MONASH UNIVERSITY
THESIS ACCEPTED IN SATISFACTION OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
ON..... 22 December 2003 .....
.....[redacted].....
Sec. Research Graduate School Committee

Errata

p 7 para 5, line 2: "[6]" for "[5]"

p 159 section title 7.3.2: "Metal-Oxide Varistor (MOV)" for "Varistor (MOV)"

p 186 para 2: Comment: The nominal low voltage line to line value is 415 V in Australia

p 212 figure caption 9.19(b): "Supply" for "LSupply"

p 212 figure 9.19: Replace entire caption with:
" Test 1 – Experimental (a) load current, and (b) active filter and source current waveforms. "

p 262 para 2, line 3: "aging" for "ageing"

Design and Control of a Universal Custom Power Conditioner (UCPC)

A thesis submitted in accordance with the regulations of Monash University in fulfillment of the requirements for the degree of Doctor of Philosophy

by

Michael John Newman

BSc, Monash University, 1998
BE(Hons), Monash University, 2000

Department of Electrical and Computer Systems Engineering
Monash University
Victoria, Australia

September 2003

Dedicated to my parents,
Lorraine and Adrian

Contents

List of Figures

List of Tables

Abstract

The cost to society due to poor electrical power quality takes the form of physical damage, nuisance, and financial consequences. In recent years the dramatic cost reduction of higher power static devices has enabled the use of power electronics to become a financially viable solution to rectifying poor power quality in areas of most need. This range of applications is known as Custom Power.

This thesis presents the design and control processes for a Custom Power application using the combined series-shunt topology, and is called the Universal Custom Power Conditioner, or UCPC. The series-shunt back-to-back power converter topology offers the flexibility to simultaneously regulate both load voltage and supply current. However, the ability of the device to compensate for Power Quality problems is dependant on the rating of the two power converters and the performance of the control systems.

It is shown that by rating the UCPC to only half the load power requirements, the device is still able to compensate for the majority of Power Quality events, without the need for large energy storage capabilities. For higher power systems the UCPC may therefore be a more cost effective option compared to systems which require full per unit ratings and large energy storage capabilities (which also leads to large maintenance costs).

Due to the strict transient and steady-state performance requirements that are necessary for adequate compensation of Power Quality events, the converter control systems are key to the performance of the UCPC, and are therefore a primary focus of this work. The aspects of control considered are: the primary control systems for both the series and shunt power converters, the digital control aspects, and the control of the series converter protection system.

The proposed control system allows the series section of the UCPC to compensate for deviations in the load voltage, including fundamental magnitude, harmonics, unbalance, sags, swells, phase jump, and flicker. The shunt section of the UCPC compensates for harmonics and unbalance in the supply current, and also controls the power flow to the common dc-bus. The series control system applies stationary frame P+Resonant controllers to this topology for the first

time, to provide improved steady-state compensation compared to existing systems, and to allow for both three-phase and single-phase operation whilst keeping the practical computational overheads to a minimum. This stationary frame form of controller is applied to the fundamental, 5^{th}, 7^{th} 9^{th}, 11^{th} and 13^{th} voltage harmonics, and is used in combination with feed-forward and active damping controllers to maintain good transient performance and stability, respectively. For the shunt controller, a deadbeat current regulation scheme is used. However, to create the reference signal for this controller a new stationary frame reference generation system was developed that provides improved computational advantages (using the same methodology as was used for the development of the P+Resonant controller), and is applicable to single-phase systems (unlike the synchronous *d-q* frame equivalent).

This thesis also investigates the digital implementation of the proposed systems and identifies the use of delta operator based Infinite Impulse Response (IIR) digital filters for use with real-time control of power converters. This provides a solution to the known problems associated with the digital implementation of stationary frame linear controllers on 16-bit fixed-point processors.

The control of the protection systems for the UCPC is the final control aspect investigated. An integrated protection system for the series topology is proposed to ensure a continuous, appropriately rated, current path under fault conditions, as well as start-up, stand-by, shut-down and recovery from faults.

The UCPC and all of the related theoretical concepts and control systems have been extensively verified in simulation, low voltage experimental work, and in a medium voltage experimental prototype of the series component.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other institution, and to the best of the author's knowledge, contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Michael John Newman

Acknowledgements

The last few years of working on a PhD has opened up my perspective on life, has allowed me to travel more, and given me the opportunity to meet new people who have also contributed towards this work in many ways. I would firstly like to thank my supervisor Assoc. Prof. Grahame Holmes. It has been a privilege to work with you and the close knit research group you have spent so much time to develop. It is hard to imagine a PhD without a wrong path here and there, but if I could make my choice of supervisor again, I would choose you again as the one to help guide me down this blind path reserved only for PhD students.

To my office mates, Dr. Daniel Zmood and Dr. Brendan McGrath, thank-you for keeping me on track, the insightful group discussions, and your friendship. Thank-you to Andrew McIver who was there without fail to answer my hardware and software questions; you definitely have 'the knack'. To Pat McGoldrick, I give thanks for your friendship, technical support, and help with all the little bits that get things done. Thank-you to Martin Lizner for help with the metal work, and to Linda, Pat, Erika, Brendan and Dan, who took the time to proof-read this thesis.

A special thank-you goes to Prof. Frede Blaabjerg of Aalborg University, Denmark. Your trust in letting me work with your medium voltage experimental system is truly appreciated. Thank-you also for your continued support long after my departure from Denmark. It has been an honor and pleasure to work both with you and your research group. For rebuilding the system I thank its creator, Dr. John Godsk Nielsen, who took time off work to help me get started. Thanks to Assoc. Prof. Hans Nielsen, and the technical staff, Walter and Kim, for their help. To Gert, Christian, Christinel and Remus, thank-you for making me feel at home in Aalborg.

Last, but not least, I would like to thank my family. To my parents, Lorraine and Adrian, I dedicate this thesis. I am indebted for the unconditional support and encouragement you have given throughout my entire education. Thank-you to my beautiful, caring and intelligent fiance, Debbie. I love you. Some questioned the sanity of us working simultaneously on our PhDs, but doing so has allowed us to grow closer and identify with each other's daily PhD woes. With this chapter of our lives coming to an end, we can now start our next adventure in life.

Publications

Several parts of the work and ideas presented in this thesis have been published by the author during the course of the research work. These publications are listed below.

IEEE Transactions

[1] **M.J. Newman** and D.G. Holmes, "An Integrated Approach for the Protection of Series Injection Inverters," *IEEE Trans. on Ind. Applicat.*, vol. 38, no. 3, May/June, pp. 679-687, 2002.

[2] **M.J. Newman**, D.N. Zmood, and D.G. Holmes, "Stationary Frame Harmonic Reference Generation for Active Filter Systems," *IEEE Trans. on Ind. Applicat.*, vol. 38, no. 6, Nov./Dec., pp. 1591-1599, 2002.

[3] **M.J. Newman** and D.G. Holmes, "Delta Operator Digital Filters for High Performance Inverter Applications," *IEEE Trans. on Pow. Elect.*, vol. 18, no. 1, Jan., pp. 447-454, 2003.

[4] **M.J. Newman**, D.G. Holmes, J.G. Nielsen, and F. Blaabjerg, "A Dynamic Voltage Restorer (DVR) with Selective Harmonic Compensation at Medium Voltage Level," *IEEE Trans. on Ind. Applicat.* (In Review)

[5] P.C. Loh, **M.J. Newman**, D.N. Zmood, and D.G. Holmes, "Improved Transient and Steady State Voltage Regulation for Single and Three Phase Uninterruptible Power Supplies," *IEEE Trans. on Pow. Elect.*, vol. 18, no. 5, Sept., 2003.

[6] J.G. Nielsen, **M.J. Newman**, H.N. Nielsen, and F. Blaabjerg, "Control and Testing of a Dynamic Voltage Restorer at Medium Voltage Level," *IEEE Trans. on Pow. Elect.* (Accepted for Publication)

IEEE Conference Proceedings

[7] **M.J. Newman** and D.G. Holmes, "An Integrated Approach for the Protection of Series Injection Inverters," *in Conf. Rec. IEEE/IAS Annual Meeting*, Chicago, IL, USA, pp. 781-788, 2001.

[8] **M.J. Newman**, D.N. Zmood, and D.G. Holmes, "Stationary Frame Harmonic Reference Generation for Active Filter Systems," *in Conf. Rec. IEEE/APEC*, Dallas, TX, pp. 1054-1060, 2002.

[9] **M.J. Newman** and D.G. Holmes, "Delta Operator Digital Filters for High Performance Inverter Applications," *in Conf. Rec. IEEE/PESC*, Cairns, Qld, Australia, pp. 1407-1412, 2002.

[10] **M.J. Newman** and D.G. Holmes, "A Universal Custom Power Conditioner (UCPC) with Selective Harmonic Compensation," *in Conf. Rec. IEEE/IECON*, Seville, Spain, Nov. 2002.

[11] **M.J. Newman**, D.G. Holmes, J.G. Nielsen, and F. Blaabjerg, "A Dynamic Voltage Restorer (DVR) with Selective Harmonic Compensation at Medium Voltage Level," *in Conf. Rec. IEEE/IAS*, Salt Lake City, UT, 2003. (In Print)

[12] P.C. Loh, **M.J. Newman**, D.N. Zmood, and D.G. Holmes, "Improved Transient and Steady State Voltage Regulation for Single and Three Phase Uninterruptible Power Supplies," *in Conf. Rec. IEEE/PESC*, Vancouver, BC, Canada, pp. 498-503, 2001.

[13] J.G. Nielsen, **M.J. Newman**, H.N. Nielsen, and F. Blaabjerg, "Control and Testing of a Dynamic Voltage Restorer at Medium Voltage Level," *in Conf. Rec. IEEE/PESC Annual Meeting*, Acapulco, Mexico, 2003.

[14] G.H. Bode, **M.J. Newman**, D.G. Holmes, "Design and Analysis of Robust Predictive Current Regulated Algorithms," *in Conf. Rec. IEEE/PEDS*, Singapore, 2003. (In Print)

[15] E. Twining, **M.J. Newman**, P.C. Loh, D.G. Holmes, "Voltage compensation in weak distribution networks using a D-Statcom", *in Conf. Rec. IEEE/PEDS*, Singapore, 2003. (In Print)

Glossary of Terms

A/D	Analog to Digital conversion
AF	Active Filter
ANSI	American National Standards Institute
APLC	Active Power Line Conditioner
BJT	Bipolar Junction Transistor
CT	Current Transformer
CRO	Cathode Ray Oscilloscope
DC	Direct Current
DF	Direct Form digital filter structure (i.e. DFI, DFIt, DFII, DFIIt)
DFT	Discrete Fourier Transform
DSP	Digital Signal Processor
DVR	Dynamic Voltage Restorer
FACTS	Flexible AC Transmission System
FIR	Finite Impulse Response
FWL	Finite Word Length
HP	High Pass
HV	High Voltage System (> 66 kV)
IEC	International Electrotechnical Commission
IEE	The Institution of Electrical Engineers
IEEE	The Institute of Electrical and Electronics Engineers
IIR	Infinite Impulse Response
IGBT	Insulated Gate Bipolar Transistor
ITIC	Information Technology Industry Council
LC	Inductor-Capacitor 2^{nd} order filter
LV	Low Voltage System (110 V - 415 V)

LVRC	Line Voltage Regulator/Conditioner
MATLAB	Numerical Analysis Program
MUPC	Multilevel Universal Power Conditioner
MV	Medium Voltage System (6 kV - 66 kV)
P+R	Proportional plus Resonant Control (i.e. P+Resonant)
PAF	Shunt/Parallel Active Filter
PCC	Point of Common Coupling
PCR	Predictive Current Regulation
PDPINT	Power Drive Protection Interrupt
PI	Proportional Integral Control
PLL	Phase Lock Loop
PLC	Power Line Conditioner
PM	Phase Margin
PWM	Pulse Width Modulation
SAF	Series Active Filter
SEMI	Semiconductor Equipment and Materials International
SRF	Synchronous Reference Frame
StatRF	Stationary Reference Frame
STATCOM	Static Synchronous Compensator
SVC	Static Var Compensator
TCR	Thyristor Controlled Reactor
THD	Total Harmonic Distortion
UCPC	Universal Custom Power Conditioner
UPFC	Universal Power Flow Controller
UPLC	Universal Active Power Line Conditioner
UPQC	Unified Power Quality Conditioner
UPQM	Universal Power Quality Manager
UPS	Uninterruptable Power Supply
USSC	Universal Series Shunt Conditioner
VSI	Voltage Source Inverter

List of Symbols Used

Lowercase Greek

α	First phase quality in the orthogonal α-β reference frame
α_k	Delta IIR digital filter denominator coefficients
β	Second phase quality in the orthogonal α-β reference frame
β_k	Delta IIR digital filter numerator coefficients
γ	Gamma (discrete version of the Laplace 's')
δ	Delta operator
δ^{-1}	Inverse delta operator
τ_{Delay}	Delay in control system loop due to digitization
ϕ_L	Phase angle of the load current (I_L)
ϕ_n	Phase offset for the n^{th} harmonic controller
ϕ_{SAF}	Phase angle of series injected voltage
ϕ_{SAG}	Voltage sag phase jump angle
ω_0	Fundamental frequency (rad/sec)
ω_c	Cut-off frequency (rad/sec)
ω_n	n^{th} harmonic frequency (rad/sec)

Uppercase Greek

Δ	Sample time / free design variable
ΔT	Switching period time (i.e. half full switching cycle for asymmetrical PWM)

Lowercase

a	First phase of a three-phase system (sometimes denoted as the red phase)
a_k	Shift IIR digital filter denominator coefficients
b	Second phase of a three-phase system (sometimes denoted as the white phase)
b_k	Shift IIR digital filter numerator coefficients
c	Third phase of a three-phase system (sometimes denoted as the blue phase)
f_0	Fundamental frequency (Hz)
f_{break}	Break point frequency of the LC filter
f_c	Cut-off frequency (Hz)
f_n	nth harmonic frequency (Hz)
f_s	Sample frequency of the digital controller
f_{sw}	Switching frequency of the PWM
$h_{dq}(t)$	Analog time-domain filter function referred to the d-q reference frame
n	Harmonic number (i.e. n^{th} harmonic). Mainly used in subscripts.
q	Shift operator
t_{SAG}	Voltage sag time duration (seconds)
$t_{SAG,Max}$	Maximum sag compensation duration by the UCPC for given conditions
$w[t]$	Windowing function for FIR digital filter creation
$x[k]$	Discrete input sequence
$x_\alpha(t)$, $x_\beta(t)$	α & β phase input variables (α-β reference frame)
$x_a(t)$, $x_b(t)$	a & b phase input variables (abc reference frame)
$y[k]$	Discrete output sequence
$y_\alpha(t)$, $y_\beta(t)$	α & β phase output variables (α-β reference frame)
$y_a(t)$, $y_b(t)$	a & b phase output variables (abc reference frame)

Uppercase

$C(s)$	Target output parameter for a control system
C_{DC}	Capacitance located across the dc-bus
C_F	Filter capacitance ($_X$=$_{SAF}$ for series inverter, and $_X$=$_{PAF}$ for parallel inverter)
$D(s)$	Disturbance input for a control system
$E(s)$	Error term for a control system
$G_1(s)$	Feed-back linear series controller
$G_2(s)$	s-domain model of the series system plant
$[\mathbf{H}_{ab}(s)]$	2 x 2 Matrix of Analog cross coupled filters between the a & b phases
$H_{dq}(s)$	Analog s-domain filter function referred to the d-q reference frame
$H_\delta(\gamma)$	Delta operator (δ) based Infinite Impulse Response (IIR) digital filter
$H_q(z)$	Shift operator (q) based Infinite Impulse Response (IIR) digital filter
I_L	Load current
$I_{L,SAG}$	Load current during a voltage sag (only applies for In-phase compensation)
I_{PAF}	Output current of the shunt (parallel) inverter
I_{PAF}*	Demanded output current of the shunt (parallel) inverter
I_{Trip}	Minimum current in the series clamping varistor required to trip the protection
K_{AD}	Series active damping gain constant
K_I	Integral gain constant
$K_{I,n}$	Integral gain constant for the nth harmonic
K_P	Proportional gain constant
$L_{F,X}$	Filter inductance ($_X$=$_{SAF}$ for series inverter, and $_X$=$_{PAF}$ for parallel inverter)
L_{Load}	Load inductance
L_S	Source inductance
N	Series transformer ratio
P_{av}	Average power dissipation
$P_{DC,NET}$	Net real power flow *out* of the dc-bus energy storage
P_{PAF}	Real power flow *out* of the shunt (parallel) converter
P_{SAF}	Real power flow *out* of the series converter
Q_{PAF}	Reactive power flow *out* of the shunt (parallel) converter
Q_{SAF}	Reactive power flow *out* of the series converter

$R(s)$	Reference input to a control system
R_{CF}	Resistance in series with the filter capacitance, C_F
R_{LF}	Internal series resistance of the filter inductor, L_F
R_{Load}	Load resistance
R_S	Source resistance
T	Discrete sample time constant (sec)
V_{DC}	Voltage of the common dc-bus
V_L	Load voltage
$V_{PAF}{}^*$	Demanded voltage by the shunt PCR controller for the PWM generator
V_{Ref}	Reference load voltage for the series controller
V_S	Source voltage (i.e. before source impedance)
$V_{SAF}{}^*$	Demanded voltage by the series control system for the PWM generator
$V_{SAF,Normalised}{}^*$	$V_{SAF}{}^*$ normalized to a modulation depth using the measured value of V_{DC}
V_T	Terminal supply voltage (i.e. after source impedance)
$V_{T,SAG}$	Terminal supply voltage during a voltage sag event
V_{Trip}	Peak ac voltage at which the series protection will trip
$X_\alpha(s)$, $X_\beta(s)$	α & β phase input variables (α-β reference frame)
$X_a(s)$, $X_b(s)$	a & b phase input variables (abc reference frame)
$Y_\alpha(s)$, $Y_\beta(s)$	α & β phase output variables (α-β reference frame)
$Y_a(s)$, $Y_b(s)$	a & b phase output variables (abc reference frame)

Chapter 1

Introduction

1.1 Background

The quality of electrical power supplied to consumers (often termed "Power Quality") has become an increasingly important topic over the last two decades. Electrical supply quality problems can lead to either failure or misoperation of consumer equipment. The consequences of lost production time, additional maintenance, and equipment replacement because of Power Quality problems are estimated to cost billions of dollars each year. Ironically, the increasing use of power electronics in part contributes to Power Quality problems. The distorted currents drawn by consumers with large non-linear power electronic loads are a major cause of the voltage distortions seen both by themselves and by nearby consumers. These distorted currents can also create safety concerns due to overheating of supply cables.

However, the increased use of power electronic systems has also led to a dramatic drop in the cost of, and a large increase in both voltage and current ratings of, the static silicon devices which are the core of this technology. As a result, the use of power electronics has also become a viable solution for the resolution of Power Quality issues. These power electronic solutions are often defined as "Custom Power" applications.

Custom Power applications can be targeted to solve a specific Power Quality condition, or they can compensate for a wide range of Power Quality problems. The suitability of the device for a given Power Quality problem depends on the topology used, the power ratings of the components, the storage capacity (if any), and also the implemented control strategy. Most converter based Custom Power applications use a variation of either a shunt or a series converter topology – or sometimes even both. The shunt topology is currently the more commercially available alternative, and is commonly used for the active compensation of current waveform distortion.

The series topology has become popular in the literature to improve the performance of cheap passive harmonic filters using low rating active devices. For compensation of voltage sags, and full supply interruptions, the Uninterruptible Power Supply (UPS) is a common example of a Custom Power application, especially in its common low power form to protect personal computers. However, for higher power systems the energy storage constraints can become expensive partly due to the capital cost, but more importantly, because of the recurring maintenance costs. For these systems, the use of a partially rated series converter with minimal energy storage has been found to be an attractive alternative. This device only compensates for some sags (and not full interruptions), but since the majority of sags are within the range of this cheaper device, the coverage of voltage events is not significantly altered. This approach has created a recent increase in research into Custom Power applications using higher rated series converters.

Combined series-shunt topologies have also attracted recent interest because the available degrees of freedom give the topology greater flexibility. Many of these reported combinations also include tuned passive harmonic filters. However, as combined series-shunt systems are proposed to provide compensation for more extreme Power Quality issues such as sags, the ratings of the active components have been increased from the originally small values. Interestingly however, tuned passive components still remain in many proposed schemes, even though their usefulness becomes unclear, as the ratings of the active components for these systems are increased to perform the primary task. Unfortunately, because of the difficulties of constructing experimental series based topologies (due to protection and other issues), many proposed series-shunt schemes have only been verified in simulation. This means that possible inadequacies in their performance of harmonic voltage compensation have not been explored, even though a review of the proposed controllers shows that practical limitations will most likely exist. Finally, many of these systems use three-phase systems and do not easily translate to single-phase systems.

Hence, further research is needed into the development of Custom Power applications that do not have the full rating and energy storage capability of a UPS, but are capable of simultaneously compensating for a high portion of Power Quality conditions.

1.2 Aim of the Research

The aim of this research is to investigate the use of active series-shunt converters to develop a single Custom Power system that is capable of improving the quality of power for either customer premises or distribution systems. The system should be capable of simultaneously compensating for voltage/current harmonics, fundamental voltage error, voltage sags and swells, voltage flicker and voltage/current unbalance. The implementation should be applicable to both three-phase and single-phase installations, and should ideally utilize a fully digital controller to allow upgradability, flexibility and intolerance to component variations with age.

Control systems are key to the performance of almost all Custom Power systems, and will therefore be a primary focus of this work. This work investigates the use of stationary frame control techniques to achieve the research aims, and to provide fast and accurate control of the voltage and current outputs of the series-shunt converters. The work also broadens the classical definition of control to include the digital signal processing aspects (i.e. digital control), as well as the control of the protection systems for the proposed Custom Power application.

1.3 Research Approach

The arrangement of the research approach described in this thesis is shown in Figure 1.1. The initial chapters provide a review and an overview of Custom Power structures, including in particular the series-shunt topology used in this research. The next chapters then focus on various control aspects, including the stationary frame linear control systems, digital control implementation, and the integrated control of the converter protection systems. Finally, simulations as well as low voltage and medium voltage experimental verification are presented to conclude the work.

Chapter 2 provides a review of current literature for both Power Quality and Custom Power (i.e. the problem and existing solutions, respectively). Firstly, a brief review of the problem of poor Power Quality is presented, as this is the core problem to be addressed by the research program. Secondly, existing Custom Power applications in the literature are presented to review the current state of the art, as well as to develop a background for the design choices discussed in later chapters. This thesis uses a distributed literature review structure to maintain the flow of the work. Therefore, detailed reviews on stationary frame control, digital control, and series protection, are included at the beginning of the relevant chapters, rather than being consolidated in an early chapter as is more conventional.

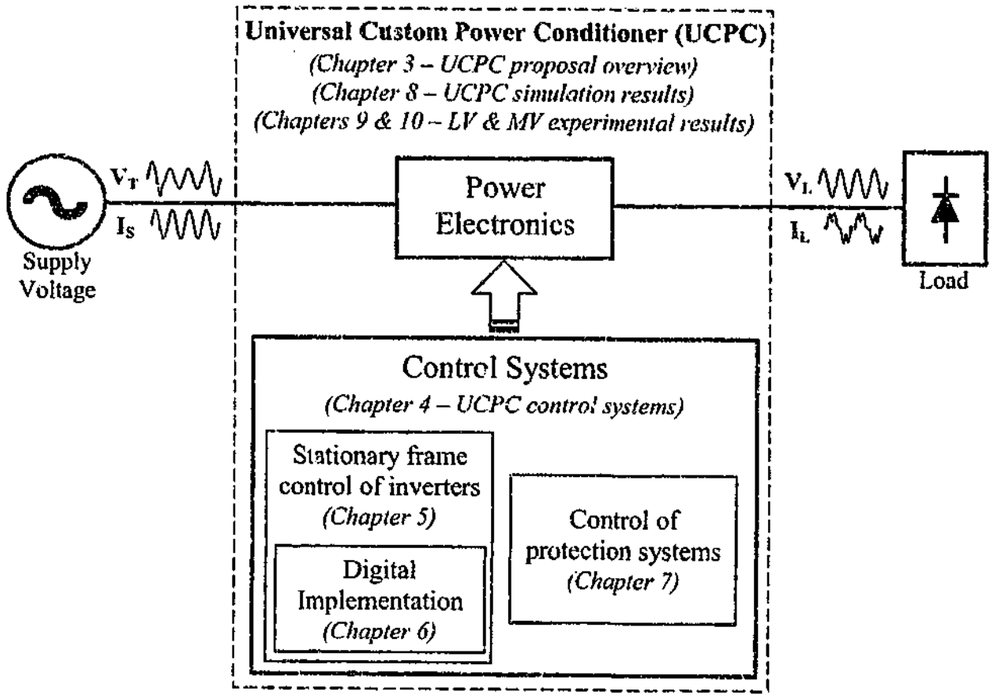


Figure 1.1: Thesis structure in relation to the Universal Custom Power Conditioner (UCPC).

Chapter 3 provides an overview of the proposed Universal Custom Power Conditioner (UCPC) system, and includes discussion of the following attributes of the system: objectives, control parameter options, physical components, ratings, injection limitations, and preferred placement of the device in a distribution network.

Chapter 4 presents a control scheme for the UCPC. This scheme aims to provide the combined capabilities of previously presented systems, without the need for additional tuned passive harmonic filters, and with improved voltage harmonic compensation capability. The predicted performance of the control scheme is discussed, as well as the mechanisms that will allow the device to compensate for power quality problems in distribution systems and consumer's premises. These include: voltage/current harmonics, voltage sag and swell compensation, flicker, voltage/current unbalance, and fundamental voltage regulation. However, while a conventional implementation of such a control scheme would use traditional *d-q* reference frame controllers, this approach is shown to have practical limitations due to computational overheads and lack of applicability to single-phase systems.

Chapter 5 develops the use of stationary frame linear control techniques with the UCPC, to solve the computational and single-phase limitations of the control scheme presented in Chapter 4. A short review of the current state of the art in stationary frame ac control is presented, particularly focusing on the P+Resonant linear controller and its development from the equivalent PI *d-q* controller. The use and design of the P+Resonant controller for selective voltage harmonic compensation in the series controller is then investigated to allow feasible implementation of multiple harmonic controllers, as well as the possibility of single-phase operation. An active damping scheme is also proposed to damp the filter resonance conditions, and to improve the system stability. The review of the P+Resonant controller also leads to the proposal of a new harmonic signal reference extraction system that is used to replace the control block proposed for the shunt component of the UCPC in Chapter 4. The new control block is a stationary frame equivalent to the commonly used high-pass filter placed in the *d-q* synchronous reference frame, and both three-phase and single-phase versions are presented. Development of the control block is given, as well as some discussion of its advantages and disadvantages (particularly with respect to practical implementations).

Chapter 6 investigates the digital implementation of the linear controllers developed in Chapter 5. The work reviews the practical problems of using signal processing techniques with digital control. Particular focus is given to the conventional techniques used for fixed-point Digital Signal Processor (DSP) implementation of the stationary frame controllers discussed (and developed) in Chapter 5. The conventional Infinite Impulse Response (IIR) and Finite Impulse Response (FIR) methods are reviewed for their suitability to the stationary frame controllers used in this research. The use of delta operator based IIR filters in high performance converter applications, such as the UCPC (Chapter 3), is then proposed. This method alleviates the problems found in Chapter 5, when implementing the stationary frame control schemes with traditional shift operator IIR filters. This chapter also provides a concise overview of the delta operator, and shows how the transition from shift to delta filters can be easily achieved. The approach is initially verified using detailed simulations, with some experimental results included.

Chapter 7 addresses the appropriate design and control of the protection mechanisms of the UCPC. Whilst protection of shunt converters is considered general knowledge for Power Electronic engineers, protection of series converters is quite different, and literature addressing this problem is scarce. The problems and inadequacies of existing series protection schemes are discussed and used as a base to propose an integrated protection scheme between the main DSP controller software and the series protection elements. Both simulation and experiment are used to verify the proposed control of the protection elements during fault conditions and normal

operation.

Chapter 8 details the different levels of simulations used in the design and verification of the UCPC. Simulation results of the full UCPC application are presented to verify the proposed system, in a theoretical sense.

Chapter 9 presents low voltage (LV) experimental verification of the UCPC. The chapter describes the LV experimental set-up for the UCPC, including the development of both the power converter and the controller boards. Experimental results are then presented for a wide range of conditions to verify the proposed control scheme.

Chapter 10 presents medium voltage (MV) experimental verification of the proposed series harmonic compensation scheme. This work was completed in Denmark in collaboration with Aalborg University, and used a Dynamic Voltage Restorer (DVR) prototype connected to a 10 kV experimental test grid. An overview of the system and test setup is provided, followed by experimental test results.

Chapter 11 concludes the thesis and offers recommendations for future work in this field of research.

1.4 Identification of Original Contribution

The work presented in this thesis includes both theoretical and applied research in the field of power electronics. In particular, the proposed Universal Custom Power Conditioner development is application level research, while the control aspects provide a more theoretical contribution. For clarity, it is therefore important to identify the original contributions reported in this thesis in both these areas.

The first contribution is the development (Chapters 3 and 5) and verification (Chapters 9 and 10) of a Custom Power application capable of simultaneously compensating for voltage/current harmonics, fundamental voltage error, voltage sags and swells, voltage flicker and voltage/current unbalance. The system provides improved steady-state voltage compensation performance compared to that previously reported for the series-shunt topology, without compromising the transient performance of the system. The new controller for the fully active series-shunt system can be applied to both single- and three-phase systems, and uses a fully digital implementation.

The second contribution is the development of a stationary frame control block for generation of a harmonic/unbalance current reference for the shunt portion of the UCPC (Chapter 5). This work extends the theory proposed by Zmood et al. [16] to develop a stationary frame equivalent

of the rotating d-q frame high-pass filter system that has previously been used. The new scheme has significant computational advantages, does not require a sine table, and can be used for single-phase implementations with equivalent performance to a three-phase system.

The third contribution adapts delta operator based Infinite Impulse Response (IIR) digital filters to real-time control of power electronic converters. This development provides a solution to the known problems [17] associated with the digital implementation of stationary frame linear controllers on 16-bit fixed-point processors.

The fourth contribution is the development of an integrated protection system for the series converter topology (Chapter 7). The control of the protection system provides a complete solution, with the integration of both software and hardware components to ensure a continuous current path under fault conditions, start-up, stand-by, shut-down, and recovery from faults, as well as providing redundancy in the protection system. The scheme is experimentally verified under short-circuit fault current conditions.

The final contribution is at a more practical level. The literature review (Chapter 2) shows that the majority of reported series-shunt power quality conditioners have only been verified in simulation. The UCPC described in this thesis has been constructed and fully experimentally verified. The series harmonic compensation controller has also been experimentally verified on a medium voltage (10 kV) prototype system (Chapter 10). To the author's knowledge, this work is the first reported experimental system achieving series active voltage harmonic compensation at a medium voltage level.

The majority of the material presented in this thesis has been published in both IEEE Journals [1]-[5] and conference proceedings [7]-[15], and these publications are noted at the start of the relevant chapters.

Chapter 2

Background Review

[1]The primary theme of this research is to explore the development of a fully active Custom Power compensation system that will improve a broad range of Power Quality phenomena. This chapter reviews existing Custom Power systems that have been developed in recent years, concentrating on systems that use a series based converter topology, since this topology is the focus of the research work presented in this thesis. A brief review of Power Quality phenomena is firstly presented to outline the compensation performance expected from the application under investigation. (A more detailed Power Quality review is provided in Appendix A.) The relationship between Power Quality (the problem), and Custom Power (the solution) is presented in Figure 2.1, which shows how an ideal Custom Power device acts as a bi-directional Power Quality buffer between the electrical supply and the consumer.

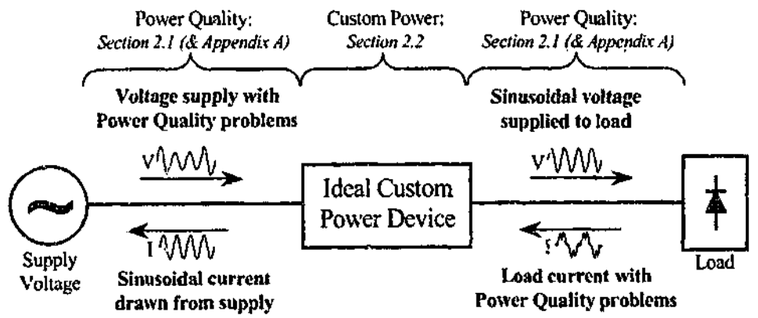


Figure 2.1: Power Quality compensation example for an ideal Custom Power device.

[1]Note that a distributed literature review format is used in this thesis to maintain the flow of the text. Therefore, further literature reviews relating to the control contributions in linear control, digital control, and control protection system, can be found at the beginning of their respective chapters later in this thesis.

2.1 Power Quality: The Problem

There are many definitions of the terms *"Power Quality"* (and *"Quality of Supply"*); and the research area is littered with vague, differing and misused terminology. This brief review of Power Quality (along with the more detailed review in Appendix A) provides background information to form a foundation for the Custom Power application proposed in this thesis, and also defines the terminology used in this thesis to ensure the clarity of the later discussions.

Perhaps the most general and appropriate definition for Power Quality has been proposed by Dugan et al.[18]:

> *A Power Quality problem is "any power problem manifested in voltage, current, or frequency deviations that results in failure or misoperation of customer equipment."*

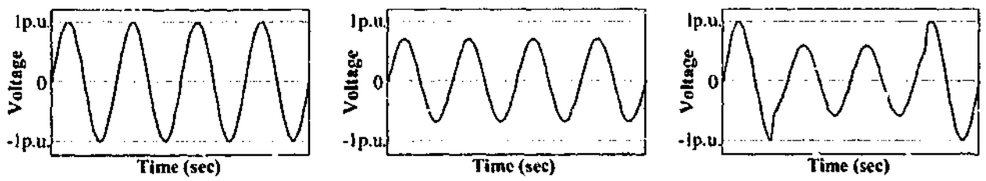
Power Quality issues such as waveform distortion, transients, voltage flicker, unbalance (also imbalance[18]), and voltage variations affect the consumer and supplier in three ways: damage, nuisance and economics. Damage may come from extra heating and fire in conductors due to harmonics, over-voltages outside the rated values of installed components, motor vibration, and other sources. Nuisance includes light flicker, tripping of microprocessor based appliances, communication noise (including telephones and T.V.s), and added audible noise from components such as transformers. Economic impacts arise from the replacement of damaged items (including components with reduced life spans due to Power Quality issues), lost production time (due to supply outages/sags), errors in the measurement of power flow used by the power distribution companies to charge their clients, to name a few. This financial cost to the consumer and/or supplier, as well as mandatory Power Quality standards, are the major driving force for the purchase and installation of Custom Power solutions.

2.1.1 Terminology and Classification

Many variations of terminology exist for Power Quality phenomena. The terminology and classification provided in Table 2.1 [18] has been selected for this thesis. The definitions used in these classifications are based on the IEEE (The Institute of Electrical and Electronics Engineers) and other U.S.A. based standards organizations' publications [18]-[21]. Further terminology clarification is required for the description of the voltage levels. For this research work, Low Voltage (LV) is used for distribution systems and consumer reticulation between 110 V and 415 V, Medium Voltage (MV) is used for distribution voltages between 6.6 kV and 22 kV, and

Classification Type	Classification Sub-Type
Type 1: Transients	Impulsive transient
	Oscillatory transient
Type 2:. Long-duration voltage variations	Over-voltage
	Under-voltage
	Sustained interruptions
Type 3: Short-duration voltage variations	Interruption
	Sags (dips)
	Swells
Type 4: Voltage unbalance	
Type 5: Waveform distortion	dc offset
	Harmonics
	Interhamonics
	Notching
	Noise
Type 6: Voltage fluctuations	Voltage flicker
Type 7: Power frequency variations	

Table 2.1: Catagories of power system Power Quality phenomena.



Nominal voltage value | Type 2 example | Type 3 example

Type 4 example | Type 5 example | Type 6 example

Figure 2.2: Example waveforms of type 2 to type 6 Power Quality problems.

High Voltage (HV) is used for 66 kV and higher for distribution and transmission systems. The research focuses only on LV and MV *radial* installations.

The Power Quality phenomena targeted for compensation by this research are types 2 through 6, and examples of each type are illustrated in Figure 2.2. Type 1 events (such as voltage spike transients) are not included as they are generally faster than the bandwidth of the power electronic solutions discussed in this research, and are usually clamped using varistors, or other similar methods. Power system frequency variations (type 7) are also not compensated, as the influence of Custom Power applications on the grid is not significant enough to be able to vary this quantity (unless the chosen topology fully decouples the supply and load – which is not the case in this research).

Short-Duration and Long-Duration Voltage Variations (Types 2 and 3)

Long and short duration voltage variations (types 2 and 3) include over-voltage, under-voltage, interruptions, sags[2], and swells. These voltage variations have more obvious financial implications to most consumers, and are therefore likely to provide the major cost justification for installing a Custom Power solution such as the one presented in this thesis.

The size of these voltage variations from the nominal value is typically much larger than variations expected from Power Quality types 4 through 6. Therefore, types 2 and 3 are also used as the basis for the converter ratings of the Custom Power application developed in this research. The voltage variation problem may be viewed from three perspectives: the susceptibility of equipment to such variations; the probability of these variations occurring; and the compatibility of equipment with its installation, based on the first two perspectives. This view creates an aid for the development of the optimum ratings required to provide the most cost effective Custom Power application to the consumer. An expanded review is included in Appendix A.

Voltage Unbalance (Type 4)

Voltage unbalance is where the fundamental voltage components in a three-phase system do not have the same magnitude and/or are not exactly 120^o displaced. The three primary ways of describing unbalance numerically are: 'negative sequence unbalance', 'magnitude unbalance' and 'phase unbalance' [22] (Appendix A).

[2]Note: The terms 'sag' and 'dip' are interchangeable (dip is used in the IEC standards), and only the term 'sag' will be used hereafter. Sag values will always be specified as a ratio of the nominal value (not the percentage drop caused by the sag). For example a sag which causes a 10% drop from the nominal voltage is noted as a 0.9 p.u. (or 90%) sag.

Unlike the instant recognition by consumers of type 2 and 3 voltage variations (when their computers and lights go out temporarily) the effects of voltage unbalance and waveform distortion are typically less obvious, although not necessarily less important. The most dominant effect of voltage unbalance is the reduction in the life of motors due to increased heating and vibrations [22] - [25], and also the reduction in developed torque [22] [23]. The Victorian Distribution Code [26] specifies a limit of 1% on the negative sequence component. However, even this 1% on an induction machine can cause a loss of life of more than 5%, and with an unbalance of 2% or 4% this loss of life increases significantly to 22% and 64%, respectively [23]. For VSDs, and other equipment using three-phase diode rectifiers with capacitive filters on the dc side, small amounts of voltage unbalance can lead to large even harmonic currents, which can cause equipment to trip because of over-current [22].

Waveform Distortion (Type 5)

Harmonic (or waveform) distortion is becoming an issue of great concern with increased use of non-linear loads. Some common examples of polluting loads are: VSDs, Switched Mode Power Supplies (SMPS) used in computers and other electronic equipment, saturation of magnetic components, arc furnaces, and lighting ballasts. Harmonics are known to have caused the malfunction of devices such as VSDs [27], capacitors [28], circuit breakers [29], fuses [29], conductors, electronic equipment, lighting, metering [30], protective relays, rotating machines [23] [25], solid state relays, static reactive power compensators [31], telephone communications [32] [33], and transformers. The causes of malfunction include: increased heating, increased dielectric stress, shifted voltage zero crossings, higher di/dt, increased magnetic fields and capacitive coupling, increased vibration, and increase susceptibility to sags due to 'flat-topping'. Additional heating is a particularly important problem as it may create safety problems with the possibility of starting fires due to overheating cables and other devices.

Voltage Fluctuation - Flicker (Type 6)

Voltage fluctuations are defined as "a series of voltage changes, or a continuous variation of the rms voltage" [34]. A common form of voltage fluctuations is 'flicker'; which, as the name suggests, causes some lights (particularly fluorescent lights) to appear to flicker. Variations as small as 0.5% in the 6-8 Hz range can cause a visible effect [18]. Flicker causes irritation to consumers working under such conditions, and can even be a trigger for illnesses such as photosensitive epilepsy. Arc furnaces are a major contributor to voltage flicker problems [18].

2.2 Custom Power: Power Electronic Solutions

In the late 1980's N.G. Hingorani [35] [36] identified that high-power electronics was reaching a stage where it was feasible to use this technology to control the transmission of power, and hence overcome problems associated with existing mechanical control based systems. This new concept was called the Flexible AC Transmission System; or FACTS . A few years later Hingorani [37] [38] noted that power electronic solutions could also perform similar roles in distribution systems; but this time mainly in the area of Power Quality. For these solutions, Hingorani coined the name *"Custom Power"*. The Custom Power philosophy proposes that existing and future products can be used as a 'value added' service by the utilities for industrial and commercial customers who require a more reliable and higher quality supply. While the original use of the term Custom Power was purely for power electronic solutions based on medium voltage distribution systems, its use has since broadened in the literature to include low voltage solutions on consumers' premises. This broader view will be used in this research[3].

It is estimated that Power Quality problems may cost tens of billions of dollars each year in the U.S.A. alone [20]. With rapidly dropping prices and increasing power levels of power electronic devices, Custom Power systems are becoming more widely used, and this trend is expected to continue.

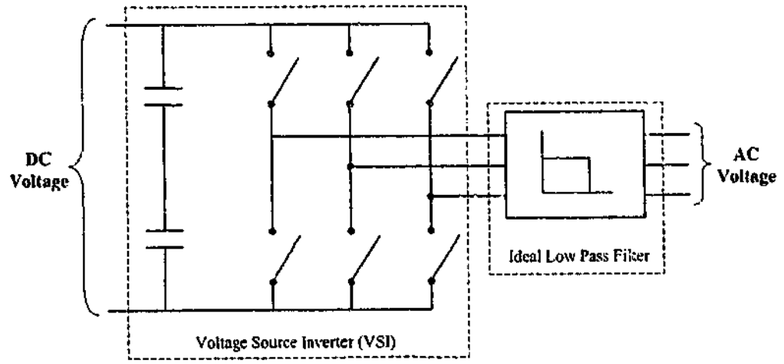
The core power electronic converter technologies of Custom Power systems are the Voltage Source Inverter (VSI) and Current Source Inverter (CSI), and hence it is appropriate to review these technologies before proceeding further.

2.2.1 Voltage Source Inverters (VSI) and Current Source Inverters (CSI)

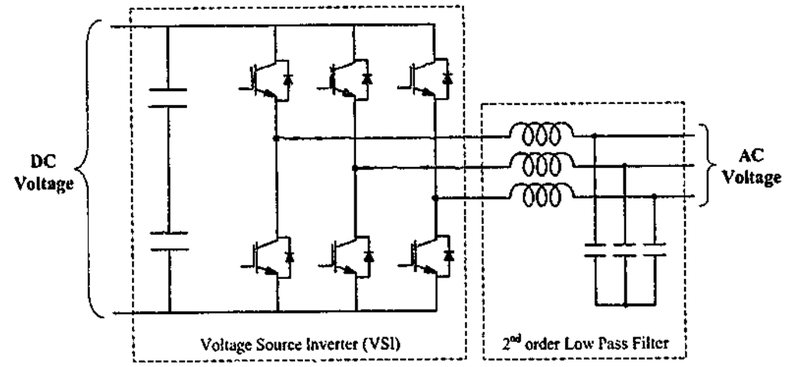
Both Voltage Source Inverters (VSI) and Current Source Inverters (CSI) have been used in the literature for the construction of active filters, with the VSI topology being reported much more extensively. Both the VSI and CSI allow for bidirectional flow. The VSI is a buck converter from the dc to ac direction, or conversely a boost converter in the ac to dc direction. The CSI is a boost converter from the dc-ac direction, or a buck converter in the ac to dc direction. Commercially, nearly all active filters are of the VSI type [39] [40], as is also the case for all other Custom Power applications.

The use of VSI and CSI topologies in Custom Power applications has been compared by Yunus and Bass [41], and by Benchaita et al. [42]. Higher efficiency and lower initial costs of the

[3]The definition of Custom Power applications also extends to other Power Quality solutions such as the static by-pass switch, but only inverter based applications are investigated in this research.



(a) Ideal Voltage Source Inverter



(b) Example of a common practical Voltage Source Inverter

Figure 2.3: Schematics of ideal and practical Voltage Source Inverters (VSI).

VSI usually make it the more popular choice. CSI converters have additional losses caused by the extra diode drop introduced, as well as the losses in the link inductor. Furthermore, under light load the current through the inductor for many applications is kept constant, thus fixing this loss regardless of the output power, and hence reducing the efficiency of the CSI topology even further. The significantly higher volume of VSIs produced commercially has also led to a higher differential in cost between the two. Because of these reasons, this research focuses only on the VSI type of converter.

Figure 2.3a shows the basic construction of an ideal VSI. It is assumed that the dc side is voltage stiff, and this is usually achieved using capacitors (as shown here), batteries, or other components in some applications. Each vertical pair of switches is known as a phase-leg, and the connection point between the switches is the output. By switching on either the upper or lower switch (never both at the same time), the phase output will connect to the positive or negative voltage rails of the dc-bus, respectively. By switching one phase leg in one direction and

another in the opposing direction, a voltage differential equal to the dc-bus voltage is achieved across the output; switching both legs in the same direction results in a zero output voltage. This provides three different voltage levels that can be generated across any two phase legs. The phase-legs are switched at high frequency, and a low pass filter (often just the load) is used to extract the required volt-second average voltage. This control strategy is known as Pulse Width Modulation, or simply PWM. The reference voltage that defines the required volt-second average is also time varying (typically a sinusoid), and hence creates the desired voltage waveform on the output.

Of course, in practice, the switches are not ideal, and the filter is typically a second order *LC* filter. A practical example using IGBTs is presented in Figure 2.3b, and is the type of converter used for this research.

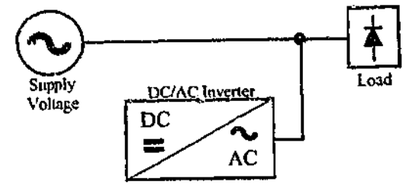
Modulation theory of power converters is now a mature field, and is taken as assumed knowledge in this research. A detailed discussion can be found in the text by Holmes and Lipo [43]. A nominal switching frequency of 5 kHz, with asymmetrically regular sampled PWM will be assumed, unless otherwise noted. Note that the nominal sample frequency of the control system will therefore be twice the switching frequency at 10kHz (i.e. due to the asymmetrical sampling)[4].

Throughout this thesis the VSI is modelled in various ways, depending on the level of detail required. For the series control model developed and used in Chapters 4, 5, and 10, the VSI is modelled as a continuous linear gain, as it is assumed to always operate in the linear region (note that the *LC* filter is included in the model separate from the VSI). For the digital control research (Chapter 6) the linear gain model is still used, but also incorporates a transport delay to account for the regular sampled process. The UCPC verification simulation in Chapter 8 models the VSI as a whole, and includes the modulation strategy, IGBT characteristics, dc-bus components and values, *LC* filter elements, and all discrete sampling effects. The limitations of the linear VSI model are discussed and investigated in the low and medium voltage experimental work in Chapters 9 and 10, respectively.

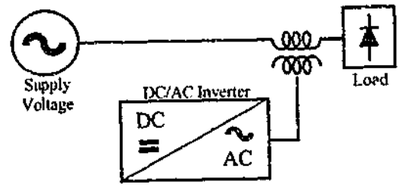
2.2.2 Custom Power Topologies

The ability of a Custom Power system to combat various Power Quality problems depends on its topology. Figure 2.4 illustrates the more common Custom Power topologies. These topologies

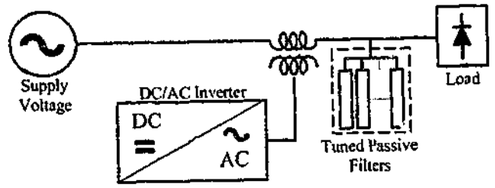
[4]For the MV DVR experimental chapter both 3 kHz and 5 kHz are tested using symmetrically sampled PWM. Therefore the sample rates for these tests match the switching frequencies.



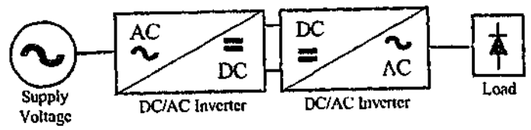
(a) Shunt (or parallel) active topology



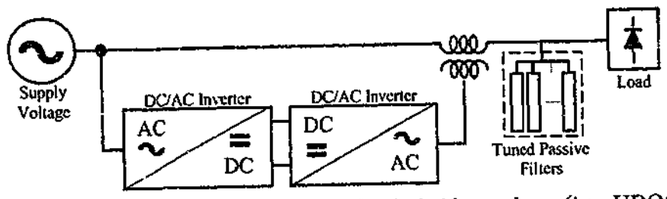
(b) Series active topology



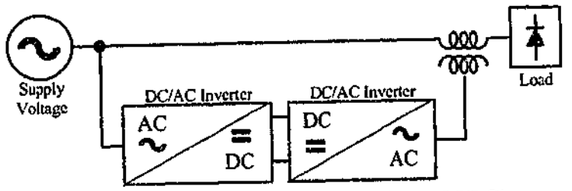
(c) Series active / shunt passive hybrid topology



(d) In-line ac-dc-ac conversion topology (i.e. Online UPS)



(e) Series-shunt active and shunt passive hybrid topology (i.e. UPQC)



(f) Series-shunt active topology (i.e. UCPC)

Figure 2.4: Fundamental Custom Power topologies.

contain either one or two converters, as well as additional tuned passive harmonic filters for some applications. It should be noted also that for some two converter topologies, the second converter may simply be a passive diode rectifier. Depending on the specific application the same topology can be used for different Custom Power systems, but with very different ratings, physical parameter values, and control objectives, their contribution to reducing Power Quality phenomena can vary significantly.

For example, the shunt topology in Figure 2.4a is used for Parallel (shunt) Active Filters (PAF), as well as for Static Compensators (STATCOM; also a FACTS device). It can be controlled directly for current injection, or indirectly for voltage stabilization/regulation. The series topology (Figure 2.4b) is used for the Series Active Filter (SAF) and for the Dynamic Voltage Restorer (DVR) using voltage feedback control. The series topology can also be used to control current, and is common in systems such as the active-passive Hybrid SAF (Figure 2.4c), where current regulation is used to isolate the passive filters from the source voltage, to improve and decouple their performance from the supply source impedance.

These topologies contain a single power electronic converter. Table 2.2 shows that most of the associated applications are only able to remove Power Quality problems from either the source current, or the load voltage – but not both. This is seen by the grey ticked boxes residing in either the source current compensation columns, or the load voltage compensation columns, but rarely in both.

A common topology for an Uninterruptible Power Supply (UPS) is the in-line topology (Figure 2.4d), with two converters, where the supply side converter may be either a passive rectifier (i.e. diode rectifier) or an active rectifier. With both converters active, this device can compensate for Power Quality problems in both the load voltage and source current. Alternatively, by extending the Hybrid SAF (discussed above) with an additional shunt converter (Figure 2.4e), load voltage control can be added to act simultaneously with the current harmonic compensation. This topology was first presented as the Universal Power Quality Conditioner (UPQC), and can also compensate for voltage harmonics and flicker. The shunt connection of the UPQC can be placed either upstream or downstream of the series injection transformer.

Removal of the passive components from the UPQC creates the unified fully active series-shunt topology (Figure 2.4f) which is the subject of this research. With two active converters this topology has enough degrees of freedom to simultaneously attenuate most types of Power Quality problems from the source current and load voltage. While there is only a small amount of literature discussing control schemes to achieve this task, prior literature from the PAF, SAF,

	Primary Author	Initial Publication Year	Device Name	Simulated Results	Experimental Results	% Series Capability	Tuned Passive Filters	Source Current Comp.		Load Voltage Compensation					
								Harmonics	Unbalance	Harmonics (due to Vs)	Steady State Unbalance	Flicker	Steady State Fundumental	Sag	Swell
Active Shunt	Sasaki	1971	None	✓	✗	N/A	✓	✓	✗	✗	✗	✗	✗	✗	✗
	Amentani	1972	None	✓	✓	N/A	✗	✓	✗	✗	✗	✗	✗	✗	✗
	Gyugyi	1976	Shunt AF	✗	✓	N/A	✗	✓	✗	✗	✗	✗	✗	✗	✗
	Akagi	1986	Shunt AF	✗	✓	N/A	✗	✓	✗	✗	✗	✗	✗	✗	✗
	Grady	1991	APLC	✓	✗	N/A	✗	✓	✗	✗	✗	✗	✗	✗	✗
	Akagi	1998	Shunt AF	✗	✓	N/A	✗	✗	✗	✓	✗	✗	✗	✗	✗
	Cheng	2000	Shunt AF	✓	✓	N/A	✓	✓	✗	✗	✗	✗	✗	✗	✗
	Mattavelli	2000	Shunt AF	✗	✓	N/A	✗	✓	✗	✗	✗	✗	✗	✗	✗
Active Series	Gyugyi	1976	Series AF	✗	✗	NS	✗	✓	✗	✗	✗	✗	✗	✗	✗
	Peng	1988	Hybrid Series AF	✓	✓	5%	✓	✓	✗	✗	✗	✗	✗	✗	✗
	Bhattacharya	1993	Hybrid Series AF	✓	✓	4%	✓	✓	✗	✗	✗	✗	✗	✗	✗
	Moran	1995	Hybrid Series AF	✓	✗	NS	✓	✓	✗	✗	✓	✗	✗	✗	✗
	Blajszczak	1995	Series AF	✗	✓	NS	✗	✗	✗	✓	✗?	✗?	✗	✗	✗
	Aredes	1996	Series AF	✓	✗	NS	✗	✗	✓	✓?	✗?	✗?	✗	✗?	✗?
	Dixon	1997	Hybrid Series AF	✓	✓	NS	✓	✓	✗	✗	✗	✗	✗	✗	✗
	Stump	1997	DVR	✓	✗	50%	✗	✗	✗	✗	✗	✓	✗	✓	✓
	Fang	1998	DVR	✗	✓(2)	30-70% (1)	✗	✗	✗	✓?	✗	✗?	✓?	✓	✓
	Cao	1999	SPQC	✓	✓	14%	✗	✗	✗	✓?	✓	✗?	✗?	✓	✓
	Woodley	1999	DVR	✗	✓	53%	✗	✗	✗	✗	✗	✗?	✗	✓	✗?
	Vilathgamuwa	1999	DVR	✓	✗	100%	✗	✗	✗	✗	✗	✗	✗	✓	✗?
	Peel	2000	DVR	✗	✓	50%	✗	✗	✗	✗	✗	✗	✗	✓	✓
	Svensson	2001	DVR	✓	✗	NS	✗	✗	✗	✓?	✗	✗	✗	✓	✗?
	Nielsen	2001	DVR	✓	✓	50%	✗	✗	✗	✗	✗	✗	✗	✓	✗?

✓? Some comp. of this type is possible, but performance not necessarily adequate (refer to discussion)
✗? This form of compensation seems to be possible, but is not quoted in the publication
NS Information Not Supplied in publication

Table 2.2: Review of *selected* Custom Power research outlining the reported Power Quality compensation capabilities of the shunt (parallel) and series topologies (including passive hybrid topologies). Note that a grey background indicates the physical and compensation capabilities of each system which are preferred by this research.

UPS, DVR and UPQC systems have some direct relevance to this role, and will therefore now be individually reviewed.

2.2.3 Active Filters (AF)

The concept of injecting harmonic currents to cancel out unwanted harmonic currents was first introduced in 1971 by Sasaki and Machida [44], and also by Ametani in 1972 [45]. Both of these were only mathematical and simulation models, but in 1976 Gyugyi introduced the terminology "Active ac Power Filters" and produced laboratory results using pulse width modulated BJT's [46]. This publication is commonly seen as the seminal paper in the field of Active Filters.

The two fundamental topologies are the shunt (or parallel) active filter (PAF) and the series active filter (SAF). All other topologies are either variations, combinations, or additions of these two forms, which primarily compensate for current and voltage, respectively. These basic structures have been shown already in Figure 2.4. Hybrid active filters are the combination of passive and active filters, and are primarily driven by the cost advantages that can be obtained using reduced rating components.

The shunt active filter is the simplest form of active filter to construct. It requires no external power supply (other than its shunt connection to the grid), and can be directly connected to the LV grid without the use of bulky and costly transformers. For these reasons the shunt active filter is the most common form of active filter in the literature, and also in the commercial environment [40] [47] [48]. There are literally hundreds of publications discussing purely shunt active filters (which is more than five times the amount of literature available on any of the other active filter topologies).

The most popular variation for the series topology is to include a passive filter system on the output. This means that the series converter can act as a harmonic isolator with a greatly reduced rating (as low as 4% of the load rating), while the cheaper passive filter elements absorb the main harmonic load currents. This is commonly reported as the Hybrid Series Active Filter [49]-[53] (Figure 2.4c). Whilst the small rating of the series converter greatly reduces the overall cost, the compromise is that the system is generally incapable of compensating for problems such as voltage sags, swells, unbalance, and fundamental voltage deviation. A number of hybrid active-passive topologies have been proposed in recent years and are systematically reviewed by Senini et al. in [54]. The series active filter has also been applied to directly control harmonic voltage in [55] and [56], although very little other literature has been reported for this application.

2.2.4 Uninterruptible Power Supplies (UPS)

Uninterruptible Power Supplies (UPS) are designed to provide an interruption free supply even during deep sags and full interruptions. Therefore, the UPS requires power components which are rated for the full voltage and power requirements of the load. The length of interruption a UPS can handle depends on the energy storage capability of the device. A small UPS can provide power to a personal computer for hours (and even days in extreme cases); but as the required power rating increases, the capital and/or maintenance cost of the energy storage devices becomes prohibitive [57]. Large volume manufacturing of UPS systems at lower power ratings means the UPS is the most common form of Custom Power product on the market. There are three main UPS topologies currently available: the on-line UPS (Figure 2.4d), the off-line UPS (Figure 2.4a), and the line interactive UPS (Figure 2.4f with the shunt converter located on the load side of the series converter). Discussion of some of the line-interactive versions is included in Section 2.2.6, as this UPS type (which may use the unified series-shunt topology) is directly relevant to this research.

2.2.5 Dynamic Voltage Restorers (DVR)

The Dynamic Voltage Restorer (DVR) was proposed to reduce the maintenance and capital costs of UPS systems, and has a typical series converter rating of 15-50% of the protected load [58]-[63]. It is known that a large majority of sag and interruption events have a short duration and are relatively shallow in depth. Therefore, a DVR is expected to lower the net Power Quality cost to the consumer, since even though the capital cost is greatly reduced (because of the reduced rating of the system) the effectiveness of the solution is mostly maintained. Dugan et al. [64] estimate the cost of a DVR (with 50% injection) as $300/kVA, versus $500/kVA for a UPS. The DVR has maintenance costs of approximately 5% of the capital cost, while the UPS has maintenance costs of 15% of the capital cost (i.e. the UPS is five times more expensive to maintain when the difference in capital costs is also factored in) [64].

The cost of a DVR can vary depending on the voltage reference strategy used. The three common strategy types are: pre-sag compensation, in-phase compensation, and energy optimized compensation. The pre-sag scheme injects a voltage such that the load voltage remains at the same magnitude and phase as before the sag occurred (Figure 2.5). With in-phase compensation, the voltage magnitude is preserved at the load, but not the phase. The main difference between the two schemes is that the series injection rating for the pre-sag scheme must be higher to compensate for the phase jump. The third option is the energy optimized scheme, which injects

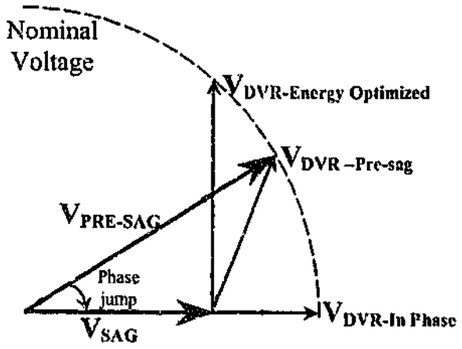


Figure 2.5: DVR control strategy options phasor diagram: Pre-sag compensation, In-phase compensation, and Energy optimized control. (Adapted from [58])

a voltage orthogonal to the sag voltage. This requires minimal real power to be supplied by the DVR, and hence the energy optimized scheme does not require a significant dc energy source. However, it still creates a phase jump and also requires a significantly larger series voltage injection system rating. Since grid connected thyristor based applications in particular are sensitive to phase angle jumps (e.g. Variable Speed Drives and Static Var Compensators) [22] [65], the energy optimized scheme is considered by some in industry [66] and academia [59] to be impractical for most installations. To compensate for sags without the load receiving a phase jump, real power injection must be supplied (at least initially) by the series injection system.

For the in-phase compensation strategy, the series injection capacity of the DVR provides a reasonable estimate of the level of sag that can be accommodated. However, this estimate should be progressively reduced for the pre-sag and energy optimized schemes, as they require a larger series injection capacity to compensate for the same sag levels.

The two common schemes for voltage control using a DVR are feed-forward (open loop), and feed-back (closed loop) control [58]. While both schemes have reported good to excellent sag compensation performance, neither of the proposed strategies have so far been shown to have good voltage harmonic compensation capability. The DVR generally only operates during a sag event, and is usually not operated in steady-state to provide harmonic voltage compensation. Furthermore, while the feed-forward control approach provides good dynamic response for fundamental voltage regulation, there will always be some output voltage error due to voltage drop in the LC filter and series transformer, and non-linearities in the converter (such as deadband). To overcome these problems some schemes have incorporated an inner deadbeat controller, but this suffers from "sensitivity to parameter variations, and dependence on load parameters" [63].

Closed loop voltage controllers typically use a Proportional-Integral (PI) regulator in the fundamental d-q rotating frame, which has a high resonant gain at the stationary frame fundamental frequency, and hence excellent steady-state performance. However, the control loop gain applied to the harmonics with this type of system is much smaller. For this reason, good harmonic compensation performance would not be expected from these control schemes if they were also to be used for steady state harmonic voltage compensation.

Svensson et al. [67] [68] presented simulations of a feedforward voltage harmonic compensation scheme for use with a DVR. However, this system has major limitations in practice. For harmonic compensation using a DVR (with say a 50% injection capacity) the controller must be able to accurately reproduce the harmonic compensation signal at modulation depths of much lower than 10%. Therefore, considering that the practical effects of deadtime, minimum pulse width, quantization, component tolerance, etc., will be significant at this modulation depth, the actual performance of this approach in a practical system is expected to be poor. These issues are considered further in Chapter 10.

2.2.6 Combined Series-Shunt Topologies

Since the combined series-shunt topology (Figure 2.4f) was presented by Moran in 1989 [70], the topology has been of particular interest for power system conditioning applications because of its inherent flexibility. The topology permits a range of specifically targeted Custom Power products to be integrated into a single unit. The shunt converter allows a mixture of reactive power, harmonic current compensation, voltage stabilization, flicker compensation and three-phase current balancing, while the series converter can support fundamental voltage regulation, sag/swell compensation, harmonic current isolation, and harmonic voltage compensation.

The terminologies used for the combined series-shunt topology have become quite varied (possibly to the point of confusion). Some of the acronyms proposed include (in alphabetical order): APLC, LVRC, MUPC, PLC, UCPC, UPLC, UPQC, UPQM, UPS, and USSC. Table 2.3 summarizes these various alternatives. For FACTS applications, the terminology is much simpler, with the term Unified Power Flow Controller (UPFC) most commonly being used. However, UPFCs are designed for transmission lines (where bi-directional power flow is possible) and should not be confused with Custom Power systems which are designed for radial systems (i.e. distribution systems). For this reason, the UPFC and other transmission based unified systems (such as the Universal Active Power Line Conditioner proposed by Aredes et al. [69]) will not be discussed further in this thesis.

Acronym	Full Expanded Name	References
APLC	Active Power Line Conditioner	[69]
LVRC	Line Voltage Regulator/Conditioner	[70]
MUPC	Multilevel Universal Power Conditioner	[71]
PLC	Power Line Conditioner	[72]
UCPC	Universal Custom Power Conditioner	[73][74]
UPLC	Universal (Active) Power Line Conditioner	[75]
UPQC	Unified Power Quality Conditioner	[76]-[88]
UPQM	Universal Power Qualtiy Manager	[89]
UPS	Uninterruptable Power Supply	[90]-[94]
USSC	Unified Series Shunt Conditioner	[95]

Table 2.3: Existing acronyms for combined series-shunt Custom Power devices.

A comparison of the various proposed unified series-shunt schemes is presented in Table 2.4. This table shows that no single system has been reported that can simultaneously compensation for voltage/current harmonics, current/voltage unbalance, fundamental voltage deviations, voltage flicker, sags and swells. The system in this thesis aims to achieve the capabilities shown in the bottom row of the table, where all boxes in the columns are grey. Furthermore, most of the systems were found to have limited capability for direct regulation of load voltages to compensate for pre-existing harmonics in the upstream supply voltage – rather than merely reducing supply harmonics as a consequence of reduced source current harmonics. Most systems which have included facilities to compensate for these harmonics use either a feed-forward scheme (which, as discussed above, has practical performance problems) or a feed-back approach with limited gain on the harmonics. Both approaches lead to a less than acceptable performance. Selected proposed unified series-shunt schemes are discussed below to illustrate these points.

The most commonly reported combined series-shunt device is the Universal Power Quality Conditioner (UPQC) [77]. The UPQC is the combination of a series hybrid active filter, and a small shunt converter. The shunt converter allows small amounts of real power to be injected into the dc-bus, for use by the series converter, so that the system can compensate for voltage flicker as well as for current and some voltage harmonics.

Li et al. [75] proposed a unified hybrid system similar to that of a UPQC, but with an

	Primary Author	Initial Publication Year	Device Name	Simulated Results	Experimental Results	% Series Capability	Tuned Passive Filters	Source Current Comp.		Load Voltage Compensation					
								Harmonics	Unbalance	Harmonics (due to Vs)	Steady State Unbalance	Flicker	Steady State Fundumental	Sag	Swell
Unified Active Series/Shunt	Moran	1989	LVRC	✗	✓	18%	✗	✓	✗	✓	✗	✗	✓	✗	✗
	Akagi	1995	PLC	✗	✓	5%	✓	✓	✗	✗	✗	✗	✗	✗	✗
	Fujita	1996	UPQC	✓	✓	5%	✓	✓	✗	✓	✓	✓	✗	✗	✗
	Jeon	1997	UPS	✓	✓	25%	✗	✓	✗?	✓?	✗?	✓	✓	✓	✓
	Singh	1998	UPQC	✓	Sh.	NS	✗	✓	✗	✓?	✗	✗	✗	✗	✗
	Kamran	1998	UPS	✓	✓	10%	✗	✓?	✓	✓?	✓	✗?	✓	✓	✓
	Kamran	1998	UPS	✓	✓	10%	✗	✓?	✓	✓?	✓	✗?	✓	✓	✓
	Enslin	1998	USSC	✗	✗	RP	✗	RP	RP	RP	RP	RP	RP	RP	RP
	Aredes	1998	APLC/UPLC	✓	✗	NS	✗	✓	✓	✓	✓	✗	✓	✓?	✓?
	Vilathgamuwa	1998	UPQC	✓	Se.	100%	✗	✓	✓	✓?	✓	✗	✓	✓	✓
	Zhan	1999	UCPC	Sh.	✗	RP	✗	RP	RP	RP	RP	RP	RP	RP	RP
	da Silva	1999	UPS	✓	✗	100%	✗	✓?	✗	✓?	✓	✗?	✓	✓	✓
	Chen	2000	UPQC	✓	✗	NS	✓	✓	✓	✓	✓	✓	✗	✗	✗
	Tolbert	2000	MUPC	✓	✓	100%	✗	✓	✗	✗	✓	✗	✓	✓	✓
	da Silva	2000	UPS	✓	✗	100%	✗	✓?	✗	✓?	✓	✗?	✓	✓	✓
	Li	2000	UPLC	✓	✗	25%	✓	✓	✓	✓?	✓	✓	✓?	✓	✓
	Elmitwally	2000	UPQC/DVR	✓	✗	90%	✓	✓	✗	✓?	✓	✓	✓	✓	✓
	Elmitwally	2000	UPQM	✓	✗	18%	✗	✓	✓	✓?	✓	✓	✓	✗	✗
	Zhan	2001	UCPC	✓	✗	DP	✗	DP	DP	DP	DP	DP	DP	DP	DP
	Basu	2001	UPQC	✓	✓	50-100%	✗	✓	✗	✓?	✗?	✗?	✓	✓	✓
	Chen	2001	UPQC	✓	✗	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP
	Chen	2001	UPQC	✓	Sh.	100%	✗	✓	✗?	✓?	✓	✗	✓	✓	✓
	Graovac	2001	UPQC	✓	✗	NS	✗	✓	✗	✓	✓	✓	✗	✓	✓
	Singh	2001	UPQC	✗	✓	10%	✓	✓	✓	✓?	✓	✗	✗	✓?	✗
	Elnady	2001	UPQC	✓	✗	30%	✗	✓	✗?	✗	✗	✗	✗	✓	✓
	Elnady	2001	UPQC	✓	✗	NS	✗	✓	✓	✗	✓	✗?	✓	✓	✓
	Jianjun	2002	UPQC	✗	✓	NS	✗	✓?	✓	✓?	✓	✗?	✓	✓	✓
	Target		UCPC	✓	✓	25-50%	✗	✓	✓	✓	✓	✓	✓	✓	✓

✓? Some comp. of this type is possible, but performance not necessarily adequate (refer to discussion)
✗? This form of compensation seems to be possible, but is not quoted in the publication
NS Infonnation Not Supplied in publication
RP Review Paper: therefore performance attributes quoted vary
DP Design Process paper; therefore capabilities given with no justification
Sh. Shunt Only
Se. Series Only

Table 2.4: Review of combined series-shunt Custom Power research outlining the reported Power Quality compensation capabilities of each proposed device. Note that a grey background indicates the physical and compensation capabilities of each system which are preferred by this research.

increased series rating to enable voltage regulation and sag/swell compensation whilst keeping the rating of the shunt converter small. However, as discussed in Sub-section 2.2.5, real power injection is required to compensate for sags without creating a voltage phase jump at the load. The small shunt converter does not have the capacity for this, and therefore the system is forced to use an energy optimized type controller [96] [97], which causes phase jumps during sag/swell events. This also means that the steady-state voltage regulation is limited, due to the increased requirement on voltage injection to provide the out of phase voltage. As with the majority of unified controller systems the proposed system has only been verified in simulation.

Elmitwally et al. [81] proposed another type of unified hybrid active-passive system to compensate for harmonics, voltage unbalance, and voltage flicker as well as to regulate the fundamental voltage. The series control system incorporates an inner current control loop with an outer fuzzy logic load voltage control loop. The results show that the total harmonic distortion (THD) of the load voltage increases, instead of decreasing, and is actually double that of the source voltage! A drop in the source voltage THD is claimed, but this is only due to the large reduction in source current harmonics achieved by the passive filters and series isolator combination. The results display acceptable voltage unbalance and flicker compensation, but the voltage sag response is very slow. Furthermore, the promise of simultaneous reactive power compensation, fundamental steady state voltage compensation, and sag compensation without phase jump contradicts the claim of a low rated shunt converter. The proposed system is once again only verified in simulation.

Table 2.4 indicates that line interactive UPS systems [90]-[94] may offer a more promising approach for a complete unified system. The shunt converter for these systems must be rated for the entire load rating and of course the systems require energy storage anyway (as discussed in Sub-section 2.2.4). However, the usual UPS control scheme requires that the series transformer has either a high leakage inductance or a separate inductance to be placed in series with the line [92], to give the shunt converter (which is run in voltage regulation mode) sufficient dynamic control over the load voltage. Unfortunately, the side effect is that the load voltage becomes quite sensitive to step changes in load current, as is clearly shown in the experimental results provided by Kamran et al. [92].

In principle it seems quite feasible for the series-shunt topology to resolve all types of Power Quality problems, with good performance. However, Table 2.4 and the related discussion above shows that this result has not yet been successfully achieved in any reported system.

2.2.7 Practical Implementation of Custom Power Systems

The fundamental converter topologies used for Custom Power applications are similar to many other converter based applications, with the main distinguishing factor being the control systems used and their target outputs. Hence, the core focus of this thesis is the control systems used. The demand for a very accurate and fast response from the control system means that the practical implementation of these systems becomes critical -- especially when digital implementations are used. Hence, in this thesis the concept of 'control' is extended beyond theoretical algorithms to include practical aspects such as the digital implementation of the primary control systems, and the control of the converter protection systems.

The flexibility, reduced size and cost of fully digital implementations has created a shift from analog to digital signal processor (DSP) based power electronic applications. However, for many control systems, digitization can alter the expected behavior of the system if this process is not fully accounted for. Furthermore, the DSPs used in many power converters have a fixed-point architecture, which can further degrade the performance of the controller. A specific review of these problems is presented in Chapter 6, as a background to the solution proposed in this thesis.

The issue of converter protection for the series topology has also been mentioned in the literature as a major difficulty for practical systems. The literature provides only limited details of possible solutions to the problem [39] [49] [98] [99]. A fault protection scheme for series converters was presented in 1996 by Moran et al. [100] [101], but there are practical issues with the implementation of the proposed system. A review of this issue is included in Chapter 7, as a background to the solution proposed in this thesis. The issues of protection against all converter faults and the control of the series converter during start-up, fault recovery, and shut-down, are also considered in this chapter.

The voltage and power ratings of the UCPC need to be considered as well. These have been investigated for the DVR by Nielsen et al. [60] and Bollen et al. [102], but for these investigations the supply source was taken to be a passive shunt rectifier. Chen et al. [83] extended this work for the active series-shunt topology (which included tuned passive filters for the harmonic compensation), but only accounted for the steady-state power from the shunt. Chapter 3 re-develops this work to suit the UCPC, and extends the theory to incorporate the transient energy capabilities into the rating model.

Finally, transformer design has been identified as an important consideration for the series topology [99]. Series based Custom Power applications requiring fast and large injection voltages

(e.g. a DVR compensating for a sag) can easily saturate the series transformer if it is not rated for "twice the normal steady-state flux" [99]. The transformer design affects the control systems in two ways. Firstly, if a double transformer rating is not feasible, then the controller must be constrained to ensure that the converter limits the current into the transformer during saturation to stop over-current faults. Secondly, the design of the transformer governs part of the design of the control system for the series protection. Hence, some consideration of transformer design and its effect on the control system performance, is presented in Chapters 4, 9 and 10.

2.3 Summary

The chapter has reviewed the Power Quality problems faced by modern consumers, and categorized them into seven types. Types 2-6 were identified as appropriate for further consideration, while types 1 and 7 were identified as unlikely to be compensated using power electronics because of practical limitations. This review also provides a basis for the rating and control design considerations used throughout this thesis for the development of the proposed Custom Power application.

The chapter then reviewed VSI based Custom Power applications that have been reported in the literature to address Power Quality problems. The review showed that series based topologies offer improved potential for Power Quality compensation because of the increased capabilities of this type of system, and also identified a research gap in the area of the use of series based converters to accurately remove voltage harmonics from load voltages where they are caused by voltage harmonics upstream of the unit's insertion point. It was further recognized that a Custom Power application for radial systems that can combine this attribute with fast and accurate simultaneous compensation for voltage sags/swells/fundamental variation/unbalance/flicker and current harmonics/unbalance is the ultimate goal. The flexibility of the combined series-shunt topology shows promise in achieving this task, but improved control strategies need to be developed to achieve a completely effective solution.

Chapter 3 establishes the physical structure and control requirements of the combined series-shunt based Custom Power device needed to achieve this ideal goal. The term Universal Custom Power Conditioner, or UCPC, is used for the proposed system. Chapter 4 then explores appropriate control schemes for the UCPC. Controller implementation issues of the UCPC are then expanded in Chapters 5, 6, and 7.

Chapter 3

The Universal Custom Power Conditioner (UCPC)

Chapter 2 has identified a need for research into a single Custom Power device, capable of compensating for Power Quality problems of types 2 through 6. For this research, it has been stipulated that the conditioner should be for radial systems, suitable for either LV or MV installations, and should be applicable to either three-phase or single-phase systems. The controller for the system should be fully digital to allow for upgrades, flexibility, and increase its tolerance to aging drift effects seen in analog systems. The system should protect downstream consumer loads from Power Quality problems in their supply voltage such as: harmonics, sags, swells, phase jump, flicker and unbalance. In turn, the currents taken from the grid supply should be regulated so that they remain free from harmonics and unbalance irrespective of the load currents drawn by consumers.

This chapter presents a general arrangement of a series-shunt topology to achieve these aims. The resulting system is termed the Universal Custom Power Conditioner, or UCPC. An overview of the UCPC concept is provided, including objectives of the series and shunt converters, physical components of the system, control system and feed-back/feed-forward parameter options, power/voltage/current ratings, and a brief discussion on the use of the UCPC in an electrical distribution system.

The concepts presented in this chapter are then applied in Chapter 4 to develop a control scheme for the topology. This control scheme is then further developed in Chapter 5 to ensure that the system can be applied to a single-phase system, and is also practically implementable.

3.1 Overall Objectives of the UCPC

Since it was first proposed [70], the unified series-shunt converter topology has been of particular interest for power system conditioning applications because of its inherent flexibility. The most appropriate terminology for the resulting device is the Universal Custom Power Conditioner, or UCPC. The term Universal Custom Power Conditioner (UCPC) was first used by Zhan et al. [73] to describe a general combination of existing Custom Power applications capable of compensating for a range of Power Quality problems [74]. The term UCPC has been adopted for this work instead of the more commonly used term of UPQC (Unified Power Quality Conditioner), because UPQC was originally used to describe the topology of a small rated series-shunt converter with larger rated tuned passive filters (a topology that only addresses some of the Power Quality goals considered in this research). In contrast, the work described in this thesis investigates a purely active approach (i.e. no tuned passive filters), and requires larger converter ratings to enable the compensation of larger voltage variations (such as sags). Therefore, the term UCPC defines a separation between the two approaches, and is more appropriate.

The operation of a UCPC in a distribution network is shown in Figure 3.1. The target objective of the control system is to ensure the downstream load sees a sinusoidal and balanced[1] voltage (of correct magnitude and phase), while the upstream grid supply sees a sinusoidal and balanced current[1] . To achieve this result, the series converter compensates for supply voltage variations from the ideal, which include fundamental magnitude and phase, unbalance, sags, swells, harmonics and flicker, while the shunt converter compensates for deviations in the supply current from the ideal, which include harmonics and unbalance. The reasoning for this choice and segregation is detailed in Section 3.3.

The primary *harmonic* purpose of the series portion of the UCPC is to attenuate pre-existing voltage harmonics that are upstream from the connection point, since any voltage distortion caused by distorted load currents will be addressed by the load current filtering action of the shunt converter. However, even without the shunt active filter portion, the UCPC still has the capability to compensate for voltage harmonics caused by both upstream and downstream distortion sources, since it directly acts to regulate the load side voltage. In contrast, the shunt active filter portion operating in isolation will only remove voltage distortion caused by downstream non-linear load current, by removing these currents from the supply. However, when combined with the series filter section, it does alleviate some of the voltage compensation work for the series part of the system.

[1]Note that for single-phase systems, balancing is not required, since there is only one voltage and current path.

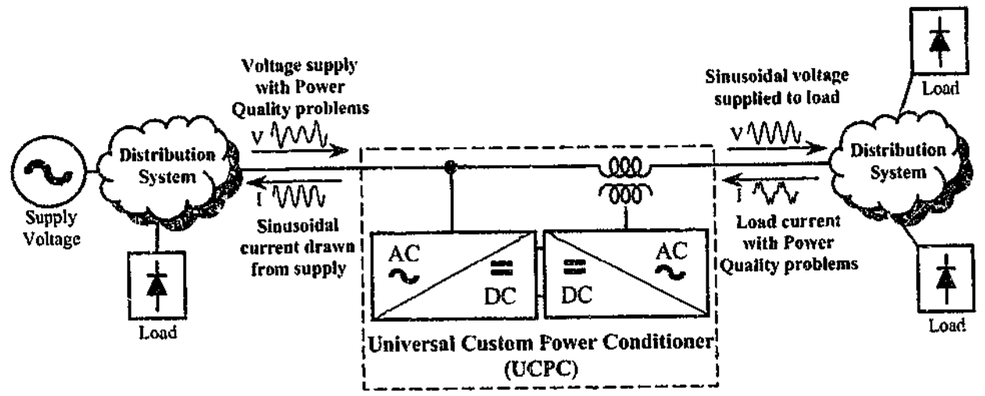


Figure 3.1: Overview of the operation and placement of a Universal Custom Power Conditioner (UCPC) in a radial distribution system.

For purely sag compensation, when compared to a DVR, the continuous operation of the UCPC will have higher power losses, as the DVR only has conduction losses in its steady state mode. However, reducing harmonics in the upstream supply current is likely to somewhat reduce the distribution system losses, and may also lead to some reduction in losses in consumer equipment due to the less distorted voltage. Hence, the ability of the UCPC to regulate the fundamental voltage may achieve some reduction in distribution losses that counteract the increased steady state losses of the device. Whether the resultant losses are higher or lower will depend on the distribution system, the magnitude of existing harmonics, the efficiency of the chosen converters, and many other parameters. This issue is not pursued further in this research.

3.2 Physical Components of the UCPC

Figure 3.2 illustrates the major physical components of the proposed UCPC. The physical components can be broken up into four sections; **(a)** primary power components, **(b)** ancillary power components, **(c)** measurement hardware, and **(d)** controller hardware, and each of these sections are discussed in detail in later chapters of this thesis.

The parameter values for the series components are denoted by the subscript SAF (i.e. Series Active Filter), and for the shunt parameters the subscript PAF (i.e. Parallel Active Filter) is used. The voltage at the supply voltage terminals is V_T, while the source voltage (i.e. the voltage behind the supply impedance) is denoted by V_S. The load voltage is identified as V_L.

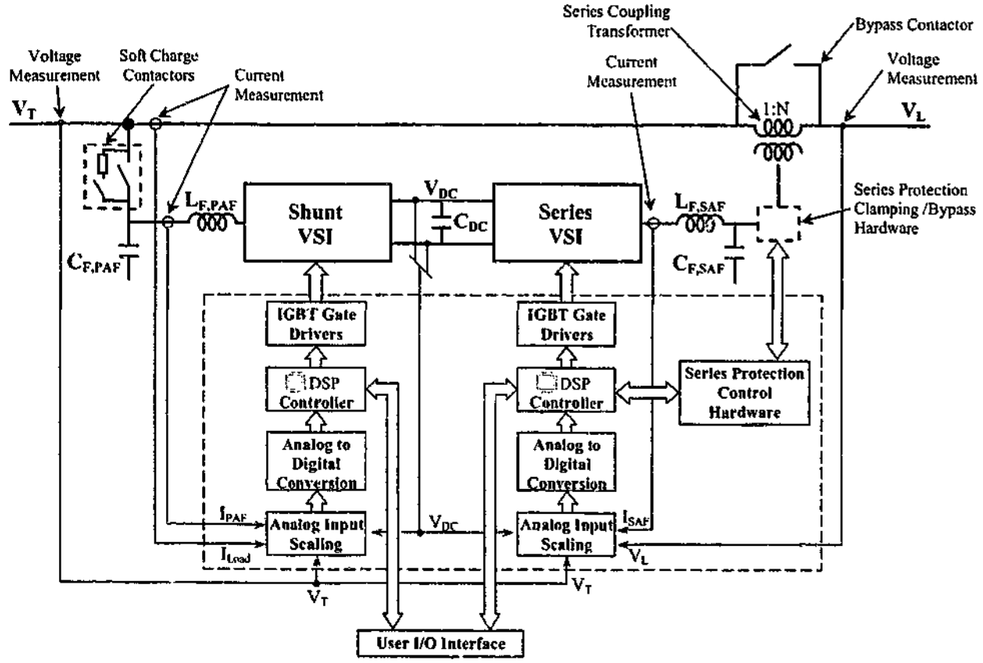


Figure 3.2: Physical components of the Universal Custom Power Conditioner (UCPC).

3.2.1 Primary Power Components

The primary power components are the power converter elements that are essential to the operation of the UCPC. Many of these elements exist in both the series and shunt topologies and are identified throughout this work using the SAF and PAF subscripts, respectively. The major components are:

- Series topology components
  - Voltage Source Inverter (VSI)
  - Filter inductors ($L_{F,SAF}$)
  - Filter capacitors ($C_{F,SAF}$)
  - Coupling transformer
- Shunt (parallel) topology components
  - Voltage Source Inverter (VSI)
  - Filter inductors ($L_{F,PAF}$)
  - Filter capacitors ($C_{F,PAF}$)
- Dc-bus capacitance

The term voltage source inverter includes the IGBTs, high frequency capacitors, heatsinks, bleed-resistors, and bus-bars and is described in detail in Chapter 9. (Note that the gate driver is taken as part of the controller hardware.) The design and effect of the LC filter parameters is described in Chapter 4, while all the magnetic components are discussed further in Chapter 9.

3.2.2 Ancillary Power Components

The ancillary components can vary depending on exactly how the UCPC is constructed. They include soft start contactors and relays, switch mode power supply (SMPS) for the controller hardware, converter/system protection elements, EMI components, cabling, connectors, and of course the UCPC housing. All of these components are discussed with the experimental work in Chapter 9, while Chapter 7 describes in particular the protection elements and their control.

3.2.3 Analog Measurement Hardware

The current and voltage measurements required by the UCPC are:

- Current measurements
  - Load current (used by shunt converter)
  - Shunt converter current (used by shunt converter)
  - Series converter current (used by series converter)
- Voltage measurements
  - Supply terminal voltage (used by both shunt and series converters)
  - Load voltage (used by series converter)
  - Dc-bus voltage measurement (used by both series and shunt converters)

Measurement of the voltage and current parameters is a critical part of the signal path, since the accuracy and performance of the system is dependant on the measurement quality. For all experimental work, currents were measured using hall effect sensors (i.e. LEMs), while voltages were measured using resistive dividers, followed by differing forms of additional isolation. The measurement devices used for the LV and MV experimental work are described in Chapters 9 and 10, respectively.

3.2.4 Controller Hardware

The controller hardware includes all the components from the raw analog measurement inputs to the inputs of the gates of the IGBTs. This includes analog conditioning, analog to digital conversion (integrated with the DSP for the LV experimental work), the DSP controller, and the IGBT gate drivers. The controller hardware also includes interface circuitry to allow interaction with the operator. This includes some (or all) of the following components: RS-232/485 serial interface, front panel buttons or keypad, or LCD. Dedicated controller hardware for the series protection system is also required and is discussed in Chapter 7. The remaining components are detailed in the experimental Chapters 9 and 10.

3.3 Control Concepts for the UCPC

The control system of the UCPC can be broken up into separate controllers for the shunt and series converters, provided the two schemes are only loosely coupled. Coupling between the converters can occur in two ways. Firstly, the converters connect via a common dc-bus, and therefore only one of the two controllers can have direct control over this parameter. This role is typically taken by the shunt controller, since demanding real power injection from the series converter will result in a magnitude and/or phase variation of the load voltage. However, due to the large capacitance used on the dc-bus, its voltage only varies slowly (compared to the other parameters in the system), and it is therefore unlikely to cause any significant coupling between the two controllers. The second possible form of coupling is between the two converter outputs. Both series and shunt topologies can regulate the system voltage or current to some degree. However, both cannot regulate the same quantity if they are to operate dc-coupled. Therefore, the only two options are:

1. Series regulation of the load voltage and shunt regulation of the supply current.
2. Series regulation of the supply current, and shunt regulation of the load voltage.

Hybrid based series-shunt solutions typically implement the second option, with the series converter acting as an isolator of current harmonics, and forcing such currents into the tuned harmonic passive filters. The shunt converter then controls the power flow (via dc-bus regulation), and some other functions. Line interactive UPS systems incorporating the series-shunt topology also typically use the second option, as when the voltage is lost the full rated shunt converter must regulate the load voltage. However, it is more common to make a shunt active filter regulate current and operate using the first option [40] [47] [48], with only a small number of alternatives having been reported as using voltage harmonic regulation/damping [103] (i.e. the second option). Lastly, the series based dynamic voltage restorer (DVR) system uses option 1 and directly regulates the load voltage [60]-[63].

The shunt converter can directly inject currents into a system (assuming a stiff supply), but can only alter the load voltage indirectly by changing the current in the grid. Likewise, the series topology can directly alter the load voltage, but can only affect the supply current indirectly (i.e. by the current drawn by the load varying because of the voltage change). Hence, it seems likely that the best performance will be gained by using the direct regulation approach as shown in Figure 3.3 (i.e. option 1 with series converter voltage regulation and shunt converter current regulation). This is the approach investigated in this research.

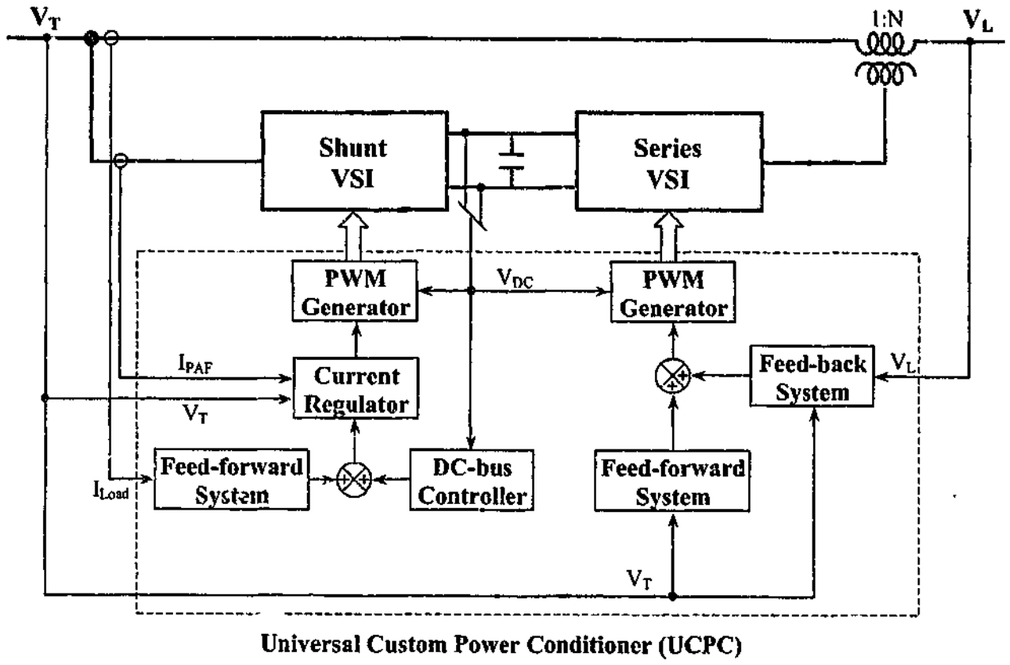


Figure 3.3: General structure of the control scheme of the Universal Custom Power Conditioner.

3.3.1 Voltage Regulation of the Series Converter

The series converter regulates the downstream load voltage to match a target sinusoidal reference. The difference between the upstream supply voltage and the target load voltage is injected by the series converter, and must be tightly controlled to ensure that the voltage harmonics and other disturbances are cancelled. The two available control variables are the supply and load voltages. The supply voltage is the disturbance and the feed-forward parameter. The load voltage is the target, and the feed-back parameter. For good transient response feed-forward is the better option, as it can directly produce the required volt-seconds output as required at the output of the VSI. This is the method used by most DVRs [57] [61] [62]. However, due to the voltage drop across the filter inductance and also because of the series transformer, the injected voltage will have some small errors. For harmonic voltage compensation, high magnitude and phase accuracy is required, yet the modulation depths are likely to be very low (e.g. typically around 1%-10% for a series converter with 0.5 p.u. injection capacity). Therefore the performance is expected to be greatly limited by effects such as digital sampling delays, dead-time, minimum pulse widths, PWM timer resolution, etc., as well as the filter voltage drop errors already mentioned. Hence, feed-back of the load voltage is also required to achieve an acceptable level of steady-state error – especially for compensation of smaller magnitude Power Quality conditions such as voltage harmonics and unbalance. Figure 3.3 presents an overview of this scheme. Note that the measurement of the dc-bus voltage is passed into the PWM generation block. This allows the PWM modulation depth to be calculated based on the instantaneous dc-bus voltage, to ensure that it always produces the required volts-seconds output. This is called dc-bus compensation in this thesis. It also has the advantage of keeping the control loops independent of variations in the dc-bus voltage, which further minimizes any interaction between the series and shunt control schemes that may occur through this link.

3.3.2 Current Regulation of the Shunt Converter

The shunt converter regulates the upstream supply current so that it remains balanced and free from distortion. The two control variables are the load and supply currents. The load current contains the disturbance, and is the feed-forward parameter. The supply current is the target, and is the feed-back variable. However, unlike the series converter, a feed-forward control loop cannot be implemented directly, because the desired output is a current, yet the demanded output of a VSI is voltage. Therefore, an inner current loop is required to implement this role. Once more, it must be decided which of the two variables to use (or both). For this research only the feed-forward (i.e. load current measurement) option is used (Figure 3.3), so the shunt converter measures the actual load current, and injects a current to cancel out any unwanted harmonics and unbalance. If correct cancellation is achieved, then only a sinusoidal balanced current will be drawn from the supply.

Feed-forward compensation has been chosen for the following reasons. Firstly, the allowable limits on individual current harmonics are higher than for voltage harmonics [21], and (assuming good regulation by the inner current loop) a feed-forward method only should be capable of meeting these limits. Using feed-forward current regulation means that it will have no stability problems (assuming the inner current loop is stable), and this reduces control interaction between the series and shunt controllers. Current measurement is also typically more expensive than voltage measurement, and a reduced number of current sensors is to be preferred (compared to the alternative of measuring both load and supply currents). Finally, the compensation current for typical diode/thyristor rectified loads generally contains sharp transitions, and can vary significantly from cycle to cycle. This type of load is better suited to a feed-forward based control system.

3.3.3 Combined Series-Shunt Operation of the UCPC

Power flow is the major joint task conducted between the series and shunt converters, and is essential for them to achieve their respective objectives. The bi-directional power flow capabilities of both the series and shunt converters are utilized by the UCPC. The series converter can either increase or decrease the load voltage, which will in turn absorb or inject power from/to the dc-link, respectively. This power flow is then moved from/to the distribution system via the shunt converter. The dc-bus of the UCPC connects the series and shunt converters and balances the power between the two converters. The shunt converter demands real power from the grid to regulate the dc-bus to a pre-determined value, which then creates the power flow required by the series converter.

The placement of the shunt converter either upstream or downstream of the series converter can greatly vary the ratings of the converters, because of the different requirements of the two options during deep transients (such as voltage sags). This is discussed in more detail later in this chapter. However, the conclusion for this research is that the shunt converter location was chosen to be upstream of the series converter (Figure 3.3).

So far the coupling between the series and shunt controllers has been separated into the dc-bus interaction discussed above, and interaction via the target grid parameters of each controller. The target regulation parameters of each controller, and the feed-forward controller only on the shunt converter, have been chosen to minimize this interaction. However, some interaction will still exist. For example, if the shunt converter compensates for distortions in the load current, these distortions will be removed from the supply current. This will, in turn, minimize the supply voltage harmonics created because of harmonic voltage drop across the supply impedance. The series controller will then compensate for this change in load voltage conditions, which may vary the current drawn by the load. This current variation is then detected by the shunt converter and the interaction loop is complete. However, since this loop depends on parameter changes which are expected to only be very small (e.g. even a large change in supply current should only create a small variation in the supply voltage), then the overall problems caused by this interaction are expected to be minimal. Hence the approach that has been used was to design and develop the series and shunt controllers separately for the majority of the work presented in this thesis. Then the operation of the two systems was verified as a combined unit, in both simulation and experiment, as described in Chapters 8, 9, and 10.

3.4 Ratings and Injection Capabilities of the UCPC

The capabilities of the proposed UCPC are determined by the ratings of the primary power components. Voltage sags are the primary issue with regard to the power, voltage and current ratings of the UCPC, since all other Power Quality conditions typically require much lower ratings.

Chapter 2 identified that a Dynamic Voltage Restorer (DVR) typically has half the power rating, and contains much smaller storage elements than does an Uninterruptable Power Supply (UPS). It is therefore approximately 40% cheaper than the UPS, and also has maintenance costs which are five times less. Also, the discussion in Chapter 2 identified that the reduced rating does not equate to a similar reduction in sag compensation capabilities, because a large majority of sags are relatively shallow. Hence, a voltage injection capability of no more that 0.5 p.u. was chosen for the series component of the UCPC developed for this project.

This section investigates in detail the compensation limits that are feasible for the proposed UCPC. In general, the voltage compensation depth and duration are limited by both the series injection voltage, and the available power flow to the series converter. This flow is regulated by both the maximum current of the shunt converter (I_{PAF}) and the storage capacity of the dc-bus (Figure 3.4). The limits are investigated and linked to obtain the net sag compensation capabilities of the UCPC with respect to depth ($V_{T,SAG}$), phase (ϕ_{SAG}), and duration (t_{SAG}), of the voltage event, as well as the load current magnitude (I_L) and load phase angle (ϕ_L). Finally, the limits of the UCPC are correlated with the equipment susceptibility limits and the probability of sag events, to estimate the compatibility of the proposed Custom Power device with consumer loads.

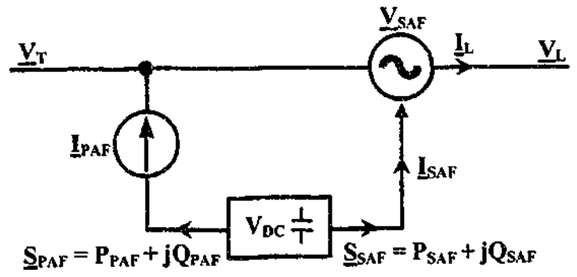
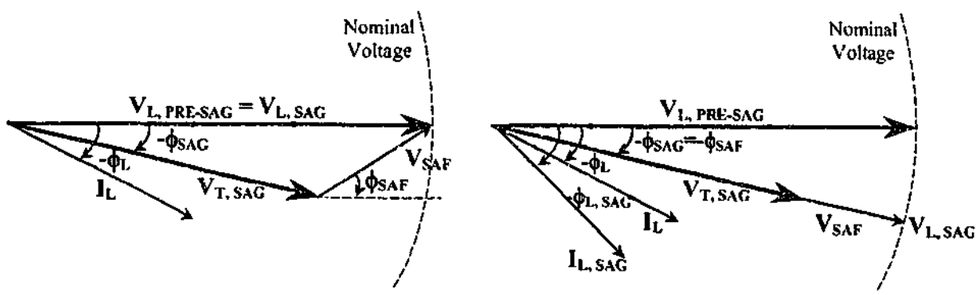


Figure 3.4: Simplified UCPC circuit diagram outlining terminology and polarities used for injection capabilities investigation.

3.4.1 UCPC Sag Injection

As discussed above, a series injection capacity (V_{SAF}) of 0.5 p.u. was chosen, and the same limit was also imposed on the shunt converter current (I_{PAF}) to maintain a more feasible overall system cost. The remaining limit is the short-term energy capacity of the capacitors, and this is determined by the difference in stored energy between the dc-bus voltage (V_{DC}) before the event and the minimum allowable voltage to which the dc-bus can reduce while continuing to compensate.

Chapter 2 discussed the three primary methods used by a DVR to compensate for sags. The proposed UCPC uses a combination of the In-phase and Pre-sag compensation methods (Figure 3.5). Full details of the control scheme are provided in Chapter 4, but for this investigation it should be noted that the Pre-sag method is initially used, then the UCPC slowly shifts towards the In-phase compensation method for steady-state voltage magnitude variations. However, the injection and rating calculations are only developed here for the Pre-sag method, as the ratings for the In-phase method can be easily derived from these results by setting the phase jump angle (ϕ_{SAG}) to zero.



(a) Pre-sag compensation phasor diagram. (b) In-phase compensation phasor diagram.

Figure 3.5: Phasor diagrams for compensation of voltage sags using Pre-sag and In-phase compensation.

3.4.2 UCPC Steady-State Power Flow

During a voltage sag, real power is injected into system by the series converter, and has a magnitude given by

$$|P_{SAF}| = \sqrt{3}\,|V_{SAF}|\,|I_{SAF}|\cos\left(\phi_{SAF} - \phi_L\right), \tag{3.1}$$

where the series converter current is equal to the load current (on a p.u. basis), ϕ_{SAF} is the phase angle of the series injected voltage relative to the pre-sag voltage, and ϕ_L is the load

current phase angle relative to the pre-sag voltage. This power flow must come from either the dc-bus storage capacitors, or the shunt converter. The power supplied by the shunt converter is given by

$$|P_{PAF}| = \sqrt{3}\,|V_{T,SAG}|\,|I_{PAF}|\,. \tag{3.2}$$

Under steady-state conditions the dc-bus voltage should remain stable, and therefore all the power should be derived from the shunt converter, so that

$$|P_{SAF}| = -\,|P_{PAF}|\,. \tag{3.3}$$

(Note that the negative sign is due to the sign convention presented in Figure 3.4.)

The reactive power flow out of the series converter is given by

$$|Q_{SAF}| = \sqrt{3}\,|V_{SAF}|\,|I_{SAF}|\cos\left(\phi_{SAF} - \phi_L\right), \tag{3.4}$$

and is provided by the converter. On the other hand, since the shunt converter current is always in phase with the supply terminal voltage (V_T), its reactive power flow is simply

$$|Q_{PAF}| = 0. \tag{3.5}$$

The power ratings of VSIs are primarily based on their converter current, and the maximum operating dc-bus voltage (Chapter 9). The dc-bus voltage is boosted from the rectified value by the shunt converter, and a default boost of 20 % was chosen for this investigation (i.e. 700 V_{dc} for a 415 V_{ac} nominal grid voltage). For the series converter the primary referred current is equal to the load current, and therefore its power rating is determined by the series transformer ratio. This ratio then determines the voltage injection capabilities of the UCPC, which in turn limits the sag compensation. Therefore, for the series converter, only the injection voltage (V_{SAF}) is of primary interest, and this is investigated in the following sub-section, 3.4.3.

For the shunt converter, the current is not limited by the load current (unlike the series converter), and it can therefore easily become quite large under certain circumstances. For example, if 0.5 p.u. real power is required by the shunt converter, and a voltage sag of 0.2 p.u. exists, then from (3.2) the shunt converter current required is 1.4 p.u. (i.e. nearly three times the required power). However, the shunt converter voltage rating is directly linked to the system voltage, and is unlikely to see such extremes. Therefore, for the shunt converter, only

the injection current (I_{PAF}) is of primary interest, and this is also investigated in Section 3.4.4.

Finally, to overcome any possible power delivery limitations of the shunt converter, the stored energy in the dc-bus can also be taken into account for short term sags. Under these circumstances the shunt converter is current limited, and the additional energy comes from the dc-bus capacitor storage. The power balance in (3.3) now becomes invalid, but the sag compensation abilities also become time dependant.

3.4.3 Series Voltage Injection Requirements

From the phasor diagram shown in Figure 3.5 the required voltage injection for the Pre-sag compensation method is

$$|V_{SAF}| = \sqrt{|V_L|^2 + |V_{T,SAG}|^2 - 2|V_L||V_{T,SAG}|\cos(\phi_{SAG})}, \qquad (3.6)$$

where the In-phase requirement is given by substitution of $\phi_{SAG} = 0$. The series voltage injection phase angle can then be derived to be

$$\phi_{SAF} = arccos\left(\frac{|V_L|\cos(\phi_{SAG}) - |V_{T,SAG}|}{\sqrt{|V_L|^2 + |V_{T,SAG}|^2 - 2|V_L||V_{T,SAG}|\cos(\phi_{SAG})}}\right) + \phi_{SAG}. \qquad (3.7)$$

Figure 3.6 presents these results in graphical form, and shows that a system with a series injection capability cannot compensate for voltage sags with phase jumps of more than 30^o, irrespective of the sag depth. However, for voltage sags with phase jumps between 0^o and 25^o, sags depths of 0.5 p.u. to 0.65 p.u. are possible, respectively. Note that the sag voltage that can be compensated is independent of the load current parameters.

3.4.4 Shunt Current Injection Requirements

Under sag conditions (assuming no energy storage contribution), using (3.1), (3.3), and (3.6), the required shunt converter current is found to be

$$|I_{PAF}| = -\frac{|I_L|}{|V_{T,SAG}|}\sqrt{|V_L|^2 + |V_{T,SAG}|^2 - 2|V_L||V_{T,SAG}|\cos(\phi_{SAG})}\cos(\phi_{SAF} - \phi_L). \qquad (3.8)$$

From this equation the relationship between the required current, voltage sag depth and phase jump is shown in Figure 3.7. As can be seen for a 0.8 p.u. load current, a shunt converter capable of only 0.5 p.u. current injection can sustain sufficient power delivery for voltage sags as deep as approximately 0.6 p.u.. Comparison of Figures 3.6 and 3.7 shows that for equivalently

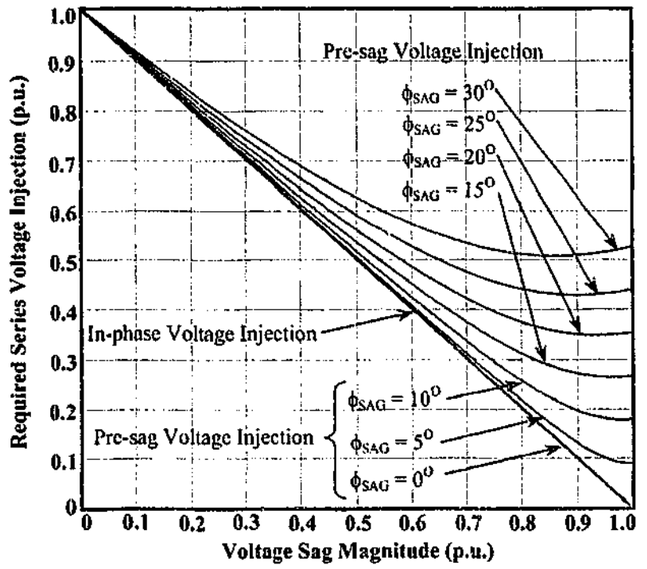


Figure 3.6: Required series voltage injection of the UCPC during voltage sag conditions (i.e. varying depth and phase jump).

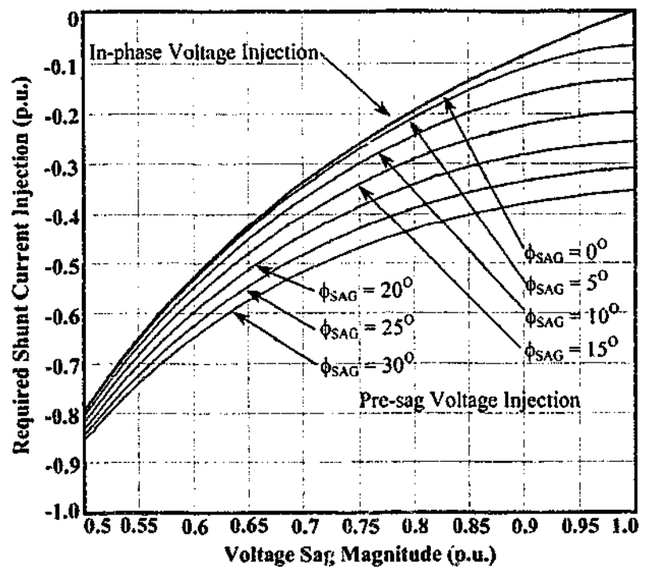


Figure 3.7: Required shunt currnet injection of the UCPC during voltage sag conditions (i.e. varying depth and phase jump), with 0.8 p.u. load current.

rated series and shunt converters (i.e. 0.5 p.u.), the limiting factor for steady-state voltage compensation is the shunt converter. However, this is not always the case under transient conditions when the dc-bus storage facilities are taken into account, as will now be shown.

3.4.5 Equating Energy Flow Constraints

When the required current of the shunt converter of the UCPC exceeds its maximum rating, the control system will limit the current to the largest steady-state permissible value. Under these conditions the series converter will continue to demand real power, which will be supplied from the energy storage of the dc-bus (Note that the dc-bus voltage is no longer regulated by the shunt converter due to the current limiting). The power flow out of the dc-bus is then given by

$$P_{DC,NET} = P_{PAF} + P_{SAF}. \tag{3.9}$$

The UCPC must stop compensating once the dc-bus voltage reaches $V_{DC,Min}$. Therefore, the maximum energy deliverable by the dc-bus is

$$\Delta E_{DC} = \frac{1}{2} C_{DC} \left(V_{DC,Nom}^2 - V_{DC,Min}^2 \right). \tag{3.10}$$

Combining (3.9) and (3.10) results in a maximum permissible sag duration of

$$t_{SAG,Max} = \frac{C_{DC} \left(V_{DC,Nom}^2 - V_{DC,Min}^2 \right)}{2 P_{DC,NET}}. \tag{3.11}$$

Figure 3.8 presents the combined compensation capabilities of the UCPC with 0.5 p.u. rated series and shunt converters, and dc-bus storage capacity equal to the experimental system in Chapter 9 (i.e. 7.05 mF). These results account for the following: voltage injection limits, current injection limits, dc-bus storage usage, voltage sag depth, voltage sag phase jump, as well as for both the Pre-sag and In-Phase sag compensation control methods. Note that for longer duration sags, the compensation method approaches the In-Phase alternative, hence allowing for continuous voltage variations as deep as 0.62 p.u. (and is limited by the shunt converter rating). For very short duration sags, the compensation limit is the series injection voltage previously discussed, and is not based on energy restrictions.

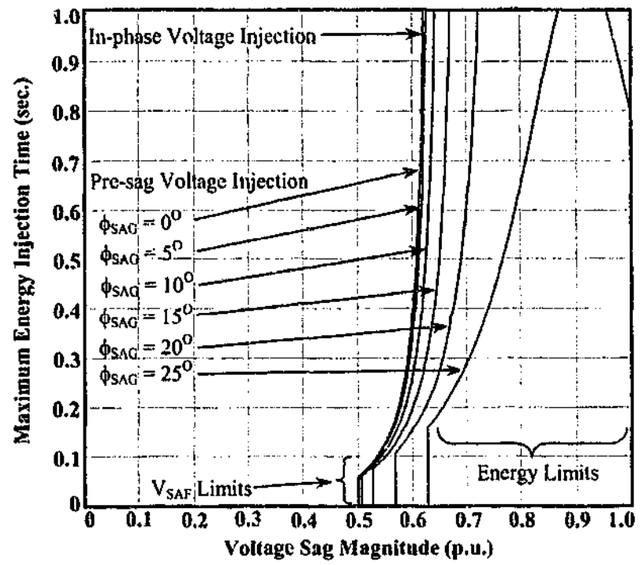


Figure 3.8: Maximum sag compensation capabilities of the UCPC with respect to; time (t_{SAG}), sag depth ($V_{T,SAG}$) and phase jump (ϕ_{SAG}), with a unity target load voltage and a 0.8 p.u. load current.

3.4.6 Compatibility Limits for UCPC Protected Equipment

The calculations so far in this section have assumed an ideal target load voltage (i.e. 1.0 p.u.). Appendix A shows that consumer equipment has a voltage susceptibility limit which varies significantly depending on the equipment. The majority of equipment is shown to be able to handle sags of 0.8 p.u., or less, without affecting its operation, and ITIC (previously CBEMA) limits recommend that consumer equipment be able to withstand sags of this depth for up to 10 seconds. Therefore, whilst the UCPC may not be able to fully compensate for sags deeper than 0.5 p.u., partial compensation of deeper sags may still achieve the objective of avoiding misoperation of consumer equipment.

Figure 3.9 shows the compensation limits for different target load voltages. For a target load minimum voltage of 0.8 p.u. the compensation range is dramatically improved. For this same target load voltage Figure 3.10 shows an updated version of the net compensation capabilities of the UCPC (compared to Figure 3.8), where a large increase in the range of operation of the UCPC has been achieved.

Sag measurement data from an extensive EPRI study [20] (see also Appendix A) has shown that a large majority of events have sag depths no lower than 0.5 p.u.. However, this data does

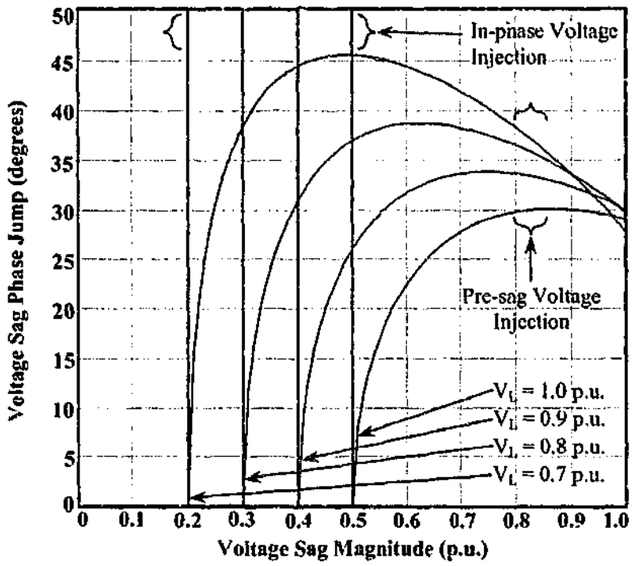


Figure 3.9: Maximum sag compensation capabilities of the UCPC (maximum 0.5 p.u. voltage injection) with respect to sag depth and phase jump, with varying minimum load voltage (V_L) equipment susceptibility limits.

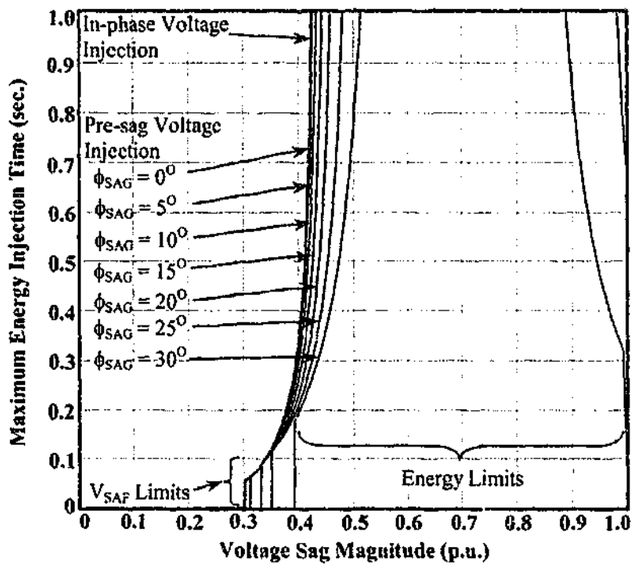


Figure 3.10: Maximum sag compensation capabilities of the UCPC with respect to time (t_{SAG}), sag depth ($V_{T,SAG}$) and phase jump (ϕ_{SAG}), with a target load voltage of 0.8 p.u. and a 0.8 p.u. load current.

not indicate what phase jumps are associated with the measured sags. Bollen et al. [65] note that whilst phase jumps over 10° are quite plausible, the existence of phase jumps "exceeding 45° seems very unlikely". Therefore, assuming that the majority of phase jumps associated with the voltage sags are less than 30°, the proposed UCPC with 0.5 p.u. series and shunt power ratings is expected to be able to provide effective supply quality compensation for consumer equipment for 85-90% of voltage sag events.

3.4.7 Series-Shunt Orientation Options

So far it has been assumed that the series-shunt orientation places the shunt converter upstream of the series converter. An alternative option is to place the shunt converter downstream of the series converter. This has the advantage for the shunt converter of a linear relationship between the required power and the demanded current, as the downstream voltage should always be kept at 1 p.u. magnitude by the series converter. However, the disadvantage is that the series converter must now be rated to handle both the load current, and the shunt converter current. For a series converter with 0.5 p.u. power rating, the design current is no longer 1 p.u. (as it was previously), and the resulting voltage injection capacity is therefore reduced to be much lower than 0.5 p.u.. Elmitwally et al. [89] showed that the difference in Power Quality compensation characteristics between the two orientation options is very minimal. For these reasons, placement of the shunt downstream of the series converter is not considered further in this research.

3.5 Network Placement of the UCPC

The diverse nature of distribution systems means that even if Power Quality problems are improved at one point in the network, this will not necessarily provide adequate compensation for the surrounding loads – irrespective of the effectiveness of the unit at its own installation point. Therefore, even though the focus of this work is on the control of one UCPC, it is important to identify situations where the system is a viable Custom Power alternative. Placement of the UCPC is considered here via a qualitative discussion using an example distribution system (Figure 3.11). The radial power system is constructed to suit the discussions, but incorporates actual parameter values and situations found in distribution networks in outer metropolitan Melbourne, Australia [104]. For this discussion the UCPC is initially assumed to create a pure sinusoidal voltage at its load side voltage terminals and a sinusoidal supply current (how to achieve this task discussed further in Chapters 4 and 5).

The loads in the Figure 3.11 are separated into four subgroups:

1. A residential load group (Load 1);
2. A single MV customer with a large motor load (Load 2);
3. A mainly commercial load group which draws high levels of harmonic currents and is sensitive to flicker and sags (Load 3); and
4. A mainly industrial area which includes flicker producing arc furnaces, and has processes which are sensitive to sags as well as motors whose aging profile is sensitive to voltage unbalance and voltage harmonics (Load 4).

The fault (and hence sag) conditions considered are:

1. Single-phase impedance faults to ground due to trees close to the long 10km feeder F4 (Fault 1);
2. A three-phase short circuit fault on the feeder F10 (Fault 2);
3. a two-phase fault on the spur off feeder F4 (Fault 3); and
4. Sags due to faults in the transmission system.

A number of UCPC placements have been considered, and are identified by UCPC 1 through UCPC 6 in Figure 3.11. Each placement option is discussed in turn below.

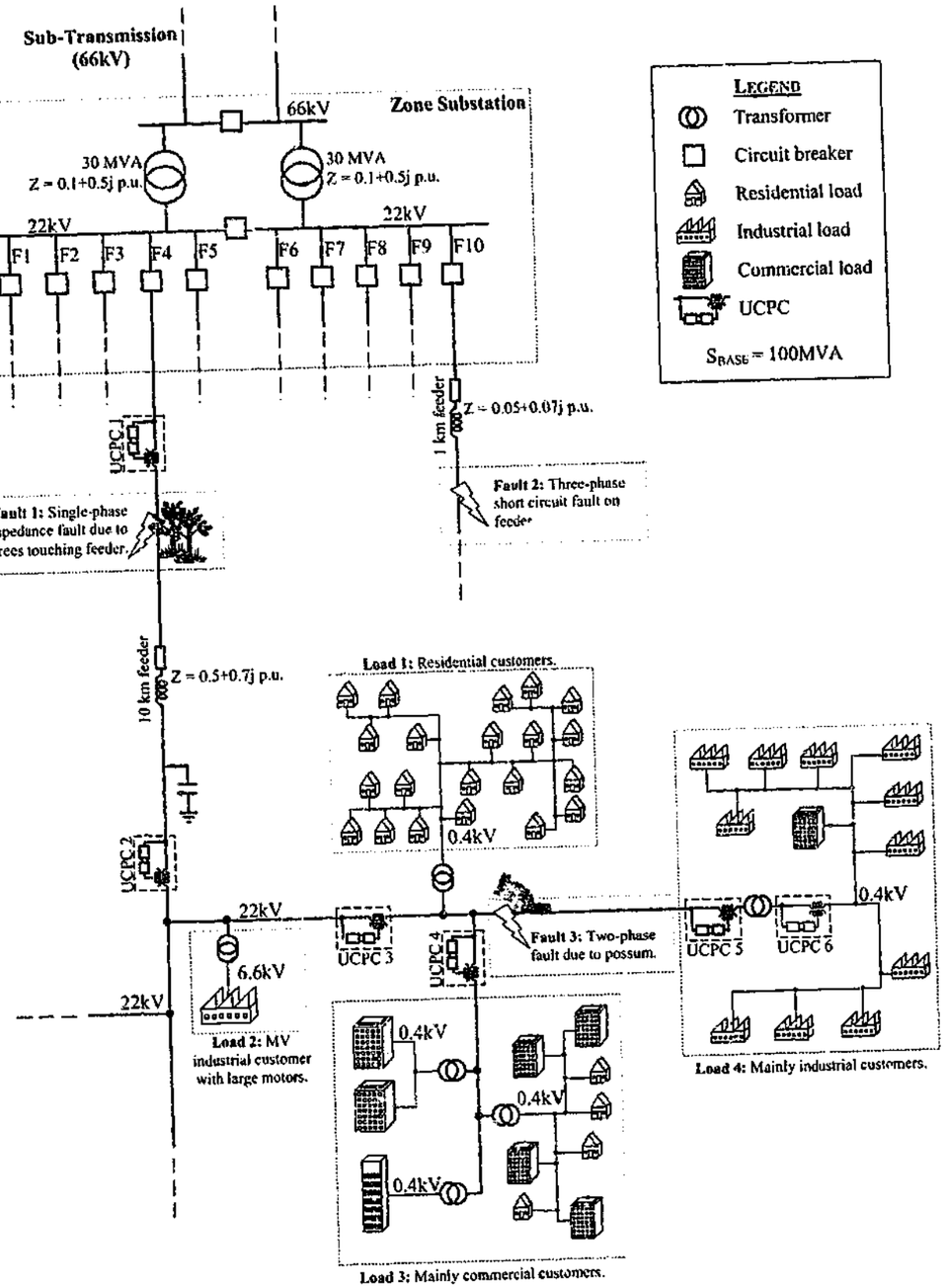


Figure 3.11: Placement options for a UCPC in an example distribution system.

The placement of UCPC 1 is unsuitable for many reasons. Firstly, the device would provide no sag protection for the commonly occurring downstream faults on the long feeder (Fault 1), and its internal protection mechanism is likely to remove the UCPC from operation during these events. (The fault level at the point of installation is a major design consideration for the protection elements of the series portion of the UCPC. Higher fault current levels may increase the overall cost of the device, and should be taken into account. This is discussed further in Chapter 7.) The large harmonic currents from Loads 3 and 4 will create a significant harmonic voltage drop down the feeder, which will work against any benefits of the harmonic voltage compensation as seen further downstream. This also applies for flicker and unbalanced loads. Finally, one major advantage of this placement would be regulation of the fundamental line voltage of the feeder. However, there are likely to be existing automatic tap changers in the Zone Substation that can perform this role, and there are also cheaper alternatives for purely fundamental voltage regulation.

Of the remaining placement options considered, UCPC 2 requires the largest power rating as it is located on the feeder itself, with UCPC 3, which is located on the feeder spur, being the second largest. The remaining locations UCPCs 4 through 6 are located on load groups and have the smallest ratings. If the implementation of a UCPC of the required rating of UCPC 2 is feasible (if not already, advancing technology dictates that it soon will be), then this device would protect all the loads on the entire feeder from all sags (as deep as 0.5 p.u.) created on the upstream feeder (Fault 1), adjacent feeder (Fault 2), as well as any sags originating from the transmission system. However, it may not protect loads from sags caused by the starting currents of the large motors in Load 2. This protection would depend on whether the installed UCPC has a high enough rating to handle these currents, and also on the voltage drop caused by these currents across the impedance between the UCPC and the motor loads. With a relatively close proximity to the load groups 1 to 4, the ability of the UCPC to supply a harmonic/unbalance/flicker free voltage to these loads is quite promising (depending on the impedance between it and the loads). The main problem lies with its adequate protection from the other loads on the feeder (shown by the dashed lines). If these are not closely located, then justification for this large power rating to cover all the loads diminishes, and placement on just the spur may be more beneficial. Due to the long feeder, the voltage regulation capacity of the UCPC is also quite attractive.

Note that no option has been provided for installation to protect the purely residential consumers. This stems from the reality that, unless mandated to do so by the regulating body,

it is unlikely that this load group can create any financial justification for a sole installation. The location of UCPC 3 has the advantage of protecting this residential load group, and by also protecting the commercial and industrial consumers, a financial justification for the installation may be achieved. The location is close enough to Loads 1, 3 and 4 to provide adequate voltage supply quality compensation (especially to Loads 1 and 4), and can also protect these loads from sags caused by the motor starting currents in Load 2. This positioning isolates the Power Quality problems of the load group from each other by directly regulating the common voltage point they share. Thus, the flicker from the industrial load should not affect the commercial buildings, and the harmonic and unbalanced current from the commercial buildings should not affect the motor loads in the industrial load group. UCPC 3 is still ineffective for downstream faults (e.g. Fault 3), but such faults have a much smaller probability in comparison to the upstream faults listed in this particular example.

Locations UCPC 4 and 5 both protect a particular load group, and will therefore have much lower individual power requirements than UCPC 3. Their location is also closer (although only marginally) to the loads and hence should provide slightly better voltage regulation. Location UCPC 4 protects the commercial buildings from nearby flicker sources and sags, and also protects the nearby loads from voltage harmonics caused by the non-linear commercial loads (i.e. PCs, VSDs in air-conditioning units, lighting ballasts, etc.). Location UCPC 5 protects the industrial loads from voltage harmonics, unbalance and sags, and also removes any harmonic supply current distortion caused by the industrial loads.

Location UCPC 6 is positioned similarly to UCPC 5, but differs in that it is a LV installation (not MV, as have been considered to date). The per unit losses for a LV installation will be slightly larger than those for a MV device (primarily due to the increased transformer losses), and the installation of one MV device is likely to be cheaper than many LV devices [60]. However, the advantages of a LV version is that lower short circuit fault levels make protection of the UCPC easier and cheaper, and the voltage isolation costs and size are significantly reduced. Whilst a LV connection will provide the same types of compensation as listed for UCPC 5 (the MV location), care must be taken to ensure that the impedance between the installation point and any non-linear voltage stiff loads (i.e. diode rectifiers with capacitive loads) is not too small. If the low voltage connection point is regulated to be a perfect sinusoid, then this will create a near zero impedance condition, and excellent voltage profiles are seen by the consumers load. The consequence is that these voltage stiff loads may demand significantly higher harmonic currents. As the shunt active filter attempts to compensate for these currents to protect the upstream

load, other downstream problems may arise. (Note that this situation will arise with any device that compensates to create a highly sinusoidal voltage close to these loads.) For the UCPC the finite gain control of the series converter (Chapter 5) will limit this effect, but the obvious compromise will be a reduction in load voltage harmonic compensation capabilities.

The final option is for installation for a single customer (not shown). The outcome would be similar to the installations at locations UCPC 4 through 6 depending on the type of consumer load and the installation voltage. However, in this case all of the cost must be worn by the single consumer, and the range of Power Quality problems must be severe enough to warrant such expenditure.

In summary, the placement of the UCPC can be considered on the basis of technical and financial considerations. The primary technical considerations are to ensure that the UCPC is placed downstream of high probability fault locations, and is positioned such that the impedance between the UCPC and the protected loads is kept to a minimum – especially when problematic loads are sharing the same connection. The financial considerations mandate that the placement be such that the value added benefits of the reduced Power Quality problems are larger than the cost perceived by the consumers (unless mandated by a regulating body). This will create a means to pass some of the financial costs back onto the consumers. Therefore, a location which protects large numbers of *non*-residential consumers, which are particularly sensitive to Power Quality problems, will be the logical solution. Higher power MV installations will likely provide this broad coverage and are the more cost effective solution on a kVA and protected consumer basis (compared to low power LV installations).

Therefore, UCPC locations 3 through to 5 can be seen to be the most attractive installation points in this example. The exact location would depend on the severity of the particular Power Quality problems and the willingness of the consumers to pay more for value added power services. Furthermore, the broad compensation capabilities of the UCPC help justify its existence, but care should be taken to consider that if only one particular Power Quality problem exists, there will most likely be a cheaper (and just as adequate) alternative solution that should be explored.

3.6 Summary

This chapter has presented the Universal Custom Power Conditioner (UCPC) which uses the active series-shunt topology to compensate for Power Quality event types 2 through 6, as discussed in Chapter 2. The separate objectives of the series and shunt converters have been outlined, and the input control parameters investigated and chosen. The physical components required to complete the UCPC have been listed, and the series and shunt ratings have been investigated to determine the limits on the depth of Power Quality problems for which the chosen ratings can compensate. Finally, placement options for the UCPC in a radial distribution system have been discussed to highlight the more beneficial usage conditions of the device, and to review the limitations of this type of system.

Chapter 4

Control of the UCPC System

[1]The series-shunt UCPC converter topology, introduced in Chapter 2 and then developed in more detail in Chapter 3, provides the most flexible alternative for the compensation of Power Quality problems. Chapter 3 has considered the application of the UCPC in terms of its overall objectives, its physical structure, general control requirements, ratings, and alternatives for its placements in an electrical distribution network.

This chapter now develops in more detail an appropriate control scheme for the UCPC to achieve these objectives. Two schemes are explored – one for the series converter and one for the shunt converter of the UCPC. For the series converter, the control scheme applies existing synchronous frame control concepts that have been used previously in other Custom Power topologies. A control model is developed and the effects of the system's physical parameters on this model are investigated. For the shunt converter, three primary control functions are established: the current reference signal generation, the dc-bus voltage controller, and the inner deadbeat digital current controller (based on a Predictive Current Regulation (PCR) strategy). The use of a synchronous reference frame signal extraction system is also presented. It should be noted that while the choice and implementation of the inner PCR current regulator is discussed for completeness, there is a wealth of literature already established in the field for this type of controller, and hence it does not form a significant part of the work presented in this thesis.

This chapter assumes the use of traditional linear synchronous reference frame controllers. However, the use of such systems is limited (if not impossible) with single-phase systems. Hence, Chapter 5 investigates the replacement of these controllers with an equivalent stationary frame version that also allow for single-phase operation as well as reduced implementation complexity.

[1]The material in this chapter was first published in part as:
M. J. Newman and D. G. Holmes, "A Universal Custom Power Conditioner with Selective Harmonic Voltage Compensation", *in Conf. Rec. IEEE/IECON*, Seville, Spain, Nov., 2002.

4.1 UCPC System Analysis

In Chapter 3 control variables were identified that allow the series and shunt converters to be only loosely coupled. Therefore, using these control variables, the series and shunt converter control systems can initially be designed and verified separately, before they are combined into a complete system. The overall proposed control scheme for the UCPC is shown in Figure 4.1.

Each of the control blocks in Figure 4.1 are discussed in this chapter, and the use of conventional synchronous (i.e. rotating) frame d-q controllers is assumed. This allows the overall functionally of the control scheme to be investigated, especially with regards to the physical parameters of the system, such as the grid source and load impedances, transformer ratio, and the LC filters. The limitations of the synchronous frame controllers are also investigated.

For the analysis of the series component of the UCPC, a default set of parameters is initially chosen (refer to Table B.1 of Appendix B). This allows for each physical parameter to be varied separately, to investigate their performance and stability contributions in isolation. This analysis approach is continued in Chapter 5 for the control parameters, after improved stationary frame controllers are discussed. The control aspects of the shunt component of the UCPC are also presented, and the use of a synchronous frame based concepts are again discussed to create a framework for the improved alternatives developed later in Chapter 5.

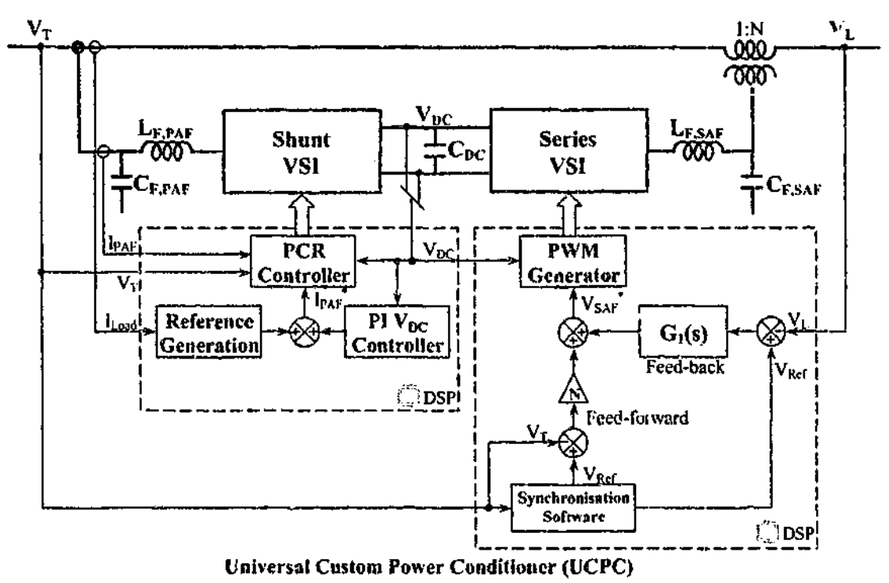


Figure 4.1: Proposed control scheme for the Universal Custom Power Conditioner.

4.2 Series Voltage Injection Control System Overview

In principle the series converter is an ideal voltage source that injects a voltage that is the difference between the terminal voltage and the target reference load voltage to remove all voltage distortion and create a load voltage that is free from Power Quality problems. To achieve this goal, a control system is required to operate the VSI such that, after the LC filter and series transformer, the appropriate voltage injection is achieved. Compensation of voltage sags requires good transient response from the control, while compensation for voltage harmonics and unbalance requires a good steady state response (particularly because of the typically small value voltages that must be compensated).

As discussed in Chapter 2, feed-forward and dead-beat voltage control schemes have been used to achieve good transient results with series converter topologies such as a DVR. However the steady-state responses are not always particularly good – especially for harmonic compensation. Such systems are also quite sensitive to parameter variations in the system, and to practical non-linear effects such as digitization delays, dead-band, and so forth. Feed-back approaches generally use PI d-q controllers synchronized to the fundamental. These systems provide very good steady state performance for the fundamental component, but provide results that are only marginally better than feed-forward approaches for harmonic compensation. Combined feed-forward and feed-back control systems have been proposed for DVRs, but only to provide zero steady-state fundamental error.

In this research a combined approach is presented that uses a feed-forward controller to achieve the transient voltage control requirements of the UCPC, and a feed-back controller to greatly reduce the steady-state voltage error in both the fundamental and harmonics. Figure 4.2 shows how these controllers work together to control the series converter. Their detailed designs will now be discussed in the following sections.

4.2.1 Supply Voltage Feed-forward

The feed-forward controller presented here to compensate for terminal supply voltage (V_T) distortion is similar to the DVR schemes reviewed in Chapter 2, bus has two major differences. Firstly, no sag detection mechanism is required as the UCPC operates continuously. This is in contrast to typical DVR schemes that only compensate for transient variations and hence only operate during these times. Secondly, the actual sag compensation scheme is chosen to be a combination of the Pre-sag and In-phase schemes that have been separately applied to DVR control. Both these schemes have relatively similar real power and voltage rating requirements,

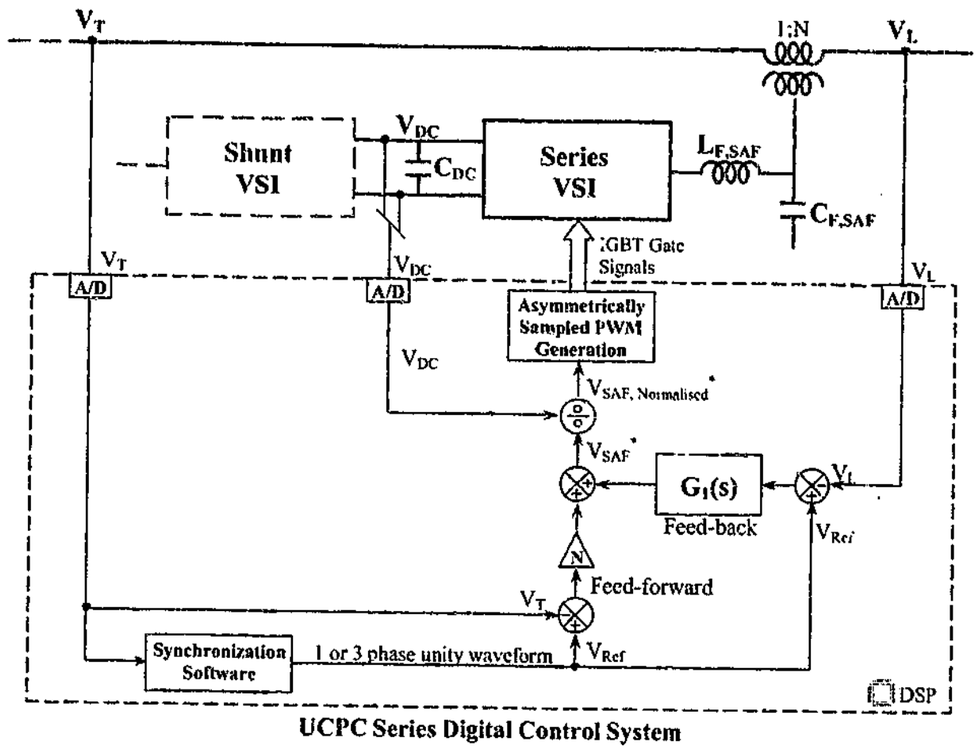


Figure 4.2: Components of the series injection control system.

although they have slight performance differences depending on the depth and phase jump of the voltage sag. However, since the need for a sag detection mechanism has been removed, the phase relationship of the series injection to the supply voltage must be maintained by a new strategy. DVR schemes either freeze the injection phase at the start of the voltage sag for Pre-sag compensation, or quickly adapt to the new phase situation for In-phase compensation. Without sag detection, freezing is not possible, and yet the initial phase jump associated with the In-phase option is not desirable. The solution developed here is to create a combination of the two schemes by slowing the phase synchronization slew rate such that, when the supply voltage phase varies, the converter phase reference only follows at a rate of change acceptable to the load. This means that initially the Pre-sag voltage phase will be retained, but it will then slowly vary until the In-phase voltage phase is achieved. A further advantage of this approach is that it allows for slow variations in the system phase to be accommodated without additions to the overall control scheme.

The feed-forward controller, illustrated in Figure 4.2, is implemented by firstly subtracting

the measured supply terminal voltage from the reference voltage (which is created using the synchronization scheme mentioned above). To minimize excitation of the resonance created by the LC filter, a rate limiter is applied to the signal, but this rate is kept high enough so that the effect on the transient response is minimal. (The LC resonance effect is discussed in more detail later in this section.) After the turns ratio of the transformer is accounted for, the resultant signal is added to the feed-back output, and the result (V_{SAF}*) is passed on to the PWM generation portion of the controller[2]. To ensure that the required volts-seconds is achieved at the output of the converter, the dc-bus voltage is also measured and incorporated into the calculation of the PWM values. It should be noted that small voltage drops across the filter inductance and transformer are not accounted for in this strategy. However, these small errors will be removed by the feed-back portion of the controller.

The result of this part of the overall control scheme is a controller that will automatically shift with variations in the power system phase (caused by voltage sags and other slow variations in the system frequency); compensate for sags with good transient response and no phase jump; and maintain continuous fundamental voltage regulation. However, due to the limitations of the feed-forward approach for smaller errors, the scheme will only provide limited voltage unbalance compensation, flicker compensation, and voltage harmonic compensation (particularly for larger magnitude and lower frequency components).

4.2.2 Load Voltage Selective Feed-back

The feed-forward control component provides the required transient response for compensation of sags and other sharp variations in voltage. A feed-back controller is now developed to remove any steady-state and harmonic voltage error in the load voltage. Conventionally, a PI d-q feed-back controller is applied to the load voltage to achieve this goal. However, this type of controller will remove positive sequence fundamental error since it has a very high gain for this component, but it will have only a minimal effect on other errors due to the lower gain that it has for the remaining error components. To remove the voltage unbalance for a three-wire system a negative sequence PI d-q controller can be used, but this still does not remove most of the voltage harmonic error.

Bhattacharya et al. [52] proposed the use of separate PI d-q controllers to act on each of a number of supply current harmonics, and hence controlled the series topology to act as a current

[2]Note from Figure 4.2 that a positive output from the VSI will cause an increase in load voltage (i.e. the positive terminal of the series injection in connected to the load). This sign notation will be used in the remainder of this thesis and should be assumed if not explicitly noted.

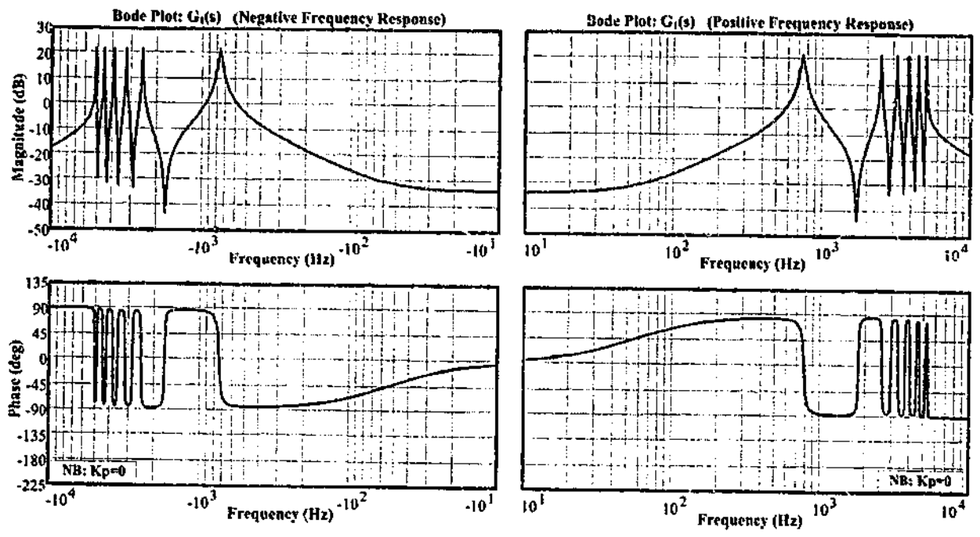


Figure 4.3: Simulated positive and negative bode plots for the series feed-back controller, $G_1(s)$. (Proportional gain equals zero).

source to achieve harmonic isolation. This approach achieves virtually zero steady-state error for each of the selected harmonic signals. A similar selective approach was also used by Mattavelli et al. [17] in an UPS in-line topology for voltage control. Thus, it seems attractive to use the same approach with the series topology for voltage control. However, to allow for all sources of harmonic distortion, separate positive and negative PI d-q controllers would be required for each of the selected harmonics. These controllers would provide the frequency response shown in Figure 4.3. But, for selected components of the fundamental and the 5^{th}, 7^{th}, 9^{th}, 11^{th} and 13^{th} voltage harmonics, a total of twelve separate PI d-q controllers would be required, and this is very computationally expensive. Furthermore, PI d-q controllers are designed for use on three-phase systems, and cannot be used on single-phase systems.

Chapter 5 explores the use of stationary frame equivalent controllers to alleviate these difficulties, and results in a feed-back controller for the series VSI, $G_1(s)$, given by

$$G_1(s) = K_P + \sum_{n \in M} 2K_{I,n}\omega_c \frac{s+\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_n^2} \text{ where: } M = \{1,5,7,9,11,13\}. \qquad (4.1)$$

However, for the following discussions in this chapter this development can be temporarily put to one side, and the use of both positive and negative PI d-q controllers to compensate for all the selected frequency components is assumed.

4.2.3 Design and Stability Analysis

The design of the series controller for the UCPC involves many hardware and software parameters that must be chosen during the design stage, as well as other parameters that must be identified, but are not under the control of the designer or installer (e.g. the power grid system and load parameters). The major hardware parameters of interest are detailed in Table 4.1. All of these parameters can in some way affect the stability of the control system, and their influence will now be investigated. The physical parameter values that provide the most stable performance will then be selected. (Parameter determination for the control variables using a similar investigation approach is left until Chapter 5, where the controller is discussed in more detail.) Only linear analysis of the system is pursued in this section, and therefore it is assumed that the VSI does not operate into its over-modulation region. Also, other non-linear effects (such as the digital implementation and magnetic saturation) are neglected for the moment.

Symbol	Name	Parameter Type
V_{Ref}	Reference voltage	Input - $R(s)$
V_L	Load voltage	Output - $C(s)$
V_S	Source voltage	Disturbance - $D(s)$
K_P	Proportional gain	Controller design parameter
K_I	Integral gain	Controller design parameter
N	Transformer ratio	Fixed by maximum desired series injection
L_F	Filter inductance	VSI design parameter
R_{LF}	Filter inductor resistance	VSI design parameter
C_F	Filter capacitance	VSI design parameter
R_{CF}	Filter capacitor resistance	VSI design parameter
L_S	Source inductance	Varying distribution system parameter
R_S	Source resistance	Varying distribution system parameter
L_{Load}	Load inductance	Varying distribution system parameter
R_{Load}	Load resistance	Varying distribution system parameter

Table 4.1: Series model parameter types.

An s-domain analytical model is used for the design and stability analysis. Figure 4.4 shows

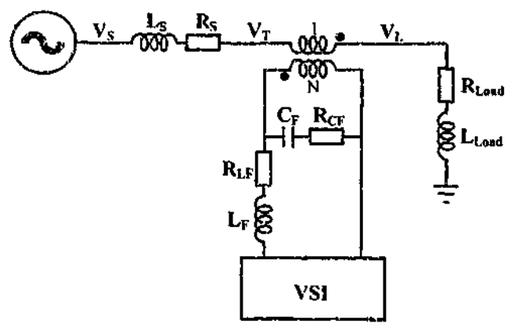


Figure 4.4: Electrical elements and labels used for stability analysis of the series control system.

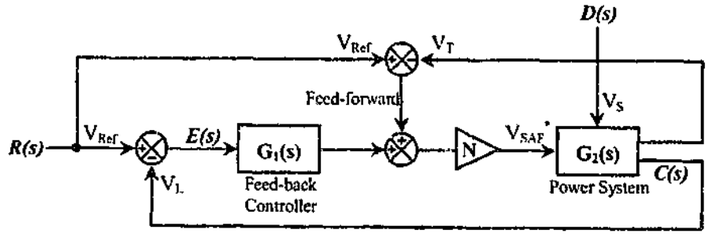


Figure 4.5: Control block overview of the series injection control system and model of the power system used for stability analysis.

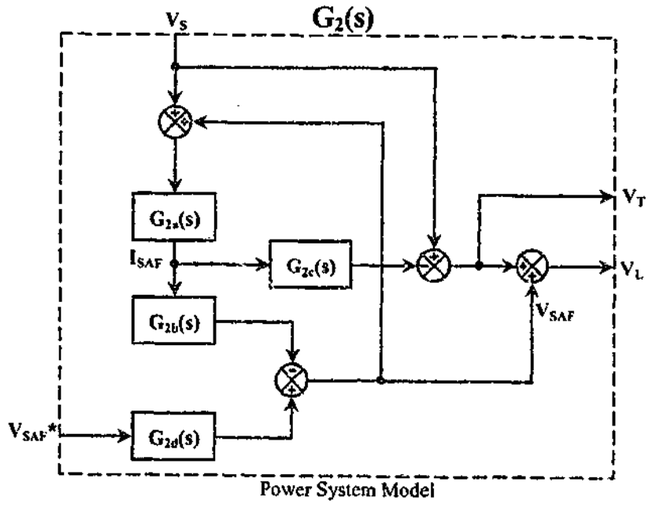


Figure 4.6: Power system model used for stability analysis.

the electrical system schematic and terminology used for the model, and the control model block layouts are presented in Figures 4.5 and 4.6. The VSI power stage can be modeled as simply a unity gain block due to the dc-bus voltage compensation included within the controller, which ensures that the required volt-seconds output is generated irrespective of the instantaneous dc-bus voltage.

The input variable to the control system, $R(s)$, is the three-phase (or single-phase) reference load voltage (V_{Ref}), and the output variable, $C(s)$, is the load voltage (V_L). The source voltage (V_S) is the disturbance of the system, $D(s)$. Note the distinction between the source voltage (V_S) and the terminal supply voltage (V_T) shown in Figure 4.4. The source voltage is located before the Thévenin equivalent supply impedance, Z_S, while the terminal supply voltage is the point of upstream connection of the UCPC (i.e. on the other side of Z_S).

The power system, $G_2(s)$, is also developed from the schematic in Figure 4.4. The inputs to the model are the output of the VSI (before the LC filter) together with the source voltage. The model outputs define both the terminal supply voltage and the load voltage of the system, which are fed back into the series controller (Figure 4.5). To implement the power system as a two port block, the model was broken into four parts (Figure 4.6), defined as:

$$G_{2a}(s) = \frac{1}{s.N\,(L_S + L_{Load}) + N\,(R_S + R_{Load})} \quad , \tag{4.2}$$

$$G_{2b}(s) = \frac{s^2.L_F.R_{CF}.C_F + s.(L_F + R_{CF}.R_{LF}.C_F) + R_{LF}}{s^2.N.C_F.L_F + s.N.C_F.(R_{LF} + R_{CF}) + N} \quad , \tag{4.3}$$

$$G_{2c}(s) = s.N.L_S + N.R_S \quad , \text{ and} \tag{4.4}$$

$$G_{2d}(s) = \frac{s.R_{CF}.C_F + 1}{s^2.N.C_F.L_F + s.N.C_F.(R_{LF} + R_{CF}) + N} \quad . \tag{4.5}$$

Due to the number of variables, as well as the wide range of possible values for each, it is not feasible (or useful) to investigate all possible parameter variations. Instead, default values are assigned to each parameter, and then each parameter variation is investigated in isolation. This allows the effect of each individual parameter to be qualified, as well as providing an approximate quantification of the effect under the set conditions. A full set of the default values that were used is provided in Table B.1 of Appendix B, and they are also noted throughout the following

text as required.

Design of the LC filter elements: L_F, R_{LF}, C_F & R_{CF}

Filters are typically used as interfaces between the VSI and the system to attenuate the PWM switching harmonics, whilst passing through the lower frequency target waveform. For active filter systems the design typically becomes a compromise between good filtering of the switching harmonics, and the effect of the filter on the target low frequency harmonics (which can be less than one decade away from the switching frequency). To achieve the sharp roll-off characteristic required, second order LC (inductor-capacitor) filters are commonly used. An LC filter is a minimum requirement for the series topology, as the series transformer approximates the performance of a current transformer (especially for higher values of N) and simple first order inductive filters typically don't operate well into this type of load. However, LC filters create resonance conditions around the break point of the filter, which must be taken into account during the design. The control model shows that this resonance effect can cause a significant amplification of disturbances from the supply voltage to the load voltage, and it is therefore an important design consideration in this section. The four parameters of the LC filter design are: the filter inductance (L_F), the resistance of the filter inductor (R_{LF}), the filter capacitance (C_F), and the series resistance of the filter capacitor (R_{CF}).

For many high performance applications the filter inductance is designed to be relatively small to minimize the voltage drop error between the converter output and the connection point, and to make the output voltage more immune from variations in the load currents. However, inductor sizes cannot be too small because of the limits caused by the high ripple currents that are then created, and hence this limit is used as the design criterion in this work. An initial default value of 2 mH was chosen for the filter inductance as this leads to acceptable maximum ripple currents in the LV experimental converter system (see Chapter 9).

The filter capacitance is initially selected based on the filter break point required, and then adjusted to meet the stability considerations and commercially available values. For this research harmonics up to the 13^{th} are compensated (650 Hz), and therefore a break point in the region of 1.0 kHz to 1.5 kHz is desired (i.e. above the highest target frequency, but sufficiently below the switching frequency to achieve acceptable attenuation). Using the approximate break point equation

$$f_{break} = \frac{1}{2\pi\sqrt{LC}} \tag{4.6}$$

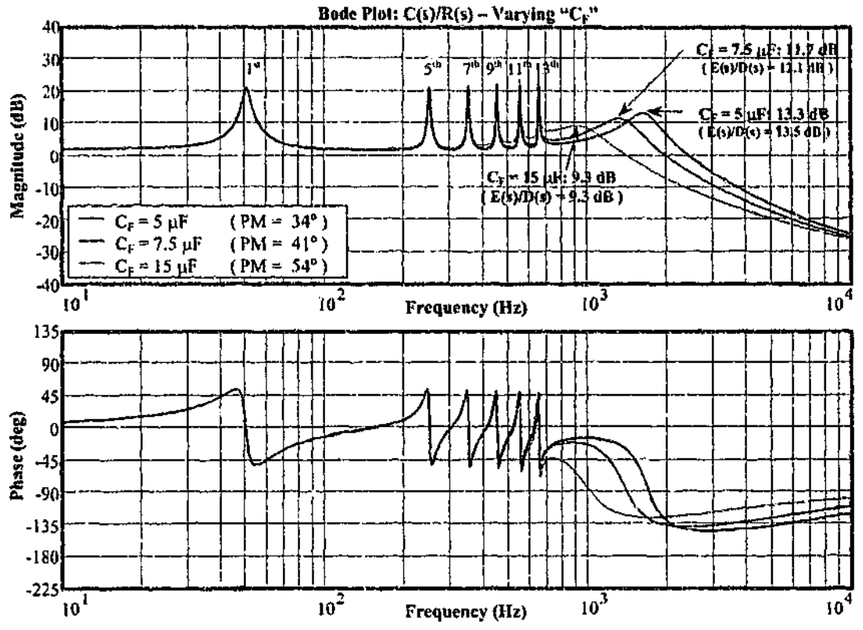


Figure 4.7: Bode plots showing effect of varying the filter capacitance (C_F). (Refer to Table B.1 in Appendix B for remaining default parameters.)

a capacitance of 7.5 μF was initially chosen as a default value to give a resonant frequency of approximately 1.3 kHz. However, is should be noted that the actual break point can vary from (4.6) due to the loading effect of the other parameters in the system (such as load and source impedances). The effect of varying C_F is shown in Figure 4.7[3], with values on either side of the default value simulated. The larger capacitance (and therefore lower frequency) has a lower resonant gain of 9.3-dB, compared to 11.7-dB and 13.3-dB for the smaller capacitances. (The larger capacitance of 15 μF was therefore adopted for the system value.)

The next consideration is the resonance damping effect of the resistance in both the filter inductor and capacitor. Since the full converter current passes through the filter inductor, and a small impedance is preferred for the filter inductor as discussed above, the resistance of the inductor must be kept small. The actual value is determined by the design of the inductor (i.e. losses in the copper and magnetic material). A default value of 50 $m\Omega$ (i.e. 0.05 Ω) was used for the following analysis based on measurements of the installed LV inductors (see Chapter 9).

[3]For all Bode plots (i.e. open loop system response) presented in this thesis, the response C(s)/R(s) is equivalent to C(s)/E(s). This is because with the feed-back set to zero, the reference, R(s), and the error, E(s), are the same.

This resistance is too low to adequately damp the *LC* filter.

The only viable option for passive damping is therefore the series resistance of the capacitor. This is the approach used by various researchers [69] [81] [89] [105] [106] [107], and since the equivalent series resistance (ESR) of capacitors is typically quite small, an external series resistor is required. Figure 4.8 presents a bode plot of the system to illustrate the dramatic effect series resistance has on the resonance damping.

For the targeted harmonic frequencies a high loop gain is ideal as this gain is derived from the controller, but high loop gain at the filter break point (derived from the plant) is not desirable, and is likely to cause amplification of voltage supply disturbances in this frequency region. This is shown in the disturbance error plot of E(s)/D(s) in Figure 4.9[4]. The controller's gains become deep notches at the selected frequencies (i.e. very little disturbance remaining), but the *LC* resonance clearly creates a large source of amplification for disturbances near its frequency. With no resistance the loop gain is 34.6-dB (39.0-dB disturbance error) at the filter break point, but it drops to only 11.7-dB (12.1-dB disturbance error) with 4.7 Ω inserted. The phase margin (PM) is also significantly improved, from unstable with no resistance, to 41° with 4.7 Ω inserted. However, the use of the resistance has two disadvantages. Firstly, high frequency attenuation becomes poorer as the resistance increases, because the ripple current creates a directly proportional voltage component across the resistor (Figure 4.8). Secondly, the ripple current through the resistor creates additional losses in the system. Since performance is more important for this type of system, this small additional power loss is acceptable as a design compromise. (Chapter 5 explores the use of an active damping scheme to provide increased damping and stability, as well as allowing the size of R_{CF} to be minimized as much as possible. However, as the active damping scheme acts as a virtual resistor, the above analysis will still hold.)

[4]Note that for this system the disturbance error, E(s)/D(s), has equivalent magnitude to the output disturbance ratio, C(s)/D(s), as the feed-back path is unity and the reference input is set to zero for the analysis.

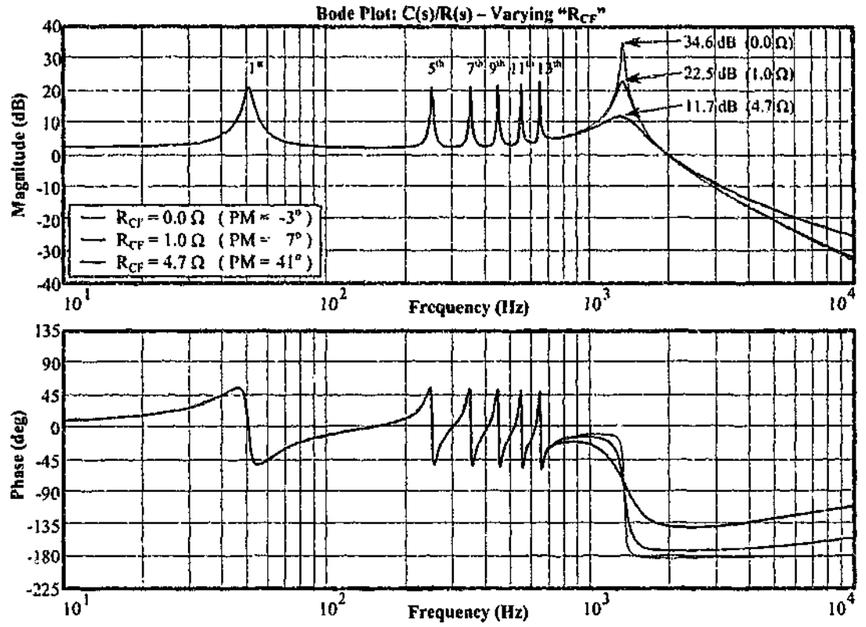


Figure 4.8: Bode plots showing effect of varying the filter capacitor series resistance (R_{CF}). (Refer to Table B.1 in Appendix B for remaining default parameters.)

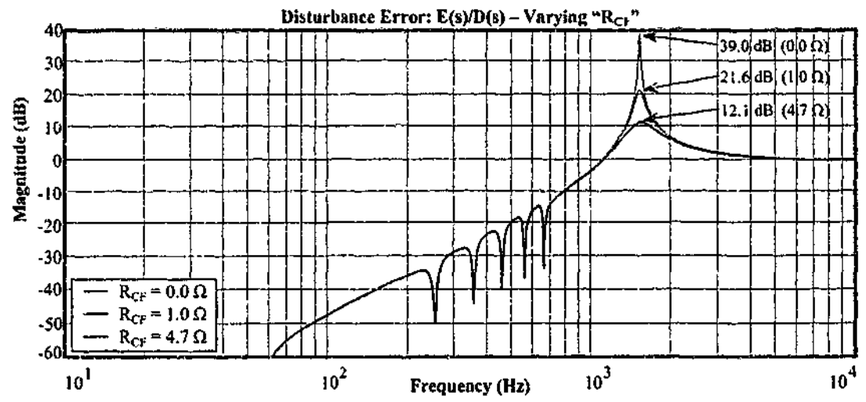


Figure 4.9: Disturbance error $E(s)/D(s)$ showing effect of varying the filter capacitor series resistance (R_{CF}). (Refer to Table B.1 in Appendix B for remaining default parameters.)

Effect of load and source impedances

Load and source impedances are closely related to each other with respect to their effect on the system loop gain. This is primarily due to their physical arrangement in the circuit. Their influence can also be seen in Equation 4.2 as the direct addition of the inductance and resistance values in the denominator. The default values for the load elements are 10 Ω (i.e. R_{Load}) and 5 mH (i.e. L_{Load}), while the default source impedance is 0.5 Ω and 1 mH[5].

Figure 4.10 shows that a purely resistive load (i.e. $L_{Load} = 0$) creates significant damping of the LC resonance compared to a combined inductive-resistive load. Conversely, the larger the inductive component, the smaller the damping. Not unexpectedly, the phase margin (PM) is once again affected by the reduction in damping, with the greatest PM created with a purely resistive load. As expected, without any inductive load, damping and PM improve as the load resistance gets smaller. Thus, stability improves during heavier real power load conditions. Finally, since the source impedance is generally significantly smaller than the load impedance, the source impedance usually has little effect on the stability and performance of the system.

Design and effect of the transformer ratio: N

As previously discussed, the target series injection capacity for the UCPC in this research is 0.5 p.u., and thus for a LV installation a default transformer ratio (N) of two is used. As the ratio varies, so does the load impedance seen on the converter side of the transformer. Figure 4.11 shows that for larger the injection capacities, the effective impedance seen by the converter reduces, the system is more damped, and the phase margin (PM) improves.

[5]The source impedance value also includes the equivalent leakage reactance and winding resistance of the transformer. The magnetising inductance of the transformer is assumed to be large and is therefore ignored.

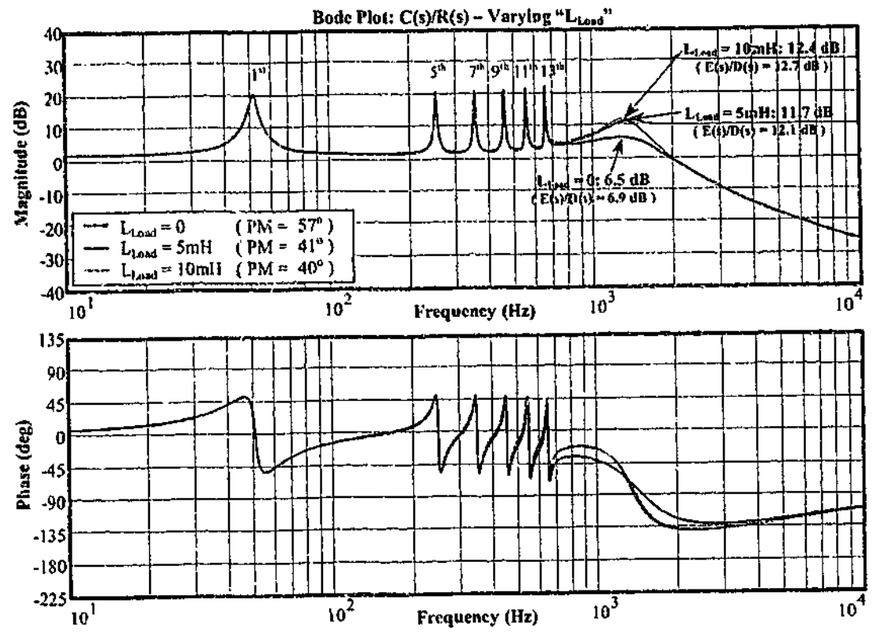


Figure 4.10: Bode plots showing effect of varying the load inductance (L_{Load}).

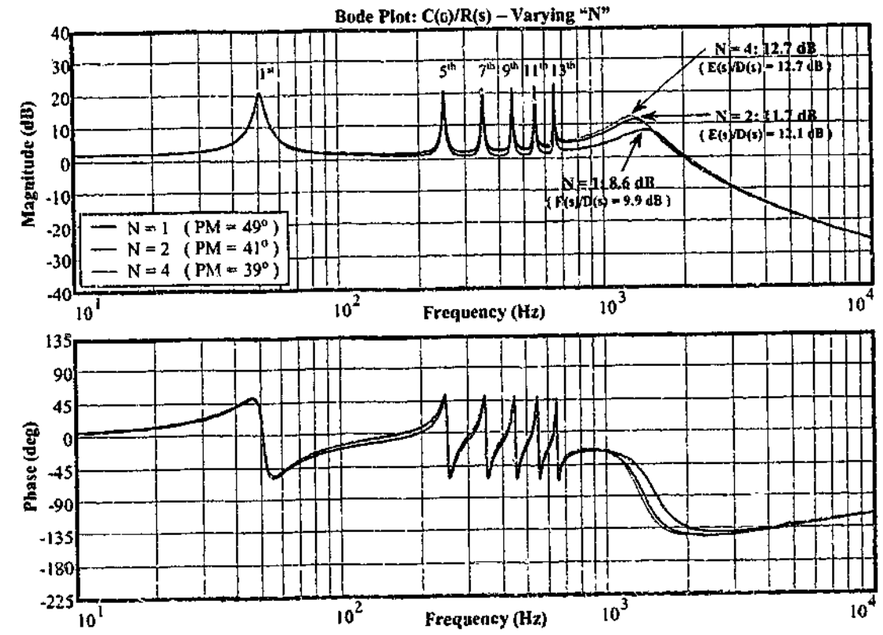


Figure 4.11: Bode plots showing effect of varying the transformer ratio (N).

Design summary

The analysis of the series converter section has shown that at the lower target frequencies, loop gains of the system are influenced mainly by the linear control gains, while at and beyond the *LC* filter break point, the loop gain is dominated by the physical system elements. The primary design concern for this filter is to minimize the disturbance gain around its resonant frequency.

The filter inductance was chosen based on the maximum allowable ripple current and converter rating, to minimize the effect of the voltage drop across the inductor. An initial default filter capacitance of 7.5 μF was selected, and was then revised to 15 μF to provide a lower disturbance gain around the *LC* break point frequency. A damping resistance of 4.7 Ω in series with the filter capacitor was then introduced to provide adequate damping, although the losses produced by the resistor are of some concern. (Minimizing this resistance using active damping techniques will be discussed in Chapter 5.) The choice of the transformer N also has a significant influence on the stability of the system, and as the value of N is lowered (i.e. larger series rating) more of the load impedance is reflected to the converter side – providing more damping.

The robustness of the controller to variations in the power system parameters was also investigated. Variations in the source impedance had little effect due to the dominance of the load components. A purely resistive load provides significantly more damping (and therefore better stability) than when combined with an inductive load. However, acceptable stability can still be maintained with an inductive load.

4.3 Shunt Current Injection Control System Overview

The shunt (parallel) portion of the UCPC removes unbalanced fundamental and harmonic currents from the supply, and also provides bi-directional power to the series converter. Figure 4.12 shows the major sections of the control system for the shunt converter, where it can be seen that the controller can be split into four separate blocks: reference current generation, dc-bus regulation, PCR current regulation, and the VSI PWM controller. The reference current (I_{PAF}*) is created from the addition of the target active filter cancellation currents (that cancel unbalanced load and harmonic currents) and the real power sinusoidal current (required to maintain a constant dc-bus voltage to control the power flow to the series converter). This reference current is tracked using a Predictive Current Regulation (PCR) scheme. The current reference generation and current regulation processes are discussed in the following sub-sections.

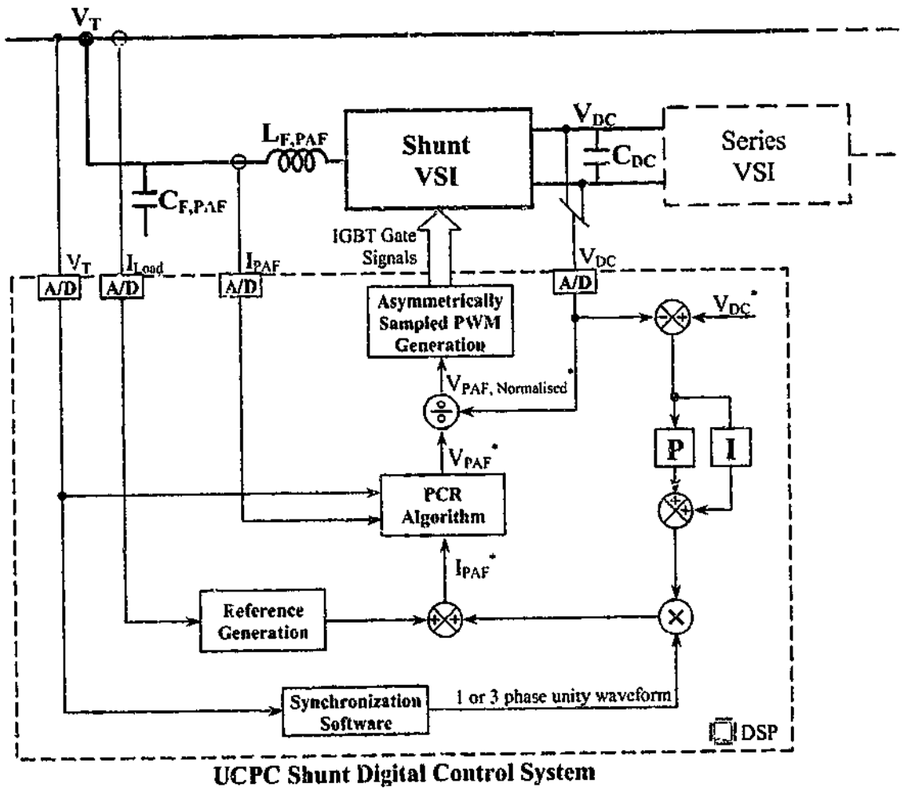


Figure 4.12: Components of the shunt control system.

4.3.1 Reference Signal Creation

Chapter 3 described the control parameters for the shunt converter, and concluded that a purely feed-forward control approach was appropriate due to the highly transient nature expected from the current to be compensated. Unlike the series controller, where the ideal voltage has a fixed magnitude, the current magnitude is dependant on the load and will vary. Therefore, the reference signal for the feed-forward controller must be extracted from the load current. (Recall also that for shunt current compensation the feed-forward parameter is the load, whereas for series voltage compensation the feed-forward parameter is the supply.) The reference generation control block is required to remove the fundamental component from the load current signal, which leaves the unbalance and distortion components remaining. If tracked faithfully by the PCR algorithm, injection of the negated signal will cancel with the unbalance and distortion components in the load current, to achieve a sinusoidal and balanced supply current.

One common method for reference generation in the literature is to shift the signal into the d-q rotating frame, which modulates the fundamental component into a dc quantity. This allows a simple first order high-pass filter to attenuate the dc portion, with a suitable cut off frequency from the harmonic components such that they are minimally affected by the filtering action. For three-wire systems this is given by

$$\begin{bmatrix} y_a(t) \\ y_b(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} \cos(\omega_0 t) & \sin(\omega_0 t) \\ -\sin(\omega_0 t) & \cos(\omega_0 t) \end{bmatrix} \left\{ [h_{dq}(t)] \circledast \left\{ \begin{bmatrix} \cos(\omega_0 t) & -\sin(\omega_0 t) \\ \sin(\omega_0 t) & \cos(\omega_0 t) \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_a(t) \\ x_b(t) \end{bmatrix} \right\} \right\} \tag{4.7}$$

and

$$H_{dq}(s) = \frac{s}{s + \omega_c}, \tag{4.8}$$

where ω_c is the high pass filter cut-off frequency, and ω_0 is the fundamental frequency.

However, this method is not applicable for single-phase systems, is computationally expensive, and requires the generation of multiple sinusoidal references. Therefore, Chapter 5 develops an exact equivalent method for the stationary reference frame which does not require modulation into the rotating d-q frame, is more computationally efficient and has no requirement for synchronized sinusoidal references. Stationary frame approaches for single-phase systems are then also developed.

4.3.2 Predictive Current Regulation (PCR)

For the series converter the transient response requirements were provided by feed-forward injection of the supply voltage error. However, for current regulation in the shunt converter, a direct open-loop feed-forward approach is not possible, because the converter output quantity is voltage, not current. Therefore an inner loop current controller is required to produce the demanded feed-forward reference current described in Section 4.3.1.

There are three basic types of current regulation systems: ramp comparison (i.e. linear PI and similar techniques), hysteresis, and predictive [108]. Linear PI (and PI d-q) can provide an adequate steady-state response, but are not suited to the rapidly changing load currents typically seen by this system. Hysteresis current controllers exhibit excellent transient performance, but implementations in their traditional form are not fully digital. They also exhibit a variable switching frequency which is more difficult to filter. So, the remaining alternative is to use a predictive current controller, which requires some knowledge of the converter parameters to calculate the output voltage required, and to therefore achieve dead-beat control of the target output current.

Predictive controllers have been widely used in the literature for shunt active filtering [109], and have also been used in combination with other linear techniques for additional improved steady state performance [110]. This research uses the direct implementation method proposed by Holmes et al. [111], which is termed the Predictive Current Regulator, or PCR. The algorithm can be used in situations where the back-emf of the connected load is either known and unknown. For active rectifier topologies (such as are used in this research) the back-emf is known, and therefore either version can be used.

Malesani et al. [109] has identified that PCR implementations where the back-emf is known are stable for 100% variations in filter inductance, whereas implementations where the back-emf is estimated are typically only stable for inductance variations of about 20% (or even less) depending on the load parameters. More stable versions (with respect to variations in the filter inductance, $L_{F,PAF}$) of the estimated back-emf scheme are possible, but they are still not as stable as the measured back-emf schemes [109]. Back-emf known PCR is also sensitive to errors in the voltage measurement, and fast voltage variations (because the back-emf is assumed to be relatively constant during the switching cycle), but little work has been done to quantify this effect and to investigate how it affects the stability of the PCR algorithm. However, resolving these issues is a separate research issue, and is not pursued further in this work as the series control portion of the UCPC is of more importance. Hence, the measured back-emf PCR

version was chosen for this research due to its wide stability regions for variance in $L_{F,PAF}$. It should be noted that the following (and future) PCR discussions are included in this thesis for completeness, but are simply the understanding, design, and application of existing knowledge in the field of digital current control and do not represent an original research contribution.

The basic back-emf known form of the PCR algorithm is given by Holmes et al. [111] as:

$$V_{PAF}^*[i] = 4.V_T[i-1] - 2.V_T[i-2] - V_{PAF}^*[i-1] + \frac{L_{F,PAF}}{\Delta T}.(I_{PAF}^*[i+1] - I_{PAF}[i-1]) \tag{4.9}$$

where the input and output parameters are identified in Figure 4.12. For many digital implementations the analog system measurements are sampled during the prior switching period[6] (i.e. during [i-1]), and the algorithm calculation result is then executed during the next switching period (i..e [i]). The resultant target converter current should then be achieved by the end of this second switching period (i.e. at the start of [i+1]). Equation (4.9) includes a linearly extrapolated prediction of the measured back-emf to compensate for this prior sampling, and also predicts the measured current at time [i] using the previously measured current and the applied voltage for the period [i-1,i]. Whilst the back-emf value during the period [i+1] is unknown before the event, it is known that it will typically not vary significantly during a switching period, and hence the use of forward estimation is seen to be adequate.

For active filter applications, the reference current (I_{PAF}^*) is likely to vary significantly from switching cycle to switching cycle, and the prediction choice becomes more important. In (4.9) the reference current is defined as $I_{PAF}^*[i+1]$; which is not causal. Holmes et al. [111] propose three options: **1.** Use the [i-1] sample; **2.** Use a one step forward linear extrapolation to give an estimate for [i]; and **3.** Use a two step forward linear extrapolation to give an estimate for [i+1]. The choice between the alternatives is a compromise between the overshoot (that can occur during rapid changes in the demanded current for options 2 and 3) and errors in the current cancellation due to the phase shift applied (for options 1 and 2).

The cancellation error caused by a single sample delay will increase for higher order harmonics and can be found by the subtraction of one sinusoid from another phase shifted sinusoid,

$$e(t) = \sin(\omega_n t) - \sin(\omega_n t - \omega_n / f_s) \tag{4.10}$$

[6]Note that for the asymetrically sampled system discussed here, the term "switching period" is used to denote half of a full switching cycle.

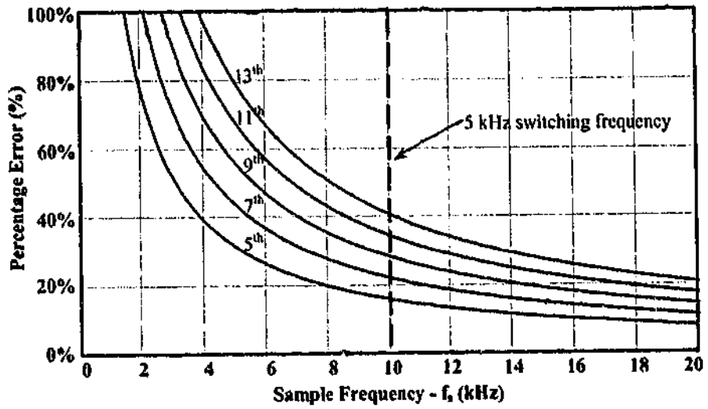


Figure 4.13: Current compensation percentage error (per harmonic) caused by a single sample delay in the cancellation current.

where ω_n is the harmonic frequency (rad/sec), and f_s is the sample frequency (Hz). The peak error (in p.u.) is then found to be

$$|e(t)|_{MAX} = 2\sin(\omega_n / 2f_s) \tag{4.11}$$

The percentage error versus sample frequency for each of the low order harmonics considered in this research is illustrated in Figure 4.13. With a switching frequency of 5 kHz, the cancellation error due to just one sample period is up to 40% (or more) for the higher harmonics, and this is before other errors in the system are accounted for. Therefore, the two step forward extrapolation option was chosen, as this provides a better estimate of the reference current with no sample delay. Substitution of this forward reference estimator into (4.9) gives the final PCR equation used:

$$\begin{aligned} V_{PAF}^*[i] = {} & 4.V_T[i-1] - 2.V_T[i-2] - V_{PAF}^*[i-1] \\ & + \tfrac{L_{F,PAF}}{\Delta T}.(3.I_{PAF}^*[i-1] - 2.I_{PAF}^*[i-2] - I_{PAF}[i-1]) \end{aligned} \tag{4.12}$$

4.4 Summary

This chapter has developed a control strategy for the Universal Custom Power Conditioner (UCPC), which fully exploits the potential functionality of the series-shunt topology. The series converter control strategy utilizes a DVR type technique for voltage regulation and sag compensation, together with an array of selective feed-back linear controllers on the fundamental and selected harmonics. The shunt active filter regulates the dc-bus voltage and compensates for load current harmonics and balance, using a Predictive Current Regulation (PCR) scheme to follow the reference signal.

A design process for the *LC* filter parameters has been proposed, with values for the LV experimental system chosen using the stability model developed. The susceptibility of the UCPC to variations in the load and source impedance was also presented using this model.

In this chapter only an overview of the proposed linear control system has been presented, together with a discussion of the limitations that are associated with common synchronous frame solutions. Chapter 5 proposes the replacement of these types of controllers with stationary frame versions that allow for reduced computation times and offer both three-phase and single-phase operation. Implementation limitations using traditional digitization techniques for these stationary frame controllers are then identified, allowing a further solution to be presented in Chapter 6.

Chapter 5

Stationary Reference Frame Control

[1]The overall control scheme of the UCPC has been developed in Chapter 4. In that development, the use of *d-q* synchronous reference frame controllers was assumed for series converter control and for the harmonic extraction required by the shunt converter control. This allowed for the physical parameters of the system to be explored in detail. However, synchronous *d-q* frame controllers are limited to three-phase systems, are computationally expensive, and require a separate sinusoidal reference signal for each selected frequency. Zmood et al. [112] [113] [114] has proposed the conversion of a conventional *d-q* PI controller into the stationary frame to alleviate these problems. The resultant stationary frame equivalent function is a second order resonant filter with an additional proportional term, and is known as the P+Resonant controller. It should be noted that no equivalent controller has been developed for the synchronous *d-q* frame reference generator required for control of the shunt converter discussed in Chapter 4.

This chapter begins by providing a brief review of P+Resonant control, and then investigates the control parameter design decisions required for its application to the selective harmonic voltage compensation scheme proposed in Chapter 4 for the series converter. Both feed-forward and feed-back series components are analyzed, and an improvement is proposed by the addition of active damping to the controller. The P+Resonant theory is then applied to the high-pass filter reference signal extractor for the shunt converter, to develop an equivalent stationary frame version for both three-phase and single-phase systems. The controllers are verified in simulation, and the practical implementation problems are identified. The digital implementation problems highlighted are then investigated further in Chapter 6.

[1]The material in this chapter was first published in part as:
M. J. Newman, D. N. Zmood, and D. G. Holmes, "Stationary frame harmonic reference generation for active filter systems," *in Conf. Rec. IEEE/APEC*, Dallas, TX, pp. 1054-1060, 2002.
M. J. Newman, D. N. Zmood, and D. G. Holmes, "Stationary frame harmonic reference generation for active filter systems," *IEEE Trans. Ind. Applicat.*, vol. 38, no. 6, Nov./Dec., pp. 1591-1599, 2002.

5.1 Stationary Frame Linear Controllers

The majority of classical control techniques are based around dc systems. The proportional-integral, or PI, linear controller is one such example. An ideal PI controller has a transfer function of

$$H_{PI}(s) = K_P + K_I\frac{1}{s}, \tag{5.1}$$

where K_P is the proportional gain and K_I is the integral gain. For dc systems the integral component provides zero steady-state error control, and the proportional component provides the transient response. A variation to this system is the finite pole integrator given by

$$H_{PI}(s) = K_P + K_I\frac{1}{s+\omega_c}, \tag{5.2}$$

where ω_c is known as the cut-off frequency. For some systems this implementation may be more practical as it is simpler to create digitally, and the finite dc gain provides less wind-up problems under saturation conditions.

Unfortunately, electrical systems typically involve ac and other continuously varying signals – not dc signals [114]. Therefore, steady-state errors will exist if a PI controller is applied directly to an ac system. Park [115] has shown (for ac machine theory) that three-phase signals can be transformed from their existing frame of reference (known as the stationary reference frame) into a rotating frame of reference, which is synchronous to the fundamental frequency (known also as the synchronous reference frame, or SRF). The signals in the rotating frame then become dc quantities, and can therefore use a PI controller to achieve zero steady-state error. The transformation has two stages. Firstly, the three-phase signal is converted into the α-β reference frame, which creates two orthogonal signals (as well as a zero sequence component for four-wire systems). This is known as the 'Clarke Transformation'. Next, the signal is transformed into the d-q rotating reference frame using the 'Park Transformation', so that a PI controller can be implemented on signals which are approximately dc in steady-state. (Note that the term 'approximately' is used, as in practical systems the input signal will have distortions that cause it to deviate from a pure three-phase set of sinusoids, and hence the result in the synchronous frame is not exactly dc.) The transformation is similar to an AM de-modulation process, which frequency shifts the signal to center around dc, but also results in components at the double frequency, $2\omega_0$. However, for three-phase systems the α-β to d-q transformation is constructed such that these double frequency components do not appear in the dc signal due to cancellation

between the phases. (Note that for single-phase systems this cancellation of the double frequency component in the synchronous frame does not occur.) Once the controller action is complete, reverse transformations are then used to frequency shift (i.e. an AM modulation process) the controller output back into the ac stationary frame for execution by the converter modulation controller. For a three-wire system (where the third phase is redundant) the synchronous d-q PI controller is given by

$$\begin{bmatrix} y_a(t) \\ y_b(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} \cos(\omega_0 t) & \sin(\omega_0 t) \\ -\sin(\omega_0 t) & \cos(\omega_0 t) \end{bmatrix} \left\{ [\mathbf{h}_{PI}(t)] * \left\{ \begin{bmatrix} \cos(\omega_0 t) & -\sin(\omega_0 t) \\ \sin(\omega_0 t) & \cos(\omega_0 t) \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_a(t) \\ x_b(t) \end{bmatrix} \right\} \right\}, \tag{5.3}$$

where $x_a(t)$ and $x_b(t)$ are the phase inputs, $y_a(t)$ and $y_b(t)$ are the phase outputs, $[\mathbf{h}_{PI}(t)]$ is the time domain equivalent of (5.1) or (5.2) applied to both d-axis and q-axis signals, and ω_0 is the fundamental frequency (in rad/sec).

The computational overhead of a d-q PI controller is quite significant, as a synchronized sinusoidal reference must be generated, as well as the overheads of multiple matrix conversions and the PI controller itself. If multiple controllers are required (such as for selective harmonic compensation discussed in sub-section 5.1.2) then using this type of controller becomes such a significant burden on the digital processor that its use is impractical for many DSP systems.

The PI d-q controller is now commonplace in three-phase converter control systems, but is not directly applicable to single-phase systems. However, a similar controller designed for single-phase systems has recently been investigated by Zmood et al. [114], but it is also quite computationally expensive. Therefore, an alternative approach is needed.

5.1.1 P+Resonant Controllers

To overcome the limitations of the d-q PI controller, Zmood et al. [112] [113] [114] has proposed a new form of linear control theory for electrical ac systems. The theory is based on ac stationary frame equivalents of the dc d-q PI controller, using both exact and approximate transformations to derive an ac stationary frame controller that achieves zero steady-state error. Early work [116] showed that the exact transformation resulted in cross-coupling terms between the phases in the stationary frame PI d-q equivalent, while the approximate transformation used a low-pass

to band-pass transformation of

$$H_{P+R}(s) = H_{PI}\left(\frac{s^2+\omega_0^2}{2s}\right) \tag{5.4}$$

to derive a stationary frame controller which has no cross-coupling terms. Interestingly, if the cross coupling terms of the exact transformation are dismissed, then the result is identical to the approximate version when applied to the ideal integrator in (5.1), and is nearly identical when applied to the finite pole integrator in (5.2). This controller was named by its designers as the P+Resonant controller, as the result is simply the addition of the proportional component (i.e. P), plus a second order resonant filter at the fundamental frequency. For the ideal PI dc controller (5.1) the P+Resonant controller has the transfer characteristic of

$$H_{P+R}(s) = K_P + 2K_I\frac{s}{s^2+\omega_0^2}. \tag{5.5}$$

Unfortunately, this ideal implementation is impractical for a digital filter implementation and will therefore not be considered any further in this work. However, using the Laplace transformations (and dismissing the cross-coupling terms) the finite pole PI controller in (5.2) can be transformed from the *d-q* frame into the stationary ac frame to generate the P+Resonant controller:

$$H_{P+R}(s) = K_P + 2K_I\frac{s+\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}. \tag{5.6}$$

Using the approximate transformation of (5.4) gives the P+Resonant form of:

$$H_{P+R}(s) = K_P + 2K_I\frac{s}{s^2+2\omega_c s+\omega_0^2}. \tag{5.7}$$

Since ω_c is typically much smaller than ω_0, the response of (5.5) is nearly identical to that of (5.7).

For this research only (5.6) will be used, as both versions require second order filter implementations, and there is no practical advantage in using the approximate form of (5.7). Note also that the proportional gain K_P does not vary between the rotating and stationary frame controllers, as it contains no frequency dependant parameter and is therefore unaffected by the frequency transformations.

Probably the most important aspect of the P+Resonant controller is that without any cross-coupling terms it can be applied to each phase separately. Therefore, it can be applied directly to single-phase systems, as well as to the zero sequence component in four-wire systems. Shortly

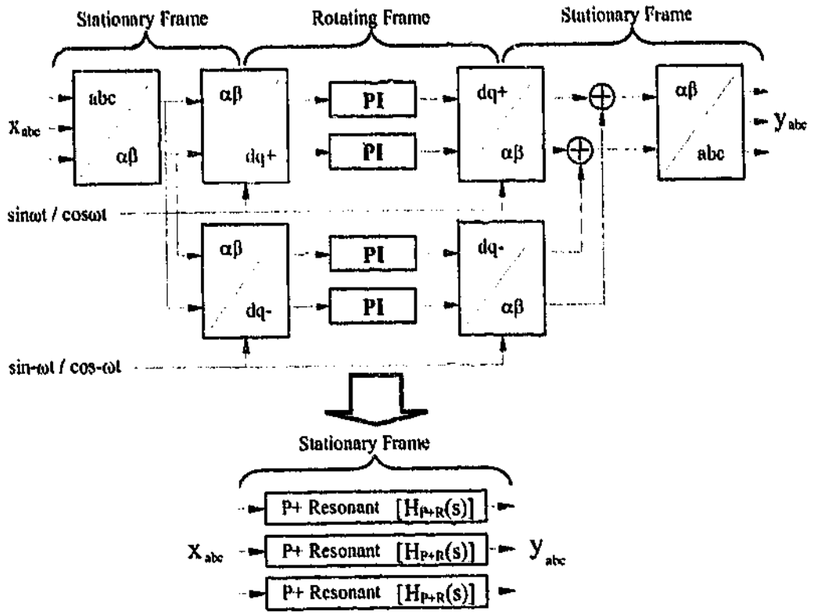


Figure 5.1: Conversion of the rotating frame *d-q* PI controller to the stationary frame P+Resonant controllers.

after the development of the P+Resonant controller, Mattavelli et al. [110] identified that it was actually the equivalent of combined positive and negative sequence *d-q* PI controllers. When the exact transformation forms of the positive and negative controllers are added together, the cross-coupling terms (previously dismissed) cancel each other out. This is illustrated in Figure 5.1. As a result, the computational overhead of the P+Resonant solution (which is already reduced compared to a synchronous frame system), is twice as fast again when both positive and negative sequence control is required.

The work by Zmood et al. [114] used the finite pole integrator form of (5.2), where the dc gain (ac fundamental gain for the equivalent P+Resonant controller) varies with both K_I and the cut-off frequency ω_c. This approach ensures that the transient response is independent of ω_c, and is dependant on K_I. For this research the finite pole integrator

$$H_I(s) = K_I\frac{\omega_c}{s+\omega_c} \tag{5.8}$$

is used instead, which has a dc gain of K_I, irrespective of the value of ω_c. (It is interesting to note that (5.8) is simply a low-pass filter with a pass-frequency gain of K_I.) Once K_I is set, the

transient response is then altered by the choice of ω_c. The resulting P+Resonant controller is given by

$$H_{P+R}(s) = K_P + 2K_I\omega_c\frac{s+\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}. \tag{5.9}$$

Figures 5.2 and 5.3 illustrate the difference between Equations (5.6) and (5.9) for a varying value of f_C (i.e. $\omega_C = 2\pi f_C$), and shows situations both without (Figure 5.2a and 5.3a) and with (Figure 5.2b and 5.3b) the proportional term. Whilst the formula difference between these two approaches is just an overall gain change of ω_c, the design process of the controller is significantly altered. This is discussed in detail in Section 5.2.

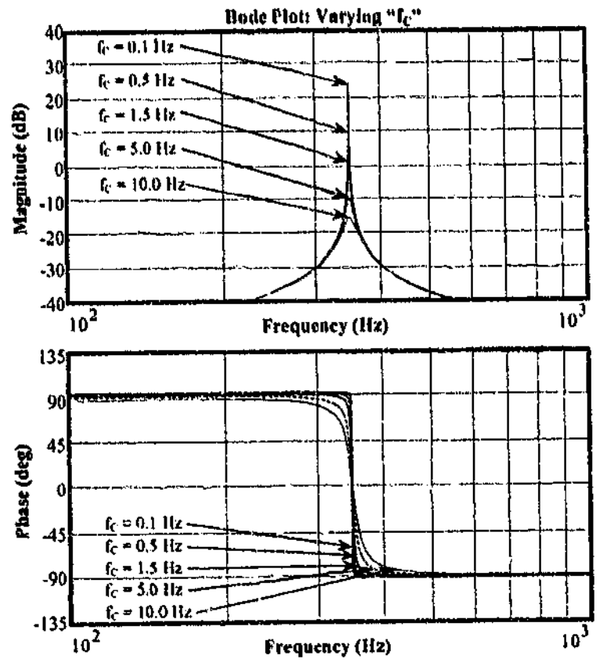
As a final note, after the development of the P+Resonant controller by Zmood et al. [114] similar stationary frame resonant based controllers were proposed by other researchers. The controller presented by Sato et al. [117] [118] [119] has very similar characteristics around the selected frequency, but also contains a low frequency gain component (due to the lack of the single pole used by the P+Resonant controller). This controller is therefore less practical for use with selective harmonic compensation, and is not investigated any further in this work.

5.1.2 P+Resonant Selective Harmonic Compensation

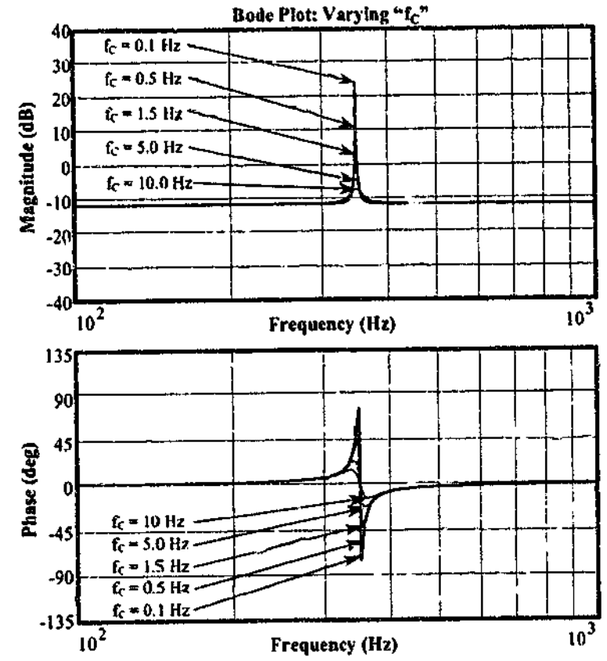
So far the P+Resonant controller has been discussed only for regulation of the fundamental frequency. However, the computational advantage of the P+Resonant controller (compared to the d-q PI) is predominant when multiple controllers are required. This is the requirement for selective harmonic compensation.

In recent years selective approaches for the attenuation of harmonics have been reported in a variety of Custom Power applications. Examples include the shunt hybrid filter [120] [121], series hybrid filter [52], shunt active filter [122] [123], as well as the UPS [124] [125] [126] and DVR [67] [127]. These selective methods have included both feed-back (e.g. d-q PI [52] [124]) and feed-forward (e.g. DFT [128], d-q with low-pass filter [67] [122], p-q theory [123], neural networks [129], Kalman filters, etc.) selective approaches, or a combination of both [120] [121]. However, most of the schemes which have presented experimental verification have operated on only one or two harmonics. In contrast, Mattavelli et al. [17] [110] has proposed the use of the P+Resonant controller for selective multiple harmonic feed-back compensation, as a computationally efficient alternative to a feed-back based d-q PI controller, and has experimentally implemented both a UPS [17] and shunt active filter [110] to prove the practical viability of the approach.

Chapters 3 and 4 identified that a combined feed-forward and feed-back controller could be

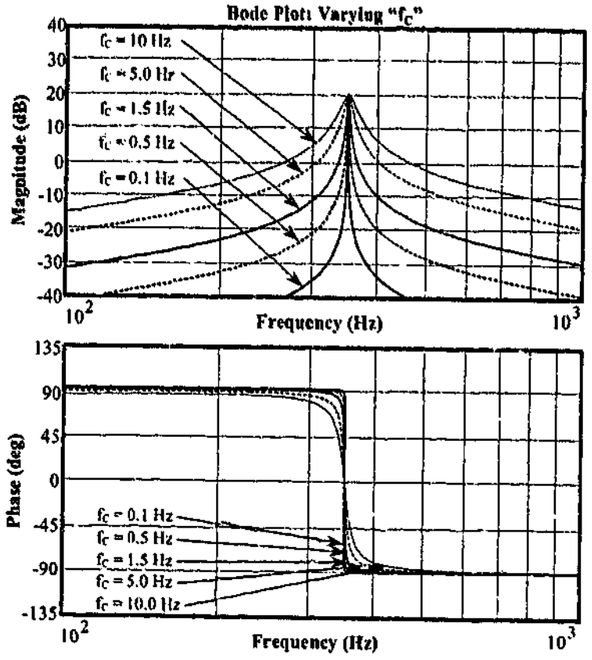


(a) P+Resonant Equation 5.6, K_P=0

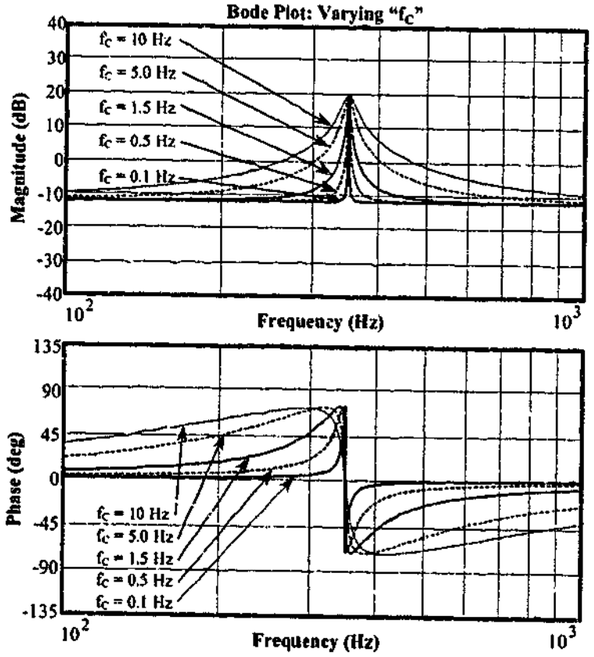


(b) P+Resonant Equation 5.6, K_P=0.25

Figure 5.2: Simulated open loop frequency response of the P+Resonant controller in Equation (5.6) with varying cut-off frequency.



(a) P+Resonant Equation 5.9, K_P=0



(b) P+Resonant Equation 5.9, K_P=0.25

Figure 5.3: Simulated open loop frequency response of the P+Resonant in Equation (5.9) controller with varying cut-off frequency.

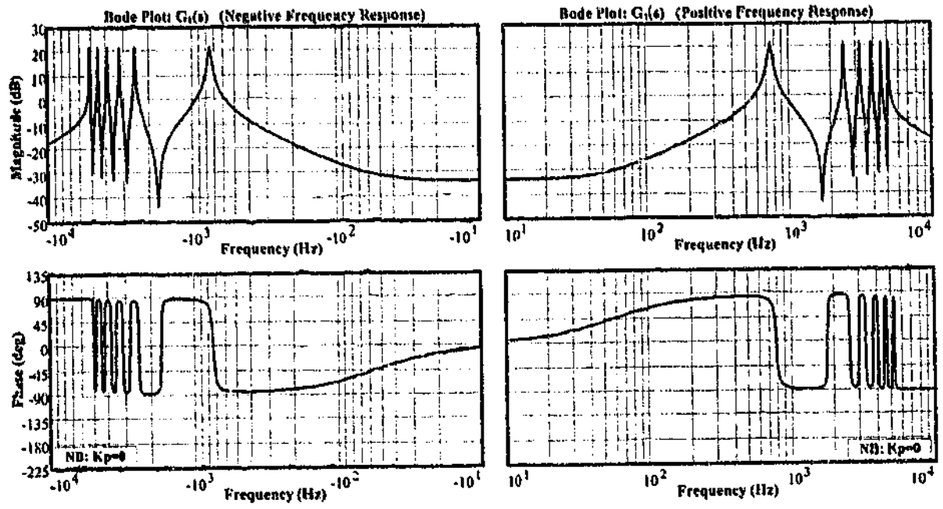


Figure 5.4: Frequency response of the harmonic P+Resonant voltage controllers.

used to control the series converter of the UCPC. The purpose of the feed-back controller was to provide accurate steady-state performance (primarily for compensation of the lower magnitude components such as the voltage harmonics and unbalance), and the use of selective feed-back compensation directly on the load voltage was investigated for its suitability for this task. The utilization of d-q PI controllers was assumed, as the P+Resonant equivalent had not yet been introduced. However, due to its computational advantages, as well as its applicability to single-phase systems, the P+Resonant controller will be used in this research from now on for the series converter selective harmonic voltage compensation. Using (5.9) the fundamental frequency ω_0 is replaced by each selected harmonic frequency ω_n to give the selective controller

$$H_n(s) = K_{I,n}\omega_c \frac{s+\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_n^2}. \tag{5.10}$$

The overall series feed-back controller is therefore given by

$$G_1(s) = K_P + \sum_{n\epsilon M} 2K_{I,n}\omega_c \frac{s+\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_n^2} \text{ where } M = \{1,5,7,9,11,13\}. \tag{5.11}$$

The frequency response of this controller is illustrated in Figure 5.4 (with $K_P = 0$), and highlights the symmetry of the controller which allows easy compensation of both positive and negative sequence disturbances.

Phase margin compensation

To improve the phase margin at each selected frequency, Mattavelli et al. [110] proposed the addition of a phase offset ϕ_n to each controller , such that

$$H_n(s) = K_{I,n}\omega_c \frac{cos\phi_n s + (\omega_c cos\phi_n - \omega_n sin\phi_n)}{s^2 + 2\omega_c s + \omega_c^2 + \omega_n^2} \qquad \text{where} \qquad \phi_n = \omega_n \tau_{Delay}. \tag{5.12}$$

The value of the compensating phase shift is coordinated to compensate for the phase shift caused by the time delay (τ_{Delay}) between the sampled input signal and the converter output reacting to the demanded signal. For the experimental system in this research a similar approach has been used, where the delay is two sample periods (i.e. τ_{Delay}=200 μs). The size and advantage of the phase compensation becomes more pronounced as the order of the harmonic increases. For example, the phase shift caused by τ_{Delay} for a 5^{th} harmonic is 18°, but it increases to 47° for the 13^{th} harmonic.

Practical Implementation

The computational cost of a single resonant controller (implemented as a 2^{nd} order IIR digital filter) was found to be approximately 2 μs on the DSP used for the LV experimental work described in Chapter 9 (i.e. 4 μs for a combined a/b phase set). Each selected harmonic therefore has a computational cost of 4 μs (24 μs for six harmonics), instead of the 27 μs (162 μs for six harmonics) required for positive and negative sequence d-q PI controllers. This computational saving is significant considering that, with a sampling frequency of 10 kHz, the absolute maximum interrupt time available is less than 100 μs. The stationary frame approach is therefore the only viable option for implementation of multiple harmonic compensators on the experimental system available.

One disadvantage of using the P+Resonant controller for selective compensation is that its digital implementation is more sensitive than its rotating frame equivalent to coefficient rounding errors, and a floating point processor is often required for satisfactory performance unless at least 32-bit word lengths are used [17]. This problem also becomes more severe as higher frequency harmonics are selected. Most of the DSP processors currently designed for PWM of power converter applications are based on 16-bit fixed-point platforms (the TI TMS320F240 DSP used in Chapter 9 is just one common example). The problems of using conventional digital filter techniques must be therefore overcome for the use of the P+Resonant to be viable. This problem is considered further in Chapter 6, where a suitable solution is identified and verified.

5.2 Parameter Design of the Stationary Frame Series Controller

Chapter 4 investigated the effect that the physical parameters (e.g. LC filter and power system parameters) have on the system. A default set of values (Table B.1 in Appendix B) was used as a starting point to investigate each parameter in turn. This section continues this approach and extends the analysis to the control parameters of the proposed series controller starting from a similar set of default parameters. The aim is to obtain an understanding of the effect of each control parameter, such that a design process for the series controller can be developed.

The series control system parameters (Figure 5.5) can be broken up into five primary terms: the proportional gain (K_P), the integral gain (K_I), the resonant cut-off frequency (f_C), the active damping gain (K_{AD}), and the feed-forward component. The contribution of the feed-forward component will be investigated first, as this is a fixed quantity. The effect of changing the feed-back controller values of K_P, K_I, and f_C (from the feed-back controller $G_1(s)$ in (5.11)) will then be considered, followed by an investigation into active damping of the LC filter resonance.

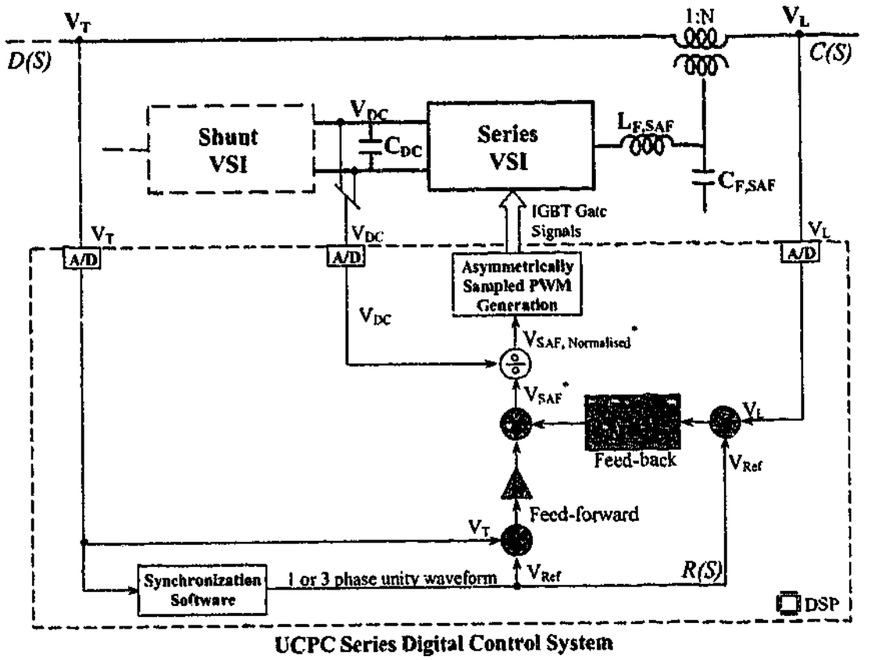


Figure 5.5: Components of the series injection control system. (Stationary frame feed-forward and feed-back components highlighted in grey.)

5.2.1 Effect of the Feed-forward Component

The contribution of the feed-forward component is fixed (i.e. to 100% of the measured difference between the supply voltage and the reference load voltage), and its primary role is to provide the transient performance of the controller. Using the control model developed in Chapter 3, Figure 5.6 illustrates the contribution of the feed-forward component in isolation (i.e. the light grey line). It can be seen that feed-forward achieves unity open loop gain in the lower frequency (taking the transformer ratio into account), but as the frequency nears the break point of the LC filter, the resonance response takes over. Furthermore, it can be seen in Figure 5.7 that only the system which includes feed-forward has a significant gain at the LC filter break point (i.e. the black line). Since both the feed-forward and proportional components provide a constant gain throughout the frequency spectrum (ignoring any practical considerations for now) it is only the smaller magnitude of the proportional gain in this example which stops it from having the same effect. However, just observing the increase in the open loop system gain at the break point is not a sufficient design test, since examining the open loop gain cannot determine whether the load voltage will see increased errors at the break frequency due to disturbances in the supply.

The disturbance plot (Figure 5.8) shows that the addition of the feed-forward increases the disturbance gain from 4-dB up to 12-dB. (Note that whilst Figure 5.6 shows that the proportional K_P component increases the system gain at the break point, the disturbance error is actually decreased with the introduction of K_P. This is due to the proportional feed-back providing a damping effect to the resonance, and is discussed further in sub-section 5.2.3.)

Figure 5.8 also shows that for lower frequencies the addition of feed-forward gives a significantly lowered disturbance error. However, care must be taken to consider the practical effects of the system on this linear theoretical result. The theoretical model assumes that the instantaneous supply voltage is subtracted from the reference, and is then added to the demanded voltage, which is also instantaneously created by the converter. In practice there is one sample delay before the reference signal is passed to the PWM modulator, and another sample period until the PWM modulator has created the demanded volts-seconds output. This delay will create an error which is not accounted for in the theoretical linear model. However, the voltage drop error caused by the LC filter is accounted for in the model, and explains why there is a gradual drop in disturbance error as the frequency decreases (i.e. as the frequency decreases the impedance of the filter inductance also decreases, creating a smaller voltage drop error). Whilst the error caused by the delay will also drop off for lower frequencies (as the effect of the fixed sample delay will be smaller), error calculations similar to those presented in Chapter 4

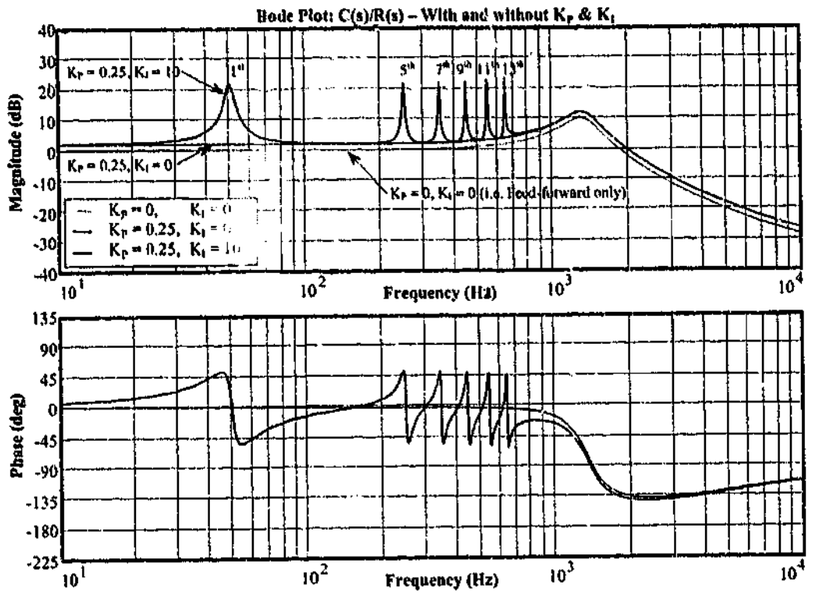


Figure 5.6: Bode plots showing effect with and without the integral gain (K_I), and the proportional gain (K_P).

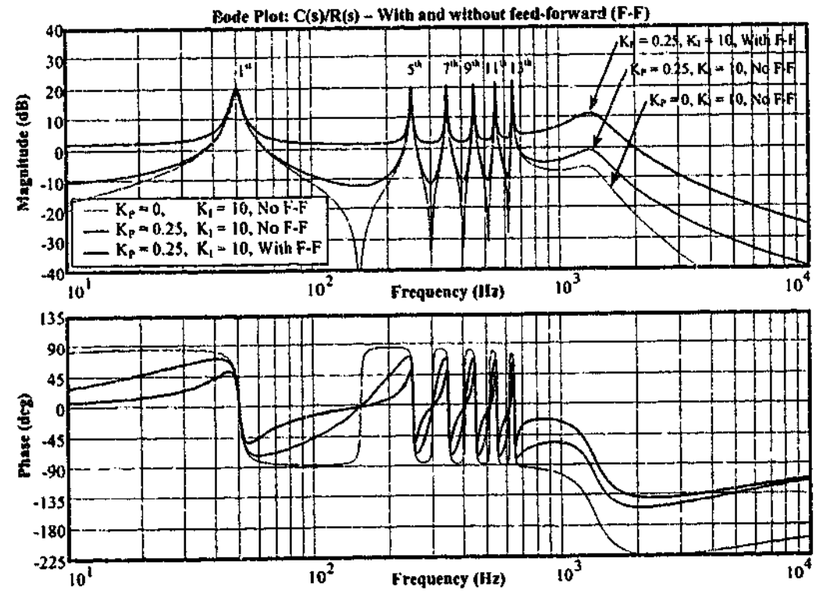


Figure 5.7: Bode plots showing effect with and without the feed-forward (F-F) controller.

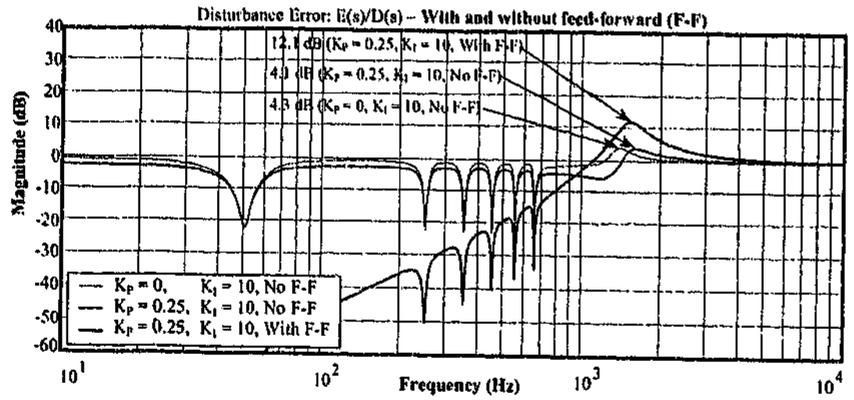


Figure 5.8: Disturbance error $E(s)/D(s)$ showing effect of with and without the feed-forward (F-F) controller. (Refer to Table B.1 in Appendix B for remaining default parameters.)

for the digital current regulation (as well as other errors in the system) mean that expecting the disturbance errors shown in Figure 5.8 to be below -60-dB is unrealistic for a practical system. One major reason for the introduction of the feed-back system is to counteract these practical errors in the feed-forward system.

5.2.2 Design of K_I

As discussed at the start of this chapter, the purpose of the resonant controller is to provide high feed-back gain at selected frequencies, which should in turn result in low disturbance errors at these values. Figure 5.7 shows that the contribution of the resonant controllers (light grey line) at frequencies away from the selected harmonics is small. We would therefore expect a change in K_I to have only a small effect on the LC resonance of the system – as shown by the very small magnitude change in system gain at the break point in Figure 5.9. However, the resonant controllers alter the phase response of the system at this frequency (Figure 5.9), and therefore also alter the disturbance error. For a change in K_I from 2 to 20 (where K_I is applied to all controllers equally), the disturbance error at the break point increases from 9.7-dB to 15.9-dB, respectively (Figure 5.10).

For series based system without a feed-forward component, the model shows that the resonant gain and the disturbance error are closely related. In fact, inspection of the system gain and disturbance error of the system without the feed-forward component (medium grey line in Figures 5.7 and 5.8, respectively) shows that the disturbance error at a selected frequency is

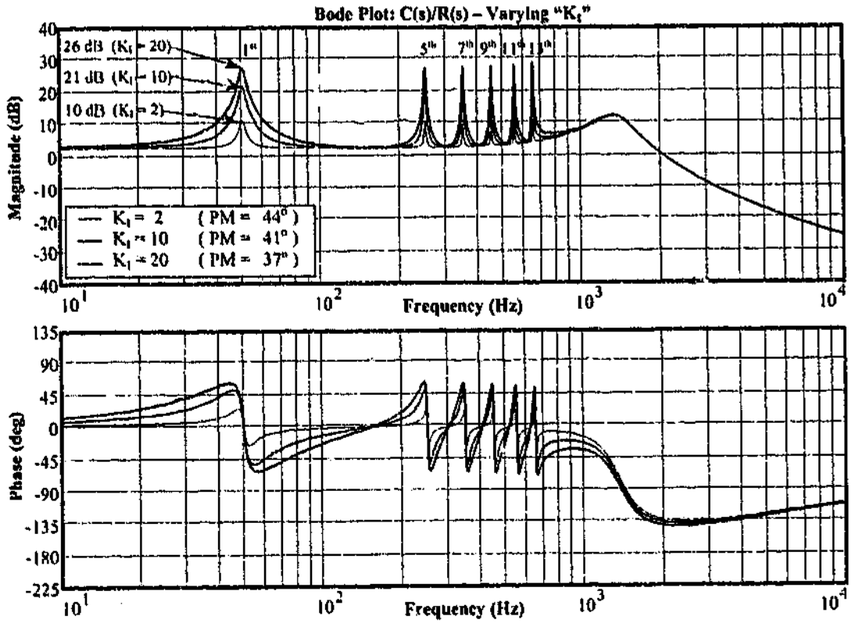


Figure 5.9: Bode plots showing effect of varying the integral gain (K_I). (Refer to Table B.1 in Appendix B for remaining default parameters.)

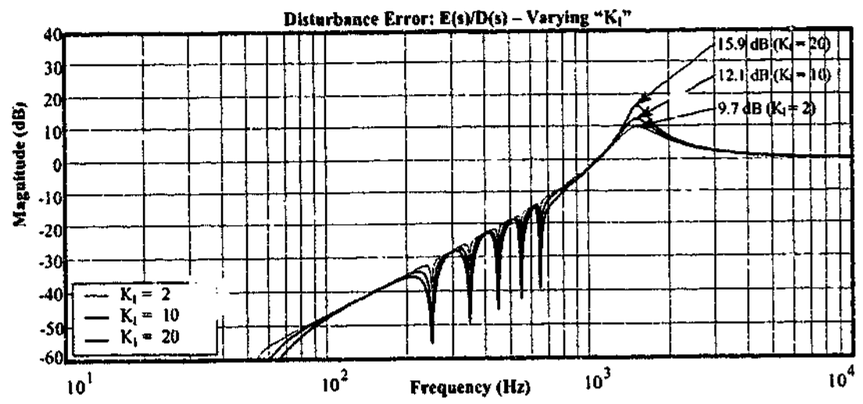


Figure 5.10: Disturbance error $E(s)/D(s)$ showing effect of varying the integral gain (K_I). (Refer to Table B.1 in Appendix B for remaining default parameters.)

approximately the inverse of the resonant gain (i.e. an inverse becomes a negation in the decibel scale). This is also confirmed in Figures 5.9 and 5.10 by comparing the response for varying values of K_I. It is expected that errors introduced by the practical system will slightly increase the theoretical disturbance error, but the disturbance error will also be slightly lowered by introduction of feed-forward. The value of K_I in the design stage can therefore initially be chosen based on the inverse of the desired disturbance error. (Note that this assumption is confirmed experimentally on the medium voltage experimental system as is presented in Chapter 10.) For example, assume that the maximum expected individual voltage harmonic in the supply is 5%, and all individual harmonic voltages in the load are to be reduced to less than 0.5%. For this case a ten times reduction of each selected supply voltage harmonic is desired (i.e. a disturbance error of -20-dB), so a minimum K_I of 10 should be chosen as a starting point in the design.

So far in this analysis K_I is assumed to be the same for all of the separate resonant controllers. This need not be the case. If different disturbance rejection values are required for particular selected frequencies, then K_I is simply replaced by $K_{I,n}$ for each resonant controller, where n is the n^{th} harmonic. Note that the higher frequency resonant controllers have more influence on the stability of the system. For example if the 11^{th} and 13^{th} resonant controller gains are reduced from 10 to 5 (with the others remaining at 10), then a phase margin increase of more than 1^o is achieved. However, if the 5^{th} and 7^{th} controller gains are reduced in the same way, then only a 0.1^o improvement is seen in the phase margin.

5.2.3 Design of K_P

The proportional feed-back gain, K_P, is used to stabilize the feed-back controller and to improve the transient response of the feed-back. The effect of varying the K_P in the system is illustrated in bode (Figure 5.11) and disturbance (Figure 5.12) plots. A variation in K_P is shown to affect the magnitude over the entire frequency spectrum shown, but has virtually no effect on the system phase response. The phase margin of the system is therefore altered by virtue of the shift in the frequency where the system gain reaches zero (i.e. the magnitude cross-over frequency), and not by a shift in the phase itself.

Proportional gain provides some damping for the LC filter resonance. The disturbance plot (Figure 5.12) shows a decrease from 14.7-dB disturbance error with no proportional gain, to 10.5-dB with a proportional gain of 0.5. A large proportional gain therefore improves both the damping of the system, and the transient response. However, the increase in proportional gain also decreases the phase margin, and a compromise must be found in the design process.

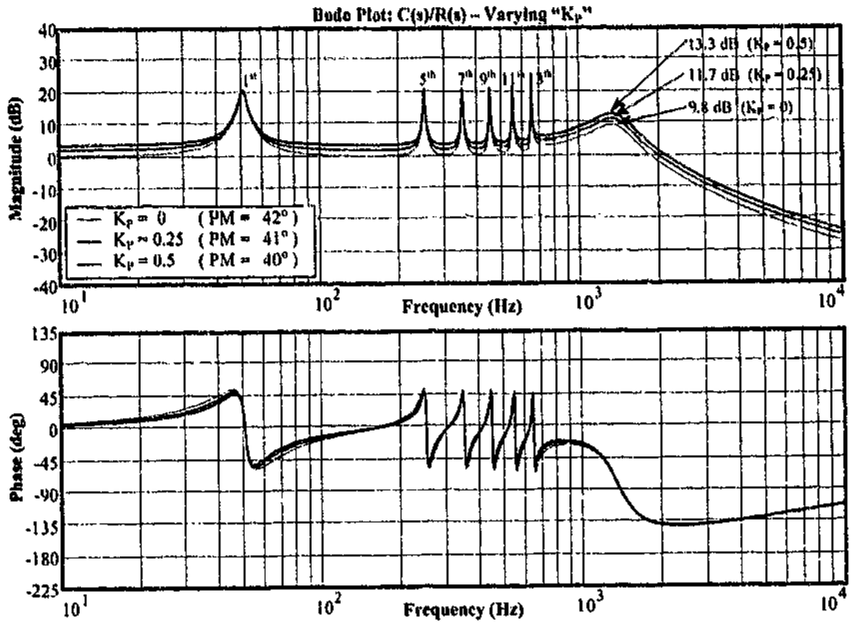


Figure 5.11: Bode plots showing effect of varying the proportional gain (K_P). (Refer to Table B.1 in Appendix B for remaining default parameters.)

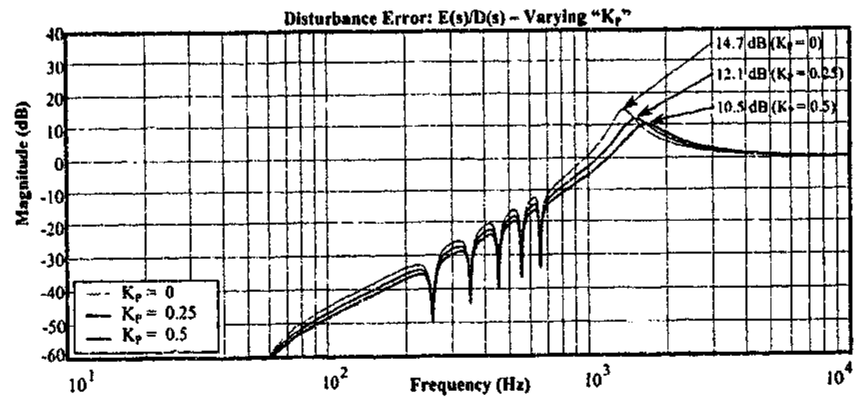


Figure 5.12: Disturbance error $E(s)/D(s)$ showing effect of varying the proportional gain (K_P). (Refer to Table B.1 in Appendix B for remaining default parameters.)

5.2.4 Design of f_C

Varying the resonant cut-off frequency f_C (i.e. $\omega_C = 2\pi f_C$) has already been shown in Figure 5.3 to effectively alter the width of the resonant term, but using (5.9) the peak gain is preserved. This is again confirmed in Figure 5.13, which shows the system response for three different values of f_C. The result is that the steady-state compensation performance at each selected harmonic will be independent of f_C, but the transient performance and stability will vary. The independence of the steady-state performance with f_C is illustrated by the disturbance plot in Figure 5.14, which shows an equivalent disturbance error for values of f_C at each target harmonic. As f_C is increased, the transient performance improves, and vice versa for a decrease in f_C. The transient performance variation can be compared to the effect of changing the cut-off frequency of a finite pole integrator in the rotating reference frame. For a finite pole integrator (simply a low-pass filter with a band-pass gain of K_I), as f_C is increased, higher frequencies are passed through into the integral controller, and therefore during transient conditions, a step signal input is seen faster by the controller. Hence, it can respond faster to the transient condition. However, Figure 5.13 shows that the phase margin of the system reduces significantly as f_C is increased.

From a design perspective a smaller value of f_C should initially be chosen, as this is the more stable option. Once a stable overall design solution is found, this value can be increased iteratively whilst maintaining the desired stability margins. For fixed-point implementations, the smaller the value of f_C, the more difficult the implementation will become. At a limit, this implementation will become near impossible to implement. Therefore, the lower value initially chosen for the design will be dictated by this practical limitation, and this will be discussed further in Chapter 6.

5.2.5 Active Damping Feed-back

The majority of the controller design to date has been devoted to damping down the resonance from the LC filter to minimize the amplification of disturbances in the supply voltage (at this frequency), and also to maintain the stability of the system. Chapter 4 has shown that the addition of physical resistors in the LC resonance path provides damping. However, the amount of damping is limited by the power losses in these resistors, as well as by the loss of filter performance when a resistor is placed in series with the series capacitor.

Ideally, the resonance should be damped without any increased power loss or loss in filtering performance. Active damping is therefore a good alternative, as this control method creates a

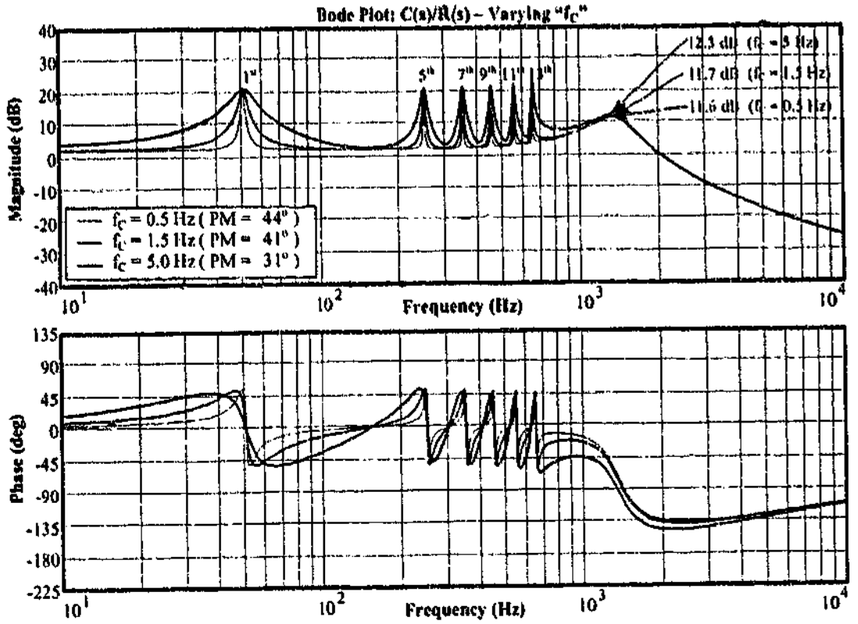


Figure 5.13: Bode plots showing effect of varying the cut-off frequency (f_C). (Refer to Table B.1 in Appendix B for remaining default parameters.)

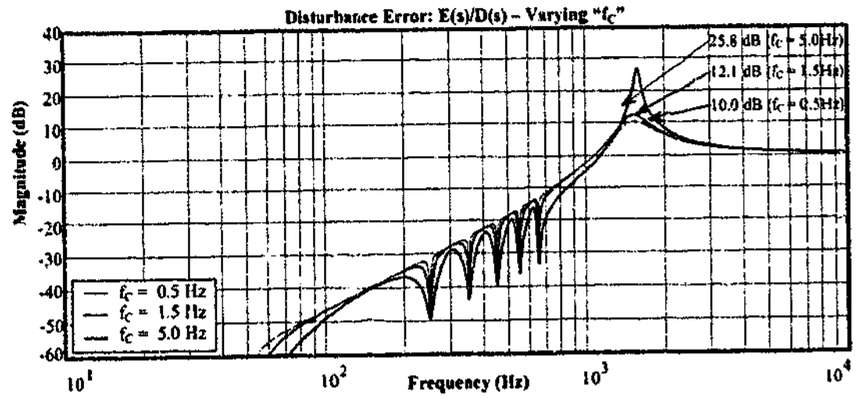
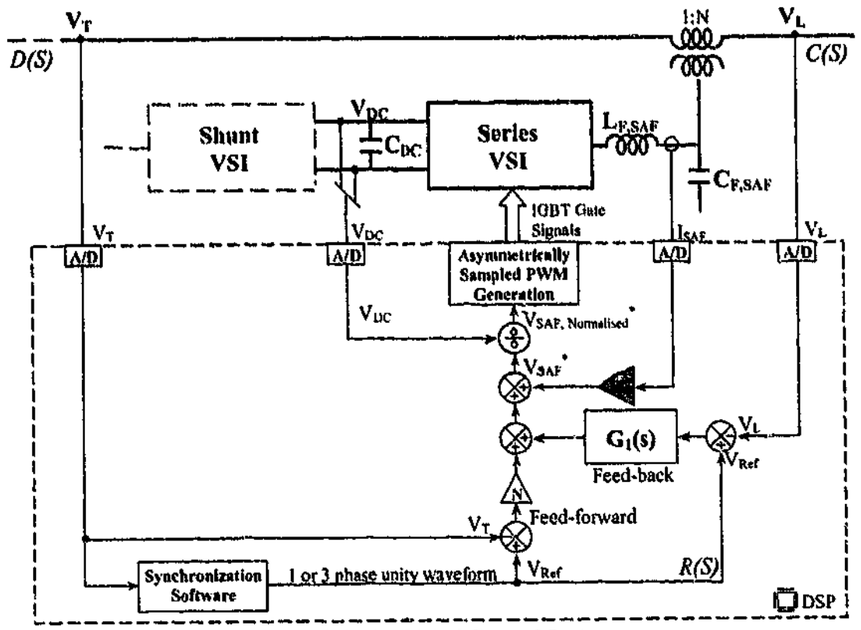


Figure 5.14: Disturbance error $E(s)/D(s)$ showing effect of varying the cut-off frequency (f_C). (Refer to Table B.1 in Appendix B for remaining default parameters.)

virtual resistor but does not have the power losses associated with a real resistor. The filtering performance for active damping schemes also remains relatively unchanged, as the frequencies of filtering interest are well outside the bandwidth of the control scheme. Active damping schemes are becoming more common for converter applications [109] [130] [131], and are especially valuable in higher power systems where using physical resistors is impractical. In the literature to date, fully active series applications (such as the dynamic voltage restorer) have either; used physical resistors for damping [69] [89] [106] [107], used deadbeat controllers which account for the LC filter components [127] [132] [133] [134], accepted the large voltage ring that occurs [60] [135] [136], or used other control schemes that inherently provide some damping [63].

Dahono et al. [131] presented four possible resistor placement options for damping a VSI converter (with an LC filter): **(a)** in series with the filter inductor; **(b)** in series with the filter capacitor; **(c)** in parallel with the filter inductor; and **(d)** in parallel with the filter capacitor. For voltage regulation each of these four options can be arranged such that an equivalent virtual resistor can be created using feed-back of various measurements in the system. Active creation of the options (a) and {(c),(d)} require measurement of the capacitor current and capacitor voltage, respectively [131]. As neither of these variables are measured by the proposed system, these options are discarded so that the cost of additional hardware is not occurred. Thus only the placement of a virtual resistor in series with the filter inductance can be used without requiring additional hardware measurements.

Placing a virtual resistor in series with the filter inductor is achieved by feeding the converter output current measurement (which is required for converter over-current protection) back into the controller as an inner current loop with gain K_{AD} (Figure 5.15). The addition of the active damping signal is placed before the dc-bus compensation scheme so that the net gain is independent of the dc-bus voltage. The value of K_{AD} becomes the equivalent virtual resistance placed in series with the filter inductor. Note that Chapter 4 discarded the use of an additional resistance in series with the filter inductance due to the voltage drop problems. However, for a virtual resistor placement the feed-back controller $G_1(s)$ will overcome these limitations, unlike the physical placement alternative. But, the introduction of the virtual resistor does lead to more errors in the feed-forward portion of the controller. This has little consequence on the steady-state performance, but will create a small temporary error (particularly at full load conditions) during fast and deep transient events. If capacitor current measurement was available, then this would be the preferable option from an overall performance perspective, but it is not used here due to cost considerations.



UCPC Series Digital Control System

Figure 5.15: Components of the series injection control system. (active damping component highlighted in grey.)

The effectiveness of the active damping scheme is presented using the bode and disturbance plots of Figures 5.16 and 5.17. The addition of the scheme provides an extra 22° of phase margin (PM), and reduces the disturbance gain from 12.1-dB to 5.8-dB (with K_{AD} set to 3 Ω). Even with the capacitor resistance R_{CF} reduced to only 1 Ω (from 4.7 Ω), the damping is still significantly better than for the original system, but this reduction in capacitor series resistance does reduce the phase margin. For this reason, a combination of both physical series capacitor resistance, and virtual resistance in series with the filter inductance is preferable (within the allowable power loss and filtering limits of the physical resistor placement).

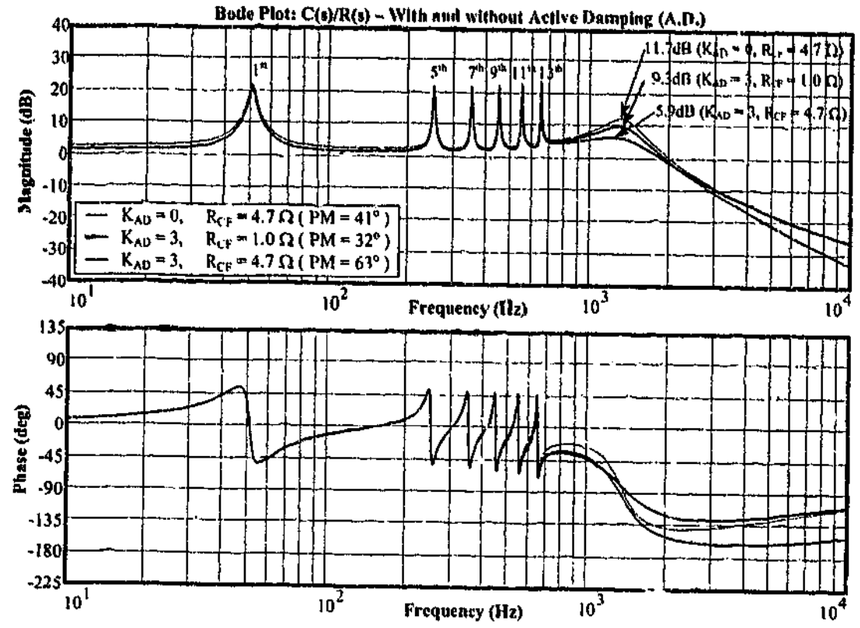


Figure 5.16: Bode plots showing effect of the active damping (A.D.) scheme. (Refer to Table B.1 in Appendix B for remaining default parameters.)

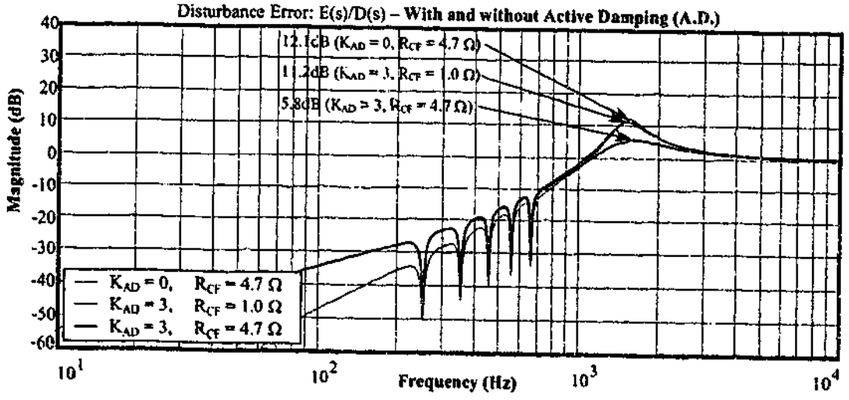


Figure 5.17: Disturbance error $E(s)/D(s)$ showing effect of the active damping (A.D.) scheme. (Refer to Table B.1 in Appendix B for remaining default parameters.)

5.2.6 Series Controller Design Process

Now that the effects and contributions of each of the separate controller parameters have been investigated (as well as the effects of the physical system parameters in Chapter 4), an overall design process for the design of the series controller can be developed. Using (5.10), the gain at selected resonant frequencies has been shown to be fixed by $K_{I,n}$, and is not affected by variations in f_C. This approach is used to define the required disturbance rejection, but the transient performance and stability of the system is controlled by more than just the resonant controller parameters (i.e. K_P, as well as filter damping elements such as R_{CF}). By fixing the target outcome first, the remaining parameters can be varied to find a suitable solution, whilst guaranteeing that the steady-state requirements will be met.

For *harmonic* voltage compensation, the steady-state performance in this research system has a higher priority than the transient performance. This is for two reasons. Firstly, harmonics are by definition steady-state, not transient, events. Secondly, the effects of voltage harmonics noted in Chapter 2 are mostly due to long term conditions, not caused by transient increases in harmonics for a few cycles. Therefore, the values of $K_{I,n}$ should only be decreased from the original target if a solution cannot be found within the scope of the remaining variables. The required disturbance error can be estimated by the ratio of the maximum voltage harmonic expected to exist at the installation and the limit down to which this harmonic must be reduced. This ratio is likely to vary for different harmonics, and may be varied for each harmonic controller if desired. The resonant gain $K_{I,n}$ is then found by the inverse of the disturbance ratio.

The design of the fundamental resonant feed-back gain $K_{I,1}$ is undertaken using the same method used for the voltage harmonic controllers, but using the voltage unbalance limits. Note that voltage unbalance limits are used as they are more challenging to achieve than voltage magnitude variation limits, and equipment is more susceptible to small variations in unbalance (Chapter 2 and Appendix A).

The design process is then as follows:

1. Create the series topology linear control model for the series system including load and source connections as discussed in Chapter 4.

2. Determine a transformer ratio N based on the required sag compensation characteristics and the desired operating voltages (Chapters 3 and 4).

3. Determine the filter parameters L_F and C_F as per the limitations given in Chapter 4.

4. Determine the disturbance rejection required and choose the inverse of this as the integral gain $K_{I,n}$ (refer above). Note that it has already been established that the higher frequency harmonic controllers have a much larger effect on the overall system stability. Therefore the choice of a smaller value of $K_{I,n}$ for the higher order harmonics (which are generally smaller in magnitude anyway) is recommended.

5. Determine the smallest value of f_C that can reasonably be digitally implemented on the chosen digital system (refer Chapter 6).

6. Determine a value of R_{CF} such that a significant damping effect is seen, yet the increased losses are within the system design parameters. (Note that the disturbance plot of $E(s)/D(s)$ should be used – not the open loop gain of $C(s)/R(s)$.)

7. Increase K_P such that the phase margin is not significantly disturbed, while still achieving further damping is still applied to the LC filter resonance.

8. Add in the active damping gain K_{AD}, and increase it until the resonance is critically damped. Also ensure that the equivalent series resistance does not become a dominant factor with regards to converter output voltage drop (i.e. the full load current multiplied by the virtual resistance K_{AD}). If a large enough value of K_{AD} is possible, investigate reducing the value of R_{CF} (i.e. to lower the resistive power losses).

9. Increase f_C whilst maintaining an acceptable phase margin and rejection of disturbance at the LC filter break point. (Note that K_I could alternatively be increased here to provide similar transient advantages, but increasing f_C will reduce the severity of the fixed-point digital implementation problems; see Chapter 6.)

10. Test using a switched simulation of the entire system (Chapter 8). If unstable (or significantly under-damped) reduce the value of f_C (or increase the value of R_{CF} if the break point resonance is the dominant stability problem and if it is acceptable to incur the increased losses). If stable, and the transient response is not sufficient, then increase the value of f_C. If stable and the transient response is more than sufficient, reduce R_{CF} to minimize the losses.

11. The final stage is to experimentally verify the system under the desired source and load conditions.

5.2.7 Final System Parameter Values

Using the design process above, the final parameters used for the LV experimental work are presented in Table 5.1. The 2 mH filter inductance (L_F) was chosen to be 9% based on the 30 kVA LV experimental VSI with a 700 V dc-bus. As discussed in Chapter 4, the values of filter capacitance and series damping resistance were chosen to be 15 μF and 4.7 Ω, respectively, to maximize the stability of the system and minimize the LC resonance. The proportional gain (K_P) was increased to 0.25, as this value was shown in the control model to provide additional damping to the system, while it was not large enough to have a noticeable effect on the stability of the system. The active damping gain/resistance (K_{AD}) was chosen to be 3 Ω. The control model shows that this value provided significant damping of the system, while the full system model (Chapter 8) verified that the value has little affect on the transient error. The filter cut-off frequency (f_C) was limited by the smallest feasible value available for the higher selected harmonics when implemented on a 16-bit fixed-point DSP (refer to Chapter 6). The integral gain $K_{I,n}$ is broken up into two groups. For the typically larger fundamental, 5^{th}, and 7^{th} harmonic voltages a disturbance gain of 20-dB was chosen (i.e. $K_{I,n} = 10$) such that expected maximum steady-state errors (after feed-forward compensation) of up to 5%, could be reduced to less than 0.5%. The higher frequency 9^{th}, 11^{th} and 13^{th} harmonics are generally much smaller and also have a more dominant effect on the stability of the overall system. Therefore, a smaller attenuation of 14-dB (i.e. $K_{I,n} = 5$) was chosen.

All of the bode and disturbance plots shown so far have clearly contained resonance problems around the LC frequency. The final system applies the results of the investigation into each parameter to maximize the damping of this resonance, and to also provide a high phase margin. Figure 5.18 presents the bode plot for the final system parameters. The resonant gain hump is now non-existent, and the phase margin is up to 85° for the series resistive-inductive load, and 100° for the purely resistive load. The disturbance plot (Figure 5.19) shows only 3.0-dB and 1.6-dB for the same loads, respectively. Whilst the most dominant factor for the improved damping and stability is the addition of the active damping scheme, the results are the culmination of tuning all the available parameters. This is illustrated by the fact that the active damping scheme improves the default system PM from 41° to 63°, but this is further increased to 85° when all parameter tuning is complete.

Note that whilst phase margins of 85° and above may seem ample, sample delays have not been included in the linear model used so far. With a magnitude cross-over frequency of 1.85 kHz, the delay from the 10 kHz sampling at this frequency corresponds to 67°. The phase

margin must therefore also take this delay into account. This is investigated further in Chapter 8 using a simulation which includes of all practical aspects of the UCPC.

Symbol	Name	Value
K_P	Proportional gain	0.25
$K_{I,\{1,5,7\}}$	Integral gain (lower harmonics)	10
$K_{I,\{9,11,13\}}$	Integral gain (higher harmonics)	5
K_{AD}	Active damping gain	3 Ω
f_C	Cut-off frequency	1.5 Hz
N	Transformer ratio	2
L_F	Filter inductance	2.0 mH
R_{LF}	Filter inductor resistance	0.05 Ω
C_F	Filter capacitance	15 μF
R_{CF}	Filter capacitor resistance	4.7 Ω

Table 5.1: Control and series converter parameters for the LV experimental work.

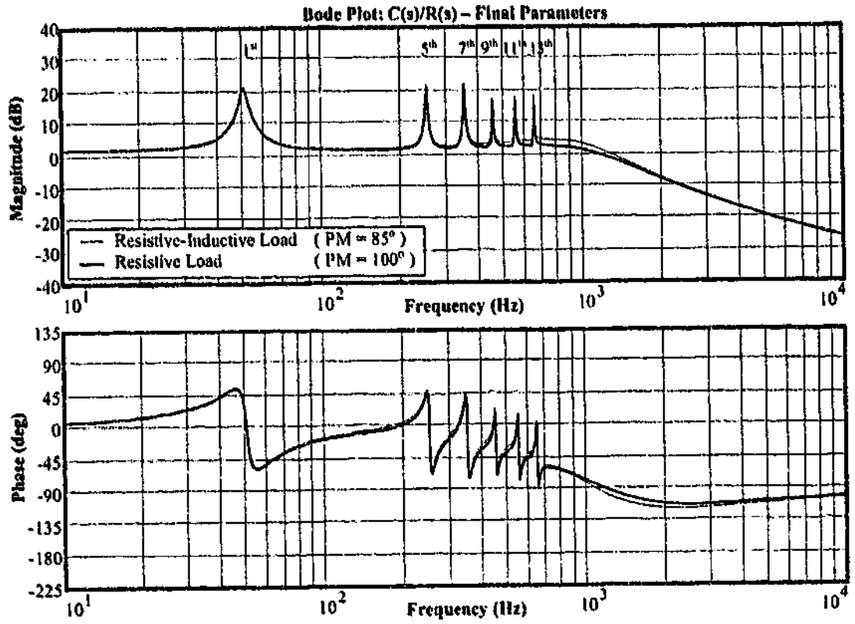


Figure 5.18: Bode plots for the final parameters used in the LV experimental work.

Figure 5.18: Bode plots for the final parameters used in the LV experimental work.

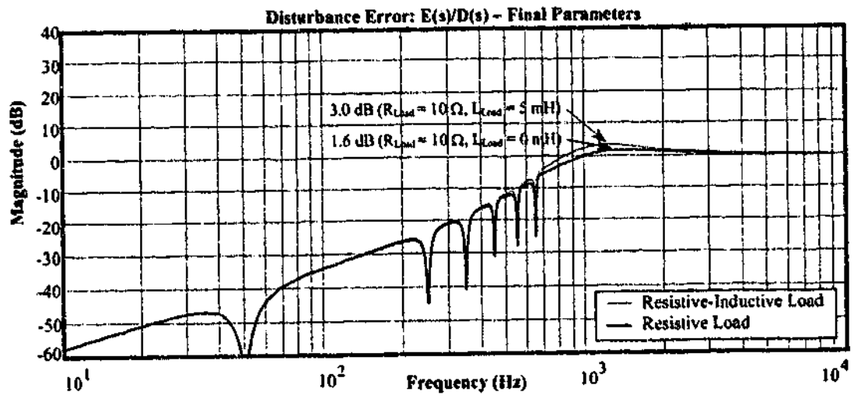


Figure 5.19: Disturbance error $E(s)/D(s)$ for the final parameters used in the LV experimental work.

5.3 Stationary Frame Shunt Current Reference Extraction

Chapter 4 presented the shunt converter control system for the UCPC, and noted that the conventional synchronous frame *d-q* harmonic extraction control block presented (highlighted in Figure 5.20) is not compatible with single-phase systems. The first section in this chapter has reviewed the development of the stationary frame P+Resonant controller which is derived from the synchronous frame *d-q* PI controller, and can be implemented on single-phase systems (unlike the synchronous frame *d-q* PI controller). This controller also has no requirement for a synchronized sinusoidal reference, and takes significantly less computational time.

This section investigates applying the conversion methodology used to develop the P+Resonant controller, to the high-pass filter based *d-q* frame harmonic signal extraction process. The aim is to achieve some, or all, of the practical and theoretical advantages of the P+Resonant controller, for the extraction of harmonic signals for use in the shunt portion of the UCPC in both single- and three-phase systems.

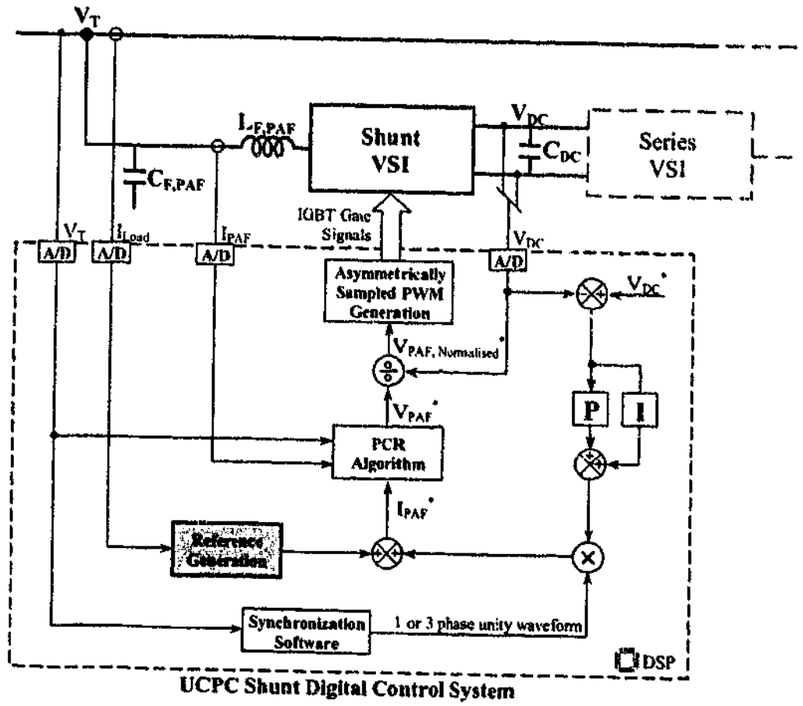


Figure 5.20: Components of the shunt control system.

5.3.1 Reference Signal Extraction Overview

The reference signal generation system is a central element of most active filter topologies. This system uses measured system variables as inputs to create current (or voltage) reference waveforms for the converter modulator that, if faithfully tracked, will cancel out all unwanted harmonic components in the target system. Figure 5.21 shows how the reference signal generation controller can be placed in a typical active filter system.

A large number of reported active filters implement their reference generation system in the Synchronous Reference Frame (SRF) [77] [94], which has the advantage of converting the harmonics of interest to a dc component so that they can be processed using a simple low-pass or high-pass filter. For example, several low-pass filters in cascaded reference frames can be used to extract particular harmonics for selective elimination [67] [122] [137]. However, this strategy does require substantial computational effort to be implemented, and is essentially the same as the selective PI *d-q* controller concept (discussed at the start of this chapter), where finite pole integrators are chosen and the gain is unity.

Alternatively, a high-pass filter in the fundamental synchronous frame can be used to retain all but the fundamental component, producing a target reference signal that is "everything else" to be compensated by the filter system modulator controller. This approach offers the benefit of minimal (if any) disturbance to the magnitude and phase of the harmonic components. (Note that some researchers implement the high-pass as one minus a low-pass filter, which is a theoretical equivalent.) Figures 5.22 and 5.23 illustrate these two approaches for active filter reference calculation. The SRF method removes only the positive sequence fundamental component, leaving both the negative sequence and the harmonics in the signal so that these components can be injected to cancel out the undesired line currents/voltages.

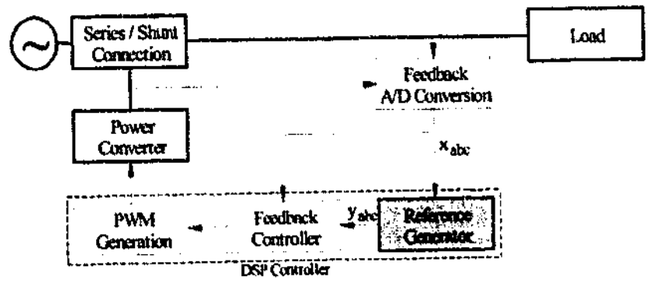


Figure 5.21: Typical active filter block structure.

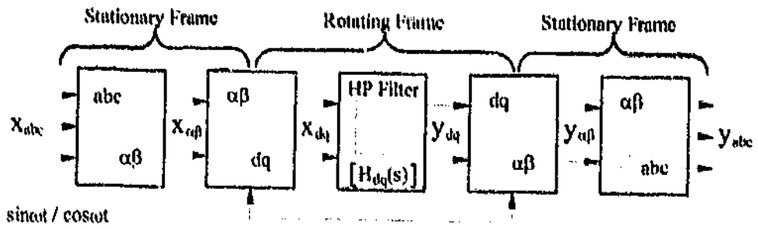


Figure 5.22: Synchronous Reference Frame (SRF) High Pass (HP) type controller.

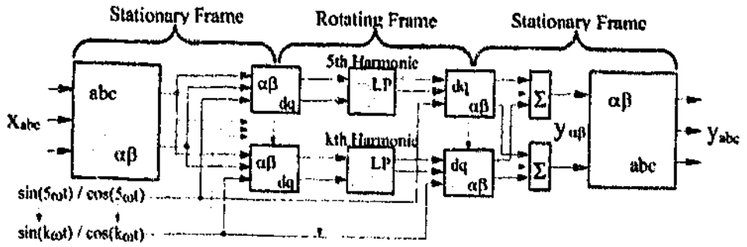


Figure 5.23: Synchronous Reference Frame (SRF) Low Pass (LP) type controller.

5.3.2 Proposed Stationary Reference Frame (StatRF) Signal Extraction

There are two primary advantages in using a stationary frame for the shunt reference generation. Firstly, translating a SRF controller into the stationary frame allows the whole converter system to be developed in a single frame of reference. Hence, classical control techniques can be used to explore the entire system performance, and improved simulation times can also be achieved. Secondly, physical implementation in the stationary frame simplifies the controller software by removing the requirement for sine/cosine reference look-up tables, and also allows specially designed digital filter operations to be readily implemented using fixed-point DSPs for the entire controller block. This significantly reduces the controller loop computation time. Of course, these advantages assume the use of a StatRF transfer function in the reference generator design that is theoretically equivalent to the SRF transfer function. Hence, it is necessary to develop these exact transformations for three-wire,four-wire, and single-phase systems.

This thesis presents an approach for transforming synchronous frame high-pass based generators into notch based active filter reference generators in the Stationary Reference frame (StatRF), which offers significant advantages in a practical implementation (Figure 5.24). The work also presents an exact transformed system including all cross coupling terms. This ensures

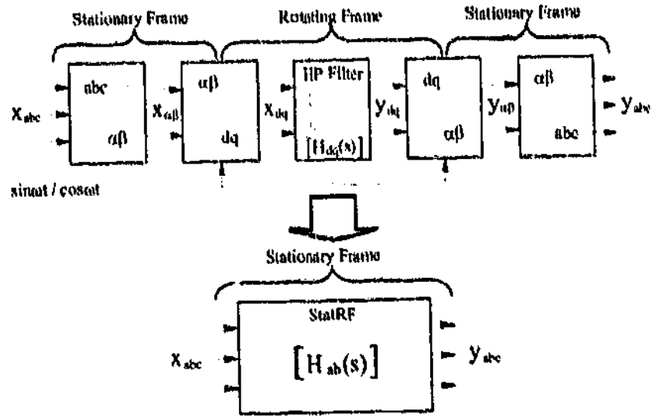
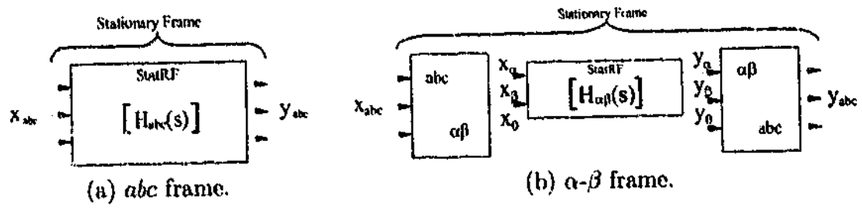


Figure 5.24: Rotating to Stationary Frame conversion of the High Pass (HP) type Synchronous Reference Frame (SRF) harmonic extraction unit.



(a) *abc* frame. (b) α-β frame.

Figure 5.25: Proposed (a) three-wire and (b) four-wire Stationary Reference Frame (StatRF) harmonic extraction units.

identical transient performance to that of a SRF generator. The performance difference between the exact and the band-pass variation is presented, outlining the positive and negative sequence differences between the two controllers. Three alternatives are considered: computation directly in the *abc* frame as shown in Figure 5.25a, which is effective for three-wire three-phase systems; computation in the stationary α-β frame as shown in Figure 5.25b, which better suits four-wire systems; and a version for single-phase systems.

Finally, coefficient rounding, internal integer truncation and pole clustering affects are significant problems when implementing shift-based Infinite Impulse Response (IIR) digital filters on 16-bit fixed-point DSPs (such as the TI TMS320F240 used in Chapter 9). These effects become even more significant at the high sample rates required for active filtering, and have also been identified as a problem for the P+Resonant controller [17]. The shift IIR implementation is briefly investigated to confirm that this problem also extends to the StatRF controller proposed here. This work leads into Chapter 6, which investigates this problem in more detail for all the linear controllers used in the UCPC.

Note that for all harmonic reference generators 'x' and 'y' denote the inputs and outputs, respectively. The sample and cut-off frequencies of the digital filters are defined by ω_s and ω_c, respectively. The choice of ω_c determines the 3-dB frequency of the high-pass filters in the SRF, or alternatively the width of the notch in the StatRF. Larger values of ω_c give better tolerance against system frequency variations, but if too large, they will affect the phase and magnitude of the low order harmonics in the target reference. It should also be noted that the use of the subscript '$_{ab}$' (when discussing three-wire systems) denotes the stationary abc frame where the c phase is not included (since for three-wire systems it can be treated as a redundant variable). When required, the c-phase signal is simply created from the negated summation of the a and b phases.

StatRF for Three-Wire Systems

For three-wire systems, a convenient compact form of the complete SRF reference generator is

$$\begin{bmatrix} y_a(t) \\ y_b(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} \cos(\omega_0 t) & \sin(\omega_0 t) \\ -\sin(\omega_0 t) & \cos(\omega_0 t) \end{bmatrix} \left\{ [\mathbf{h}_{dq}(\mathbf{t})] * \left\{ \begin{bmatrix} \cos(\omega_0 t) & -\sin(\omega_0 t) \\ \sin(\omega_0 t) & \cos(\omega_0 t) \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_a(t) \\ x_b(t) \end{bmatrix} \right\} \right\} \tag{5.13}$$

where $*$ represents a convolution. The transformation matrices are (from right to left): $ab \longrightarrow \alpha$-$\beta$; α-$\beta \longrightarrow d$-q; application of high-pass filter $h_{dq}(t)$ (or $H_{dq}(s)$); d-$q \longrightarrow \alpha$-β; α-$\beta \longrightarrow ab$.

Using the same Laplace methods as proposed by Zmood et al. [112] [113] [114] for the P+Resonant controller, (5.13) can be transformed into an equivalent stationary frame system. The first step in the derivation of the StatRF reference generator is to move the high-pass filter from the rotating d-q frame into the stationary α-β frame using

$$\begin{bmatrix} y_\alpha(t) \\ y_\beta(t) \end{bmatrix} = \begin{bmatrix} \cos(\omega_0 t) & \sin(\omega_0 t) \\ -\sin(\omega_0 t) & \cos(\omega_0 t) \end{bmatrix} \left\{ [\mathbf{h}_{dq}(\mathbf{t})] * \left\{ \begin{bmatrix} \cos(\omega_0 t) & -\sin(\omega_0 t) \\ \sin(\omega_0 t) & \cos(\omega_0 t) \end{bmatrix} \begin{bmatrix} x_\alpha(t) \\ x_\beta(t) \end{bmatrix} \right\} \right\}. \tag{5.14}$$

Expanding (5.14) produces (5.15) and (5.16), which connects the input and output α-β frames of reference.

$$y_\alpha(t) = + \left([\mathbf{h}_{dq}(\mathbf{t})] * \begin{pmatrix} x_\alpha \cos(\omega_0 t) \\ -x_\beta \sin(\omega_0 t) \end{pmatrix} \right) \cos(\omega_0 t) + \left([\mathbf{h}_{dq}(\mathbf{t})] * \begin{pmatrix} x_\alpha \sin(\omega_0 t) \\ +x_\beta \cos(\omega_0 t) \end{pmatrix} \right) \sin(\omega_0 t) \tag{5.15}$$

$$y_\beta(t) = - \left([\mathbf{h}_{dq}(\mathbf{t})] * \begin{pmatrix} x_\alpha \cos(\omega_0 t) \\ -x_\beta \sin(\omega_0 t) \end{pmatrix} \right) \sin(\omega_0 t) + \left([\mathbf{h}_{dq}(\mathbf{t})] * \begin{pmatrix} x_\alpha \sin(\omega_0 t) \\ +x_\beta \cos(\omega_0 t) \end{pmatrix} \right) \cos(\omega_0 t) \tag{5.16}$$

Each of the two outer terms in both (5.15) and (5.16) can be re-arranged using Laplace transforms:

$$\begin{aligned} & \mathcal{L}\left\{ (h_{dq}(t) * (x_\alpha \cos(\omega_0 t) - x_\beta \sin(\omega_0 t))) \cos(\omega_0 t) \right\} \qquad (5.17) \\ = \; & (H_{dq}(s) \mathcal{L}\left\{ x_\alpha \cos(\omega_0 t) - x_\beta \sin(\omega_0 t) \right\}) * \frac{s}{s^2 + \omega_0^2} \\ = \; & \frac{1}{2} \left(H_{dq}(s) \left(X_\alpha(s + j\omega_0) + X_\alpha(s - j\omega_0) - jX_\beta(s + j\omega_0) + jX_\beta(s - j\omega_0) \right) \right) * \frac{s}{s^2 + \omega_0^2} \\ = \; & \frac{1}{4} \begin{pmatrix} H_{dq}(s + j\omega_0) \left(X_\alpha(s + 2j\omega_0) + X_\alpha(s) - jX_\beta(s + 2j\omega_0) + jX_\beta(s) \right) \\ + H_{dq}(s - j\omega_0) \left(X_\alpha(s) + X_\alpha(s - 2j\omega_0) - jX_\beta(s) + jX_\beta(s - 2j\omega_0) \right) \end{pmatrix} \end{aligned}$$

$$\begin{aligned} & \mathcal{L}\left\{ h_{dq}(t) * (x_\alpha \sin(\omega_0 t) + \cos(\omega_0 t))) \sin(\omega_0 t) \right\} \qquad (5.18) \\ = \; & (H_{dq}(s) \mathcal{L}\left\{ x_\alpha \sin(\omega_0 t) x_\beta + \cos(\omega_0 t) \right\}) * \frac{\omega_0}{s^2 + \omega_0^2} \\ = \; & \frac{1}{2} \left(H_{dq}(s) \left(jX_\alpha(s + j\omega_0) - jX_\alpha(s - j\omega_0) + X_\beta(s + j\omega_0) + X_\beta(s - j\omega_0) \right) \right) * \frac{\omega_0}{s^2 + \omega_0^2} \\ = \; & \frac{1}{4} \begin{pmatrix} H_{dq}(s + j\omega_0) \left(-X_\alpha(s + 2j\omega_0) + X_\alpha(s) + jX_\beta(s + 2j\omega_0) + jX_\beta(s) \right) \\ + H_{dq}(s - j\omega_0) \left(+X_\alpha(s) - X_\alpha(s - 2j\omega_0) - jX_\beta(s) - jX_\beta(s - 2j\omega_0) \right) \end{pmatrix} \end{aligned}$$

$$\begin{aligned} & \mathcal{L}\left\{ -(h_{dq}(t) * (x_\alpha \cos(\omega_0 t) - x_\beta \sin(\omega_0 t))) \sin(\omega_0 t) \right\} \qquad (5.19) \\ = \; & (-H_{dq}(s) \mathcal{L}\left\{ x_\alpha \cos(\omega_0 t) - x_\beta \sin(\omega_0 t) \right\}) * \frac{\omega_0}{s^2 + \omega_0^2} \\ = \; & \frac{1}{2} \left(H_{dq}(s) \left(-X_\alpha(s + j\omega_0) - X_\alpha(s - j\omega_0) + jX_\beta(s + j\omega_0) - jX_\beta(s - j\omega_0) \right) \right) * \frac{\omega_0}{s^2 + \omega_0^2} \\ = \; & \frac{1}{4} \begin{pmatrix} H_{dq}(s + j\omega_0) \left(-jX_\alpha(s + 2j\omega_0 - jX_\alpha(s) - X_\beta(s + 2j\omega_0) + X_\beta(s) \right) \\ + H_{dq}(s - j\omega_0) \left(jX_\alpha(s) + jX_\alpha(s - 2j\omega_0) + X_\beta(s) - X_\beta(s - 2j\omega_0) \right) \end{pmatrix} \end{aligned}$$

$$
\begin{aligned}
& \mathcal{L}\left\{(h_{dq}(t) * (x_\alpha \sin(\omega_0 t) + x_\beta \cos(\omega_0 t))) \cos(\omega_0 t)\right\} \qquad (5.20)\\
= \; & (H_{dq}(s)\mathcal{L}\left\{x_\alpha \sin(\omega_0 t) + x_\beta \cos(\omega_0 t)\right\}) * \frac{s}{s^2+\omega_0^2}\\
= \; & \frac{1}{2}\left(H_{dq}(s)\left(jX_\alpha(s+j\omega_0) - jX_\alpha(s-j\omega_0) + X_\beta(s+j\omega_0) + X_\beta(s-j\omega_0)\right)\right) * \frac{s}{s^2+\omega_0^2}\\
= \; & \frac{1}{4}\begin{pmatrix} H_{dq}(s+j\omega_0)\left(jX_\alpha(s+2j\omega_0) - jX_\alpha(s) + X_\beta(s+2j\omega_0) + X_\beta(s)\right)\\ +H_{dq}(s-j\omega_0)\left(jX_\alpha(s) - jX_\alpha(s-2j\omega_0) + X_\beta(s) + X_\beta(s-2j\omega_0)\right)\end{pmatrix}
\end{aligned}
$$

Combining equations (5.17), (5.18), (5.19) and (5.20) together, and re-arranging the results back into matrix form, results in (5.21).

$$
\begin{aligned}
[\mathbf{Y}_{\alpha\beta}(s)] &= \frac{1}{2}\begin{bmatrix} \begin{pmatrix} +H_{dq}(s+j\omega_0)\\ +H_{dq}(s-j\omega_0)\end{pmatrix} & \begin{pmatrix} +jH_{dq}(s+j\omega_0)\\ -jH_{dq}(s-j\omega_0)\end{pmatrix}\\ \begin{pmatrix} -jH_{dq}(s+j\omega_0)\\ +jH_{dq}(s-j\omega_0)\end{pmatrix} & \begin{pmatrix} +H_{dq}(s+j\omega_0)\\ +H_{dq}(s-j\omega_0)\end{pmatrix}\end{bmatrix}[\mathbf{X}_{\alpha\beta}(s)] \qquad (5.21)\\
&= [\mathbf{H}_{\alpha\beta}(s)]\,[\mathbf{X}_{\alpha\beta}(s)]
\end{aligned}
$$

Note that the double frequency components shown in (5.17) cancel with the other outer terms in (5.15) and (5.16). This cancellation is peculiar to three-phase systems, and is the reason why this form of controller does not work directly for single-phase systems.

Finally, the three-wire stationary reference equivalent to the SRF generator of (5.13) is achieved by substituting the high-pass filter function

$$H_{dq}(s) = \frac{s}{s+\omega_c} \qquad (5.22)$$

into (5.21), and then applying the abc/α-β transformations from (5.13) to achieve (refer Appendix C for derivation):

$$
[\mathbf{H}_{ab}(s)] = \begin{bmatrix} \frac{s^2+\omega_c s+\frac{\omega_0\omega_c}{\sqrt{3}}+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} & \frac{\frac{2\omega_0\omega_c}{\sqrt{3}}}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}\\ \frac{\frac{-2\omega_0\omega_c}{\sqrt{3}}}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} & \frac{s^2+\omega_c s-\frac{\omega_0\omega_c}{\sqrt{3}}+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}\end{bmatrix} . \qquad (5.23)
$$

The StatRF generator of Figure 5.25a and (5.23) is now the exact mathematical stationary frame equivalent to the SRF generator of (5.13) with a high-pass filter substituted for $h_{dq}(t)$. Figure 5.26 verifies this equivalence by comparing the frequency characteristics for the two reference generators.

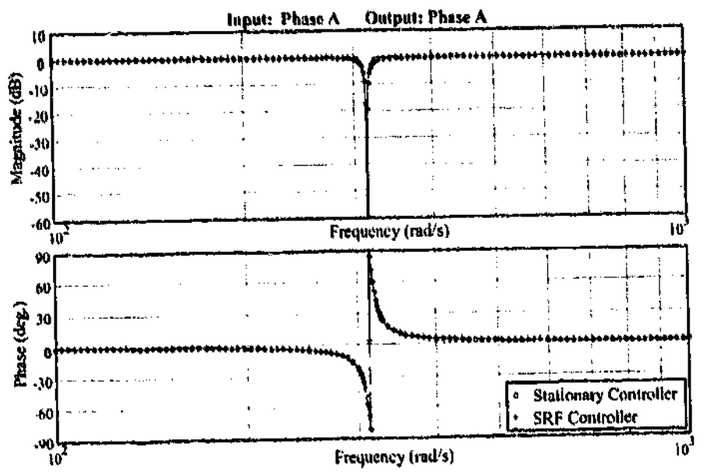


Figure 5.26: Comparison of StatRF and SRF frequency characteristics (phase a only): Simulink simulation.

StatRF for Four-Wire Systems

For unbalanced systems, the full matrix for the StatRF controller ($H_{abc}(s)$) becomes a three by three matrix of sub-generators. This system is easily derived by applying the same method as used in the balanced case, and replacing the abc/α-β transformations applied in the final stage. However, from a practical implementation perspective, the increase in the number of sub transfer functions from 4 to 9 for the StatRF system means that the computational advantages achieved for the SRF generator are reversed. Hence, the stationary generator can only usefully be derived as far as the α-β frame, as shown in Figure 5.25b. The transform is directly obtained by substituting the high-pass filter (5.22) into the α-β generalised formula (5.21) which gives (refer Appendix C for derivation)

$$
[\mathbf{H}_{\alpha\beta}(s)] = \begin{bmatrix} \frac{s^2+\omega_c s+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} & \frac{-\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}\\ \frac{\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} & \frac{s^2+\omega_c s+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}\end{bmatrix} . \qquad (5.24)
$$

StatRF for Single-Phase Systems

To obtain an exact theoretical stationary generator for the SRF, the high-pass filter ($H_{dq}(s)$) was transformed using Laplace transformation techniques. However, the existence of cross-coupling terms exclude its use for single-phase systems. Earlier in this chapter, the review of the

P+Resonant controller showed that if the approximate "low-pass to band-pass formula" [113] is used instead for the transformation, then the result is a symmetrical filter with no cross-coupling terms. This is ideal for use on single-phase systems. The transformation is achieved by performing the substitution

$$H_a(s) = H_b(s) = H_c(s) = H_{dq}\left(\frac{s^2+\omega_0^2}{2s}\right) . \tag{5.25}$$

Application of (5.25) to the SRF generator high-pass filter (5.22) results in a basic narrow-band notch (refer Appendix C for derivation):

$$H_a(s) = H_b(s) = H_c(s) = \frac{s^2+\omega_0^2}{s^2+2\omega_c s+\omega_0^2} . \tag{5.26}$$

As there are now no resulting cross-coupling terms, the matrix presentation format can be omitted, and for single-phase usage the functions $H_b(s)$ and $H_c(s)$ are not required.

The P+Resonant review also noted the work by Mattavelli [110], where it was shown that the summation of both the positive and negative results from the Laplace transformation version resulted in full cancellation of the cross-coupling terms and a result very similar to the P+Resonant controller was developed using the approximate transformation. If the positive and negative (i.e. ω_0 becomes -ω_0) sequence versions of (5.24) are added together, then the result is

$$\begin{aligned}[\mathbf{H}_{\alpha\beta}(s)] &= \begin{bmatrix} \left(\frac{s^2+\omega_c s+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}+\frac{s^2+\omega_c s+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}\right) & \left(\frac{-\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}+\frac{\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}\right) \\ \left(\frac{\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}+\frac{-\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}\right) & \left(\frac{s^2+\omega_c s+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}+\frac{s^2+\omega_c s+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}\right) \end{bmatrix} \\ &= \begin{bmatrix} \frac{2s^2+2\omega_c s+2\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} & 0 \\ 0 & \frac{2s^2+2\omega_c s+2\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} \end{bmatrix} \end{aligned} \tag{5.27}$$

which, as expected, also gives full cancellation of the coupling terms. Since the *abc* to α-β conversion makes no sense for single-phase systems, the direct terms in (5.27) can therefore be used and this results (after scaling down by two) in the single-phase form of the StatRF reference generator of:

$$H_a(s) = H_b(s) = H_c(s) = \frac{s^2+\omega_c s+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} . \tag{5.28}$$

It can be seen that the difference between this form and the earlier narrowband notch is the additional ω_c terms, and considering that $\omega_0 \gg \omega_c$, then this difference is very minimal (as was the case for the equivalent three-phase P+Resonant versions).

Since (5.28) has no cross-coupling terms, it has the advantages of half the computational effort of the StatRF controllers previously defined, and the ability to work on single-phase systems. The filter is symmetrical around 0 Hz, and will therefore also remove negative sequence components from the input signal. In contrast, the SRF and three-phase StatRF generators do not remove the negative-sequence component from the signal. This comparison is shown in Figure 5.27. Thus, active filters employing the single-phase methods on three-phase systems are unable to compensate for negative-sequence fundamental currents/voltages.

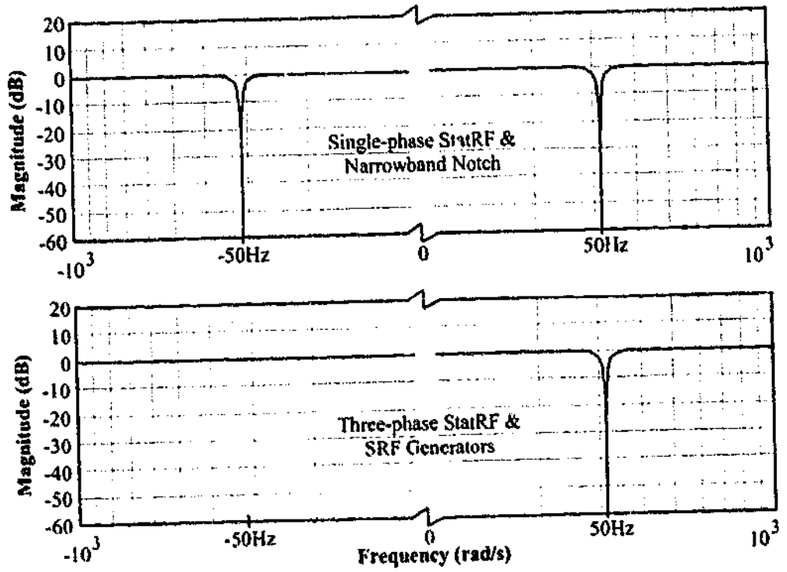


Figure 5.27: SRF, StatRF and narrow-band notch comparison.

5.3.3 Practical StatRF Fixed-Point DSP Implementations

Switching frequencies in excess of 15 kHz are not uncommon for active filters, and for systems employing asymmetrical regular sampled PWM, the sampling frequency is twice this value. The total software interrupt time for these systems is therefore an important design consideration when developing a system. The advantage of the StatRF is that it reduces the total number of required multiplications compared to the SRF, and also shifts the whole controller into a digital filter form such that instructions specifically designed for this purpose can be fully utilized on a fixed-point DSP.

Furthermore, the StatRF does not require sine and cosine references, in contrast to the SRF controller. These references typically require large look-up tables to be stored within the DSP, often in off-chip memory, which can also increase the code execution time. The use of sine/cosine look-up tables also introduces an extra source of quantization noise into the discrete system. This source of noise was measured to be relatively significant during the experimental development of the SRF generator, especially since the aim of the active filter is to compensate for harmonics that may only be a few percent of the fundamental signal (see Chapter 9 for experimental results).

In this section the available practical approaches will now be investigated that allow for drift in the fundamental frequency when implementing the StatRF (with comparison back to the SRF method). In addition, this section begins the investigation into the digital implementation of the StatRF controller, establishing the basis for the further work presented in Chapter 6.

Parameter Selection for Systems with Frequency Drift

The choice of the cut-off frequency ω_c and the sample frequency ω_s can greatly affect how the StatRF generator is implemented. For systems designed just for harmonic compensation (not sub-harmonic flicker compensation) a wider (i.e. larger) value of ω_c can be chosen. Depending on the maximum allowable system frequency swing (and on the requirements of other controllers in the system), it may then be possible to eliminate the need for synchronization hardware and software, and still obtain the required fundamental attenuation.

For all other systems, the coefficients can be dynamically created in background software with minimal effort using pre-calculated versions of (5.31). If this option is too computationally expensive, a small look-up table of filter parameters can be used for selected frequencies around the nominal. For example, a StatRF with a f_c of 5 Hz will still achieve 40-dB attenuation with frequency variations of up to 0.05 Hz. If the maximum swing of the connected system is say

± 0.5 Hz, then this corresponds to 11 sets of coefficients and therefore 99 16-bit values (198 bytes) for a three-phase StatRF (and even less for a single-phase implementation). Note that the overlap shown in (5.31) means that only nine of the coefficients actually need to be stored. This is compared to the 1024 values (2048 bytes) needed for even a small sine lookup table for a synchronous frame implementation.

Digital Implementation

Fixed-point DSPs offer single-chip advantages for many power electronic applications [17], although accurate implementation of digital filters in the presence of quantization, coefficient rounding and fixed-point truncation errors requires careful attention. It is already known that problems exist with the implementation of the P+Resonant controller using infinite impulse response (IIR) digital filters on 16-bit fixed-point DSPs [17] – especially when used for selective harmonic compensation. This section briefly investigates these implementation issues for the proposed StatRF controller to verify that such problems also exist here.

For discrete shift operator based IIR implementations a bilinear (Tustin) transformation was applied to the s-domain transfer functions to achieve z-domain digital filters in the commonly used form of

$$H_q(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-1}} \tag{5.29}$$

where constants a_n and b_n are the digital filter coefficients.

Using the StatRF controller of (5.23), the digital filter matrix $[\mathbf{H}_{ab}(\mathbf{z})]$ given in (5.31) is derived (with or without pre-warping) using the Tustin (bilinear) transformation

$$s = k\frac{z-1}{z+1}. \tag{5.30}$$

$$[\mathbf{H}_{ab}(z)] = \begin{bmatrix} \dfrac{b_{0\_11} + b_{1\_11}z^{-1} + b_{2\_11}z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} & \dfrac{b_{0\_12} + b_{1\_12}z^{-1} + b_{2\_12}z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \\ \dfrac{b_{0\_21} + b_{1\_21}z^{-1} + b_{2\_21}z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} & \dfrac{b_{0\_22} + b_{1\_22}z^{-1} + b_{2\_22}z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \end{bmatrix}$$

where

$$a_1 = \frac{-2k^2+2\omega_c^2+2\omega_0^2}{k^2+2\omega_c k+\omega_c^2+\omega_0^2}, \qquad a_2 = \frac{k^2-2\omega_c k+\omega_c^2+\omega_0^2}{k^2+2\omega_c k+\omega_c^2+\omega_0^2} \tag{5.31}$$

$$b_{0\_11} = \frac{k^2+\omega_c k+\omega_c\omega_0/\sqrt{3}+\omega_0^2}{k^2+2\omega_c k+\omega_c^2+\omega_0^2},\ b_{1\_11} = \frac{-2k^2+2\omega_c\omega_0/\sqrt{3}+2\omega_0^2}{k^2+2\omega_c k+\omega_c^2+\omega_0^2},\ b_{2\_11} = \frac{k^2-\omega_c k+\omega_c\omega_0/\sqrt{3}+\omega_0^2}{k^2+2\omega_c k+\omega_c^2+\omega_0^2}$$

$$b_{0\_22} = \frac{k^2+\omega_c k-\omega_c\omega_0/\sqrt{3}+\omega_0^2}{k^2+2\omega_c k+\omega_c^2+\omega_0^2},\ b_{1\_22} = \frac{-2k^2-2\omega_c\omega_0/\sqrt{3}+2\omega_0^2}{k^2+2\omega_c k+\omega_c^2+\omega_0^2},\ b_{2\_22} = \frac{k^2-\omega_c k-\omega_c\omega_0/\sqrt{3}+\omega_0^2}{k^2+2\omega_c k+\omega_c^2+\omega_0^2}$$

$$b_{0\_12} = -b_{0\_21} = b_{2\_12} = -b_{2\_21} = \frac{2\omega_c\omega_0/\sqrt{3}}{k^2+2\omega_c k+\omega_c^2+\omega_0^2}$$

$$b_{1\_12} = -b_{1\_21} = 2b_{0\_12},\ \textit{and}\ k = \left(\frac{2}{T}\ \textit{or}\ \frac{\omega_c}{\tan(\omega T/2)}\right).$$

As the ratio between the sample frequency and the cut-off frequency increases for an IIR digital filter, the minimum required integer bit size also increases. Whilst this problem occurs for both the high-pass filters of the SRF, and the notch based filters of the StatRF, substitution into the discrete formula for both the controllers shows that the StatRF controller is more sensitive.

It has also been found that truncation due to the fixed-point operations in the shift-based filters introduces dc offsets in the outputs of the digital filters. Figure 5.28 shows how the cross-coupling terms of the digital filters in $H_{ab}(s)$ are implemented. However, negating the signs of the numerator coefficients in (5.31) such that the cross coupling terms (H_{12} & H_{21}) are subtracted from the direct terms (H_{11} & H_{22}) (instead of summed) causes a 30-dB drop in the unwanted dc offset. This occurs since the integer truncation is always in the downward direction, so by subtracting one filter from another, the dc offset from the direct filter is cancelled out by that from the cross coupled filter.

Simulations were developed using the Simulink package in Matlab to accurately model the

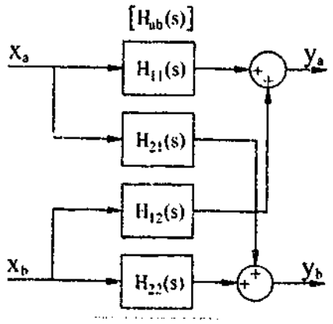
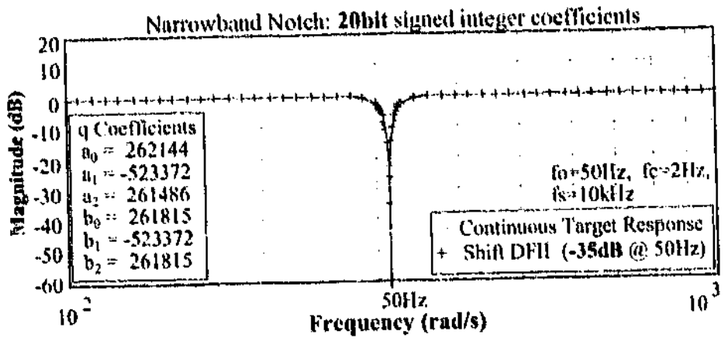
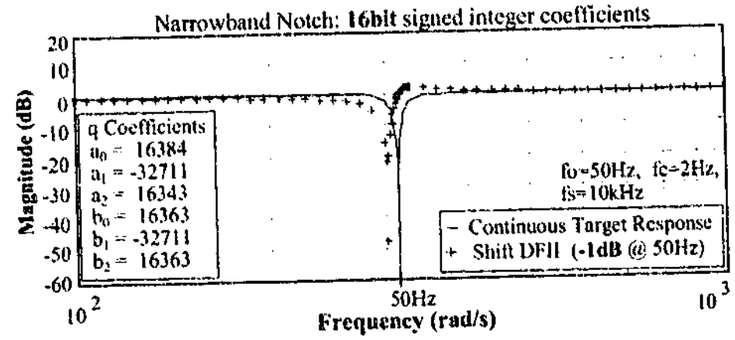


Figure 5.28: Connection of controllers in Matrix $[\mathbf{H}_{ab}(s)]$.



(a) 20 bit implementation



(b) 16-bit implementation

Figure 5.29: Frequency response for continuous and shift IIR filters with: (a) 20-bit; (b) 16-bit; coefficients.

shift-based IIR digital filters as they were experimentally implemented on the TI TMS320F240 DSP. The simulations included all integer rounding and truncation effects of each DSP calculation (details of the model are presented in Chapter 8). The results were extensively compared against the actual DSP implementation, and were used to produce the frequency characteristic of the shift-based implementation of the StatRF generator shown in Figures 5.29a and 5.29b. This figure shows that a shift-based StatRF filter can be satisfactorily implemented using 20-bit signed integer coefficients, with a possible notch attenuation of -35-dB. However, using 16-bit calculations the shift filter causes the notch centre frequency to shift, and only achieves -1-dB attenuation at the required frequency.

For the TI TMS320F240 DSP used in the experimental work in Chapter 9, 16-bit arithmetic is highly desirable, but for most high ratios between ω_c and ω_s the StatRF filter is not realizable unless either 32-bit arithmetic is used or other solutions are implemented. For the proposed solution to be viable, a practical implementation solution must be found. The experimental results are therefore deferred until Chapter 9, after this practical problem has been resolved.

5.4 Summary

This chapter has investigated the use of stationary frame control to enable fast implementation of the UCPC (even with multiple harmonic controllers), as well as to allow for single-phase operation, which was not possible using the rotating frame controllers initially presented in Chapter 4.

The P+Resonant controller proposed by Zmood et al. [114] was reviewed, and has provided the basis for the majority of the following work. The effects of each of the parameters of a P+Resonant controller were investigated to design the selective harmonic compensation controller of the series converter of the UCPC. The use of feed-forward control was also discussed, along with the proposed use of an active damping scheme to provide better stability and improved *LC* filter resonance damping for the system. This investigation allowed a design process to be established, and the final parameters for the LV experimental system (Chapter 9) developed using this approach were shown to have good stability, as well as good rejection of the selected voltage error components.

For the shunt converter, Chapter 4 explored the use of a rotating frame signal extraction unit to create the reference current required by the inner deadbeat current controller. However, once again this unit is unable to work with single-phase systems, as well as having other undesirable implementation overhead problems. This chapter has developed a stationary frame equivalent version of this controller, applying the same techniques as were used for the development of the P+Resonant controller. Both three-phase (three-wire and four-wire) and single-phase alternatives were presented, noting that the single-phase alternative was nearly identical to a basic narrow-band notch filter type controller.

Finally, the conversion of the proposed control schemes to a 16-bit fixed-point DSP platform has been shown to have problems if traditional shift IIR digital filters are used. Therefore, a solution to this problem is required if the proposed controller is to be viable in a practical system. The following chapter investigates these practical problems in more detail, and explores various options to develop an effective solution.

Chapter 6

Digital Control: The Delta Operator

[1]Most Infinite Impulse Response (IIR) digital filter implementations in power electronic converter applications are based on the time shift operator q and its associated z-transform. However, Chapter 5 concluded by identifying the problems associated with using shift-based IIR digital filters for the implementation of stationary frame linear controllers which are based on high Q second order filters. This problem is particularly severe for the higher sample frequencies (compared to the system poles) typically used in Custom Power applications, and it is also aggravated by commonly used 16-bit fixed-point DSP hardware implementations. Therefore, an alternative approach to implement the proposed stationary frame controllers must be found.

The first part of this chapter presents the current state of the art in digital signal processing techniques used for real-time digital control of Custom Power applications, and explores possible techniques from the signal-processing field that could be applied to solve these problems. The primary focus is on the suitability of IIR and FIR (Finite Impulse Response) digital filtering techniques for this research, which are the defacto standards for such implementations.

The delta operator applied to IIR digital filters is then proposed as a solution for the implementation of stationary frame controllers at high sample rates on fixed-point processors. Existing knowledge on this technique is formed to present the relevant information required for implementation of delta-based IIR digital filters in real-time control of power converters. A design process for the conversion of conventional shift-based IIR knowledge is presented to allow an easy transition from conventional methods. Two practical examples (from the UCPC shunt and series controllers) are provided to highlight the advantages of the proposed approach.

[1]The material in this chapter was first published in part as:
M. J. Newman and D. G. Holmes, "Delta Operator Digital Filters for High Performance Inverter Applications," *IEEE Trans. on Pow. Electronics*, vol. 18, no. 1, Jan., pp. 447-454, 2003.
M. J. Newman and D. G. Holmes, "Delta Operator Digital Filters for High Performance Inverter Applications," in *Conf. Rec. IEEE/PESC*, Cairns, Qld, Australia, 2002.

6.1 Digital Control in Custom Power Applications

Direct digital control of high performance converter applications (such as Custom Power) is now commonplace because of advantages such as the ability to readily upgrade the control algorithm, insensitivity to component parameters, aging and thermal drift effects, and flexibility. Figure 6.1 shows the signal flow of a Custom Power application (with a fully digital controller) including both the analog and digital components.

A single DSP (Digital Signal Processor) is used for both the digital controller and the generation of the PWM signals in the Figure 6.1 example. Dedicated DSP's optimized for this combined purpose are readily available, and include specific hardware such as PWM generation units. Such devices include the C24x and C2000 ranges in the Texas Instruments (TI) TMS320 family, the Motorola DSP56800 family, and the Analog Devices ADMC401 – all of which are 16-bit fixed-point DSPs. Therefore, the accurate fixed-point representation of controller transfer functions is particularly important.

Figure 6.1 is an example of a fully digital controller, where once the measurement signals are converted to digital form there are no further analog components before the gate driver circuitry that controls the power converter devices. (In contrast, an example of a partially digital controller would be a shunt active filter with an analog hysteresis current regulator, located after a digital reference signal processing controller.) Of course, even for a fully digital controller, the actual power system variables are analog, so there will always be a need for some analog circuitry to convert the measured signal into digital form. This can be either done using either a measurement sub-system within the DSP, or with separate circuitry. For converter applications, most measurement devices have analog outputs (e.g. hall effect and current transformers for current measurement, and resistive and capacitive dividers for voltage measurement), and therefore analog input conditioning circuitry is generally required. As well as the analog magnitude scaling, which is required to convert the signals to a voltage suitable for analog to digital conversion, analog anti-aliasing filters are usually necessary. Anti-aliasing filters attenuate input signals close to (and above) the sampling frequency of the controller, so that the aliasing of the digital controller at multiples of the sample frequency does not affect the system performance.

The alternatives to a single DSP controller solution are to use either multiple DSPs / Microcontrollers, or to use a combination of a DSP and external digital hardware for the PWM generation (typically a FPGA, or similar, is used). These solutions allow a floating-point processor to be used, which reduces the susceptibility of the system to errors caused by coefficient

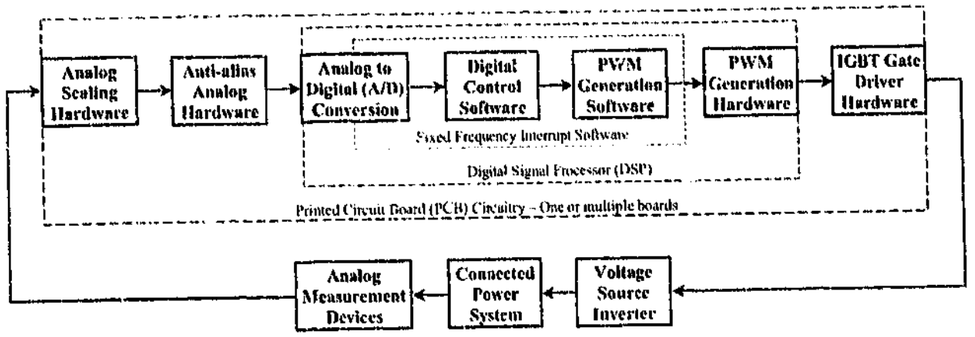


Figure 6.1: Control implementation block diagram for a general Custom Power application with a single-chip DSP digital controller.

rounding and numerical truncation. Increased processing speed of available processors (compared to the speed of converter control DSPs) can also be a driving force for a split solution. However, the disadvantage of these solutions is an increase in size and cost, as well as increased hardware and software design complexity (since extra hardware and software interfaces need to be developed).

The experimental work presented in this thesis uses both a single DSP (fixed-point), and a combined DSP/Microcontroller (floating-point/fixed-point) for the low voltage (Chapter 9) and medium voltage (Chapter 10) experimental work, respectively. However, since fixed-point digital controllers are in common use, and are much more susceptible to errors than finite word length floating-point controllers, only the implementation of fixed-point digital controllers is presented in this thesis.

6.2 Typical Digital Filter Solutions

Filter transfer functions make up a large portion of linear control systems. For the stationary frame controllers used in this work, the entire controller is a filter. Therefore, for these systems, the digital control implementation is simply the application of digital filters to the chosen processing platform. However, for applications where precise accuracy is required, quantization, truncation and rounding effects (caused by digitization and finite word length/fixed-point processes) in these digital filters can become significant issues that may compromise the performance of the digital controller.

The fundamental forms of digital filters are the Infinite Impulse Response (IIR) filter

$$y[n] = \sum_{k=0}^{\infty} h[k]\, x[n-k], \tag{6.1}$$

and the Finite Impulse Response (FIR) filter

$$y[n] = \sum_{k=0}^{n} h[k]\, x[n-k], \tag{6.2}$$

where $x[k]$ is the discrete sampled input signal, $y[k]$ is the sampled output signal, and $h[k]$ is the filter function. This section reviews both of these filters to assess their suitability for the implementation of the stationary frame filters required by the UCPC. The reasons for their limitation (noted already in Chapter 5 for the IIR method) are also detailed. Alternative methods are then briefly reviewed and this allows a solution to the fixed-point implementation problem to then be proposed.

To review the suitability of a digital filter system, a set of criteria has been established. Digital filters suitable for the control systems in this research should:

- Criterion 1: Be implementable on a 16-bit fixed-point DSP with the second order resonant/notch filter functions discussed in Chapter 5, even with large pole to sample ratios and/or a relatively narrow band/notch.
- Criterion 2: Have a close relationship to the response of the corresponding s-domain transfer function (even on a fixed-point 16-bit implementation) and should allow for an easy design transition from the continuous s-domain transfer function to its digital equivalent.
- Criterion 3: Have minimal software computational complexity and variable memory space requirements.
- Criterion 4: Be stable if the corresponding continuous equivalent controller is stable.

6.2.1 Shift operator based Infinite Impulse Response (IIR) Filters

Currently, most real time Infinite Impulse Response (IIR) digital filter implementations in power electronic applications use the time shift operator q and its associated z-transform. However, it is known that, as the sample period approaches zero for z-transform discrete systems, the dynamic response does not converge smoothly to its continuous counterpart [138]. Hence, as the sample rate of a digital controller increases, problems that are inherent with shift-based digital

filters become more severe. Furthermore, the digital resonant/notch filters in the stationary frame controllers used in this research are more sensitive to coefficient rounding errors in IIR implementation than their rotating frame equivalents, and so often a floating-point or a 32-bit fixed-point processor is required for satisfactory performance [17]. The reasons for this sensitivity are outlined below, but for completeness, a brief review of a conventional shift-based IIR filter is firstly presented to provide a basis for this discussion, and to allow the solution to be developed in the following section.

The IIR filter function presented in (6.1) is written as having infinite length – which is obviously non-causal. However, it is the presence of this infinite prior knowledge of the past signal that distinguishes the IIR from the FIR, and which allows it to more accurately match the response of a continuous filter. The prior knowledge therefore cannot be discarded (otherwise it would simply be an FIR filter), but it can be managed using recursive functions, which enable the required prior signal knowledge to be remembered without the need for infinite, or even large, storage requirements. The disadvantages of this approach are opportunities for instability, as well as additional sensitivities due to the nature of the recursive system.

A recursive system requires the use of previous input and output samples. There is typically a fixed time shift (also called a time delay) between these samples. A mathematical operator has been defined to denote this delay, and is known as the shift (or delay) operator, q. The shift operator q, and its associated z-transform, are synonymous with the implementation of IIR digital filters, and in many texts the discussion of one is almost certainly followed by the other [139]- [142]. The application of the shift operator to a given input discrete-time sample $u[k]$ is simply the future sample, as in

$$u[k+1] = qu[k] \qquad \text{or equivalently} \qquad u[k+1] = zu[k]. \tag{6.3}$$

Note that if initial conditions are ignored, the shift operator q can simply be replaced by z [143]. In a practical sense this is obviously not causal, but the inverse shift operator can be applied as a causal alternative

$$z^{-1}u[k] = u[k-1]. \tag{6.4}$$

Any linear discrete-time system (2^{nd} order in this case) can be described by a linear difference

equation [140]

$$y[k] + a_1 y[k-1] + a_2 y[k-2] = b_0 y[k] + b_1 y[k-1] + b_2 y[k-2]. \tag{6.5}$$

Note that only a second order transfer function has been considered, as it is usually preferable to implement higher order transfer functions as a cascade of first and second order functions, to minimize rounding and truncation effects [141]. Furthermore, the maximum filter order for all the linear controllers used in this research is second order.

Applying (6.4) results in the general form for the output, $y[k]$, of an second order Infinite Impulse Response (IIR) filter, if

$$y[k] = b_0 y[k] + b_1 z^{-1} y[k-1] + b_2 z^{-2} y[k-2] - a_1 z^{-1} y[k-1] - a_2 z^{-2} y[k-2]. \tag{6.6}$$

Finally, rearrangement of (6.6) provides the canonical form of the shift-based IIR filter, $H_q(z)$, which gives an output of $Y(z)$ when applied to an input sequence $X(z)$:

$$H_q(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-1}}. \tag{6.7}$$

With the structure of the shift-based 2^{nd} order IIR filter realized, implementation of (6.7) still requires the shift filter coefficients (a_k & b_k). Whilst there are many methods for acquisition of these coefficients (including impulse invariance [139] [140], pole zero placement method [139], step invariance [140], and match z transformation methods [140]), the bilinear (Tustin) transformation is the more popular choice [139], as it will always provide a stable digital filter as long as the continuous filter is stable. This method involves the substitution of

$$s = \frac{2}{T}\frac{z-1}{z+1} \tag{6.8}$$

into the desired s-domain transfer function, and rearranging the result into the form of (6.7) to obtain the associated coefficient values.

The transformation from the continuous Laplace s-domain into the discrete-time z-domain, maps the infinite frequency range $[0, \infty)$ to the finite range $[0, \pi/T)$ [142]. This mapping is non-linear and the frequency axis is compressed, with the effect becoming particularly significant for higher sampling frequencies. The importance of the compression for shift-based IIR filters is that as the sampling frequency increases, the poles of the filter will converge towards 1 on the z-plane. This clustering effect makes very small changes in the pole locations (i.e. because

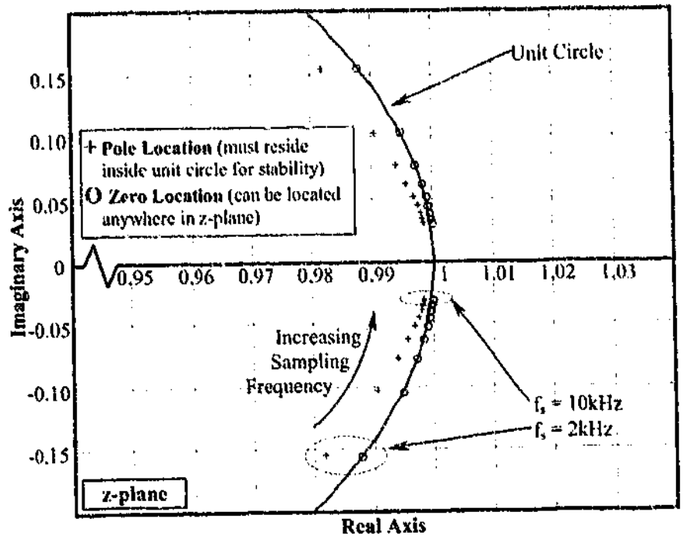


Figure 6.2: Pole clustering effect of a narrowband notch shift-based IIR digital filter (sampling frequency = 2 kHz - 10 kHz).

of coefficient rounding) cause very large deviations from the intended transfer function. Figure 6.2 illustrates this effect for a shift implementation of a narrowband notch. Furthermore, for stability, the poles of $H(z)$ must lie within the unit circle on the z-plane (the zeros can be anywhere), and therefore as the frequency increases the poles move closer to this stability limit (Figure 6.2). Finally, for practical controller implementations these problems will be further aggravated by Finite Word Length (FWL) floating point and fixed-point calculation errors.

The suitability of the IIR method for the implementation of the stationary frame controllers in this research can be summarized as shown below.

Suitability Summary of IIR Filters for Implementation of Stationary Frame Controllers:

- Criterion 1 is partially satisfied. Implementation is possible on a 16-bit fixed point DSP, but only for some controllers where the sample frequency is sufficiently low and the narrowband width is not too narrow.
- Criterion 2 is partially satisfied. The IIR filter has a relatively close relationship to the corresponding s-domain transfer function (i.e. continuous filter), but will diverge as the sample frequency increases or the notch/resonant width of the controller is decreased (i.e.

when f_C is decreased).

- Criterion 3 is fully satisfied. The computational and variable memory overheads are low.
- Criterion 4 is partially satisfied. IIR filters designed using the bilinear transformation are stable if their corresponding continuous filters are stable. However, when the coefficients of the controllers used in this research are converted to fixed-point values, they have been found to cause unstable situations sometimes because of rounding errors.

6.2.2 Finite Impulse Response (FIR) Filters

Unlike the IIR filter, the filter function for an FIR filter (6.2) is finite – as the name suggests. The advantage is that a non-recursive implementation is possible, which cannot become unstable. The disadvantage is that the loss of information in the filter function means that the continuous equivalent may not be properly represented. The discrete representation of the continuous filter (criterion 2) is of significant importance, especially for the proposed stationary reference frame harmonic extraction control block (Chapter 5) as it relies on the correct cancellation of the direct and cross-coupled filters to achieve the desired response. For the P+Resonant filter, the error between the continuous and digital versions is not as critical, as the primary objective is to provide a high gain at the selected frequency, and very low gains elsewhere. However, a close match for both phase and magnitude is desirable, otherwise the results of the stability analysis may become invalid.

The implementation of an FIR filter can be described by

$$y[n] = \sum_{k=0}^{N-1} h_w[k]\, x[n-k]. \tag{6.9}$$

where in this instance $h_w[k]$ is a time windowed set of samples from the discrete filter infinite impulse $h[k]$ [144] [139]. Note that $h[k]$ is the sampled form of the time domain impulse response $h(t)$ of the continuous filter. The most basic form of windowing is the rectangular window

$$w_{rect}[k] = \begin{cases} 1, & 0 < k < N \\ 0, & \text{otherwise} \end{cases}, \tag{6.10}$$

such that the windowed sample values are unchanged from the original, and the values outside

the finite range are removed. i.e.

$$h_w[k] = h[k]\, w_{rect}[k] \tag{6.11}$$

Using the shift operator, z, already defined, the filter transfer function for the FIR digital filter is then formed [144] to be

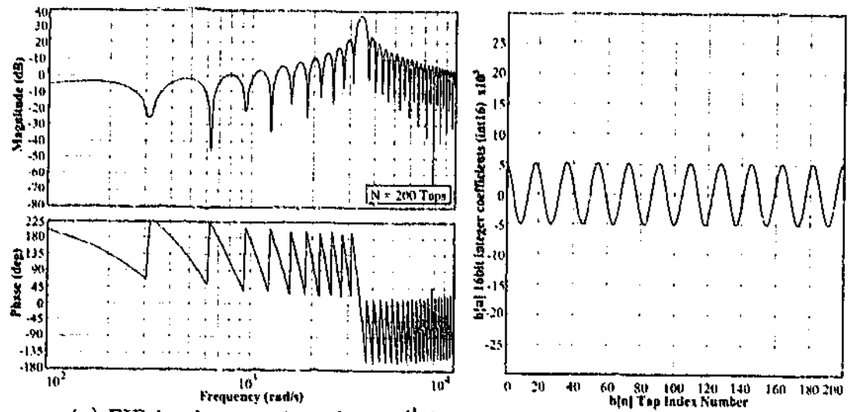
$$H(z) = \sum_{k=0}^{N-1} h_w[k]\, z^{-k} \tag{6.12}$$

To illustrate the effects of windowing (and therefore loss of information from the continuous filter) an FIR filter for the resonant portion of a P+Resonant filter is developed. Note that while many methods are available for the calculation of the filter functions coefficients $h_w[k]$, most of these methods are aimed at directly creating a custom response [139] [140], not creating an equivalent response to a continuous filter as required here. In this instance, the filter coefficients are found directly from the continuous filter by using the Laplace transform of the continuous resonant controller to give the impulse response $h(t)$. For the ideal P+Resonant controller (5.5) the impulse response is given by

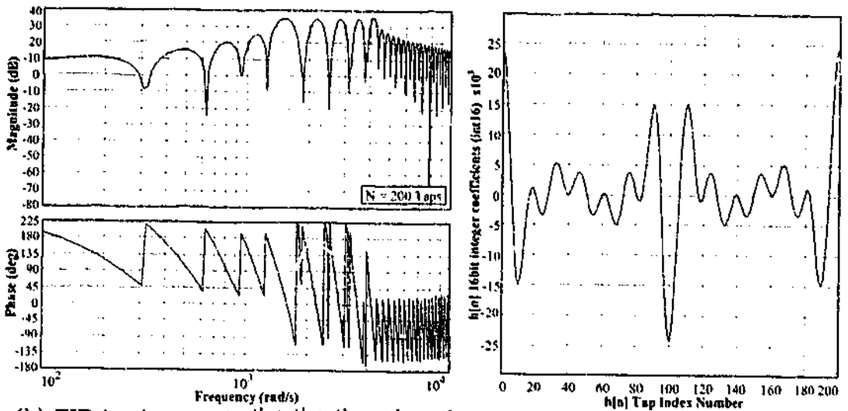
$$\begin{aligned} h(t) &= \mathcal{L}\{H_{P+R}(s)\} \\ &= \mathcal{L}\left\{2K_I \tfrac{s}{s^2+\omega_n^2}\right\} \\ &= 2K_I cos\omega_n t \end{aligned} \tag{6.13}$$

which can then be directly sampled to produce $h_w[k]$, as illustrated in Figure 6.3a for the 11^{th} harmonic (i.e. $\omega_n = 100\pi \times 11$). Note that this response is very different to that of the desired continuous filter, with the most distinctive difference being the introduced side-lobes centered around the 11^{th} harmonic. These side-lobes are a consequence of the loss of information due to the windowing, which is known as the leakage (Gibbs) effect. Figure 6.3b shows the result of a single FIR implementation for multiple resonant controllers.

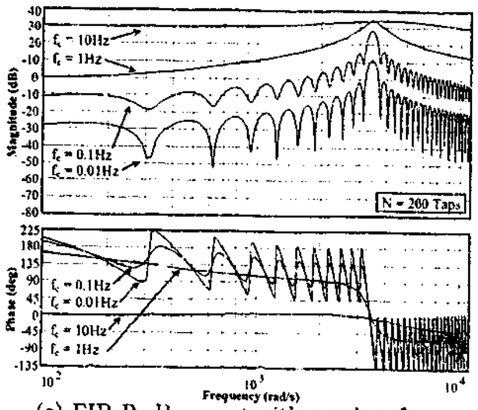
Since the impulse response of the ideal resonant controller is a sinusoid, the discarded values outside the window are significant. For the damped form of the P+Resonant filter, the impulse will decay, and therefore the leakage should be less. Once again the impulse response is developed



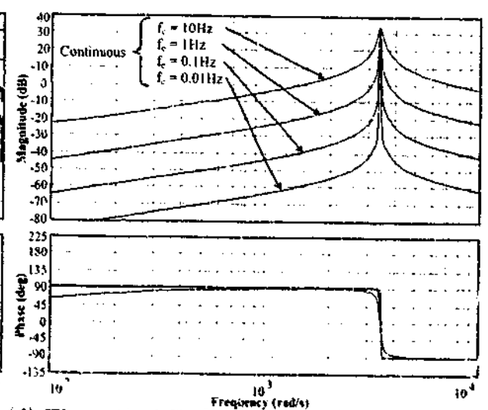
(a) FIR implementation of an 11^{th} harmonic P+Resonant controller.



(b) FIR implemented $5^{th}/7^{th}/9^{th}/11^{th}/13^{th}$ harmonic P+Resonant controllers.



(c) FIR P+Resonant with varying f_C.



(d) Theoretical P+Resonant with varying f_C.

Figure 6.3: Simulations of FIR 16-bit fixed point digital implementations.

to give

$$\begin{aligned} h(t) &= \mathcal{L}\{H_{P+R}(s)\} \\ &= \mathcal{L}\left\{2K_I\omega_c\frac{s+\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_n^2}\right\} \\ &= 2K_I\omega_c\mathcal{L}\left\{\frac{(s+\omega_c)}{(s+\omega_c)^2+\omega_n^2}\right\} \\ &= 2K_I\omega_c e^{-\omega_c t}cos\omega_n t \end{aligned} \tag{6.14}$$

The effect of varying f_C is illustrated in Figure 6.3c, with the target theoretical frequency responses given in Figure 6.3d. This shows that the two larger values of f_C show no obvious leakage effects. This is confirmed in Figure 6.4 which shows that the impulse responses of these two filters drop to near zero before the windowing occurs. However, even with this reduction in leakage, the 200 samples (also known as Taps) in the example are not sufficient to converge to the response of the continuous filters in Figure 6.3d. Increasing the number of Taps allows the FIR implementation to converge on the continuous filter (Figure 6.5), but an unrealistically large number of Taps is required to produce a reasonable frequency response.

From this example it can be seen that to use the FIR method to accurately implement a digital equivalent of the narrowband filters would be impractical, due to the large number of samples required. However, while the StatRF harmonic extraction controller used in this research requires the digital controller implementation to match the continuous controller, this may not be the case for the P+Resonant controller in general. Mattavelli et al. [17] used a FIR filter to overcome 16-bit fixed-point limitations when implementing a UPS with selective removal of voltage harmonics from its output. To provide a sharper roll-off characteristic a positive feedback term of z^{-N} was also used. However, the large variation in response (especially in phase) means that the controller design values are no longer valid. It is also unclear how the rapid phase variations seen at higher frequencies might affect the stability of the system.

The FIR filter has the advantage that as the number of selected harmonics increase, the computational time is not affected, but this is offset by the large initial overheads. For example, if 200 Taps were used for a FIR P+Resonant implementation (as per the previous example and [17]), then with a 50 ns instruction time (plus initial overheads) the TI TMS320F240 DSP would take approximately 11 μs to execute the algorithm (assuming pipelined operation of the MAC instruction), and would require 800 bytes of memory for each phase to store the previous input samples and the filter array. For an IIR implementation, each separate P+Resonant controller was measured to execute in less than 2 μs (i.e. less than 12 μs overall for the six selected controllers). The overall execution difference between the IIR and FIR methods depends on

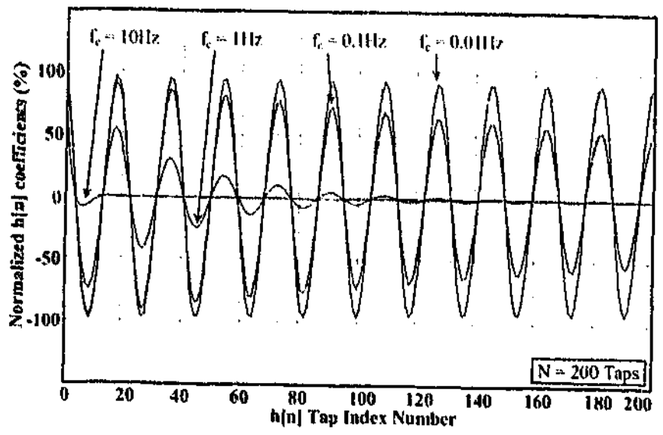


Figure 6.4: Comparison of 11^{th} harmonic P+Resonant FIR $h[n]$ coefficients with varying f_c.

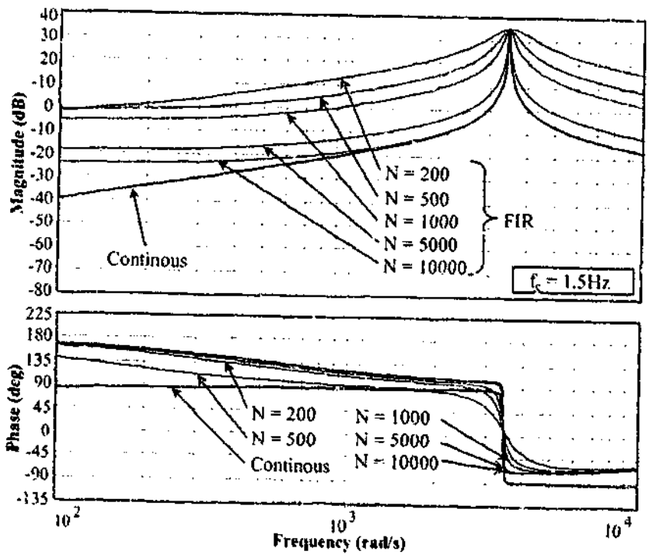


Figure 6.5: Comparison of 11^{th} harmonic P+Resonant FIR with varying N.

the number of required parallel controllers, with the cross-over point being approximately six. However, the IIR also only requires 16 bytes of memory for each controller (48 bytes overall for six) to store the previous samples, which is orders of magnitude smaller than the FIR method. Finally, for changes in fundamental frequency the FIR method would have to calculate new values for, and then update, the entire 200 Tap array, while only 5 values for each IIR P+Resonant controller would need to be updated (i.e. 30 values for 6 controllers).

The suitability of the FIR method for the implementation of the stationary frame controllers in this research can be summarized as follows:

Suitability Summary of FIR Filters for Implementation of Stationary Frame Controllers:

- Criterion 1 is partially satisfied. Implementation is possible on a 16-bit fixed point DSP, but only for the P+Resonant controller where the relationship to the continuous filter is not critical.
- Criterion 2 is not satisfied. The FIR method does not have a close relationship with its corresponding s-domain transfer function (i.e. continuous filter), and has a design process that may be difficult for discretization of the transfer function (i.e. a Laplace transform for the controller must be found, and it may not always be easily obtainable). However, if the continuous-discrete relationship requirement is relaxed, then many excellent design methods are available for various kinds of FIR filters with arbitrary specifications. This may be suitable for the P+Resonant filter in general, but is not viable for the StatRF systems.
- Criterion 3 is partially satisfied. For large numbers of selected controllers (i.e. more than 6) the FIR approach may become more efficient than the IIR method. However, the variable memory space requirements are significantly larger.
- Criterion 4 is satisfied. Non-recursive FIR filters are always stable [139].

6.2.3 Alternative Filtering Techniques

As neither of the conventional IIR and FIR digital implementation methods have been found to be completely satisfactory for the implementation of the stationary frame controller on a fixed-point processor, another solution must be found. One possibility is to use Multirate digital signal processing, which involves the down sampling of the input signal to decrease the sample rate

[139]. This would minimize the clustering effects of IIR filters, and also reduce the large number of Taps required for an FIR filter to track the response of a narrowband filter. However, there are a number of problems with this approach. Firstly, the consequence of down sampling is a loss of information, and the high accuracy required for the system (especially for the high frequency components of harmonic active filtering) cannot tolerate this loss. Secondly, selective harmonic compensation would require a different sample rate for each of the selected P+Resonant controllers. A separate FIR controller would therefore be required for each harmonic (which is not realistic), and for the IIR method the overhead would be greatly increased.

Another common digital filter variation is the Adaptive filter (of which the Kalman filter is one example) [139], where the filter coefficients (typically an FIR filter) are varied in real-time to suit the current signal characteristics. Whilst adaptive filters are good for matching to variations in system parameters, this research requires the digital filter to track a known continuous transfer function, which does not vary. Therefore, adaptive filters are not a suitable solution.

This review of the suitability of IIR and FIR filters has shown that an IIR filter solution is preferred, as it is more capable of emulating narrowband continuous filters and has a more straightforward design process. An IIR solution which is less sensitive to fixed-point processes would therefore be an ideal solution. So far the IIR filters considered have been implemented using the shift operator q – as is the case in most signal processing texts [139] - [142]. But this does not need to be the case.

The next option investigated here is the use of the delta operator to implement the IIR digital filters. In contrast to shift operator based IIR filters, it is shown that the response of the delta operator does converge to its continuous counterpart as the sample period approaches zero, and it is therefore much better suited to applications with sampling frequencies significantly higher than the filter system poles - as is the case in this research. Middleton et al. [138] has also shown that the delta operator has superior coefficient representation, rounding error performance and numerical properties in most calculations when compared to the shift operator. Therefore the focus of this chapter now turns to explore the application of the delta operator to the problem of practically implementing the desired stationary frame controllers on a fixed-point DSP. The remainder of this chapter provides an introduction to the delta operator methodology, the fixed-point controller implementation considerations, as well as application examples specific to the UCPC converter controllers presented in this thesis.

6.3 The Delta Operator

This section reviews the delta operator and considers its application to IIR digital filtering for the real-time control of power converters. The material presented is focused to provide a clear path for the conversion of conventional shift-based IIR filter design to the delta-based design approach. This allows a practical discussion for the implementation of delta IIR digital filters on fixed-point DSPs to be presented in the following section, and the final section develops and verifies the fixed-point implementation of the stationary frame controllers used in both the series and shunt controllers of the proposed UCPC, as well as a comparison with their shift-based counterparts.

The delta operator has previously been used for induction machine model parameter estimation [145] and for power system modelling [146]. However, despite the growing acceptance of the delta operators' superior high sample rate performance in the signal-processing field, little literature exists detailing the use of delta operators in real time digital control of PWM converter applications.

The delta operator δ was named and actively promoted for its use in digital control by Middleton and Goodwin in 1986 [138] [143] [147]. However, the technique was known in the numerical analysis field some decades before as the "divided difference operator", or "difference operator". A complete history of the origins of the delta operator by Goodwin et al. is contained in [147]. For discrete systems the delta operator is an Euler's approximation [148] to a derivative

$$\delta u[k] = \frac{u([k+1]\Delta) - u(k\Delta)}{\Delta} \approx \left.\frac{du(t)}{dt}\right|_{u=u(k\Delta)} \tag{6.15}$$

where Δ is the sample time.

Just as the continuous time derivative operator d/dt has an s-domain equivalent 's' (ignoring initial conditions) using Laplace transforms, the *Delta transform* can be used to convert the discrete time δ operator into its equivalent; 'γ'. Feuer and Goodwin [143] show that the Delta transform can be derived from the Laplace transform to illustrate the relationship between the two, and is summarized as follows.

If the Laplace transformation formula (single sided)

$$H(s) = L\{h(t)\} = \int_0^\infty e^{-st} h(t)\, dt \tag{6.16}$$

is discretized by the substitution of $k\Delta$ for time and an infinite summation for the integral, then

$$H'(s) = \sum_{k=0}^{\infty} \Delta e^{-sk\Delta} h(k\Delta). \tag{6.17}$$

Substitution of $e^{s\Delta} = 1 + \Delta\gamma$ results in the single sided Delta transform

$$H_\delta(\gamma) = D\{h[k]\} = \sum_{k=0}^{\infty} \Delta (1+\Delta\gamma)^{-k} h[k]. \tag{6.18}$$

Equation (6.18) can now be used to find the Delta transform of the derivative approximation δ (see proof by Feuer and Goodwin [143]), viz:

$$D\{\delta h[k]\} = \gamma H_\delta(\gamma) - (1+\Delta\gamma) h[0]. \tag{6.19}$$

Comparison of (6.19) to the Laplace transform of the derivative operator in

$$L\left\{\frac{d}{dt} h(t)\right\} = sH(s) - h(0) \tag{6.20}$$

shows that if $\Delta = 0$ then 'γ' becomes interchangeable with 's'.

Therefore, the delta operator has the particular property that as the sample time, Δ, approaches zero, the delta implementation converges towards its continuous counterpart, the Laplace transform variable s. In fact, it can be said that the continuous domain is actually just a subset of the discrete-time delta representation, where $\Delta = 0$ [143]. It is this property that gives the delta operator its superior performance at high sample rates compared to the shift operator (which does not converge).

Note that $e^{s\Delta}$ is a time shift of Δ, and is equivalent to z. Therefore, for the same substitution used to create (6.19), the shift and delta operators are found to be related by

$$z = 1 + \gamma\Delta. \tag{6.21}$$

The delta operator δ is associated with the γ-transform in exactly the same way that the shift operator q is associated with the z-transform, so it follows from (6.21) that

$$q = 1 + \delta\Delta. \tag{6.22}$$

This illustrates how the forward shift q is made up of the present sample plus the difference

(which is the derivative δ, multiplied by the time step Δ).

6.3.1 Delta operator based IIR Digital Filtering

This research requires the discrete implementation of continuous filters using delta operator based IIR digital filters. To achieve this implementation the required transfer function $H(s)$ must be defined with coefficients α'_x and β'_x (in this case second order)

$$H(s) = \frac{\beta'_0 s^2 + \beta'_1 s + \beta'_2}{s^2 + \alpha'_1 s + \alpha'_2}. \tag{6.23}$$

Equation (6.24) can then formed where the coefficients are equal to those in (6.23) for $\Delta = 0$, but they then diverge as Δ increases.

$$H_\delta(\gamma) = \frac{\beta_0\gamma^2 + \beta_1\gamma + \beta_2}{\gamma^2 + \alpha_1\gamma + \alpha_2} \tag{6.24}$$

A method of obtaining the α_x and β_x coefficients must therefore be found for finite sampling times. These coefficients could be found directly by applying the Delta transform defined in (6.18) to the sampled prototype system $h[n]$, but a more straight-forward approach is desirable.

Another possible solution is to use the relationship between the shift and delta domains in (6.21) to utilize the coefficients developed using the bilinear transformation. Note that it is the implementation method, and not the design method, that provides the delta IIR with its performance advantages over the shift IIR at high sample rates [143], and hence it is quite acceptable to use the same design process to determine coefficients. If the substitution

$$H_\delta(\gamma) = H_q(z)|_{z=1+\gamma\Delta} \tag{6.25}$$

is applied to (6.7), the conversions shown in Table 6.1 result. This allows for simple derivation of the delta filter coefficients using commonly known shift coefficient design techniques, to facilitate the usage of the proposed method by requiring minimal alterations to design methods familiar to power electronic researchers.

$\beta_0 = b_0$	$\alpha_0 = 1$
$\beta_1 = \frac{2b_0+b_1}{\Delta}$	$\alpha_1 = \frac{2+a_1}{\Delta}$
$\beta_2 = \frac{b_0+b_1+b_2}{\Delta^2}$	$\alpha_2 = \frac{1+a_1+a_2}{\Delta^2}$

Table 6.1: 2^{nd} order delta and shift coefficient conversions.

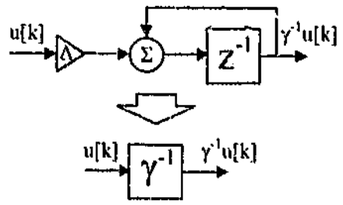


Figure 6.6: Delta operator construction diagram.

For the delta operator to be useful in digital control applications, a causal form must also be available. This is the inverse delta operator

$$\delta^{-1} = \frac{\Delta q^{-1}}{1 - q^{-1}} \qquad \text{or equivalently} \qquad \gamma^{-1} = \frac{\Delta z^{-1}}{1 - z^{-1}} \tag{6.26}$$

whose construction is illustrated in Figure 6.6. Using this operator, the IIR canonical form of the delta domain transfer function is found to be

$$H_\delta(\gamma) = \frac{\beta_0 + \beta_1\gamma^{-1} + \beta_2\gamma^{-2}}{1 + \alpha_1\gamma^{-1} + \alpha_2\gamma^{-2}}, \tag{6.27}$$

and has the same form as its shift counterpart in (6.7). Therefore, implementation of delta operator based IIR filters can be performed in the same manner as for shift operator based IIR filters, except with γ^{-1} replacing z^{-1}, and with different coefficients. Hence, the digital filter software structure of existing shift-based filters can remain mostly untouched when converting to delta-based filters, and the added computational overhead is also very minimal.

Figure 6.7 shows the discrete stability regions for the shift and delta implementations [143], as well as the region for the continuous Laplace domain. These plots also provide some illustration of the mapping that occurs between the continuous and discrete shift/delta planes. It can be seen that as Δ approaches zero, the stability region for the delta implementation will grow to approach that of the Laplace domain (i.e., the whole left hand plane), while the stability region of the shift implementation remains fixed, causing the clustering at 1 on the real axis of the pole-zero shown in Figure 6.2 as the sample rate increases.

6.3.2 Choice of Δ

For shift-based filters, the spread of the coefficient values are fixed, and if this spread is too large then the coefficients reach a point where they cannot be implemented on a 16-bit fixed-point

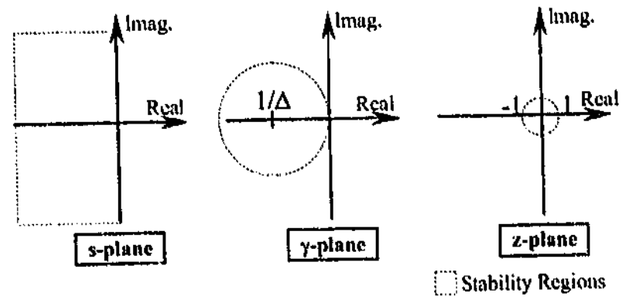


Figure 6.7: Stability regions for the continuous Laplace plane, discrete delta, and discrete shift planes.

system. However, for delta operator based digital filters (despite the variable Δ initially being defined as the sample period) Δ is free to be varied to optimize the numerical properties of the design [147]. Therefore, since the coefficients are a function of Δ (Table 1), the spread of the coefficients can be optimized to allow lower percentage rounding errors. In general, the choice of Δ determines the coefficient rounding and sensitivity, as well as the maximum internal variable size of the delta filter, as discussed later.

6.3.3 Direct Form Structures

IIR digital filters can be implemented using either direct form I (DFI) or direct form II (DFII) structures, or their transposed versions DFIt and DFIIt (Figure 6.8). The DFI form can be seen to be simply a diagrammatic version of (6.6), illustrating the order in which the additions and subtractions should take place. The DFIt, DFII and DFIIt forms are rearrangements developed to achieve different numerical responses.

Whilst most shift-based digital filters can be implemented with any of these forms, this is not necessarily the case for fixed-point delta-based digital filters [149]. For the DFI form (Figure 6.8a) the unstable pole at $z = 1$ of the γ^{-1} formula in (6.26) is not cancelled prior to this inverse delta operation, and this implementation is therefore unstable. This causes the output of the operation to overflow. For the DFIt form (Figure 6.8b) the unstable poles of (6.26) are cancelled prior to the inverse, but require double precision to function properly [149]. These problems are mostly avoided by use of the DFII and DFIIt forms (Figures 6.8c and 6.8d). Since they also have half the number of γ^{-1} operations compared to the DFI and DFIt forms, only the DFII and DFIIt forms are considered further for fixed-point delta-based filter implementations.

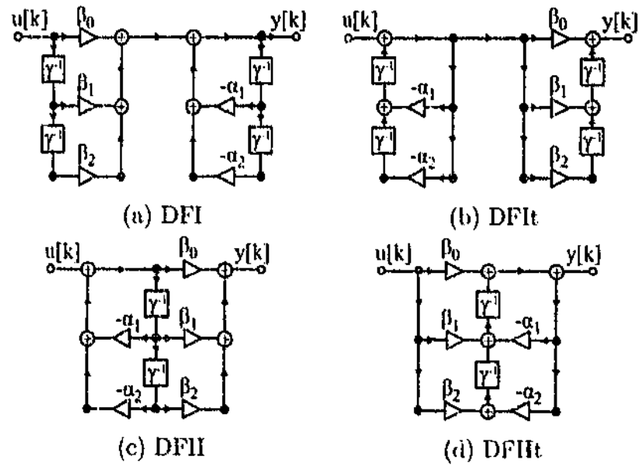


Figure 6.8: Direct Form (DF) digital filter implementation structures: (a) DFI; (b) DFIt; (c) DFII; (d) DFIIt.

6.4 Fixed-point Implementation

This section explores the fixed-point issues to be considered during the implementation of delta-based digital filters, and provides practical approaches to aid in development of a reliable and improved performance implementation. For this research work a TI TMS320F240 16-bit fixed-point DSP has been used for the LV experimental verification, implementing a DFII type delta-based digital filter using the software interrupt procedure shown in Figure 6.9. Note that this DSP does provide a 32-bit accumulator, which can be utilized to minimize overflow and truncation problems in the IIR filters.

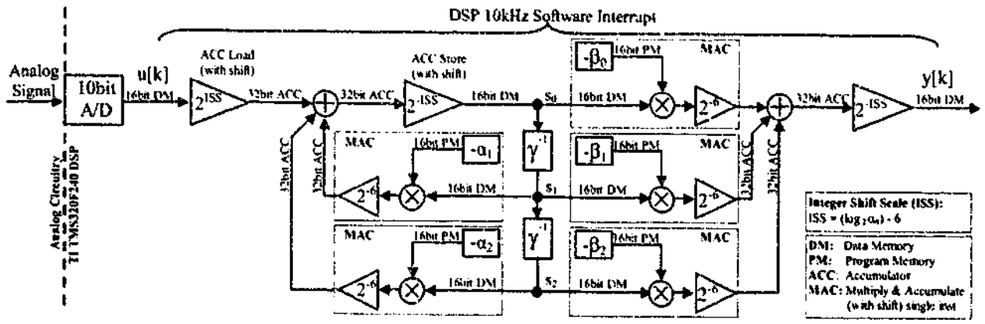


Figure 6.9: 16-bit fixed-point TI TMS320F240 DSP delta-based IIR software implementation diagram.

6.4.1 Overflow Minimization and Clamping Techniques

To manage overflows, the TI DSP instruction set allows the output of the 16-bit x 16-bit multiplications to be automatically shifted down by 6 bits before being added to the accumulator. This allows up to 128 consecutive multiplications and accumulations before an overflow can occur. The multiply, shift and accumulate steps are all conducted together using a Multiply and Accumulate (MAC) instruction in a single cycle of the DSP (Figure 6.9). A similar instruction is also available on the equivalent Analog Devices and Motorola product families, although for these products the shift is not required since they use larger length accumulators to reduce the overflow occurrences.

Clamping must be applied to points S_0, S_1 and S_2 of the filter implementation (see Figure 6.9) to avoid overflows when the accumulated results are stored back into the 16-bit data memory [149].

6.4.2 Internal Truncation Compensation

While 32-bit accumulations minimize the truncation effect that would occur if the resultants were added in 16-bit, ultimately the result still has to be converted back to 16-bit, and this leads to various problems for IIR digital filters. A shift-based implementation is particularly sensitive to this truncation, in contrast to the minimal effect it has on a delta-based implementation (as shown in the later examples).

Some of the truncation problems can be reduced by converting the truncation (floor) into a rounding event. This is done by adding an offset of half the magnitude of the divisor to the signal preceding the downward bit shift of the multiplication shift. For example, Figure 6.10 shows the truncation compensation applied to the inverse delta operator. If Δ is chosen to be 1/32, then this corresponds to a right bit shift of 5 (i.e. $log_2 32$) to the input signal. Therefore, an offset of magnitude 16 would, in isolation, shift and truncate down to 0, but in combination with the incoming signal, it converts the original truncation of the incoming signal into a rounding operation.

6.4.3 Coefficient Rounding

Another problem faced by fixed-point systems is coefficient rounding, which occurs during the filter design stages. The percentage of rounding in the coefficients is dependant on the scaling of the rational coefficients, so that they can be represented in integer form. The scaling is achieved by replacing the 1 in the IIR filter canonical form denominator of the shift and delta filters.

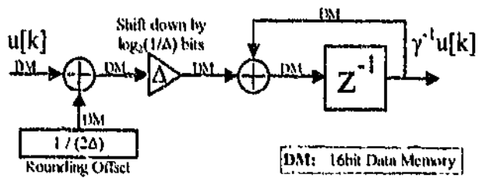


Figure 6.10: Delta operator fixed-point DSP software implementation with truncation compensation.

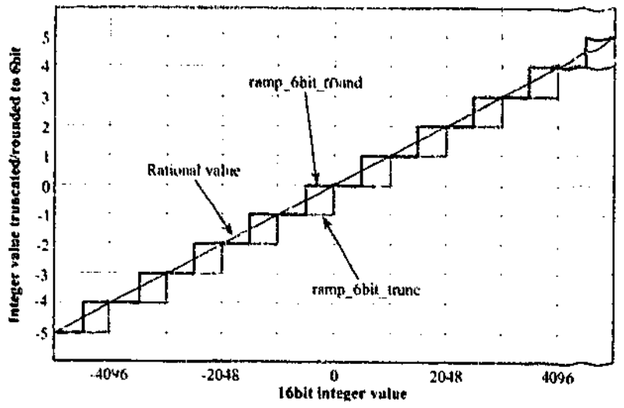


Figure 6.11: Truncation and rounding effects from a 16-bit fixed-point DSP.

with the coefficients a_0 and α_0, respectively. This will scale all the other coefficients by α_0, so that if α_0 is chosen as 214, then a coefficient of magnitude 1.546, would become 25329.664, and round to an integer of 25330. As discussed earlier, for higher frequencies the roots of shift-based IIR filters cluster toward 1 on the z-plane, and therefore the effect of this minor loss of information becomes more severe as the sampling frequency increases. For faster sampling rates, this sensitivity for the shift operator is proportional to $1/T_n$ (where n is the order of the filter), while it is a relatively small constant value for delta-based implementations [143].

Example 1 (Sub-section 6.5.1) confirms how this rounding affects the shift implementations, while the delta operator for the same example remains quite resilient to such rounding, even with 6-bit coefficients.

6.4.4 Integer Scaling and Δ Selection

For fixed-point shift filters the only parameter that can be varied is the scaling factor a_0, which is typically set to the maximum value that allows all the coefficients to fit within the available integer bit size. This choice minimizes the percentage effect of the coefficient rounding and also decreases the internal truncation problems, although the scaling factor may sometimes need adjustment and lowering if internal variables become too large.

For delta filters, the design becomes more complicated as both the scaling factor α_0 and the value of Δ need to be selected, and the ideal values of each do not necessarily agree. Table 6.1 shows that α_1 and β_1 are inversely proportional to Δ, α_2 and β_2 are inversely proportional to Δ^2, while β_0 is independent of Δ. Therefore, as Δ decreases the coefficients will spread, and care must be taken that the maximum spread remains significantly smaller than the allowed coefficient size (i.e. $\pm 2^{15}$ for a 16-bit signed integer). Also, if Δ becomes too small, the quantization noise introduced in the γ^{-1} operation will become significant, and other quantization effects also change as Δ is varied. A detailed quantization noise optimization method for choosing Δ by Kauraniemi et al. can be found in [149]. That said, decreasing the size of Δ was found to decrease the internal variables for both of the following practical examples, and therefore a compromise had to be reached to obtain an acceptable filter input range.

Another consideration for the design is that both Δ and α_0 (or a_0) should ideally be a power of 2. This facilitates their implementation as simple right bit shifts, instead of using costly division operations. Figure 6.9 shows that Δ is implemented as a right shift of $log_2(1/\Delta)$ bits. For α_0 this shift is denoted in Figure 6.10 as the Integer Shift Scale (ISS), which also incorporates the 6-bit shift used for the overflow protection discussed earlier.

It should also be noted that as the width of the notch and resonant filters in the following examples was decreased (i.e. ω_c was decreased) it was found that the internal variables of the DFII form tended to increase around the notch/resonant frequency. Thus, the choice of ω_c is also significant in the design of the scaling and Δ coefficients.

6.5 Delta IIR versus Shift IIR

This section develops two filter examples to compare shift- and delta-based IIR implementations. The examples are the stationary frame reference generation used for the shunt portion of the UCPC (proposed in Chapter 5), and the P+Resonant controller used by the series portion of the UCPC. These controllers are illustrated in Figure 6.12.

The objective of the stationary reference frame controller (Example 1) used in the shunt converter controller is to remove the positive sequence fundamental from the load current measurement, leaving only the harmonic and unbalanced currents (for three-phase systems). The negative of this remaining signal is then injected back into the grid to cancel out these unwanted quantities. This stationary frame controller has a second order notch based transfer function.

Conversely, the series converter P+Resonant controllers (Example 2) are feed-back based, and aim to provide a high gain at selected frequencies to minimize steady-state load voltage error. The transfer function of this controller is a high Q resonant function. Example 2 first applies the P+Resonant controller to a 50 Hz current regulator to compare the shift- and delta-based implementations, and then presents the harmonic voltage controllers of the UCPC for only the delta approach (as this controller could not be implemented using the shift method).

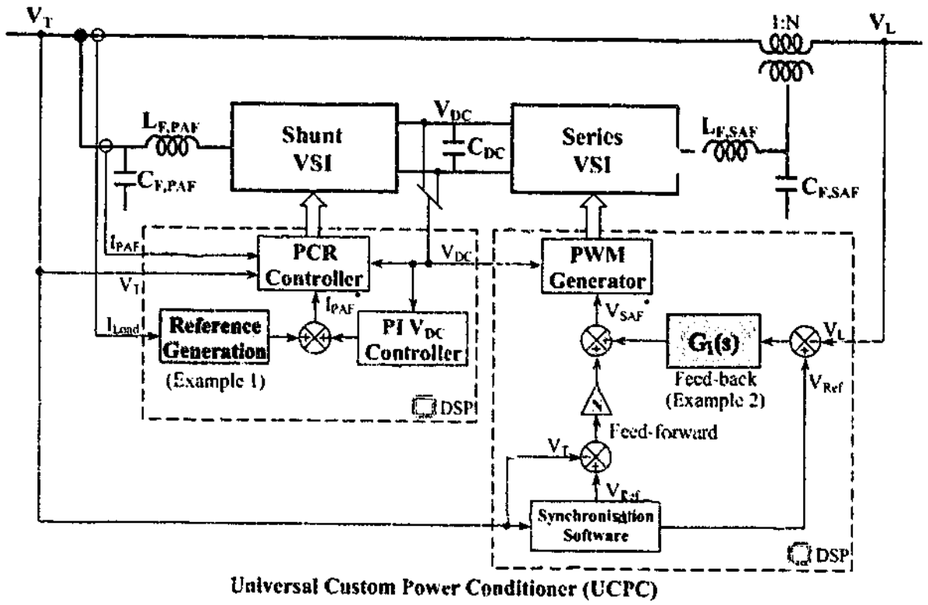


Figure 6.12: Proposed control scheme for the Universal Custom Power Conditioner, with controllers used for shift/delta comparison examples shown in grey.

6.5.1 Example 1: StatRF Harmonic Reference Generator

The first example is the harmonic reference generator block used in the shunt portion of the UCPC (Figure 6.12). Chapter 5 showed that for three-phase (and three-wire) systems both the load current harmonics and negative sequence fundamental current must be extracted, and this can be achieved using the cross-coupled notch based filters defined by

$$[\mathbf{H}_{ab}(\mathbf{s})] = \begin{bmatrix} \frac{s^2+\omega_c s+\frac{\omega_0\omega_c}{\sqrt{3}}+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} & \frac{\frac{2\omega_0\omega_c}{\sqrt{3}}}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} \\ \frac{-\frac{2\omega_0\omega_c}{\sqrt{3}}}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} & \frac{s^2+\omega_c s-\frac{\omega_0\omega_c}{\sqrt{3}}+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} \end{bmatrix}. \tag{6.28}$$

This filter arrangement was shown in Chapter 5 to be an exact theoretical equivalent to a synchronous d-q frame high-pass filter method (Figure 6.13). Note that the fixed-point shift-based IIR implementation problems also apply to the high-pass filter used in the d-q implementation, but since the d-q version uses only a first order filter, the effects are less severe. Also the filter frequency is set by the Park Transformation, and therefore filter problems cannot cause a skew in the selected frequency (unlike the stationary frame controller). However, the first order filters used can become unstable due to the combination of the rounding and clustering effects discussed earlier in this chapter. The d-q implementation also has other practical problems which are described in Chapter 9, which make the StatRF option in (6.28) more desirable once delta-based IIR filters are used.

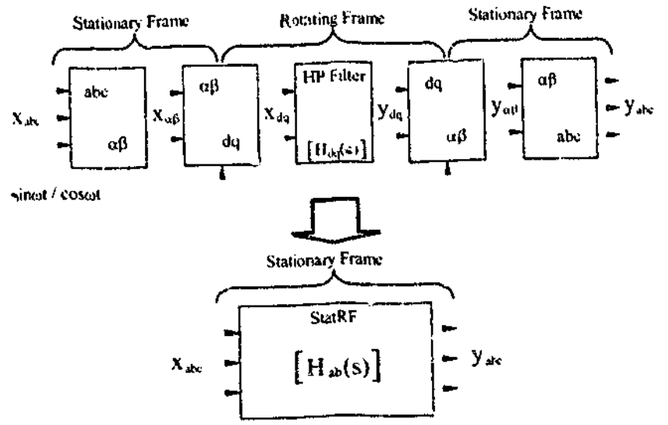


Figure 6.13: Rotating to Stationary Frame conversion of the High Pass (HP) type Synchronous Reference Frame (SRF) harmonic extraction unit.

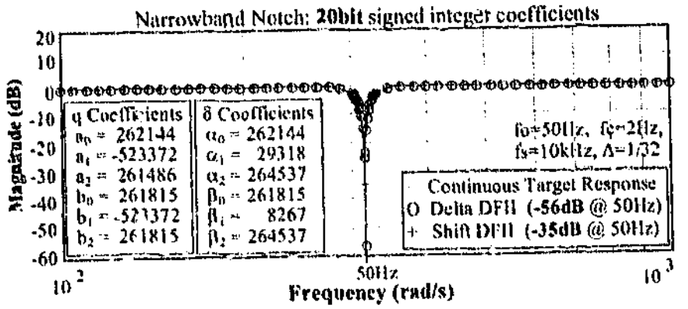
For single-phase systems only a harmonic reference is required (since single-phase systems do not have negative sequence components) and therefore the extraction takes the form of a narrowband notch (see Chapter 5) defined by

$$H(s) = \frac{s^2 + \omega_o^2}{s^2 + 2\omega_c s + \omega_0^2}. \tag{6.29}$$

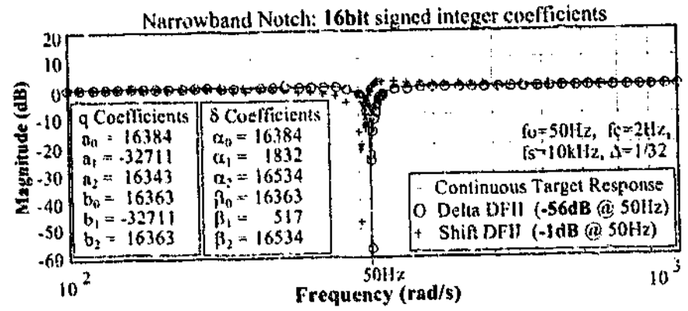
From an implementation perspective, the filters in both (6.28) and (6.29) have very similar characteristics. For this reason, only (6.29) is used in the following comparison, so that the presentation of the filter coefficients can also be included. However, both versions are implemented in the LV experimental work described in Chapter 9.

Figure 6.14 compares the frequency responses of shift- and delta-based filter implementations for a narrowband notch (6.29). The shift-based filter only achieves 35-dB attenuation of the 50 Hz signal even using a 20-bit implementation, and with a 16-bit implementation virtually none of the 50 Hz signal is attenuated because of the shifted notch frequency. On the other hand, the delta-based filter achieves 56-dB attenuation with a 16-bit implementation, and surprisingly still tracks the continuous filter locus for both 12-bit and 6-bit implementations (where the shift-based filter becomes useless). The frequency plots were generated using Matlab and Simulink simulations which modelled all quantization, coefficient rounding and internal truncations of the actual DSP practical implementation. This simulation method is discussed in detail in Chapter 8. Also, the coefficients used for the delta-based filter examples were generated from the shift coefficients (before rounding) using Table 6.1, which confirms that the limitations of the shift-based filter are not attributable to the Tustin (i.e. Bilinear) transformation.

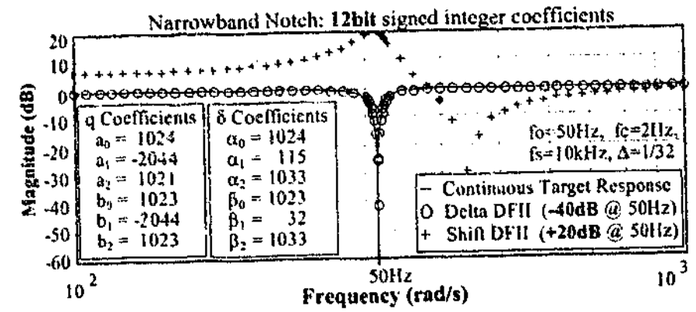
This example shows one situation where a shift-based IIR implementation is not possible on a 16-bit fixed-point DSP. By increasing the value of f_C the shift-based implementation will become more effective, but as f_C increases it affects the phase of the extracted harmonics and therefore impacts on the performance of the active harmonic filtering process. As the implementation overheads of the shift method versus the delta method are very minimal, there is no reason for such compromises to be made.



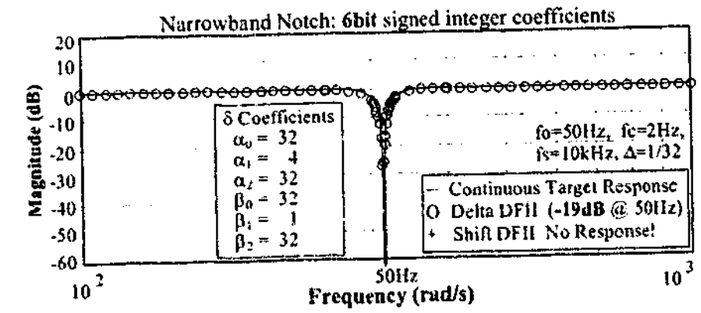
(a) 20 bit implementation



(b) 16-bit implementation



(c) 12-bit implemetation



(d) 6-bit implementation

Figure 6.14: Frequency response for continuous, shift and delta filters with: (a) 20-bit; (b) 16-bit; (c) 12-bit; (d) 6-bit coefficients.

6.5.2 Example 2: P+Resonant Controller

The second example is the P+Resonant controller (6.30) used for the selective harmonic compensation in the series controller of the UCPC (Figure 6.12). Once more both shift- and delta-based IIR methods are compared, but the comparison firstly looks at a single P+Resonant controller

$$H_{P+R}(s) = K_P + 2K_I\omega_c\frac{s+\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} \tag{6.30}$$

at the fundamental frequency (i.e. not considering the selected harmonic controllers), as some of the higher frequency selected harmonics simply cannot be implemented on the 16-bit DSP using a shift-based method.

Figure 6.15 shows the magnitude and phase plots for the shift and delta-based 16-bit fixed-point implementations of (6.30), where it can be seen that the delta-based filter exactly tracks the desired continuous function, whilst the notch of the shift-based filter skews away from the target frequency. This causes a 50 Hz gain of only 4-dB, instead of the ideal 12-dB, and the phase of the shift-based filter also reduces (causing a loss in the phase margin of the system near the notch frequencies of up to 30°).

The P+Resonant controllers shown in Figure 6.15 were implemented on a 16-bit experimental system to regulate a fundamental current into an R-L load. The results show that for this example the shift-based version contains a significant steady-state error (Figure 6.16), while the delta-based implementation creates very little error (Figure 6.17). The integral gain K_I was chosen to just give a visual steady state error of zero, so that the effect of the shift and delta implementations could be easily seen in the comparison of the steady state errors.

Furthermore, the shift-based implementation required truncation compensation (Sub-section 6.4.2) for it to even be possible, since the truncations caused by the transfer of the 32-bit accumulator summation to the 16-bit data memory were found to cause very large dc offsets which quickly saturated the PWM modulator. The delta-operator version, on the other hand, was readily implemented without any truncation compensation. However, compensation was added for the final version before the Δ bit shift (Figure 6.10) to remove a small dc offset found at low demanded current levels. This is one of many examples where the sensitivity of shift-based filters was found to cause problems requiring attention, whereas delta-based implementations did not suffer from nearly the same severity (if at all).

For the selective harmonic compensation used in the series controller of the UCPC the

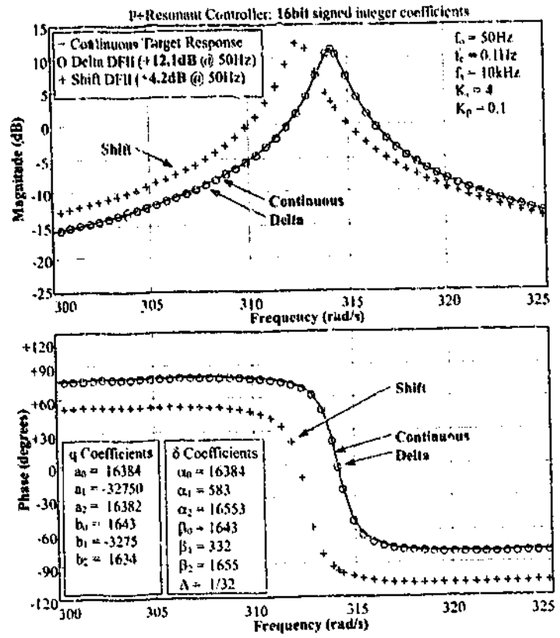


Figure 6.15: Open loop frequency response for continuous, shift and delta-based, 16-bit P+Resonant filters.

following function is required:,

$$G_1(s) = K_P + \sum_{n\in M} 2K_{I,n}\omega_c\frac{s+\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_n^2} \text{ where } M = \{1,5,7,9,11,13\}. \tag{6.31}$$

Figure 6.18 shows the continuous and delta-based IIR implementations of (6.31) using the parameters from the experimental work in Chapter 9. The delta IIR parameters are given in Table 6.2. All the resonant filters were designed with an integral gain K_I of 5, with post filter gains provided where necessary (i.e. to boost the gain to 10 for the fundamental, 5^{th} and 7^{th} harmonics). This figure shows that the delta-based method closely follows the target continuous filters. Note that the shift-based method is not shown, since the 16-bit fixed-point implementation was not practical with an f_C of 1.5 Hz and sample frequency f_S of 10 kHz, due to significant internal variable overflow for all the selected harmonics (for both DFI and DFII types) even for small input signal magnitudes. However, a clear divergence between the continuous and delta-based filters can be seen as the Nyquist frequency is approached, which is an unavoidable consequence of the digitization process. An increasing divergence in the phase

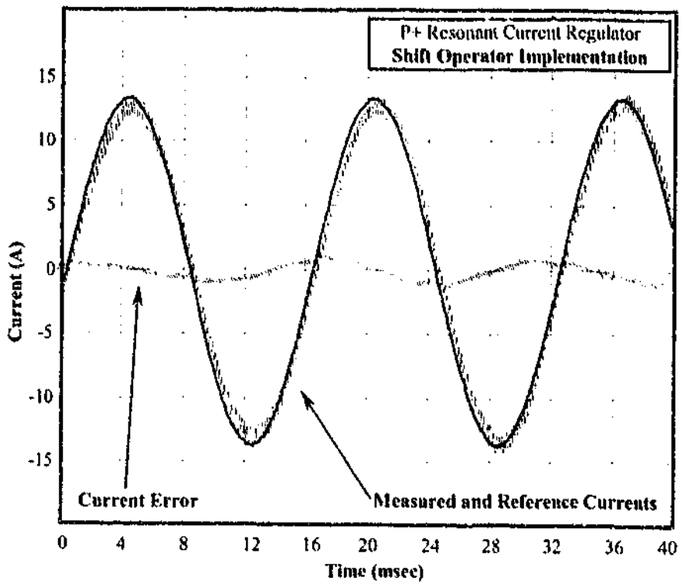


Figure 6.16: Experimental results for a shift-based P+Resonant current regulator.

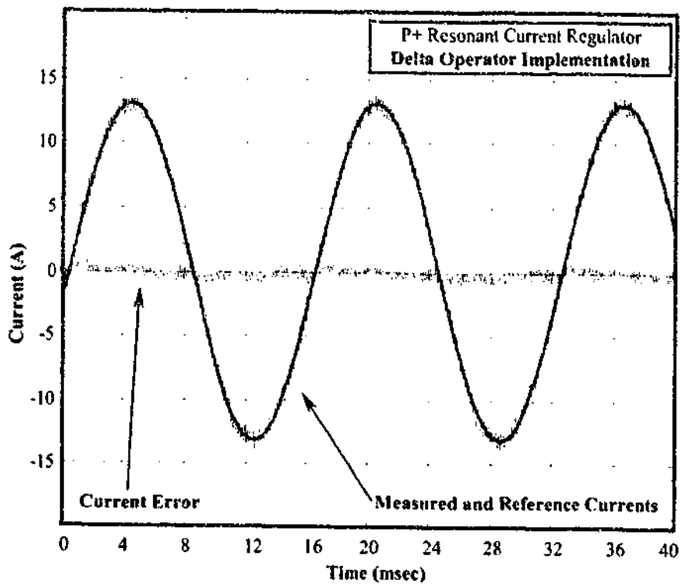


Figure 6.17: Experimental results for a delta-based P+Resonant current regulator.

Selected Frequency	n	K_I, n	f_C	α_0	Δ
50 Hz (Fundamental)	1	10	1.5 Hz	2^{13}	1/32 (2^{-5})
250 Hz (5^{th} Harmonic)	5	10	1.5 Hz	2^{13}	1/8 (2^{-3})
350 Hz (7^{th} Harmonic)	7	10	1.5 Hz	2^{13}	1/8 (2^{-3})
450 Hz (9^{th} Harmonic)	9	5	1.5 Hz	2^{13}	1/4 (2^{-2})
550 Hz (11^{th} Harmonic)	11	5	1.5 Hz	2^{13}	1/4 (2^{-2})
650 Hz (13^{th} Harmonic)	13	5	1.5 Hz	2^{13}	1/4 (2^{-2})

Table 6.2: Parameters of the delta-based IIR digital filters used for the UCPC series feed-back compensation.

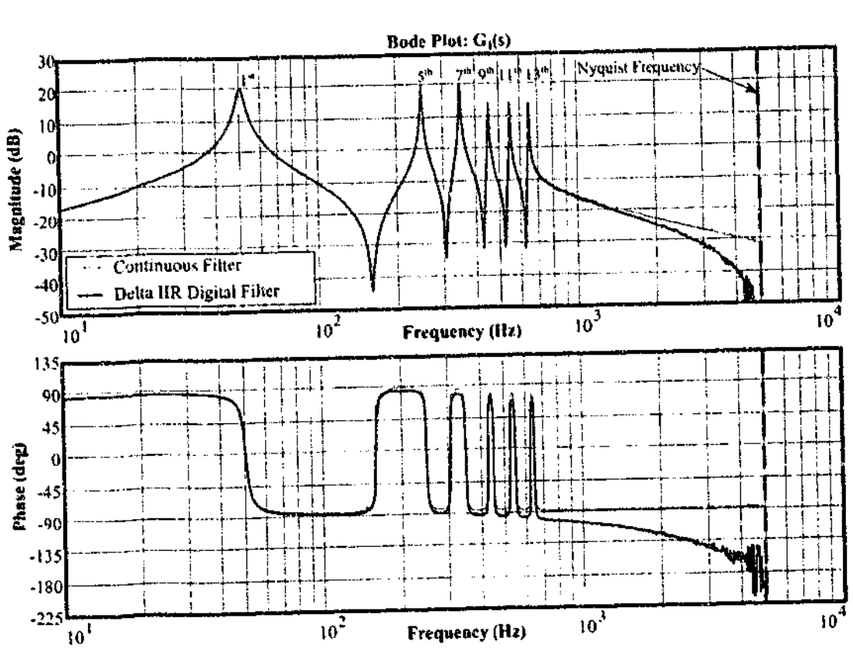


Figure 6.18: Simulated open loop frequency response for ideal continuous and delta-based 16-bit parallel selective P+Resonant controllers (with parameters from the series controller of the UCPC).

can also be seen, and also is due to the digital sampling process.

Previous selective harmonic compensation systems which used the P+Resonant controller have used either floating-point DSPs with a shift-based IIR method [110] [150] (both shunt active filters), or have been forced to use approximations to the P+Resonant controller using an FIR filter to implement a selective harmonic controller on a 16-bit fixed-point DSP [17]. Using the delta operator method, Figure 6.18 shows that fixed-point implementation is possible, and this is verified experimentally in Chapter 9.

It should be noted that this chapter only investigates the use of the delta operator for a particular set of digital controllers, to suit a particular problem. The application performance of the delta IIR digital filter to other linear controllers is not within the scope of this research, and is left for further work. Furthermore, Kauraniemi et al. [149] indicate that for some particular filter arrangements the shift-based approach may actually provide better sensitivity performance than the delta method. However, this is generally for situations where the sample frequency is significantly lower than is typically used for modern PWM converter applications.

6.6 Summary

This chapter has investigated the problems of digital implementation of the stationary frame linear controllers described in Chapter 5. Conventional shift-based IIR and FIR digital filters were reviewed to identify why each of these methods did not meet the desired criteria. In some cases, while the shift-based IIR could not implement the desired controllers on a fixed-point 16-bit DSP, the FIR was able to implement these controllers. However, due to leakage and other effects the response of the FIR filter was significantly different to that of the desired continuous filter. The FIR filter was therefore considered to be unsuitable for systems such as the StatRF harmonic extraction controller proposed in Chapter 5, because of the need for correct cancellation of the cross-coupling terms for this controller.

Delta operator based IIR digital filters were then proposed to solve these implementation problems. The delta operator δ allows high performance digital filters to be implemented with 16-bit (or less) fixed-point arithmetic without significant error, and is particularly effective for applications involving high sampling frequencies and high Q narrowband filters – such as the stationary frame controllers used in this research. While the delta operator is well known in the signal processing field, there is little literature discussing its application for real time digital control of PWM converters. A review of the delta operator and its application to digital filtering then led to a method of applying it to the desired stationary frame controllers to achieve significantly improved correlation between the continuous and discrete filters – especially using fixed-point arithmetic. Table 6.3 summarizes the differences between the shift IIR, FIR, and delta IIR implementations of the controllers used in this research.

	Shift IIR	**Non-recursive FIR**	**Delta IIR**
Computational cost	Excellent	Average	Excellent
Variable space	Excellent	Poor	Excellent
Coefficient sensitivity	Poor	Excellent	Excellent
Calculation truncation sensitivity	Poor	Excellent	Very Good
Continuous filter comparison	Average	Poor	Excellent
Ease of design from continuous filter	Excellent	Poor	Excellent
Overflow clamping and design minimsation	Good	Excellent	Very Good

Table 6.3: Comparison of digital filtering techniques for 16bit fixed-point implementations of narrowband type controllers.

Chapter 7

Control of UCPC Protection System

[1]Chapters 4 and 5 have investigated the feed-back/feed-forward control systems of the proposed UCPC, and Chapter 6 has investigated the digital control issues required to implement these controllers. This chapter continues with this theme, and investigates the protection systems required for the UCPC, to complete the knowledge base required to build the system experimentally as described in Chapters 9 and 10.

The protection systems required for shunt converters are straightforward and are common knowledge in both industry and academia. However, relatively little appears to have been reported regarding the protection of series converter systems [101], although the problem's existence (even under stand-by conditions) has been raised by multiple researchers over the last decade [39] [49] [98] [99]. This chapter starts with a short review of the protection requirements for shunt converter systems, and then considers in more detail the protection requirements for the series converter section of the UCPC.

The protection requirements for series injection converters are quite different to those of shunt protection systems, and in most cases using a shunt converter protection scheme for a series topology is likely to cause failure – even under normal operating conditions. This work proposes a series protection scheme which integrates series converter protection hardware with hardware and software controllers, to achieve protection of the system under fault conditions, start-up, stand-by, shut-down, and recovery from faults. The proposed scheme has been verified in both simulation and experimentally under fault conditions, and is used to protect the experimental UCPC system presented in Chapter 9.

[1]The material in this chapter was first published in part as:
M. J. Newman and D. G. Holmes, "An Integrated Approach for the Protection of Series Injection Inverters," *in Conf. Rec. IEEE/IAS Annual Meeting*, Chicago, IL, USA, pp. 781-788, 2001. and
M. J. Newman and D. G. Holmes, "An Integrated Approach for the Protection of Series Injection Inverters," *IEEE Trans. on Ind. Applicat.*, vol. 38, no. 3, May/June, pp. 679-687, 2002.

7.1 Shunt converter Protection

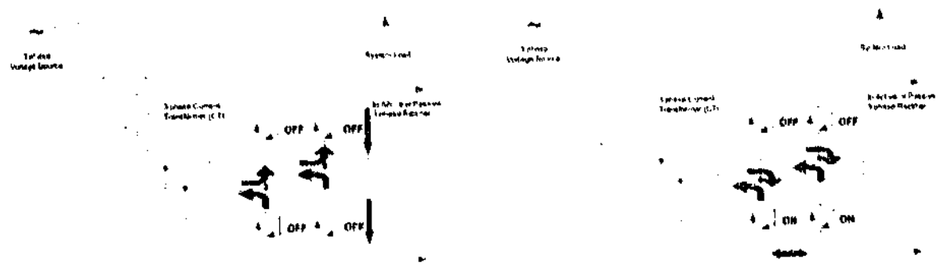
For a conventional shunt converter, the most common protection provisions are for over-current, over-temperature, and over-voltage, switching device failure, and control system failure. The usual response when a fault is detected is to switch the converter off (perhaps with a controlled turn off for an over-current trip to avoid an over-voltage surge), and to let the load freewheel down to the OFF state. Most commercial drive converters implement this type of protection in one form or another.

For the experimental shunt converter described in Chapter 9, over-current, dc-bus over-voltage, and IGBT de-saturation protection are incorporated, as well as provision for over-temperature protection. Over-currents are detected using the same LEMs as are used for the PCR current regulation system (Chapter 4), and the dc-bus over-voltage protection uses the dc-bus measurement input in the same way. To detect and limit shoot-through of the IGBT phase-legs, the HCPL-316J gate driver integrated circuit (IC) detects deep saturation of the IGBT, which is characteristic of this condition. The IGBT is then switched off by the gate driver IC, and cannot be over-ridden by the DSP. All these external fault detection mechanisms pass into the Power Drive Protection Interrupt (PDPINT) of the Texas Instruments DSP (TMS320F240), which in turn sends a signal to all the converter IGBTs to turn off. This operation is implemented in the hardware of the DSP (not software), and can only be over-ridden by disabling the overall feature on the DSP. (The consequence of this default protection mechanism for series converters will be made clear in the following section, as the difference between series protection requirements and shunt protection requirements are developed.)

7.2 Series converter Protection

7.2.1 Introduction to Protection Issues for Series Injection Converters

Unlike the shunt converter, simple disconnection does not work for the series converter. This is because the primary complication for protection of a series converter is that current always flows in the primary side of the injection transformer as long as the system load or fault current exists. Unless this current is magnetomotive force (mmf) balanced with a secondary current, it will over-magnetize the series injection transformer and generate substantial secondary side voltages. Even if the VSI is disabled, it still acts as a diode rectifier (Figure 7.1a), and hence the dc-bus voltage will quickly charge up beyond the voltage rating of the power stage. This can happen in well under a millisecond for fault level over-currents, and can occur within one



(a) OFF state (b) NULL state

Figure 7.1: VSI current paths for a series injection topology.

fundamental cycle even under normal load current conditions. Hence, an effective protection scheme must at all times provide a continuous secondary current path, and divert this current to appropriately rated elements when the converter is not switching, depending on the level and duration of current to be passed. It is also important that the protection system does not disrupt the external supply to the load during a fault. Many power equipment protection systems rely on the detection of substantial over-currents or voltages to shut down the plant, and it is important that the series injection converter system does not interfere with the proper operation of such systems.

Figure 7.2 shows the simulated response of a 250 kVA converter system to a light fault (1 kA), without a series protection system operating. The dc voltage clearly becomes unacceptably high (> 2 kV) within a few milliseconds of the fault occurrence, as the switched off IGBTs act as rectifying diodes for the secondary referred fault current.

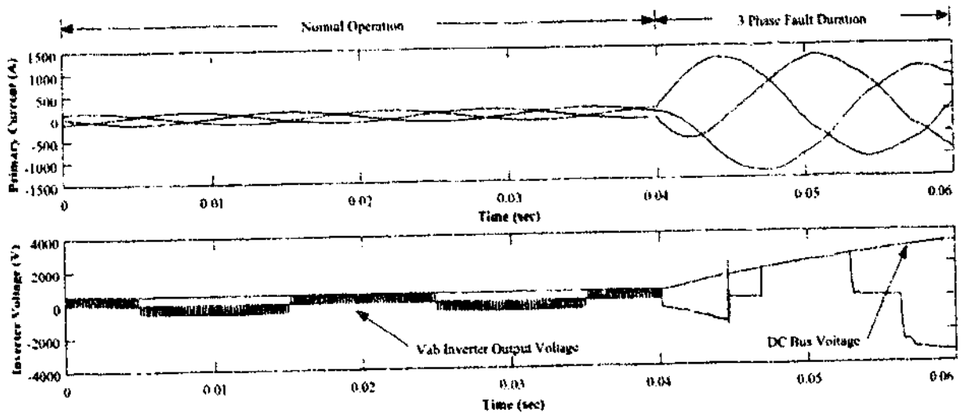


Figure 7.2: Fault response of a 250 kVA series inverter to an inverter trip with no series protection system.

7.2.2 Previous Literature

The existence of the series protection problem has been noted by a few authors over the last decade [39] [49] [98] [99], but none of this literature discusses schemes to overcome the problem. (It is commented that most reported systems only use small prototypes which can avoid most of the problems faced.) Bhattacharya et al. [98] noted that a *"sophisticated protection and sequencing scheme"* is required for series protection, and that a state based sequencing scheme is required to coordinate *"start-up, shut-down, fault coordination and bypass of the series active filter under appropriate conditions"*. However, no actual details of any of the protection or coordination schemes used were reported.

In 1996 Moran et al. [100] [101] presented the first of only a few dedicated publications discussing a fault protection scheme for series converters, and this work will now be considered in more detail. However, it is noted at the outset that while Moran et al. has presented all the components required to protect a series system, this thesis disagrees with the approach taken to control and rate these devices.

Moran et al. [101] addressed the problem of series converter protection by using a combination of varistors and thyristors to switch a shunt resistance across the injection transformer secondary when an over-voltage is detected. The fault energy is shared between the varistors and the large shunt resistance, and therefore requires these protection elements to be rated for significant transient power dissipation. This is a particular issue for the case where the primary side grid protection system fails, and the fault is only cleared by back-up protection several seconds later. Moran et al.'s system also requires a specially designed series injection transformer which saturates with only moderate magnetizing current overload to reduce the over-voltage generated by a primary side fault current. Such complications should be avoided (if possible) for a practical commercial system.

The problems found with this scheme are as follows:

1. A transformer with a very low saturation voltage (i.e. it saturates even under rated conditions) is required, whereas most series Custom Power applications require the opposite [151] (i.e. a high transformer saturation voltage) to ensure high accuracy for good cancellation of Power Quality problems in the voltage/current.

2. Even if a transformer with a very low saturation voltage is used, the voltage at which the system will operate still needs to be approximately twice the rated voltage, due to the clamping characteristic of the varistor. Therefore, all the system devices need to be rated for twice the rated voltage, which is not attractive.

3. The scheme requires the majority of the fault energy to be passed into the series system (via the varistors and inserted series resistance), which may in turn cause other problems for the application. This situation is what a protection scheme is meant to avoid, not encourage!

4. The system significantly changes the characteristics of the fault as seen by the external grid protection systems, and therefore these protection mechanisms may not operate as designed unless specifically adjusted to account for the installation [152] [153] (assuming this is possible).

5. The use of, and reliance on, varistors for such sustained high energy operations is not seen as a practical, or reliable, solution by industry [66].

6. Post fault recovery is not considered, and the ability of the system to transfer the current path to a safe steady-state condition is unclear. Furthermore, only a unidirectional control link from the thyristor controller to the primary DSP controller is shown, and therefore if a switching device fails, the DSP cannot control the thyristors to protect the system from failure.

Finally, an issue that appears not to have been considered in the literature is how to start-up a series injection system when a load current is present. Since current continuity on the secondary side must be maintained at all times, it is essential to coordinate the operation of any protection elements and the injection converter as current transfers from bypass circuitry to the converter during start-up.

7.2.3 Overview of the proposed Integrated Series Protection Scheme

This work proposes an integrated control/protection scheme for a series injection converter, which manages internal converter faults, external system faults, and start-up issues in a single coordinated approach managed by the converter controller. The scheme does not require a special transformer design, and substantially reduces the power ratings of the protection components compared to previous approaches.

Figure 7.3 shows the basic topology of a series injection converter system, including the major components of the proposed integrated protection system. As discussed, the primary role of a series protection scheme is to provide an appropriately rated current path under all feasible

conditions. The three stages of current path provided are: the IGBTs themselves using NULL state (Figure 7.1b); the thyristors/triacs; and the by-pass contactor. Varistors are only used to clamp the first transient edge of an over-voltage, and to detect and trigger the thyristors to clamp the voltage. Note that the small shunt resistance that is in series with the thyristors offers little (if any) fault current limiting (since the mmf balance in the CT is maintained during faults due to the significantly lower voltages), with its only purpose being to limit any capacitive transient for topologies using LC filters. This approach is opposite to the arrangement proposed Moran et al. [101].

A key aspect to ensuring a continuous current path is the integration of the protection hardware controllers with the primary DSP controller, via an effective communications link. Furthermore, the system should be capable of protecting itself, even with the failure of either the DSP controller or the external protection controller. This is achieved by allowing either the DSP or the protection controller to trigger the clamping thyristors. Then, by allowing only the DSP to reset the thyristor clamping, transfer to normal operation can easily be achieved as the DSP can set the VSI to the NULL state prior to this reset. The communication of all faults is also passed to the DSP controller to coordinate the transfer of the fault current, and to allow it to take appropriate action in the case of failure of the thyristors themselves.

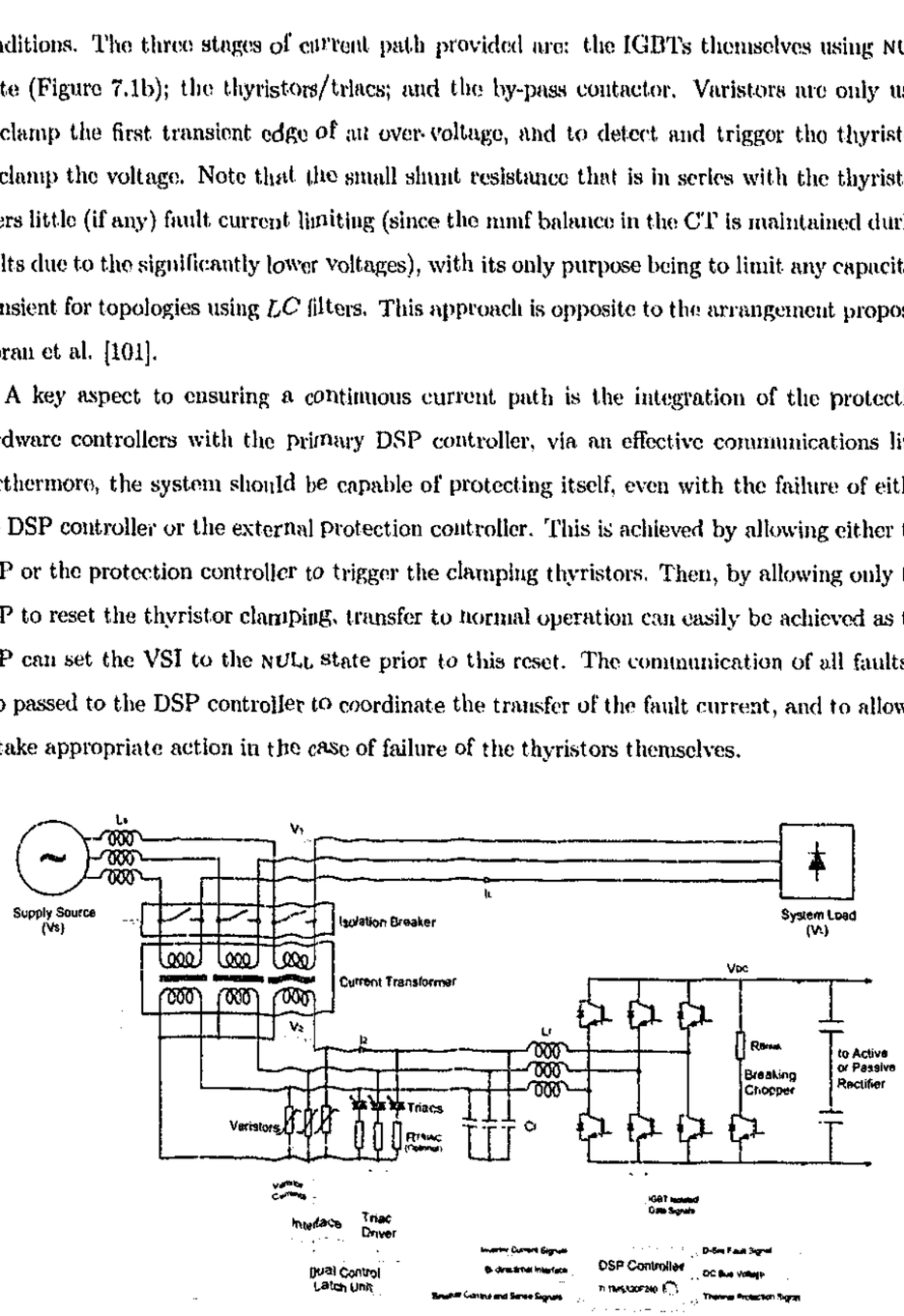


Figure 7.3: Series injected voltage source inverter, with integrated protection elements and controllers.

7.3 Elements of the Integrated Series Protection System

This section outlines the role and ratings of each of the elements in the proposed series protection system. The elements are the IGBTs (NULL and OFF state operation), varistors, thyristors/triacs, resistors, primary bypass breaker, dc-bus braking chopper, and the design of the series transformer.

7.3.1 Insulated Gate Bipolar Transistor (IGBT) NULL State

An obvious way to create a secondary current path is to switch the converter power stage to a NULL state as shown in Figure 7.1b. This state short circuits the filter inductances across the secondary (and hence the primary) of the series injection transformer, thus preventing an over-voltage. This strategy can be used both for start-up control, when the currents would be expected to be within the IGBT steady-state ratings, and temporarily at the start of an external load fault condition, until the short circuiting triacs can be triggered to provide a longer term fault current carrying capability.

7.3.2 Varistor (MOV)

Varistors are well known components that are often used to clamp over-voltage transients that may be developed in a system. However, as they have relatively limited thermal ratings, they usually cannot be used to provide a continuous fault current path, especially because they continue to sustain their breakover voltage while passing current. Hence, their role in this system is to clamp the first edge of any over-voltage, and then to provide a trigger to the short circuiting triacs so that they can provide a longer term current path.

The selection of a varistor requires determination and calculation of the steady state working voltage, breakover (or clamp) voltage, peak transient power dissipation and maximum continuous (average) power dissipation. For most varistor applications, working and breakover voltages are chosen based on the voltage rating of the connected system, the transformer ratio and the voltage rating of the devices, while the energy dissipated during an over-voltage event is the significant design criterion. However, in this scheme the peak transient power dissipation is determined by how rapidly the triac is triggered to clamp the secondary winding once the varistor current exceeds the trip set point. (Not that this trigger signal is filtered by a capacitor in the current detection circuitry. This avoids high-frequency voltage spikes causing unnecessary converter protection trips.)

For example, for the fault current levels of the experimental system, the varistors would have

to absorb more than 25 J over each fundamental cycle using the approach proposed by Moran et al. [101], and this would continue for as long as the fault existed. On the other hand, if even a very slow thyristor (for example 80 μs turn-on) were used to provide the longer term current path, the total energy dissipated in the varistor for the entire fault condition would be less than 1 J for the scheme proposed here. (In the experimental work described in Section 7.5 the thyristor turn-on time was measured to be less than 5 μs, so that only 55 mJ would be dissipated in the varistors). Since the Harris V130LA20B varistors that were used can absorb up to 70 J they were well overrated for this duty. However, if they were used as the core energy absorption devices, after three fundamental cycles they would have been well outside their ratings and much larger devices would be required.

In fact, for this application, the maximum continuous power dissipation for the varistor turns out to be the dominant factor in the varistor selection process, as follows. Maximum continuous power dissipation in the varistor occurs when the ac voltage across the devices causes a current just below the thyristor/triac trigger current set point to flow through the device. If this trigger set point is too high, a minor over-voltage can overload the varistors without the protection system operating. For example, the Harris V130LA20B varistors have a maximum continuous power rating of 1 W. This equates to a 30 mA current at their rated breakover voltage. Hence, if the trigger point was set to 40 mA, for example, and a minor sustained over-voltage caused a peak current of between 30-40 mA to flow, the varistors may fail thermally.

To calculate the average power dissipation of a varistor under ac voltage conditions, the commonly used V-I relationship (7.1) can be used to model the varistor

$$I = (kV)^{\alpha} \tag{7.1}$$

where α and k are taken from the varistor datasheet. α is the slope of the V-I characteristic on a log-log scale, and k is the inverse of the breakover voltage with a 1 A varistor current. Figure 7.4 shows the theoretical versus experimental performance for the Harris V130LA20B varistors using this model, for a peak current of just less than 5 mA. This result confirms (7.1) as a fairly accurate representation of a varistor.

Using (7.1), the instantaneous power VI dissipated by the varistor can be readily found, and integrating this power gives the average power dissipation. This provides an approximate method for calculating the average power dissipation with a sinusoidal voltage of V_{Trip} (peak)

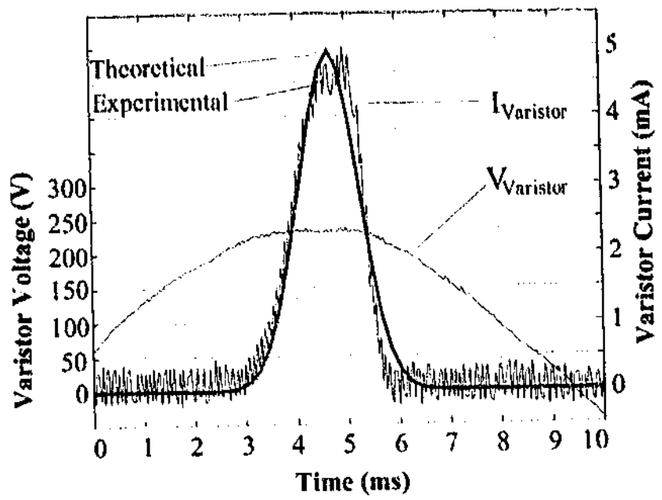


Figure 7.4: Varistor V130LA20B voltage and current waveforms.

across the varistor as

$$P_{av} = V_{Trip}^{\alpha+1} k^{\alpha} \frac{1}{2\pi} \int_0^{2\pi} \sin\omega t^{(\alpha+1)} d\omega t. \tag{7.2}$$

Figure 7.5 shows the value of the bounded integration term in (7.2) as a function of α. Replacing this integration term with a constant k_i (for a particular α) gives

$$P_{av} = \frac{V_{Trip}^{(\alpha+1)} k^{\alpha} k_i}{2\pi}. \tag{7.3}$$

Substituting (7.1) into (7.3) gives the average power dissipation of a varistor under sinusoidal voltage conditions as a function of the trigger current, viz:

$$P_{av} = \frac{I_{Trip}^{\left(\frac{\alpha+1}{\alpha}\right)} k^{-1} k_i}{2\pi}. \tag{7.4}$$

Equation (7.4) in conjunction with Figure 7.5 allows the thyristor trigger point current to be easily set based on the allowable continuous power dissipation of the varistor. The usage of these design equations is shown in the following worked example.

Worked Example: A Harris V130LA20B varistor with a maximum working voltage of 130 V_{rms} is chosen for the 110 $V_{rms-phase}$ system. The device is rated for a maximum average power dissipation of 1 W. From the datasheet, the values of α and k are determined to be 29 and 1/275,

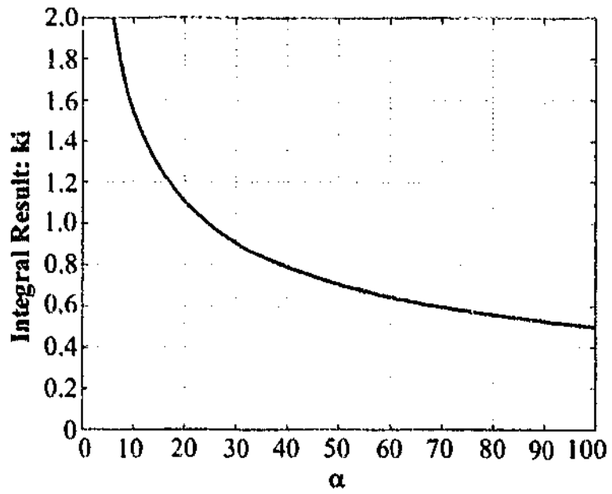


Figure 7.5: Numerical integration alpha constant graph.

respectively. Figure 7.5 gives k_i as 0.9 for $\alpha = 29$. Equation 7.4 then identifies the maximum trigger current I_{Trip} to be 30 mA for these values, to keep the continuous power dissipation of the varistor below 1 W. If less sensitive operation is required, a varistor with a higher average power dissipation would have to be used.

7.3.3 Thyristors/Triacs and Series Resistors

The short circuiting thyristors/triacs provide the main continuous current path for secondary currents during abnormal operation. They are triggered from a number of inputs, and latch on, to effectively short circuit the injection transformer through the (small) resistance R_{TRIAC}. This approach has a number of advantages, with only the relatively minor disadvantage of requiring higher current rated triacs than might otherwise be necessary.

The triacs should be rated to sustain the maximum secondary fault current ($I_{2-Fault-MAX}$) for the maximum fault time allowed by the power system protection relays. Allowance should also be made for the time taken for the breaker to close onto the fault in cases where the grid system relay protection does not operate as designed.

For systems with only an inductive filter, the resistance R_{TRIAC} is not needed. When the triacs are latched via the varistors, the DSP is notified to shut the converter down to stop the rising filter inductor current. If the notification process is not acceptably fast, then the over-current protection will act instead. In either case, the peak current from the converter is limited

by the filter inductance and will always be less than the peak pulse rating of the IGBTs.

For systems using an LC filter, when the thyristors/triacs short circuit the filter capacitors (C_f), R_{TRIAC} is required to limit the capacitive current surge to within the maximum short term current rating of both the thyristor/triac ($I_{Triac-MAX}$) and the filter capacitor (I_{Cf-MAX}). R_{TRIAC} should also be small enough to keep the voltage rise caused by the maximum secondary fault current ($I_{2-Fault-MAX}$) to well within the maximum allowable secondary voltage (V_{2-MAX}). These constraints are defined in (7.5), and should be used in conjunction with normal engineering safety margins.

$$\left(\frac{V_{2-MAX}}{\min\left(I_{Triac-MAX}, I_{Cf-MAX}\right)}\right) < R_{Triac} \ll \left(\frac{V_{2-MAX}}{I_{2-Fault-MAX}}\right) \tag{7.5}$$

Note that the smaller the value of R_{TRIAC}, the less impedance will be added to the connected power system by the series injection system. This is important both to minimize interference with the relay protection systems, and to keep the primary voltage drop within acceptable limits when the bypass breaker is open, but the converter is not yet operating. Note also that for higher power systems where suitably rated triacs may not be available, back-to-back thyristors can be used with separate isolated gate signals for each device. All thyristors should be simultaneously gated ON during the fault state (i.e., do not gate ON each thyristor separately depending on the current direction).

Worked example: ON Semiconductor MAC224A10 triacs were used for the experimental tests in the chapter. The continuous maximum current rating of the devices is 40 A_{rms}, with a peak non-repetitive surge current of 350 A_{peak}. The maximum allowable voltage (V_{2-MAX}) was set to 130 V_{rms}, with a maximum secondary fault current ($I_{2-Fault-MAX}$) of 30 A_{rms}. Under these constraints, (7.5) gives an allowable range for R_{TRIAC} of between 0.5-4.3 Ω. A value of 0.9 Ω was chosen so that during a fault the maximum secondary phase voltage is 27 V_{rms}, and the peak surge current from the capacitor is limited to 200 A_{peak}.

7.3.4 Primary Breaker

The main purpose of the primary side breakers across the injection transformer is to provide a bypass function, particularly since the breaker will not respond sufficiently quickly in most cases to provide a primary protection function. However, mechanical breakers do provide a "last ditch" option if the faster acting protection elements fail to operate correctly, or if the power system relay protection is delayed longer than expected during a system fault. Thus, the breaker should be rated to be able to close onto all fault conditions for the system, although a

limited number of operations at this maximum rating may be acceptable since it is not part of the primary protection system.

7.3.5 DC Bus "Braking" Chopper

This chopper provides the same functionality as for a variable speed drive, i.e. a control mechanism to avoid dc over-voltages. It provides a way of reducing the dc-bus voltage following a fault if rapid return to service is required. This may not be required in applications where the dc bus can be left to discharge slowly through resistors following a fault trip (as was the case in this work).

7.3.6 Transformer Design

Moran et al. [101] suggest that a protection function can be achieved by designing the series injection transformer to saturate under moderate over-currents and, hence, divert some of the fault current through the magnetizing branch of the transformer. However, for the integrated protection system described here, a custom injection transformer design with a low voltage saturation characteristic is not proposed. The higher voltage saturation characteristic allows applications such as Custom Power filters (which required precise control through the transformer) to achieve increased performance.

7.4 Series Protection System Operation

A key element of the protection system is its integration with the normal control system of the series converter application. Figure 7.6 illustrates how this integration is achieved, showing all major control states, including initializing the system, ramping the converter up to full operation and down to an off state, normal operation, and the extra control states required to implement the protection strategy proposed in this work.

For all faults, including external system faults, the converter operation must be properly shut down since the fault current referred through the secondary winding must be diverted away from the converter power stage. Converter shut down is done as a two-stage process – initially, the converter power stage is forced to a NULL state, since this can be done very quickly, and then the protection thyristors/triacs are triggered. Once it is confirmed that the thyristors/triacs have latched on, the converter power stage is switched off, forcing the full secondary current to the thyristors/triacs.

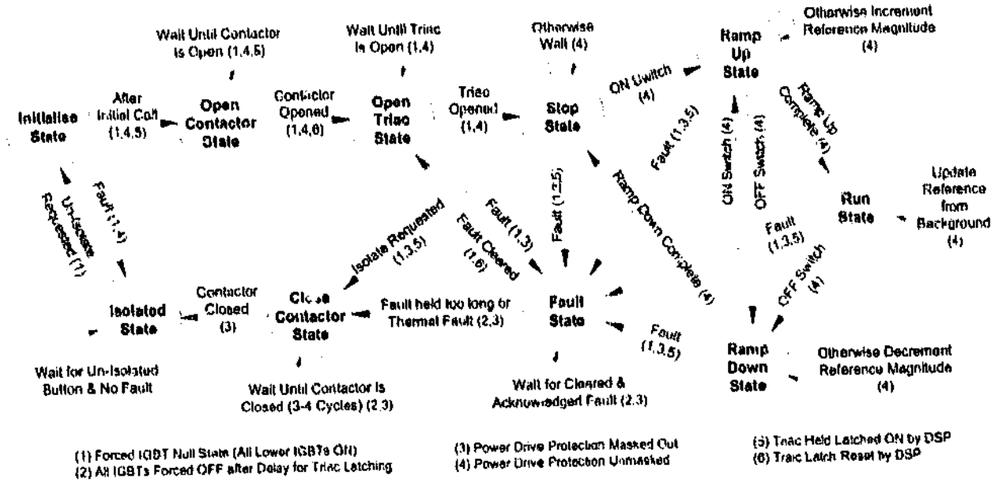


Figure 7.6: Series protection and control state diagram.

The control of the thyristors/triacs is arranged such that both the varistor current detection circuitry and the DSP can trigger the triacs, but *only* the DSP has the ability to reset the triacs. This is an important aspect of the protection system, ensuring that a current path is maintained at all times during the recovery from fault conditions.

If a longer term shutdown is required, the primary breakers can eventually be closed to bypass the series injection system from the external supply. This action is taken for thermal trips and for any faults that are held longer than expected. For external fault conditions, the maximum expected fault time is known from the power system protection relay settings. If a secondary over-voltage is the cause of the trip (as may be the case for an internal converter fault), the MOVs will initially clamp the over-voltage, thus directly triggering the protection thyristors/triacs, and immediately notify the DSP to take over control of the protection sequence.

A reverse sequence is used to control the start-up of the series injection system. Initially, the primary breakers are opened (if not already open). At the same time, the protection triacs are triggered to take over the load current from the breakers as they are opened and, finally, the converter operation is started with a NULL state (zero output), which is held for at least one half cycle while the triacs turn off after they are unlatched.

Note that the protection system can trip from a number of inputs, including thermal shutdown, converter over-current, IGBT desaturation detection, dc-bus over-voltage, and ac transient over-voltages. Fault conditions are detected by either the over-current protection or the

varistor over-voltage detection, or otherwise via the dc-bus over-voltage (as a last resort). It can be quite complex to determine exactly what part of the protection system will trigger first for a particular external fault situation. For systems using a filter capacitor, the rate of change of the secondary voltage is limited, typically allowing the over-current protection to clamp the voltage well before it has time to rise excessively. Without a filter capacitor, identifying which detection system is likely to act first for external faults becomes more complicated. The particular trip initiator is determined by the external system load conditions, the rate of change of the fault, the transformer capacitance and, of course, the sensitivity settings of each of the detection systems.

The protection system has inherent built in redundancies. Firstly, if either the over-current or ac over-voltage detection fails (i.e. a varistor fails), then the other detection system will act as a backup, with the dc-bus over-voltage detection providing further redundancy. Even if the DSP is not functional, the ac over-voltage protection will still operate and latch the triacs on. If the triacs themselves fail, then the varistors take over until the DSP closes the primary circuit breakers after the detection of a continuing fault.

7.5 Simulation and Experimental Verification

The operation of the integrated protection system has been investigated both in simulation and as part of the development of an experimental active series injection filter based around a TMS320F240 DSP (Figure 7.7). The results shown in this chapter use triacs for clamping, but for the series protection of the final low voltage experimental system described in Chapter 9, these were upgraded to larger rated thyristors to account for the increase in rating of the system. Initial tests were undertaken with a series shunt impedance to limit the current, before finally allowing the full fault current to be drawn to obtain the results shown in this section. Note that the Power Drive Protection Interrupt (PDPINT) in the DSP, which forces the IGBT's off upon interrupt, cannot be used in this system because the initial converter response upon a fault detection is to go to a NULL converter state, not an OFF state. Due to the fixed hardware relationship between the gate signals and the interrupt, the masking out of this interrupt is mandatory as noted in the footnotes provided in Figure 7.6.

Simulation studies have been used in particular to verify the proposed control logic for fault conditions that are difficult or hazardous to investigate in a laboratory experimental situation. The Simulink package in Matlab 5.3 was used in conjunction with the Power System Toolbox to complete the simulations. Once confirmed by simulation, the experimental system was exposed to the same fault conditions for final verification.

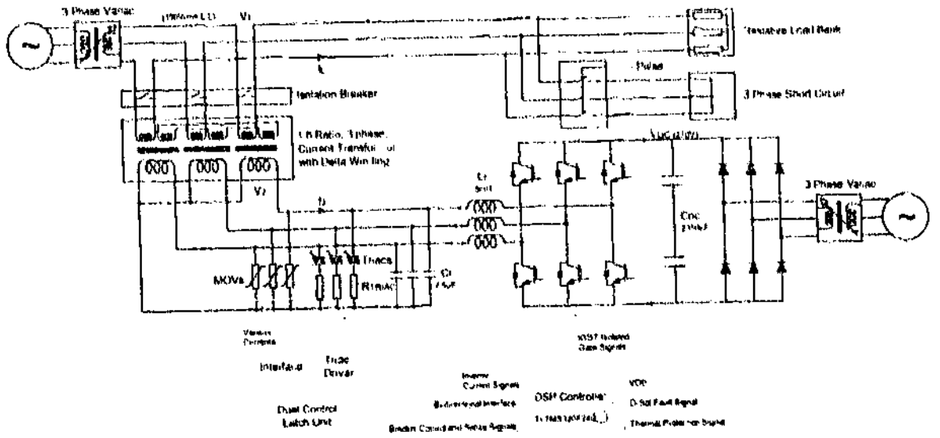


Figure 7.7: Experimental series protection and fault test setup schematic.

Figure 7.8 shows a simulated response of the complete protection system to a primary fault, and subsequent recovery. This simulation uses the exact setup configuration and parameters of the actual experimental system shown in Figure 7.7, and was used extensively to verify the protection system under all fault conditions before the experimental system was tested. The various stages of protection operation can be clearly seen, and the dc-bus voltage is maintained almost constant at all times. The simulation also shows how the converter can be quickly, and safely, returned to normal operation after a fault.

The fault configuration used for the experimental results given in Figure 7.9 was a three-phase short circuit. The fault was pulsed on and off by a purpose built timer connected to a contactor. Multiple oscilloscopes were used to obtain the results, all triggered from this timer. A delta tertiary winding was used to give a path for the zero sequence magnetizing harmonics generated by the transformers. These results show an example of the over-current detection sub-system tripping well before the secondary voltage had risen enough to trip the varistor detection circuitry.

Figure 7.10 presents experimental results for a protection trip initiated by an ac over-voltage detection by the varistors. As described before, this type of fault causes a direct triac trip, bypassing the converter NULL state stage. For this experiment the over-current trip was disabled and the converter switched off for safety reasons. The dc bus was stabilized using a fixed dc load. The effectiveness of the system in limiting the over-voltage and then diverting the injection transformer fault current into the protection triacs can be clearly seen. This result, together

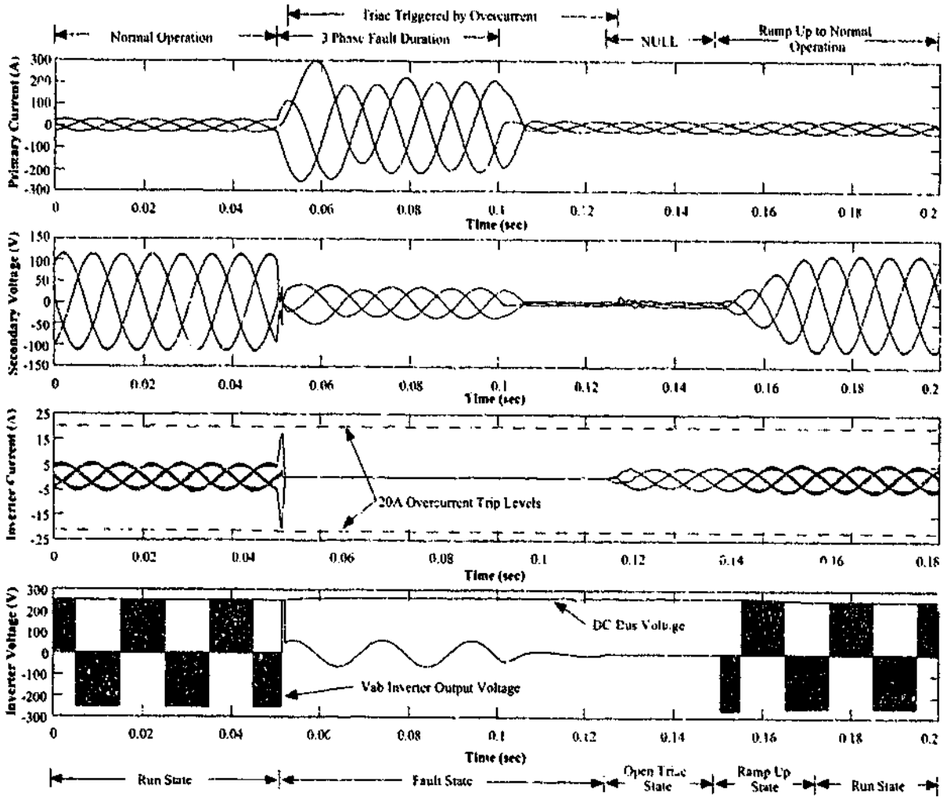


Figure 7.8: Three-phase fault and recovery simulation (over-current protection triggered).

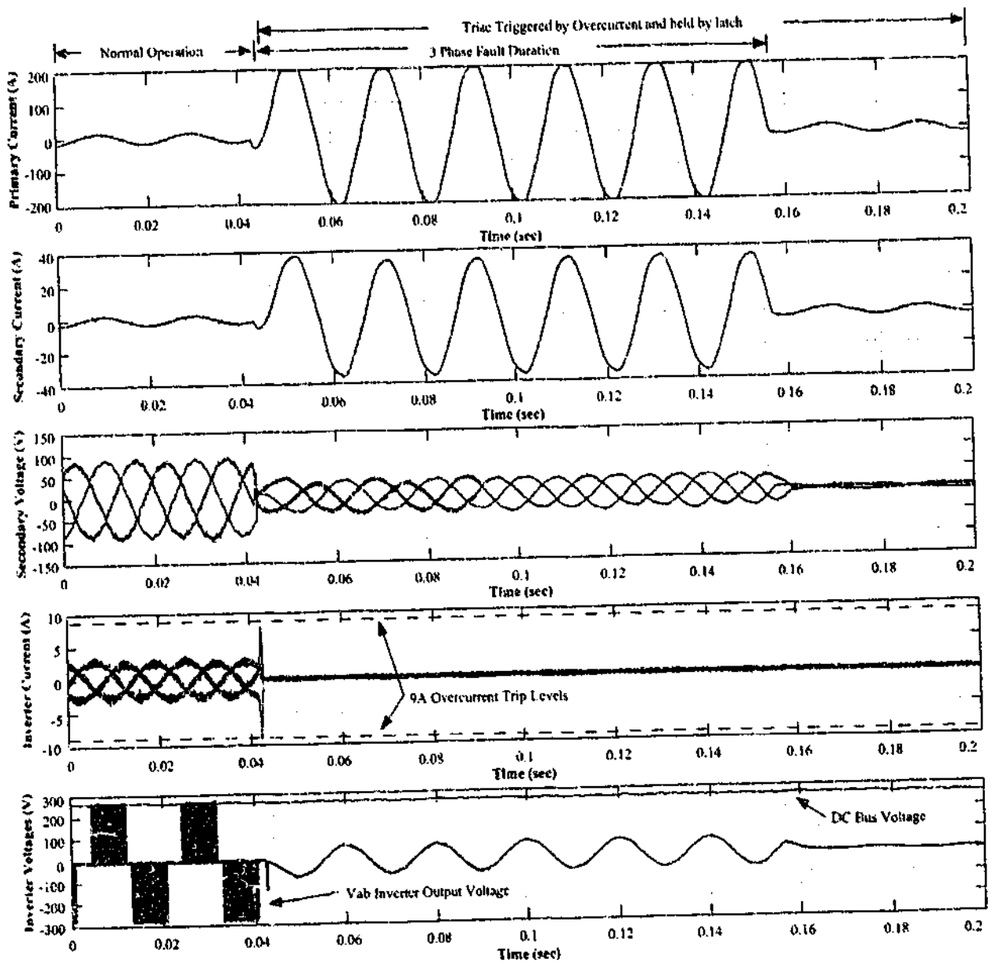
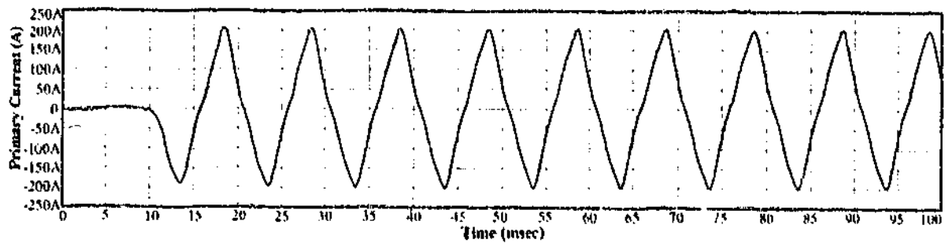
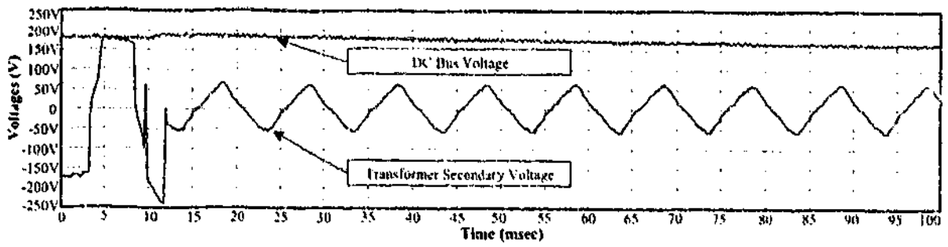


Figure 7.9: Three phase primary fault and recovery experimental results (over-current triggered protection).



(a) Primary Fault Current



(b) dc Bus and Transformer Secondary Voltages

Figure 7.10: 200 A-peak fault experimental results (ac over-voltage triggered protection).

with Figure 7.9, shows how both the over-current and over-voltage can protect the system independently, providing redundancy. Ideally, of course, both should be operational, as the effectiveness of each fault detection subsystem varies depending on the fault situation.

7.6 Summary

Control of an integrated scheme for protecting the UCPC has been presented, primarily focusing on the protection of the series component. The scheme protects the series injection converter against both internal and external fault conditions, and involves the coordinated operation of a number of elements, with the overall aim of providing a continuous path for the secondary referred current through the injection transformer at all times. Hardware and software considerations are discussed for various types of faults. The resulting scheme provides fault protection and also supports normal operations such as start-up, shutdown, and recovery from an external fault to restart the system.

The effectiveness of the protection scheme has been confirmed using Matlab simulations for a wide range of fault conditions, and has been shown to protect a complete experimental series injection system under three-phase fault conditions for both over-current and ac over-voltage trip indicators.

This protection scheme is used to control the states and to protect the low voltage experimental system in Chapter 9. But firstly, the overall UCPC and its controller design are verified via a detailed simulation as described in Chapter 8.

Chapter 8

Simulation Verification

Simulation studies are widely used in the field of power electronics. Simulations are useful design tools, and allow verification of a design prior to construction of the experimental system - therefore saving time and money. Simulation studies also enable investigation into events which are difficult (or unsafe) to generate experimentally.

The use of simulations is generally a compromise between execution time and detail, and therefore in this work varying levels of simulations have been used to design, analyze and verify the UCPC and its control systems. For the series component of the UCPC, a linear model was developed in Chapter 4, and has been used extensively to understand and design the system parameters to suit the controller. To account for the digitization of the control systems, simulation results were presented in Chapters 5 and 6 which included the numerical fixed-point effects of the DSP implementation. Finally, to verify the operation of the UCPC and its interaction with the grid, the last simulation developed is a full representation of the entire application.

This chapter details each of these levels of simulation, and outlines the purposes of the different models for the development of the UCPC. All of these simulations have been developed using MATLAB and its graphical extension Simulink, with the full system simulation also employing the Power System Blockset (PSB) add-on simulation components. Final simulation results for the UCPC are also presented to verify its operation before the experimental system test results are reported in Chapter 9.

8.1 Continuous Series Model Simulations

The first level of simulation is a design tool for the series controller, using the relevant equations that were developed in Chapter 4. Results from this simulation were presented in Chapters 4 and 5. The simulation is purely a continuous s-domain model, and does not account for any discretization effects. This allows classical control techniques such as Bode analysis to be quickly and easily used for the initial design phase.

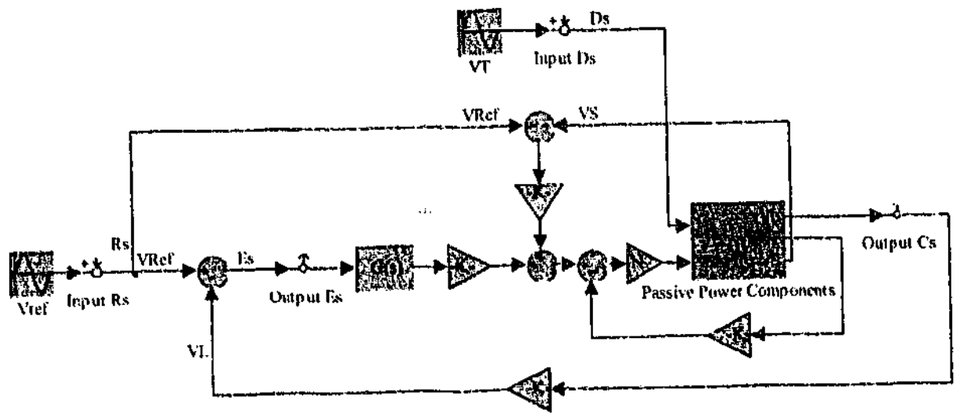
As with all the simulations reported in this thesis, the model uses a combination of MATLAB m-file scripts in conjunction with the MATLAB Simulink block diagram interface shown in Figure 8.1. The scripts are used to set the chosen parameters of the system, and Simulink is used to create the functional model. Simulink is generally used to create output waveforms from a system based on a given set of inputs (as is the case for the other Simulink models described later in this chapter). However, in this model, Simulink is simply used as a tool to create the state-space analytical model.

The model is separated into the separate control and physical elements of the system. The "Linear Analysis" option (part of the MATLAB Control Systems Toolbox) is then used to insert the input and output ports of the system into the model (Figure 8.1a), and consequently export the model into a user interface which allows investigation of the state-space model using a range of methods such as: Bode plots, Nyquist plots, pole-zero maps, impulse response and step response. Bode plots are used as the primary tool of investigation in this work.

The multiple P+Resonant feed-back controller was modelled using the s-domain functions presented in Equation 5.11 of Chapter 5. The VSI was assumed to be a unity gain block because of the dc-bus compensation portion of the controller, and also because of the regulation of the dc-bus voltage by the shunt portion of the UCPC. The remaining physical elements of system (including both the converter LC filter, transformer, and grid elements) were modelled using Equations 4.2 to 4.5 developed in Chapter 4. The arrangement of these elements is shown in Figure 8.1b. Note that these models have been purposely broken into separate blocks to allow access to different parameters of the system for investigation (such as disturbance rejection), and to also keep the size of the equations to a minimum to reduce the likelihood of errors. Gain blocks were placed in both the current and voltage feed-back paths to allow the feed-back to be either enabled or disabled. For the system Bode plot creation, the voltage feed-back gain was set to zero, to achieve the required open loop system response. For the disturbance plots, this gain was set to unity.

Note that a similar model is not presented for the shunt converter, as the outer linear

controller is open loop, and therefore does not require the same stability analysis. The main stability concern for the shunt converter is with regard to the inner dead-beat current regulator, but this is not a primary focus in this work, and has been addressed by Bode et al.[14] and Malesani et al.[109].



(a) Entire series system

(b) Inside the "Passive Power Components" block

Figure 8.1: Linear model (Simulink) of the series controller and physical system elements.

8.2 Digital Controller Frequency Sweep Simulations

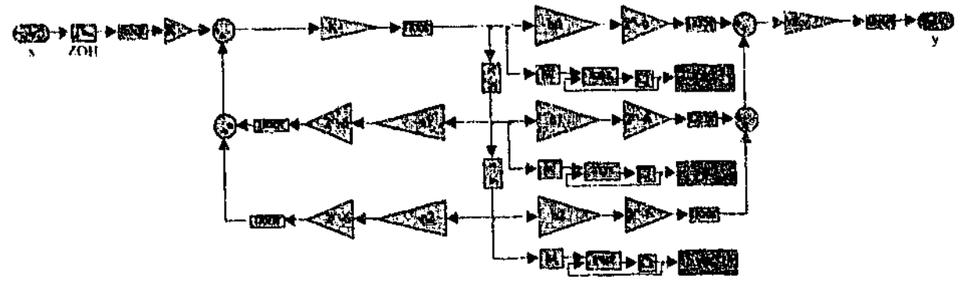
To complement the continuous linear model presented in the previous section, the digital controllers must be investigated to ensure that they do actually achieve the desired transfer functions. Whilst MATLAB already includes analysis tools for discrete digital filter systems, these tools do not account for numerical fixed-point errors, which were shown in Chapter 6 to be a significant problem for the stationary frame controllers used in this work. To resolve this problem, an exact model of the proposed digital filters was created in Simulink, including all the fixed-point numerical errors created by the DSP software implementation. Figure 8.2a shows an example of this Simulink model, and illustrates the delta based IIR (DFII form) machine code software form used in the final experimental work for the digital linear controllers in both the series and shunt converters. The models were verified by comparison with their experimental counterparts in Chapter 9.

To create the required frequency plots, the simulation is run for each selected frequency (until the response has settled), and the output phase and magnitude is then measured with respect to the input driving signal. The Simulink model for the single filter (e.g. P+Resonant) and cross-coupled filters (e.g. StatRF harmonic extraction unit) are shown in Figures 8.2b and 8.2c, respectively. (This method is extended from work by Daniel Zmood [114], who used a similar simulation to create more accurate Bode plots for the continuous versions of the P+Resonant controller.) The Simulink file is managed and executed by associated m-file scripts which perform the following tasks; define the controller parameters, create the digital filter coefficients, run the Simulink simulation for each frequency under the desired conditions, and plot the output results. This simulation is used to create the frequency plots of the shift IIR, delta IIR and FIR digital filters presented in Chapters 5 and 6. As each plots requires between 100 to 1000 data points (and therefore an equivalent number of simulation runs), this simulation method can become very time consuming. For this reason this method is only used to create and check the digital design of the controllers, and is not used as part of the series system design tool presented in the previous section.

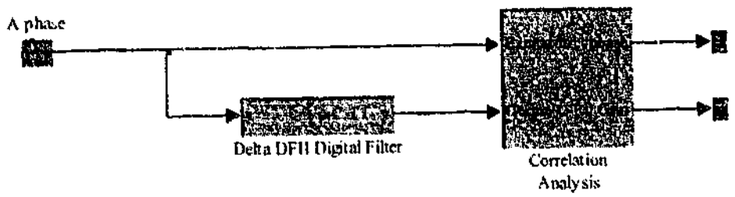
Since the model was an exact replica of the software implementation, the simulation was also used to investigate the maximum size of the internal variables (e.g. var1, var2, and var3 in Figure 8.2a). The simulations found that the largest variables always occurred for input driving frequencies around the resonant frequency of the filter. Using this simulation, the design was altered to ensure that the maximum available sinusoidal input at the resonant frequency did not cause the internal variables to exceed $\pm 2^{15}$, and therefore overflows did not occur in practice. If

a design solution is not possible, the simulation can instead be used to find the maximum input limit to allow other clamping limits to be set to stop overflow events occurring.

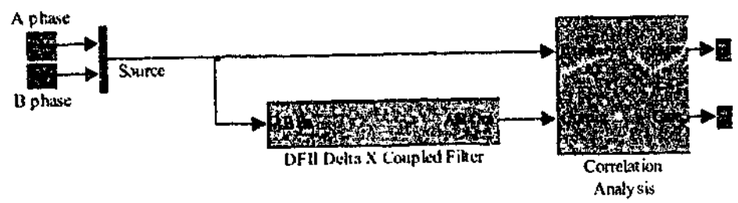
The ode45 solver was chosen for use with the Simulink simulation. Ode45 is a variable step Runge-Kutta based solver for the ordinary differential equations (ode) created by MATLAB to generate the required output. This is one of the recommended solvers for models which are not stiff (refer to the definition of stiff in Section 8.3) and contain both continuous and non-continuous states [154].



(a) Simulink delta based DFII IIR digital filter simulation (based on DSP implementation).



(b) Simulink digital filter analysis simulation (single filter)



(c) Simulink digital filter analysis simulation (cross coupled filters)

Figure 8.2: Simulink models used to create the frequency sweep simulations of the digital systems.

8.3 Complete UCPC Detailed Simulation

So far two levels of simulation have been presented. The first, an analytical simulation, is primarily used as a design tool for the control parameters and physical parameters of the series portion of the UCPC. The second, a numerical simulation, is a frequency based design and verification tool for the fixed-point DSP software implementation of the digital filters (used for both the series and shunt controllers). The final numerical simulation presented in this section is a detailed time based model of the entire UCPC and the connected grid system, as shown in Figure 8.3.

Once again a combination of MATLAB and MATLAB Simulink is used to develop the simulation, and the Power System Blockset (PSB) [155] is now also used to model the electrical system components. The PSB elements used in the system model are: the VSI, transformers, inductors, capacitors, resistors, and the non-linear diode rectified load. The VSI model (Figure 8.4) uses the PSB Universal Bridge to model the three phase-leg converter. The model incorporates the forward voltage drops of the IGBT and diodes, as well as the positive and negative switching times of the phase legs, but does not allow for any further detail to be modelled. However, considering that the quantities of interest are the resulting power system voltages and currents, any further modelling of the VSI would only slow the simulation down, while providing very little (if any) variation to the outcome. The PSB series injection transformer model includes typical transformer model parameters, as well as the saturation characteristic of the transformer. The values used in the simulation are based on the transformer used in the LV experimental work (Chapter 9), and were obtained by experimental measurement (refer to Table B.4 in Appendix B for the actual values).

The series controller model shown in Figure 8.5 includes the appropriate scaling of the measurement variables, such that the magnitudes of the integer values are equivalent to those in the experimental DSP, and account for all the numerical errors of the fixed-point DSP. To achieve this, the digital filter models presented in the previous section (Figure 8.2a) are incorporated into the controller model (Figure 8.6). The controller model for the shunt controller is presented in Figure 8.7, and again includes the same scaling as is used in the experimental system.

One problem with such a detailed simulation is the choice of a suitable solver and time step resolution, such that each simulation run can be completed within a realistic time-frame, but will also provide sufficiently accurate results. This simulation can clearly be defined as a stiff system, and a solver designed for such conditions must be chosen. (A stiff system is one where the "solutions can change on a time scale that is very short compared to the interval

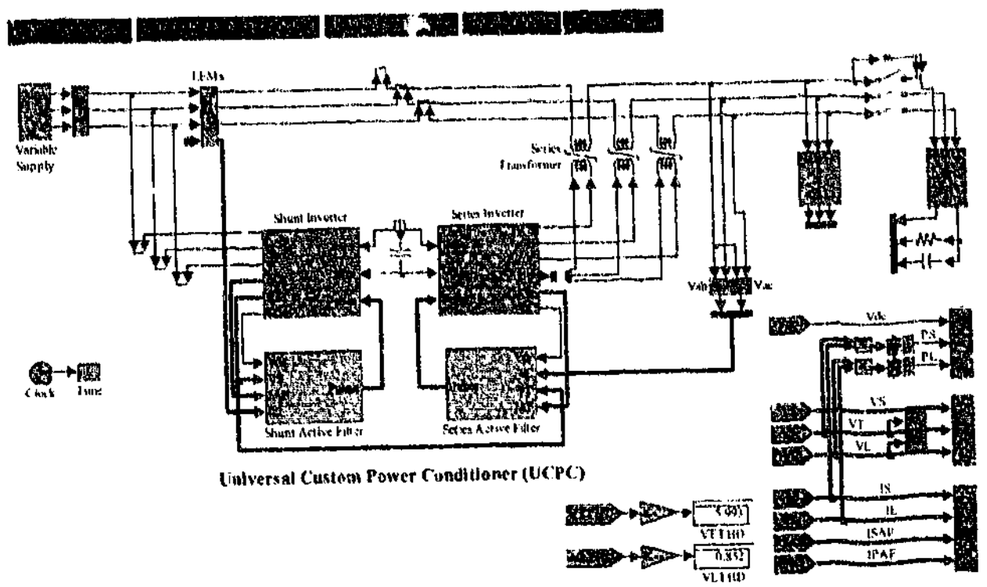


Figure 8.3: Power System Blockset based Simulink model of the UCPC and test grid system.

of integration, but the solution of interest changes on a much longer time scale" [154].) The ode23tb (TR-BDF2) solver option was therefore used. This variable step solver is designed for use with stiff systems and uses a Runge-Kutta based method (as per the ode45 solver used in the previous simulation), with a two stage implementation. The first stage uses the trapezoidal rule (TR), while the second stage uses a second order backward differentiation formula (BDF2) [154]. This method typically provides a faster simulation speed for stiff systems (compared to the ode15s and ode45 methods [155]). To further increase the simulation time, the PSB elements were also discretized (using a 2 μs step time). (Note that the fully discrete variable step solver could not be used as the model still contained some continuous states.)

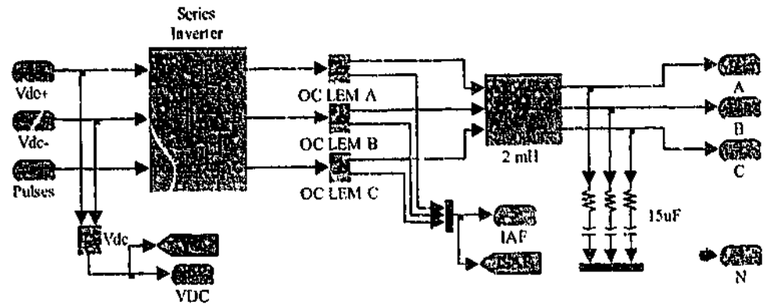


Figure 8.4: Series VSI sub-system shown in Figure 8.3.

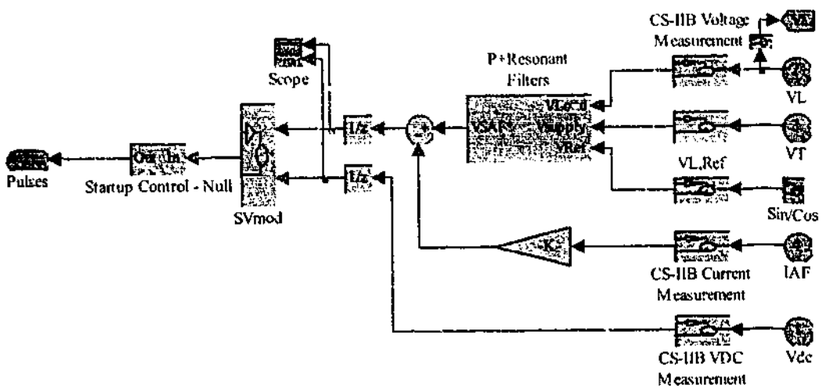


Figure 8.5: Series controller sub-system shown in Figure 8.3.

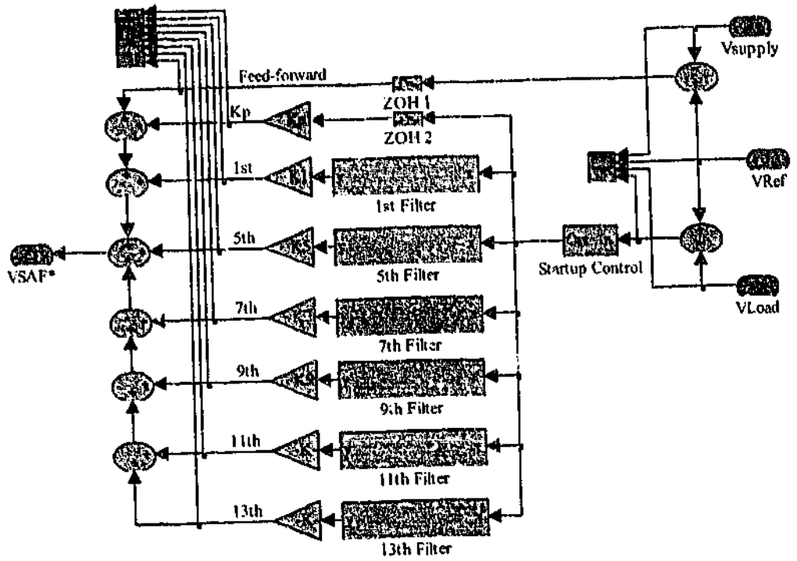


Figure 8.6:

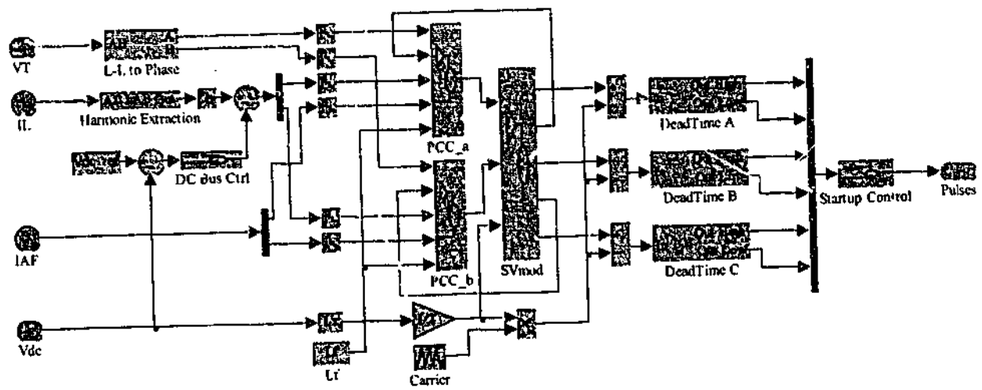


Figure 8.7:

8.4 UCPC Simulation Results

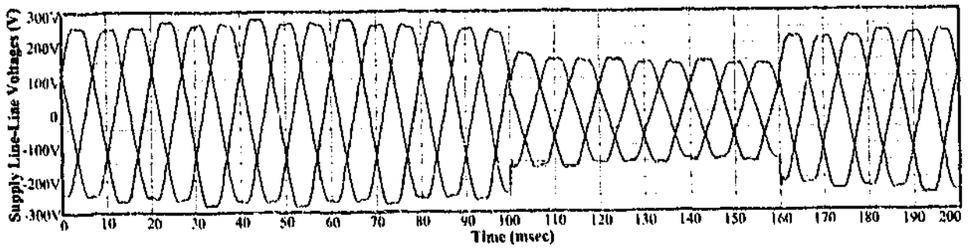
This section presents results from the simulation model described in Section 8.3. The system parameters used for these results match those of the low voltage experimental prototype described in Chapter 9, with the exclusion that the supply impedance is set to be much stiffer. This allows the simulation to be used to investigate the proposed UCPC under more realistic conditions than can be achieved with the weaker laboratory supply voltage that was available[1]. The low voltage experimental parameters can be found in Table B.2 of Appendix B.

Figure 8.8 presents system voltage and current waveforms with the UCPC in operation. The supply voltage (Figure 8.8a) contains unbalance (0.9 p.u.), flicker (0.1 p.u. at 5 Hz), harmonics (3.4% THD), and a balanced sag (0.7 p.u.). The load current (Figure 8.8c) is made up of a combined linear/non-linear load. Figure 8.8b shows that the UCPC compensates for the flicker, unbalance, harmonics and sag in the supply voltage to create a clean sinusoidal waveform for the load. Note that at the leading and trailing edges of the sag, small sharp disturbances can still be seen in the load voltage. These come about because of sample delay before the series control system can begin to react to the change in conditions. Figure 8.8d shows the active filtering and balancing function of the shunt portion of the UCPC, to create a sinusoidal and balanced current drawn from the supply. During the sag the supply current can be seen to increase due to additional power demanded by the dc-bus regulation scheme to supply the power injected by the series UCPC converter. The temporary transient in the dc-bus voltage can be seen in Figure 8.8e as the dc-bus PI controller reacts to regulate the dc-bus voltage back to its demanded value.

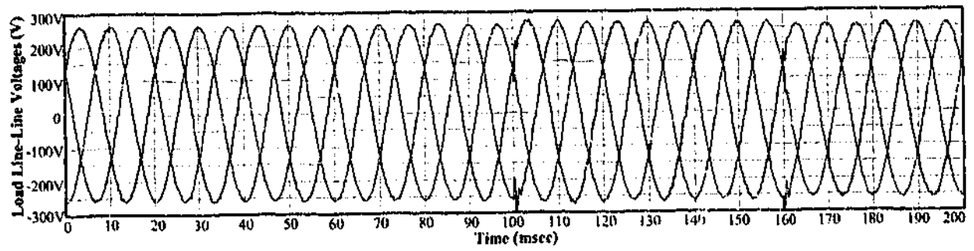
8.5 Summary

This chapter has described the analytical and numerical simulations used in this research work as a tool for both the design and theoretical verification of the proposed UCPC. All simulations use MATLAB together with its graphical extension Simulink. The Power System Toolbox is also used to model the electrical components in the full system model. With the system designed and verified in simulation, the next step is to experimentally verify the proposed UCPC and its control systems. Chapter 9 presents the low voltage experimental prototype and results for the entire UCPC. Chapter 10 extends this work for the series component of the UCPC to a medium voltage experimental prototype.

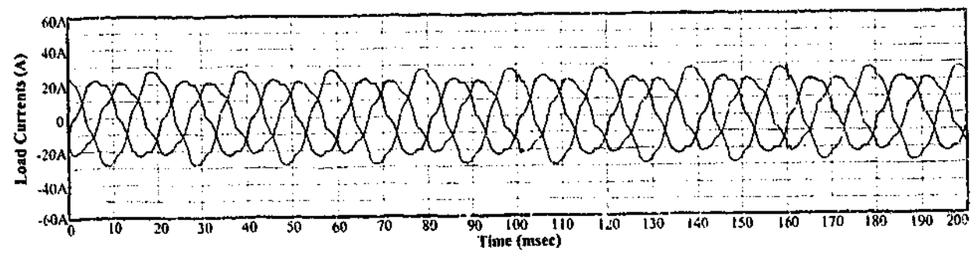
[1]Note that the simulation was also run using the same supply parameters as the low voltage experimental system to verify the accuracy of the model, but these results are not included here.



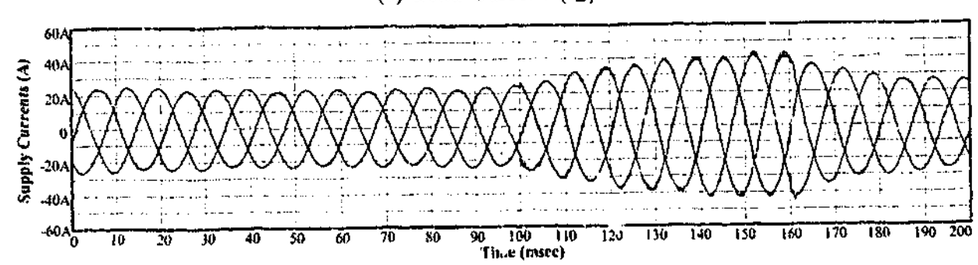
(a) Supply Voltage (V_T)



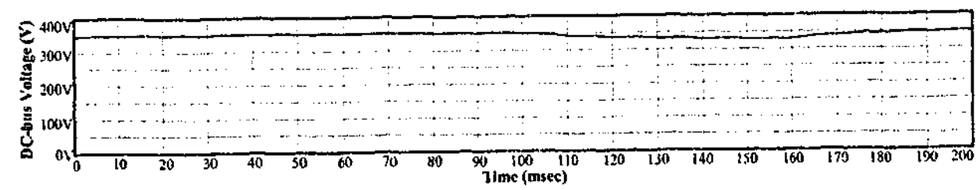
(b) Load Voltage (V_L)



(c) Load Current (I_L)



(d) Supply Current (I_S)



(e) Dc-bus Voltage (V_{DC})

Figure 8.8: Simulated results of the UCPC compensating for a distortions in the load voltage and supply current.

Chapter 9

Low Voltage Experimental Verification

[1]In the previous chapters the design and application of the UCPC control systems have been discussed in detail, and have also been theoretically verified in simulation. This chapter describes the experimental verification of the UCPC operation using a low voltage experimental prototype.

The low voltage (LV) experimental work presented in this chapter involved the development of: suitable hardware controller boards; DSP software systems for both converters; design and construction of back-to-back voltage source inverters; magnetic design; protection system development; construction and housing of the UCPC; design and construction of ancillary components; the development of the suitable testing environment; and the final experimental testing to analyze and verify the research work.

Chapter 6 presented the software techniques used for the digital controllers, and the protection systems were discussed in detail in Chapter 7. The remaining hardware and software components of the LV experimental system will be presented here, and the final results will then be presented and discussed. These results verify the capabilities of the UCPC to compensate for voltage fundamental magnitude/harmonics/unbalance/sags/swells/flicker and current harmonics/unbalance. Chapter 10 then extends some of the series control components of this work to a medium voltage (MV) system.

[1]The experimental results in this chapter were first published in part as:
M. J. Newman and D. G. Holmes, "A Universal Custom Power Conditioner with Selective Harmonic Voltage Compensation", *in Conf. Rec. IEEE/IECON*, Seville, Spain, Nov., 2002.
M. J. Newman, D. N. Zmood, and D. G. Holmes, "Stationary frame harmonic reference generation for active filter systems," *in Conf. Rec. IEEE/APEC*, Dallas, TX, pp. 1054-1060, 2002.
M. J. Newman, D. N. Zmood, and D. G. Holmes, "Stationary frame harmonic reference generation for active filter systems," *IEEE Trans. Ind. Applicat.*, vol. 38, no. 6, Nov./Dec., pp. 1591-1599, 2002.

9.1 Experimental System Overview

Throughout the course of this research work, the experimental UCPC system evolved and has taken many different forms. Originally the constructed system used pre-existing components such as controller hardware developed by the Power Electronics Group at Monash, converters, contactors and magnetics. Each of these components were then redesigned, re-built and upgraded to better suit the UCPC, and the entire system was housed in a 19" rack. However, only the final version of the system is significant for this work, and this is the experimental system that will be discussed. The UCPC can be divided into its hardware and software components, and each of these will be detailed in the following sections.

Figure 9.1 presents the hardware components and interconnections for the LV experimental system. The converters and magnetics of the UCPC are rated for 20 kVA each, and are designed for operation on systems up to 415 V. The series injection system can be configured for either 0.5 p.u. or 0.25 p.u. voltage injection, and can therefore protect loads of up to 40 kVA and 80 kVA, respectively. (All tests presented in this chapter were conducted using the 0.5 p.u. series injection configuration.) The controller hardware, current measurements, series transformers, and cabling have all also been constructed for possible use with a four-wire system, to allow for future work. However, in this chapter only a three-wire experimental verification is presented, and any redundant components are therefore not included in relevant diagrams and discussions.

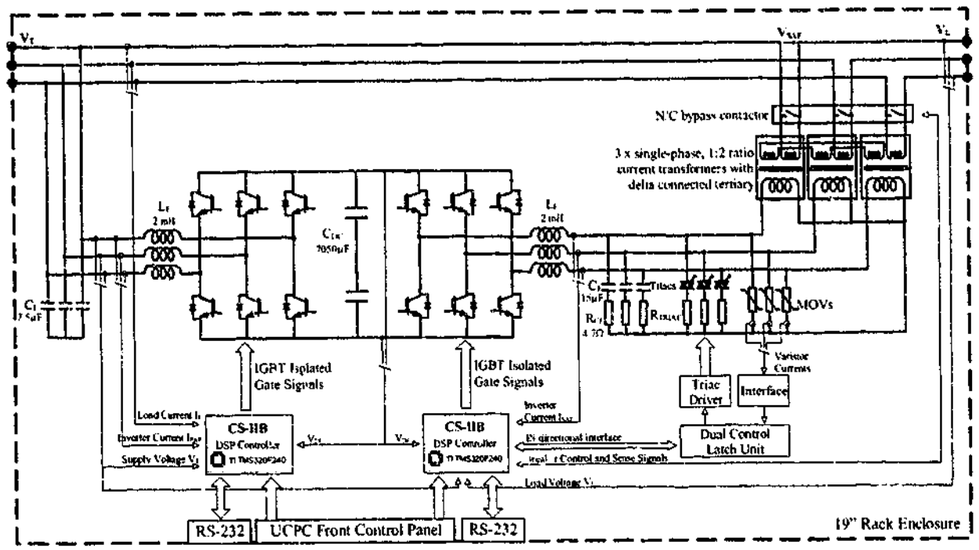


Figure 9.1: Components of the experimental UCPC prototype.

9.2 Hardware Development and Configuration

The hardware development of the UCPC required the design and construction of a wide range of elements ranging from analog and digital electronics to power semiconductor stages. The primary hardware elements are: the controller hardware, converters, magnetics, protection hardware, ancillary components, cabling, and housing (Figure 9.1). These elements are discussed in the following sub-sections. (Note that more hardware detail, than is required for this research, is provided in the following discussions to assist in the use of the hardware by other researchers who may follow on from this work.)

9.2.1 Controller Hardware - Integrated Inverter Board (CS-IIB)

A controller for the system was specifically designed to meet the requirements of the proposed UCPC. Separate controllers were used for each of the series and shunt converters. They were designed to be general enough so that the same controller hardware could be used for each system. The controller hardware unit is broken up into two PCB boards. The first is an existing DSP controller card developed by the Power Electronics Group (PEG) at Monash University, known as the CS-MiniDSP. The card is based on the Texas Instruments TMS320F240 DSP, which is a 16-bit fixed-point processor with internal hardware dedicated for use with PWM based systems. The second PCB is a daughterboard for the MiniDSP, and the circuit design and initial PCB layout were undertaken as part of this research, with the final PCB layout being completed in conjunction with other members of PEG. The daughterboard (named the Integrated Inverter Board, or CS-IIB) integrates all the required analog and digital circuitry for the converter control and interface into one unit, and also can incorporate a low power (2 kVA) VSI on the same PCB. The on-card power stage allows low power tests of the system to be conducted before the transition to a higher power system. For higher power and full voltage testing the built in power stage is removed and the gate driver outputs are connected to an external power stage. This is the configuration used for the results presented in this chapter.

The functional capabilities of the CS-IIB are shown in Figure 9.2, and a photograph is presented in Figure 9.3. The system is designed to be fully self contained to minimize the requirements for additional circuitry. The board allows for direct measurement of up to six current measurement devices (either current transformers or hall effect sensors), as well as direct measurement of two sets of three-phase voltages and two dc-bus voltages. Eight fully isolated gate drivers are included for controlling either the on-board power stage or external IGBTs. The separate low voltage dc supplies required by all the sections of the CS-IIB are

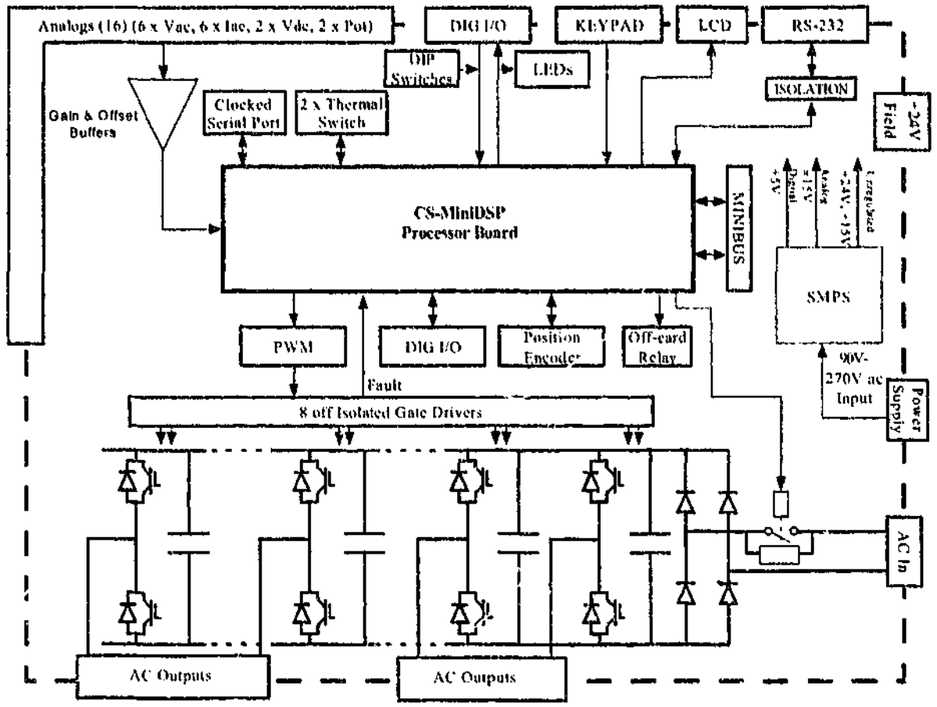


Figure 9.2: Functional specification diagram for the Integrated Inverter Board (CS-IIB). (Courtesy of P. McGoldrick)

derived internally via a fly-back based switched mode power supply (SMPS). Digital interface circuitry includes digital I/O, key-pad input, LCD display output, Serial Peripheral Interface (SPI), position encoder, RS-232, and also a Mini Bus port for connection of other peripheral devices. Full details can be found in the CS-IIB [156] and the CS-MiniDSP [157] technical manuals.

Analog Circuitry

The CS-IIB contains all the analog gain and offset circuitry required to convert the current and voltage measurements to levels suitable for the DSP's on-chip analog to digital (A/D) conversion sub-system. Both ac and dc voltages of the system can be directly connected to the CS-IIB analog inputs. Two sets of three-phase ac voltages can be measured, as well as two dc voltages. The current measurement inputs accept either hall effect sensors (such as LEMs) or current transformers. Burden resistors are included on posts (to enable easy substitution), as well as ± 15 V analog voltage supply for use with the hall effect sensors. For the system described

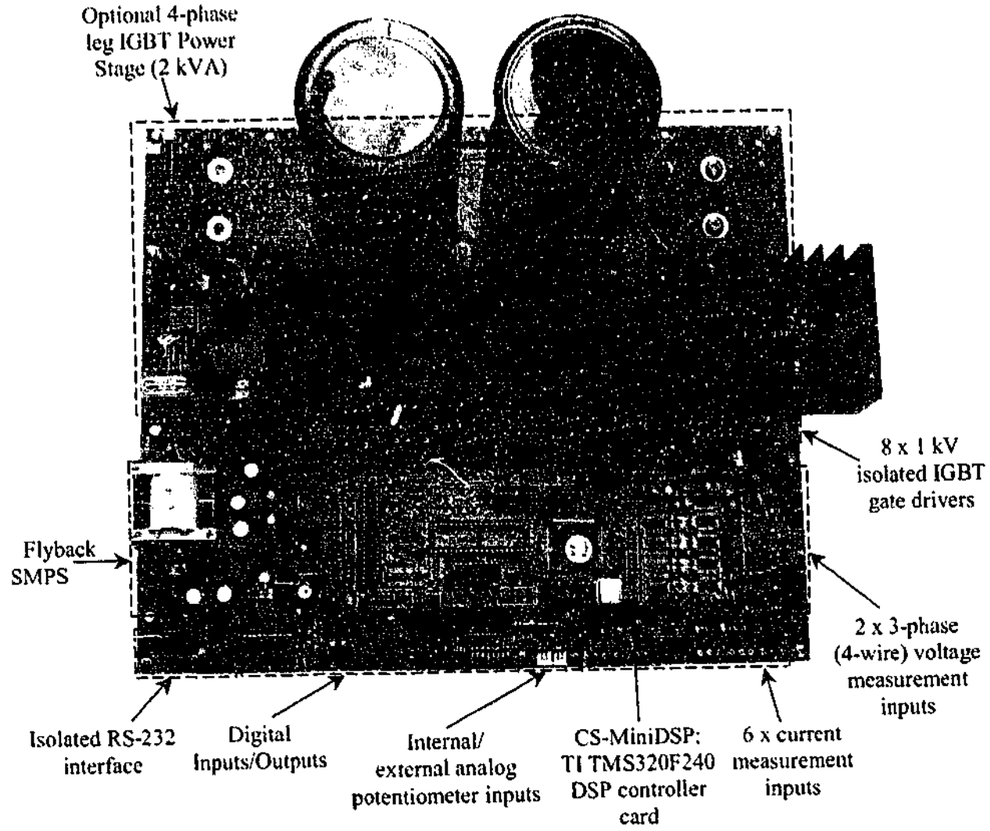


Figure 9.3: Photograph of CS-IIB controller hardware, with component layout diagram.

in this chapter LTA-100P/SP1 (1000:1) LEMs were used for all current measurements.

There is also additional circuitry incorporated with the voltage and current measurements to detect converter over-currents and dc-bus over-voltages. These signals are connected to the protection hardware described later in this section. The first phase on both sets of ac voltage inputs also includes circuitry to convert the ac signals into a square wave digital signal. The DSP can use this signal for implementation of a software based phase-locked-loop (PLL) to synchronize to the ac signal. Finally, in addition to the sample and holds provided with the A/D conversion on the DSP, four external sample and holds are provided to enable simultaneous sampling of up to six analog quantities at the start each PWM sampling period.

There are two potentiometer inputs, which can use either an on-card multi-turn pot, or accept an external ±10 V input. The external potentiometer inputs are used by the shunt section of the UCPC to allow the user to choose the demanded dc-bus voltage. A second input is used to select the percentage harmonic compensation provided by the shunt active filter (typically 100% is chosen)

Digital Circuitry and Communications Interfaces

A range of digital input and output interfaces are required to allow both interaction with the operator, and communication and control of other devices connected to the system. The operator interface uses both digital I/O and the RS-232 interface to control the UCPC, and is programmed to allow simultaneous operation. Digital I/O is used for the start/stop/un-isolate/isolate buttons for both the series and shunt converters. The RS-232 interface to a PC duplicates these operations and also allows more detailed control of other parameters in the controllers. Digital I/O is also used to provide the interface to the series protection control hardware.

The Mini-bus interface on the CS-IIB can be used for the connection of other analog and digital boards (including the digital to analog (D/A) board used occasionally in this project for test and measurement of the DSP control variables in real-time). The CS-IIB also provides capabilities for connection of an LCD and for a key-pad (up to 4 x 5 key grid), as well as a Serial Peripheral Interface (SPI) and a position encoder interface.

Finally, the CS-IIB includes a Mosfet digital output, and when the on-board power stage is not loaded (as in this case) the soft start relay is also available for use. These two outputs were used in conjunction with the 24 V field supply to control the shunt soft start contactors, and to control the series bypass contactor.

DSP Controller

The DSP is located on the CS-MiniDSP board. The controller uses a single-chip DSP solution for both the controller and the PWM generation. The Texas Instruments TMS320F240 16-bit fixed-point DSP was chosen because of its specific hardware designed for digital control of PWM converters. The DSP contains an event manager which uses an up/down counter as the PWM carrier reference, and up to 6 registers can be loaded with reference compare values. Three of these are for complementary switch pairs (i.e. a VSI phase-leg), and the other 3 are for individual device control. This allows up to 9 PWM signals to be produced, but only 8 are passed to the CS-IIB for control of the four phase-legs. Built in hardware dead-time is provided for the three complementary pairs, and these are the outputs used by both the series and shunt converters in this work. An external interrupt is also linked to the PWM outputs, such that all outputs are switched low when the interrupt is triggered (and enabled). This is called the Power Drive Protection Interrupt, or PDPINT, and is used to stop the converters once a fault is detected on either the IGBTs (via the gate driver protection), converter over-current, dc-bus over-voltage, or thermal trip. These trip indicators are added together into one signal by the CS-IIB, before connection to the DSP on the CS-MiniDSP.

The DSP has both internal Flash ROM and RAM, and the CS-MiniDSP card also includes an extra 128 kb of external RAM, and 64 kb EPROM for use with the DSP.

IGBT Gate Drivers

Eight fully isolated 900 V IGBT gate drivers are included on the CS-IIB for use with either the internal power stage, or for driving an external power stage (as used in this work). The PWM signals are optically isolated using the HCPL-316J IC, which also provides a gate driver facility in conjunction with other circuitry. The isolated power supplies for this IC are generated using a common high frequency supply into the pulse-transformer of each gate driver. The rectified output is then regulated into +17 V and -12 V supplies to drive the gate driver IC.

Switched Mode Power Supply (SMPS)

The SMPS accepts ac inputs in the range from 90 V to 270 V, or dc inputs from 130 V to 370 V. There are multiple outputs from the flyback transformer. The output dc voltages are: regulated 5 V for digital circuitry, ±15 V regulated supply for on-board analogs and external hall effect current sensors; an unregulated +16 V supply for gate drivers; an unregulated +12 V supply (linked to digital supply) for the Mosfet driver and miscellaneous external use; and an

unregulated +24 V field supply for external use. An isolated 5 V supply is generated from the digital supply for use by the RS 232 communications port.

9.2.2 Voltage Source Inverters (VSI)

The series and shunt voltage source inverters (VSI) both use the same power semiconductor devices and are therefore equivalently rated. The design of the VSI included a thermal model, rating analysis, and lifetime estimation of the electrolytic capacitors. The mechanical design was drawn up in AutoCAD to aid in the construction. (The thermal/rating model for the VSIs was completed and experimentally confirmed in previous work by the author, and can be found in [158].) Each VSI is rated for approximately 35 kVA with fan forced cooling (13 kVA with natural convection cooling), based on a 700 V dc-bus voltage, 40 °C ambient temperature, 20 °C head-room in the maximum junction temperature, and a 5 kHz switching frequency. However, since the remainder of the UCPC components (i.e. magnetics, etc.) are only rated at 20 kVA, the maximum steady-state operation of the VSIs has been reduced to this value. The 20 kVA limitation also extends the expected life of the dc-bus capacitors to greater than 10 years, due to the reduced ripple current [158] [159].

The voltage source inverters (VSI) were constructed using Siemens BSM 50 GB 170 DN2 IGBT [160] phase-leg packages. Cooling of the IGBTs is via H31 type heatsinks, and they can be either operated with fan cooling (0.07 °C/W), or naturally convected (0.28 °C/W), depending on the application. Planar bus bars, made from nickel plated copper, are used to reduce the parasitic inductance between the IGBTs and the dc capacitors, as well as to provide an easy mechanical structure that was cost effective for manufacture. By reducing this inductance, the overshoot on the switching edges was significantly decreased, and no snubber components are required. The bus plates were dimpled at some of the connection points to remove the need for spacers. This made assembly significantly simpler, removing unwanted connection resistances, and minimized sharp conducting angles that are prone to increasing EMI effects. The dc-bus capacitors are mounted between the two converters. Six FELSIC 85/S 4700 μF 450 V electrolytic capacitors [159] are used in a series/parallel combination to create a net dc-bus rating of 7.05 mF and a maximum voltage of 900 V. Small 0.1 μF WIMA high frequency film capacitors were located on the dc-bus directly next to each phase-leg to overcome the problems caused by the large parasitic inductance internal to the electrolytic capacitors. A 3D rendering of the two VSI is shown in Figure 9.4 to illustrate the layout of the components of the two converters.

The output of each VSI has an LC filter to attenuate the PWM switching harmonics. The

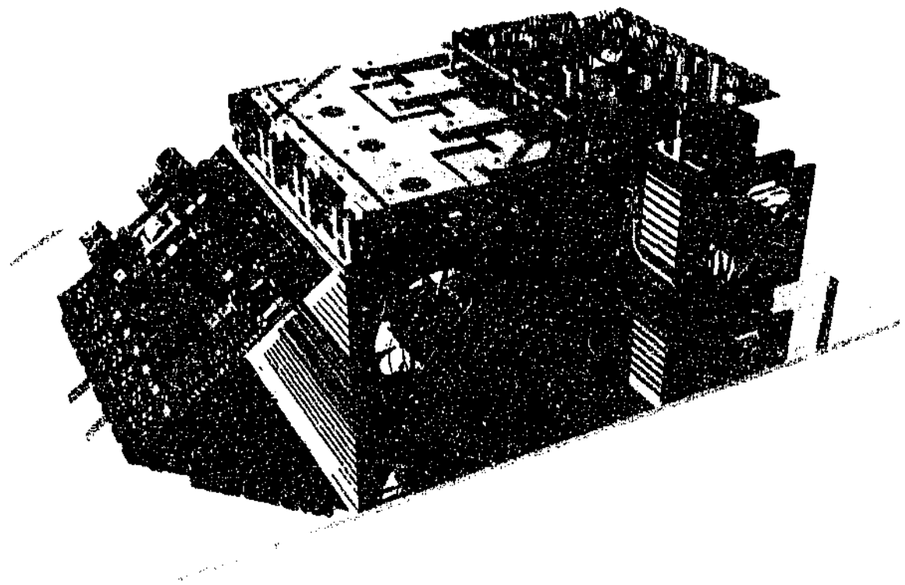


Figure 9.4: AutoCAD/3DMax rendering of the constructed back to back 30 kVA VSI.

inductance and capacitance values are given in Figure 9.1 and also in Appendix B (Table B.2). The inductors were constructed by Macro Power Pty. Ltd. and 400 V RIFA PHP 467R metalized polypropylene 30 μF capacitors were arranged to create the required filter capacitance.

9.2.3 Series Injection Transformers

The design of the series injection transformers required careful consideration of a variety of factors, including the transformer injection ratio, the saturation characteristic, zero-sequence flux path, and shielding of EMI effects to avoid them reflecting into the grid. Figure 9.5 shows the internal winding arrangement used for each of the three single-phase transformers (also seen at the top of the photo in Figure 9.6). The wiring connection between the transformers and the rest of the three-phase system is shown in Figure 9.1. The transformers were constructed to specification by Macro Power Pty. Ltd., and the nameplate and measured parameters of the transformers are provided in Appendix B (Table B.4).

The transformer ratio affects the sag compensation depth capability of the UCPC. Chapter 3 showed that a series injection capability of 0.5 p.u. can compensate for a high proportion of the expected sags occurring in a distribution system. Chapter 4 showed how the transformer

ratio N affected the damping of the LC filter, as well as the stability of the system. This effect is due to the fact that the system load parameters seen by the series converter will vary with N. For this experimental work a series voltage injection capability of 0.5 p.u. is used, but the transformer is designed to allow reconfiguration to 0.25 p.u. if required (Figure 9.5). This is done using two identical windings which can be connected either in series or parallel, whilst still maintaining the same output power (i.e. 20 kVA). The rated phase voltage from the converter is 240 V (i.e. 415 V line voltage), and the three transformers are connected in star (to allow for expansion to a four-wire system). Therefore, the output voltage is either 120 V or 60 V, depending on the chosen injection voltage.

Saturation is likely to occur during the large voltage injections that occur when compensating for a sag, if the series transformer is only designed for operation at rated conditions [99]. Depending where compensation occurs on the waveform, this may cause large magnetizing currents to be drawn from the VSI, which may in turn trip the UCPC from operation. Therefore the series transformer was designed to have a much higher saturation characteristic than is normally required by a power transformer. Even with this high saturation characteristic, a path for the non-linear portions of flux is required. Otherwise any small distortion will be directly transferred to the output grid voltage, which is highly undesirable. Therefore, the transformer also requires tertiary windings which are connected together in delta to convert the triplen harmonics in the flux to circulating currents in the delta windings.

Injection of EMI into the grid is also a significant problem for the series-shunt topology. The voltage level of the dc-bus, with respect to the grid voltage, will have a large dv/dt during each switching transition of the shunt VSI. The output of the series converter, with respect to the grid voltage, will then change due to switching transitions in both the shunt and series VSIs. The internal inter-winding capacitance of the series transformer will couple these high dv/dt changes back into the grid. To reduce this EMI injection into the grid, a new current path must be created which diverts these capacitive currents. This is achieved by using an earth shield between the primary and secondary windings, which is connected back to the grid side of the shunt converter. This encloses the EMI current loop within the back-to-back VSIs.

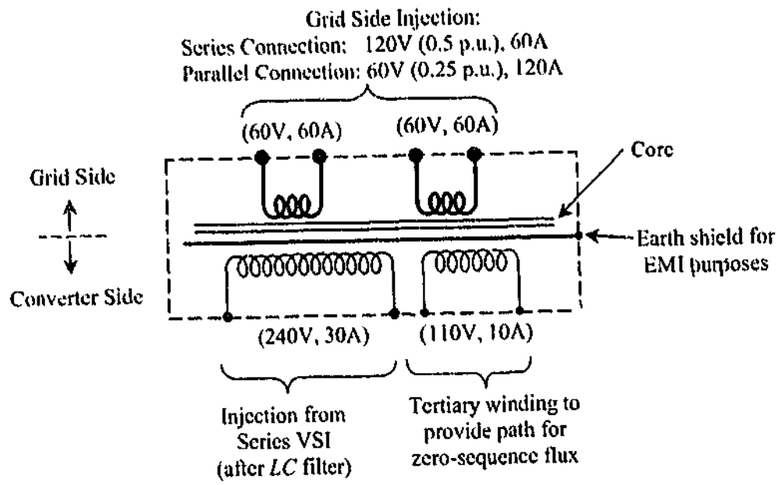


Figure 9.5: Winding arrangement schematic for one of the three identical single-phase series injection transformers.

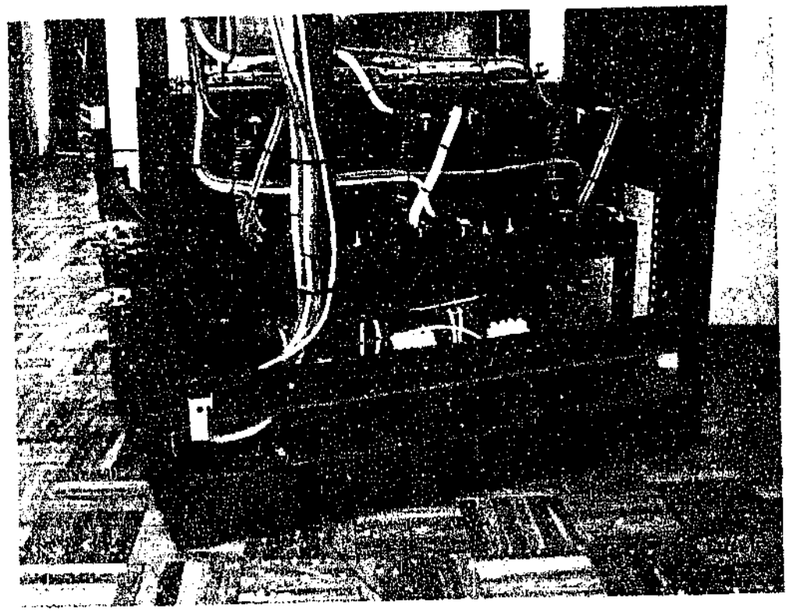


Figure 9.6: Photo of the UCPC magnetics. Above: Three separate single-phase multiple winding transformers. Below: Two three-phase sets of filter inductors.

9.2.4 Protection Components

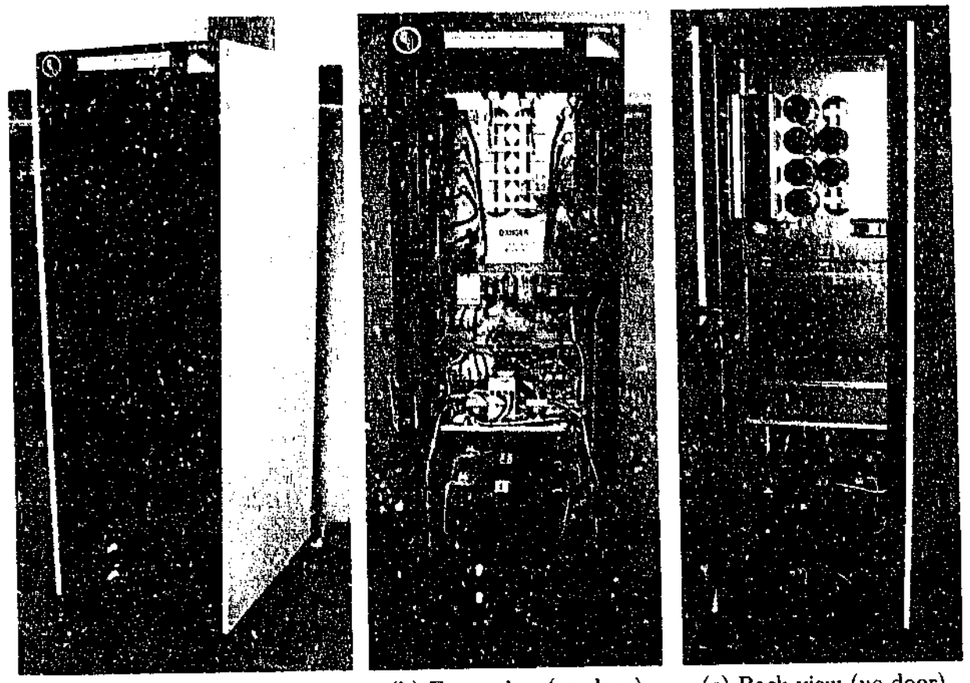
The protection elements were discussed in detail in Chapter 7. However, some of the components used to provide alternative current paths were upgraded to suit the increased power requirements of the final UCPC prototype. Firstly, the triacs were replaced with three sets of back-to-back thyristors. These EUPEC TT-105-N-16-LOF thyristor phase-leg modules are rated for 105 A in steady-state, and up to 2.2 kA under transient conditions. Therefore, it is permissible for the fault level at the load-side terminal point of the UCPC to increase up to 4 kA (for the 0.5 p.u. injection configuration used here). The thyristor gates were controlled using an existing six output thyristor gate driver board that was available.

The series bypass contactor was upgraded to a Ghisalba GH15 GN-04-240. This is a four-pole normally closed 415 V contactor rated for operation up to 63 A. An auxiliary contact block was also mounted on the contactor to provide the DSP with information relating to the state of the component. It should be noted that the choice and availability of normally closed contactors becomes increasingly difficult (and expensive) as the power rating increases, and custom designed units may be required for systems with only marginally higher ratings than were used in this work. (For the larger power system described in the following section, a normally open contactor protected by a UPS was used because of this difficulty.)

9.2.5 Housing and Construction

The low voltage UCPC prototype was housed in a standard 19" rack (Figure 9.7a). The enclosure is segregated into three sections. Firstly, the upper part is split front to back vertically into two sections, such that the air flow past the heatsinks (Figure 9.7c) is separate from all the electronics and control equipment (Figures 9.7b and 9.8). This maintains a forced air and clean air section for the heatsinks and electronics, respectively. The division was achieved using 2 mm aluminium to minimize weight and also because of the thermal properties of the material. Holes were cut into the dividers between each heatsink to mount the semiconductor devices on the opposite side to allow wiring access to these components. The third section of the enclosure mounts the magnetics at the base of the cubicle. This placement is primarily due to the weight of these components – especially the transformers. The three single-phase transformers weigh more than 200 kilograms combined, and therefore required the construction of a dedicated re-enforced mounting structure (Figure 9.6).

The majority of the wiring for the power components uses single core V90HT cables ranging from 1.5 mm^2 to paralleled 6 mm^2 cables, depending on the steady-state current load. Shielded



(a) 19" rack enclosure (b) Front view (no door) (c) Back view (no door)

Figure 9.7: Photo of the low voltage UCPC experimental test setup.

multi-strand signal cable was used to wire up the signal and control connections. The shields were earthed to a common point, at one end only, to minimize the susceptibility of the cables to radiated EMI. Additional shielding was also wrapped around the cables passing through the current transformer that was used to detect current drawn by the varistors. The large dv/dt transitions (discussed in Section 9.2.3) that can occur at the output of the series converter capacitively couple noise into the detection circuitry (which is referenced to earth). This was found to falsely trip the protection system until the shield was installed and earthed. Earthing the shield diverts these transient currents away from the detection circuitry.

For the prototype system a 240 V supply was provided via a plug at the front of the UCPC enclosure. This supply was used to power the two CS-IIB controller boards, the thyristor controllers, protection hardware controller, and the contactor coils. Note that if this supply was lost, the series system remains protected (even though control would be lost) because the coil to the normally closed bypass contactor would be de-energized. All powered components were connected to the same junction with a common earth point. There is only one earth connection within the housing. (It is commented that for a commercial system this arrangement would not

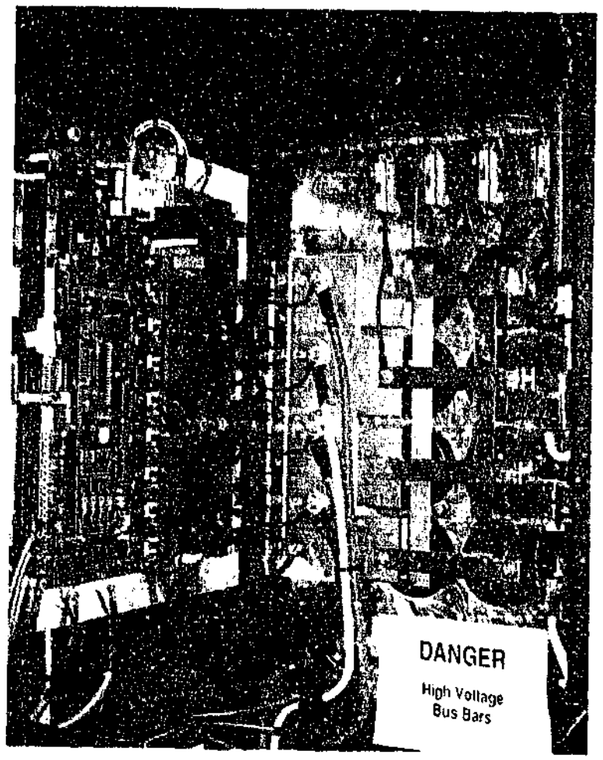


Figure 9.8: Photo of inside the UCPC prototype enclosure. Left: CS-IIB controller card used for series component. Center-Right: Busbars of the two VSIs.

be suitable as this supply cannot be guaranteed. Therefore, a small UPS supplied from the grid would need to be installed internally to the UCPC. This would be used as the auxiliary power supply, thus removing the need for any additional single-phase plug at the front of the unit.)

9.3 DSP Software Systems

The general DSP software structure for the series and shunt control systems is quite similar and therefore will be discussed together in the following sub-sections. The software is almost all written in C, and was developed and ported to the new controller hardware from open loop VSI software that was originally written by Andrew McIver. It is separated into two primary sections: the background software, and the interrupt software. The code for the series and shunt controllers can be found in Appendices D and E, respectively.

9.3.1 Background Software

The function of the background software is to control the controller's state machine and to handle the operator interface. The background code for the series and shunt converters is given in Appendices D.2 and E.2, respectively. The background software is executed in a continuous loop using the processing time that is remaining after the interrupt software has executed. Therefore, only software that is not real-time critical is placed in this section. All variables defined in the background code are stored in the (much larger) external memory of the CS-MiniDSP. Whilst the access time to the external memory is the same as for the internal memory (i.e. no wait states are used), use of the external memory does slow the processor somewhat as both the program and data memory cannot be accessed simultaneously (as they can when data is located internal to the DSP).

The state machine is used to handle the system settings during the different phases of operation (e.g. running, stopped, isolated, fault condition, etc.). For control of the series controller the state machine was presented and discussed in Chapter 7 (Figure 7.6). For the shunt converter, a similar state machine is used, but it excludes the triac opening state used for the protection of the series converter (the IGBT null states are also not used).

9.3.2 Interrupt Software

The interrupt software includes all the time critical components of the software including the main control system and DSP register setup, as well as other interrupts for synchronization timing and fault detection. The code is provided in Appendices D.1 and E.1 for the series and shunt controllers, respectively. No function calls are used in the controller software, with the exception of the digital filter assembler function, to minimize the overheads associated with such calls. Instead, duplicated code is generated using macro functions, which are expanded into full in-line code by the pre-processor during compilation. No local variables are used, so that for

each controller interrupt routine, variables do not need to be popped/pushed from/onto the stack. The makefile also ensures that all variables defined in the interrupt software are stored into the internal DSP memory. This gives all instructions in the interrupt software simultaneous access to both program and data memory.

The controller software is the major component of the interrupt software. This interrupt is called by the overflow/underflow events of the PWM up/down carrier reference counter, and therefore occurs once every half switching period. For the 5 kHz switching frequency used in this work, this equates to 10 kHz operation of this interrupt. This is also known as the sampling frequency. The absolute maximum execution time available for the interrupt is 100 μs, but a more realistic maximum value is closer to 70 μs, as spare processing time must be retained for the background software operation. The major parts of the controller software are: feed-back and feed-forward control systems, variable scaling and clamping, synchronization software, analog conversions, protection, and PWM compare register value creation.

Control Software

Delta based digital filters are used for the software implementation of the linear controllers in both the series and shunt systems. These were discussed in detail in Chapter 6. These digital filter controllers are written in assembler, and are provided in Appendices D.3 and E.3, for the series and shunt controllers, respectively. For the series controller, the digital filters implement the multiple P+Resonant harmonic controllers, and for the shunt controller they implement the reference signal extraction control block. These filters were written in assembler to take full advantage of the DSP instructions that are designed specifically for use with digital filtering. All the controllers are placed into a single function call to minimize overheads, and the parameters are passed to this function via an integer array pointer. For the shunt controller an inner current regulator is also required, and the PCR scheme detailed in Chapter 4 is used. The PCR controller was adapted from a version written by Andrew McIver for a different controller card (and system parameters) that did not include reference prediction.

Variable Scaling and Clamping

Due to the 16-bit fixed-point capability of the DSP a large portion of the software design and implementation time is dedicated to management of the scaling of the variables to both minimize numerical truncation errors and to protect the variables from overflowing. 32-bit operation is possible on the DSP, but each instruction executes orders of magnitude slower, and therefore

this alternative was not a feasible option for this work. All signed variables are restricted to the range of $\pm 2^{15}$, and only the high byte of the 32-bit resultant from the multiplication of two 16-bit integers is typically used.

To stop variable overflow, either the scaling is designed such that overflow is not possible, or where this is not viable, clamping is included. Clamping is also applied to various signals in the controllers to ensure the demanded values are maintained within the rated limits of the external components. Examples include clamping of the shunt current reference and the series voltage reference (both magnitude and rate of change limiting).

Synchronization Software

Synchronization to the supply fundamental voltage is implemented using software triggered by the zero-crossing hardware discussed previously in this chapter. At each positive zero-crossing of the supply voltage, the hardware triggers an interrupt, which captures the time of the crossing event. During each sampling interrupt the presence of this crossing is detected and the crossing time is used to determine any phase error, which is then slowly nudged in the correct direction to reduce the error. If phase error remains from zero crossing to zero crossing, the frequency is also nudged towards a new value to reduce the error. The slew rate of the system is easily controlled by the step size of this frequency tracking. For the series converter the slew rate is intentionally kept slow to minimize any phase jump effects at the load bus. Double zero crossing events (e.g. due to severe harmonics or notching) are discarded by using both hardware, and software lock-out regions.

Analog Conversion and Code Timing

A 10-bit analog to digital (A/D) conversion sub-system is built into the DSP, with 16 input channels multiplexed through two A/D converters. The conversion time for each set of two analog conversions is 6.6 μs. The first set of conversions is started automatically by the start of the controller interrupt, and since the calculations of the controller cannot start until these conversions are finished, code not associated with these measurements must be executed during this period. For both the series and shunt systems this period is used by the synchronization software and the digital filter inverse delta operations. The succeeding sets of analog measurement conversions are then started manually in turn as the interrupt code proceeds, and are arranged to shadow executing analog conversions. Hence a new conversion cycle will start immediately the present one finishes, with the results being stored into a 2 deep FIFO buffer. The reliance of

the system on the various analog measurements means that the structure of the interrupt code is primarily designed around the analog conversions to maximize the processor usage throughout the interrupt code execution.

9.4 Test Setup and Measurement

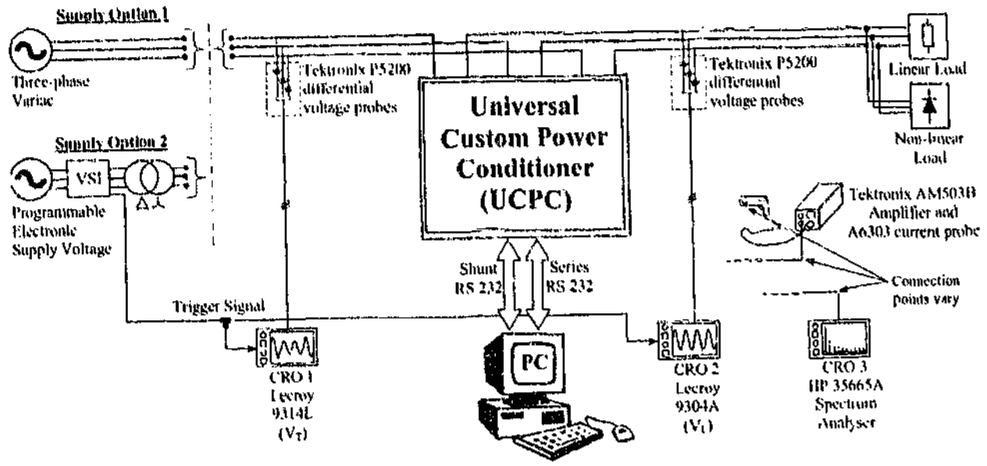
The experimental test setup is shown in Figure 9.9. The components of interest are the supply voltage, the loads, the UCPC, and the test measurement devices.

A supply voltage capable of producing sags, swells, distortion, unbalance and flicker was required to test the UCPC. To achieve this a three-phase programmable supply was designed, constructed and programmed in conjunction with Andrew McIver and Erika Twining to generate these waveforms. This supply was coupled to the system via an isolation transformer. The interface to the programmable supply is via both the front panel and a connected PC (using a RS-232 interface).

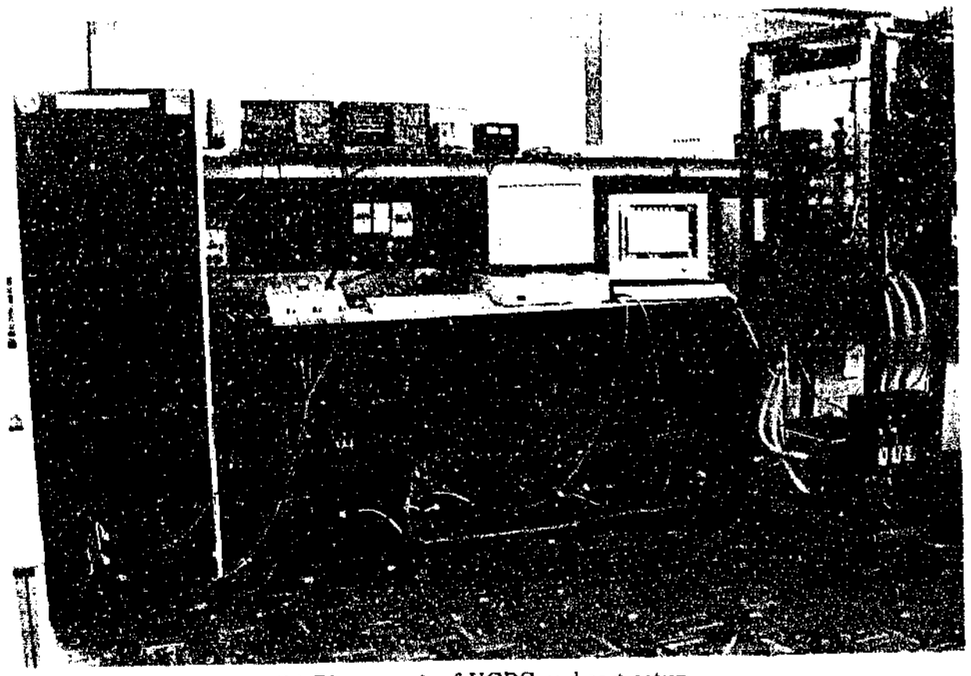
Both linear, non-linear, and combined loads were tested. The linear loads consisted of either a purely resistive load, or a resistive-inductive series load combination. The non-linear load was a diode rectifier with a parallel resistive-capacitive dc load, with series ac line inductors between the load and the UCPC. The particular load used is noted with each test in the following section.

Two four-channel Lecroy (9314L and 9304A) 1 MS/s digital oscilloscopes (CROs) were used to simultaneously capture the three-phase voltage supply and load waveforms. A Hewlett Packard HP 35665A dynamic signal analyzer was also connected to the load and supply voltages (across the *ab* line voltage). Voltages were measured using six Tektronix P5200 high voltage (1 kV) differential probes. Currents were measured using a Tektronix A6303 current probe with an AM503B current amplifier. When required, measurement data was also recorded from the DSP variables, using either data storage internal to the MiniDSP (and collected via the RS-232 interface), or via an external D/A card (interfaced to the CS-HB Mini-bus) connected to a CRO channel.

An isolated trigger signal output was generated from the programmable supply, and connected to both CROs. This was used as a simultaneous trigger that could either be activated by the supply's keyboard interface, or at the start of a generated sag. A contactor controller unit was also constructed to allow timed pulsed operation of an external contactor. This unit was used in Chapter 7 to create the momentary short circuit used to test the protection circuit, and was also used here to switch loads in and out using its manual setting function.



(a) UCPC and test circuit schematic.



(b) Photograph of UCPC and test setup.

Figure 9.9: Experimental test configuration for the low voltage UCPC prototype. (a) Circuit Schematic, and (b) Photograph.

9.5 Experimental Results

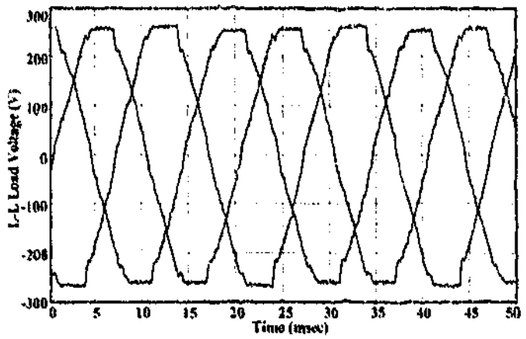
This section presents further results from the low voltage experimental UCPC system. The aim is to verify the compensation capability of the UCPC for Power Quality events of types 2 through 6 (Chapter 2). The results are broken up into steady-state voltage compensation (i.e. series), transient voltage compensation, and current compensation (i.e. shunt). The UCPC has been tested under a wide range of combined Power Quality conditions, but due to the large number of possible combinations not all results can be shown here. Therefore, only selected results from these combinations are presented. All tests were conducted at a nominal fundamental line voltage of 190 V rms, and load currents are noted with each test.

Throughout these results, supply options 1 and 2 (Figure 9.9a) are used. Where possible a mains supply through a variac is used (option 1) as this provides a lower source impedance than the programmable supply (option 2). However, for voltage sags, flicker, and large unbalance conditions this supply is not suitable, and the programmable supply (option 2) must be used. Unfortunately, also the variac supply is relatively weak, which is not an ideal operating condition for the PCR current regulation scheme that was used (as discussed in Chapter 4). Chapter 4 showed that the series control system (which is the primary focus in this thesis) is relatively unaffected by the source impedance of the supply, but if the shunt section of the UCPC resonates with the supply, this will in turn affect the series section. To combat this problem gains in the series controller were slightly lowered during the tests that were conducted using the programmable supply, and as a consequence a small steady-state error sometimes exists. This error is not seen in the simulation results shown in Chapter 8, where a more realistic supply impedance was able to be used).

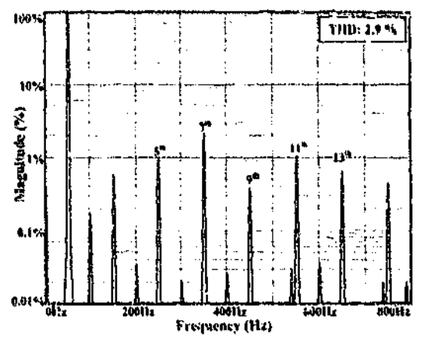
9.5.1 Steady State Voltage Compensation

The first test verifies the steady-state voltage harmonic compensation operation of the series portion of the UCPC, and uses supply option 1 (Figure 9.9a). Voltage distortion is generated by the insertion of a non-linear load directly onto the source terminals (i.e. upstream of the UCPC). A combined linear/non-linear load is also applied downstream of the UCPC, and the currents are active filtered by the shunt portion of the UCPC. These current results are presented later in Section 9.5.3 (Figure 9.19).

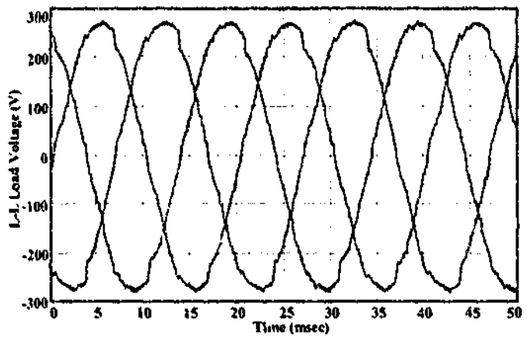
Figure 9.10 presents the waveform and frequency spectrum for the supply and load voltages in Test 1, and the numerical harmonic results are given in Table 9.1. The load voltage regulation is shown for the case with only feed-forward compensation, and also with both feed-forward and



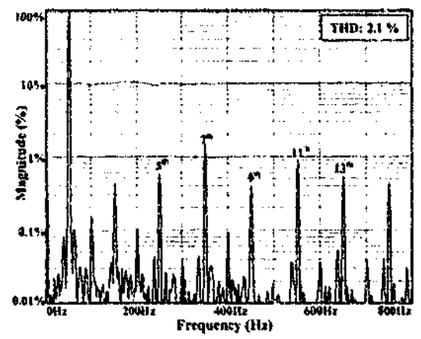
(a) Supply voltage (V_T)



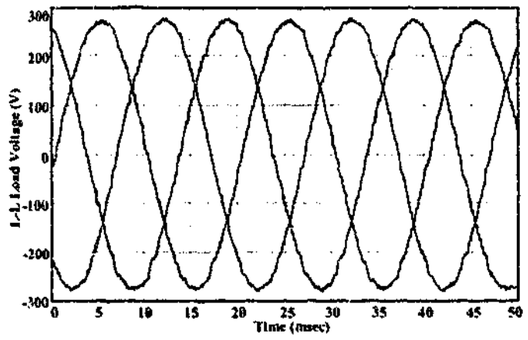
(b) Supply voltage (V_T)



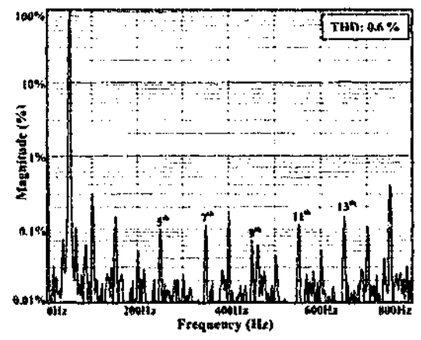
Feed-forward only; (c) Load voltage (V_L),



(d) Load voltage (V_L).



Feed-forward and feed-back; (e) Load voltage (V_L),



(f) Load voltage (V_L).

Figure 9.10: Test 1 - Experimental L-L source and load voltages: (a),(c),(e) Time waveform; (b),(d),(f) Frequency spectrum.

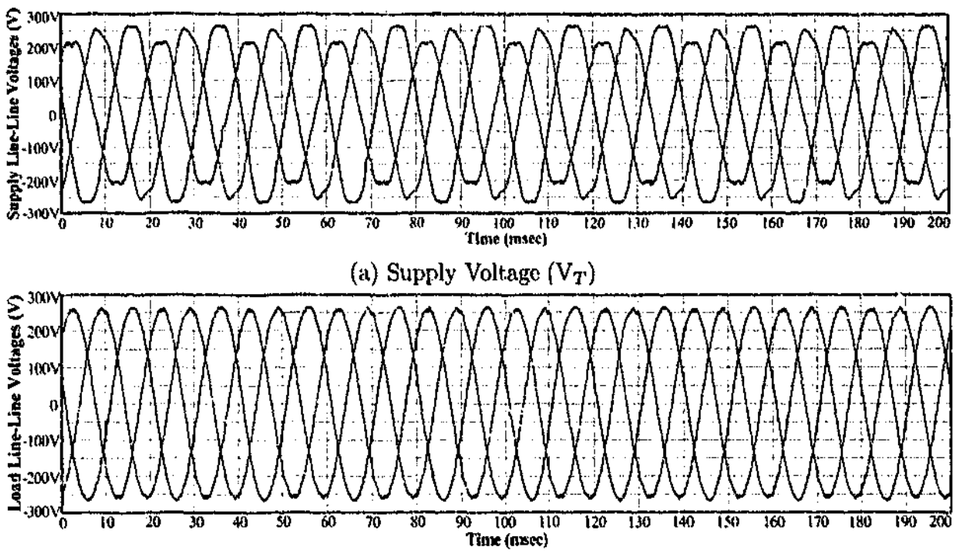
Compensation Type	5^{th}	7^{th}	9^{th}	11^{th}	13^{th}	THD
None	1.0%	2.1%	0.4%	1.1%	0.6%	2.9%
Feed-forward only	0.6%	1.6%	0.4%	0.9%	0.5%	2.1%
Feed-forward and Resonant Feedback	0.1%	0.12%	0.07%	0.13%	0.15%	0.6%

Table 9.1: Load voltage harmonics before and after series injection

selective feed-back compensation. The supply voltage contains a Total Harmonic Distortion (THD) of 2.9% (Figures 9.10a-b). The use of only feed-forward compensation reduces the load voltage THD to 2.1% (Figures 9.10c-d). Note that the only significant reduction is to the lower order 5^{th} and 7^{th} harmonics, because the higher frequencies have a larger phase error which is caused by the sample delay (see Figure 4.13 in Chapter 4). With the resonant harmonic controllers added (Figures 9.10c-d) each of the selected harmonics in the load voltage is reduced to approximately 0.1% (Table 9.1), and a final THD of only 0.6% is obtained. Furthermore, this THD is now dominated by the 15^{th} and 17^{th} harmonics that have not been selected for compensation. These harmonics could also be compensated to further reduce the load voltage THD if required.

The second test shows the combined compensation of voltage harmonics and unbalance, and is conducted using supply option 2 (with a 10 A linear load). The voltage harmonics and unbalance are created by the programmable supply. Figure 9.11 shows the supply and load waveforms of the test.

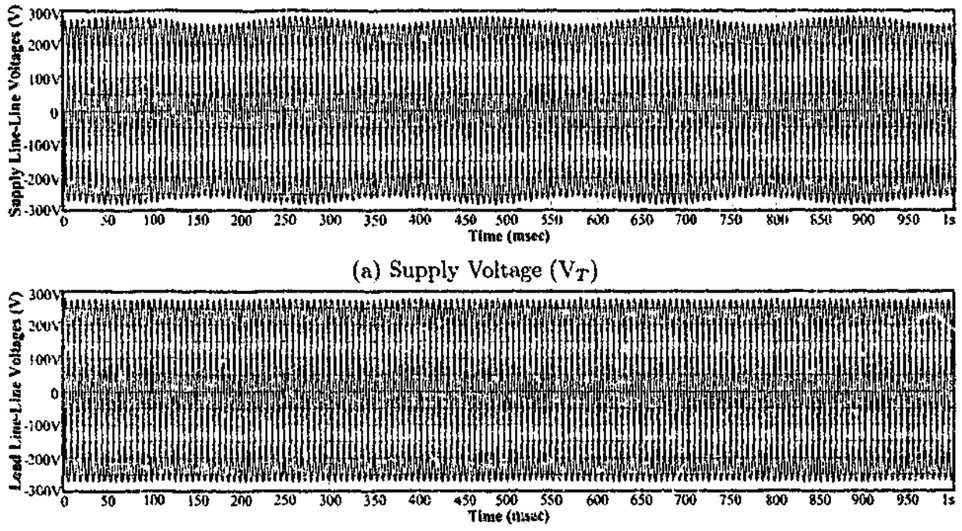
The third test verifies the flicker compensation of the UCPC. The programmable supply of option 2 is used to create a 5 Hz 10% flicker component modulated onto the fundamental supply voltage. After compensation by the UCPC, the flicker is clearly seen to be removed from the load voltage (Figure 9.12).



(a) Supply Voltage (V_T)

(b) Load Voltage (V_L)

Figure 9.11: Test 2 - Experimental results of the UCPC compensating for a supply voltage with harmonics and an unbalanced fundamental component.



(a) Supply Voltage (V_T)

(b) Load Voltage (V_L)

Figure 9.12: Test 3 - Experimental results of the UCPC compensating for a supply voltage with a 5 Hz 0.1 p.u. flicker component.

9.5.2 Transient Voltage Compensation

The primary purpose of this section is to verify the operation of the series portion of the UCPC under transient conditions. For all tests in this section, the programmable supply (option 2 in Figure 9.9a) was used with a linear load (10 A).

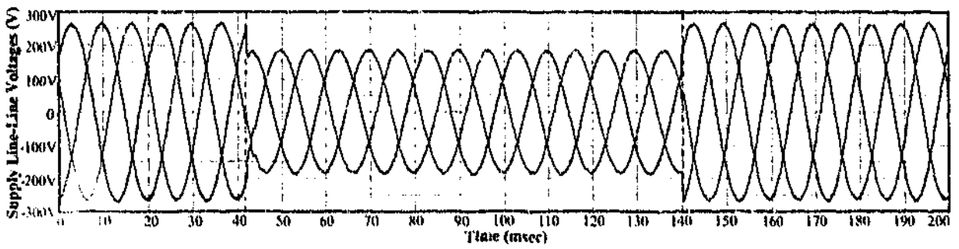
The fourth test (Figure 9.13) shows the transient response of the UCPC to a 0.7 p.u. symmetrical sag. These results show that the load voltage magnitude is maintained throughout the sag, and only small deviations can be seen at the start and end of the event. These small deviations are due to the delay between the time that the change in condition is sampled by the controller, and the time that the controller implements the required change in the next switching cycle.

The fifth test (Figure 9.14) verifies the operation of the UCPC under a 1.2 p.u. swell. Under this condition excess power from the voltage swell is injected back into the grid via the bi-directional shunt connection.

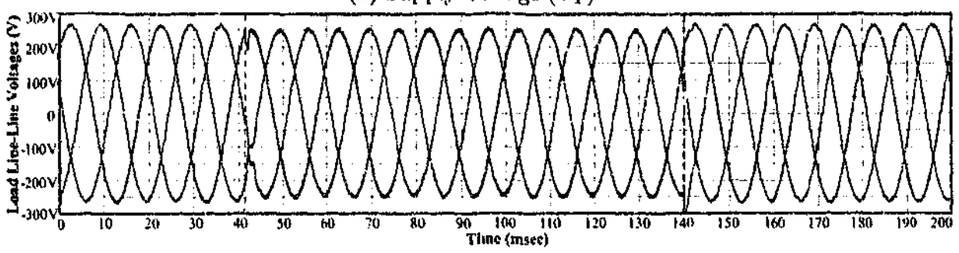
The sixth test (Figure 9.15) shows a 0.8 p.u. symmetrical sag with voltage harmonics introduced. The load voltage magnitude remains constant during the event, and the voltage harmonics are compensated during the sag. A small oscillation can be seen at the start and end of the sag, and this is attributed to the transient response of the chosen resonant harmonic controller with an f_c of 1.5 Hz. This value provides a fast transient response, but has some overshoot for large steps in the reference signal. (Refer to the medium voltage experimental results in Chapter 10 for a comparison of the transient response with varying values of f_c).

The seventh test (Figure 9.16) investigated the compensation by the UCPC of an asymmetrical sag which also contains harmonics. Once again the load voltage magnitude on all phases remains nearly constant, and the introduced voltage harmonics are also adequately compensated.

The eighth test shows compensation of a combination of voltage conditions including unbalance (0.9 p.u.), flicker (0.1 p.u. at 5 Hz), harmonics (3.3% THD), and a sag (0.8 p.u.). Figure 9.17 presents the supply and load waveforms over a period of half a second, and Figure 9.18 provides a closer view of the same results during the sag event. Once again a small transient is seen at the edges of the sag event, but otherwise the load voltage is kept at a constant sinusoidal magnitude and phase. The load voltage THD is reduced from 3.3% down to less than 0.7%, with the 2.6% 5^{th} harmonic reduced to 0.2%, and the 1.2% 7^{th} reduced to 0.1%.

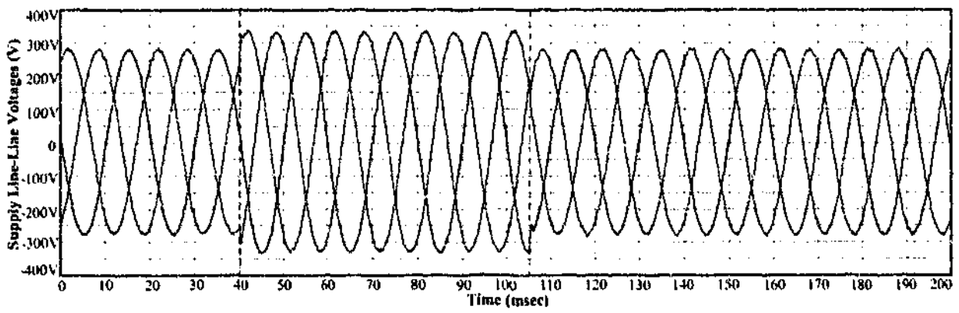


(a) Supply Voltage (V_T)

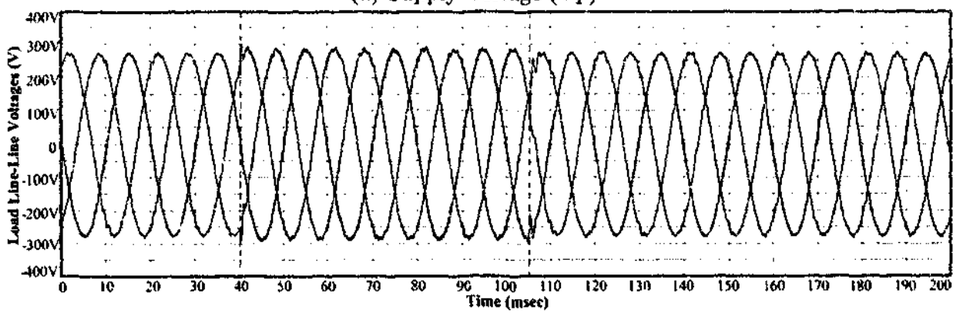


(b) Load Voltage (V_L)

Figure 9.13: Test 4 - Experimental results of the UCPC compensating for a 0.7 p.u. symmetrical sag.

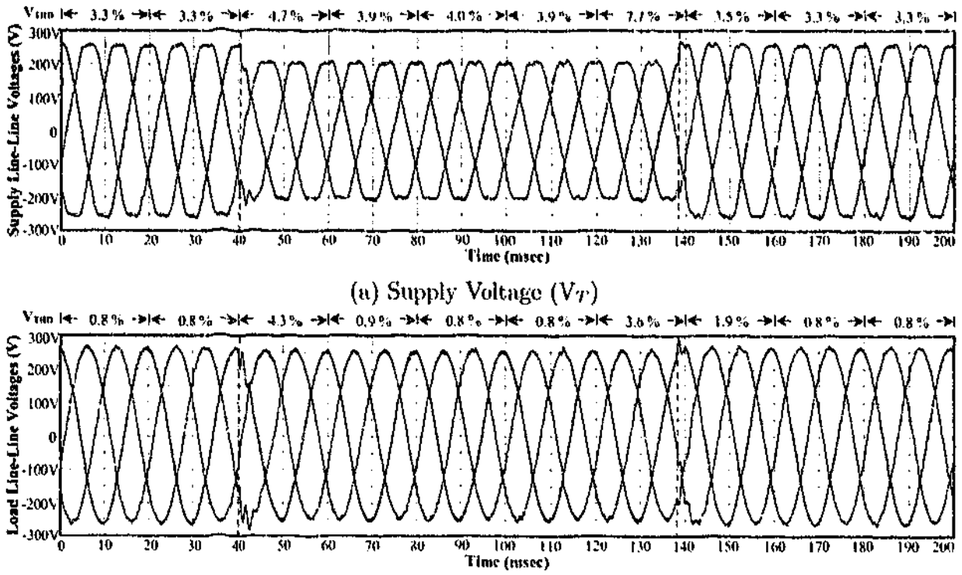


(a) Supply Voltage (V_T)



(b) Load Voltage (V_L)

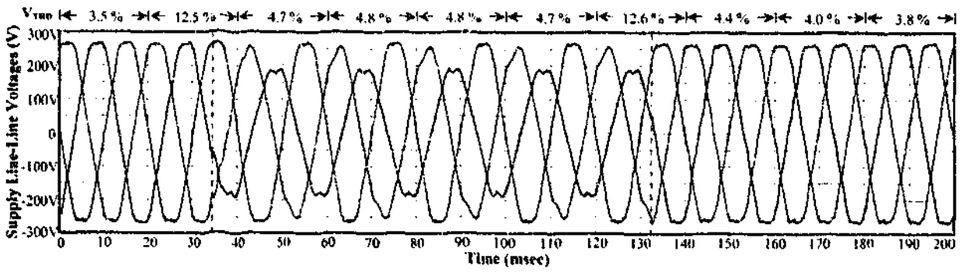
Figure 9.14: Test 5 - Experimental results of the UCPC compensating for a 1.2 p.u. symmetrical swell.



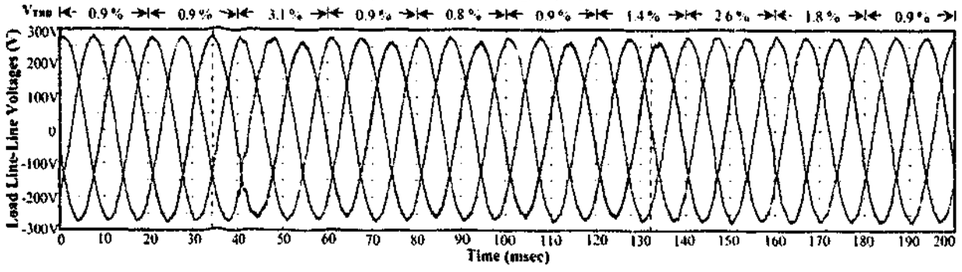
(a) Supply Voltage (V_T)

(b) Load Voltage (V_L)

Figure 9.15: Test 6 - Experimental results of the UCPC compensating for a 0.8 p.u. symmetrical sag with harmonics.

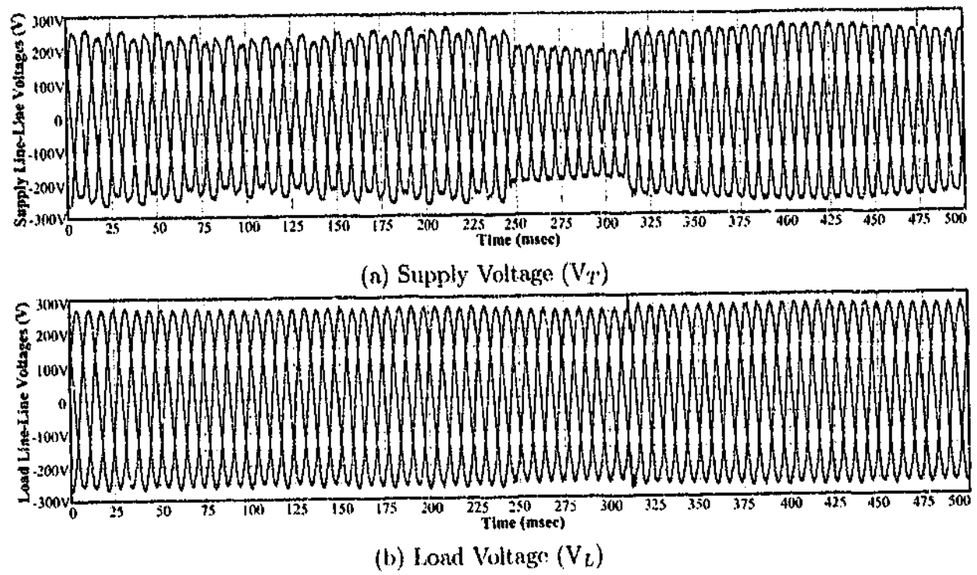


(a) Supply Voltage (V_T)



(b) Load Voltage (V_L)

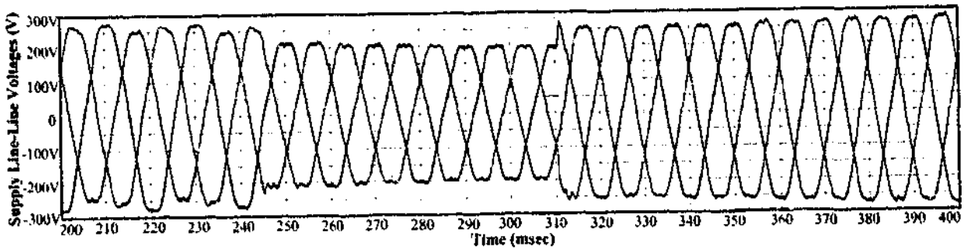
Figure 9.16: Test 7 - Experimental results of the UCPC simultaneously compensating for voltage harmonics, 0.1 p.u. 5 Hz flicker, 0.9 p.u. unbalance, and a 0.8 p.u. sag.



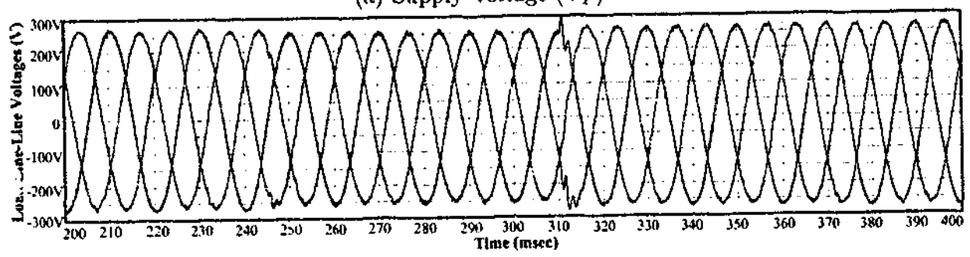
(a) Supply Voltage (V_T)

(b) Load Voltage (V_L)

Figure 9.17: Test 8 - Experimental results of the UCPC simultaneously compensating for voltage harmonics, 0.1 p.u. 5 Hz flicker, 0.9 p.u. unbalance, and a 0.8 p.u. sag.



(a) Supply Voltage (V_T)



(b) Load Voltage (V_L)

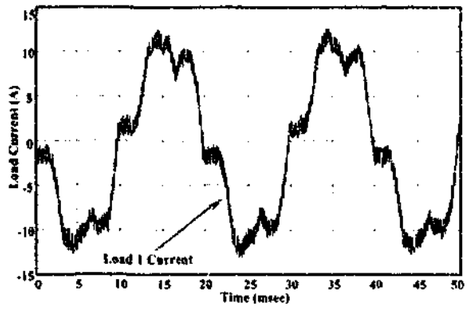
Figure 9.18: Test 8 - Experimental results of the UCPC simultaneously compensating for voltage harmonics, 0.1 p.u. 5 Hz flicker, 0.9 p.u. unbalance, and a 0.8 p.u. sag (Zoomed view of Figure 9.17).

9.5.3 Current Compensation

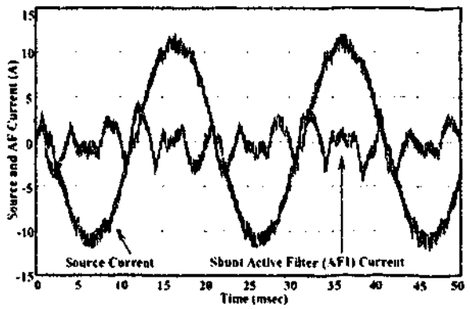
This section presents the current compensation results for the shunt portion of the UCPC, and outlines both the current harmonic and balancing capabilities of the system. All tests were conducted using supply option 1 (Figure 9.9a).

Figure 9.19 presents the current harmonic compensation results obtained during Test 1 presented earlier in this section (Figure 9.10). The highly distorted load current shown in Figure 9.19a was measured by the shunt controller, and the injected current and resulting sinusoidal supply currents are shown in Figure 9.19b. (Note that only one of the three phase currents is shown to maintain the clarity of the figure).

The ninth test was conducted with only the shunt portion of the UCPC in operation, and was used to investigate the practical outcomes of the StatRF controller presented in Chapter 5, compared to the traditional Synchronous Reference Frame (SRF) signal extraction method. Both the three-wire StatRF and SRF signal extraction methods were implemented and the results are shown in Figures 9.20a and 9.20b, respectively. The StatRF presented requires no sine table storage for synchronous frame transformation, and is computationally quicker than its equivalent synchronous counterpart. However, without the use of the delta operator, it was quite difficult to implement this scheme properly using conventional shift-based IIR filters. Therefore, no shift IIR based StatRF implementation results are included here for comparison. The total execution time for the four stationary delta filters of the three-wire StatRF was 6 μs, whereas an equivalent synchronous reference frame implementation took 13.5 μs to execute. If only harmonics are required to be extracted (i.e. an unbalance component is not required), the single-phase version of the StatRF can be used, which will provide a further 50% reduction in computational time.

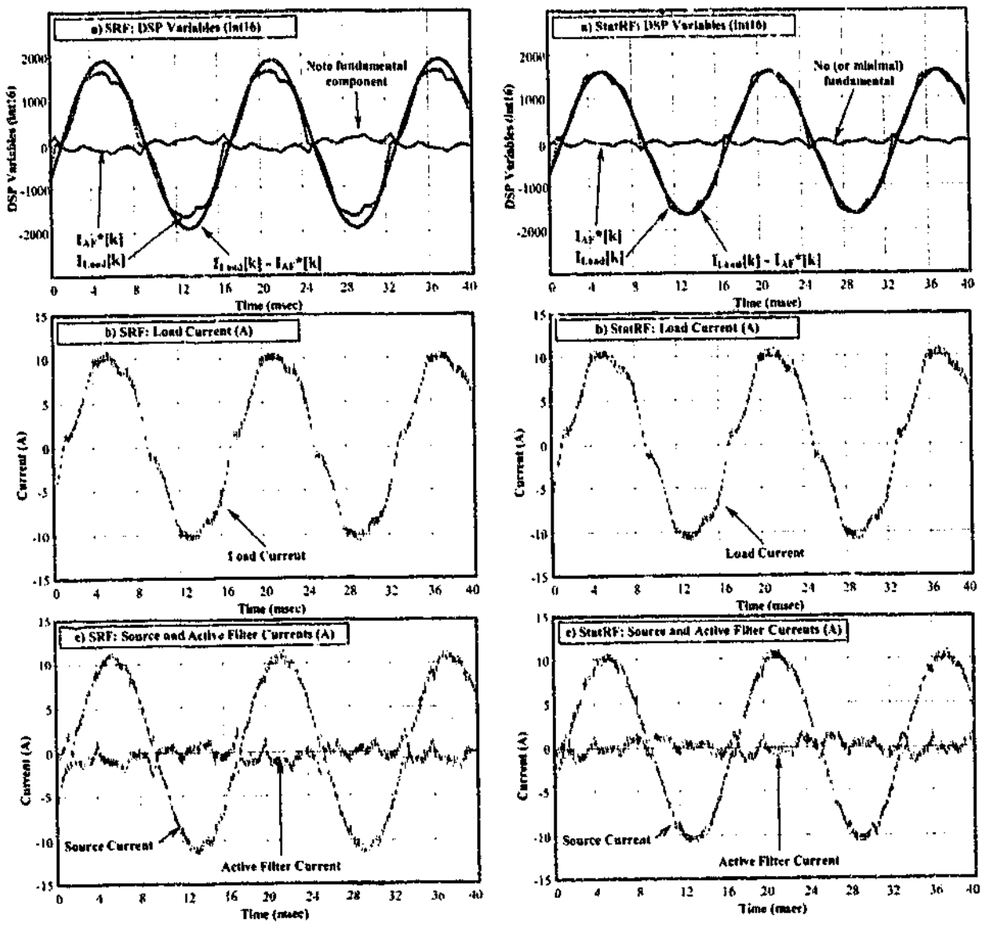


(a) Load current



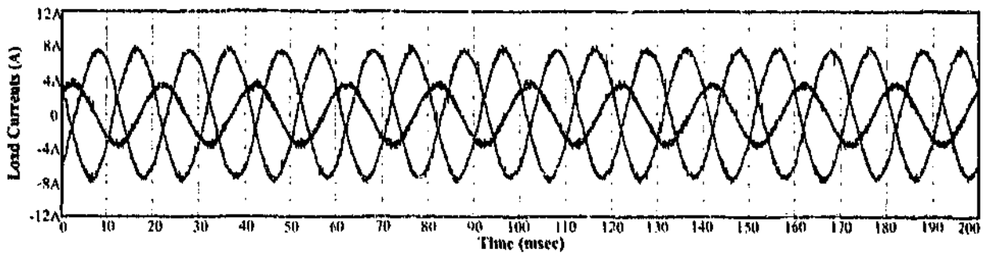
(b) LSupply and shunt active filter currents

Figure 9.19: Test 1 - Experimental source, load and active filter current (a),(c)waveforms and (b),(d) frequency spectrums.

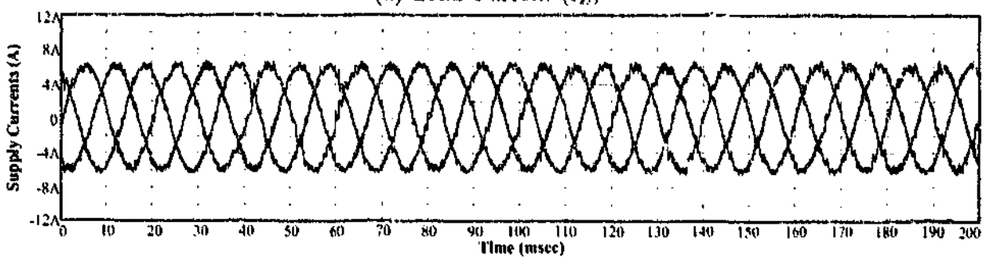


(a) (b)

Figure 9.20: Test 9 - Comparison SRF and StatRF harmonic extraction implementations.



(a) Load Current (I_L)



(b) Supply Current (I_S)

Figure 9.21: Test 10 - Experimental results of the UCPC balancing the supply current due to unbalanced load currents.

For the 16-bit fixed-point implementation it was difficult to achieve more than 30-dB attenuation in the SRF generator. This was found to be due to dc offsets created from integer truncation in the high-pass filter, and also due to the sine/cosine phase index quantization. The outcome of this lack of attenuation is clearly seen in the fundamental component still left in the active filter reference current (I_{AF}*) in Figure 9.20a. Note that this extra fundamental component is not as dominant in the final active filter current shown in Figure 9.20b, which is due to the interaction of the PI dc bus controller with this component. For the StatRF implementation, Figure 9.20b shows very little fundamental component remaining, because of the accuracy of the delta operator based digital filter used, and the lack of errors introduced by the use of a sine lookup table.

The ninth experimental test (Figure 9.21) shows the balancing capabilities of the UCPC. The results show that load current unbalance is greatly reduced in the supply current by the shunt portion of the UCPC. However, as mentioned previously, the soft nature of the supply results in a small additional resonance which can be seen in the supply current.

9.6 Summary

This chapter has described the hardware and software systems that were developed to build the low voltage UCPC experimental prototype. The hardware development involved the design and construction of dual back-to-back voltage source inverters, series protection systems, series transformer, as well as a hybrid digital/analog controller board to control the converters. The DSP software was developed for both the series and shunt converters, and required the development of control, protection and interface software. The final UCPC prototype is housed in a standard 19" rack for convenience.

To verify the operation of the UCPC control systems proposed in this research, the prototype was tested under all of the required Power Quality conditions, as well as with combinations of these phenomena. The voltage conditions tested included fundamental magnitude variations, harmonics, unbalance, flicker, swells, symmetrical sags, asymmetrical sags. Current conditions tested included harmonics and unbalance. The results showed that the UCPC compensated for all of these Power Quality conditions almost exactly as predicted by the design and simulations already presented. The following chapter presents further results describing the operation of the series control system on a 10 kV medium voltage system.

Chapter 10

Medium Voltage Experimental Verification

[1]Chapter 9 has presented low voltage (LV) experimental verification of the entire UCPC system. This chapter extends this work and presents medium voltage (MV) experimental verification for the series control system (which has been the dominant focus in this research work). The experimental testing was conducted using a prototype medium voltage (10 kV) Dynamic Voltage Restorer (DVR) in collaboration with Aalborg University, Denmark.

Whilst the primary objective of this testing was to verify the series control system under medium voltage conditions, this chapter also presents a further application for the proposed series controller – a DVR that incorporates voltage harmonic compensation. DVRs are now becoming more established in industry as a method of reducing the impact of voltage sags on sensitive loads. However, DVRs mostly spend most of their time in standby mode, since voltage sags occur very infrequently, and hence their utilization is low. In principle, it would be advantageous if the series connected converter of a DVR could also be used to compensate for load voltage harmonics (similar to the UCPC), with minimal alterations to existing hardware.

This chapter applies the series selective voltage compensation scheme of the UCPC to a DVR, to provide voltage harmonic compensation with minimal effect on the sag compensation capability of the basic DVR. The system was designed using the simulation models developed in Chapter 4, and its experimental performance was verified under a range of conditions, including distorted supply voltages, non-linear loads, and operation during distorted voltage sags.

[1]The material in this chapter is also published in part as:
M.J. Newman, D.G. Holmes, J.G. Nielsen, F. Blaabjerg, "A Dynamic Voltage Restorer (DVR) with Selective Harmonic Compensation at Medium Voltage Level," *in Conf. Rec. IEEE/IAS Annual Meeting*, Salt Lake City, UT, USA, 2003. (In Print)

10.1 Overview and Objectives

The medium voltage DVR prototype used for the work in this chapter was developed by Dr. John Godsk Nielsen, under the supervision of Prof. Frede Blaabjerg. The experimental testing was completed on site in Aalborg, Denmark, during a stay at Aalborg University as part of this research. I would once again like to thank Prof. Blaabjerg for making this collaboration possible. Details of the existing system are given in Section 10.2, and further hardware details are reported in the related Ph.D. thesis [60] by Dr. Nielsen. Only the selective harmonic controllers were added to the DVR (i.e. no feed-forward or proportional control) so that near zero net real power flow requirement of the series-only topology (Figure 10.1) could be preserved in steady-state. Details of the additional controller design and verification are presented in Section 10.3.

This work serves a variety of objectives in relation to the theory presented in this thesis. These objectives are:

1. To verify the series harmonic voltage compensation scheme on a medium voltage system. The results show that the medium voltage experimental parameters of the transformers, measurement sensors, etc., allow for the same high level of compensation performance achieved for voltage harmonics in the low voltage system;

2. To present a further application for the series control scheme;

3. To demonstrate the capability of the system controller to operate at significantly lower sample frequencies;

4. To experimentally verify the use of delta IIR digital filters on a floating-point processor;

5. To provide an experimental example of the single-phase operation of the series control scheme (since the three-phase DVR is constructed using three individual single-phase H-bridges acting on each phase of the system independently).

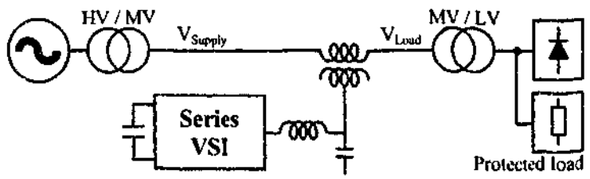


Figure 10.1: Series topology of the Dynamic Voltage Restorer (DVR).

10.2 Medium Voltage DVR Prototype

This section outlines the experimental details of the MV DVR including the physical parameters, test measurement setup, existing controller setup, as well as initial testing results and alterations that were made to the system before the addition of the harmonic control scheme.

10.2.1 DVR System Overview

Table 10.1 provides the specifications for the 10 kV DVR platform used for this work (refer to [60] for full details), with the schematic of the test setup shown in Figure 10.2. The 200 kVA converter is made up of three individual H-bridges (i.e. a six-phase-leg voltage source converter), and is constructed from a three-phase UPS donated by APC-Denmark (i.e. three phase legs for the active rectifier and the other three for the UPS output). The dc-bus of the converter is charged to 600 V using a 3 kW unidirectional dc power supply (two 300 V_{dc} supplies in series, with current limiting to reduce surge currents on start-up). This provides 4.7 kJ of energy (at 600 V) in the 26 mF of electrolytic dc-bus capacitors, for transient use by the DVR during a sag ride-through event.

Three specially made 67 kVA 0.29 kV/2.9 kV single-phase series injection transformers are installed in series on each phase to connect the LV converter to the MV test grid. This provides a maximum injection capability of 200 kVA at 5 kV line-line, and the system can therefore protect loads of up to 400 kVA.

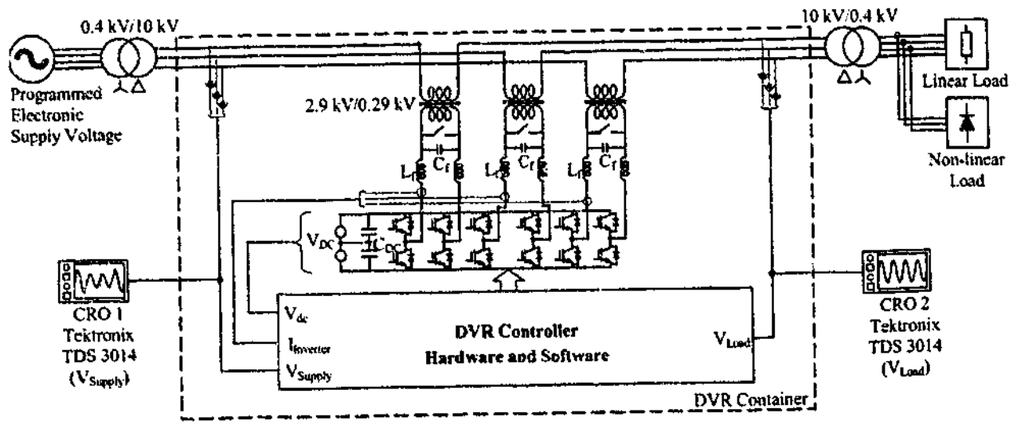


Figure 10.2: Experimental DVR test system and measurement setup.

Parameter	Value
Nominal grid voltage (l-l)	$V_{Supply} = 10$ kV
Nominal load loltage (l-l)	$V_{Load} = 380$ V
Maximum series voltage injection (l-l)	$V_{DVR} = 5$ kV
Maximum series power injection	$S_{DVR} = 200$ kVA
DC charger maximum power rating	$P_{DC} = 3$ kW
Switching frequency	$f_{SW} = 3.0$ kHz & 5.0 kHz
Sampling frequency	$f_S = 3.0$ kHz & 5.0 kHz
Maximum converter dc-bus voltage	$V_{DC} = 600$ V
Capacitance of the dc-bus	$C_{DC} = 26$ mF
Filter inductance	$L_F = 260$ μH
Filter capacitance	$C_F = 120$ μF
Transformer leakage inductance (referred to LV side)	$L_{Leakage} = 201$ μH

Table 10.1: Main specifications of the experimental DVR.

The series protection scheme used was similar to the scheme presented in Chapter 7[2], but at the time the tests were undertaken, the thyristor protection current path was not yet complete. To minimize cost, the bypass contactor was located on the LV side of the series injection transformers, and a normally open contactor was used (instead of the preferred normally closed) which therefore had to be protected by a low rated UPS to ensure its ability to bypass the system under all conditions.

10.2.2 Medium Voltage Test Grid

The medium voltage grid was fed from a low voltage supply that was stepped up to a medium voltage level for connection to the DVR. The supply voltage profile could be arbitrarily created using a California Instruments (iX Series) 30 kVA programmable ac power supply. The programmable supply was used to create harmonic distortion and symmetrical/asymmetrical voltage sags at a line voltage of 380 V, before stepping up to a line voltage of 10 kV (i.e. 5.8 kV phase voltage) using a 50 kVA star-delta distribution transformer (seen in Figures 10.2 and

[2]The initial publication [7] associated with the series protection work in Chapter 7 is a primary reference in the relevant section of the PhD Thesis describing the construction of the MV DVR[60].

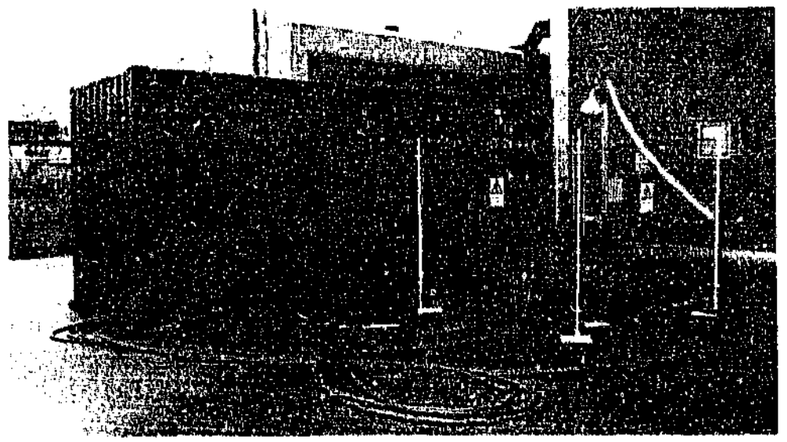


Figure 10.3: Photo of the DVR container and the 10 kV/0.4 kV distribution tranformers for the supply and load. Location: Aalborg University, Denmark.

10.3). After the series connection of the DVR at the medium voltage level, the load voltage was stepped down using another distribution transformer to allow low voltage linear and non-linear loads to be connected.

The non-linear load was a diode rectifier with a parallel resistive/capacitive dc load. All tests were conducted at low power (a few kVA) due to limitations of the programmable power supply at the time of the tests. Whilst the majority of the tests were undertaken with a line voltage of 10 kV, tests containing the non-linear loads with high current peaks were conducted at only 5 kV, once again due to limitations of the supply.

A 20 foot shipping container (Figure 10.3) was used to house the medium voltage DVR system (including injection transformers, converter and controller). This allowed relocation of the system for off site testing. The DVR container and all other medium voltage components (i.e. cabling and distribution transformers) were all located externally in the high voltage yard. The low voltage supply and load, as well as the user interface and oscilloscopes, were located inside the primary building for safety and other practical reasons (including the Danish winter weather conditions).

10.2.3 Measurement Systems

The primary system variables measured during the tests were the three-phase medium voltage supply and load voltages, the low voltage converter output currents, the dc-bus voltage of the DVR, and the low voltage load current during non-linear load testing. These measurements

(excluding the load current) were all shared by the DVR controller and the two Tektronix TDS 3014 four-channel oscilloscopes (CROs 1 and 2 in Figure 10.2). The oscilloscopes were triggered via a common signal from the DSP controller to synchronize the results and used the controller sag detection as a trigger timing mechanism.

The medium voltage supply and load voltages were measured using 24 kV ABB KEVA 24 A1 voltage sensors (1:10000 ratio) with galvanic isolation achieved using HCPL-7800 isolation amplifiers. LEMs similar to those in Chapter 9 were used to measure the low voltage converter output currents. The dc-bus voltage was measured using resistive dividers.

10.2.4 Controller Hardware and DVR Control Software

The digital controller comprised an Analog Devices AD21062 floating-point Sharc DSP and a Siemens SAB 80C167 Micro Controller linked via dual access RAM. This structure is categorized as the dual chip solution discussed in Chapter 6. The floating-point DSP is used for the primary control algorithm, and the fixed-point Micro Controller controls the PWM signal generation for the twelve IGBTs (i.e. six phase legs).

The existing control structure for a typical DVR sag compensation system was based on a combination of supply voltage feed-forward and d-q PI load voltage feed-back (Figure 10.4). A software-based phase-locked-loop (PLL) was used to create d-q coordinate sinusoidal references from the supply, which were used to determine the target phase of the output voltages. The phases of these references were controlled to only slowly vary during transients from their pre-sag values, to minimize phase jump effects. The feed-forward control then calculated the appropriate modulation depth to inject compensating voltages between the supply voltages (V_{Supply}) and the load voltages (V_{Load}), to restore the load voltages to the target references. However, this did not account for the voltage drop across the filter inductor and other parameters such as the transformer, and hence closed loop load voltage feed-back control was added in the d-q frame to minimize any steady state error in the fundamental component.

When the grid voltage is normal, the DVR system is held in a null state to lower its losses. Once a voltage sag is detected, the DVR switches into active mode to react as fast as possible, thus injecting the required ac voltages between the grid and the load. The voltage sag detection (Figure 10.4) is based on the rms of the error vector which allows for detection of symmetrical and non-symmetrical sags, as well as the associated phase jump. The detection formula is given

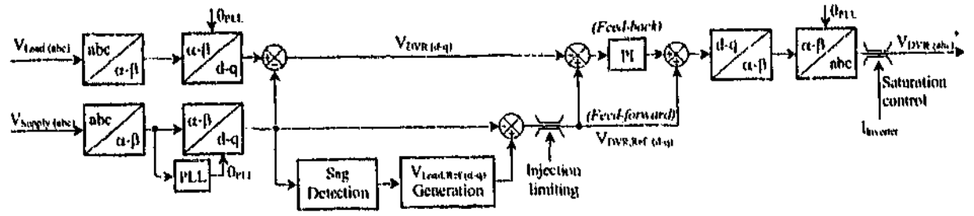


Figure 10.4: Existing sag compensation portion of the control structure (i.e. not including the harmonic voltage compensation).

by

$$|V_{error,dq}| > V_{Threshold} \tag{10.1}$$

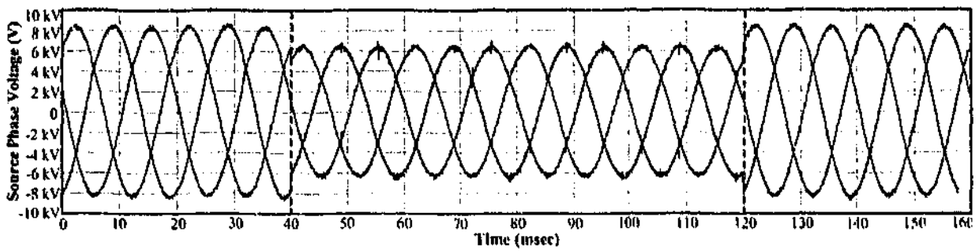
where the rms of the error is

$$|V_{error,dq}| = \sqrt{(V_{ref,d} - V_{Supply,d})^2 + (V_{ref,q} - V_{Supply,q})^2}. \tag{10.2}$$

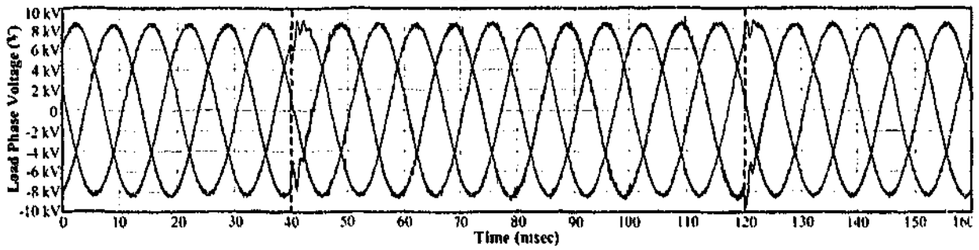
Due to limitations in both the computation time available in the DSP, as well as hardware limitations of the Micro Controller, only symmetrically sampled PWM was used by the DVR system. Therefore, the switching and sampling frequencies are matched (instead of sampling at double the switching frequency as was done with the asymmetrical PWM strategy used in Chapter 9). This results in a significantly lower control bandwidth. The default switching/sample frequency of the DVR was 3 kHz, but it was revised to allow both 3 kHz and 5 kHz after some optimization of the original DVR controller software. Whilst the system presented in the following section was successfully operated at both 3 kHz and 5 kHz, the majority of the following results are for the 5 kHz version, and this should be assumed unless otherwise noted.

10.2.5 Initial DVR Experimental Results

Figure 10.5 shows the response of the DVR to an 80 ms symmetrical voltage sag. A small resonance can be seen in the load voltage at both the start and finish of the sag. This is due to the LC filter resonance being excited by the high dv/dt of the sag. With harmonics programmed into the supply voltage, Figure 10.6 shows that the DVR still compensates for the voltage sag, but does not reduce the harmonic content. Without the DVR running and with the bypass contactor open, the non-linear load increases the voltage total harmonic distortion (THD) from

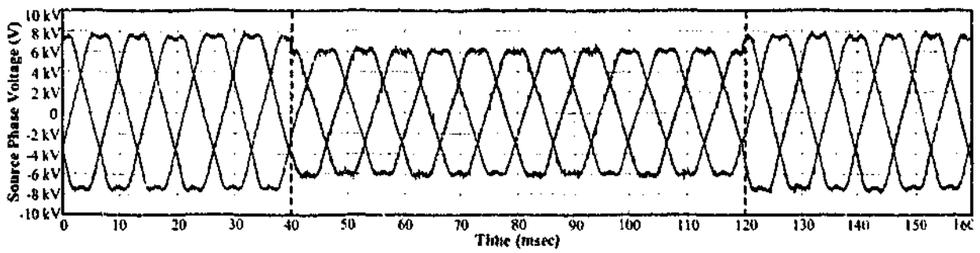


(a) Supply Voltage

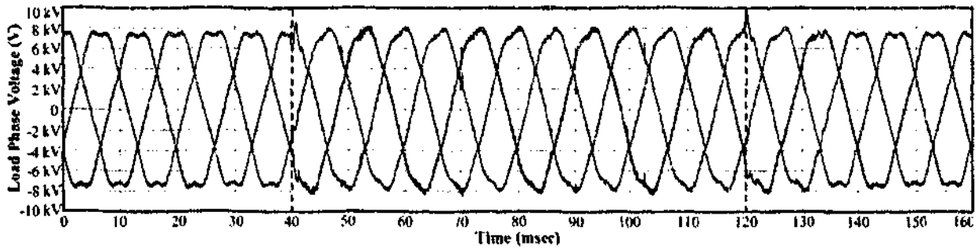


(b) Load Voltage

Figure 10.5: Operation of the DVR to a 0.76 p.u. dip with no supply voltage harmonics.



(a) Supply Voltage



(b) Load Voltage

Figure 10.6: Operation of the DVR to a 0.76 p.u. dip with supply voltage harmonics and no harmonic compensation.

1.6% to 1.9%, due to the added impedance introduced by the series transformer. For a much larger non-linear load (i.e. 45 kW), Nielsen [60] showed that the DVR in standby increased the voltage THD from 1.9% to 2.9%.

10.3 DVR with Voltage Harmonic Compensation

Since voltage sags generally only occur a few times each year at any particular location, a DVR system will generally spend most of its time in standby mode waiting for a sag to occur. Unfortunately, the DVR will still introduce extra impedance into the line, primarily due to the series transformer; and Section 10.2.5 has shown that this impedance will (in turn) cause a voltage drop to the load and increased load voltage harmonics when non-linear loads are present. As the DVR does not conventionally operate during steady-state conditions, it does not compensate for these added harmonics, or for other voltage harmonics in the system.

In principle, it would be advantageous if the series connected converter of the DVR can also be used to compensate for any steady state load voltage harmonics. This would increase the Power Quality 'value added' benefits to the system (which is the definition and driving force of Custom Power applications) with minimal extra capital cost, but of course with some increase in converter steady-state losses. The limitations on achieving this objective are steady state power flow constraints and the low modulation depths that must be used with a DVR that has a typical voltage injection capacity.

Unlike the bi-directional power supply in the UCPC, the energy storage system of a DVR is typically recharged using a small unidirectional power supply. Hence, most DVR's cannot supply significant steady-state real power and also cannot absorb almost any steady-state real power back through the series connection. Therefore any steady-state harmonic voltage compensation strategy that is implemented must ensure that the steady state real power flow through the DVR is kept close to zero.

The typical voltage injection capability of a DVR is in the range of 50%. Hence, to compensate for harmonics as low as 1% (or even lower) the system must operate at modulation depths of around 2%; but high magnitude and phase accuracy must still be maintained for the compensation to be effective. This chapter applies the selective harmonic feed-back control strategy presented in Chapter 4. This can be easily added to medium voltage DVR systems to provide voltage harmonic compensation with minimal effect on the sag compensation performance of the original DVR. Due to the power flow constraints listed above, only the resonant controllers are used in the steady-state operation, and the feed-forward, active damping and proportional

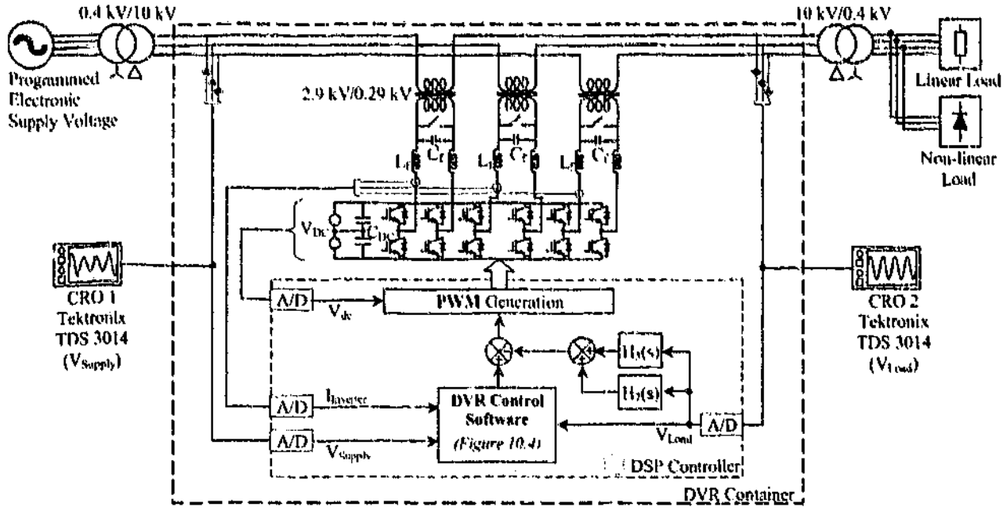


Figure 10.7: Circuit and control block diagram of the DVR (placed in a 20 foot container), and the experimental power system test setup.

components are not implemented. Note that the consequence of the loss of the proportional gain and active damping is a loss of control over the damping of the system. The DVR also does not have any additional series resistance to damp any LC resonance. However, as illustrated in Chapter 5, the main contributor to resonance problems is the excitation provided by the feed-forward component. As this component is not implemented in steady-state, many of the resonance problems do not exist, and this is investigated further in the following section using the control model developed in Chapter 4.

10.3.1 Proposed Additional Selective Voltage Harmonic Control

To add voltage harmonic compensation to the DVR system the converter must now switch continuously, instead of only when sags occur. Therefore, the standby mode must be revised to accommodate this change. For most DVR systems this will have little effect on the device conduction losses, but the additional switching losses must be traded against any power quality savings and possible reduction in downstream line and load losses by virtue of the reduced harmonics.

As identified before, it is critical that the harmonic compensation scheme has a net real power flow of zero (or very close to zero) at all times. Otherwise, since the system only has a low rated unidirectional power supply, a net power flow out of the converter will deplete the

available energy stored in the capacitors, and a net power flow into the converter will increase the dc-bus voltage past its ratings and possibly damage the system. To minimize the net real power flow, narrowband resonant based controllers [112] [113] are used to compensate for each selected harmonic, with no proportional term.

Figure 10.8 shows the frequency response of the stacked 5^{th} and 7^{th} resonant controllers, which have a high attenuation to any 50 Hz components to minimize disruption to the sag compensation system and the real power flow. Furthermore, it should be noted that any small 50 Hz component left in the controllers will be phase shifted by approximately 90^o, and will therefore only demand reactive power (again having no effect on the real power flow).

From Chapter 5 the resonant control filter is

$$H_n(s) = 2K_{I,n}\omega_c \frac{cos\phi_n s + (\omega_c cos\phi_n - \omega_n sin\phi_n)}{s^2 + 2\omega_c s + \omega_n^2 + \omega_0^2} \quad \text{where} \quad \phi_n = \omega_n \tau_{Delay}. \tag{10.3}$$

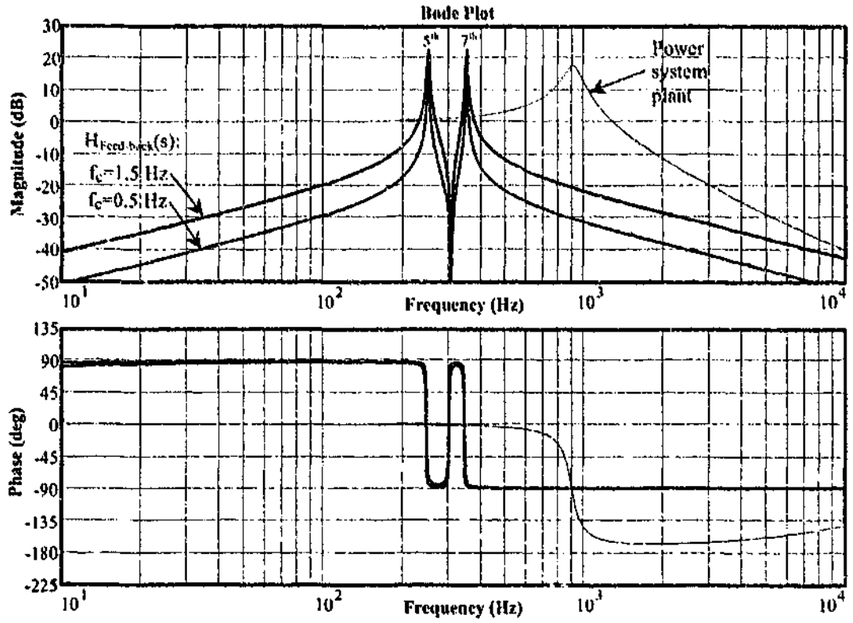
The full harmonic feed-back controller (without proportional component) is therefore given by

$$H_{Feed-back}(s) = \sum_{n \epsilon M} 2K_{I,n}\omega_c \frac{cos\phi_n s + (\omega_c cos\phi_n - \omega_n sin\phi_n)}{s^2 + 2\omega_c s + \omega_n^2 + \omega_0^2} \text{ where: } M = \{5, 7\}. \tag{10.4}$$

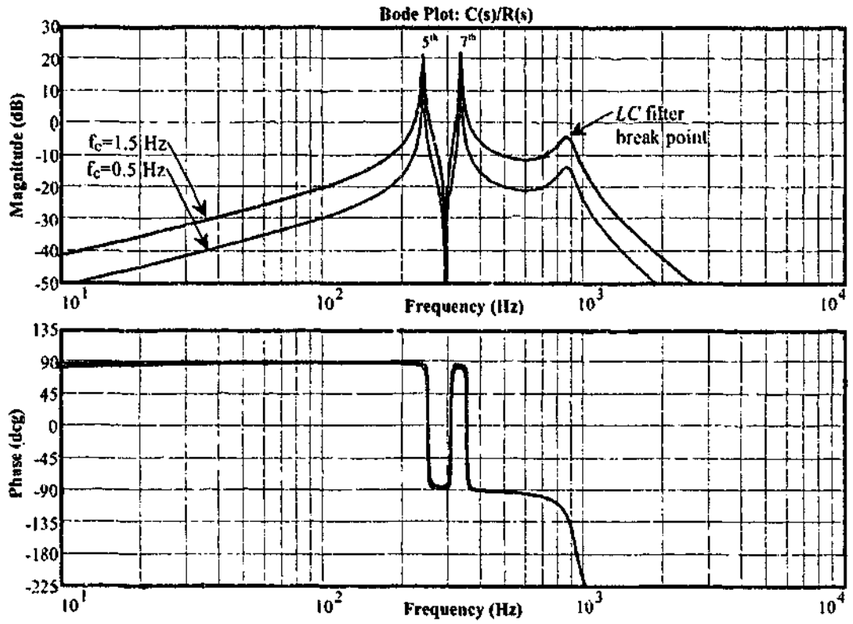
Figure 10.8a shows the bode plot for the feed-back controller (10.4), as well as for the system plant. From Chapter 4 it is known that changing the value of ω_c provides a compromise between the transient harmonic performance and the stability of the controllers. This is verified once again for the MV DVR by the system bode plot (Figure 10.8b), with a gain margin drop from 14.3-dB to 3.8-dB with f_c set to 0.5 Hz and 1.5 Hz, respectively.

The disturbance rejection simulation of the voltage harmonic controllers is illustrated in Figure 10.9, where the disturbance is taken as the supply voltage (V_{Supply}) and the target is the load voltage (V_{Load}). This result shows that the controllers have virtually no effect on the fundamental voltage, and they have more than 20-dB of attenuation of the 5^{th} and 7^{th} selected harmonics. But they also have a small effect on supply disturbances around the break point of the LC filter. At this frequency the 1.5 Hz controller has the higher disturbance gain of 8-dB, whereas this gain is less than 2-dB for the 0.5 Hz controller.

The Delta operator based IIR digital filters discussed in Chapter 6 were used to implement (10.4) to ensure high accuracy. The total DSP computational time for the feed-back controller is 11 μs (including digital filters and δ^{-1} operations for all phases at both the 5^{th} and 7^{th} harmonics), which is relatively small compared to the 200 μs time window that is available at 5 kHz operation (333 μs for 3 kHz). After optimization of the original controller software (reduced



(a) Separate feed-back controller and power system plant.



(b) Full system (i.e. feed-back controller and power system plant).

Figure 10.8: Simulated open loop frequency response of the (a) voltage harmonic controllers, and (b) the series injection system (both control and plant).

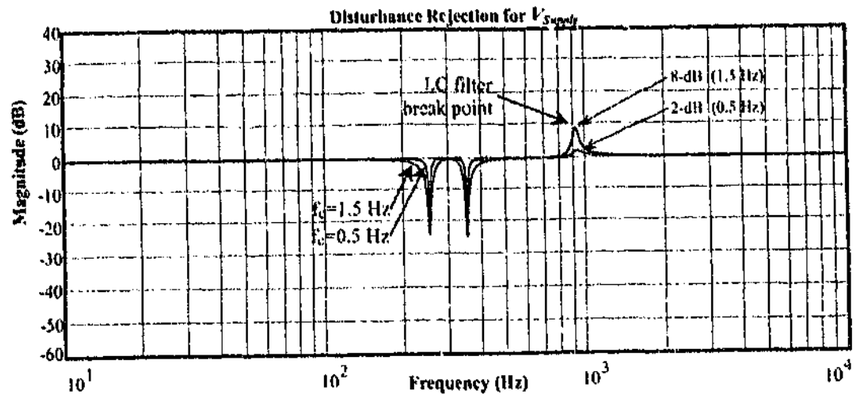


Figure 10.9: Simulated supply voltage disturbance rejection from the load voltage due to the proposed harmonic controllers.

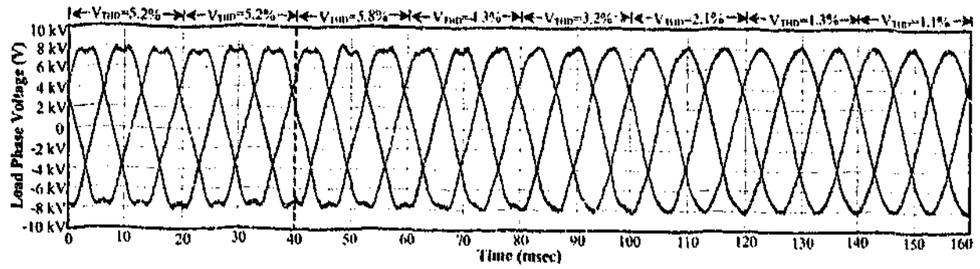
from over 200 μs, down to 110 μs), addition of the feed-back controller $G_1(s)$ and extra rate clamping required a total interrupt execution time of 127 μs.

10.3.2 Start-up Tests

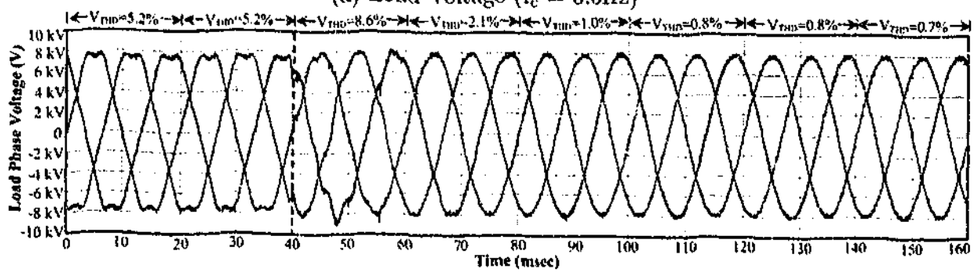
Figure 10.10 presents the experimentally measured start-up transient responses for both values of f_c. The response of Figure 10.10a (i.e. $f_c = 0.5$ Hz) is much slower than that in Figure 10.10b (i.e. f_c =1.5 Hz), with settling times of 5 cycles and 2 cycles, respectively. The reduced stability margin of the 1.5 Hz version is well illustrated by the large injection overshoot of the first settling cycle in Figure 10.10b.

10.3.3 Steady-State Harmonic Compensation Tests

The steady-state performance of the proposed system is summarized in Table 10.2, and includes results from: **(a)** a distorted supply voltage; **(b)** a non-linear load current (which in turn creates a distorted supply voltage); and **(c)** a combination of the two (i.e. both distorted supply voltage and a non-linear load). For test setup (a), Figures 10.11 and 10.12 show the waveforms and frequency spectrums of the supply and load phase voltages for switching frequencies of 3 kHz and 5 kHz, respectively. The faster switching frequency provides slightly higher attenuation of the 5^{th} and 7^{th} harmonics, however, the overall THD is similar between the two results due to other remaining harmonics. Figures 10.12, 10.13 and 10.14 show results for test setups (a), (b) and (c), respectively (f_{sw} = 5 kHz). In all cases the selected 5^{th} and 7^{th} harmonics are significantly



(a) Load Voltage (f_c = 0.5Hz)



(b) Load Voltage (f_c = 1.5Hz)

Figure 10.10: Experimentally measured startup transient response of the voltage harmonic compensation controllers.

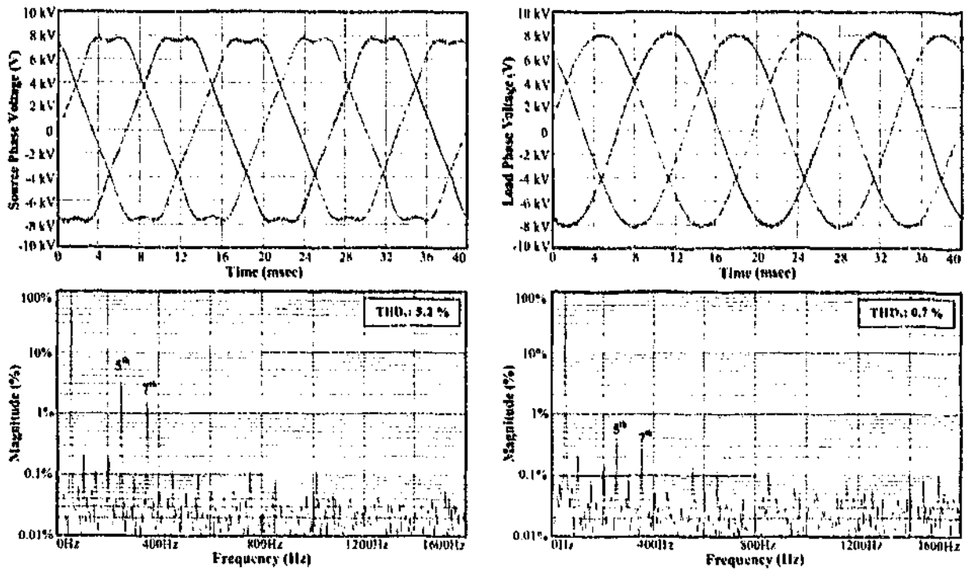
Harmonic Source	Voltage Point	5^{th}	7^{th}	THD
Supply Voltage Distortion	Supply	4.7 %	2.1 %	5.2 %
	Load	**0.4 %**	**0.1 %**	**0.7 %**
Non-linear Load	Supply	1.1 %	1.0 %	1.7 %
	Load	**0.1 %**	**0.1 %**	**0.9 %**
Supply Voltage Distortion	Supply	5.8 %	3.1 %	6.7 %
and Non-linear Load	Load	**0.5 %**	**0.3 %**	**1.2 %**

Table 10.2: Steady-state test results of the load voltage harmonics before and after compensation.

reduced by a factor of 10 (i.e. 20-dB) or more, and the THD is also substantially reduced since these harmonics are the most dominant in the system. Note that compensation has only been incorporated for the 5^{th} and the 7^{th} harmonics in this system, and other higher order harmonics will still exist. This can be seen in Figures 10.13 and 10.14, where the 5^{th} and 7^{th} voltage harmonics are attenuated in the load voltage, but the other low order harmonics such as the 11^{th} and 13^{th} still remain. (Note that shifting to an asymmetrical PWM scheme would double the controller bandwidth and would allow these higher harmonics to be easily compensated using the same switching frequency, as verified with the LV UCPC system. However, the *LC* filter break point frequency would also need to be increased.)

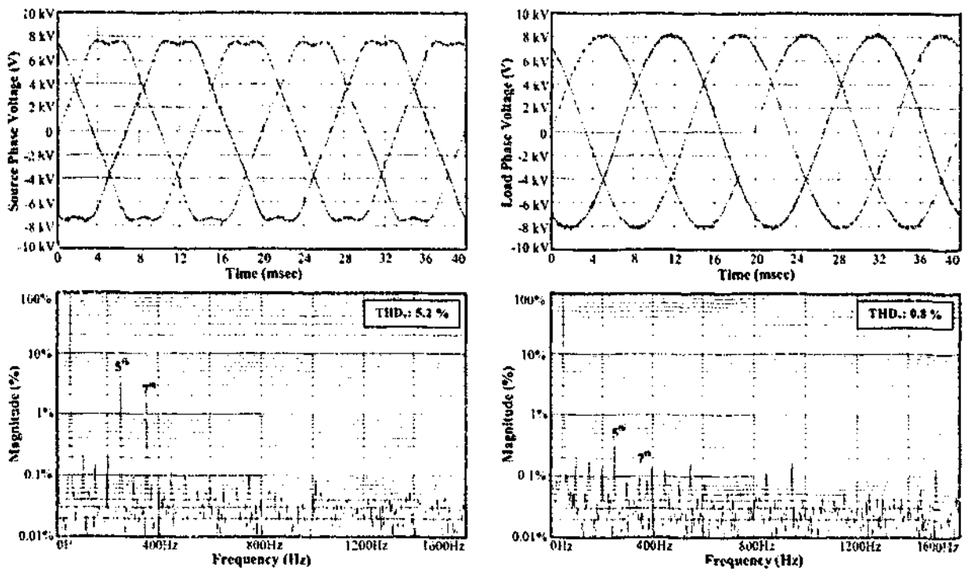
The simulated disturbance rejection plot for the supply voltage (Figure 10.9) shows that both the 0.5 Hz and 1.5 Hz versions should provide identical attenuation of the selected voltage harmonics. Theoretically, since the VSI is modelled as a linear gain (with compensation for V_{DC}), the resulting attenuation at the selected harmonics should also be independent of modulation depth. In practice, the VSI is not a perfect linear gain due to practical errors and other second order effects. Unlike the experimental UCPC in Chapter 9, the dc-bus voltage (V_{DC}) and the supply voltage (V_{Supply}) are not linked, and therefore they can be used to investigate these effects experimentally.

Figure 10.15 illustrates the practical operation of the DVR with varying modulation depths to compensate for the harmonics presented in test setup (a) (i.e. 5% 5^{th} and 2% 7^{th} supply voltage harmonics). These results were recorded using a HP 35660A spectrum analyzer. Figure 10.15a illustrates the compensation of the system for a varying dc-bus voltage (with all other parameters fixed), such that a larger dc voltage represents a lower modulation depth. The 5^{th} and 7^{th} harmonic levels remain steady within 0.05% for the entire range of results taken, but the voltage THD increases significantly as the modulation depth reduces (i.e. for a higher dc-bus voltage). Figure 10.15b confirms this result by varying the supply voltage from less than 1 kV up to the rated 10 kV, with the dc-bus voltage fixed (the lower the supply voltage, the lower the modulation depth required). Once again the controlled harmonics remain relatively constant, but the THD more than doubles in magnitude as the required modulation depth lowers. These results confirm that whilst the feed-back controller is able to regulate the selected harmonics independently of the DVR injection capability, the larger the injection capability (i.e. a higher dc-bus voltage or a smaller transformer ratio) the higher the expected overall THD. Note that in all cases presented, the THD was still significantly reduced from its original value of more than 5%, and this discussion serves only to explore the ideal operating conditions of the system.



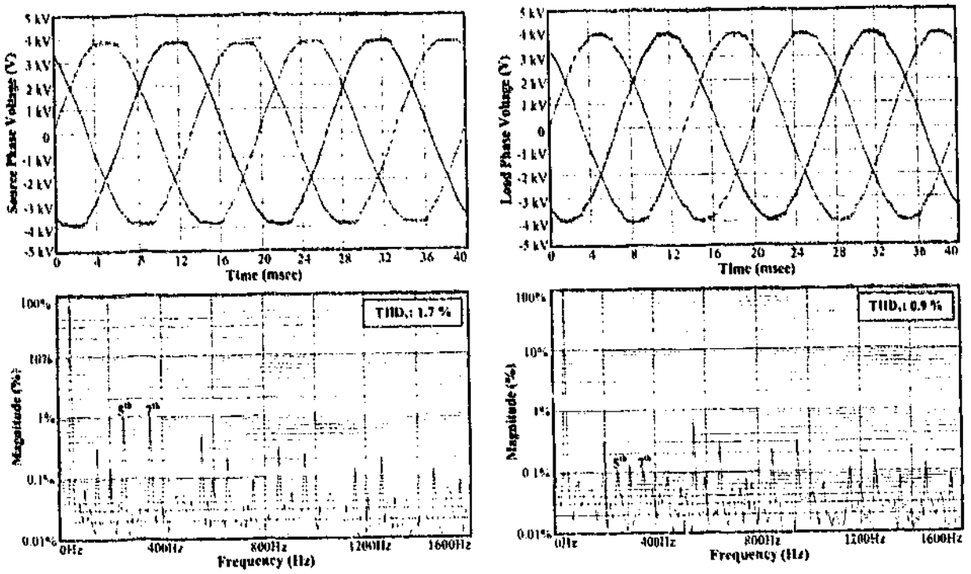
a) Supply Voltage b) Load Voltage

Figure 10.11: Experimentally measured steady-state voltage harmonic compensation performance in time and frequency domains: Source voltage distortion (f_{sw} = 3 kHz).



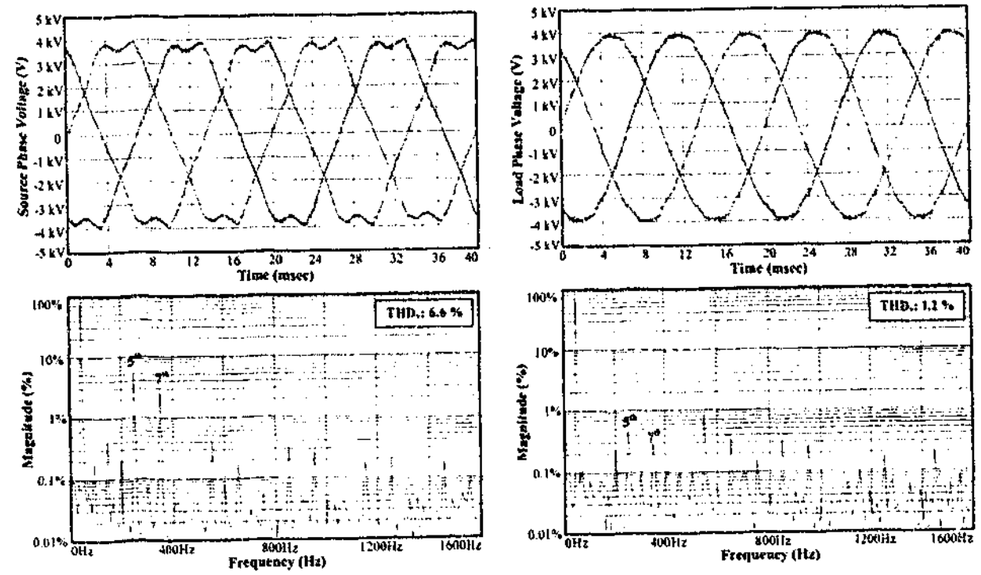
a) Supply Voltage b) Load Voltage

Figure 10.12: Experimentally measured steady-state voltage harmonic compensation performance in time and frequency domains: Source voltage distortion (f_{sw} = 5 kHz).



a) Supply Voltage b) Load Voltage

Figure 10.13: Experimentally measured steady-state voltage harmonic compensation performance in time and frequency domains: Non-linear load.



a) Supply Voltage b) Load Voltage

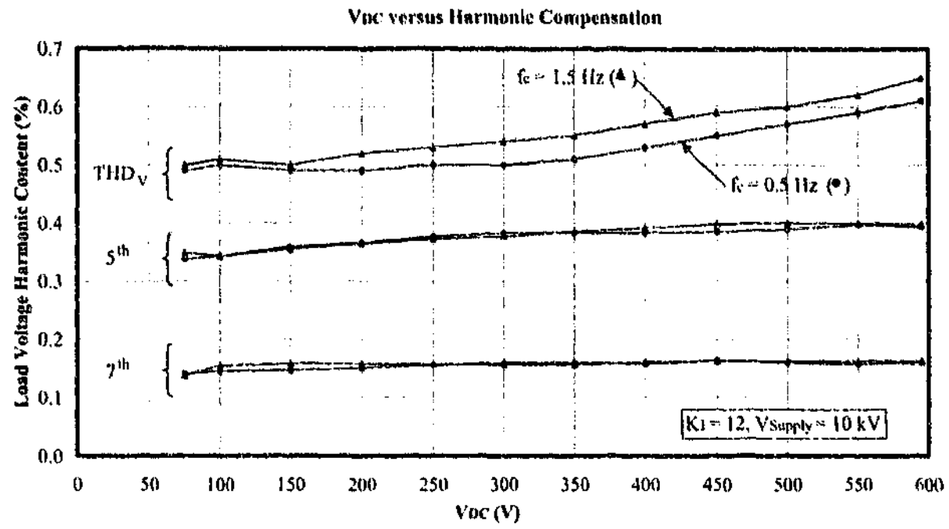
Figure 10.14: Experimentally measured steady-state voltage harmonic compensation performance in time and frequency domains: Source voltage distortion and non-linear load.

The design process for the controllers was presented in Chapter 4, and selected the resonant gain K_I based on the inverse of the required harmonic attenuation. This approximation is experimentally confirmed in Figure 10.16, which shown operation of the DVR with varying values of the resonant gain K_I. Figure 10.16a presents the absolute harmonic content for the varying gain, and this is then normalized to the achieved attenuation factor relative to the original supply harmonic in Figure 10.16b. In this normalized plot, the 5^{th} and 7^{th} results (both 0.5 Hz and 1.5 Hz versions) all perform equivalently at each gain tested. More interestingly, a plot of $1/K_I$ is also superimposed on the plot, and very clearly verifies the relationship used for the design process.

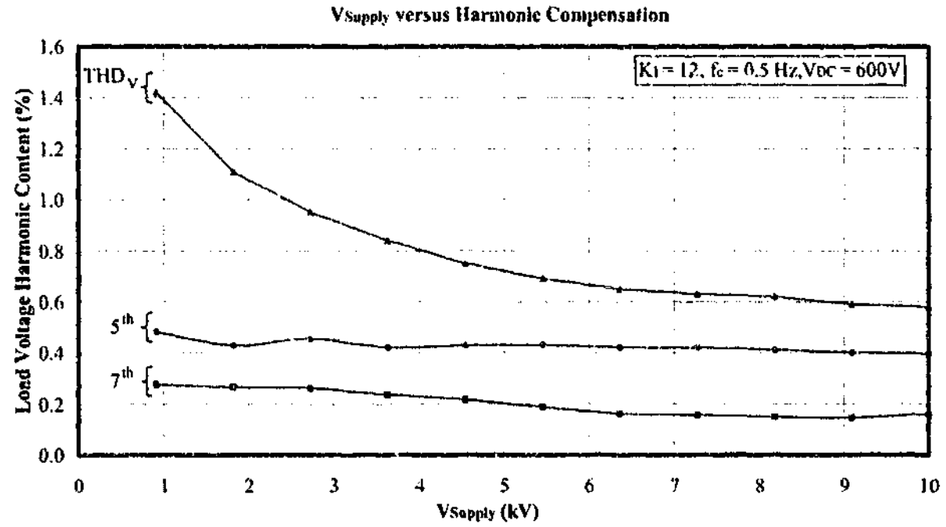
In Figure 10.12b it can be seen that some small magnitude harmonics have been introduced in the load voltage (i.e. they do not exist in the supply voltage). During the experimental testing of the DVR under a variety of conditions, harmonics in the 800 Hz to 1000 Hz range were consistently found to appear. Whilst these harmonics were generally only in the order of 0.1%-0.3%, the effectiveness of the 5^{th} and 7^{th} compensation means that in many cases these harmonics became more dominant in the overall load voltage THD. Since this frequency region contains the break point of the LC filter, it seemed logical to conclude that the primary reason for the increase was LC filter resonance, especially since Figure 10.9 had already shown the expected amplification of supply voltage harmonics in this frequency region. However, correlating the disturbance plot in Figure 10.9 with the 17^{th} and 19^{th} supply voltage harmonics in Figure 10.12a (which are in this case the disturbance input to the model), the predicted gain of these disturbances does not match the magnitude of the load voltage harmonics presented in Figure 10.12b. Another source of excitation for this resonance must therefore exist. This source may well be the harmonics generated by the symmetrically sampled PWM switching scheme.

Typically with a switching frequency of 5 kHz or more, the effects of switching harmonics are very minimal. For this system, however, the reference signals are 250 Hz and 350 Hz, rather than the typical 50/60 Hz. Hence the sidebands are spread much wider around the carrier, wide enough to intrude into the lower frequency harmonic region. Open loop PWM simulations with a combined 5^{th} and 7^{th} reference (using the magnitudes required to compensate for the harmonics shown in Figure 10.12) show harmonics of slightly less than 0.1% in the LC break point region, confirming this as the extra harmonic source.

Figure 10.17 presents a plot for the effect on the load voltage caused by disturbances injected at the output of the converter. In this case the disturbance is caused by the switching harmonics discussed. (Note that the low frequency components have an average disturbance rejection of

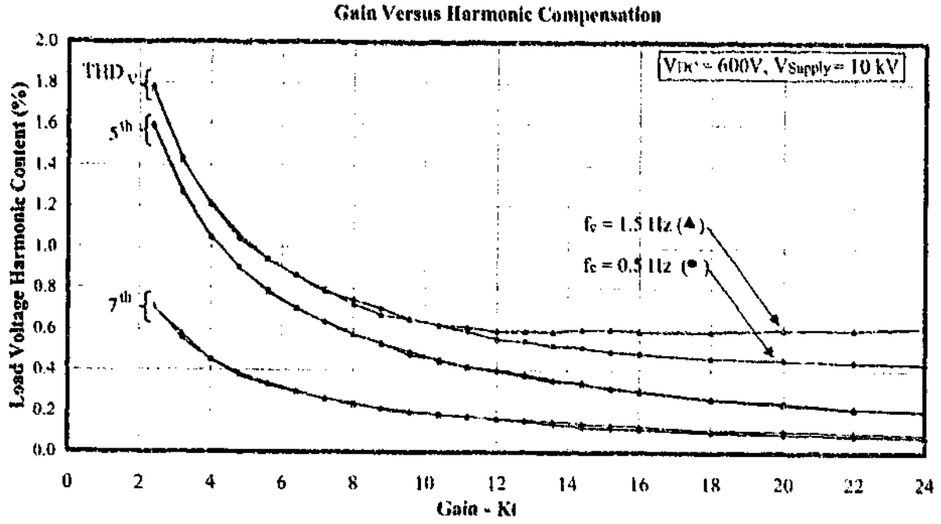


(a) Sweep of load harmonics for changing dc-bus voltage (i.e. varying modulation depth).

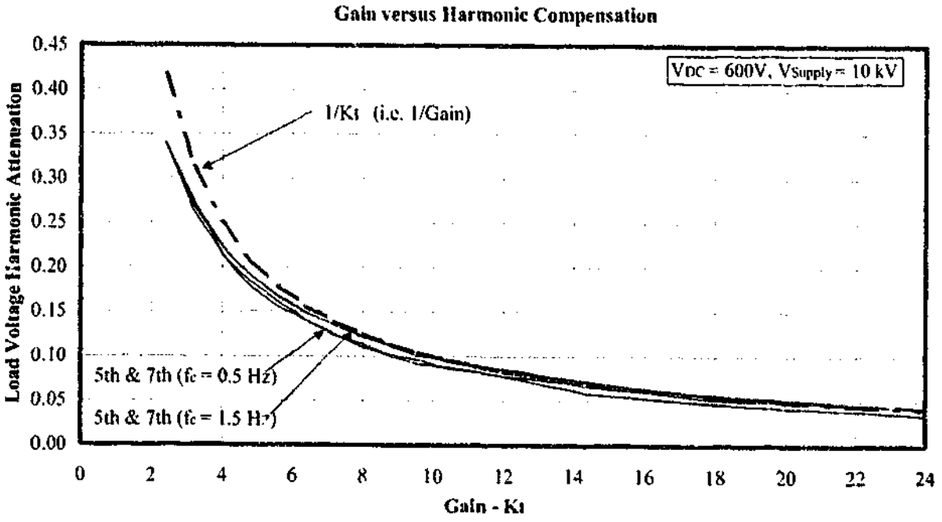


(b) Sweep of load harmonics for changing supply voltage (i.e. varying modulation depth).

Figure 10.15: Experimentally measured results of the DVR with varying, (a) dc-bus voltage, and (b) supply voltage, to highlight the effect of varying the modulation depth.



(a) Sweep of load harmonics for changing resonant gain K_f.



(b) Sweep of load harmonic attenuation for changing resonant gain K_f.

Figure 10.16: Experimentally measured results of the DVR with varying gain. Figure (a) shows the actual harmonic content, and (b) is the attenuation relative to the original supply harmonics.

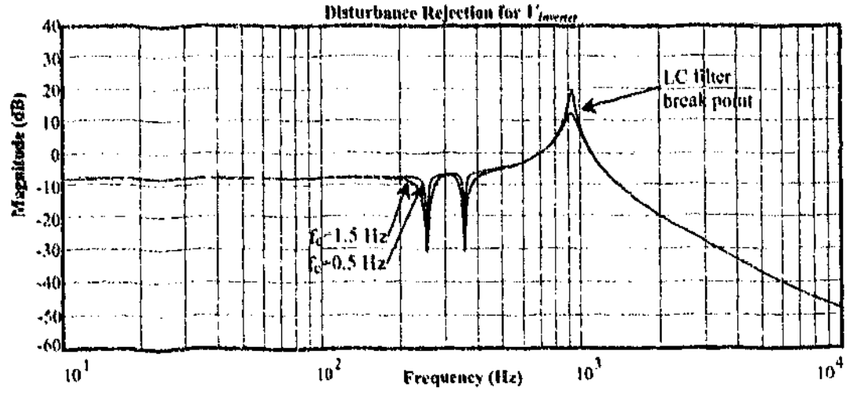


Figure 10.17: Simulated inverter output voltage disturbance rejection from the load voltage due to the proposed harmonic controllers.

less than 0-dB due to the transformer ratio.) This shows that the system is very sensitive to the introduction of disturbances in the LC filter break point region, which is expected since this point is directly connected to the LC filter input. Furthermore, correlation between this plot and switching harmonics already discussed, results in a very similar magnitude of introduced harmonics as shown in Figure 10.12. Whilst further analysis is beyond the scope of this research, it highlights the fact that for optimum overall performance, effects such as the switching scheme, LC filter values, nominal dc-bus voltage and the switching frequency, should be taken into account during the design of the converter parameters (particularly to avoid extra excitation in this region from the converter). Note that the active damping scheme presented in Chapter 4 would also be useful here, although the real power flow constraints mentioned previously will limit its use.

Finally, as discussed in Chapters 3 and 4, selective harmonic compensation creates an equivalent low impedance condition at the selected frequencies (i.e. zero magnitude and zero source impedance). For these tests the effect of voltage compensation on the non-linear load current harmonics was minimal, as expected, since the impedance of the distribution transformer between the DVR and this load is the dominant influence. However, if the harmonic compensated DVR was connected directly to the low voltage side at the load, then there is a possibility for large increases in current harmonics. This particularly refers to voltage stiff non-linear loads (such as the diode rectifier with capacitive dc load used here for the experimental work) where large increases in peak current will occur for very low equivalent impedances at certain harmon-

ics (including the 5^{th} and 7^{th}). Therefore, the medium voltage installation presented here is seen as a more practical installation level for the proposed scheme.

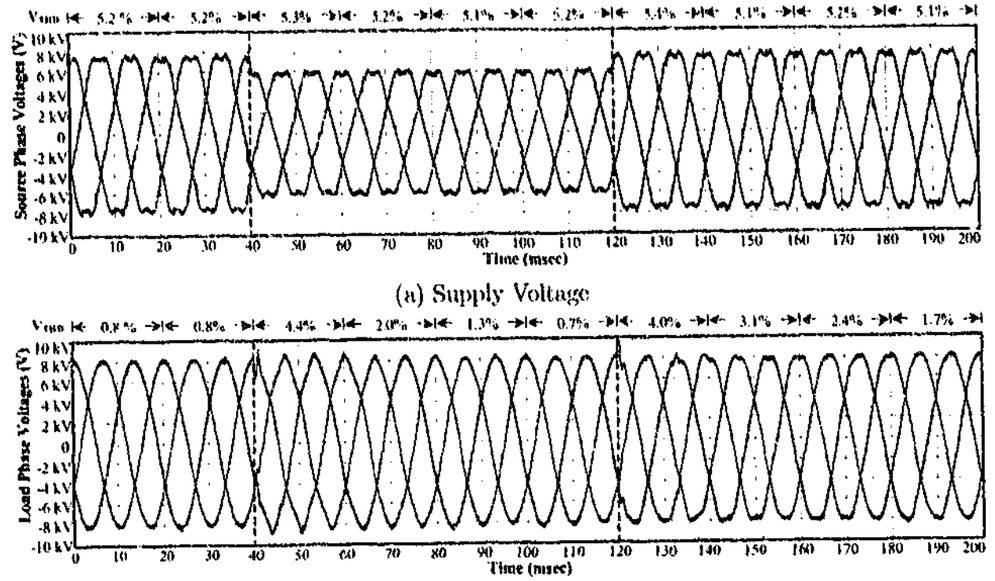
10.3.4 Dynamic Sag and Harmonic Compensation Tests

The dynamic tests on the system were conducted using simultaneous harmonics and sags (symmetrical and asymmetrical), and included combinations of distorted supply and linear/non-linear loads similar to the steady state tests. During the transient events, the combined feed-forward/feed-back controller shown in Figure 10.4 was enabled, and hence the system characteristics were changed. In particular the added feed-back proportional component provided some damping of the LC resonance. The control and disturbance plots that apply during these transient events therefore vary from those provided earlier.

Figure 10.18 shows the response of the DVR to a 0.79 p.u. symmetrical sag and a linear load, with the harmonic compensation enabled. The sag compensation scheme can be seen to continue to function as expected, and the harmonic compensation settles within about two fundamental cycles with an f_c of 1.5 Hz. Note that as a part of the sag, the harmonic magnitudes will also typically vary, and hence a transient is seen in the harmonic compensation. As with the previous start-up transient (Figure 10.10b) a small overshoot is seen during the first fundamental cycle when the larger value of f_c is used.

Figures 10.19 and 10.20 show results for the compensation of another symmetrical sag, but this time with a non-linear load (Figure 10.19 also includes voltage harmonics in the supply). For these results a smaller f_c of 0.5 Hz is used to allow comparison with the previous results. The response time to the transient is visibly slower, but has no initial overshoot. Note that as with the steady-state results in Figures 10.13 and 10.14, some harmonics still remain as only the 5^{th} and 7^{th} harmonics are targeted for compensation.

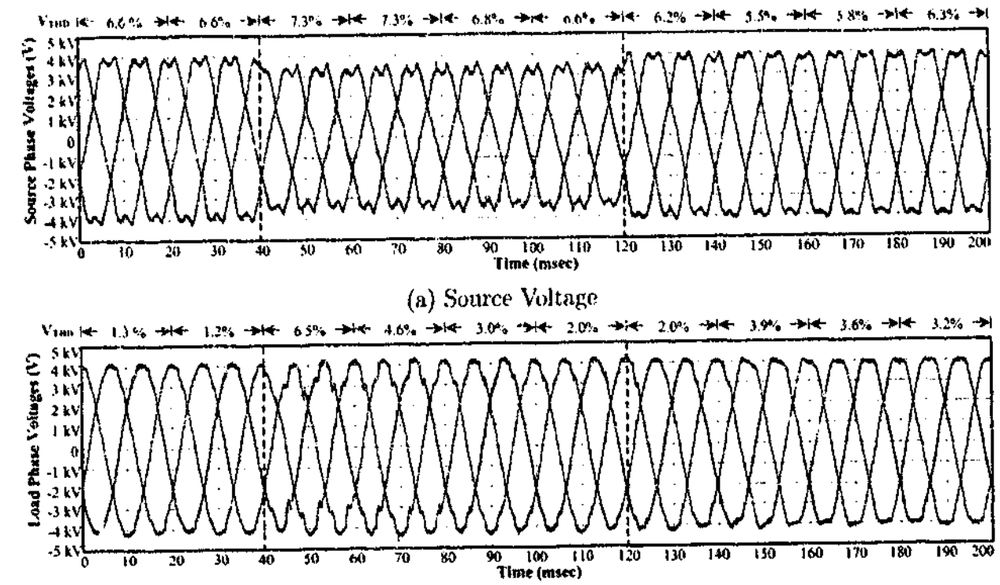
The final results in Figure 10.21 are for an asymmetrical sag with a linear load. At the low voltage terminals of the programmable supply a 0.63 p.u. single-phase sag was created, which became a shallower two-phase sag on the medium voltage side of the star-delta transformer. Once again the sag is compensated as required, with a small harmonic overshoot created due to the 1.5 Hz cut-off frequency used for this result.



(a) Supply Voltage

(b) Load Voltage

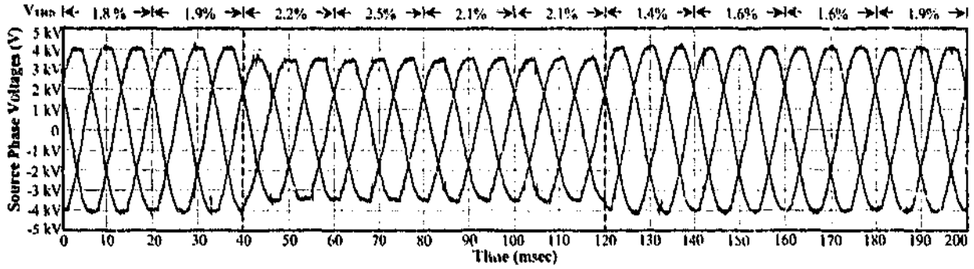
Figure 10.18: Experimentally measured results of the DVR with 5^{th} and 7^{th} voltage harmonic compensation for a dip with supply voltage harmonics and linear load.



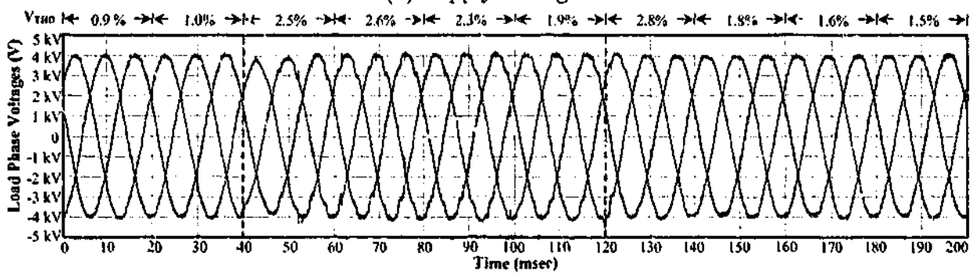
(a) Source Voltage

(b) Load Voltage

Figure 10.19: Experimentally measured results of the DVR with 5^{th} and 7^{th} voltage harmonic compensation for a dip with supply voltage harmonics and non-linear load.

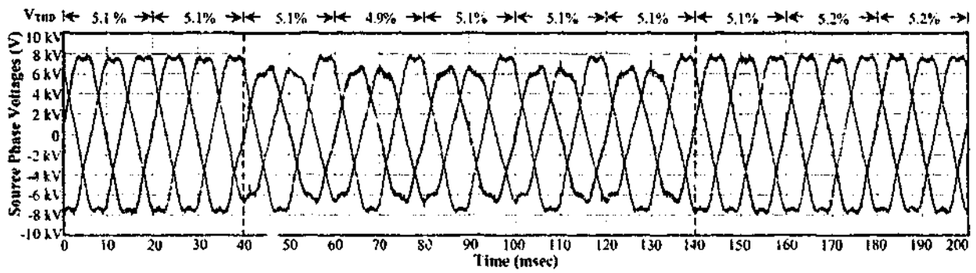


(a) Supply Voltage

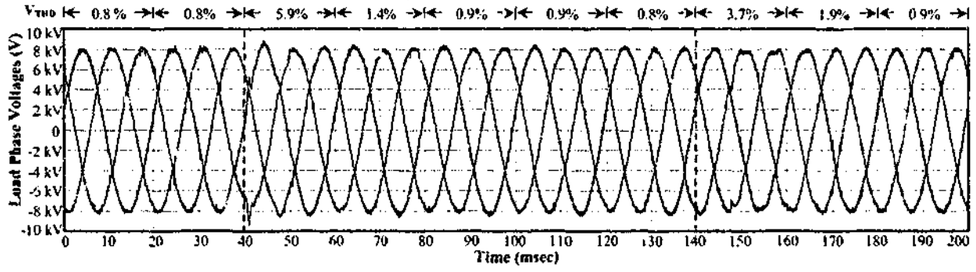


(b) Load Voltage

Figure 10.20: Experimentally measured results of the DVR with 5^{th} and 7^{th} voltage harmonic compensation for a dip with a non-linear load.



(a) Supply Voltage



(b) Load Voltage

Figure 10.21: Experimentally measured results of the DVR with 5^{th} and 7^{th} voltage harmonic compensation for an un-symmetrical dip with supply voltage harmonics.

10.4 Summary

This chapter has verified the application of P+Resonant selective harmonic compensation to a medium voltage connected series topology. The experimental results provide harmonic attenuation equivalent to the low voltage results presented in Chapter 9, verifying that stationary frame linear controllers can easily be applied to medium voltage systems. The 10 kV DVR prototype was tested under a range of steady state and transient conditions containing harmonics in the supply voltage. Both linear and non-linear loads were tested, as well as combinations of symmetrical and asymmetrical voltage sags. The work has extended the capabilities of the DVR to include harmonic voltage compensation, with minimal (if any) alterations required to the existing hardware system, and also without affecting the operation of the sag compensation control.

The resonant controllers were implemented and successfully tested for sample frequencies of 3 kHz (less than a third of that used for the LV UCPC) and 5 kHz, confirming the operation of the proposed scheme for lower sample frequency operation. Chapters 6 and 9 investigated the operation of delta based IIR digital filters on fixed-point implementations, and they have now also been shown in this chapter to be suitable for floating-point operation.

Steady-state testing of the medium voltage system, over a wide range of operating parameters, has verified experimentally the relationship between the harmonic attenuation and the resonant gain used in the design of the controllers. These tests have also confirmed the ability of the controllers to provide consistent harmonic attenuation, irrespective of the operating modulation depth, with the overall voltage THD only varying slightly due to the small increase of other harmonics under some operating conditions. However, the overall load voltage THD was still significantly lower than the original supply voltage under all cases tested.

Chapter 11

Conclusion

Custom Power technologies have now been of interest for more than a decade, and are becoming more widely used as the financial burden of Power Quality phenomena on industrial and commercial electrical consumers increases. Most commercially available Custom Power applications for power distribution networks are either shunt or series converter based, but are rarely both. Existing systems reported in the literature are yet to take full advantage of the capability of the combined active series-shunt topology to simultaneously compensate for a broad selection of the Power Quality problems, and most control systems are only applicable to three-phase systems.

This thesis describes the design and application of a Universal Custom Power Conditioner, or UCPC. The UCPC is capable of simultaneously compensating for Power Quality problems in both the load voltage and the supply current. For the load voltage it can compensate for deviations in fundamental magnitude, harmonics, unbalance, sags, swells, phase jump, and flicker. For the supply current, both harmonics and unbalance can be compensated. The work has focused particularly on the control aspects of the UCPC in order to provide improved performance compared to existing systems, and to allow a broader application to address both more types of Power Quality problems and single-phase systems. The control aspects explored include linear control techniques, digital implementation, and the integrated control of the converter protection systems. Each of these aspects has provided contributions to the field of power electronics, which are applicable not only to the UCPC, but also to a wider range of applications.

This concluding chapter presents a summary of the main findings and outcomes of the thesis, as well as some discussion of possible future work.

11.1 Summary of the Work

11.1.1 The Universal Custom Power Conditioner (UCPC)

The thesis has presented an extended development of the Universal Custom Power Conditioner. The UCPC is a single Custom Power device suitable for broad compensation of Power Quality problems, and can achieve improved steady-state voltage compensation compared to existing systems. It is designed for insertion into either low or medium voltage radial systems, on either three-phase or single-phase networks. A fully digital implementation is preferable to allow for future upgrades, flexibility, and to avoid the aging drift effects seen in analog systems.

In this work, both the series and shunt converters of the UCPC are rated for only 0.5 p.u. of the protected load, and therefore cannot compensate for very deep sags and interruptions. Despite this, the UCPC is still expected to be able to protect the load for up to 85-90% of sag events. This is because the majority of predicted sags occur within the range of the device, and also because most loads can withstand any shallow sags that remain because of incomplete compensation. The ratings and load sag protection limitations of the UCPC has been investigated with respect to sag depth, phase jump, load power and angle, and stored energy limitations. The placement of the UCPC in the network was also investigated in order to achieve maximum protection from Power Quality phenomena.

The control system that has been developed for the UCPC consists primarily of stationary frame linear controllers which, unlike synchronous frame controllers, provide for the option of either single- or three-phase operation. Separate P+Resonant feed-back controllers are used for the series converter to attenuate each selected harmonic (including the fundamental) by feed-back directly from the load voltage. The low computational overheads of this control technique allow many harmonic controllers to be easily implemented. This is the first reported application of the stationary frame P+Resonant controller for use with the series injection topology. The use of selective harmonic feed-back control of the load voltage is also an original contribution for the series topology. In addition, an active damping scheme was presented for use with the series topology, to damp the resonance effect arising from the output *LC* filter of the VSI. An analytical model for the series system was developed, and used in conjunction with an original design methodology to determine the hardware and control system parameters of the system.

11.1.2 Stationary Frame Signal Extraction

This research has developed a new stationary frame active filter signal extraction control block, building on techniques that were used to develop the P+Resonant controller. This work achieves an exact stationary frame equivalent to a high-pass filter placed in the *d-q* reference frame, and is called the StatRF. The proposed scheme has significant computational advantages, and does not require a sine table. Due to its exact equivalence, the StatRF can also be used as a convenient simulation block for existing synchronous frame extraction controllers. However, as with its synchronous frame counterpart, the StatRF only removes the positive sequence fundamental component, leaving both the harmonics and negative sequence components in the output signal. This cross-coupling between the phases means that the controller is not applicable for single-phase systems. To resolve this problem, two different methods were used to develop single-phase versions of the StatRF. This work showed that the single-phase equivalent is very closely related to the narrowband notch, and using an approximate conversion technique the outcome actually becomes a narrowband notch.

11.1.3 Digital Implementation

The next part of the work applied delta operator based Infinite Impulse Response (IIR) digital filters to implement the proposed stationary frame controllers on a 16-bit fixed-point DSP. This work investigated the problems associated with using the common approach of shift-based IIR digital filters, as well as using the alternative of Finite Impulse Response (FIR) digital filters. For 16-bit fixed point operation IIR filters were shown to diverge from their continuous prototypes as the sample frequency increased, while FIR filter were found to require an unrealistic number of taps to maintain a close relationship with their continuous counterpart for the narrowband types of filters required with the UCPC system. Reduced tap numbers might be suitable for the P+Resonant controller in applications where the aim is only to obtain a high gain at the selected frequency. However, correct signal cancellation between the forward and cross-coupled signals is critical to the StatRFs operation, and hence the FIR approach cannot be used.

Delta based IIR digital filters were then identified as potentially alleviating the problems presented by these typical approaches, and a detailed review of the method was presented. To the author's knowledge, this is the first reported application of delta based IIR digital filters for real-time control of converter applications. The effectiveness of the delta approach was verified for both the P+Resonant and StatRF controllers. The design (i.e. coefficient and parameter selection) and implementation (i.e. overflow control, truncation compensation, and fixed-point

scaling) were examined and discussed for fixed-point implementations. For the StatRF implementation the shift IIR filter was shown to only be feasible with a 20-bit coefficient word length, whereas the delta IIR still operated with a coefficient resolution as low as 6-bit. Nearly all existing DSPs dedicated for use with power converters are fixed-point, and therefore the recognition of the delta IIR filter's advantages for converter control implementation is of significant importance for a wide range of high performance converter applications – especially those using the stationary frame controllers discussed in this work.

11.1.4 Protection

The fourth contribution of this thesis is the development of an integrated protection scheme for the series portion of the UCPC. The protection of shunt converters is straightforward and well known, but the application of the same approach to the series topology can lead to disastrous results. The series protection problem has been previously identified by other researchers, but was only more recently addressed in detail in the literature. This thesis identifies and addresses several problems with the approach described by Moran et al. [101]. In contrast, the series protection control proposed in this work integrates both software and hardware components. This ensures a continuous, and appropriately rated, current path under fault conditions, start-up, stand-by, shut-down, and recovery from faults, as well as providing redundancy in the protection system. The series protection scheme was verified using both simulation and under experimental short-circuit conditions.

11.1.5 Experimental Verification Processes

The last step of the research program was the experimental verification processes. The UCPC was thoroughly tested experimentally using a low voltage prototype. A dedicated controller board was developed for use in the system, and is now also used by other researchers across the globe (including Australia, U.S.A., Nepal, Singapore and Korea). The series component was also experimentally verified on a medium voltage (10 kV) prototype system. To the author's knowledge, this is the first reported experimental system which incorporates series active voltage harmonic compensation at medium voltage levels.

11.2 Suggestions for Future Work

11.2.1 Multi-level Implementations

This research has assumed the use of a typical two-level VSI for both the series and shunt converters in the UCPC. For medium voltage applications a relatively large power rating is likely to be required for many suitable network placement options. For this reason, the medium voltage experimental system was also verified at switching frequencies as low as 3 kHz. However, even this frequency may not be sufficiently low for some higher power applications when using a two-level VSI. Multi-level converters offer a possible alternative, and have become a popular research topic in recent years for higher power applications. Multi-level converters have already been applied to shunt active filters [161] [162] [163], and been applied also to a Universal Power Conditioner using back-to-back five-level diode-clamped VSIs [71]. For series applications it has been established that for the majority of the time the converter will run at a low modulation depth, and this can cause balancing and utilization problems for diode-clamped converters. This problem has initially been investigated by Tolbert et al. [71], but further investigation is required. The UCPC back-to-back application of cascaded, capacitor-clamped, and hybrid multilevel topologies also needs investigation to quantify the problems which may exist, and to compare the advantages and disadvantages of each topology for the implementation of the type of UCPC presented in this work.

11.2.2 Further Experimental and Control Investigations

The medium voltage experimental system discussed in this thesis only investigated the series section of the UCPC. Further work is needed to extend this medium voltage experimental system to the full UCPC series-shunt topology. Investigation into the problems associated with using the PCR algorithm in conjunction with a connection transformer also needs to be undertaken, or another suitable current regulation scheme chosen. Higher power testing of the medium voltage system is also desirable.

The experimental work so far has been based on three-wire systems. A next step could be to experimentally apply the UCPC to both four-wire and single-phase systems. The medium voltage system verified that the series controller can control each phase independently, but the effect of having a fluctuating supply to the dc-bus under single-phase conditions needs to be considered (i.e. in contrast the power flow of a three-phase system is constant). The most likely application for the UCPC on single-phase systems is for compensation of SWER (Single Wire

Earth Return) lines, but these are nearly always very weak systems, and the robustness of the PCR algorithm on such systems has been noted as being of some concern. Further work on this problem is currently being undertaken at Monash University [14]. For four-wire systems the experimental low-voltage UCPC that was developed already contains the majority of the control and hardware functionality (e.g. four-phase leg converters and wiring) required to begin testing, and this is left as a starting point for future work.

The feed-forward component of the series controller could also be replaced with a deadbeat voltage controller. Whilst this would introduce additional stability considerations, it may also reduce the *LC* filter resonance problems and make the active damping scheme redundant. To reduce the size and cost of the system slightly, the shunt soft start contactor (but not the isolation contactor) could also be removed, and the soft start operation undertaken by the series converter. However, further work needs to be done to consider ways of achieving this without affecting the load voltage (i.e. by using phase offset voltage injection).

11.2.3 UCPC Network Application Research

This research has approached the UCPC application from a power electronics control perspective, and has therefore had little focus on the power system aspects of such an installation. The preferred placement of the UCPC was discussed, but before an installation could be considered in practice, more detailed simulation investigations into the wider network affects of the UCPC are required. This work would include investigations into the effects of the UCPC on the network protection coordination, as well as a grid simulation which takes into account the entire surrounding network. For this simulation the analytical series model presented in this work is ideal, but it still requires the development of an equivalent model for the shunt converter system.

The insertion of the UCPC into the distribution system also has small loss implications for the network, due to losses in both the converters and the passive components. It is expected that a reduction in Power Quality problems is likely to reduce the losses in the network (especially the reduced harmonic content) and this loss reduction could be offset against the UCPC losses. However, a detailed investigation into this trade-off would be necessary to assess the net impact and long term financial justification of the UCPC.

11.2.4 Effects of PWM on Closed-loop Linear Controllers

The medium voltage system investigation used the series analytical model to create a disturbance plot of the sensitivity of the load voltage to disturbances at the output of the converter. This

plot was used to consider the mitigation of PWM switching harmonics into the load. PWM is now a very mature field, and the switching harmonics produced by power converters can be easily determined for open loop operation using both switched simulations, and by pre-calculated analytical equations [43]. However, the exact correlation of this work with closed loop systems remains unclear. With further research, the use of this type of disturbance plot should provide a useful link between this wealth of knowledge and the knowledge base of linear closed loop converter controllers.

11.3 Closure

The high precision and fast transient response requirements of Custom Power applications has attracted interest from many power electronics researchers over the last couple of decades. The aim of simultaneously compensating for a broad range of Power Quality phenomena has directed the work presented in this thesis towards finding new technologies to make this goal more realistically achievable.

New stationary frame control techniques have been developed with reduced practical overheads and improved performance, and previously developed stationary frame controllers have been applied in a new application. Digital control aspects have been investigated in detail, and new techniques applied to the digital control of power converters. A scheme for the integrated control of protection systems for the series topology has also been proposed. Whilst these contributions have been developed to achieve the ultimate goal of this research – to compensate for Power Quality phenomena – the potential for their broader application is clear. For example, the series protection scheme has already been used and referenced by several other research groups [60] [132]. Furthermore, during the course of this research work, the delta IIR has already been further applied to a power conditioner for traction systems, a stat-com, a UPS, and in renewable energy power conversion applications. This diversity of new applications using the contributions of this thesis is expected to continue.

Appendix A

Power Quality

This appendix is a review of Power Quality and is expanded from the summary presented in Section 2.1 for readers requiring further background information in this field.

The research in this thesis focuses on the development of a Custom Power system to compensate for a variety of Power Quality issues in a single unit. *But what is Power Quality? Which of these phenomena are important?; why?; and what are the consequences to both the consumer and the utility? What is the probability of such phenomena occurring? What limits must Custom Power devices achieve, and which of these targets on each phenomena are mandatory or recommended?* This section answers these questions, and provides background information on Power Quality to outline the research goals for the design and final implementation. Power Quality is also littered with vague, differing, and misused terminology, and therefore this section will define such terms for use in this research, to ensure the clarity of discussions.

A.1 Power Quality and the Consumer

There are many definitions of the terms *"Power Quality"* (and *"Quality of Supply"*); the following definition given by Dugan et al.[18] may be said to be the most general and appropriate definition:

> *A Power Quality problem is "any power problem manifested in voltage, current, or frequency deviations that results in failure or misoperation of customer equipment."*

Power Quality issues such as waveform distortion, transients, voltage flicker, unbalance (also imbalance[18]), and voltage variations affect the consumer and supplier in three ways: damage, nuisance and economics. Damage may come from extra heating and fire in conductors due to harmonics, over-voltages outside the rated values of installed components, motor vibration, and

other sources. Nuisance includes light flicker, tripping of microprocessor based appliances, communication noise (including telephones and T.V.s), and added audible noise from components such as transformers. Economic impacts arise from the replacement of damaged items (including components with reduced life spans due to Power Quality issues), lost production time (due to supply outages/sags), errors in the measurement of power flow used by the power distribution companies to charge their clients, to name a few. This financial cost to the consumer and/or supplier, as well as mandatory Power Quality standards, are the major driving force for the purchase and installation of Custom Power solutions.

A.2 Terminology and Classification

Many variations of terminology exist for Power Quality phenomena. The terminology and classification provided in Table A.1 [18] has been chosen for this thesis.

Only types 2 through 6 are discussed in the following sections, as these are the Power Quality events to be compensated for in this research. Type 1 events (such as voltage spike transients) are not included as they are generally faster than the bandwidth of the power electronic solutions discussed in this research, and are usually clamped using varistors, or other similar methods. Power system frequency variations (type 7) are also not compensated, as the influence of Custom Power applications on the grid is not significant enough to be able to vary this quantity (unless the chosen topology fully decouples the supply and load – which is not the case in this research).

The definitions used in the classifications are based on the IEEE (The Institute of Electrical and Electronics Engineers) and other U.S.A. based standards organizations' publications[18]-[21], and examples of each type are illustrated (Figure A.1).

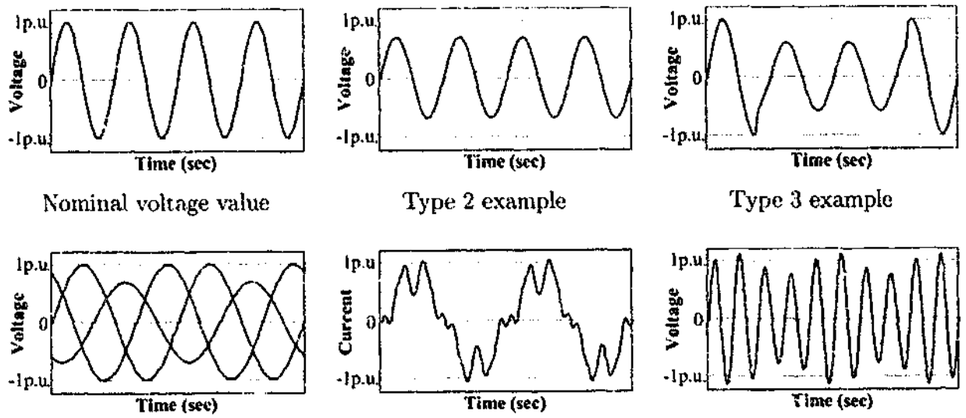
Further terminology clarification is required for the description of the voltage levels. For this research work, Low Voltage (LV) is used for distribution systems and consumer reticulation between 110 V and 415 V, Medium Voltage (MV) is used for distribution voltages between 6.6 kV and 22 kV, and High Voltage (HV) is used for 66 kV and higher for distribution and transmission systems. The research focuses only on LV and MV *radial* installations.

A.3 Power Quality Standards

The serious and widespread nature of Power Quality problems has led to the development of a wide range of standards by professional bodies worldwide. The majority of the existing standards are either IEC or IEEE, with other well known standards from Semiconductor Equipment and

Classification Type	Classification Sub-Type
Type 1: Transients	Impulsive transient
	Oscillatory transient
Type 2:. Long-duration voltage variations	Over-voltage
	Under-voltage
	Sustained interruptions
Type 3: Short-duration voltage variations	Interruption
	Sags (dips)
	Swells
Type 4: Voltage unbalance	
Type 5: Waveform distortion	dc offset
	Harmonics
	Interhamonics
	Notching
	Noise
Type 6: Voltage fluctuations	Voltage flicker
Type 7: Power frequency variations	

Table A.1: Catagories of power system Power Quality phenomena [18].



Nominal voltage value | Type 2 example | Type 3 example

Type 4 example | Type 5 example | Type 6 example

Figure A.1: Example waveforms of type 2 to type 6 Power Quality problems.

Materials International (SEMI), Information Technology Industry Council (ITIC), and American National Standards Institute (ANSI). Such standards are important as they determine target performance levels for Custom Power devices. Also, some standards provide information and sample data regarding the ways of testing the suitability of Power Quality solutions such as Custom Power devices (i.e. [19] and [20]). Due to the large quantity of Power Quality standards available, only those which affect the design choices of the Custom Power device described in this research will be reviewed here. These include: the Victorian Distribution Code [26]; IEEE and IEC standards relating to harmonics [21] [164], voltage sags [19] [20] [165], voltage unbalance [166], and flicker [34] [167]; and the ITIC (replaces CBEMA–Computer and Business Equipment Manufacturers' Association) curve [163] [168] and SEMI standards [169][170]. Each of these are discussed below in their respective classification sections.

A.4 Short-Duration and Long-Duration Voltage Variations (Types 2 and 3)

The most obvious Power Quality issues for many consumers are long and short duration voltage fluctuations which cover both classification types 2 and 3 (Table A.1). In some instances the terms interruption, outage, sag, dip and undervoltage are used somewhat interchangeably, and therefore to maintain the clarity in the discussions that follow, these terms need to be more accurately defined[1]. Once again the definitions from the text by Dugan et al.[18] will be utilized, and have been summarized in Figure A.2. All events shorter than 60 seconds are defined as type 3 short-duration, whilst all events past this limit are type 2 long-duration.

The voltage variation problem may be viewed from three perspectives: the susceptibility of equipment to such variations; the probability of these variations occurring; and the compatibility of equipment with its installation, based on the first two perspectives.

A.4.1 Equipment Susceptibility

Equipment known to be susceptible to voltage variations includes (but is not limited to) Personal Computer (PC) equipment, PLCs, Variable Speed Drives (VSD), ac relays, motor starters and High-Intensity Discharge (HID) lamps [18] [20] [165] [171] [172]. Table A.2 provides example

[1]Note: The terms 'sag' and 'dip' are interchangeable (dip is used in the IEC standards), and only the term 'sag' will be used hereafter. Sag values will always be provided as a ratio of the nominal value (not the percentage drop caused by the sag). For example a sag which causes a 10% drop from the nominal voltage is noted as a 0.9 p.u. (or 90%) sag.

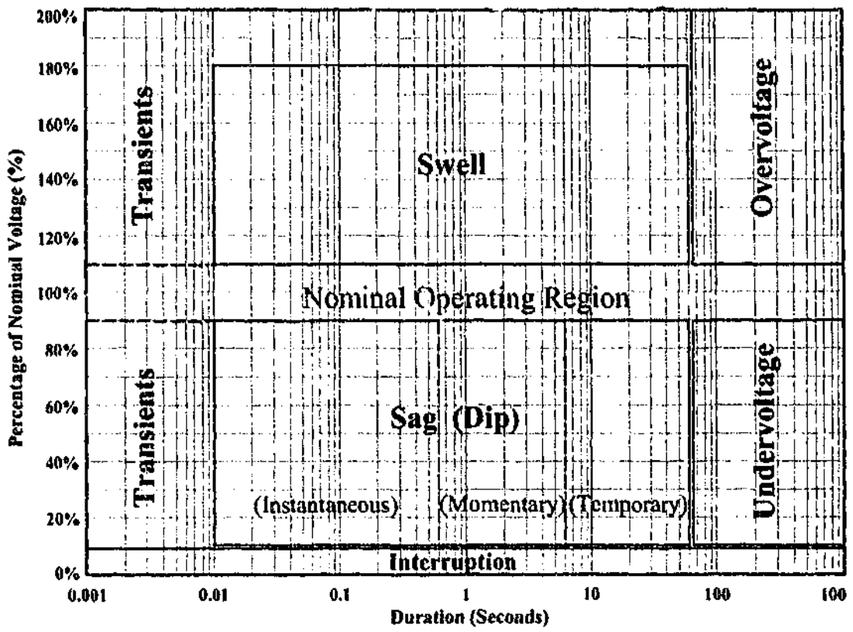


Figure A.2: Diagrammatic classification of long (type 2) and short (type 3) duration voltage fluctuations.

voltage tolerances for some of these devices [20]. The values in each entry give the time for a full interruption and/or the steady state voltage that will cause the device to fail or misoperate, respectively. This data shows that it is plausible for a sag of only 1 or 2 cycles to cause misoperation of some equipment, and steady state voltages of only 80% of the nominal can also cause problems.

The reason for the susceptibility of equipment varies greatly depending on the type of device, and the more common examples follow. The temporary failure of many forms of electronic equipment during a sag is commonly due to insufficient voltage on the dc capacitors, and many devices will become significantly more susceptible to sags as the steady state voltage drops [173]. This is because the energy storage of a capacitor is proportional to the square of the voltage ($\frac{1}{2}CV^2$), and a drop of 20% in the steady state voltage will result in a 36% loss in the stored energy available to 'ride through' a sag. The susceptibility of VSDs (and other similar equipment) can also be due to over-current tripping, which arises from the reduced supply voltage causing the VSD to draw more current to feed its constant power load [22] [27]. Note also that it becomes even more susceptible when the drive is running closer to its rated power

	Voltage Tolerance		
Equipment	**Upper Range**	**Average**	**Lower Range**
PLC	20ms, 75%	260ms, 60%	620ms, 45%
PLC Input Card	20ms, 80%	40ms, 55%	40ms, 30%
5 h.p. ac Drive	30ms, 80%	50ms, 75%	80ms, 60%
ac Control Relay	10ms, 75%	20ms, 65%	30ms, 60%
Motor Starter	20ms, 60%	50ms, 50%	80ms, 40%
Personal Computer	30ms, 80%	50ms, 60%	70ms, 50%

Table A.2: Susceptability of equipment to voltage sags [Maximum interruption time, Minimum steady state voltage] (Source: IEEE Std 1346:1998).

during the sag, and at both rated speed and rated load a two cycle sag of only 0.75 p.u. could trip the drive [27]. For control relays, many are "sealed in by their own contacts", and are likely to drop out with an interruption of only half a cycle or more [174]. Some high pressure sodium HID lamps can also have their arc extinguished with an interruption of less than a cycle; and new lamps are susceptible to steady voltages of 75%, with this value deteriorating quickly with age [171]. Once the arc is removed, the lamp takes several minutes to turn back on, which is highly undesirable for situations such as car park security lighting.

As well as voltage magnitude, the voltage phase is also quite important for some loads. Susceptible loads are primarily grid connected thyristor rectifiers including some VSDs and Static Var Compensators (SVC) [22] [27] [65]. A voltage phase jump is associated with many voltage sags and should be considered for compensation depending on the type of load.

A.4.2 Event Probability

The probability of sags and other Power Quality events occurring depends on such a high number of known/partially known/unknown variables for each site that a purely theoretical calculation is impractical. An extensive study was undertaken by Electrical Power Research Institute (EPRI) which measured 222 distribution feeders across the U.S.A. for two years from June 1993. This data is included in the IEEE Standard 1346[20], and a normalized version of that data is presented in Table A.3. This data shows that nearly half of all sags are in the 0.8-0.9 p.u. range, with high proportions of the remaining sags in the 0.7-0.8 p.u. and 0.6-0.7 p.u. ranges. A total of 78% of sags are 0.6 p.u. and higher. Also, most of the sags (72%) were

	Time in seconds					
Magnitude	**0.0 < 0.2**	**0.2 < 0.4**	**0.4 < 0.6**	**0.6 < 0.8**	**≥ 0.8**	**Totals**
80% - 90%	35.9 %	5.6 %	2.4 %	1.0 %	4.2 %	**49.0 %**
70% - 80%	15.3 %	1.4 %	0.8 %	0.4 %	1.0 %	**18.9 %**
60% - 70%	7.8 %	1.2 %	0.4 %	0.2 %	0.4 %	**10.0 %**
50% - 60%	4.6 %	0.8 %	0.2 %	0.2 %	0.2 %	**6.0 %**
40% - 50%	2.8 %	0.4 %	0.2 %	0.2 %	0.2 %	**3.8 %**
30% - 40%	2.0 %	0.4 %	0.2 %	0.0 %	0.2 %	**1.9 %**
20% - 30%	0.8 %	0.2 %	0.2 %	0.0 %	0.0 %	**1.2 %**
10% - 20%	0.8 %	0.2 %	0.2 %	0.0 %	0.2 %	**1.4 %**
0% - 10%	2.0 %	0.6 %	0.2 %	0.0 %	4.2 %	**7.0 %**
Totals	**71.9 %**	**10.8 %**	**4.8 %**	**2.0 %**	**10.6 %**	**100.0 %**

Table A.3: Normalised average voltage sag data from EPRI study incorporating 222 distribution feeders for 2 years. Nominal value is 50.2 events per site per year. (Adapted from IEEE Std 1346:1998).

less than 0.2 seconds. The average total number of sags per site per year was 50.2. This is nearly one per week

Whilst this data will not provide a true indication of the probabilities of events at a new site elsewhere, it does provide a general overview as to where the more common problem areas lie. It also allows for rough estimates to be made on the likely annual effect of voltage sags on consumer equipment, once the susceptibility data is obtained for the device. This comparison is known as the compatibility of the device and is discussed in the following subsection.

A.4.3 Equipment Compatibility

The compatibility of equipment with its supply quality must be considered, and a decision made as to whether the equipment should be upgraded, or alternatively, if the supply quality issue should be addressed. In many instances the option of upgrading multiple pieces of equipment would be too costly, and a single solution to improve the Power Quality of the site (or a portion of the site) is pursued. Custom Power devices are one of many options. This work focuses on Custom Power solutions which only work to improve the high probability ranges listed above, resulting in a more financially viable Power Quality solution which encompasses problems of

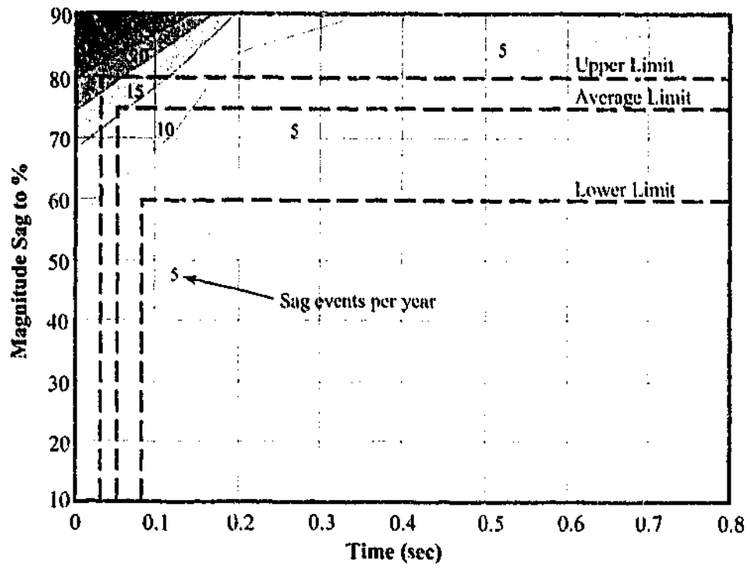


Figure A.3: Voltage sag coordination chart for a 5 h.p. ac drive (Data Source: IEEE Std 1346-1998).

types 2 through 6.

The compatibility of equipment with its supply can be ascertained by the use of coordination charts outlined in IEEE Standard 1346 [20]. These are simply the combination of the equipment susceptibility limits and the sag event probabilities presented in a graphical form (e.g. Figure A.3), and will be used later in the thesis to show the advantages of this work in reducing the compatibility problems.

Recommendations have been set down by the ITIC [165] [168] and SEMI [169] [170] for the design of equipment with regard to voltage sag immunity; but these are rarely achieved due to increased cost [175]. From the utility supply perspective in Victoria, the Victorian Electricity Distribution Code [26] governs the voltage variations which may occur without penalty to the utility. Figure A.4 presents the ITIC, SEMI and the Victorian Electricity Distribution Code in a single graph to highlight the gap between these standards. The graph shows that the utility can produce voltage variations, without penalty, which will cause possible temporary failure of equipment adhering to the ITIC or SEMI standards[2].

[2]The SEMI F47 standard does not include overvoltage limits, and notes this lack due to the "extremely low number of" interruptions caused to semiconductor equipment by this category [169].

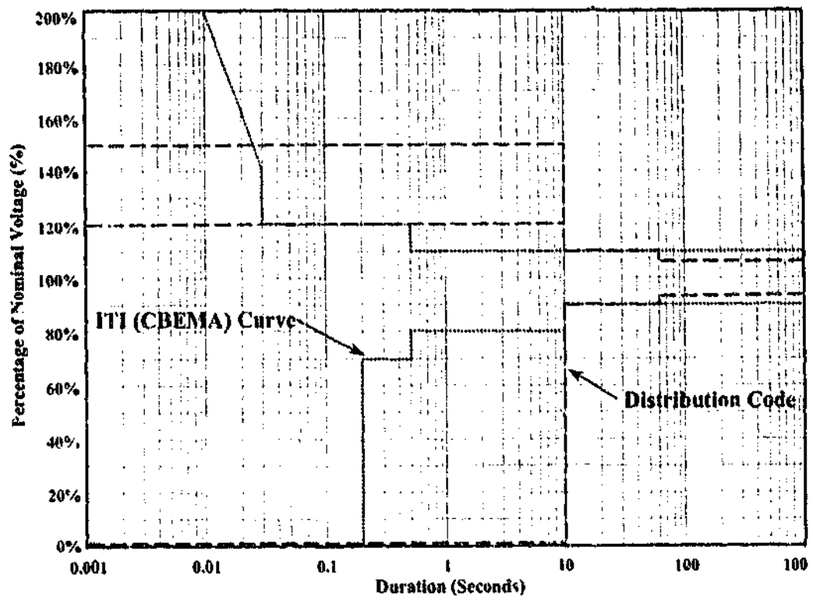


Figure A.4: Comparison of ITIC curve and limits specified by the Victorian Distribution Code.

A.5 Voltage Unbalance (Type 4)

Voltage unbalance (also imbalance [18]) can be defined in three ways: (1) "the ratio of the negative sequence and the positive voltage component"; (2) "the ratio of the difference between the highest and the lowest magnitude", and (3) "the average of the three voltage magnitudes; and the difference between the largest and the smallest phase difference between consecutive phases" [22]. These descriptions can therefore be termed 'negative sequence unbalance', 'magnitude unbalance' and 'phase unbalance', respectively [22]. For four-wire systems voltage unbalance is both the negative and zero sequence components, but for three-wire systems (where zero sequence does not exist) it is simply the negative sequence component. Whilst, theoretically, some voltage unbalance also occurs during type 2 sags caused by unbalanced faults (i.e. single and two phase faults), the type 4 voltage unbalance particularly refers to the steady state variations caused by unbalanced loads and other power system asymmetries. The Victorian Distribution Code specifies a limit of 1% on the negative sequence component [26].

Unlike the instant recognition by consumers of type 2 and 3 voltage variations (when their computers and lights go out temporarily) the effects of voltage unbalance and waveform distortion are typically less obvious, although not necessarily less important. The most dominant

effect of voltage unbalance is the reduction in the life of motors due to increased heating and vibrations [22] [23] [24] [25], and also the reduction in developed torque [22] [23]. Voltage unbalance of 1% on an induction machine can cause a loss of life of more than 5%, but with an unbalance of 2% or 4% this loss of life increases significantly to 22% and 64%, respectively [23]. For VSDs, and other equipment using, three-phase diode rectifiers with capacitive filters on the dc side, small amounts of voltage unbalance can lead to large even harmonic currents. This in turn can cause equipment to trip on over-current [22] and introduces further problems as discussed in the following section.

A.6 Waveform Distortion (Type 5)

Harmonic (or waveform) distortion is becoming an issue of great concern with increased use of non-linear loads. Some common examples of polluting loads are: VSDs, Switched Mode Power Supplies (SMPS) used in computers and other electronic equipment, saturation of magnetic components, arc furnaces, and lighting ballasts. For individual devices (below 16A) the IEC 61000-3-2 (AS/NZS 61000.3.2 in Australia) [164] states the harmonic current limits for different classes of equipment. This standard is only valid for new equipment (not pre-existing equipment), and still allows relatively high levels of current distortion from some classes of equipment.

Harmonics are known to have caused the malfunction of devices such as VSDs [27], capacitors [28], circuit breakers [29], fuses [29], conductors, electronic equipment, lighting, metering [30], protective relays, rotating machines [23] [25], solid state relays, static reactive power compensators [31], telephone communications [32] [33], and transformers. The causes of malfunction include: increased heating, increased dielectric stress, shifted voltage zero crossings, higher di/dt, increased magnetic fields and capacitive coupling, and increased vibration. Additional heating is a particularly important problem as it may create safety problems with the possibility of starting fires due to overheating cables and other devices. Diode rectified loads are also likely to cause 'flat-topping' of the voltage waveform, which describes a reduced peak voltage. The consequence of this common occurrence is a reduction of the dc bus voltage in other electronic equipment, which increases susceptibility to voltage sags [173] (as previously discussed). Both the IEEE 519 Standard [21] and the IEEE Task Force on the Effects of Harmonics on Equipment Report [173] present excellent summaries on the "Effects of Harmonics on Equipment"; therefore further discussion on these well known effects will not be pursued.

The IEEE 519 standard recommends that utilities should regulate the voltage harmonics

on LV systems to 5.0% THD (Total Harmonic Distortion) [3], and that consumers draw current with no more than 5.0% THD [21]. Voltage and current limits are also imposed on each individual harmonic. These limits vary as the short circuit to load current ratio changes, and are made mandatory in Victoria by the governing body, the Office of Regulator General (ORG)[26]. Whilst harmonic limits (and others) are made mandatory for many utilities and their consumers worldwide, utilities in the U.S.A. do not typically monitor the harmonic levels unless a customer raises a complaint or a particular problem arises [176]. This is usually also the case for utilities in Australia.

A.7 Voltage Fluctuations - Flicker (Type 6)

Voltage fluctuations are defined as "a series of voltage changes, or a continuous variation of the r.m.s. voltage" [34]. A common form of voltage fluctuations is 'flicker'; which, as the name suggests, causes some lights (particularly fluorescent lights) to appear to flicker. Variations as small as 0.5% in the 6-8 Hz range can cause a visible effect [18]. Flicker causes irritation to consumers working under such conditions, and can even be a trigger for illnesses such as photosensitive epilepsy. Arc furnaces are a major contributor to voltage flicker in distribution networks [18].

In Australia these fluctuations are required to remain within maximum limits[4]. The IEC standards use a variety of measures for voltage fluctuations, which include: short-term flicker indicator, long-term flicker indicator, and the relative voltage change 'd(t)'. Each of these measures and their IEC limits are given in Appendix .

A.8 Summary

This Power Quality review provides further background information to aid design choices to be made for the Custom Power application in this work. A large portion of the discussion was devoted to long and short duration voltage variations (types 2 and 3), as this background information is critical to the design and rating of the series inverter portion of the chosen Custom Power topology (the primary focus of this work). Furthermore, these variations have more obvious financial implications to most consumers, and are likely to provide the major cost

[3] For consumers with a short circuit to load ratio of less than 20

[4] Flicker limits in Australia are set by the AS/NZS 61000.3.3 and AS/NZS 61000.3.5 standards, which are relabeled from the IEC 61000-3-3 [34] and 61000-3-5 [167] standards, respectively. The U.S.A. based ANSI C84.1 is used by other countries for the same purpose.

justification for installing a Custom Power solution. The review shows that a device capable of protecting sensitive loads from sags down to 0.5 p.u. will be effective in more than 80% of sags and interruption events. Also, since the susceptibility of most equipment will allow for a shallow dip of say 0.8-0.9 p.u. (or even deeper), then to ensure compatibility of these devices to their supply for a 0.5 p.u. sag, only the difference of 0.3-0.4 p.u. is required. This reduced demand on the Custom Power device should further decrease the capital cost. This work investigates systems with maximum voltage injection capabilities of no more than 0.25-0.50 p.u. of nominal.

Steady state Power Quality problems such as voltage unbalance (type 4), waveform distortion (type 5) and flicker (type 6), are more difficult to quantify in purely financial terms. The voltage unbalance and voltage/current distortion can cause premature ageing and increased audible noise in equipment, but these effects are rarely detected (if at all) until years later when a device fails. Safety is also a problem, due to possible fires started by increased heating caused by current waveform distortion (particularly in cables). On the other hand, flicker (type 6) is more related to irritation and health issues for selected people in the community. For these reasons Custom Power devices which compensate for Power Quality types 4 through 6 are important to the consumer, but are more likely to be installed due to the mandatory (or recommended) standards listed in this review; instead of financial incentives.

The review has discussed numerous recommended and mandatory standards which exist for limiting the Power Quality phenomena supplied to consumers, as well as the susceptibility of their equipment to such events. Even with these standards, there is still a large gap between the emission limits and the equipment susceptibility limits. Also, due to financial considerations, these recommendations are in many instances ignored, until such limits are enforced (which is typically only in extreme cases when actively pursued). This issue is predominant in the mandatory limits set down by bodies such as the Victorian Office of Regulator General (ORG). The ORG's Electricity Distribution Code[26] sets out distortion limits for the load current drawn by consumers, such that neighboring consumers (on the same connection) are not affected by the resulting voltage distortion. However, most office buildings contain large numbers of computers or industrial plants with variable speed drives and are unlikely to meet these limits.

Appendix B

Series Parameter Values

B.1 Default Series Stability Design Parameters

Table B.1 lists the default parameters used for the frequency plots in Chapters 4 and 5 for the investigation into the effects of varying each parameter.

Symbol	Name	Value
V_{Ref}	Reference voltage	190 V_{rms}
V_L	Load voltage	190 V_{rms}
V_S	Source voltage	190 V_{rms}
K_P	Proportional gain	0.25
K_I	Integral gain	10
N	Transformer ratio	2
L_F	Filter inductance	2.0 mH
R_{LF}	Filter inductor resistance	0.05 Ω
C_F	Filter capacitance	7.5 μF
R_{CF}	Filter capacitor resistance	4.7 Ω
L_S	Source inductance	1.0 mH
R_S	Source resistance	0.5 Ω
L_{Load}	Load inductance	5 mH
R_{Load}	Load resistance	10.0 Ω

Table B.1: Default design parameters for stability model tests.

B.2 Final LV Experimental Parameters

Tables B.2, B.3 and B.4 list the parameters used for the final LV experimental system.

Symbol	Name	Value
V_{Ref}	Reference voltage	190 V_{rms}
V_L	Load voltage	190 V_{rms}
V_S	Source voltage	Varies
K_P	Proportional gain	0.25
K_I	Integral gain	10
K_{AD}	Active Damping Gain	1 Ω
N	Transformer ratio	2
L_F	Filter inductance	2.0 mH
R_{LF}	Filter inductor resistance	0.05 Ω
C_F	Filter capacitance	15 μF
R_{CF}	Filter capacitor resistance	4.7 Ω
L_S	Source inductance	>1.0 mH
R_S	Source resistance	0.5 Ω
L_{Load}	Load inductance	Varies
R_{Load}	Load resistance	Varies

Table B.2: Final parameters used for the LV experimental work.

Selected Frequency	n	K_1, n	f_C	α_0	Δ
50 Hz (Fundamental)	1	10	1.5 Hz	2^{13}	1/32 (2^{-5})
250 Hz (5^{th} Harmonic)	5	10	1.5 Hz	2^{13}	1/8 (2^{-3})
350 Hz (7^{th} Harmonic)	7	10	1.5 Hz	2^{13}	1/8 (2^{-3})
450 Hz (9^{th} Harmonic)	9	5	1.5 Hz	2^{13}	1/4 (2^{-2})
550 Hz (11^{th} Harmonic)	11	5	1.5 Hz	2^{13}	1/4 (2^{-2})
650 Hz (13^{th} Harmonic)	13	5	1.5 Hz	2^{13}	1/4 (2^{-2})

Table B.3: Parameters of the delta-based IIR digital filters used for the UCPC series feed-back compensation.

Transformer Component	Parameter	Value
Inverter side winding (Primary)	Rated voltage	240 V
	Rated current	35 A
	Rated apparent power	8.4 kVA
	Leakage inductance	842 μH
	Winding dc resistance	36 mΩ
Grid side windings (Secondary) - 2 identical windings	Rated voltage	60 V
	Rated current	60 A
	Rated apparent power	3.6 kVA
	Leakage inductance	52 μH
	Winding dc resistance	9 mΩ
Tertiary winding (Tertiary)	Rated voltage	110 V
	Rated current	10 A
	Rated apparent power	1.1 kVA
	Leakage inductance	174 μH
	Winding dc resistance	71 mΩ
Overall Transformer Parameters (Primary referred)	Magnetizing inductance	220 mH
	Magnetizing resistance	460 kΩ
	Net leakage inductance	860 μH
	Net dc resistance	144 mΩ

Table B.4: Nameplate and measured specifications of the LV experimental single-phase series injection transformers.

Appendix C

Mathematical Working

This appendix provides additional mathematical working to support the stationary reference frame signal extraction functions (or StatRF) presented in Chapter 5.

C.1 Four-Wire StatRF using Exact Transformation

The following mathematical working is for the derivation of the four–wire three-phase StatRF using a transformation based on the exact Laplace transformation developed in Chapter 5.

In the synchronous frame, the high-pass function filter

$$H_{dq}(s) = \frac{s}{s+\omega_c} \tag{C.1}$$

is applied to both the d and q signals, which removes the dc component (i.e. removes the positive sequence fundamental component). By applying the transformation developed in Chapter 5

$$[\mathbf{H}_{\alpha\beta}(s)] = \frac{1}{2}\begin{bmatrix} \begin{pmatrix} +H_{dq}(s+j\omega_0) \\ +H_{dq}(s-j\omega_0) \end{pmatrix} & \begin{pmatrix} +jH_{dq}(s+j\omega_0) \\ -jH_{dq}(s-j\omega_0) \end{pmatrix} \\ \begin{pmatrix} -jH_{dq}(s+j\omega_0) \\ +jH_{dq}(s-j\omega_0) \end{pmatrix} & \begin{pmatrix} +H_{dq}(s+j\omega_0) \\ +H_{dq}(s-j\omega_0) \end{pmatrix} \end{bmatrix} \tag{C.2}$$

to (C.1), the result is

$$[\mathbf{H}_{\alpha\beta}(s)] = \frac{1}{2}\begin{bmatrix} \left(+\frac{s+j\omega_0}{s+j\omega_0+\omega_c} + \frac{s-j\omega_0}{s-j\omega_0+\omega_c}\right) & \left(+j\frac{s+j\omega_0}{s+j\omega_0+\omega_c} - j\frac{s-j\omega_0}{s-j\omega_0+\omega_c}\right) \\ \left(-j\frac{s+j\omega_0}{s+j\omega_0+\omega_c} + j\frac{s-j\omega_0}{s-j\omega_0+\omega_c}\right) & \left(+\frac{s+j\omega_0}{s+j\omega_0+\omega_c} + \frac{s-j\omega_0}{s-j\omega_0+\omega_c}\right) \end{bmatrix}. \tag{C.3}$$

This can be broken up into the sub-functions

$$[\mathbf{H}_{\alpha\beta}(s)] = \frac{1}{2}\begin{bmatrix} \mathbf{H}_{\alpha\beta-11}(s) & \mathbf{H}_{\alpha\beta-12}(s) \\ \mathbf{H}_{\alpha\beta-21}(s) & \mathbf{H}_{\alpha\beta-22}(s) \end{bmatrix}. \tag{C.4}$$

Re-arranging these sub-functions gives

$$\begin{aligned} \mathbf{H}_{\alpha\beta-11}(s) &= \left(+\frac{(s+j\omega_0)(s-j\omega_0+\omega_c)}{(s+j\omega_0+\omega_c)(s-j\omega_0+\omega_c)} + \frac{(s-j\omega_0)(s+j\omega_0+\omega_c)}{(s-j\omega_0+\omega_c)(s+j\omega_0+\omega_c)}\right) \quad \text{(C.5)} \\ &= \left(+\frac{s^2+\omega_c s+\omega_0^2+j\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} + \frac{s^2+\omega_c s+\omega_0^2-j\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}\right) \\ &= \frac{2s^2+2\omega_c s+2\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} \\ &= \mathbf{H}_{\alpha\beta-22}(s) \end{aligned}$$

and

$$\begin{aligned} \mathbf{H}_{\alpha\beta-12}(s) &= \left(+j\frac{(s+j\omega_0)(s-j\omega_0+\omega_c)}{(s+j\omega_0+\omega_c)(s-j\omega_0+\omega_c)} - j\frac{(s-j\omega_0)(s+j\omega_0+\omega_c)}{(s-j\omega_0+\omega_c)(s+j\omega_0+\omega_c)}\right) \quad \text{(C.6)} \\ &= \left(+\frac{js^2+j\omega_c s+j\omega_0^2-\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} - \frac{js^2+j\omega_c s+j\omega_0^2+\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}\right) \\ &= \frac{-2\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} \\ &= -\mathbf{H}_{\alpha\beta-21}(s) \end{aligned}$$

Combining these sub-functions back into matrix form, gives

$$[\mathbf{H}_{\alpha\beta}(s)] = \begin{bmatrix} \frac{s^2+\omega_c s+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} & \frac{-\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} \\ \frac{\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} & \frac{s^2+\omega_c s+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} \end{bmatrix} \tag{C.7}$$

which is the four-wire form of the StatRF in the stationary α-β reference frame presented in Chapter 5.

C.2 Three-Wire StatRF using Exact Transformation

For a three-wire three-phase system a reduced α-$\beta \longrightarrow a$-b transformation matrix is used to transform the α-β reference frame StatRF (C.7) into the abc reference frame (where the c-phase is redundant). The result is then simplified to achieve the form shown in Chapter 5. This is shown as follows:

$$\begin{aligned} [\mathbf{H}_{ab}(s)] &= \begin{bmatrix} 0 & 1 \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{s^2+\omega_c s+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} & \frac{-\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} \\ \frac{\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} & \frac{s^2+\omega_c s+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} \left(\frac{s^2+\omega_c s+\omega_0^2}{\sqrt{3}(s^2+2\omega_c s+\omega_c^2+\omega_0^2)} - \frac{\omega_0\omega_c}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}\right) & \left(\frac{2s^2+2\omega_c s+2\omega_0^2}{\sqrt{3}(s^2+2\omega_c s+\omega_c^2+\omega_0^2)}\right) \\ \left(\frac{\omega_0\omega_c}{\sqrt{3}(s^2+2\omega_c s+\omega_c^2+\omega_0^2)} + \frac{s^2+\omega_c s+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2}\right) & \left(\frac{2\omega_0\omega_c}{\sqrt{3}(s^2+2\omega_c s+\omega_c^2+\omega_0^2)}\right) \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{s^2+\omega_c s-\sqrt{3}\omega_0\omega_c+\omega_0^2}{\sqrt{3}(s^2+2\omega_c s+\omega_c^2+\omega_0^2)} & \frac{2s^2+2\omega_c s+2\omega_0^2}{\sqrt{3}(s^2+2\omega_c s+\omega_c^2+\omega_0^2)} \\ \frac{s^2+\omega_c s+\frac{\omega_0\omega_c}{\sqrt{3}}+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} & \frac{2\omega_0\omega_c}{\sqrt{3}(s^2+2\omega_c s+\omega_c^2+\omega_0^2)} \end{bmatrix} \\ &= \begin{bmatrix} \frac{s^2+\omega_c s+\frac{\omega_0\omega_c}{\sqrt{3}}+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} & \frac{2\omega_0\omega_c}{\sqrt{3}(s^2+2\omega_c s+\omega_c^2+\omega_0^2)} \\ \frac{s^2+\omega_c s-\sqrt{3}\omega_0\omega_c+\omega_0^2}{2(s^2+2\omega_c s+\omega_c^2+\omega_0^2)} - \frac{s^2+\omega_c s+\frac{\omega_0\omega_c}{\sqrt{3}}+\omega_0^2}{2(s^2+2\omega_c s+\omega_c^2+\omega_0^2)} & \frac{s^2+\omega_c s+\omega_0^2}{(s^2+2\omega_c s+\omega_c^2+\omega_0^2)} - \frac{\omega_0\omega_c}{\sqrt{3}(s^2+2\omega_c s+\omega_c^2+\omega_0^2)} \end{bmatrix} \\ &= \begin{bmatrix} \frac{s^2+\omega_c s+\frac{\omega_0\omega_c}{\sqrt{3}}+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} & \frac{\frac{2\omega_0\omega_c}{\sqrt{3}}}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} \\ \frac{-\frac{2\omega_0\omega_c}{\sqrt{3}}}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} & \frac{s^2+\omega_c s-\frac{\omega_0\omega_c}{\sqrt{3}}+\omega_0^2}{s^2+2\omega_c s+\omega_c^2+\omega_0^2} \end{bmatrix} \end{aligned} \tag{C.8}$$

C.3 Single-Phase StatRF using Approximate Transformation

The following mathematical working is for the derivation of the single-phase StatRF using the approximate low-pass to band-pass transformation. (Note that the derivation of the single-phase StatRF using the combination of the positive and negative sequence StatRF is already given in Chapter 5)

By applying the transformation

$$H_{abc}(s) = H_a(s) = H_b(s) = H_c(s) = H_{dq}\left(\frac{s^2+\omega_0^2}{2s}\right) \tag{C.9}$$

to (C.1) the result is

$$H_{abc}(s) = \frac{\left(\frac{s^2+\omega_0^2}{2s}\right)}{\left(\frac{s^2+\omega_0^2}{2s}\right)+\omega_c}. \tag{C.10}$$

Rearranging gives

$$H_{abc}(s) = \frac{\left(s^2+\omega_0^2\right)}{\left(s^2+\omega_0^2\right)+2\omega_c s} \tag{C.11}$$

and finally the result shown in Chapter 5 is obtained

$$H_{abc}(s) = \frac{s^2+\omega_0^2}{s^2+2\omega_c s+\omega_0^2}. \tag{C.12}$$

which is simply a narrowband notch.

Appendix D

Series DSP Software

D.1 pwmint.c: C Interrupt Code

```
/************************************************************************/
/*                                                                      */
/* Application: Series VSI interrupt using variable switching frequency */
/* Developed By:Monash University: Power Electronics Group              */
/* Author:      M.Newman and A.McIver                                   */
/* File:        pwmint.c                                                */
/* Processor:   TI TMS320F240 (MiniDSP / CS-IIB)                        */
/* History:                                                             */
/*      13/05/99 AM - modified from lpiint.c                            */
/*      03/06/99 AM - moved sine table defintions to tables.c           */
/*      08/07/99 AM - correcting a comment in SetMag()                  */
/*      20/07/99 AM - correcting PWM init bug                           */
/*      17/02/00 ZM - added phase D (neutral) control                   */
/*      03/03/00 AM - corrected 4th leg initialization                  */
/*      21/11/00 MN - PWM Fast Stop altered to suit series injection    */
/*      22/11/00 MN - PDPINT structure forced in interrupt              */
/*      23/11/00 MN - PWM_fast_stop altered to force a null state on PDPINT*/
/*      12/12/00 MN - Zero crossing code added                          */
/*      19/01/01 MN - Debug Variable Grabbing Software Added            */
/*      13/02/01 MN - XINT2 and XINT3 added for fault protection        */
/*      14/02/01 MN - Analog Conversion Software added                  */
/*      16/02/01 MN - P+Resonant Software Added                         */
/*      20/02/01 MN - "period" variable removed and replaced by period_2*2 */
/*      21/02/01 MN - Control Code moved to a macro in control.h        */
/*      16/10/01 MN - Altered software to work with CS-IIB/MiniDSP board */
/*      27/11/01 MN - ZX captures altered for CS-IIB/Series Implementation */
/*      13/12/01 MN - All fourth phase leg code removed for safety reasons */
/*      19/02/02 MN - Feedforward injection software now in and working */
/*      19/04/02 MN - Voltage error slew limit added to stop LC resonance */
/*      22/04/02 MN - 5th Harmonic Load Voltage P+Resonant feedback added */
/*      23/04/02 MN - 7th/11th/13th Harmonic VL P+Resonant feedback added */
/*      24/04/02 MN - Timing optimisation done to speed up code         */
/*      24/04/02 MN - Control.h and other legacy series software removed */
/*      29/04/02 MN - Further optimisations on resonant filters implemented*/
/*      04/03/03 MN - Feedback fundamental P+R added and tested         */
/*      12/06/03 MN - Calabrated analog offset values ready for final tests*/
```

```
/*      12/06/03 MN - Active Damping current feed-back added                */
/*      15/06/03 MN - Proportional voltage feed-back added                  */
/*                                                                          */
/****************************************************************************/

#include <mu_pcb.h>                  /* Defines Board Specific Variables  */
#include <c240.h>
#include <conio.h>
#include <minibus.h>                 /* Defines DAC interface information */
#include <adc.h>                     /* Defines A/D Conversion Values     */
#include <iib.h>
#include <fastdiv.h>                 /* Used for DC bus voltage division  */
#include "sintab.h"                  /* Lookup table used for trig. funct.*/
#include "series.h"                  /* Defines Series VSI Specific Values*/
#include "filters.h"

/****************************************************************************/
/*                          VARIABLE DECLARATIONS                           */
/****************************************************************************/

unsigned int
    phase_step,
    fault,
    is_modulating,
    int2_vect,
    int4_vect,
    index,
    phase,
    phase_trim,
    period_2,           /* Half a period                                    */
    V_Asat = 0,
    V_Bsat = 0,
    V_Csat = 0;

signed int
    V_A, V_B, V_C,      /* Demanded voltages                                */
    t_A, t_B, t_C, t_D, /* Phase Switching times                            */
    Voff,               /* Offset for third harmonics                       */
    max_time,
    mag,
    VsmeasBA, VsmeasCA, /* Supply Voltage Vs Measurement Variables          */
    VsmeasA,  VsmeasB,  /* Supply Voltage Vs Phase Measurement Equivalent   */
    VlmeasBA, VlmeasCA, /* Load Voltage Vl Measurement Variables            */
    VlmeasA,  VlmeasB,  /* Load Voltage Vl Phase Measurement Equivalent     */
    I_A, I_B,           /* Inverter output current measurements             */
    Vdc,                /* DC Bus Voltage measured variable                 */
    Vdc_Invert,         /* Scaled and Inverted DC Bus Voltage variable      */
    ImeasA,   ImeasB,   /* Inverter Output Current Measurement Variables    */
    VrefA,    VrefB,    /* Demanded Reference Voltages                      */
    VerrA=0,  VerrB=0,  /* Voltage error between reference and measured     */
    VlerrA=0, VlerrB=0, /* Voltage error between reference and measured     */
    VerrA_1,  VerrB_1,  /* Previous sample's voltage error values           */
    kp=0,     ki=0,     /* Proportional and Integral Gain Constants         */
    Vs_sum,             /* Average L-L Source Voltage ADC Value             */
    Vs_sum_tmp,         /* L-L Source Voltage ADC Value Summation           */
    Vdc_remove=FALSE;   /* Flag for enabling Vdc voltage removal            */
```



```
/* Zero Crossing Variables                                                  */
unsigned int
    in_sync,        /* Flag to indicate that snc is achieved                */
    ZX_in_sync,     /* > ZX_SYNC_LIMIT means that sync has been achieved    */
    ZX_state,       /* State of the zero crossing synchronization process   */
    ZX_count,       /* The number of switching cycles between ZX interrupts*/
    ZX_seen,        /* Set to TRUE when a zero crossing is detected         */
    ZX_cycles,      /* Count of number of ZXs during averaging              */
    ZX_sum;         /* Running sum for average                              */

signed int
    ZX_time,        /* Time of captured ZX in timer units                   */
    ZX_time_phase,  /* Time of captured ZX in phase units                   */
    ZX_phase_scale, /* Scale factor between timer and phase units           */
    ZX_phase_err,   /* Difference in phase units (2^16 == 360deg)           */
    ZX_err_sum;     /* Integral for frequency control                       */

/* Debug Variables                                                          */
signed int
    DACTest,            /* Toggle Variable for calabrating DAC Outpus       */
    spare,              /* Spare temporary variable for occasional use      */
    g_con = GRAB_READY, /* Grab control variable                            */
    g_sel = 0,          /*                                                  */
    g_idx = 0;          /*                                                  */

/* Assembler IIR Shift and Delta Filter Variables                           */
signed int
    Res_Array[HARM_NUM*2*6],        /* 2 phases, 6 variables per filter     */
    *PRA_5th  =  Res_Array,         /* Variables: [ y   x   x2  x1  x0  * ]*/
    *PRB_5th  = (Res_Array+ 6),
    *PRA_7th  = (Res_Array+12),
    *PRB_7th  = (Res_Array+18),
    *PRA_9th  = (Res_Array+24),
    *PRB_9th  = (Res_Array+30),
    *PRA_11th = (Res_Array+36),
    *PRB_11th = (Res_Array+42),
    *PRA_13th = (Res_Array+48),
    *PRB_13th = (Res_Array+54);

signed int
    PRes_1st[2*6],
    *PRA_1st  =  PRes_1st,                  /* [ y   x   x2  x1  x0  * ]   */
    *PRB_1st  = (PRes_1st+6);               /* [ y   x   x2  x1  x0  * ]   */

/* Controller Choice Variables */
signed int
    ON_5th  = FALSE,    /* Determines whether 5th harm. VL feedback is ON  */
    ON_7th  = FALSE,    /* Determines whether 7th harm. VL feedback is ON  */
    ON_9th  = FALSE,    /* Determines whether 9th harm. VL feedback is ON  */
    ON_11th = FALSE,    /* Determines whether 11th harm. VL feedback is ON */
    ON_13th = FALSE,    /* Determines whether 13th harm. VL feedback is ON */
    ON_damp = FALSE,    /* Determines whether active damping is ON         */
    Proportional = FALSE,/* Determines whether proportional feedback is ON */
    Feedforward = TRUE, /* Determines whether Source Verr feedforward is ON*/
    Feedback    = FALSE;/* Determines whether Load fund. feedback is ON    */
```

```
/* Found in tables.c                                                    */
extern int sin_table_A[TABLE_SIZE];
extern int sin_table_B[TABLE_SIZE];


/*************************************************************************/
/*                          FUNCTION DEFINITIONS                         */
/*************************************************************************/

/*************************************************************************/
/*  NAME:        INT2Service()                                           */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Interrupt for EV Timer and PDPINT                       */
/*  NOTES:       coltrollers, pwm calculations and the loading compare   */
/*               registers is perfomed in this interrupt.                */
/*  HISTORY:     12/12/2000  MN  ZX Code added                           */
/*               14/02/2001  MN  Test for tripped Triac added            */
/*               16/02/2001  MN  P+Resonant Voltage Controller Added     */
/*               20/02/2001  MN  Simultaneous Timer 1 & 2 Update Included*/
/*               16/10/2001  MN  Altered to work with CS-IIB/MiniDSP board*/
/*               13/12/2001  MN  Fourth phase leg code removed           */
/*               19/02/2002  MN  Feedforward injection software implemented*/
/*               19/04/2002  MN  Verr slew limit added to stop LC resonance*/
/*               22/04/2002  MN  5th Harmonic VL P+Resonant feedback added*/
/*               23/04/2002  MN  7th/11th/13th Harm. VL P+R feedback added*/
/*               24/04/2002  MN  Timing optimisation done to speed up code*/
/*               29/04/2002  MN  Further optimisations on resonant filters*/
/*               03/03/2003  MN  ZX Parameters changed to improve robustnes*/
/*               04/03/2003  MN  Feedback fundamental P+R added and tested*/
/*************************************************************************/
interrupt void INT2Service(void)
{
    int2_vect = reg(EVIVRA);    /* Reading this register seems to clear it */

/*  Timing_R2_on();*/

    if (int2_vect == VECT_PDPINT)               /* Check for PDPINT fault  */
    {
        fault |= GATE_FAULT;
        PWM_fault_stop();
    }

    if (int2_vect == VECT_T1PINT)       /* ( 0.3us -> 1.6us )              */
    {
        reg(T2CMPR) = 3*period_2/2;     /* 25% of time at end of interrupt */
                                        /* left for acquistion. Min = 4us  */
        reg(T1PR)   =  period_2*2 ;     /* Both timer values need to be    */
        reg(T2PR)   = (period_2*2) - 1;/* updated simultaneously to stop   */
                                        /* them from going out of sync.    */
    }

    if ( (TriacON()) && (is_modulating==TRUE) )/*MOV fault trip check (2us)*/
    {
        fault |= TRIAC_TRIP;
        PWM_fault_stop();
```



```
    }

    /* Second ADC conversion starts when first finishes                 */
    reg(ADCTRL1) = ADCEMULATOR|ADC1EN|ADC2EN|ADC_VAC1|ADC_VAC2|ADCSTART;


    /* Zero crossing processing                                         */
    ZX_count++;
    if (ZX_count > ZX_MAX_COUNT) /* Zero crossing signal lost           */
    {
        PWM_fast_stop();         /* Halt modulation                     */
        fault |= LOST_SYNC;
        ZX_state   = ZX_LOST;    /* Restart searching for sync          */
        ZX_in_sync = 0;
        ZX_count   = 0;
    }
    if (ZX_state == ZX_LOST)     /* No idea of anything : start freq est. */
    {
        fault |= LOST_SYNC;
        if (ZX_seen == TRUE)
        {
            ZX_cycles = 0;
            ZX_sum    = 0;
            ZX_count  = 0;
            ZX_seen   = FALSE;
            ZX_state  = ZX_EST;
        }
    }
    else if (ZX_state == ZX_EST) /* Roughly measure period and average  */
    {
        if (ZX_seen == TRUE)
        {
            ZX_cycles++;
            ZX_sum  += ZX_count;
            ZX_count = 0;                /* Reset counter              */
            ZX_seen  = FALSE;
        }
        if (ZX_cycles >= ZX_CYCLE_AVG)
        {
            ZX_sum = ZX_sum/ZX_CYCLE_AVG;
            phase_step = 0xffff/ZX_sum; /* Approximate frequency       */
            ZX_sum -= ZX_sum/16;        /* Also use for glitch filter  */
            phase = phase_step + ZX_OFFSET + phase_trim; /* within phase_step */
            ZX_state = ZX_MISC;         /* Calculate ZX_phase_scale first */
        }
    }
    else if (ZX_state == ZX_SYNC)       /* Accurately measure phase error */
    {
        if (ZX_seen == TRUE)
        {
            ZX_seen = FALSE;
            if (ZX_count > ZX_sum)      /* Ignore glitches             */
            {
                ZX_count = 0;
                /* Rescale to phase units                               */
                ZX_time_phase = ZX_OFFSET + phase_trim + (((ZX_time>>2)*ZX_phase_scale)>>3);
```

```
            /* Calculate phase error captured time                    */
            ZX_phase_err = phase - ZX_time_phase;
            /* Limit size of phase change                             */
            if (ZX_phase_err > ZX_BIG_ERR)
            {
                phase -= ZX_BIG_ERR;
                ZX_err_sum = (ZX_err_sum+ZX_BIG_ERR)>>1;/* Integrate   */
            }                                          /* phase errors*/
            else if (ZX_phase_err < -ZX_BIG_ERR)
            {
                phase += ZX_BIG_ERR;
                ZX_err_sum = (ZX_err_sum-ZX_BIG_ERR)>>1;
            }
            else
            {
                phase -= ZX_phase_err;
                ZX_err_sum = (ZX_err_sum+ZX_phase_err)>>1;
            }
            ZX_state = ZX_FREQ;
        }
    }
}
else if (ZX_state == ZX_FREQ)    /* Nudge frequency if needed          */
{

    if (ZX_err_sum > ZX_FREQ_ERR)/*If too large nudge freq (phase_step)*/
    {
        phase_step--;
    }
    else if (ZX_err_sum < -ZX_FREQ_ERR)
    {
        phase_step++;
    }
    ZX_state = ZX_LOCK;
}
else if (ZX_state == ZX_LOCK)    /* Tost to see if still in sync       */
{
    if (ZX_in_sync >= ZX_SYNC_LIMIT)
    {
        if ((ZX_phase_err>ZX_PHASE_ERR)||(ZX_phase_err<-ZX_PHASE_ERR))
        {                            /* Gone out of sync               */
            PWM_fast_stop();
            ZX_in_sync = 0;
            in_sync = FALSE;
            fault |= LOST_SYNC;
        }
        else
        {
            in_sync = TRUE;
        }
    }
    else if ((ZX_phase_err<ZX_PHASE_ERR)&&(ZX_phase_err>-ZX_PHASE_ERR))
    {                            /* In sync this cycle                 */
        ZX_in_sync++;
    }
    else
```



```
    {
        ZX_in_sync = 0;
    }
    ZX_state = ZX_MISC;
}
else if (ZX_state == ZX_MISC)
{
    ZX_phase_scale = (phase_step<<5)/(period_2*2);/*THIS IS SLOW!(~8us)*/
    ZX_state = ZX_SYNC;
}

Delta_Neg1(PRA_1st,5);          /* Delta var. shuffle for A/B phase */
Delta_Neg1(PRB_1st,5);          /* P+ Res. controllers (800ns each) */

Delta_Neg1(PRA_5th,3);          /* Delta var. shuffle for A/B phase */
Delta_Neg1(PRB_5th,3);          /* P+ Res. controllers (800ns each) */

Delta_Neg1(PRA_7th,3);          /* Delta var. shuffle for A/B phase */
Delta_Neg1(PRB_7th,3);          /* P+ Res. controllers (800ns each) */

Delta_Neg1(PRA_9th,2);          /* Delta var. shuffle for A/B phase */
Delta_Neg1(PRB_9th,2);          /* P+ Res. controllers (800ns each) */

Delta_Neg1(PRA_11th,2);         /* Delta var. shuffle for A/B phase */
Delta_Neg1(PRB_11th,2);         /* P+ Res. controllers (800ns each) */

Delta_Neg1(PRA_13th,2);         /* Delta var. shuffle for A/B phase */
Delta_Neg1(PRB_13th,2);         /* P+ Res. controllers (800ns each) */


VerrA_1 = VerrA;
VerrB_1 = VerrB;

phase += phase_step;                        /* Update phase angle     */
index = phase>>6;

max_time = period_2 - (DEADCNT+PULSECNT); /* Update the switching freq.*/

/* Calculate A and B phase voltage references                         */
VrefA = ((long)sin_table_A[index]*(long)(mag<<SHIFT3))>>16;
VrefB = ((long)sin_table_B[index]*(long)(mag<<SHIFT3))>>16;

while(!ADC_ready());              /* Check analog conversion is ready */
/* First ADC Results - Source Voltage                                 */
VsmeasBA = VSBA_OFFSET - get_ADC1(); /* Volt. Measurements Inverted on IIB*/
VsmeasCA = VSCA_OFFSET - get_ADC2();
VsmeasA  =(-VsmeasBA   -VsmeasCA)<<SHIFT3;/* Convert to phase voltages */
VsmeasB  =( VsmeasBA*2 -VsmeasCA)<<SHIFT3;/* with added scaling shift  */


/* Third ADC conversion starts when second finishes                   */
reg(ADCTRL1) = ADCEMULATOR|ADC1EN|ADC2EN|ADC_I1|ADC_I2|ADCSTART;

/* Calculate DC voltage scaling factor                                */
if ((Vdc<V_DC_MIN) && is_modulating && (!Vdc_remove)) /* Test low Vdc  */
{
```

```
    fault |= LOW_DC_VOLTS;
    PWM_fault_stop();
}
else if (Vdc > V_DC_MAX)                /* Test for high volts          */
{
    fault |= HIGH_DC_VOLTS;
    PWM_fault_stop();
}
else
{

    Vdc_Invert = ((signed int)fastdivu(SCALE1, Vdc))<<SHIFT2;
}

if (VsmeasA > 0 )                       /* Source Volt. Average Value  */
    Vs_sum_tmp += (VsmeasA)>>(SHIFT3+2);
else
    Vs_sum_tmp -= (VsmeasA)>>(SHIFT3+2);


while(!ADC_ready());                /* Check analog conversion is ready */

/* Second ADC Results - Load Voltage                                    */
VlmeasBA = VLBA_OFFSET - get_ADC1(); /* Volt. Measurements Inverted on IIB*/
VlmeasCA = VLCA_OFFSET - get_ADC2();
VlmeasA  =(-VlmeasBA   -VlmeasCA)<<SHIFT3;/* Convert to phase voltages */
VlmeasB  =( VlmeasBA*2 -VlmeasCA)<<SHIFT3;/* with added scaling shift  */


/* Fourth ADC conversion starts when third finishes                     */
reg(ADCTRL1) = ADCEMULATOR|ADC1EN|ADC2EN|ADC_VDC1|ADC_VDC2|ADCSTART;


if(Feedforward) {
    VerrA = (VrefA - VsmeasA);     /* Calculate A & B phase errors      */
    VerrB = (VrefB - VsmeasB);
}
else {
    VerrA = 0;
    VerrB = 0;
}

/* Clamp maximum swing to minimise series inverter LC filter resonance */
if ( (VerrA - VerrA_1) >  200 ) VerrA = VerrA_1 + 200;
if ( (VerrA - VerrA_1) < -200 ) VerrA = VerrA_1 - 200;
if ( (VerrB - VerrB_1) >  200 ) VerrB = VerrB_1 + 200;
if ( (VerrB - VerrB_1) < -200 ) VerrB = VerrB_1 - 200;


VlerrA = VrefA - VlmeasA;      /* Calculate A & B phase errors         */
PRA_5th[1]  = VlerrA;          /* Harmonic Filter Input Assignments    */
PRA_7th[1]  = VlerrA;
PRA_9th[1]  = VlerrA;          /* Input assignments and VlerrX         */
PRA_11th[1] = VlerrA;          /* creation takes 1.2us per phase       */
PRA_13th[1] = VlerrA;
```



```
VlerrB = VrefB - VlmeasB;
PRB_5th[1]  = VlerrB;
PRB_7th[1]  = VlerrB;
PRB_9th[1]  = VlerrB;
PRB_11th[1] = VlerrB;
PRB_13th[1] = VlerrB;


if(!ON_5th) {
    PRA_5th[1] = 0;
    PRB_5th[1] = 0;
}

if(!ON_7th) {
    PRA_7th[1] = 0;
    PRB_7th[1] = 0;
}

if(!ON_9th) {
    PRA_9th[1] = 0;
    PRB_9th[1] = 0;
}

if(!ON_11th) {
    PRA_11th[1] = 0;
    PRB_11th[1] = 0;
}

if(!ON_13th) {
    PRA_13th[1] = 0;
    PRB_13th[1] = 0;
}

Delta_All(Res_Array);  /* This function calculates all of the harmonic */
                       /* resonant filters for both A & B phases       */
                       /* Total execution time for 5/7/9/11/13th =14us */

if(ON_5th) {        /* 0.5us each when ON */
    VerrA += (PRA_5th[0]<<1);
    VerrB += (PRB_5th[0]<<1);
}

if(ON_7th) {
    VerrA += (PRA_7th[0]<<1);
    VerrB += (PRB_7th[0]<<1);
}

if(ON_9th) {
    VerrA += (PRA_9th[0]);
    VerrB += (PRB_9th[0]);
}

if(ON_11th) {
    VerrA += (PRA_11th[0]);
    VerrB += (PRB_11th[0]);
```

```
}

if(ON_13th) {
    VerrA += (PRA_13th[0]>>1);
    VerrB += (PRB_13th[0]>>1);
}


/* P+R Voltage Feedback Controller */
if(Feedback) {

    PRA_1st[1] = (VlerrA>>1);
    PRB_1st[1] = (VlerrB>>1);

    Delta_2nd_x2(PRes_1st);

    VerrA += (PRA_1st[0]);
    VerrB += (PRB_1st[0]);
}
else
{
    PRA_1st[1] = 0;
    PRB_1st[1] = 0;

    Delta_2nd_x2(PRes_1st);
}


/* Proportional Voltage Feed-back */
if(Proportional) {

    VerrA += (VlerrA>>3);                              /* Kp = 0.25   */
    VerrA += (VlerrA>>3);
}

/* Third ADC Results - Inverter Currents for Active Damping           */
I_A = I1_OFFSET - get_ADC1();
I_B = I2_OFFSET - get_ADC2();

if(ON_damp) {

    VerrA += I_A>>1;
    VerrB += I_B>>1;
}

/* Calculate A & B phase controller outputs                           */
Clamp(VerrA, 10000);              /* Clamp size of the error (C macro) */
Clamp(VerrB, 10000);              /*                                   */

/* Determine target switching times                                   */
if(Vdc_remove)
{
    V_A = ((long)sin_table_A[index]*(long)(mag<<2))>>16;
    t_A = ((long)V_A*(long)(period_2<<3))>>16;
    V_B = ((long)sin_table_B[index]*(long)(mag<<2))>>16;
    t_B = ((long)V_B*(long)(period_2<<3))>>16;
```



```
}
else
{
    V_A = ((long)VerrA*(long) Vdc_Invert  )>>16;
    t_A = ((long)( V_A<<SHIFT4 )*(long)( (int)(period_2<<4) ))>>16;
    V_B = ((long)VerrB*(long) Vdc_Invert  )>>16;
    t_B = ((long)( V_B<<SHIFT4 )*(long)( (int)(period_2<<4) ))>>16;
}

t_C = -t_A - t_B;                  /* Calculate t_C from t_A and t_B    */


/* Calculate t_C from t_A and t_B */
t_C = -t_A - t_B;

if (t_A > t_B)
{
    if (t_A > t_C)
    {
        if (t_B > t_C)  Voff = t_B>>1;
        else            Voff = t_C>>1;
    }
    else
    {
        Voff = t_A>>1;
    }
}
else
{
    if (t_B > t_C)
    {
        if (t_A > t_C)  Voff = t_A>>1;
        else            Voff = t_C>>1;
    }
    else
    {
        Voff = t_B>>1;
    }
}

/* Add offset into t_A, t_B and t_C */
t_A = t_A + Voff;
t_B = t_B + Voff;
t_C = t_C + Voff;

/* Clamp switch times for pulse deletion and saturation               */
if (t_A > max_time)                                        /* A phase */
{
    reg(CMPR1) = 0;
}
else if (t_A < -max_time)
{
    if (!V_Asat && (int2_vect == VECT_T1UFINT))
        reg(CMPR1) = (period_2<<1) - 1;
    else
        reg(CMPR1) = (period_2<<1);
```

```
    V_Asat = 1;
}
else
{
    if (V_Asat && (int2_vect == VECT_T1UFINT))
        reg(CMPR1) = (period_2<<1);
    else
        reg(CMPR1) = period_2 - t_A;
    V_Asat = 0;
}

if (t_B > max_time)                                    /* B phase */
{
    reg(CMPR2) = 0;
}
else if (t_B < -max_time)
{
    if (!V_Bsat && (int2_vect == VECT_T1UFINT))
        reg(CMPR2) = (period_2<<1) - 1;
    else
        reg(CMPR2) = (period_2<<1);
    V_Bsat = 1;
}
else
{
    if (V_Bsat && (int2_vect == VECT_T1UFINT))
        reg(CMPR2) = (period_2<<1);
    else
        reg(CMPR2) = period_2 - t_B;
    V_Bsat = 0;
}

if (t_C > max_time)                                    /* C phase */
{
    reg(CMPR3) = 0;
}
else if (t_C < -max_time)
{
    if (!V_Csat && (int2_vect == VECT_T1UFINT))
        reg(CMPR3) = (period_2<<1) - 1;
    else
        reg(CMPR3) = (period_2<<1);
    V_Csat = 1;
}
else
{
    if (V_Csat && (int2_vect == VECT_T1UFINT))
        reg(CMPR3) = (period_2<<1);
    else
        reg(CMPR3) = period_2 - t_C;
    V_Csat = 0;
}


if (g_con == GRAB_GO)                          /* Debug Grabbing Software */
{
```



```
    if (g_idx < G_COUNT)
    {
        g[g_idx][0] = VlmeasA;
        g[g_idx][1] = VerrA;
        g[g_idx][2] = VlmeasB;
        g[g_idx][3] = VerrB;
        g[g_idx][4] = VrefA;
        g[g_idx][5] = V_A;
        g[g_idx][6] = VsmeasA;
        g[g_idx][7] = Vdc_Invert;
        g[g_idx][8] = t_A;
        g[g_idx][9] = Vdc;
        g[g_idx][10]= VsmeasBA;
        g[g_idx][11]= VsmeasCA;
        g[g_idx][12]= VlmeasBA;
        g[g_idx][13]= VlmeasCA;
        g[g_idx][14]= I_A;
        g[g_idx][15]= I_B;
        g[g_idx][16]= fault;
        g_idx++;
    }
    else
    {
        g_con = GRAB_DONE;
        g_idx = 0;
    }
}

/* Forth ADC Results - DC Bus Voltage                                    */
spare = get_ADC1();              /* Ignore first DC bus                  */
Vdc   = get_ADC2();              /* Used Vdc2 for DC bus measurement     */
Vdc  -= ADC_ZERO;

reg(ADCTRL1) = ADCEMULATOR|ADC1EN|ADC2EN|ADC_VAC4|ADC_VAC5;

/*  Timing_R2_off();*/

} /* End timer1 interrupt */


/*************************************************************************/
/*  NAME:        IN42Service()                                           */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Interrupt for Zero Crossing Detection                   */
/*  NOTES:                                                               */
/*  HISTORY:     13/12/2000  MN  Created from my 1999 LPI ActiveFilter Code*/
/*               27/11/2001  MN  Altered for CS-IIB Implementation       */
/*************************************************************************/
interrupt void INT4Service(void)
{
    int4_vect = reg(EVIVRC);
    ZX_seen = TRUE;

#if   (CAP_PORT == CAP_PORT3)
    ZX_time = (period_2*2) - reg(CAP3FIFO);     /* ZX from V1           */
#elif (CAP_PORT == CAP_PORT4)
```

```
    ZX_time = (period_2*2) - reg(CAP4FIFO);     /* ZX from V4              */
#endif

    Vs_sum = Vs_sum_tmp;
    Vs_sum_tmp = 0;

} /* End INT4Service */


/*************************************************************************/
/*  NAME:        PWM_Init()                                              */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Sets up registers for PWM                               */
/*  NOTES:                                                               */
/*  HISTORY:     04/12/2000  MN  Added Series Injection Settings         */
/*               27/11/2001  MN  Capture ports init changed to suit CS-IIB */
/*               13/12/2001  MN  Fourth phase leg code removed           */
/*               29/04/2002  MN  Resonant Filter zeroing optimized       */
/*************************************************************************/
void PWM_Init(void)
{
    int i;

    /* Calculate sin tables */
    sin_table(sin_table_A,TABLE_SIZE, 32767,    0   , 0);
    sin_table(sin_table_B,TABLE_SIZE, 32767, -21845, 0);

    /* set up registorsfor ADC measurements */
    reg(ADCTRL1) = ADCEMULATOR|ADC1EN|ADC2EN|ADC_VAC4|ADC_VAC5;
    reg(ADCTRL2) = ADCEVSOC|ADCPSCALE; /*0x0403;*/

    is_modulating = FALSE;
    fault = NO_FAULT;
    Clear_Gate_Fault();

    reg(CAPFIFO) = 0x00ff;/* Clear the fifo capture inputs                */

    reg(COMCON) = 0x2b17; /* Disable full compare - un-shadows registers  */
    reg(T1CON)  = 0xa806; /* Stop timer                                   */

    reg(ACTR)   = 0x0CCC; /* Force low gates on, high gates off           */
    reg(SACTR)  = 0x0000; /* Force PWM outputs low                        */

    period_2    = 1000;

    reg(GPTCON) = 0x047a; /* 000 00 01 00 1 11 10 10                      */
     /* timer control       |   |  |  |  | |  |  +- T1CMPR active high
      *                     |   |  |  |  | |  +- T2CMPR active high
      *                     |   |  |  |  | +- T3CMPR forced high
      *                     |   |  |  |  +- compare output enable
      *                     |   |  |  +- T1toADC
      *                     |   |  +- T2toADC starts on period
      *                     |   +- T3toADC
      *                     +- timer status bits                         */

    /* Action control register */
```



```
    reg(ACTR)  = 0x0CCC; /* Force low gates on, high gates off           */


    /* simple action control register */

    reg(SACTR)  = 0x0000; /* 0000 0000 00 00 00 00                       */
     /*                      \       /  |  |  +--- PWM7/CMP7 off
      *                       \   /     |  +------ PWM8/CMP8 off
      *                        \/       +- sample and hold active high
      *                         - reserved                               */


    /* deadband timer control */
    reg(DBTCON) = 0x00e0|(DEADCNT<<8); /* xxxx xxxx 111 00 000          */
     /*                                  |       | |   |  +- reserved
      *                                  \     /  |   +- prescaler
      *                                    \ /    +- enabled all pins
      *                                     +- deadband time            */

    reg(CMPR1)  = period_2;
    reg(CMPR2)  = period_2;
    reg(CMPR3)  = period_2;
    reg(SCMPR1) = 0;
    reg(SCMPR2) = 0;
    reg(SCMPR3) = 0;

    /* compare control register */
    reg(COMCON) = 0x2b17; /* 0 01 0 10 1 1 0 00 10 111                   */
    reg(COMCON) = 0xab17; /* 1 01 0 10 1 1 0 00 10 111                   */
     /*                     | |  | |  | | | |  |  +- pins in pwm mode
      *                     | |  | |  | | | |  +- reload SACTR immediately
      *                     | |  | |  | | | +- reload SCMPRx on zero or period
      *                     | |  | |  | | +- use timer1
      *                     | |  | |  | +- simple compare enabled
      *                     | |  | |  +- full compare enabled
      *                     | |  | +- reload ACTR immediately
      *                     | |  +- space vector mode disabled
      *                     | +- CMPRx reload on zero or period
      *                     +- compare enable                            */

    reg(T2PR) = (2 * period_2)-1; /* Timer 2 added for control of S'n'H  */
    reg(T2CNT) = 0;

    reg(T2CMPR) = 3*period_2/2;   /* 25% of time at end of interrupt left */
                                  /*  for acquistion. Datasheet min = 4us */

    reg(T2CON) = 0x9082; /* 10 010 000 1 0 00 00 1 0 */
     /* timer 2            | |   |   | | | |  | +- reserved
      *                    | |   |   | | | |  +- timer compare enable
      *                    | |   |   | | | +- timer compare reload on zero
      *                    | |   |   | | +- clock source - internal
      *                    | |   |   | +- timer enable
      *                    | |   |   +- enable on timer 1 enable
      *                    | |   +- input clock prescaler
      *                    | +- count mode selection - continuous up mode
      *                    +- emulation control bits - timer continues */
```

```
reg(T1PR)  = 2 * period_2;
reg(T1CNT) = 0;
/* NOTE: must start timer the first time, and load the various registers
 * which are shadowed and only reload on timer overflow                  */
reg(T1CON) = 0xa806; /* 10 101 000 0 0 00 01 1 0                         */
 /* timer 1            | |   |   | | |  | | +- reserved
  *                    | |   |   | | |  | +- timer compare enable
  *                    | |   |   | | |  +- timer compare reload 0/PER
  *                    | |   |   | | +- clock source - internal
  *                    | |   |   | +- timer enable
  *                    | |   |   +- reserved
  *                    | |   +- input clock prescaler
  *                    | +- count mode selection - continuous up mode
  *                    +- emulation control bits - timer continues    */


/* Enable timer 1 period and underflow interrupt                        */
reg(EVIFRA)  = (T1PINT|T1UFINT|PDPINT);
reg(EVIMRA) |= (T1PINT|T1UFINT|PDPINT);
reg(EVIMRB)  = 0;

/* Enable INT2 interrupt                                                */
reg(IFR)     = ENABLE_INT2;
reg(IMR)    |= ENABLE_INT2;

/* Start timer 1                                                        */
reg(T1CON) = 0xa846; /* 10 101 000 0 1 00 01 1 0                        */
 /* timer 1            | |   |   | | |  | | +- reserved
  *                    | |   |   | | |  | +- timer compare enable
  *                    | |   |   | | |  +- timer compare reload 0/PER
  *                    | |   |   | | +- clock source - internal
  *                    | |   |   | +- timer enable
  *                    | |   |   +- reserved
  *                    | |   +- input clock prescaler
  *                    | +- count mode selection - continuous up mode
  *                    +- emulation control bits - timer continues    */


    /* Zero Crossing Interrupt etc Setup                                */
    reg(CAPCON) = 0xB855;  /* 1 01 1 1 0 0 0 01 01 01 01 */
     /* Capture control       | |  | | | | | |  |  +--+- positive edge detect
                              | |  | | | | | |  +--+- positive edge detect
                              | |  | | | | +- cap4 starts adc
                              | |  | | +-+- cap 34/12 use timer 2
                              | |  +-+- enable cap 3/4
                              | +- enable cap 12 / disable QEP
                              +- don't reset capture unit */

    reg(EVIFRC) = 0x000f;      /* Reset capture input flags             */

#if   (CAP_PORT == CAP_PORT3)
       reg(EVIMRC) = CAP3INT;      /* Enable capture 3 input interrupt  */
#elif (CAP_PORT == CAP_PORT4)
       reg(EVIMRC) = CAP4INT;      /* Enable capture 4 input interrupt  */
#endif
```



```
    reg(IFR)  = ENABLE_INT4;   /* Enable interrupt 4                    */
    reg(IMR) |= ENABLE_INT4;

    mag = 0;
    phase_trim = 0;

    for(i=0;i<(HARM_NUM*2*6);i++)    /* Zero all Harmonic Resonant Filters */
        Res_Array[i]=0;

    for(i=0;i<(2*6);i++)
        PRes_ist[i]=0;


    is_modulating = FALSE;

} /* End PWM_Init */


/*************************************************************************/
/*  NAME:        PWM_Start()                                             */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Sets up registers to start PWM.                         */
/*  NOTES:       Called from background code.                            */
/*  HISTORY:     08/02/2001  MN  Phase init. removed to fix bug.         */
/*               14/02/2001  MN  Added A/D Initialisation.               */
/*               13/12/2001  MN  Fourth phase leg code removed           */
/*               19/04/2002  MN  Removed mag=0 initialisation to fix a bug */
/*               29/04/2002  MN  Resonant Filter zeroing optimized       */
/*************************************************************************/
void PWM_Start(void)
{
    unsigned int int_mask;
    int i;

    int_mask    = reg(EVIMRA);
    reg(EVIMRA) &= ~(T1PINT|T1UFINT|PDPINT);  /* Disable PWM interrupt   */

    reg(COMCON) = 0x2b17;              /* Disable full compare outputs   */
    reg(T1CON)  = 0xa806;              /* Stop timer                     */
    reg(SACTR)  = 0x0000;
    reg(ACTR)   = 0x0000;

    reg(GPTCON) = 0x047a;              /* 000 00 01 00 1 11 10 10        */
    reg(ACTR)   = 0x0666;              /* 0 000 01 10 01 10 01 10        */
    reg(SACTR)  = 0x0000;

    reg(DBTCON) = 0x00e0|(DEADCNT<<8); /* xxxx xxxx 111 00 000           */

    reg(CMPR1)  = period_2;
    reg(CMPR2)  = period_2;
    reg(CMPR3)  = period_2;
    reg(SCMPR1) = 0;
    reg(SCMPR2) = 0;
    reg(SCMPR3) = 0;
```

```
/* Compare control register */
reg(COMCON) = 0x2b17;              /* 0 01 0 10 1 1 0 00 10 111      */
reg(COMCON) = 0xab17;              /* 1 01 0 10 1 1 0 00 10 111      */
reg(T2PR)   = (2*period_2)-1;      /* Timer 2 added for control of S&H*/

/* Stop Timer 1 and 2 */
reg(T1CON)  = 0xa806;              /* 10 101 000 0 0 00 01 1 0       */

reg(T2CNT) = 0;

reg(T2CMPR) = 3*period_2/2;   /* 25% of time at end of interrupt left  */
                              /*  for acquistion. Datasheet min = 4us  */

reg(T2CON) = 0x9082; /* 10 010 000 1 0 00 00 1 0 */

reg(T1PR)   = 2 * period_2;
reg(T1CNT)  = 0;

reg(T1CON)  = 0xa806;              /* 10 101 000 0 0 00 01 1 0       */
reg(T1CON)  = 0xa846;              /* 10 101 000 0 1 00 01 1 0       */

is_modulating = TRUE;

/* Set up registersfor ADC measurements */
reg(ADCTRL1) = ADCEMULATOR|ADC1EN|ADC2EN|ADC_VAC4|ADC_VAC5;
reg(ADCTRL2) = ADCEVSOC|ADCPSCALE; /*0x0403;*/

/* Enable timer 1 period, underflow and Power Drive Protection intpts  */
reg(EVIFRA) = (T1PINT|T1UFINT|PDPINT);
reg(EVIMRA) = int_mask | PDPINT; /* PDPINT Disabled in FAST_FAULT_STOP */

for(i=0;i<(HARM_NUM*2*6);i++)    /* Zero all Harmonic Resonant Filters */
    Res_Array[i]=0;

for(i=0;i<(2*6);i++)
    PRes_1st[i]=0;

} /* End PWM_Start */


/*************************************************************************/
/*  NAME:        PWM_Stop()                                              */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Sets up registers to stop PWM into a NULL state.        */
/*  NOTES:       Called from background code.                            */
/*************************************************************************/
void PWM_Stop(void)
{
    PWM_fast_stop();

} /* End PWM_Stop */


/*************************************************************************/
/*  NAME:        PWM_OFF()                                               */
/*  RETURNS:     void                                                    */
```



```
/*  DESCRIPTION: Sets up registers to turn all gates OFF.                */
/*  NOTES:       Called from background code.                            */
/*  HISTORY:     03/01/2000  MN  Created                                 */
/*************************************************************************/
void PWM_OFF(void)
{
    PWM_off_stop();

} /* End PWM_OFF */


/*************************************************************************/
/*  NAME:        fault_init()                                            */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Uses fault_mask to initialise the correct fault triggers. */
/*  NOTES:       Called from background code.                            */
/*  HISTORY:     09/02/2001  MN  Created from LPI Active Filter Code     */
/*************************************************************************/
void fault_init(unsigned int fault_mask)
{
    fault = NO_FAULT;

    if (fault_mask&LC_VDC1) /* Initialize the level comparison interrupts  */
    {
        reg(XINT1) = 0x8004;/* Clear interrupt flag, rising edge triggered */
    }

    if (fault_mask&(LC_VDC1))                  /* Enable INT1 interrupt   */
    {
        reg(IFR)  = ENABLE_INT1;
        reg(IMR) |= ENABLE_INT1;
        if (fault_mask&LC_VDC1)  reg(XINT1) |= 0x0001; /* Enable interrupt */
    }

} /* End fault_init */
        .

/*************************************************************************/
/*  NAME:        XINT1_int()                                             */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Over DC bus fault detection interrupt - XINT2           */
/*  NOTES:                                                               */
/*  HISTORY:     09/02/2001  MN  Created from LPI Active Filter Code     */
/*               13/09/2001  MN  Altered to suit CS-IIB                  */
/*************************************************************************/
interrupt void XINT1_int(void)
{
    PWM_fast_stop();
    fault |= LC_VDC1;

} /* End XINT1_int */


/*************************************************************************/
/*  NAME:        check_faults()                                          */
/*  RETURNS:     unsigned integer - value of fault, if any.              */
```

```
/*  DESCRIPTION: Clears the faults and checks that they have cleared.      */
/*  NOTES:       Called from background code.                              */
/*  HISTORY:     27/11/2001  MN  Altered to suit CS-IIB                    */
/*************************************************************************/
unsigned int check_faults(unsigned int fault_mask)
{
    unsigned int val = NO_FAULT;

    Clear_Gate_Fault();
    for (val=0; val<40; val++)
    {
        val++;
    }
    reg(EVIFRA) = 0x0001;                   /* Clear                     */
    val = (reg(EVIFRA)&0x0001)?GATE_FAULT:0;/* For PDPINT                */


    if (fault_mask&LC_VDC1)                 /* Test LC_VDC1 pin (XINT2)  */
    {
        val |= (reg(XINT2)&0x0040)?LC_VDC1:0;
    }

    /* Test level of ZX_in_sync for re-synchronization                   */
    if (fault_mask&LOST_SYNC)
    {
        val |= (ZX_in_sync<ZX_SYNC_LIMIT)?LOST_SYNC:0;
    }

    if (fault_mask&TRIAC_TRIP)
    {
        val != (TriacON())?TRIAC_TRIP:0;
    }

    /* Test DC voltage                                                   */
    if (fault_mask&LOW_DC_VOLTS)
    {
        val |= (Vdc<=V_DC_MIN)?LOW_DC_VOLTS:0;
    }

    if (fault_mask&HIGH_DC_VOLTS)
    {
        val |= (Vdc>=V_DC_MAX)?HIGH_DC_VOLTS:0;
    }

    fault |= val;
    return val&fault_mask;

} /* End check_faults */


/*************************************************************************/
/*  NAME:        SetMag(unsigned int)                                    */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Sets the modulation depth.                              */
/*  NOTES:       Called from background code.                            */
/*               Mag = 32767 == 200%                                     */
```



```
/*               m has 8 bits afters the point. ie 256d == 1% mod depth  */
/*************************************************************************/
void SetMag(unsigned int m)
{
    if (m < 0) m = 0;
    if (m > 32767) m = 32767;
    mag = (int)m;

} /* End SetMag */


/*************************************************************************/
/*  NAME:        GetFreq(signed int)                                     */
/*  RETURNS:     integer - fundamotal frequency achieved                 */
/*  DESCRIPTION: Gets the fundamental frequency.                         */
/*  NOTES:       Called from background code.                            */
/*               ff has 8 bits afters the point. ie 256d == 1Hz          */
/*************************************************************************/
int GetFreq(unsigned int sw_freq)
{

    return (int)( (((long)phase_step)*(long)sw_freq)/1281);

} /* End GetFreq */


/*************************************************************************/
/*  NAME:        SetSwFreq(signed int)                                   */
/*  RETURNS:     integer - switching frequency achieved                  */
/*  DESCRIPTION: Sets the switching frequency.                           */
/*  NOTES:       Called from background code.                            */
/*               fsw is in Hz                                            */
/*************************************************************************/
int SetSwFreq(int fsw)
{
    unsigned int half_period;

    half_period = (unsigned int)((CPU_FREQ/2/(long)fsw + 1)/2);
    if (half_period > MAX_PER_2) half_period = MAX_PER_2;
    else if (half_period < MIN_PER_2) half_period = MIN_PER_2;

    period_2 = half_period;

    return (int)((CPU_FREQ/2/(long)half_period + 1)/2);

} /* End SetSwFreq */


/*************************************************************************/
/*  NAME:        GetStatus()                                             */
/*  RETURNS:     integer - PWM Status                                    */
/*  DESCRIPTION: Returns the PWM status.                                 */
/*  NOTES:       Called from background code.                            */
/*               0 - stopped, 1 - running, -1 - fault                    */
/*************************************************************************/
int GetStatus(void)
```

```
{
    int ret_val = 0;

    if ( (fault) != NO_FAULT)
    {
        ret_val = -1;
    }
    else if (is_modulating == TRUE)
    {
        ret_val = 1;
    }
    return ret_val;

} /* End GetStatus */


/*************************************************************************/
/*  NAME:        GetFaults()                                             */
/*  RETURNS:     unsigned intoger - Detected Faults                      */
/*  DESCRIPTION: Returns the current detected faults.                    */
/*  NOTES:       Called from background code.                            */
/*  HISTORY:     08/12/2001  MN  Created                                 */
/*************************************************************************/
unsigned int GetFaults(void)
{
    return fault;

} /* End GetFaults */


/*************************************************************************/
/*  NAME:        ClearFaults()                                           */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Clears all previously detected faults.                  */
/*  NOTES:       Called from background code.                            */
/*  HISTORY:     08/12/2001  MN  Created                                 */
/*************************************************************************/
void ClearFaults(void)
{
    fault = NO_FAULT;

} /* End ClearFaults */
```


D.2 pwmback.c: C Background Code

```
/*************************************************************************/
/*                                                                       */
/* Application: Series VSI background routines with variable switching   */
/*              frequency                                                */
/* Developed By:Monash University: Power Electronics Group               */
/* Author:      M.Newman and A.McIver                                    */
/* File:        pwmback.c                                                */
/* History:                                                              */
/*      13/05/99 AM - Modified from lpiint.c                             */
/*      19/05/99 AM - Added version number                               */
/*      17/02/00 ZM - Added LPI v2.0 definition                          */
/*      23/11/00 MN - Constant V/Hz removed and comment headers added    */
/*      01/12/00 MN - Triac Control and Initialisation Mechanisms added  */
/*      12/12/00 MN - New states added to account for series protection  */
/*      19/01/01 MN - Debug Variable Grabbing Software Added             */
/*      14/02/01 MN - Trip Detection Software for Triac/MOV Added        */
/*      16/10/01 MN - Amalgamated C file with pwmmain.c                  */
/*      16/10/01 MN - Altered software to work with CS-IIB/MiniDSP board */
/*      19/02/02 MN - Feedforward injection scaling implemented          */
/*      19/04/02 MN - Bug in startup initialisation function rectified   */
/*      22/04/02 MN - Load voltage harmonic selection interface added    */
/*      24/04/02 MN - Legacy PI/Res series control stuff removed         */
/*      12/06/03 MN - Active Damping current feed-back added             */
/*      15/06/03 MN - Proportional voltage feed-back added               */
/*                                                                       */
/*************************************************************************/

#define VERSION     "2.0"

#include <mu_pcb.h>                 /* Defines Board Specific Variables    */
#include <c240.h>
#include <conio.h>
#include <stdlib.h>
#include <intrpt.h>
#ifndef _INLINE                     /* Work around for preprocessor bug    */
#define _INLINE 0                   /* in math.h                           */
#endif
#include <iib.h>
#include "series.h"                 /* Defines Series VSI Specific Values  */


/*************************************************************************/
/*                          VARIABLE DECLARATIONS                        */
/*************************************************************************/

unsigned int
    mag_bak,                        /* Actual mod_depth in 1/256ths of a % */
    mag_Vs,                         /* Source Voltage in mod. depth scale  */
    mod_depth,                      /* Target mod_depth in 1/256ths of a % */
    half_period,
    sw_freq = 0,
    temp,
    phase_step_bak,
    is_running = FALSE,
    is_isolated = TRUE,
```

```
    display_stop = 0,
    digin,                          /* Mirror of the digital inputs       */
    serial_mag = MAG_MIN,
    serial_fsw = FSW_MAX/4,
    tested_faults;

signed int
    freq = 0,
    serial_ff = FF_50HZ,
    scale3 = SCALE3;

/**** State stuff ****/
typedef void (* funcPtr)(void);
funcPtr state;

/**** Debug Variables ****/
signed int
    g[G_COUNT][G_SIZE];             /* Grab Data                          */


/************************************************************************/
/*                        FUNCTION DECLARATIONS                         */
/************************************************************************/

/**** Board Initialisation ****/
void init_series_pcb(void);

/**** State Functions ****/
void StateInit(void);
void StateStop(void);
void StateRun(void);
void StateRampUp(void);
void StateRampDown(void);
void StateFault(void);
void StateIsolated(void);
void StateOpenCont(void);
void StateCloseCont(void);
void StateOpenTriac(void);

/**** Voltage Regulator Functions ****/
double round(double);

/**** Serial port display of variables ****/
void init_display(void);
void display(void);
void display_state(funcPtr state);
void display_fault(unsigned int fault_type);
void print_help(void);
int  keyboard(void);                /* Handles serial port input          */
void buttons(void);                 /* Checks for and handles buttons     */

/**** Zero Corssing Functions ****/
void display_sync(unsigned int state);

/**** Misc ****/
void wait(unsigned int count);      /* Kills Time                         */
```



```
void set_refs(void);                /* Set freq. and magnitude references  */


/************************************************************************/
/*                        FUNCTION DEFINITIONS                          */
/************************************************************************/


/************************************************************************/
/*  NAME:         main()                                                */
/*  RETURNS:      integer - (always 0)                                  */
/*  DESCRIPTION:  Controls the flow of the background software          */
/*  NOTES:                                                              */
/*  HISTORY:      01/12/2000  MN  Triac Initialisation Added            */
/*                13/12/2000  MN  Interrupt Serive Routine 4 Added for ZX */
/*                14/12/2000  MN  Fault initialisation added            */
/*                16/10/2001  MN  Amalgamated with main() from old pwmmain.c*/
/*                16/10/2001  MN  Altered to work with CS-IIB/MiniDSP board */
/************************************************************************/
int main()
{
    char
        quit = FALSE,
        str[10];
    signed int
        i, count = 0;


/* Standard initialization for the board                                */
iib_init();
init_series_pcb();

/* Set up interrupt vectors */
set_GISR_vector(INT2Service,INT2);        /* PWM/PDPint Interrupt       */
set_GISR_vector(INT4Service,INT4);        /* Zero Crossing Interrupt    */
set_SISR_vector(XINT1_int,INT6,XINT1_ISR); /* Over DC Bus Volt. Fault   */

TriacReset();                   /* Reset the triac latch                */

puts("\n\n\t3 phase PWM for Series Injection\n\t\tVersion " VERSION "\n");

print_help();                   /* Also initialises display             */
set_ledout(LED_OFF,LOAD);       /* Turn all leds off                    */
state = &StateIsolated;         /* Initialize state machine             */
ZX_state = ZX_LOST;             /* Initialize ZX state                  */
tested_faults = GATE_FAULT|LC_VDC1;
fault_init(tested_faults);
PWM_Init();
EnableInts();

while (quit != 1)
{
    display();                  /* Update serial port display           */
    (*state)();                 /* Call state function                  */
    quit = keyboard();          /* Handle keyboard input                */
```

```
        buttons();
        set_refs();                 /* Set freq. and magnitude references  */
    }
    return 0;

} /* End main */


/*************************************************************************/
/*  NAME:        init_series_pcb()                                       */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Sets up the PCB board on startup                        */
/*  NOTES:       Replaces the standard VSI version in mu_pcb.h           */
/*  HISTORY:     01/12/2000  MN  Created from function in mu_pcb.h       */
/*               27/11/2001  MN  Altered to suit CS-IIB                  */
/*************************************************************************/
void init_series_pcb(void)
{

    set_waitstates(WAITSTATES_PS, WAITSTATES_DS, WAITSTATES_IS, WAITSTATES_EX);

    reg(OCRA) = 0x130f; /* 0001 0011 0000 1111                           */
                        /* \ /| |||| \  / \  /                           */
                        /*  | | ||||  \/   ++-- use IOPA0-3 as ADC in    */
                        /*  | | ||||   +------- not used                 */
                        /*  | | ||++------------ IOPB0&1 as PWM outputs   */
                        /*  | | ++-------------- IOPB2&3 as dig out      */
                        /*  | +----------------- IOPB4 as T2PWM (S&H trig.) */
                        /*  +------------------- IOPB5-7 as dig out       */

    reg(OCRB) = 0x00fd; /* 0000 0000 1111 1101                           */
                        /*  \    /   \|\| |||+- ADCSOC                   */
                        /*   \/      | | ||+-- not used                  */
                        /*    |      | | ++--- IOPC2&3 as dig out        */
                        /*    |      | +------ CAP1 & CAP2               */
                        /*    |      +-------- IOPC6&7 as CAP3 and CAP4   */
                        /*    +---------------- not used                 */

    reg(PBDATDIR) = 0xff00;
    reg(PCDATDIR) = 0xce00;     /* Bits are outputs except for bits 0,4,5  */

    reg(GPTCON)   = 0xe04c;     /* Set sample and hold to sample           */

                        /* Set up port for to suit SROM                    */
    reg(SPIPC1) = 0x11; /* 0001 0001                                       */
                        /* |||| |||+-- SPICLK data dir is output           */
                        /* |||| ||+--- SPICLK is an I/O pin                */
                        /* |||| |+---- pin is output 0                     */
                        /* |||| +----- current value of SPICLK            */
                        /* ++++------- as above for SPISTE                 */
    reg(SPIPC2) = 0x10; /* as SPIPC1 but for SPISIMO (hi) and SPISOMI (lo) */
    WriteControlLatch(0,LOAD) ;
    set_ledout(0,LOAD);

} /* End init_series_pcb */
```



```
/*************************************************************************/
/*  NAME:        set_refs()                                              */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Passes frequency and magnitude references to the interrupt*/
/*  NOTES:       Functions used are defined in pwmint.c and declared in  */
/*               pwm.h .                                                 */
/*  HISTORY:     23/11/2000  MN  Const V/Hz removed                      */
/*               16/02/2001  MN  Filter Parameter Set Function added     */
/*               20/02/2001  MN  PI Gain Variable Set Function added     */
/*************************************************************************/
void set_refs(void)
{

    sw_freq = SetSwFreq(serial_fsw);

    freq = GetFreq(sw_freq);

    mod_depth = serial_mag;

} /* End scale_refs */


/*************************************************************************/
/*  NAME:        display()                                               */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Displays variables and state through serial port        */
/*  NOTES:       Case 0 never happens                                    */
/*  HISTORY:     23/11/2000  MN  Const V/Hz removed                      */
/*************************************************************************/
void display(void)
{
    display_step++;
    scale3 = SCALE3;
    mag_Vs = ((long)Vs_sum*(long)scale3)>>16;
    switch(display_step)
    {
        case 1:  puts("  "); break;
        case 2:  if (mag_Vs>0)
                    putf(mag_Vs<<SHIFT5,SCALE2,0);
                 else
                    putd(mag_Vs);          break;/* Display Source Volt.  */
        case 3:  puts("V   ");             break;
        case 4:  putf(mod_depth<<SHIFT5,SCALE2,0);/* Display Demanded Volt.*/
                                           break;
        case 5:  puts("V   ");             break;
        case 6:  putf(freq,256,1);         break;
        case 7:  puts("Hz   ");            break;
        case 8:  putu(sw_freq);            break;
        case 9:  puts("Hz   ");            break;
        case 10: display_state(state);     break;
        case 11: puts("    ");             break;
        case 12: display_sync(ZX_state);   break;
        case 13: puts("  ");               break;
        case 14: putd(phase_trim);         break;
        case 15: puts("  ");               break;
```

```
        case 16: putu(mag_bak);             break;
        case 17: puts("    \r");            break;
        default:display_step = 0;
    }

} /* End display */


/*****************************************************************************/
/*  NAME:        StateIsolated()                                             */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: State used to keep the inverter isolated from the primary */
/*  NOTES:                                                                   */
/*  HISTORY:     12/12/2000  MN  Created                                     */
/*               14/12/2000  MN  Triac Trip Fault Test Removal Added         */
/*               10/12/2001  MN  Fault checking flow changed                 */
/*****************************************************************************/
void StateIsolated(void)
{
    tested_faults &= ~TRIAC_TRIP;
    check_faults(tested_faults);

    CloseContactor();
    /*TriacLatch();*/

    if (!is_isolated)
    {
        tested_faults |= LOST_SYNC;
        ClearFaults();
        check_faults(tested_faults);

        if (GetFaults()&tested_faults)
        {
            PWM_Stop();
            puts("\nCannot Un-Isolate as there is a fault condition.\n");
            display_fault(check_faults(tested_faults));
            is_isolated = TRUE;
        }
        else
        {
            state = &StateInit;
        }
    }

} /* End StateIsolated */


/*****************************************************************************/
/*  NAME:        StateInit()                                                 */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: State used to initialise PWM variables                      */
/*  NOTES:       On completion state is changed to "StateStop".              */
/*  HISTORY:     14/12/2000  MN  Triac Trip Fault Test Removal Added         */
/*****************************************************************************/
void StateInit(void)
{
```



```
    mod_depth = mag_Vn;
    mag_bak = mag_Vs;
    SetMag(mag_bak);
    PWM_Init();
    EnableInts();
    TriacLatch();
    tested_faults &= ~TRIAC_TRIP;
    state = &StateOpenCont;
    is_running = FALSE;

} /* End StateInit */


/*****************************************************************************/
/*  NAME:        StateOpenCont()                                             */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: State used to wait for contactor to open                    */
/*  NOTES:       Once contactor is open, state becomes StateOpenTriac        */
/*  HISTORY:     12/12/2000  MN  Created                                     */
/*               14/12/2000  MN  Triac Trip Fault Test Removal Added         */
/*****************************************************************************/
void StateOpenCont(void)
{
    check_faults(tested_faults);
    if (GetStatus() == -1)
    {
        PWM_Stop();
        state = &StateIsolated;
        CloseContactor();
        puts("\nFault During Contactor Opening.  Closing Contactor!\n");
        display_fault(check_faults(tested_faults));
    }
    else if (is_isolated)
    {
        CloseContactor();
        state = &StateCloseCont;
    }
    else
    {
        PWM_Stop();
        TriacLatch();
        tested_faults &= ~TRIAC_TRIP;
        OpenContactor();
        if ( ContOpened() )
        {
            state = &StateOpenTriac;
        }
    }

} /* End StateOpenCont */


/*****************************************************************************/
/*  NAME:        StateCloseCont()                                            */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: State used to wait for contactor to close                   */
```

```
/*  NOTES:      Once contactor is closed, state becomes StateIsolated     */
/*  HISTORY:    12/12/2000  MN  Created                                   */
/*              10/12/2001  MN  PWM_Stop added to ensure a null condition */
/*                              when coming from the fault state          */
/****************************************************************************/
void StateCloseCont(void)
{
    PWM_Stop();
    CloseContactor();
    if ( ContClosed() )
    {
        state = &StateIsolated;
    }

} /* End StateCloseCont */


/****************************************************************************/
/*  NAME:       StateOpenTriac()                                            */
/*  RETURNS:    void                                                        */
/*  DESCRIPTION: State used to wait for triac to open                       */
/*  NOTES:      Once triac is open, state becomes StateStop                 */
/*  HISTORY:    12/12/2000  MN  Created                                     */
/*              14/12/2000  MN  Triac Trip Fault Test Inclusion Added       */
/****************************************************************************/
void StateOpenTriac(void)
{
    check_faults(tested_faults);
    if (GetStatus() == -1)
    {
        state = &StateFault;
        PWM_Stop();
    }
    else if (!MovTripped())
    {
        PWM_Init();
        PWM_Stop();

        TriacReset();

        tested_faults |= TRIAC_TRIP;
        state = &StateStop;
    }
    else
    {
        puts("\nMOV Trip Sensed. Cannot Open Triac\n");
        state = &StateFault;
    }

} /* End StateOpenTriac */


/****************************************************************************/
/*  NAME:       StateStop()                                                 */
/*  RETURNS:    void                                                        */
/*  DESCRIPTION: Function called when inside a Stop State                   */
```



```
/*  NOTES:      State is held until either a fault is found or an         */
/*              external start command is recieved.                       */
/*  HISTORY:                                                              */
/****************************************************************************/
void StateStop(void)
{
    mod_depth = mag_Vs;
    mag_bak = mag_Vs;
    SetMag(mag_bak);

    check_faults(tested_faults);
    if (GetStatus() == -1)
    {
        state = &StateFault;
        PWM_OFF();
    }
    else if(is_isolated)
    {
        state = &StateCloseCont;
    }
    else if (SwitchOn())
    {
        if(Vdc_remove)
            tested_faults &= ~LOW_DC_VOLTS;
        else
            tested_faults |= LOW_DC_VOLTS;

/*        state = &StateRampUp;*/
        state = &StateRun;   /* Go directly to Run State */
        PWM_Start();
    }

} /* End StateStop */


/****************************************************************************/
/*  NAME:       StateRampUp()                                               */
/*  RETURNS:    void                                                        */
/*  DESCRIPTION: State used to ramp up the demanded magnitude               */
/*  NOTES:      State is held until either a fault is found or until        */
/*              demanded magitude is achieved.                              */
/*  HISTORY:                                                                */
/****************************************************************************/
void StateRampUp(void)
{
    state_led(LED_RUN);
    if (GetStatus() == -1)
    {
        state = &StateFault;
        PWM_OFF();
    }
    else if(is_isolated)
    {
        state = &StateStop;
    }
    else if (!SwitchOn())
```

```
    {
        state = &StateRampDown;
    }
    else if (mag_bak > mod_depth)
    {
        mag_bak = mod_depth;
        SetMag(mag_bak);
        state = &StateRun;          /* Ramping finished                    */
    }
    else
    {
        mag_bak += MAG_STEP;
        SetMag(mag_bak);
    }

} /* End StateRampUp */


/*****************************************************************************/
/*  NAME:        StateRun()                                                  */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: State is held whilst PWM is switching with no faults        */
/*  NOTES:       State is held until either a fault is found or until        */
/*               stop command is reached.                                    */
/*  HISTORY:                                                                 */
/*****************************************************************************/
void StateRun(void)
{
    state_led(LED_RUN);

    if (GetStatus() == -1)
    {
        state = &StateFault;
        PWM_OFF();
    }
    else if(is_isolated)
    {
        state = &StateStop;
    }
    else if (!SwitchOn())
    {
        /*state = &StateRampDown;*/
        PWM_Stop();                 /* Bypass StateRampDown                */
        state = &StateStop;         /* Ramping finished                    */
    }
    else
    {
        mag_bak = mod_depth;
        SetMag(mag_bak);
    }

} /* End StateRun */


/*****************************************************************************/
/*  NAME:        StateRampDown()                                             */
```



```
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: State used to ramp down the demanded magnitude              */
/*  NOTES:       State is used following an external stop command, but is    */
/*               overidden in the case of a fault.                           */
/*  HISTORY:                                                                 */
/*****************************************************************************/
void StateRampDown(void)
{
    state_led(LED_RUN);
    if (GetStatus() == -1)
    {
        state = &StateFault;
        PWM_OFF();
    }
    else if(is_isolated)
    {
        state = &StateStop;
        tested_faults &= ~LOW_DC_VOLTS;
    }
    else if (SwitchOn())
    {
        state = &StateRampUp;
    }
    else if ( mag_bak < MAG_STEP)
    {
        mag_bak = mag_Vs;
        SetMag(mag_bak);
        PWM_Stop();
        state = &StateStop;         /* Ramping finished                    */
    }
    else
    {
        mag_bak -= MAG_STEP;
        SetMag(mag_bak);
    }

} /* End StateRampDown */


/*****************************************************************************/
/*  NAME:        StateFault()                                                */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: State used whilst a fault condition is held and also when   */
/*               a fault has not been acknowledged.                          */
/*  NOTES:       Once cleared and acknowledged, the next state is given to   */
/*               the stop state.                                             */
/*  HISTORY:     14/12/2000  MN  Triac Trip Fault Test Removal Added         */
/*****************************************************************************/
void StateFault(void)
{
    state_led(LED_FAULT);
    tested_faults &= ~TRIAC_TRIP;
    do
    {
        while(kbhit()) getc();
        display_fault(check_faults(tested_faults));
```

```
        puts("Press any key to reset/Spacebar to isolate.\n");
        PWM_OFF();          /* Gates should already be all off by now anyway */
        while( (!kbhit()) && (!IsolatePressed()) )
        {
            digin = ReadDigIn();
        }
        if( (getc() == ' ') || IsolatePressed() )
        {
            CloseContactor();
            is_isolated = TRUE;
            break;
        }

    } while (check_faults(tested_faults) != NO_FAULT);

    print_help();

    if (is_isolated)
        state = &StateCloseCont;
    else
        state = &StateOpenTriac;

    tested_faults &= ~LOW_DC_VOLTS;

    ClearFaults();

    is_running = FALSE;

} /* End StateFault */


/*************************************************************************/
/*  NAME:         init_display()                                         */
/*  RETURNS:      void                                                   */
/*  DESCRIPTION: Sets up heading for variable display and resets rotating */
/*                variable display counter.                              */
/*  NOTES:                                                               */
/*  HISTORY:                                                             */
/*************************************************************************/
void init_display(void)
{
    puts("\n\n  mod      freq      sw_freq    state    ZX_State   Kp      Ki \n");
    display_step = 0;

} /* End init_display */


/*************************************************************************/
/*  NAME:         display_state()                                        */
/*  RETURNS:      void                                                   */
/*  DESCRIPTION: Function used to display the text for the current state. */
/*  NOTES:                                                               */
/*  HISTORY:                                                             */
/*************************************************************************/
void display_state(funcPtr state)
{
```



```
    if (state == &StateInit)            puts("Init      ");
    else if (state == &StateStop)       puts("Stop      ");
    else if (state == &StateRampUp)     puts("Ramp Up   ");
    else if (state == &StateRun)        puts("Run       ");
    else if (state == &StateRampDown)   puts("Ramp Down ");
    else if (state == &StateFault)      puts("Fault     ");
    else if (state == &StateOpenCont)   puts("OpenCont  ");
    else if (state == &StateCloseCont)  puts("CloseCont ");
    else if (state == &StateOpenTriac)  puts("OpenTriac ");
    else if (state == &StateIsolated)   puts("Isolated  ");
    else puts("    ???   ");

} /* End display_state */


/*************************************************************************/
/*  NAME:         display_fault()                                        */
/*  RETURNS:      void                                                   */
/*  DESCRIPTION: Function used to display the text for the ZX state.     */
/*  NOTES:                                                               */
/*  HISTORY:      08/12/2001  MN  Created                                */
/*************************************************************************/
void display_fault(unsigned int fault_type)
{
    puts("\nFault Value:  ");  putu(GetFaults());
    puts("\nMasked Fault: ");  putu(fault_type);  putc('\n');
    if (fault_type&GATE_FAULT   ) puts("PDPINT       \n");
    if (fault_type&LC_VDC1      ) puts("Over DC      \n");
    if (fault_type&LOST_SYNC    ) puts("Lost Sync    \n");
    if (fault_type&LOST_SUPPLY  ) puts("Lost Supply  \n");
    if (fault_type&CONTACT_FAULT) puts("Contact Fault\n");
    if (fault_type&TRIAC_TRIP   ) puts("Triac Trip   \n");
    if (fault_type&LOW_DC_VOLTS ) puts("Low DC Volts \n");
    if (fault_type&HIGH_DC_VOLTS) puts("High DC Volts\n");
    putc('\n');

} /* End display_fault */


/*************************************************************************/
/*  NAME:         display_sync()                                         */
/*  RETURNS:      void                                                   */
/*  DESCRIPTION: Function used to display the text for the ZX state.     */
/*  NOTES:                                                               */
/*  HISTORY:      13/12/2000  MN  Created                                */
/*************************************************************************/
void display_sync(unsigned int state)
{
    if (state == ZX_LOST)       puts("ZX Lost   ");
    else if (state == ZX_EST)   puts("ZX Est.   ");
    else if (state == ZX_SYNC)  puts("ZX in Sync");
    else if (state == ZX_PHASE) puts("ZX Phase  ");
    else if (state == ZX_FREQ)  puts("ZX Nudge f");
    else if (state == ZX_LOCK)  puts("ZX Lock   ");
    else if (state == ZX_MISC)  puts("ZX Realloc");
    else
```

```
    {
        putxx((unsigned int)status);
        puts("   ?      ");
    }

} /* End display_sync */


/*************************************************************************/
/*  NAME:        keyboard()                                              */
/*  RETURNS:     integer - 1 on a quit command and 0 otherwise           */
/*  DESCRIPTION: Takes user keboard commands  from the serial port and   */
/*               updates corresponding variables where appropriate.      */
/*  NOTES:                                                               */
/*  HISTORY:     23/11/2000  MN  Const V/Hz removed                      */
/*               24/11/2000  MN  Contactor Code Added                    */
/*               01/12/2000  MN  Triac Latch and Reset added             */
/*               12/02/2001  MN  Quit Unisolated Lock added for security */
/*               22/04/2002  MN  5th/7th Harmonic VL P+R toggle added    */
/*               23/04/2002  MN  9th/11th/13th Harmonic VL P+R toggle added*/
/*               24/04/2002  MN  Vdc volts removal mode toggle added     */
/*************************************************************************/
int keyboard(void)
{
    char c,i,j;
    int ret_val = 0;

    if (kbhit())
    {
        c = getc();
        switch(c)
        {
            case 'i': if (serial_mag < MAG_MAX-1 ) serial_mag+= 1; break;
            case 'I': if (serial_mag < MAG_MAX-25) serial_mag+=25; break;
            case 'm': if (serial_mag > MAG_MIN+1 ) serial_mag-= 1; break;
            case 'M': if (serial_mag > MAG_MIN+25) serial_mag-=25; break;
            case 'l': if (serial_ff <  FF_MAX-12  ) serial_ff += 12; break;
            case 'L': if (serial_ff <  FF_MAX-256 ) serial_ff +=256; break;
            case 'j': if (serial_ff > -FF_MAX+12  ) serial_ff -= 12; break;
            case 'J': if (serial_ff > -FF_MAX+256 ) serial_ff -=256; break;
            case '>': if (serial_fsw < FSW_MAX    ) serial_fsw+=0;  break;
            case '<': if (serial_fsw > FSW_MIN    ) serial_fsw-=0;  break;
            case 'R':
                if(Vdc_remove) {
                    Vdc_remove = FALSE;
                    puts("\n\n  - Closed loop control mode -\n\n");
                }
                else {
                    Vdc_remove = TRUE;
                    puts("\n\n  - Open loop Vdc removal mode\n\n");
                }
                break;
            case 'v': phase_trim-=25; break;
            case 'b': phase_trim+=25; break;
            case 'f':
                if(Feedforward) {
```



```
                    Feedforward = FALSE;
                    puts("\nFeedforward OFF\n");
                }
                else {
                    Feedforward = TRUE;
                    puts("\nFeedforward ON\n");
                }
                break;
            case 'F':
                if(Feedback) {
                    Feedback = FALSE;
                    puts("\nFeedback OFF\n");
                }
                else {
                    Feedback = TRUE;
                    puts("\nFeedback ON\n");
                }
                break;
            case '5':
                if(ON_5th) {
                    ON_5th = FALSE;
                    puts("\n5th Harmonic Feedback OFF\n");
                }
                else {
                    ON_5th = TRUE;
                    g_idx = 0;
                    g_con = GRAB_GO;
                    puts("\n5th Harmonic Feedback ON\n");
                }
                break;
            case '7':
                if(ON_7th) {
                    ON_7th = FALSE;
                    puts("\n7th Harmonic Feedback OFF\n");
                }
                else {
                    ON_7th = TRUE;
                    g_idx = 0;
                    g_con = GRAB_GO;
                    puts("\n7th Harmonic Feedback ON\n");
                }
                break;
            case '9':
                if(ON_9th) {
                    ON_9th = FALSE;
                    puts("\n9th Harmonic Feedback OFF\n");
                }
                else {
                    ON_9th = TRUE;
                    g_idx = 0;
                    g_con = GRAB_GO;
                    puts("\n9th Harmonic Feedback ON\n");
                }
                break;
            case '1':
                if(ON_11th) {
```

```
            ON_11th = FALSE;
            puts("\n11th Harmonic Feedback OFF\n");
        }
        else {
            ON_11th = TRUE;
            g_idx = 0;
            g_con = GRAB_GO;
            puts("\n11th Harmonic Feedback ON\n");
        }
        break;
    case '3':
        if(ON_13th) {
            ON_13th = FALSE;
            puts("\n13th Harmonic Feedback OFF\n");
        }
        else {
            ON_13th = TRUE;
            g_idx = 0;
            g_con = GRAB_GO;
            puts("\n13th Harmonic Feedback ON\n");
        }
        break;
    case 'a':
        if(ON_damp) {
            ON_damp = FALSE;
            puts("\nActive Damping OFF\n");
        }
        else {
            ON_damp = TRUE;
            g_idx = 0;
            g_con = GRAB_GO;
            puts("\nActive Damping ON\n");
        }
        break;
    case 'p':
        if(Proportional) {
            Proportional = FALSE;
            puts("\nFeed-back Proportional OFF\n");
        }
        else {
            Proportional = TRUE;
            g_idx = 0;
            g_con = GRAB_GO;
            puts("\nFeed-back Proportional ON\n");
        }
        break;
    case 'q':                       /* Quit                         */
        PWM_Stop();
        DisableInts();

        if (is_isolated) {
            ret_val = 1;
            puts("\n");
            CloseContactor();
        }
        else {
```



```
            is_isolated = TRUE;
            state = &StateCloseCont;
            puts("\nNot in isolated state to quit. Isolating Now.\n");
        }
        break;
    case 'h':                       /* Help                         */
        print_help();
        break;
    case 'd':                       /* Disable switching            */
        is_running = FALSE;
        break;
    case 'e':                       /* Enable switching             */
        is_running = TRUE;
        break;
    case 't':                       /* Latch Triac                  */
        TriacLatch();
        tested_faults &= ~TRIAC_TRIP;
        puts("\nTriac Latched\n");
        break;
    case 'r':                       /* Reset Triac                  */
        TriacReset();
        tested_faults |= TRIAC_TRIP;
        puts("\nTriac Reset\n");
        break;
    case 'u':
        is_isolated = FALSE;        /* Request to Unisolate System */
        puts("\nUn-Isolate Requested.\n");
        break;
    case ' ':
        is_isolated = TRUE;         /* Request to Isolate System   */
        puts("\nIsolate Requested.\n");
        break;
    case 'c':                       /* Clear grab code              */
        g_idx = 0;
        g_con = GRAB_READY;
        puts("\nGrab cleared\n");
        break;
    case 'g':                       /* Start grab now               */
        puts("\nGrab started\n");
        g_idx = 0;
        g_con = GRAB_GO;
        break;
    case 's':                       /* Show grabbed data to serial */
        puts("\n\ni\t\t1\t2\t3\t4\t5\t6\t7\t8\t9\t10\n");
        for (i=0; i<G_COUNT; i++)
        {
            putd(i);
            for (j=0; j<G_SIZE; j++)
            {
                putc('\t');
                putd(g[i][j]);
            }
            putc('\n');
        }
        break;
}
```

```
    }
    return ret_val;

} /* End keyboard */


/*************************************************************************/
/*  NAME:        buttons()                                               */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Checks for external start and stop buttons and updates  */
/*               corresponding variables where appropriate.              */
/*  NOTES:                                                               */
/*  HISTORY:     24/11/2000  MN  Initial Creation                        */
/*************************************************************************/
void buttons(void)
{

    digin = ReadDigIn();

    if( UnIsolatePressed() )
    {
        is_isolated = FALSE;                /* Request to Unisolate System */
        puts("\nUn-Isolate Requested.\n");
    }

    if( IsolatePressed() )
    {
        is_isolated = TRUE;                 /* Request to Isolate System   */
        CloseContactor();
        puts("\nIsolate Requested.\n");
    }

    if( StartPressed() && (!StopPressed()) && (!is_isolated) )
    {
        OpenContactor();
        is_running = TRUE;
    }

    if( StopPressed() )
    {
        is_running = FALSE;
    }

} /* End buttons */


/*************************************************************************/
/*  NAME:        print_help()                                            */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Outputs the help information to the serial port.        */
/*  NOTES:                                                               */
/*  HISTORY:     23/11/2000  MN  Const V/Hz removed                      */
/*************************************************************************/
void print_help(void)
{
    puts("\n\n\th\tHelp.\n");
```



```
    puts("\tq\tQuit.\n");
    puts("\te\tEnable switching.\n");
    puts("\td\tDisable switching.\n");
    puts("\ti/I\tVoltage Reference slow/fast increase.\n");
    puts("\tm/M\tVoltage Reference slow/fast decrease.\n");
    puts("\tl/L\tRef freq slow/fast increase.\n");
    puts("\tj/J\tRef freq slow/fast decrease.\n");
    puts("\t>\tSwitching freq increase (Disabled).\n");
    puts("\t<\tSwitching freq decrease (Disabled).\n");
    puts("\tc\tClear Grabbed Data.\n");
    puts("\tg\tGrab Data.\n");
    puts("\ts\tShow Grabbed Data to Serial Port.\n");
    puts("\tu\tUn-Isolate the system.\n");
    puts("\tv\tPhase Trim Subtract.\n");
    puts("\tb\tPhase Trim Addition.\n");
    puts("\tR\tToggle between Closed loop and Vdc removal modes.\n");
    puts("\tf\tToggle Feedforward Voltage Compensation ON/OFF\n");
    puts("\tF\tToggle Feedback Fundamental Voltage Compensation ON/OFF\n");
    puts("\tp\tToggle Proportional Feedback Voltage Compensation ON/OFF\n");
    puts("\t5\tToggle  5th Harmonic Voltage Compensation ON/OFF\n");
    puts("\t7\tToggle  7th Harmonic Voltage Compensation ON/OFF\n");
    puts("\t9\tToggle  9th Harmonic Voltage Compensation ON/OFF\n");
    puts("\t1\tToggle 11th Harmonic Voltage Compensation ON/OFF\n");
    puts("\t3\tToggle 13th Harmonic Voltage Compensation ON/OFF\n");
    puts("\tSpace Bar:\tIsolate the system.\n");

    init_display();

} /* End print_help */
```

D.3 filters.asm: Assembler Digital Filter Code

```
;*****************************************************************************
;  Application:     Assembly IIR filter functions for use with C code
;  Developed By:    Monash University, Power Electronics Group
;  Author:          M.Newman
;  File:            filters.asm
;  History:
;       07/08/2001 MN - Initial Creation
;       10/08/2001 MN - IIR_1stx2 added
;       10/08/2001 MN - IIR_2nd and IIR_2ndx2 added
;       22/08/2001 MN - DFII delta filter function added
;       14/11/2001 MN - 4 x cross coupled StatRF delta filter fcn added
;       08/01/2002 MN - File stripped down for use with P+Resonant Code
;       11/01/2002 MN - DFI Second order IIR shift filter added
;       22/04/2002 MN - 5th P+R Resonant Filter Implemented and tested
;       23/04/2002 MN - 7th/11th/13th P+R Filter Implemented and tested
;       24/04/2002 MN - 9th P+R Resonant Filter Implemented and tested
;       29/04/2002 MN - Single call 5th/7th/9th/11th/13th P+Rs Impl. & tested
;       04/02/2003 MN - Fundamental P+R filters added for feedback trimming
;       26/06/2003 MN - Additional coefficient options added
;
;*****************************************************************************
;
; 1. void Delta_2nd(signed int* arg1)      arg1 --> [ y   x   x2  x1  x0  * ]
;
; 2. void Delta_2nd_x2(signed int* arg1)   arg1 --> [ y   x   x2  x1  x0  *
;                                                     y   x   x2  x1  x0  * ]
;
;*****************************************************************************
;
; Typical calling process    r = IIR_Xxx(arg1,..,argx);
;   SACL    * ,AR1       <- arp always equals 1 entering into a function
;   BLKD    _arg2+0,**
;   BLKD    _arg1+0,**   <- arguments pushed on stack in reverse order
;   CALL    _IIR
;   SBRK    2
;   LDPK    _r
;   SACL    _r           <- result returned in low acc.
;
;*****************************************************************************
                                ;
    .text                       ; For macd one coefficient table must be
                                ; defined in program space (ie. Values A & B)
B_Del                           ;
    .word    1762, 8017, 1654   ; B_Del=[b0 b1 b2] P+Res D: 10kHz/1.5Hz/2^14/5/fund

A_Del                           ;
    .word    -16544, -1175      ; A_Del=[-a2 -a1]  P+Res D: 10kHz/1.5Hz/2^14/5/fund


                                ;
B_Del_5th                       ;
    .word    38, 614,  5        ; B_Del=[b0 b1 b2] P+Res D: 10kHz/1.5Hz/2^13/3/5th
A_Del_5th                       ;
    .word    -12897, -1735      ; A_Del=[-a2 -a1]  P+Res D: 10kHz/1.5Hz/2^13/3/5th
                                ;
B_Del_7th                       ;
```



```
    .word    38, 610,  5        ; B_Del=[b0 b1 b2] P+Res D: 10kHz/1.5Hz/2^13/3/7th
A_Del_7th                       ;
    .word    -25252, -3277      ; A_Del=[-a2 -a1]  P+Res D: 10kHz/1.5Hz/2^13/3/7th
                                ;
B_Del_9th                       ;
    .word    38, 303,  1        ; B_Del=[b0 b1 b2] P+Res D: 10kHz/1.5Hz/2^13/2/9th
A_Del_9th                       ;
    .word    -10398, -2660      ; A_Del=[-a2 -a1]  P+Res D: 10kHz/1.5Hz/2^13/2/9th
                                ;
B_Del_11th                      ;
    .word    37, 300,  1        ; B_Del=[b0 b1 b2] P+Res D: 10kHz/1.5Hz/2^13/2/11th
A_Del_11th                      ;
    .word    -15509, -3937      ; A_Del=[-a2 -a1]  P+Res D: 10kHz/1.5Hz/2^13/2/11th
                                ;
B_Del_13th                      ;
    .word    37, 296,  1        ; B_Del=[b0 b1 b2] P+Res D: 10kHz/1.5Hz/2^13/2/13th
A_Del_13th                      ;
    .word    -21529, -5441      ; A_Del=[-a2 -a1]  P+Res D: 10kHz/1.5Hz/2^13/2/13th
                                ;
                                ;
;h_FIR                          ; 7th Harmonic undamped P+R FIR coefficients
;   .word   4797,   4447,   3884,   3133,   2231,   1222,    154,   -921,  -1952,  -2889
;   .word  -3687,  -4307,  -4720,  -4906,  -4855,  -4570,  -4065,  -3365,  -2502,  -1519
;   .word   -463,    616,   1665,   2634,   3476,   4150,   4625,   4876,   4893,   4675
;   .word   4231,   3583,   2763,   1809,    769,   -309,  -1371,  -2368,  -3250,  -3976
;   .word  -4511,  -4828,  -4913,  -4761,  -4379,  -3787,  -3013,  -2093,  -1072,      0
;   .word   1072,   2093,   3013,   3787,   4379,   4761,   4913,   4828,   4511,   3976
;   .word   3250,   2368,   1371,    309,   -769,  -1809,  -2763,  -3583,  -4231,  -4675
;   .word  -4893,  -4876,  -4625,  -4150,  -3476,  -2634,  -1665,   -616,    463,   1519
;   .word   2502,   3365,   4065,   4570,   4855,   4906,   4720,   4307,   3687,   2889
;   .word   1952,    921,   -154,  -1222,  -2231,  -3133,  -3884,  -4447,  -4797,  -4915
;   .word  -4797,  -4447,  -3884,  -3133,  -2231,  -1222,   -154,    921,   1952,   2889
;   .word   3687,   4307,   4720,   4906,   4855,   4570,   4065,   3365,   2502,   1519
;   .word    463,   -616,  -1665,  -2634,  -3476,  -4150,  -4625,  -4876,  -4893,  -4675
;   .word  -4231,  -3583,  -2763,  -1809,   -769,    309,   1371,   2368,   3250,   3976
;   .word   4511,   4828,   4913,   4761,   4379,   3787,   3013,   2093,   1072,      0
;   .word  -1072,  -2093,  -3013,  -3787,  -4379,  -4761,  -4913,  -4828,  -4511,  -3976
;   .word  -3250,  -2368,  -1371,   -309,    769,   1809,   2763,   3583,   4231,   4675
;   .word   4893,   4876,   4625,   4150,   3476,   2634,   1665,    616,   -463,  -1519
;   .word  -2502,  -3365,  -4065,  -4570,  -4855,  -4906,  -4720,  -4307,  -3687,  -2889
;   .word  -1952,   -921,    154,   1222,   2231,   3133,   3884,   4447,   4797,   4915
                                ;
;   .globl  _FIR_200T           ; 200 Tap FIR filter
                                ;
                                ;
    .globl  _Delta_5th          ; Single second order Delta Filter
    .globl  _Delta_7th          ; Single second order Delta Filter
    .globl  _Delta_9th          ; Single second order Delta Filter
    .globl  _Delta_11th         ; Single second order Delta Filter
    .globl  _Delta_13th         ; Single second order Delta Filter
    .globl  _Delta_All          ; 10 second order Delta Filters
    .globl  _Delta_2nd_x2       ; 50Hz Resonant filters for feedback trimming
                                ;
                                ;
;*****************************************************************************
                                ;
```

```
_Delta_5th:                     ; Note: arp = 1 from calling process
                                ;
                                ;****       Setup Input Arguments        ****
    SAR   AR1, *-               ;*ar1 =  ar1, ar1-- :Put ar1 onto stack
    LAR   AR2, *+, AR2          ; ar2 = *ar1, ar1++ :Load arg1 value into ar2
    ADRK  #1                    ; Point arg2 to the second element
                                ;
                                ;****      Perform Multiplications       ****
    SPM   3                     ; Set PREG Shift mode to right shift by 6
    LACC  *+, 7                 ; Load 'x' into 32bit Accum. with shift X left
                                ; X=7 for 13bit, X=8 for 14bit, X=9 for 15bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_Del_5th+0, *+       ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling or less
    ROL                         ; Use this for 13bit scaling or less
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MAC   B_Del_5th+0, *-       ; Positive pma inc. & negative dma increment
    APAC                        ; Add final result to Accumulator
    MAR   *-                    ; Point ar2 to the dma address of y
    ROL                         ; Use this for 14bit scaling or less
    ROL                         ; Use this for 13bit scaling or less
    SACH  *, 7, AR1             ; Store result back into arg1[0]
    SPM   0                     ; Reset PREG Shift mode to no shift
    RET                         ; Return from function back to C code
                                ;
;*******************************************************************************
                                ;
_Delta_7th:                     ; Note: arp = 1 from calling process
                                ;
                                ;****       Setup Input Arguments        ****
    SAR   AR1, *-               ;*ar1 =  ar1, ar1-- :Put ar1 onto stack
    LAR   AR2, *+, AR2          ; ar2 = *ar1, ar1++ :Load arg1 value into ar2
    ADRK  #1                    ; Point arg2 to the second element
                                ;
                                ;****      Perform Multiplications       ****
    SPM   3                     ; Set PREG Shift mode to right shift by 6
    LACC  *+, 7                 ; Load 'x' into 32bit Accum. with shift X left
                                ; X=7 for 13bit, X=8 for 14bit, X=9 for 15bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_Del_7th+0, *+       ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling or less
    ROL                         ; Use this for 13bit scaling or less
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MAC   B_Del_7th+0, *-       ; Positive pma inc. & negative dma increment
    APAC                        ; Add final result to Accumulator
    MAR   *-                    ; Point ar2 to the dma address of y
```



```
    ROL                         ; Use this for 14bit scaling or less
    ROL                         ; Use this for 13bit scaling or less
    SACH  *, 7, AR1             ; Store result back into arg1[0]
    SPM   0                     ; Reset PREG Shift mode to no shift
    RET                         ; Return from function back to C code
                                ;
;*******************************************************************************
                                ;
_Delta_9th:                     ; Note: arp = 1 from calling process
                                ;
                                ;****       Setup Input Arguments        ****
    SAR   AR1, *-               ;*ar1 =  ar1, ar1-- :Put ar1 onto stack
    LAR   AR2, *+, AR2          ; ar2 = *ar1, ar1++ :Load arg1 value into ar2
    ADRK  #1                    ; Point arg2 to the second element
                                ;
                                ;****      Perform Multiplications       ****
    SPM   3                     ; Set PREG Shift mode to right shift by 6
    LACC  *+, 7                 ; Load 'x' into 32bit Accum. with shift X left
                                ; X=7 for 13bit, X=8 for 14bit, X=9 for 15bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_Del_9th+0, *+       ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling or less
    ROL                         ; Use this for 13bit scaling or less
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MAC   B_Del_9th+0, *-       ; Positive pma inc. & negative dma increment
    APAC                        ; Add final result to Accumulator
    MAR   *-                    ; Point ar2 to the dma address of y
    ROL                         ; Use this for 14bit scaling or less
    ROL                         ; Use this for 13bit scaling or less
    SACH  *, 7, AR1             ; Store result back into arg1[0]
    SPM   0                     ; Reset PREG Shift mode to no shift
    RET                         ; Return from function back to C code
                                ;
;*******************************************************************************
                                ;
_Delta_11th:                    ; Note: arp = 1 from calling process
                                ;
                                ;****       Setup Input Arguments        ****
    SAR   AR1, *-               ;*ar1 =  ar1, ar1-- :Put ar1 onto stack
    LAR   AR2, *+, AR2          ; ar2 = *ar1, ar1++ :Load arg1 value into ar2
    ADRK  #1                    ; Point arg2 to the second element
                                ;
                                ;****      Perform Multiplications       ****
    SPM   3                     ; Set PREG Shift mode to right shift by 6
    LACC  *+, 7                 ; Load 'x' into 32bit Accum. with shift X left
                                ; X=7 for 13bit, X=8 for 14bit, X=9 for 15bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_Del_11th+0, *+      ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling or less
```

```
    ROL                         ; Use this for 13bit scaling or less
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MAC   B_Del_11th+0, *-      ; Positive pma inc. & negative dma increment
    APAC                        ; Add final result to Accumulator
    MAR   *-                    ; Point ar2 to the dma address of y
    ROL                         ; Use this for 14bit scaling or less
    ROL                         ; Use this for 13bit scaling or less
    SACH  *, 7, AR1             ; Store result back into arg1[0]
    SPM   0                     ; Reset PREG Shift mode to no shift
    RET                         ; Return from function back to C code
                                ;
;*****************************************************************************
                                ;
_Delta_13th:                    ; Note: arp = 1 from calling process
                                ;
                                ;****       Setup Input Arguments        ****
    SAR   AR1, *-               ;*ar1 =  ar1, ar1-- :Put ar1 onto stack
    LAR   AR2, *+, AR2          ; ar2 = *ar1, ar1++ :Load arg1 value into ar2
    ADRK  #1                    ; Point arg2 to the second element
                                ;
                                ;****      Perform Multiplications       ****
    SPM   3                     ; Set PREG Shift mode to right shift by 6
    LACC  *+, 7                 ; Load 'x' into 32bit Accum. with shift X left
                                ; X=7 for 13bit, X=8 for 14bit, X=9 for 15bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_Del_13th+0, *+      ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling or less
    ROL                         ; Use this for 13bit scaling or less
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MAC   B_Del_13th+0, *-      ; Positive pma inc. & negative dma increment
    APAC                        ; Add final result to Accumulator
    MAR   *-                    ; Point ar2 to the dma address of y
    ROL                         ; Use this for 14bit scaling or less
    ROL                         ; Use this for 13bit scaling or less
    SACH  *, 7, AR1             ; Store result back into arg1[0]
    SPM   0                     ; Reset PREG Shift mode to no shift
    RET                         ; Return from function back to C code
                                ;
;*****************************************************************************
                                ;
_Delta_All:                     ; Note: arp = 1 from calling process
                                ;
                                ;****       Setup Input Arguments        ****
    SAR   AR1, *-               ;*ar1 =  ar1, ar1-- :Put ar1 onto stack
    LAR   AR2, *+, AR2          ; ar2 = *ar1, ar1++ :Load arg1 value into ar2
    ADRK  #1                    ; Point arg2 to the second element
                                ;
                                ;****      Perform Multiplications       ****
```



```
    SPM   3                     ; Set PREG Shift mode to right shift by 6
                                ;
                                ; Filter   --  1  -- A phase 5th
    LACC  *+, 7                 ; Load 'x' into 32bit Accum. with shift X left
                                ; X=7 for 13bit, X=8 for 14bit, X=9 for 15bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_Del_5th+0, *+       ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling or less
    ROL                         ; Use this for 13bit scaling or less
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MAC   B_Del_5th+0, *-       ; Positive pma inc. & negative dma increment
    APAC                        ; Add final result to Accumulator
    MAR   *-                    ; Point ar2 to the dma address of y
    ROL                         ; Use this for 14bit scaling or less
    ROL                         ; Use this for 13bit scaling or less
    SACH  *, 7                  ; Store result back into arg1[0]
    ADRK  #7                    ; Move arg2 to point to 2nd element in next filter
                                ;
                                ; Filter   --  2  -- B phase 5th
    LACC  *+, 7                 ; Load 'x' into 32bit Accum. with shift X left
                                ; X=7 for 13bit, X=8 for 14bit, X=9 for 15bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_Del_5th+0, *+       ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling or less
    ROL                         ; Use this for 13bit scaling or less
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MAC   B_Del_5th+0, *-       ; Positive pma inc. & negative dma increment
    APAC                        ; Add final result to Accumulator
    MAR   *-                    ; Point ar2 to the dma address of y
    ROL                         ; Use this for 14bit scaling or less
    ROL                         ; Use this for 13bit scaling or less
    SACH  *, 7                  ; Store result back into arg1[0]
    ADRK  #7                    ; Move arg2 to point to 2nd element in next filter
                                ;
                                ; Filter   --  3  -- A phase 7th
    LACC  *+, 7                 ; Load 'x' into 32bit Accum. with shift X left
                                ; X=7 for 13bit, X=8 for 14bit, X=9 for 15bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_Del_7th+0, *+       ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling or less
    ROL                         ; Use this for 13bit scaling or less
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
```

```
RPT   #3-1              ; Repeat the Multiply & Data Move X times
MAC   B_Del_7th+0, *-   ; Positive pma inc. & negative dma increment
APAC                    ; Add final result to Accumulator
MAR   *-                ; Point ar2 to the dma address of y
ROL                     ; Use this for 14bit scaling or less
ROL                     ; Use this for 13bit scaling or less
SACH  *, 7              ; Store result back into arg1[0]
ADRK  #7                ; Move arg2 to point to 2nd element in next filter
                        ;
                        ; Filter   --  4  -- B phase 7th
LACC  *+, 7             ; Load 'x' into 32bit Accum. with shift X left
                        ; X=7 for 13bit, X=8 for 14bit, X=9 for 15bit
MPY   #0                ; Multiply TREG by 0 to clear PREG
RPT   #2-1              ; Repeat the Multiply & Data Move X times
MAC   A_Del_7th+0, *+   ; Positive pma inc. & positive dma increment
APAC                    ; Add final result to Accumulator
ROL                     ; Use this for 14bit scaling or less
ROL                     ; Use this for 13bit scaling or less
SACH  *, 7              ; Store result x0 for use in delta shifts
LACL  #0                ; Set 32bit Accumulator to 0
MPY   #0                ; Multiply TREG by 0 to clear PREG
RPT   #3-1              ; Repeat the Multiply & Data Move X times
MAC   B_Del_7th+0, *-   ; Positive pma inc. & negative dma increment
APAC                    ; Add final result to Accumulator
MAR   *-                ; Point ar2 to the dma address of y
ROL                     ; Use this for 14bit scaling or less
ROL                     ; Use this for 13bit scaling or less
SACH  *, 7              ; Store result back into arg1[0]
ADRK  #7                ; Move arg2 to point to 2nd element in next filter
                        ;
                        ; Filter   --  5  -- A phase 9th
LACC  *+, 7             ; Load 'x' into 32bit Accum. with shift X left
                        ; X=7 for 13bit, X=8 for 14bit, X=9 for 15bit
MPY   #0                ; Multiply TREG by 0 to clear PREG
RPT   #2-1              ; Repeat the Multiply & Data Move X times
MAC   A_Del_9th+0, *+   ; Positive pma inc. & positive dma increment
APAC                    ; Add final result to Accumulator
ROL                     ; Use this for 14bit scaling or less
ROL                     ; Use this for 13bit scaling or less
SACH  *, 7              ; Store result x0 for use in delta shifts
LACL  #0                ; Set 32bit Accumulator to 0
MPY   #0                ; Multiply TREG by 0 to clear PREG
RPT   #3-1              ; Repeat the Multiply & Data Move X times
MAC   B_Del_9th+0, *-   ; Positive pma inc. & negative dma increment
APAC                    ; Add final result to Accumulator
MAR   *-                ; Point ar2 to the dma address of y
ROL                     ; Use this for 14bit scaling or less
ROL                     ; Use this for 13bit scaling or less
SACH  *, 7              ; Store result back into arg1[0]
ADRK  #7                ; Move arg2 to point to 2nd element in next filter
                        ;
                        ; Filter   --  6  -- B phase 9th
LACC  *+, 7             ; Load 'x' into 32bit Accum. with shift X left
                        ; X=7 for 13bit, X=8 for 14bit, X=9 for 15bit
MPY   #0                ; Multiply TREG by 0 to clear PREG
RPT   #2-1              ; Repeat the Multiply & Data Move X times
```



```
MAC   A_Del_9th+0, *+   ; Positive pma inc. & positive dma increment
APAC                    ; Add final result to Accumulator
ROL                     ; Use this for 14bit scaling or less
ROL                     ; Use this for 13bit scaling or less
SACH  *, 7              ; Store result x0 for use in delta shifts
LACL  #0                ; Set 32bit Accumulator to 0
MPY   #0                ; Multiply TREG by 0 to clear PREG
RPT   #3-1              ; Repeat the Multiply & Data Move X times
MAC   B_Del_9th+0, *-   ; Positive pma inc. & negative dma increment
APAC                    ; Add final result to Accumulator
MAR   *-                ; Point ar2 to the dma address of y
ROL                     ; Use this for 14bit scaling or less
ROL                     ; Use this for 13bit scaling or less
SACH  *, 7              ; Store result back into arg1[0]
ADRK  #7                ; Move arg2 to point to 2nd element in next filter
                        ;
                        ; Filter   --  7  -- A phase 11th
LACC  *+, 7             ; Load 'x' into 32bit Accum. with shift X left
                        ; X=7 for 13bit, X=8 for 14bit, X=9 for 15bit
MPY   #0                ; Multiply TREG by 0 to clear PREG
RPT   #2-1              ; Repeat the Multiply & Data Move X times
MAC   A_Del_11th+0, *+  ; Positive pma inc. & positive dma increment
APAC                    ; Add final result to Accumulator
ROL                     ; Use this for 14bit scaling or less
ROL                     ; Use this for 13bit scaling or less
SACH  *, 7              ; Store result x0 for use in delta shifts
LACL  #0                ; Set 32bit Accumulator to 0
MPY   #0                ; Multiply TREG by 0 to clear PREG
RPT   #3-1              ; Repeat the Multiply & Data Move X times
MAC   B_Del_11th+0, *-  ; Positive pma inc. & negative dma increment
APAC                    ; Add final result to Accumulator
MAR   *-                ; Point ar2 to the dma address of y
ROL                     ; Use this for 14bit scaling or less
ROL                     ; Use this for 13bit scaling or less
SACH  *, 7              ; Store result back into arg1[0]
ADRK  #7                ; Move arg2 to point to 2nd element in next filter
                        ;
                        ; Filter   --  8  -- B phase 11th
LACC  *+, 7             ; Load 'x' into 32bit Accum. with shift X left
                        ; X=7 for 13bit, X=8 for 14bit, X=9 for 15bit
MPY   #0                ; Multiply TREG by 0 to clear PREG
RPT   #2-1              ; Repeat the Multiply & Data Move X times
MAC   A_Del_11th+0, *+  ; Positive pma inc. & positive dma increment
APAC                    ; Add final result to Accumulator
ROL                     ; Use this for 14bit scaling or less
ROL                     ; Use this for 13bit scaling or less
SACH  *, 7              ; Store result x0 for use in delta shifts
LACL  #0                ; Set 32bit Accumulator to 0
MPY   #0                ; Multiply TREG by 0 to clear PREG
RPT   #3-1              ; Repeat the Multiply & Data Move X times
MAC   B_Del_11th+0, *-  ; Positive pma inc. & negative dma increment
APAC                    ; Add final result to Accumulator
MAR   *-                ; Point ar2 to the dma address of y
ROL                     ; Use this for 14bit scaling or less
ROL                     ; Use this for 13bit scaling or less
SACH  *, 7              ; Store result back into arg1[0]
```

```
ADRK  #7                    ; Move arg2 to point to 2nd element in next filter
                            ;
                            ; Filter   --  9  -- A phase 13th
LACC  *+, 7                 ; Load 'x' into 32bit Accum. with shift X left
                            ; X=7 for 13bit, X=8 for 14bit, X=9 for 15bit
MPY   #0                    ; Multiply TREG by 0 to clear PREG
RPT   #2-1                  ; Repeat the Multiply & Data Move X times
MAC   A_Del_13th+0, *+      ; Positive pma inc. & positive dma increment
APAC                        ; Add final result to Accumulator
ROL                         ; Use this for 14bit scaling or less
ROL                         ; Use this for 13bit scaling or less
SACH  *, 7                  ; Store result x0 for use in delta shifts
LACL  #0                    ; Set 32bit Accumulator to 0
MPY   #0                    ; Multiply TREG by 0 to clear PREG
RPT   #3-1                  ; Repeat the Multiply & Data Move X times
MAC   B_Del_13th+0, *-      ; Positive pma inc. & negative dma increment
APAC                        ; Add final result to Accumulator
MAR   *-                    ; Point ar2 to the dma address of y
ROL                         ; Use this for 14bit scaling or less
ROL                         ; Use this for 13bit scaling or less
SACH  *, 7                  ; Store result back into arg1[0]
ADRK  #7                    ; Move arg2 to point to 2nd element in next filter
                            ;
                            ; Filter   --  10  -- B phase 13th
LACC  *+, 7                 ; Load 'x' into 32bit Accum. with shift X left
                            ; X=7 for 13bit, X=8 for 14bit, X=9 for 15bit
MPY   #0                    ; Multiply TREG by 0 to clear PREG
RPT   #2-1                  ; Repeat the Multiply & Data Move X times
MAC   A_Del_13th+0, *+      ; Positive pma inc. & positive dma increment
APAC                        ; Add final result to Accumulator
ROL                         ; Use this for 14bit scaling or less
ROL                         ; Use this for 13bit scaling or less
SACH  *, 7                  ; Store result x0 for use in delta shifts
LACL  #0                    ; Set 32bit Accumulator to 0
MPY   #0                    ; Multiply TREG by 0 to clear PREG
RPT   #3-1                  ; Repeat the Multiply & Data Move X times
MAC   B_Del_13th+0, *-      ; Positive pma inc. & negative dma increment
APAC                        ; Add final result to Accumulator
MAR   *-                    ; Point ar2 to the dma address of y
ROL                         ; Use this for 14bit scaling or less
ROL                         ; Use this for 13bit scaling or less
SACH  *, 7, AR1             ; Store result back into arg1[0]
SPM   0                     ; Reset PREG Shift mode to no shift
RET                         ; Return from function back to C code
                            ;
;***************************************************************************
;                           ;
;_FIR_200T:                 ; Note: arp = 1 from calling process
;                           ;
;                           ;****      Setup Input Arguments       ****
;   SAR   AR1, *-           ;*ar1 =  ar1, ar1-- :Put ar1 onto stack
;   LAR   AR2, *+, AR2      ; ar2 = *ar1, ar1++ :Load arg1 value into ar2
;   ADRK  #201-1            ; Point arg2 to the end of the table
;                           ;
;                           ;****      Perform Multiplications      ****
;   SPM   3                 ; Set PREG Shift mode to right shift by 6
```



```
;   LACL  #0                ; Set 32bit Accumulator to 0
;   MPY   #0                ; Multiply TREG by 0 to clear PREG
;   RPT   #200-1            ; Repeat the Multiply & Data Move X times
;   MACD  h_FIR+0, *-       ; Positive pma inc. & negative dma increment
;   APAC                    ; Add result to Accumulator
;   SPM   0                 ; Reset PREG Shift mode to no shift
;   SACH  *, 5, AR1         ;
;   RET                     ; Return from function back to C code
;                           ;
;***************************************************************************
                            ;
_Delta_2nd_x2:              ; Note: arp = 1 from calling process
                            ;
                            ;****      Setup Input Arguments       ****
    SAR   AR1, *-           ;*ar1 =  ar1, ar1-- :Put ar1 onto stack
    LAR   AR2, *+, AR2      ; ar2 = *ar1, ar1++ :Load arg1 value into ar2
    ADRK  #1                ; Point arg2 to the second element
                            ;
                            ;****      Perform Multiplications      ****
    SPM   3                 ; Set PREG Shift mode to right shift by 6
                            ; Filter   --  1  --
    LACC  *+, 8             ; Load 'x' into 32bit Accum. with shift 8 left - 14bit
;   LACC  *+, 9             ; Load 'x' into 32bit Accum. with shift 9 left - 15bit
    MPY   #0                ; Multiply TREG by 0 to clear PREG
    RPT   #2-1              ; Repeat the Multiply & Data Move X times
    MAC   A_Del+0, *+       ; Positive pma inc. & positive dma increment
    APAC                    ; Add final result to Accumulator
    ROL                     ; Use this for 14bit scaling
    SACH  *, 7              ; Store result x0 for use in delta shifts
    LACL  #0                ; Set 32bit Accumulator to 0
    MPY   #0                ; Multiply TREG by 0 to clear PREG
    RPT   #3-1              ; Repeat the Multiply & Data Move X times
    MAC   B_Del+0, *-       ; Positive pma inc. & negative dma increment
    APAC                    ; Add final result to Accumulator
    MAR   *-                ; Point ar2 to the dma address of y
    ROL                     ; Use this for 14bit scaling
    SACH  *, 7              ; Store result back into arg1[0]
    ADRK  #7                ; Move arg2 to point to 2nd element in next filter
                            ; Filter   --  2  --
    LACC  *+, 8             ; Load 'x' into 32bit Accum. with shift 8 left - 14bit
;   LACC  *+, 9             ; Load 'x' into 32bit Accum. with shift 9 left - 15bit
    MPY   #0                ; Multiply TREG by 0 to clear PREG
    RPT   #2-1              ; Repeat the Multiply & Data Move X times
    MAC   A_Del+0, *+       ; Positive pma inc. & positive dma increment
    APAC                    ; Add final result to Accumulator
    ROL                     ; Use this for 14bit scaling
    SACH  *, 7              ; Store result x0 for use in delta shifts
    LACL  #0                ; Set 32bit Accumulator to 0
    MPY   #0                ; Multiply TREG by 0 to clear PREG
    RPT   #3-1              ; Repeat the Multiply & Data Move X times
    MAC   B_Del+0, *-       ; Positive pma inc. & negative dma increment
    APAC                    ; Add final result to Accumulator
    MAR   *-                ; Point ar2 to the dma address of y
    ROL                     ; Use this for 14bit scaling
    SACH  *, 7, AR1         ; Store result back into arg1[0]
    SPM   0                 ; Reset PREG Shift mode to no shift
```

```
    RET                         ; Return from function back to C code
                                ;
;***************************************************************************
    .end
```

Appendix E

Shunt DSP Software

E.1 pcrint.c: C Interrupt Code

```
/*************************************************************************************/
/*                                                                                   */
/* Application: PCR interrupt routine                                                */
/* Developed By:Monash University                                                    */
/* Author:      M.Newman and A.McIver                                                */
/* File:        pcrint.c                                                             */
/* History:     derived from VSI code   and EC PCR code (...\ec\c\pwm.c)             */
/*                                                                                   */
/*    30/06/1998  AM  Added zero crosing code                                        */
/*    20/07/1999  AM  Moving to new code library and HPI hardware                    */
/*    05/09/2001  MN  Altered to work with the MiniDSP and CS-IIB                    */
/*    24/09/2001  MN  Delta Filter software added and linked to assembly             */
/*    01/10/2001  MN  Analogs re-arranged to allow load current measurement         */
/*    03/10/2001  MN  Notch based Active filter included and tested                  */
/*    14/11/2001  MN  StatRF based AF added and tested                               */
/*    15/11/2001  MN  SRF based AF added                                             */
/*                                                                                   */
/*************************************************************************************/

#include <c240.h>
#define MiniDSPpcb      1
#include <mu_pcb.h>
#include <fastdiv.h>
#include <minibus.h>                    /* Defines DAC interface information */
#include <adc.h>
#include <iib.h>
#include "pcr.h"
#include "sintab.h"
#include "shunt.h"
#include "filters.h"


/*************************************************************************************/
/*                          VARIABLE DECLARATIONS                                    */
/*************************************************************************************/
```

```
/* Shared variables */
unsigned int
    fault;                  /* Bits set indicate faults                    */

/* Shared for debug only */
unsigned int
    s1,s2,s3,s4,s5,s6,      /* Magic shift numbers                         */
    v_scale_int,            /* Calculated scale factor for voltage to time */
    Ctrl_Type=0;            /* Determines which controller type is used    */
signed int
    i_max_err,              /* Maximum current error for accurate scaling  */
    v_max_err,              /* Maximum intermediate voltage value          */
    DACTest,            /* Toggle Variable for calabrating DAC Outpus      */
    i_scale_int;   /* Calculated scale factor for current to voltage (L/dt)*/
signed int
    g_con = GRAB_READY,     /* Grab control variable                       */
    g_a = 0,                /* Grab variables                              */
    g_b = 0,
    g_c = 0,
    g_d = 0,
    g_e = 0,
    g_idx=0;

/* pcrint.c only variables */
unsigned int
    is_modulating,  /* Flag set if the PWM outputs are active              */
    in_sync,        /* Flag to indicate that snc is achieved               */
    is_PCR,         /* Run as PCR, not VSI                                 */
    is_sync,        /* Sync to ZX                                          */
    is_mag,         /* Magnitude control                                   */
    is_Vdc,         /* Indicator of whether DC bus control is used         */
    phi,            /* Phase angle offset for power factor control         */
    phase,          /* Current phase angle 0 .65535 == 0..360deg           */
    phase_step,     /* Change in phase angle in half a switching cycle     */
    dead_band,      /* Level of compensation for deadband in clock cycles  */
    adc_index_d=0,  /* For cyclic reading of misc analog values            */
    Vdc_count,      /* Count variable to slow down Vdc PI Loop             */
    Vdc_sc,         /* Scaled DC link voltage                              */
    Vscale,         /* Voltage scale factor                                */
    int1_vect,      /* Specific interrupt vector for interrupt 1           */
    int2_vect,      /* Specific interrupt vector for interrupt 2           */
    int4_vect,      /* Specific interrupt vector for interrupt 2           */
    index_d,        /* Mapping of phase to sine tables for direct          */
    index_q,        /* Mapping of phase to sine tables for quadrature      */
    ZX_in_sync,     /* > ZX_SYNC_LIMIT means that sync has been achieved   */
    ZX_state,       /* State of the zero crossing synchronization process  */
    ZX_count,       /* The number of switching cycles between ZX interrupts*/
    ZX_seen,        /* Set to TRUE when a zero crossing is detected        */
    ZX_cycles,      /* Count of number of ZXs during averaging             */
    ZX_sum;         /* Running sum for average                             */

signed int
    mag_d,          /* Scaled direct magnitude reference                   */
    mag_q,          /* Quadrature magnitude reference                      */
    mag_af,         /* Magnitude for active filter compensation            */
    Ki_dc=5,Kp_dc=5,/* Proportional and Integral DC Bus control gains      */
```



```
    spare1,         /* Dummy variable                                      */
    adc_Vdc,        /* DC link voltage [adc units]                         */
    Vdc_meas,       /* DC link voltage after input hysteresis [adc units]  */
    Vdc_ref,        /* DC link voltage reference [adc units]               */
    Vdc_err,        /* DC Voltage Error Reference                          */
    Vdc_sum,        /* DC Voltage Integrator Sum                           */
    sum_round_err,  /* Holds rounding errors of the Vdc control integrator */
    IrefA, IrefB,   /* Demanded currents [adc units]                       */
    IrefA2, IrefB2, /* Previous Demanded currents [adc units]              */
    ImeasA =0, ImeasB =0,/* Measured currents [adc units]                  */
    ImeasA2=0, ImeasB2=0,/* Previous measured currents  [adc units]        */
    ImeasA3=0, ImeasB3=0,
    ILmeasA, ILmeasB,    /* Measured load currents [adc units]             */
    VrefA_0, VrefB_0,    /* Demanded voltages [Emf units]                  */
    VactA_1=0, VactB_1=0,/* Actual voltages [time units]                   */
    VactA_2=0, VactB_2=0,/* Previous actual voltages [adc units]           */
    VactA_3=0, VactB_3=0,
    V_Anx, V_Bnx, V_Cnx,/* Actual switching times sans deadband comp.      */
    V_A, V_B, V_C,  /* Phase voltage switching times [time units]          */
    EmfBA, EmfCA,   /* Measured line to line Emfs [adc units]             */
    EmfA_1, EmfB_1, /* Line Emfs [Emf units]                               */
    EmfA_2, EmfB_2, /* Previous line Emfs [Emf units]                      */
    Voff,           /* Offset for third harmonic injection [time units]    */
    IcalcC,         /* Calculated 3rd I for deadband comp. [adc units]     */
    Vthird,         /* = 65536/3 to trick compiler                         */
    ZX_time,        /* Time of captured ZX in timer units                  */
    ZX_time_phase,  /* Time of captured ZX in phase units                  */
    ZX_phase_scale, /* Scale factor between timer and phase units          */
    ZX_phase_err,   /* Difference in phase units (2^16 == 360deg)          */
    ZX_err_sum,     /* Integral for frequency control                      */
    A_sat, B_sat, C_sat;    /* Counters to detect loss of supply           */


/* Assembler IIR Filter Variables */
signed int
    notchA[]   = {0, 0, 0, 0, 0, 0},    /* [ y   x   x2  x1  x0  * ]      */
    notchB[]   = {0, 0, 0, 0, 0, 0},    /* [ y   x   x2  x1  x0  * ]      */
    StatRF[]   = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    *StatRF_11 =  StatRF,               /* [ y   x   x2  x1  x0  * ]      */
    *StatRF_12 = (StatRF+6 ),           /* [ y   x   x2  x1  x0  * ]      */
    *StatRF_21 = (StatRF+12),           /* [ y   x   x2  x1  x0  * ]      */
    *StatRF_22 = (StatRF+18);           /* [ y   x   x2  x1  x0  * ]      */


/* SRF Variables */
signed int
    iaf, ibf,           /* phase load currents after filtering            */
    kc,                 /* current cos(wt) variable                       */
    ks,                 /* current sin(wt) variable                       */
    tmp1, tmp2,         /* temporary variables                            */
    xy_2[] = {0, 0, 0, 0, 0, 0, 0, 0};  /* [ yd1 xd0 xd1 * yq1 xq0 xq1 * ] */


signed int              /* abc->dq->abc Transform Constants */
    k1 = 26755,         /*  root(2/3) x 32768 */
```

```
    k2 = 13376,         /*  1/root(6) x 32768 */
    k3 = 23171;         /*  1/root(2) x 32768 */


/* Sine tables in tables.c */
extern signed int sin_table_A[TABLE_SIZE];
extern signed int sin_table_B[TABLE_SIZE];



/*************************************************************************/
/*                          FUNCTION DEFINITIONS                         */
/*************************************************************************/


/*************************************************************************/
/*  NAME:        INT2Service()                                           */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Interrupt for EV Timer and PDPINT                       */
/*  NOTES:       coltrollers, pwm calculations and the loading compare   */
/*               registers is perfomed in this interrupt.                */
/*  HISTORY:     13/09/2001  MN  Altered to suit CS-IIB                  */
/*               24/09/2001  MN  Delta Filter software added             */
/*               01/10/2001  MN  Analogs re-arranged for IL measurement  */
/*               03/10/2001  MN  Active filter now working               */
/*               04/10/2001  MN  DC Bus Control upgraded again           */
/*               03/10/2001  MN  Notch based AF included and tested      */
/*               14/11/2001  MN  StatRF based AF added and tested        */
/*               15/11/2001  MN  SRF based AF added                      */
/*               03/01/2003  GB  PCR with estimated backemf added        */
/*************************************************************************/
interrupt void INT2Service(void)
{
    Timing_R2_on();

    int2_vect = reg(EVIVRA);    /* Reading this register seems to clear it */

    if (reg(EVIFRA)&PDPINT)
    {
        PWM_fast_stop();
        fault |= PDP_FAULT;
    }

    /* Second ADC conversion starts when first finishes */
    reg(ADCTRL1) = ADCEMULATOR|ADC1EN|ADC2EN|ADC_I1|ADC_I2|ADCSTART;

    /* Zero crossing processing */
    ZX_count++;
    if (ZX_count > ZX_MAX_COUNT)    /* Zero crossing signal lost          */
    {
        if (is_sync == TRUE)        /* Only use ZX if need to             */
        {
            PWM_fast_stop();        /* Halt modulation                    */
            fault |= LOST_SYNC;
            ZX_state = ZX_LOST;     /* Restart searching for sync         */
            ZX_in_sync = 0;
```



```
        }
        ZX_count = 0;
    }
    if (ZX_state == ZX_LOST)        /* No idea of anything: start freq est.*/
    {
        if (is_sync == TRUE) fault |= LOST_SYNC;
        if (ZX_seen == TRUE)
        {
            ZX_cycles = 0;
            ZX_sum = 0;
            ZX_count = 0;
            ZX_seen = FALSE;
            if (is_sync == TRUE) ZX_state = ZX_EST;
        }
    }
    else if (ZX_state == ZX_EST)    /* Roughly measure period and average  */
    {
        if (ZX_seen == TRUE)
        {
            ZX_cycles++;
            ZX_sum += ZX_count;
            ZX_count = 0;               /* Reset counter                  */
            ZX_seen = FALSE;
        }
        if (ZX_cycles >= ZX_CYCLE_AVG)
        {
            ZX_sum = ZX_sum/ZX_CYCLE_AVG;
            phase_step = 0xffff/ZX_sum; /* Approximate frequency          */
            ZX_sum -= ZX_sum/8;         /* Also use for glitch filter     */
            phase = phase_step + ZX_OFFSET; /* Within phase_step          */
            ZX_state = ZX_MISC;         /* Calculate ZX_phase_scale first */
        }
    }
    else if (ZX_state == ZX_SYNC)       /* Accurately measure phase error */
    {
        if (ZX_seen == TRUE)
        {
            ZX_seen = FALSE;
            if (ZX_count > ZX_sum)      /* Ignore glitches                */
            {
                ZX_count = 0;
                /* Rescale to phase units */
                ZX_time_phase = ZX_OFFSET + (((ZX_time>>2)*ZX_phase_scale)>>3);
                /* Calculate phase error captured time */
                ZX_phase_err = phase - ZX_time_phase;
                /* Limit size of phase change */
                if (ZX_phase_err > ZX_BIG_ERR)
                {
                    phase -= ZX_BIG_ERR;
                    ZX_err_sum = (ZX_err_sum+ZX_BIG_ERR)>>1;/*Int. ph. errs*/
                }
                else if (ZX_phase_err < -ZX_BIG_ERR)
                {
                    phase += ZX_BIG_ERR;
                    ZX_err_sum = (ZX_err_sum-ZX_BIG_ERR)>>1;
                }
```

```
            else
            {
                phase -= ZX_phase_err;
                ZX_err_sum = (ZX_err_sum+ZX_phase_err)>>1;
            }
            ZX_state = ZX_FREQ;
        }
    }
}
else if (ZX_state == ZX_FREQ)   /* Nudge frequency if needed          */
{
    /* If too large, nudge freq (phase_step) */
    if (ZX_err_sum > ZX_FREQ_ERR)
    {
        phase_step--;
    }
    else if (ZX_err_sum < -ZX_FREQ_ERR)
    {
        phase_step++;
    }
    ZX_state = ZX_LOCK;
}
else if (ZX_state == ZX_LOCK)   /* Test to see if still in sync       */
{
    if (ZX_in_sync >= ZX_SYNC_LIMIT)
    {
        if ((ZX_phase_err>ZX_PHASE_ERR)||(ZX_phase_err<-ZX_PHASE_ERR))
        {                           /* Gone out of sync */
            PWM_fast_stop();
            ZX_in_sync = 0;
            in_sync = FALSE;
            fault |= LOST_SYNC;
        }
        else
        {
            in_sync = TRUE;
        }
    }
    else if ((ZX_phase_err<ZX_PHASE_ERR)&&(ZX_phase_err>-ZX_PHASE_ERR))
    {                               /* In sync this cycle                 */
        ZX_in_sync++;
    }
    else
    {
        ZX_in_sync = 0;
    }
    ZX_state = ZX_MISC;
}
else if (ZX_state == ZX_MISC)
{

    ZX_phase_scale = (phase_step<<5)/PERIOD;  /* THIS IS SLOW!! (~5us) */
    ZX_state = ZX_SYNC;
}

/* Update phase angle */
```



```
phase  +=  phase_step;
index_d = (phase+phi)>>6;
index_q = (phase+phi+16384)>>6;

/* Reference Step Creation at top of A phase waveform */
if(step_flag==1)
{
    if( (index_d>250) && (index_d<260) )
    {
        step_flag = 0;
        ref_double_bak = ref_double;
    }
}

/* Shuffle old measurements */
EmfA_2 = EmfA_1;
EmfB_2 = EmfB_1;

IrefA2 = IrefA;
IrefB2 = IrefB;

notchA[2] = (notchA[3]>>5) + notchA[2]; /* x2 = Delta(x1)              */
notchA[3] = (notchA[4]>>5) + notchA[3]; /* x1 = Delta(x0)              */

notchB[2] = (notchB[3]>>5) + notchB[2]; /* x2 = Delta(x1)              */
notchB[3] = (notchB[4]>>5) + notchB[3]; /* x1 = Delta(x0)              */

StatRF_11[2] = (StatRF_11[3]>>5) + StatRF_11[2]; /* x2 = Delta(x1)     */
StatRF_11[3] = (StatRF_11[4]>>5) + StatRF_11[3]; /* x1 = Delta(x0)     */

StatRF_22[2] = (StatRF_22[3]>>5) + StatRF_22[2]; /* x2 = Delta(x1)     */
StatRF_22[3] = (StatRF_22[4]>>5) + StatRF_22[3]; /* x1 = Delta(x0)     */

StatRF_12[2] = (StatRF_12[3]>>5) + StatRF_12[2]; /* x2 = Delta(x1)     */
StatRF_12[3] = (StatRF_12[4]>>5) + StatRF_12[3]; /* x1 = Delta(x0)     */

StatRF_21[2] = (StatRF_21[3]>>5) + StatRF_21[2]; /* x2 = Delta(x1)     */
StatRF_21[3] = (StatRF_21[4]>>5) + StatRF_21[3]; /* x1 = Delta(x0)     */


/* Vdc control */
Vdc_count++;

if ( is_Vdc  && (Vdc_count>=10) )
{
    Vdc_count = 0;
    Vdc_err = Vdc_ref - adc_Vdc;        /* Error without hysteresis    */
    Vdc_sum += (Vdc_err>>Ki_dc);        /* Integral Gain Sum           */

    sum_round_err += Vdc_err - ((Vdc_err>>Ki_dc)<<Ki_dc);
    if ( sum_round_err > (1<<Ki_dc))    /* Add back in rounding errors */
    {                                   /* caused by fixed point ops.  */
        sum_round_err = 0;
        Vdc_sum++;
    }
    else if ( sum_round_err < -(1<<Ki_dc))
```

```
    {
        sum_round_err = 0;
        Vdc_sum--;
    }

    if (Vdc_sum >  30000)              /* Clamp integrator sum        */
    {
        Vdc_sum =  30000;
    }
    else if (Vdc_sum < -30000)
    {
        Vdc_sum = -30000;
    }

    spare1 = adc_Vdc - Vdc_meas;       /* Add an input hysteresis to  */
    if ( (spare1 > 1) || (spare1 < -1) )/* KP to stop constant hunting */
    {
        Vdc_meas = adc_Vdc;
    }
    Vdc_err = Vdc_ref - Vdc_meas;      /* Error with small hysteresis */

    mag_d = (Vdc_err<<Kp_dc) + Vdc_sum; /* Add Proportional Gain: KP   */
}

/* Clamp magnitude */
if (mag_d > (signed int)(MAG_MAX) )
{
    mag_d = MAG_MAX;
}
else if (mag_d < (signed int)(-MAG_MAX) )
{
    mag_d = -MAG_MAX;
}

/* Calculate A phase current reference */
IrefA =  ((long)sin_table_A[index_d]*(long)mag_d)>>16;
IrefA += ((long)sin_table_A[index_q]*(long)mag_q)>>16;
IrefA =  (IrefA + 2)>>2;  /* Shift down to give a max. swing of +-4096 */

/* Calculate B phase current reference */
IrefB =  ((long)sin_table_B[index_d]*(long)mag_d)>>16;
IrefB += ((long)sin_table_B[index_q]*(long)mag_q)>>16;
IrefB =  (IrefB + 2)>>2;  /* Shift down to give a max. swing of +-4096 */


/*DAC_Test();*/
load_DAC(DAC0,((IrefA>>1  )+DAC_OFFSET));
load_DAC(DAC1,((IrefB>>1  )+DAC_OFFSET));
load_DAC(DAC2,((IrefA     )+DAC_OFFSET));
load_DAC(DAC3,((IrefB     )+DAC_OFFSET));
update_DACS();

Timing_R2_off();
Timing_R2_on();

/* Read first ADC result : should be ready by now */
```



```
ILmeasA = (ADC_ZERO - (signed)(get_ADC1()))<<3; /* Current Measurements */
ILmeasB = (ADC_ZERO - (signed)(get_ADC2()))<<3; /* Inverted on CS-IIB   */
ILmeasA -= 0;                                   /* Trim DC offsets      */
ILmeasB += 0;

notchA[1] = (ILmeasA>>1);
notchB[1] = (ILmeasB>>1);

StatRF_11[1] = (ILmeasA>>1);
StatRF_12[1] = (ILmeasB>>1);
StatRF_21[1] = (ILmeasA>>1);
StatRF_22[1] = (ILmeasB>>1);

/* Third ADC conversion start */
reg(ADCTRL1) = ADCEMULATOR|ADC1EN|ADC2EN|ADC_VAC1|ADC_VAC2|ADCSTART;


if (Ctrl_Type==NOTCH)   /* Delta Operator based Notch Filter */
{
    Delta_2nd(notchA);
    Delta_2nd(notchB);

    IrefA += ((long)(notchA[0]<<4)*(long)mag_af)>>16;
    IrefB += ((long)(notchB[0]<<4)*(long)mag_af)>>16;
}
else if (Ctrl_Type==STATRF)
{
    Delta_2nd_x4(StatRF);

    IrefA += ((long)((StatRF_11[0]+StatRF_12[0])<<4)*(long)mag_af)>>16;
    IrefB += ((long)((StatRF_22[0]-StatRF_21[0])<<4)*(long)mag_af)>>16;
}
else if (Ctrl_Type==SRF)
{
    ks = sin_table_A[index_d];
    kc = sin_table_A[index_q];

    /*  abc --> dq transformation  */
    tmp1 = ((int)(((long)k1*(long)((ILmeasA>>1)<<2))>>16) +
           (int)(((long)k2*(long)((ILmeasA>>1)<<2))>>16))<<2;

    tmp2 = (((ILmeasB>>1)<<2)<<1) + ((ILmeasA>>1)<<2);
    tmp2 = (((long)k3*(long)tmp2)>>16)<<2;

    xy_2[1]  =  (int)(((long)kc*(long)tmp1)>>16) +
                (int)(((long)ks*(long)tmp2)>>16);
    xy_2[5]  =  (int)(((long)kc*(long)tmp2)>>16) -
                (int)(((long)ks*(long)tmp1)>>16);

    IIR_1stx2(xy_2);

    /*  dq --> abc transformation  */
                                        /* k1*(kc*idf0 - ks*iqf0) */
    iaf = (((long)k1*(long)( (int)(((long)kc*(long)(xy_2[0]))>>16) -
                             (int)(((long)ks*(long)(xy_2[4]))>>16)  ))>>16);
```

```
                    /* idf0*(k3*ks - k2*kc) + iqf0*(k2*ks + k3*kc) */
    ibf = (int)(((long)(xy_2[0])*(long)( (int)((k3*(long)ks)>>16) -
                                         (int)((k2*(long)kc)>>16)   ))>>16)
        + (int)(((long)(xy_2[4])*(long)( (int)((k2*(long)ks)>>16) +
                                         (int)((k3*(long)kc)>>16)   ))>>16);

    IrefA += ((long)(iaf<<4)*(long)mag_af)>>16;
    IrefB += ((long)(ibf<<4)*(long)mag_af)>>16;
}

reg(PBDATDIR) = 0x0c00;     /* Timing off                              */
reg(PBDATDIR) = 0x0c04;     /* Timing on                               */

/* Store the previous measured currents */
ImeasA3 = ImeasA2;
ImeasB3 = ImeasB2;
ImeasA2 = ImeasA;
ImeasB2 = ImeasB;
/* Read second ADC result : should be ready by now */
ImeasA = (ADC_ZERO - (signed)(get_ADC1()))<<3; /* Current Measurements */
ImeasB = (ADC_ZERO - (signed)(get_ADC2()))<<3; /* Inverted on CS-IIB   */
ImeasA -= 64;                                  /* Trim DC offsets      */
ImeasB += 40;

IcalcC = - ImeasA - ImeasB;

/* Fourth ADC conversion start */
reg(ADCTRL1) = ADCEMULATOR|ADC1EN|ADC2EN|ADC_VDC1|ADC_VDC2|ADCSTART;


/* Calculate DC voltage scaling factor */
Vdc_sc = adc_Vdc>>s2;
if ((adc_Vdc < V_DC_MIN) && is_modulating) /* Test for low volts */
{
    Vscale = 32767;
    fault |= LOW_DC_VOLTS;
    PWM_fast_stop();
}
else if ((adc_Vdc > V_DC_MAX) && (adc_Vdc < 513)) /* Test for high volts */
{
    fault |= HIGH_DC_VOLTS;
    PWM_fast_stop();
}
else
{
    Vscale = fastdivu(v_scale_int,Vdc_sc)<<s1;
}

/* Calculate A Phase current error and scale */
VrefA_0 = (IrefA<<1) + IrefA - (IrefA2<<1)- ImeasA; /* Feedforward     */

if (VrefA_0 > i_max_err)
{
    VrefA_0 = i_max_err<<s4;
}
else if (VrefA_0 < -i_max_err)
```



```
{
    VrefA_0 = -i_max_err<<s4;
}
else
{
    VrefA_0 <<= s4;
}
VrefA_0 = ((long)i_scale_int*(long)VrefA_0)>>16;
VrefA_0 = (VrefA_0+1)/2;
g_a = VrefA_0;

/* Calculate B Phase current error and scale */
VrefB_0 = (IrefB<<1) + IrefB - (IrefB2<<1)- ImeasB; /* Feedforward     */

if (VrefB_0 > i_max_err)
{
    VrefB_0 = i_max_err<<s4;
}
else if (VrefB_0 < -i_max_err)
{
    VrefB_0 = -i_max_err<<s4;
}
else
{
    VrefB_0 <<= s4;
}
VrefB_0 = ((long)i_scale_int*(long)VrefB_0)>>16;
VrefB_0 = (VrefB_0+1)/2;

Timing_R2_off();
Timing_R2_on();

/* Read third ADC result : should be ready by now */
EmfBA = ADC_ZERO - get_ADC1(); /* Voltage Measurements Inverted on IIB */
EmfCA = ADC_ZERO - get_ADC2();

/* Calculate line voltages from line-to-line for Emfs */
EmfA_1 = - EmfBA   - EmfCA;
EmfB_1 =   EmfBA*2 - EmfCA;

/* Fourth ADC conversion start - misc conversion */
reg(ADCTRL1) = ADCEMULATOR|ADC1EN|ADC2EN|ADCSTART
                    |((adc_index_d + (adc_index_d<<3))<<1);

if (is_PCR)
{
    /* Finish Vref calculation */

    VrefA_0 = 4*EmfA_1 - 2*EmfA_2 + VrefA_0;
    if (VrefA_0 > v_max_err)
    {
        VrefA_0 = (v_max_err<<s3);
    }
    else if (VrefA_0 < -v_max_err)
    {
        VrefA_0 = -(v_max_err<<s3);
```

```
    }
    else
    {
        VrefA_0 = VrefA_0<<s3;
    }

    /* Scale voltage to switching time */
    V_A = (((long)Vscale*(long)VrefA_0)>>16) - VactA_1;

    /* Finish Vref calculation */
    VrefB_0 = 4*EmfB_1 - 2*EmfB_2 + VrefB_0;
    if (VrefB_0 > v_max_err)
    {
        VrefB_0 = (v_max_err<<s3);
    }
    else if (VrefB_0 < -v_max_err)
    {
        VrefB_0 = -(v_max_err<<s3);
    }
    else
    {
        VrefB_0 = VrefB_0<<s3;
    }

    /* Scale voltage to switching time */
    V_B = (((long)Vscale*(long)VrefB_0)>>16) - VactB_1;
}
else /* is_PCR == FALSE */
{
    /* Instant VSI */
    VrefA_0 = PERIOD_2<<3;
    V_A = (((long)IrefA<<3)*(long)VrefA_0)>>16;
    V_A = (V_A+1)>>1;
    V_B = (((long)IrefB<<3)*(long)VrefA_0)>>16;
    V_B = (V_B+1)>>1;
}

V_C = -V_A - V_B;                /* Find Vb from Va and Vc              */

/* Determine offset for effective 3rd harmonic injection */
/* Voff = -(max(Va,Vb,Vc)+min(Va,Vb,Vc))/2;                */
if (V_A > V_B)
{
    if (V_A > V_C)
    {
        if (V_B > V_C)  Voff = V_B>>1;
        else            Voff = V_C>>1;
    }
    else
    {
        Voff = V_A>>1;
    }
}
else
{
    if (V_B > V_C)
```



```
    {
        if (V_A > V_C)  Voff = V_A>>1;
        else            Voff = V_C>>1;
    }
    else
    {
        Voff = V_B>>1;
    }
}

/* Add offset into V_A, V_B and V_C to center the vectors */
V_A   = V_A + Voff;
V_Anx = V_A;
V_B   = V_B + Voff;
V_Bnx = V_B;
V_C   = V_C + Voff;
V_Cnx = V_C;

/* Deadband compensation */
if (int2_vect == VECT_T1PINT)        /* Upper switch turn on interrupt */
{
    if (ImeasA > 80) V_A += dead_band;/* Switch on earlier             */
    if (ImeasB > 80) V_B += dead_band;/* 160 is about 1 amp            */
    if (IcalcC > 80) V_C += dead_band;
}
else
{
    if (ImeasA < 80) V_A += dead_band;
    if (ImeasB < 80) V_B += dead_band;
    if (IcalcC < 80) V_C += dead_band;
}

/* Clamp switch times for pulse deletion and load registers */
if (V_A > MAX_TIME)
{
    reg(CMPR1) = 0;
    V_Anx = PERIOD_2;
    A_sat--;
}
else if (V_A < -MAX_TIME)
{
    if ((A_sat==0) && (int2_vect == VECT_T1UFINT))
        reg(CMPR1) = PERIOD - 1;
    else
        reg(CMPR1) = PERIOD;
    A_sat++;
    V_Anx = -PERIOD_2;
}
else
{
    if ((A_sat>0) && (int2_vect == VECT_T1UFINT))
    {
        reg(CMPR1) = PERIOD;
        V_Anx = -PERIOD_2;
    }
    else
```

```
        reg(CMPR1) = PERIOD_2 - V_A;
    A_sat = 0;
}
if ((A_sat > SAT_LIMIT)||(A_sat < -SAT_LIMIT))
{
    if ((is_modulating == TRUE) && (is_sync == TRUE))
    {
        PWM_fast_stop();
        fault |= LOST_SUPPLY;
        ZX_state = ZX_LOST;             /* Force ZX machine to restart */
    }
}

if (V_B > MAX_TIME)
{
    reg(CMPR2) = 0;
    V_Bnx = PERIOD_2;
    B_sat--;
}
else if (V_B < -MAX_TIME)
{
    if ((B_sat==0) && (int2_vect == VECT_T1UFINT))
        reg(CMPR2) = PERIOD - 1;
    else
        reg(CMPR2) = PERIOD;
    B_sat++;
    V_Bnx = -PERIOD_2;
}
else
{
    if ((B_sat>0) && (int2_vect == VECT_T1UFINT))
    {
        reg(CMPR2) = PERIOD;
        V_Bnx = -PERIOD_2;
    }
    else
        reg(CMPR2) = PERIOD_2 - V_B;
    B_sat = 0;
}
if ((B_sat > SAT_LIMIT)||(B_sat < -SAT_LIMIT))
{
    if ((is_modulating == TRUE)&&(is_sync==TRUE))
    {
        PWM_fast_stop();
        fault |= LOST_SUPPLY;
        ZX_state = ZX_LOST;             /* Force ZX machine to restart */
    }
}

if (V_C > MAX_TIME)
{
    reg(CMPR3) = 0;
    V_Cnx = PERIOD_2;
    C_sat--;
}
else if (V_C < -MAX_TIME)
```



```
{
    if ((C_sat==0) && (int2_vect == VECT_T1UFINT))
        reg(CMPR3) = PERIOD - 1;
    else
        reg(CMPR3) = PERIOD;
    C_sat++;
    V_Cnx = -PERIOD_2;
}
else
{
    if ((C_sat>0) && (int2_vect == VECT_T1UFINT))
    {
        reg(CMPR3) = PERIOD;
        V_Cnx = -PERIOD_2;
    }
    else
        reg(CMPR3) = PERIOD_2 - V_C;
    C_sat = 0;
}
if ((C_sat > SAT_LIMIT)||(C_sat < -SAT_LIMIT))
{
    if ((is_modulating == TRUE)&&(is_sync==TRUE))
    {
        PWM_fast_stop();
        fault |= (LOST_SUPPLY|LOST_SYNC);
        ZX_state = ZX_LOST;             /* Force ZX machine to restart */
    }
}

VactA_3 = VactA_2;
VactA_2 = VactA_1;
spare1  = (V_Anx<<1) - V_Bnx - V_Cnx;
VactA_1 = ((long)spare1*(long)Vthird)>>16;

VactB_3 = VactB_2;
VactB_2 = VactB_1;
spare1  = (V_Bnx<<1) - V_Anx - V_Cnx;
VactB_1 = ((long)spare1*(long)Vthird)>>16;



if (g_con == GRAB_GO)
{
    if (g_idx < G_COUNT)
    {
        g[g_idx][ 0] = ImeasA;
        g[g_idx][ 1] = IrefA;
        g[g_idx][ 2] = VrefA_0;
        g[g_idx][ 3] = ImeasB;
        g[g_idx][ 4] = IrefB;
        g[g_idx][ 5] = VrefB_0;
        g[g_idx][ 6] = adc_Vdc;
        g[g_idx][ 7] = EmfBA;
        g[g_idx][ 8] = EmfCA;
        g[g_idx][ 9] = Voff;
        g[g_idx][10] = VactA_1;
```

```
            g[g_idx][11] = VactB_1;
            g[g_idx][12] = mag_d;
            g[g_idx][13] = ILmeasA;
            g[g_idx][14] = ILmeasB;
            g[g_idx][15] = StatRF_11[0]+StatRF_12[0];
            g[g_idx][16] = StatRF_22[0]-StatRF_21[0];
            g[g_idx][17] = Vdc_sum;
            g[g_idx][18] = Vdc_meas;
            g_idx++;
        }
        else
        {
            g_con = GRAB_DONE;
            g_idx = 0;
        }
    }
    Timing_R2_off();
    Timing_R2_on();

    /* Read fourth ADC result : should be ready by now */
    adc_Vdc  = get_ADC2();              /* Grab Vdc                    */
    adc_Vdc -= ADC_ZERO;                /* Remove offset               */
    spare1   = get_ADC1();

    /* Read fifth ADC result : should be ready by now
    a_val[adc_index_d]   = get_ADC1();
    a_val[adc_index_d+8] = get_ADC2();
    adc_index_d = (adc_index_d+1)&0x07;

    /* Ready for next interrupt */
    reg(ADCTRL1) = ADCEMULATOR|ADC1EN|ADC2EN|ADC_I4|ADC_I5;

    Timing_R2_off();

} /* End INT2Service */


/*************************************************************************/
/*  NAME:        IN42Service()                                           */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Interrupt for Zero Crossing Detection                   */
/*  NOTES:                                                               */
/*  HISTORY:     13/09/2001  MN  Altered to suit CS-IIB                  */
/*************************************************************************/
interrupt void INT4Service(void)
{
    int4_vect = reg(EVIVRC);
    ZX_seen = TRUE;

#if   (CAP_PORT == CAP_PORT3)
    ZX_time = PERIOD - reg(CAP3FIFO);                   /* ZX from V1 */
#elif (CAP_PORT == CAP_PORT4)
    ZX_time = PERIOD - reg(CAP4FIFO);                   /* ZX from V4 */
#endif

} /* End INT4Service */
```



```
/*************************************************************************/
/*  NAME:        PWM_Init()                                              */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Sets up registers for PWM                               */
/*  NOTES:       Need to initialize period_2, max_time, dead_band        */
/*               before calling.                                         */
/*  HISTORY:     13/09/2001  MN  Altered to suit CS-IIB                  */
/*************************************************************************/
void PWM_Init(Mode op_mode)
{
    reg(EVIFRA) = PDPINT;

    reg(ADCTRL1) = ADCEMULATOR|ADC1EN|ADC2EN|ADC_I4|ADC_I5;

    reg(COMCON) = 0x2b17; /* disable full compare - un-shadows registers */
    reg(T1CON)  = 0xa806; /* stop timer */
    reg(SACTR)  = 0x0000;
    reg(ACTR)   = 0x0000;

    reg(CAPFIFO) = 0x00ff;  /* clear the fifo capture inputs */

    ImeasA = 0;
    ImeasB = 0;
    IcalcC = 0;

    /* Set operating mode */
    is_PCR  = op_mode.is_PCR;
    is_sync = op_mode.is_sync;
    is_mag  = op_mode.is_mag;
    is_Vdc  = ( is_sync && is_PCR && (!is_mag) )?TRUE:FALSE;

    /* Calculate sin tables */
    sin_table(sin_table_A,TABLE_SIZE,32767,0,0);
    sin_table(sin_table_B,TABLE_SIZE,32767,-21845,0);

    reg(GPTCON) = 0x047a; /* 000 00 10 00 1 11 10 10 */
     /* Timer control       |   |  |  |  | |  |  +- T1CMPR active high
      *                     |   |  |  |  | |  +- T2CMPR active high
      *                     |   |  |  |  | +- T3CMPR forced high
      *                     |   |  |  |  +- compare output enable
      *                     |   |  |  +- T1toADC
      *                     |   |  +- T2toADC starts on period
      *                     |   +- T3toADC
      *                     +- timer status bits */

     /* Action control register */
     reg(ACTR) = 0x0000; /* 0 000 00 00 00 00 00 00 */
      /*                  |  |  |  |  |  |  |  |
       *                  |  |  +--+--+--+--+--+- all pwm pins forced low
       *                  |  +- space vector bits
       *                  +- space vector disabled */

     /* Simple action control register */
     reg(SACTR) = 0x0000; /* 0000 0000 00 00 00 00 */
```

```
 /*                    \       /  |  |  |
  *                      \   /    |  +--+- all pwm pins forced low
  *                        \/     +- sample and hold active high
  *                         - reserved */

/* Deadband timer control */
reg(DBTCON) = 0x00e0|(DEADCNT<<8); /* xxxx xxxx 111 00 000 */
 /*                                  |      | |   |  +- reserved
  *                                   \    /  |   +- prescaler
  *                                     \ /    +- enabled all pins
  *                                      +- deadband time */

reg(CMPR1)  = PERIOD_2;
reg(CMPR2)  = PERIOD_2;
reg(CMPR3)  = PERIOD_2;
reg(SCMPR1) = 0;
reg(SCMPR2) = 0;
reg(SCMPR3) = 0;

/* Compare control register */
reg(COMCON) = 0x2b17; /* 0 01 0 10 1 1 0 00 10 111 */
reg(COMCON) = 0xab17; /* 1 01 0 10 1 1 0 00 10 111 */
 /*                    | |  | |  | | | |  |  +- pins in pwm mode
  *                     | |  | |  | | | |  +- reload SACTR immediately
  *                     | |  | |  | | | +- reload SCMPRx on zero or period
  *                     | |  | |  | | +- use timer1
  *                     | |  | |  | +- simple compare enabled
  *                     | |  | |  +- full compare enabled
  *                     | |  | +- reload ACTR immediately
  *                     | |  +- space vector mode disabled
  *                     | +- CMPRx reload on zero or period
  *                     +- compare enable */

reg(T2PR)   = PERIOD-1;
reg(T2CNT)  = 0;
reg(T2CMPR) = 3*PERIOD/4;

reg(T2CON)  = 0x9082;/* 10 010 000 1 0 00 00 1 0 */
 /* Timer 2           |  |   |   | | |  |  | +- reserved
  *                   |  |   |   | | |  |  +- timer compare enable
  *                   |  |   |   | | |  +- timer compare reload on zero
  *                   |  |   |   | | +- clock source - internal
  *                   |  |   |   | +- timer enable
  *                   |  |   |   +- enable on timer 1 enable
  *                   |  |   +- input clock prescaler
  *                   |  +- count mode selection - continuous up mode
  *                   +- emulation control bits - timer continues */

reg(T1PR)  = PERIOD;
reg(T1CNT) = 0;

/* NOTE: must start timer the first time, and load the various registers
 * which are shadowed and only reload on timer overflow */
reg(T1CON) = 0xa802; /* 10 101 000 0 0 00 00 1 0 */
 /* Timer 1           |  |   |   | | |  |  | +- reserved
  *                   |  |   |   | | |  |  +- timer compare enable
```



```
  *                   |  |   |   | | |  +- timer compare reload on zero
  *                   |  |   |   | | +- clock source - internal
  *                   |  |   |   | +- timer enable
  *                   |  |   |   +- reserved
  *                   |  |   +- input clock prescaler
  *                   |  +- count mode selection - continuous up mode
  *                   +- emulation control bits - timer continues */


/* Enable timer 1 period and undeflow interrupt */
reg(EVIFRA)  = (T1PINT|T1UFINT);
reg(EVIMRA) |= (T1PINT|T1UFINT);
reg(EVIMRB)  = 0;

/* Enable INT2 interrupt */
reg(IFR)  = ENABLE_INT2;
reg(IMR) |= ENABLE_INT2;

/* start timer 1 */
reg(T1CON) = 0xa842; /* 10 101 000 0 1 00 00 1 0 */
 /* Timer 1           |  |   |   | | |  |  | +- reserved
  *                   |  |   |   | | |  |  +- timer compare enable
  *                   |  |   |   | | |  +- timer compare reload on zero
  *                   |  |   |   | | +- clock source - internal
  *                   |  |   |   | +- timer enable
  *                   |  |   |   +- reserved
  *                   |  |   +- input clock prescaler
  *                   |  +- count mode selection - continuous up mode
  *                   +- emulation control bits - timer continues */

reg(CAPCON) = 0xB855;  /* 1 01 1 1 0 0 0 01 01 01 01 */
 /* Capture control       | |  | | | | | |  |  +--+- positive edge detect
                          | |  | | | | | +--+- positive edge detect
                          | |  | | | | +- cap4 starts adc
                          | |  | | +-+- cap 34/12 use timer 2
                          | |  +-+- enable cap 3/4
                          | +- enable cap 12 / disable QEP
                          +- don't reset capture unit */
    if (is_sync == TRUE) /* don't need ZX interrupt if running as current source */
    {
        reg(EVIFRC) = 0x000f;       /* Reset capture input flags          */

#if   (CAP_PORT == CAP_PORT3)
        reg(EVIMRC) = CAP3INT;      /* Enable capture 3 input interrupt   */
#elif (CAP_PORT == CAP_PORT4)
        reg(EVIMRC) = CAP4INT;      /* Enable capture 4 input interrupt   */
#endif

        reg(IFR) |= ENABLE_INT4;    /* Enable interrupt 4                 */
        reg(IMR) |= ENABLE_INT4;
    }
    phase = 0;
    phi = 0;
    phase_step = 328;               /* Just a reasonable value ~50Hz      */
    index_d = 0;
    mag_d = 0;
```

```
    mag_q = 0;
    mag_af = 0;
    Vdc_ref = 0;
    Vdc_sum = 0;
    ZX_state = ZX_LOST;
    ZX_count = 0;
    ZX_in_sync = 0;
    ZX_seen = FALSE;
    ZX_cycles = 0;
    ZX_sum = 0;
    ZX_phase_err = 0;
    ZX_err_sum = 0;
    in_sync = FALSE;
    A_sat = 0;
    B_sat = 0;
    C_sat = 0;
    is_modulating = FALSE;

    s5 = log2((long)(32767/I_SCALE+0.5));
    s6 = log2((long)(65535/V_SCALE+0.5));
    v_scale_int = (unsigned int)(V_SCALE*(1<<s6)+0.5);
    i_scale_int = (signed int)(I_SCALE*(1<<s5)+0.5);
    s2 = log2((long)(Vdc_MAX/250.0/V_DC_SC+0.5));
    s1 = log2((long)(Vdc_MIN*32000.0/(V_DC_SC*v_scale_int)+0.5)) - s2;
    s3 = 16 - s1 - s2 - s6;
    s4 = 14 - s5;
    i_max_err = (1<<(15-s4)) - 1;
    v_max_err = (1<<(15-s3)) - 1;
    if ((s3 > 3)||((s1*s2*s3*s4*s5*s6) < 0))
    {
        fault |= SCALE_ERROR;
    }

} /* End PWM_Init */


/*****************************************************************************/
/*  NAME:        PWM_Start()                                                 */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: Sets up registers to start PWM modulation.                  */
/*  NOTES:       Called from background code.                                */
/*               Due to deadband/ACTR bug - re-initializes the entire        */
/*               PWM timers stuff                                            */
/*  HISTORY:     13/09/2001  MN  Altered to suit CS-IIB                      */
/*****************************************************************************/
void PWM_Start(void)
{
    unsigned int int_mask,i;

    int_mask = reg(EVIMRA);
    reg(EVIMRA) &= ~(T1PINT|T1UFINT);   /* Disable PWM interrupt */

    /* Force PWM outputs low before initializing them */
    reg(COMCON) = 0x2b17;               /* Disable full compare outputs */
    reg(T1CON) = 0xa806;                /* Stop timer */
    reg(SACTR) = 0x0000;
```



```
    reg(ACTR) = 0x0000;

    reg(GPTCON) = 0x047a;               /* 000 00 10 00 1 11 10 10 */
    reg(ACTR) = 0x0666;                 /* 0 000 01 10 01 10 01 10 */
    reg(SACTR) = 0x0000;                /* 0000 0000 00 10 00 00 */
    reg(DBTCON) = 0x00e0|(DEADCNT<<8);  /* xxxx xxxx 111 00 000 */

    reg(CMPR1) = PERIOD_2;
    reg(CMPR2) = PERIOD_2;
    reg(CMPR3) = PERIOD_2;
    reg(SCMPR1) = 0;
    reg(SCMPR2) = 0;
    reg(SCMPR3) = 0;

    /* Compare control register */
    reg(COMCON) = 0x2b17;               /* 0 01 0 10 1 1 0 00 10 111 */
    reg(COMCON) = 0xab17;               /* 1 01 0 10 1 1 0 00 10 111 */
    reg(T2PR) = PERIOD-1;
/* <<< stop timer 1 & 2 */
    reg(T1CON) = 0xa802;                /* 10 101 000 0 0 00 00 1 0 */

    reg(T2CNT) = 0;
    reg(T2CMPR) = 3*PERIOD/4;

    reg(T2CON) = 0x9082;                /* 10 010 000 1 0 00 00 1 0 */

    reg(T1PR) = PERIOD;
    reg(T1CNT) = 0;

    reg(T1CON) = 0xa802;                /* 10 101 000 0 0 00 00 1 0 */

    reg(T1CON) = 0xa842;                /* 10 101 000 0 1 00 00 1 0 */

    mag_d = 0;
    mag_q = 0;
    mag_af = 0;
    Vdc_ref = 0;
    Vdc_sum = 0;
    Vthird = (65536L/3);
    is_modulating = TRUE;
    A_sat = 0;
    B_sat = 0;
    C_sat = 0;

    /* Set up registersfor ADC measurements */
    reg(ADCTRL1) = ADCEMULATOR|ADC1EN|ADC2EN|ADC_I4|ADC_I5;
    reg(ADCTRL2) = ADCEVSOC|ADCPSCALE;  /*0x0403;*/

    /* Re-enable interrupts */
    reg(EVIFRA) = (T1PINT|T1UFINT);
    reg(EVIMRA) = int_mask;

} /* End PWM_Start */

/*****************************************************************************/
/*  NAME:        PWM_Stop()                                                  */
```

```
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Stops the PWM but leaves the interrupt running          */
/*  NOTES:       Called from background code.                            */
/*************************************************************************/
inline void PWM_Stop(void)
{
    PWM_fast_stop();
} /* End PWM_Stop */


/*************************************************************************/
/*  NAME:        fault_init()                                            */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Uses fault_mask to initialise the correct level         */
/*               comparison fault interrupt triggers.                    */
/*  NOTES:       Called from background code.                            */
/*  HISTORY:     13/09/2001  MN  Altered to suit CS-IIB                  */
/*************************************************************************/
void Fault_init(unsigned int fault_mask)
{
    fault = NO_FAULT;

    /* Initialize the level comparison interrupts */
    if (fault_mask&LC_VDC1)
    {
        reg(XINT1) = 0x8004; /* Clear interrupt flag, rising edge triggered */
    }

    /* Enable INT1 interrupt */
    if (fault_mask&(LC_VDC1))
    {
        reg(IFR)  = ENABLE_INT1;
        reg(IMR) |= ENABLE_INT1;
        if (fault_mask&LC_VDC1)  reg(XINT1) |= 0x0001; /* enable interrupt */
        /*if (fault_mask&LC_I_MOD) reg(XINT3) |= 0x0001;*/ /* enable interrupt */
    }
} /* End Fault_init */


/*************************************************************************/
/*  NAME:        XINT1_int()                                             */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Over DC bus fault detection interrupt - XINT2           */
/*  NOTES:                                                               */
/*  HISTORY:     13/09/2001  MN  Altered to suit CS-IIB                  */
/*************************************************************************/
interrupt void XINT1_int(void)
{
    PWM_fast_stop();
    fault |= LC_VDC1;
} /* end XINT1_int */


/*************************************************************************/
/*  NAME:        check_faults()                                          */
/*  RETURNS:     unsigned integer - value of fault, if any.              */
```



```
/*  DESCRIPTION: Clears the faults and checks that they have cleared.    */
/*  NOTES:       Called from background code.                            */
/*  HISTORY:     13/09/2001  MN  Altered to suit CS-IIB                  */
/*************************************************************************/
unsigned int check_faults(unsigned int fault_mask)
{
    unsigned int val = NO_FAULT;

    if (fault_mask&PDP_FAULT)
    {
        reg(EVIFRA) = 0x0001; /* clear and test for return */
        val = (reg(EVIFRA)&0x0001)?PDP_FAULT:0; /* for PDPINT */
    }

    /* Test LC_VDC1 pin (XINT1) */
    if (fault_mask&LC_VDC1)
    {
        val |= (reg(XINT1)&0x0040)?LC_VDC1:0;
    }

    /* Test level of ZX_in_sync for re-synchronization */
    if (fault_mask&LOST_SYNC)
    {
        val |= (ZX_in_sync<ZX_SYNC_LIMIT)?LOST_SYNC:0;
    }

    /* Clear X_sat counters for lost supply */
    if (fault_mask&LOST_SUPPLY)
    {
        A_sat = 0;
        B_sat = 0;
        C_sat = 0;
    }

    /* Test DC voltage */
    if (fault_mask&LOW_DC_VOLTS)
    {
        val |= (adc_Vdc<V_DC_MIN)?LOW_DC_VOLTS:0;
    }

    if (fault_mask&HIGH_DC_VOLTS)
    {
        val |= (adc_Vdc>V_DC_MAX)?HIGH_DC_VOLTS:0;
    }

    fault = val;
    return val&fault_mask;

} /* End check_faults */

/* gets the operating mode in *m and returns the fault status */
unsigned int GetMode(Mode *m)
{
    m->is_mod  = is_modulating;
    m->in_sync = in_sync;
    m->is_PCR  = is_PCR;
```

```
    m->is_mag  = is_mag;
    m->is_sync = is_sync;

    return fault;

} /* End GetMode */


/*************************************************************************/
/*  NAME:        SetMag(signed int, signed int, signed int)              */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Sets the magnitude reference with direct and quadrature */
/*               magnitudes.                                             */
/*  NOTES:       Called from background code.                            */
/*               Mag = 32767 == 200%                                     */
/*               mX has 8 bits afters the point. ie 256d == 1% mod depth */
/*************************************************************************/
inline void SetMag(signed int md, signed int mq, signed int maf)
{
    mag_d = md;
    mag_q = mq;
    mag_af = maf;

} /* End SetMag */


/*************************************************************************/
/*  NAME:        GetMagD(void)                                           */
/*  RETURNS:     signed int                                              */
/*  DESCRIPTION: Gets the direct (real) magnitude reference.             */
/*  NOTES:       Called from background code.                            */
/*               Mag = 32767 == 200%                                     */
/*               mX has 8 bits afters the point. ie 256d == 1% mod depth */
/*************************************************************************/
inline signed int GetMagD(void)
{
    return mag_d;

} /* End GetMag */


/*************************************************************************/
/*  NAME:        SetMagAF(signed int)                                    */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Sets the active filter magnitude reference              */
/*  NOTES:       Called from background code.                            */
/*************************************************************************/
inline void SetMagAF(signed int maf)
{
    mag_af = maf;

} /* End SetMagAF */


/*************************************************************************/
/*  NAME:        SetMagDC(signed int)                                    */
```



```
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Sets the DC bus magnitude reference                     */
/*  NOTES:       Called from background code.                            */
/*************************************************************************/
inline void SetMagDC(signed int mdc_ref)
{
    Vdc_ref = mdc_ref;

} /* End SetMagDC */


/*************************************************************************/
/*  NAME:        GetMagDC(void)                                          */
/*  RETURNS:     signed int                                              */
/*  DESCRIPTION: Gets the DC voltage magnitude reference.                */
/*  NOTES:       Called from background code.                            */
/*************************************************************************/
inline signed int GetMagDC(void)
{
    return Vdc_ref;

} /* End GetMagDC */


/*************************************************************************/
/*  NAME:        SetPhase(unsigned int)                                  */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Sets the phase offset for the reference (65536 == 360deg) */
/*  NOTES:       Called from background code.                            */
/*************************************************************************/
inline void SetPhase(unsigned int phase_offset)
{
    phi = phase_offset;

} /* End SetPhase */


/*************************************************************************/
/*  NAME:        GetPhase(void)                                          */
/*  RETURNS:     unsigned int - current phase offset (65536 == 360deg)   */
/*  DESCRIPTION: Gets the phase offset for the reference                 */
/*  NOTES:       Called from background code.                            */
/*************************************************************************/
inline unsigned int GetPhase(void)
{
    return phi;

} /* End GetPhase */


/*************************************************************************/
/*  NAME:        IncrPhase(signed int)                                   */
/*  RETURNS:     unsigned int - incremented phase offset (65536 == 360deg) */
/*  DESCRIPTION: Increments the phase offset for the reference           */
/*  NOTES:       Called from background code.                            */
/*************************************************************************/
```

```
/* moves the phase offset, returning the new value, 32767 == 180deg */
inline unsigned int IncrPhase(signed int dphi)
{
    return (phi += dphi);

} /* End IncrPhase */


/*************************************************************************/
/*  NAME:        GetFundFreq(void)                                       */
/*  RETURNS:     integer - fundametal frequency achieved (1/256th of a Hz) */
/*  DESCRIPTION: Gets the fundamental frequency.                         */
/*  NOTES:       Called from background code.                            */
/*************************************************************************/
signed int GetFundFreq(void)
{
    return (signed int)( (((long)phase_step)*(CPU_FREQ/PERIOD))/256L);

} /* End GetFundFreq */


/*************************************************************************/
/*  NAME:        SetFundFreq(signed int)                                 */
/*  RETURNS:     integer - fundametal frequency achieved                 */
/*  DESCRIPTION: Sets the fundamental frequency.                         */
/*  NOTES:       Called from background code.                            */
/*               ff has 8 bits afters the point. ie 256d == 1Hz          */
/*************************************************************************/
void SetFundFreq(signed int ff)
{
    if (is_sync == FALSE) /* cannot set fund freq if synchronizing */
    {
        phase_step = (unsigned int)((256L*(long)ff)/(CPU_FREQ/PERIOD));
    }
} /* End SetFundFreq */

/* Sets the level of deadband compensation in nanoseconds */
inline void SetDeadBand(unsigned int d)
{
    if (d > MAX_DEADBAND) d = MAX_DEADBAND;
    dead_band = d / (unsigned int)(1e9 / CPU_FREQ);
} /* End SetDeadBand */
```


E.2 pwmback.c: C Background Code

```
/*************************************************************************/
/*                                                                       */
/* Application: PCR background routines                                  */
/* Developed By:Monash University                                        */
/* Author:      M.Newman and A.McIver                                    */
/* File:        pcrmain.c                                                */
/* History:     derived from VSI code                                    */
/*                                                                       */
/*    19/07/1999  AM  Porting to new library and HPI hardware            */
/*    05/09/2001  MN  Altered to work with the MiniDSP and CS-IIB        */
/*                                                                       */
/*************************************************************************/

#include <c240.h>
#define  MiniDSPpcb     1
#include <mu_pcb.h>
#include <stdlib.h>
#include <conio.h>
#include <adc.h>
#include <intrpt.h>
#include <iospace.h>
#include <math.h>
#include <iib.h>
#include "pcr.h"
#include "timer.h"
#include "shunt.h"


/*************************************************************************/
/*                          VARIABLE DECLARATIONS                        */
/*************************************************************************/

signed int
    Idemand,                /* Demanded current in 256th of an amp       */
    Ia = 0,                 /* Measured current in 256th of an amp       */
    mod_depth,              /* Mod depth in 8ths of a percent            */
    Vdc = 0,                /* DC bus voltage in volts                   */
    mag_af_bak,             /* Background level active filter mag. ref.  */
    mag_dc_bak,             /* Background level dc voltage mag. ref.     */
    mag_bak;                /* Background level magnitude reference      */


unsigned int
    dead_band_comp = 0xffff,/* Dead time compensation level in nanoseconds */
    sw_freq    = 0,         /* Switching frequency                       */
    Ipeak,                  /* Peak measured current in 256th of an amp  */
    old_fault  = NO_FAULT,  /* Known faults                              */
    freq,                   /* Fundamental frequency                     */
    digin,                  /* Mirror of the digital inputs              */
    a_val[16]  = {0},       /* Latest analog values                      */
    is_af_on   = FALSE,     /* Indicates whether active filtering is on  */
    ref_double = FALSE,     /* Flag to indicate a reference step of double */
    ref_double_bak = FALSE, /* Flag to force reference step of double    */
    step_flag  = FALSE,     /* Indicates that step change is requested   */
    tested_faults;          /* Which faults are tested for               */
```

```
short unsigned int
    ramp_count = 0;         /* Used to slow down VDC Ramp up              */

Mode
    op_mode;                /* Operating mode                             */

timerNumber
    onesec_timer,           /* Times out seconds                          */
    fault_timer;            /* Timer for fault clearing                   */


signed int
    g[G_COUNT][G_SIZE];     /* Grab data                                  */

/* State stuff */
typedef void (* funcPtr)(void);
typedef struct
{
    funcPtr f;
    int first;
} StateType;
StateType state;


/*****************************************************************************/
/*                          FUNCTION DECLARATIONS                            */
/*****************************************************************************/

#define NextState(_s_,_f_)      { _s_.f = &_f_; \
                                  _s_.first = 1; }
#define IsFirstState(_s_)       (_s_.first == 1)
#define DoneFirstState(_s_)     _s_.first = 0
#define IsLastState(_s_)        (_s_.first == 1)
#define DoState(_s_)            ((*(_s_.f))())
#define IsCurrentState(_s_,_f_) (_s_.f == &_f_)

void StateInit(void);
void StateRestart(void);
void StateIsolated(void);
void StateCharge(void);
void StateCloseWait(void);
void StateOpenWait(void);
void StateSync(void);
void StateStop(void);
void StateRun(void);
void StateRampUp(void);
void StateRampDown(void);
void StateFault(void);

void CheckFault(void);      /* Fault detection - uses tested_faults       */

/* Serial port display of variables */
void init_display(StateType state);
void display(char c);
void display_fault(unsigned int fault);
```



```
void display_state(StateType state);
void display_ZXstate(unsigned int zx_state);
void display_CTRLstate(unsigned int ctrl_type);

int keyboard(void);         /* Serial input handling : returns quit status */
void scale_adcs(void);      /* Scales analog values to real units          */
void onesec_events(void);   /* Performs one second events                  */
void get_variables(void);   /* Asks user for relevant parameters           */
void set_variables(void);   /* Sets the parameters with default values     */


/*****************************************************************************/
/*                          FUNCTION DEFINITIONS                             */
/*****************************************************************************/

/*****************************************************************************/
/*  NAME:        main()                                                      */
/*  RETURNS:     integer - (always 0)                                        */
/*  DESCRIPTION: Controls the flow of the background software                */
/*  NOTES:                                                                   */
/*  HISTORY:     05/09/2001  MN  iib_init() added for CS_IIB operation       */
/*               13/09/2001  MN  OCR-A & B overidden to fix iib_init bug     */
/*****************************************************************************/
int main()
{
    char
        quit = 0;
    signed int
        autostart = FALSE,
        count = 0,
        freq_scale,
        mag_scale;
unsigned int i;

/* Standard initialization for the board                                     */
init_pcb();
iib_init();

tested_faults = PDP_FAULT|LC_VDC1|LC_I_MOD|HIGH_DC_VOLTS;

reg(OCRA) = 0x130f; /* 0001 0011 0000 1111
                       \ /| |||| \  / \  /
                        | | ||||  \/   ++-- use IOPA0-3 as ADC in
                        | | ||||   +------- not used
                        | | ||++----------- IOPB0&1 as PWM outputs
                        | | ++------------- IOPB2&3 as dig out
                        | +----------------- IOPB4 as T2PWM (S&H trigger)
                        +------------------- IOPB5-7 as dig out         */

reg(OCRB) = 0x00fd; /* 0000 0000 1111 1101 */
                    /*  \    /   \|\| |||+- ADCSOC
                         \/      | | ||+-- not used
                          |      | | ++--- IOPC2&3 as dig out
                          |      | +------ CAP1 & CAP2
                          |      +-------- IOPC6&7 as CAP3 and CAP4
                          +--------------- not used */
```

```
puts("\n\n\tPCR/VSI controller\n\n");

/* Set up interrupt vectors */
set_SISR_vector(RTI_int,  INT1, WDINT);
set_SISR_vector(XINT1_int,INT6, XINT1_ISR);
set_GISR_vector(INT2Service,INT2);
set_GISR_vector(INT4Service,INT4);

/* Set up registers for ADC measurements */
reg(ADCTRL1) = ADCEMULATOR|ADC1EN|ADC2EN|ADC_I4|ADC_I6;
reg(ADCTRL2) = ADCEVSOC|ADCPSCALE; /* 0x0403 */

digin = ReadDigIn();

/* Test for auto start */
if (get_dips()&DIP_AUTO)        /* Autostart Parameters              */
{
    autostart = TRUE;
    set_variables();
}
else
{
    get_variables();
}

sw_freq = (CPU_FREQ/2/PERIOD_2 + 1)/2;
puts("Switching frequency is ");    putu(sw_freq);
puts("Hz\nhalf_period is ");        putu(PERIOD_2);       puts("\n");
puts("Deadband compensation is: "); putu(dead_band_comp); puts("ns\n");

if (autostart == FALSE)
{
    puts("\nPress any key to start\n");
    getc();
}

freq_scale = (50*163841 + sw_freq/2)/sw_freq;

StateInit();            /* Initialize state machine                  */

init_display(state);    /* Initialise display stuff                  */
while (quit != 1)
{
    GetMode(&op_mode);  /* Retrieve operating mode                   */
    CheckFault();       /* Check for faults                          */
    DoState(state);     /* Call state function                       */
    digin = ReadDigIn();/* Read IIB digital inputs (inverted in hardw.)*/
    quit  = keyboard(); /* Handle keyboard input                     */
    TimerFuncPoll();    /* Run any queued timer functions            */
    scale_adcs();       /* Scale adc measurements                    */
    display(0);         /* Serial port display                       */
}

return 0;
} /* End main */
```



```
/*****************************************************************************/
/*  NAME:        onesec_events()                                             */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: It is the background process task to clear the flags        */
/*  NOTES:                                                                   */
/*  HISTORY:     XX/XX/2001  MN                                              */
/*****************************************************************************/
void onesec_events(void)
{
    ReloadTimer(onesec_timer, 1000 mSEC);

    /* Check for low speed faults */
    if (IsCurrentState(state,StateFault))
    {
        display_fault(fault&tested_faults);
    }
    display(1);

} /* End onesec_events */


/*****************************************************************************/
/*  NAME:        display()                                                   */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: Displays variables and state through serial port            */
/*  NOTES:       Case 0 never happens                                        */
/*  HISTORY:     10/10/2001  MN  Full re-work of display variables           */
/*****************************************************************************/
void display(char c)
{
    signed int
        temp;

    static unsigned int step = 0;

    if (c == TRUE)
    {
        step = 1;
        puts("   \r");
    }
    else if (step > 0)
    {
        step++;
    }
    switch(step)
    {
        case 1:  display_state(state);       break;
        case 2:  puts(" ");                  break;
        case 3:  display_ZXstate(ZX_state);  break;
        case 4:  puts(" ");                  break;
        case 5:  putf(freq,256,1);           break;
        case 6:  puts("Hz ");                break;
        case 7:
                 if(is_af_on)
```

```
            puts("E:");
        else
            puts("D:");
        putf((a_val[AVAL_POT2]-512)*100,128,1);
        break;
case 8: puts("%  ");                    break;
case 9:
        if (IsCurrentState(state,StateCharge))
        {
            putf(Ipeak,256,2);
            puts("A  ");
        }
        else if (op_mode.is_PCR)
        {
            if (op_mode.is_mag)
            {
                putf(Idemand,256,2);
                puts("A  ");
            }
            else
            {
                temp = V_DC_SC*(1L<<14);
                putd( ((long)(mag_dc_bak<<2)*(long)temp)>>16 );
                puts("Vdc  ");
            }
        }
        else
        {
            putf(mod_depth,8,1);
            puts("%  ");
        }
        break;
    case 10: putd(Vdc);                 break;
    case 11: puts("Vdc  ");             break;
    case 12: display_CTRLstate(Ctrl_Type);  break;

    default: step = 0;
  }
} /* End display */


/*************************************************************************/
/*  NAME:        scale_adcs()                                            */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Scales analog values to real units (ie V, A, %)         */
/*  NOTES:                                                               */
/*  HISTORY:     05/09/2001  MN  Analog offsets changed to suit CS-IIB   */
/*               01/10/2001  MN  Pot 2 analog added for active filtering */
/*************************************************************************/
void scale_adcs(void)
{
    signed int
        temp;

    mag_af_bak = (a_val[AVAL_POT2]-512)*MAG_SCALE;  /* +128 is 100%      */
```



```
    if (op_mode.is_mag)                     /* Magnitude control       */
    {
        if(!ref_double_bak)
            mag_bak = (a_val[AVAL_POT1]-512)*MAG_SCALE;
        else
            mag_bak = ((a_val[AVAL_POT1]-512)*MAG_SCALE)<<1;
    }
    else
    {
        mag_dc_bak = a_val[AVAL_POT1];
    }

    if (op_mode.is_PCR)
    {
        temp = (I_AC_SC*(1L<<18));
        Idemand = ((long)mag_bak*(long)temp)>>16;
    }
    else
    {
        temp = 100L*(1<<5);
        mod_depth = ((long)mag_bak*(long)temp)>>16;
    }
    freq = GetFundFreq();

    temp = V_DC_SC*(1L<<14);
    Vdc = ((long)(adc_Vdc<<2)*(long)temp)>>16;

    Ia = (int)(((long)((signed)a_val[AVAL_I1]-504)*(I_AC_SC*1024L))/4);

} /* End scale_adcs */


/*************************************************************************/
/*  NAME:        CheckFault()                                            */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Fault detection - uses tested_faults                    */
/*  NOTES:                                                               */
/*************************************************************************/
void CheckFault(void)
{
    if ((fault&tested_faults)&&(!IsCurrentState(state,StateFault)))
    {
        PWM_Stop();
        ContactorOpen(DIG_CONT|DIG_AUX_CONT);
        old_fault = fault;
        NextState(state,StateFault);
        display_fault(old_fault);
    }
} /* End CheckFault */


/*************************************************************************/
/*  NAME:        StateInit()                                             */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: State used to initialise PWM, state, and other variables */
/*  NOTES:       On completion state is changed to "StateIsolated".      */
```

```
/*  HISTORY:     XX/XX/2001  MN  XXX                                          */
/*****************************************************************************/
void StateInit(void)
{
    SetFundFreq(100*256);
    SetDeadBand(dead_band_comp);
    tested_faults = PDP_FAULT|LC_VDC1|LC_I_MOD|HIGH_DC_VOLTS;
    Fault_init(tested_faults);
    PWM_Init(op_mode);
    EnableInts();

    TimerInit();                /* Start timer for background processes    */
    onesec_timer = GetTimer();  /* For ticking off 1 sec events            */
    StartTimer(onesec_timer, 1000 mSEC, &onesec_events);
    fault_timer = GetTimer();   /* For fault clearing                      */
    NextState(state,StateIsolated);

    SetLamp(LAMP_ON|LAMP_READY|LAMP_FAULT,CLEAR);

} /* End StateInit */


/*****************************************************************************/
/*  NAME:        StateIsolated()                                             */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: State used to keep the inverter isolated from the primary */
/*  NOTES:                                                                   */
/*  HISTORY:     10/09/2001  MN  Created                                     */
/*****************************************************************************/
void StateIsolated(void)
{
    tested_faults = ALL_FAULTS&~(LOW_DC_VOLTS|LOST_SUPPLY|LOST_SYNC);
    ContactorOpen(DIG_CONT|DIG_AUX_CONT);

    if (IsFirstState(state))
    {
        init_display(state);            /* Initialise display header       */
        DoneFirstState(state);
    }

    if ( UnIsolatePressed() )
    {
        NextState(state,StateRestart);
    }

} /* End StateIsolated */


/*****************************************************************************/
/*  NAME:        StateRestart()                                              */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: Restart everything either after a fault or on start up     */
/*  NOTES:                                                                   */
/*****************************************************************************/
void StateRestart(void)
{
```



```
    SetMag(0,0,0);
    SetMagDC(0);
    NextState(state,StateCharge);
    tested_faults = ALL_FAULTS&~(LOW_DC_VOLTS|LOST_SUPPLY|LOST_SYNC);
    SetLamp(LAMP_READY,SET);

} /* End StateRestart */


/*****************************************************************************/
/*  NAME:        StateCloseWait()                                            */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: Closes charge contactor and waits for DC bus to charge     */
/*  NOTES:                                                                   */
/*  HISTORY:     10/09/2001  MN  Isolated State accounted for                */
/*****************************************************************************/
void StateCharge(void)
{
    static int maxIa = 0, oldmaxIa = 0;
    static timerNumber peak_timer;      /* Timer for peak Ia measurement   */
    static timerNumber charge_timer;    /* Timer for dc bus charging       */
    if (IsFirstState(state))
    {
        init_display(state);            /* Initialise display header       */
        DoneFirstState(state);
        maxIa = 0;                      /* Clear maxIa before starting wait*/
        oldmaxIa = 0;
        ContactorClose(DIG_AUX_CONT);
        peak_timer = GetTimer();        /* Timer for the peak meas.        */
        charge_timer = GetTimer();      /* Timer for the charge delay      */
        StartTimer(charge_timer, 10000 mSEC, NULL);
        StartTimer(peak_timer,    2000 mSEC, NULL);
    }

    if ( IsolatePressed() )
    {
        ContactorOpen(DIG_CONT|DIG_AUX_CONT);
        NextState(state,StateIsolated);
    }

    else if (StopPressed())
    {
        NextState(state,StateRestart);
        ContactorOpen(DIG_AUX_CONT);
    }
    else if (IsTimerFinished(charge_timer))
    { /* If still drawing >1Amp after 10 seconds > fault */
        if (oldmaxIa > (1.0*256))                       /* Ipk > 1.0A  */
        {
            fault |= DC_CHARGE_FAULT;
            NextState(state,StateFault);
            ContactorOpen(DIG_AUX_CONT);
        }
        else /* No fault, so keep charging */
        {
            StartTimer(charge_timer, 10000 mSEC, NULL);
```

```
        }
    }

    /* Calc max(Ia) */
    if (Ia > maxIa) maxIa = Ia;
    else if (-Ia > maxIa) maxIa = -Ia;

    Ipeak = maxIa;
    /* Test for charged DC bus */
    if (IsTimerFinished(peak_timer) && (!IsolatePressed()))
    {
        if ((maxIa > (0.3*256))||(Vdc < Vdc_MIN+5))          /* Ipk > 0.3A  */
        { /* Not charged yet, wait further */
            StartTimer(peak_timer, 200 mSEC, NULL);
            oldmaxIa = maxIa;
            maxIa = 0;
        }
        else /* Charged! */
        {
            NextState(state,StateCloseWait);
            fault &= ~LOW_DC_VOLTS; /* Clear fault flag */
            tested_faults |= LOW_DC_VOLTS; /* Enable low Vdc fault checking */
        }
    }
    if (IsLastState(state))
    {
        FreeTimer(peak_timer);
        FreeTimer(charge_timer);
    }
} /* End StateCharge */


/*****************************************************************************/
/*  NAME:        StateCloseWait()                                            */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: Closes the main contactor and waits for it to close         */
/*  NOTES:       Once closed, state becomes StateSync or StateStop           */
/*  HISTORY:     10/09/2001  MN  Isolated State accounted for                */
/*****************************************************************************/
void StateCloseWait(void)
{
    static timerNumber cont_timer;      /* Contactor closing delay          */
    if (IsFirstState(state))
    {
        DoneFirstState(state);
        ContactorClose(DIG_CONT);
        cont_timer = GetTimer();        /* The contactor closing delay      */
        StartTimer(cont_timer, 500 mSEC, NULL);
    }
    if ( StopPressed() || IsolatePressed() )
    {
        NextState(state,StateOpenWait);
    }
    else if (IsTimerFinished(cont_timer))
    {
        if (ContClosed())
```



```
        {
            ContactorOpen(DIG_AUX_CONT);
            if (op_mode.is_sync)        /* Next state depends on op. mode  */
            {
                NextState(state,StateSync);
            }
            else
            {
                NextState(state,StateStop);
                init_display(state);
            }
        }
        else
        {
            fault |= CONTACT_FAULT;
            ContactorOpen(DIG_CONT|DIG_AUX_CONT);
            old_fault = fault;
            NextState(state,StateFault);
            display_fault(old_fault);
            StartTimer(fault_timer, 10000 mSEC, NULL); /* Delay before retry */
        }
    }
    if (IsLastState(state))
    {
        FreeTimer(cont_timer);
    }
} /* End StateCloseWait */


/*****************************************************************************/
/*  NAME:        StateOpenWait()                                             */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: State used to wait for contactor to open                    */
/*  NOTES:       Once contactor is open, state becomes StateIsolated         */
/*  HISTORY:     XX/XX/2001  MN  XXX                                         */
/*****************************************************************************/
void StateOpenWait(void)
{
    static timerNumber cont_timer;  /* Contactor opening delay */
    if (IsFirstState(state))
    {
        DoneFirstState(state);
        ContactorOpen(DIG_CONT|DIG_AUX_CONT);
        cont_timer = GetTimer(); /* The contactor opening delay */
        StartTimer(cont_timer, 500 mSEC, NULL);
    }

    if (IsTimerFinished(cont_timer))
    {
        if (!ContClosed())
        {
            NextState(state,StateIsolated);
        }
        else
        {
            fault |= CONTACT_FAULT;
```

```
            old_fault = fault;
            NextState(state,StateFault);
            display_fault(old_fault);
            StartTimer(fault_timer, 10000 mSEC, NULL); /* Delay before retry */
        }
    }

    if (IsLastState(state))
    {
        FreeTimer(cont_timer);
    }
} /* End StateOpenWait */


/*************************************************************************/
/*  NAME:        StateSync()                                             */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Waits until synchronisation is achieved                 */
/*  NOTES:                                                               */
/*  HISTORY:     10/09/2001  MN  Adjusted to include "StateIsolated" state */
/*************************************************************************/
void StateSync(void)
{
    static timerNumber sync_timer;              /* Time to achieve sync  */
    if (IsFirstState(state))
    {
        DoneFirstState(state);
        sync_timer = GetTimer();
        StartTimer(sync_timer, 10000 mSEC, NULL); /* 10 sec to sync in     */
    }
    if ( IsolatePressed() )
    {
        ContactorOpen(DIG_CONT|DIG_AUX_CONT);
        NextState(state,StateIsolated);
    }
    else if (StopPressed())
    {
        NextState(state,StateRestart);
        PWM_Stop();
        ContactorOpen(DIG_CONT|DIG_AUX_CONT);
    }
    else if (op_mode.in_sync)
    {
        fault &= ~(LOST_SUPPLY|LOST_SYNC);      /* Clear these faults    */
        NextState(state,StateStop);
        init_display(state);
        tested_faults |= (LOST_SUPPLY|LOST_SYNC);
    }
    else if (IsTimerFinished(sync_timer))       /* Failed to sync        */
    {
        fault |= LOST_SYNC;
        PWM_Stop();
        ContactorOpen(DIG_CONT|DIG_AUX_CONT);
        old_fault = fault;
        NextState(state,StateFault);
        display_fault(old_fault);
```



```
    }
    if (IsLastState(state))
    {
        FreeTimer(sync_timer);
    }
} /* End StateSync */


/*************************************************************************/
/*  NAME:        StateStop()                                             */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: State for when the inverter isn't running               */
/*  NOTES:                                                               */
/*  HISTORY:     10/09/2001  MN  Isolated State accounted for            */
/*************************************************************************/
void StateStop(void)
{
    SetMag(0,0,0);
    SetMagDC(adc_Vdc+1);
    Vdc_sum = 0;

    if ( IsolatePressed() )
    {
        ContactorOpen(DIG_CONT|DIG_AUX_CONT);
        NextState(state,StateIsolated);
    }
    else if (StartPressed())
    {
        NextState(state,StateRampUp);
        PWM_Start();
        SetLamp(LAMP_ON,SET);
    }
} /* End StateStop */


/*************************************************************************/
/*  NAME:        StateRampUp()                                           */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: State for ramping the magnitude up to the reference     */
/*  NOTES:                                                               */
/*  HISTORY:     10/09/2001  MN  Isolated State accounted for            */
/*************************************************************************/
void StateRampUp(void)
{
    if ( IsolatePressed() )
    {
        ContactorOpen(DIG_CONT|DIG_AUX_CONT);
        NextState(state,StateIsolated);
    }
    else if ( StopPressed() )
    {
        NextState(state,StateRampDown);
    }
    else if ( op_mode.is_sync && op_mode.is_PCR && (!op_mode.is_mag) )
    {
        ramp_count++;
```

```
        if ( ( GetMagDC() < mag_dc_bak ) && (ramp_count >= 5) )
        {
            ramp_count = 0;
            SetMagDC(GetMagDC()+1);
        }
        else
        {
            ramp_count = 0;
            SetMagDC(mag_dc_bak);
            NextState(state,StateRun);
        }
    }
    else if ( (GetMagD()>mag_bak-10)&&(GetMagD()<mag_bak+10) ) /* Ramping finished */
    {
        NextState(state,StateRun);
    }
    else
    {
        signed int mag_targ;
        mag_targ = GetMagD();
        if (mag_targ < mag_bak) mag_targ++; \
        else if (mag_targ > mag_bak) mag_targ--; \
        SetMag(mag_targ,0,0);
    }
} /* End StateRampUp */


/*****************************************************************************/
/*  NAME:        StateRun()                                                  */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: State for whilst the inverter is running                    */
/*  NOTES:                                                                   */
/*  HISTORY:     10/09/2001  MN  Isolated State accounted for                */
/*****************************************************************************/
void StateRun(void)
{
    if ( IsolatePressed() )
    {
        ContactorOpen(DIG_CONT|DIG_AUX_CONT);
        NextState(state,StateIsolated);
    }
    else if (StopPressed())
    {
        NextState(state,StateRampDown);
    }
    else if ( !( op_mode.is_sync && op_mode.is_PCR && (!op_mode.is_mag) ) )
    {
        SetMag(mag_bak, 0, 0);
    }
    else
    {
        SetMagDC(mag_dc_bak);

        if(is_af_on)
        {
            SetMagAF(mag_af_bak);
```



```
        }
        else
        {
            SetMagAF(0);
        }
    }

} /* End StateRun */


/*****************************************************************************/
/*  NAME:        StateRampDown()                                             */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: State for ramping the magnitude down to zero before         */
/*               turning inverter off                                        */
/*  NOTES:                                                                   */
/*  HISTORY:     10/09/2001  MN  Isolated State accounted for                */
/*****************************************************************************/
void StateRampDown(void)
{
    if ( IsolatePressed() )
    {
        ContactorOpen(DIG_CONT|DIG_AUX_CONT);
        NextState(state,StateIsolated);
    }
    else if (StartPressed())
    {
        NextState(state,StateRampUp);
    }
    else if ( op_mode.is_sync && op_mode.is_PCR && (!op_mode.is_mag) )
    {
        PWM_Stop();
        NextState(state,StateStop);
        SetLamp(LAMP_ON,CLEAR);
        Vdc_sum = 0;
    }
    else if (GetMagD() == 0) /* Ramping finished */
    {
        PWM_Stop();
        NextState(state,StateStop);
        SetLamp(LAMP_ON,CLEAR);
    }
    else
    {
        signed int mag_targ;
        mag_targ = GetMagD();
        if (mag_targ < 0) mag_targ++; \
        else if (mag_targ > 0) mag_targ--; \
        SetMag(mag_targ,0,0);
    }
} /* End StateRampDown */


/*****************************************************************************/
/*  NAME:        StateFault()                                                */
/*  RETURNS:     void                                                        */
```

```
/*  DESCRIPTION: State for handling faults                                  */
/*  NOTES:                                                                  */
/*  HISTORY:     10/09/2001  MN  Isolated State accounted for               */
/*************************************************************************/
void StateFault(void)
{
    static unsigned int fault_state = 0;
    if (IsFirstState(state))
    {
        DoneFirstState(state);
        PWM_Stop();
        ContactorOpen(DIG_CONT|DIG_AUX_CONT);
        SetLamp(LAMP_ON|LAMP_READY,CLEAR);
        SetLamp(LAMP_FAULT,SET);
    }
    tested_faults &= ~LOW_DC_VOLTS; /* Open contactor, so low Vdc is likely */
    if ((check_faults(tested_faults) == NO_FAULT)&&(fault_state == 0))
    {
        puts("\nFaults cleared - press stop to continue\n");
        fault_state = 1;
    }

    if ((StopPressed())&&(fault_state == 1))
    {
        StartTimer(fault_timer, 1000 mSEC, NULL); /* Delay */
        fault_state = 2;
        puts("\nRestarting...\n");
    }
    if ((fault_state == 2)&&(IsTimerFinished(fault_timer)))
    {
        NextState(state,StateIsolated);
        fault_state = 0;
        SetLamp(LAMP_FAULT,CLEAR);
    }
    if ( IsolatePressed() )
    {
        ContactorOpen(DIG_CONT|DIG_AUX_CONT);
        NextState(state,StateIsolated);
        tested_faults = ALL_FAULTS&~(LOW_DC_VOLTS|LOST_SUPPLY|LOST_SYNC);
    }
} /* End State Fault */


/*************************************************************************/
/*  NAME:        keyboard()                                              */
/*  RETURNS:     integer - returns 1 if quit is chosen, 0 otherwise      */
/*  DESCRIPTION: Handles serial port input                               */
/*  NOTES:                                                               */
/*  HISTORY:     03/10/2001  MN  Active filter enable/disable added      */
/*               10/12/2001  MN  Soft start bypass added                 */
/*************************************************************************/
int keyboard(void)
{
    char c,i,j;
    int ret_val = 0;
```



```
    if (kbhit())
    {
        c = getc();
        switch(c)
        {
            case 'q':       /* Quit */
                PWM_Stop();
                ContactorOpen(DIG_CONT|DIG_AUX_CONT);
                DisableInts();
                ret_val = 1;
                puts("\n\n\tQuitting\n\n");
                break;
            case 'h':       /* Help */
                puts("\tq\tquit\n");
                puts("\tt\ttiming analysis\n");
                puts("\tf\tdisplay fault status\n");
                puts("\ts\tdisplay the scaling factors\n");
                puts("\t1\tincrease phase offset\n");
                puts("\t2\tdecrease phase offset\n");
                puts("\tc\treset grab system\n");
                puts("\td\tdisplay grabbed results\n");
                puts("\tg\tstart grab\n");
                puts("\t:\tBypass soft start\n");
                break;
            case 'a':
                if (!is_af_on) {
                    is_af_on = TRUE;
                    puts("\nActive Filter Enabled\n");
                }
                else {
                    is_af_on = FALSE;
                    puts("\nActive Filter Disabled\n");
                }
                break;
            case 'n':
                Ctrl_Type = (Ctrl_Type>=2)?0:Ctrl_Type+1;
                if (Ctrl_Type == NOTCH) {
                    puts("\nNotch Filter\n");
                }
                else if (Ctrl_Type == STATRF)
                {
                    puts("\nStatRF Filter\n");
                }
                else if (Ctrl_Type == SRF)
                {
                    puts("\nSRF Filter\n");
                }
                else
                {
                    puts("\nNo Filter\n");
                }
                break;
            case 't':       /* Interrupt timing analysis */
                puts("\n\nStarting timing analysis\n\n");
                while(1)
                {
```

```
            Timing_R2_off();
            Timing_R2_on();
        }
        break;
    case 's':       /* Scaling Factor Constants Display */
        puts("\nScaling factors\n\ts1 = ");
        putu(s1);
        puts("\n\ts2 = ");
        putu(s2);
        puts("\n\ts3 = ");
        putu(s3);
        puts("\n\ts4 = ");
        putu(s4);
        puts("\n\ts5 = ");
        putu(s5);
        puts("\n\ts6 = ");
        putu(s6);
        puts("\n\tv_scale_int = ");
        putu(v_scale_int);
        puts("\n\ti_scale_int = ");
        putu(i_scale_int);
        puts("\n\ti_max_err = ");
        putu(i_max_err);
        puts("\n\tv_max_err = ");
        putu(v_max_err);
        puts("\n\n");
        break;
    case 'f':       /* Display current faults */
        if (fault==NO_FAULT)
        {
            puts("\nNo faults\nGate state:");
            /*putxx(get_fault());*/
            puts("\n");
        }
        else
        {
            display_fault(fault);
        }
        break;
    case '1':
        /*IncrPhase(16);*/
        Kp_dc++;
        puts("\nKp_dc = "); putd(Kp_dc); puts("\n");
        break;
    case '2':
        /*IncrPhase(-16);*/
        Kp_dc--;
        puts("\nKp_dc = "); putd(Kp_dc); puts("\n");
        break;
    case '!':
        /*IncrPhase(256);*/
        Ki_dc++;
        puts("\nKi_dc = "); putd(Ki_dc); puts("\n");
        break;
    case '@':
        /*IncrPhase(-256);*/
```



```
        Ki_dc--;
        puts("\nKi_dc = "); putd(Ki_dc); puts("\n");
        break;
    case 'c': /* Clear grab code */
        g_idx = 0;
        g_con = GRAB_READY;
        puts("Grab cleared\n");
        break;
    case 'g': /* Start grab now */
        puts("Grab started\n");
        g_idx = 0;
        g_con = GRAB_GO;
        break;
    case 'd': /* Display grabbed data */
        puts("\n");
        for (i=0; i<G_COUNT; i++)
        {
            putd(i);
            for (j=0; j<G_SIZE; j++)
            {
                putc('\t');
                putd(g[i][j]);
            }
            putc('\n');
        }
        break;
    case ':': /* Bypass contactor soft start */
        if (state.f == &StateCharge)
        {
            NextState(state,StateCloseWait);
        }
        break;
    case 'b': /* Create a current reference step */
        ref_double = TRUE;
        step_flag = 1;
        break;
    case 'v': /* Create a current reference step */
        ref_double = FALSE;
        step_flag = 1;
        break;
    }
  }
  return ret_val;

} /* End keyboard */


/*****************************************************************************/
/*  NAME:        init_display()                                              */
/*  RETURNS:     void                                                        */
/*  DESCRIPTION: Sets up heading for variable display and resets rotating    */
/*               variable display counter.                                   */
/*  NOTES:                                                                   */
/*  HISTORY:     XX/XX/2001  MN  XXX                                         */
/*****************************************************************************/
void init_display(StateType state)
```

```
{
    if (state.f == &StateInit)          puts("Init    ");
    else if (state.f == &StateIsolated)
        puts("\n\nState   ZX State   freq    AF      Ref     Vdc\n");
    else if (state.f == &StateCharge)
        puts("\n\nState   ZX State   freq    AF      Ia Max  Vdc\n");
    else
        puts("\n\nState   ZX State   freq    AF      Ref     Vdc\n");

} /* End init_display */

/*************************************************************************/
/*  NAME:        display_fault()                                         */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Displays the type of current fault to the serial port   */
/*  NOTES:                                                               */
/*  HISTORY:     XX/XX/2001  MN  XXX                                     */
/*************************************************************************/
void display_fault(unsigned int fault)
{
    if (fault!=NO_FAULT)
    {
        puts("\nFault detected :\n");
        if (fault&PDP_FAULT)
        {
            puts("\tPDPINT Fault: ");
            puts("\n");
        }
        if (fault&LC_VDC1         ) puts("\tDC over voltage fault\n");
        if (fault&LC_I_MOD        ) puts("\tOver current fault   \n");
        if (fault&LOST_SYNC       ) puts("\tLoss of sync fault   \n");
        if (fault&LOST_SUPPLY     ) puts("\tLoss of supply       \n");
        if (fault&LOW_DC_VOLTS    ) puts("\tLow DC volts         \n");
        if (fault&HIGH_DC_VOLTS   ) puts("\tHigh DC volts        \n");
        if (fault&MISC_FAULT      ) puts("\tMisc fault           \n");
        if (fault&SCALE_ERROR     ) puts("\tScale error          \n");
        if (fault&DC_CHARGE_FAULT ) puts("\tDC bus charge fault  \n");
        if (fault&CONTACT_FAULT   ) puts("\tContactor fault      \n");
    }
} /* End display_fault */

/*************************************************************************/
/*  NAME:        display_state()                                         */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Function used to display the text for the current state.*/
/*  NOTES:                                                               */
/*  HISTORY:     XX/XX/2001  MN  XXX                                     */
/*************************************************************************/
void display_state(StateType state)
{
    if (state.f == &StateInit)          puts("Init    ");
    else if (state.f == &StateIsolated) puts("Isolated");
    else if (state.f == &StateRestart)  puts("Restart ");
```



```
    else if (state.f == &StateCharge)    puts("CloseCh ");
    else if (state.f == &StateCloseWait) puts("CloseWt ");
    else if (state.f == &StateOpenWait)  puts("OpenWait");
    else if (state.f == &StateSync)      puts("Sync    ");
    else if (state.f == &StateStop)      puts("Stop    ");
    else if (state.f == &StateRampUp)    puts("RampUp  ");
    else if (state.f == &StateRun)       puts("Run     ");
    else if (state.f == &StateRampDown)  puts("RampDown");
    else if (state.f == &StateFault)     {puts("Fault:"); putu(fault);}
    else
    {
        putxx((unsigned int)state.f);
        putc('?');
    }
} /* End Display_state */

/*************************************************************************/
/*  NAME:        display_ZXstate()                                       */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Function to display text for current zero crossing state*/
/*  NOTES:                                                               */
/*  HISTORY:     XX/XX/2001  MN  XXX                                     */
/*************************************************************************/
void display_ZXstate(unsigned int zx_state)
{
    if      (zx_state == ZX_LOST ) puts("ZX Lost ");
    else if (zx_state == ZX_EST  ) puts("ZX Est. ");
    else if (zx_state == ZX_SYNC ) puts("ZX Sync.");
    else if (zx_state == ZX_PHASE) puts("ZX Phase");
    else if (zx_state == ZX_FREQ ) puts("ZX Freq ");
    else if (zx_state == ZX_LOCK ) puts("ZX Lock ");
    else if (zx_state == ZX_MISC ) puts("ZX Misc ");
    else
    {
        putxx(zx_state);
        putc('?');
    }
} /* End display_ZXstate */

/*************************************************************************/
/*  NAME:        display_CTRLstate()                                     */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Function to display text for current zero crossing state*/
/*  NOTES:                                                               */
/*  HISTORY:     XX/XX/2001  MN  XXX                                     */
/*************************************************************************/
void display_CTRLstate(unsigned int ctrl_type)
{
    if      (ctrl_type == NOTCH  ) puts("Notch   ");
    else if (ctrl_type == STATRF ) puts("StatRF  ");
    else if (ctrl_type == SRF    ) puts("SRF     ");
    else if (ctrl_type == NO_CTRL) puts("None    ");
    else
    {
```

```
        putu(ctrl_type);
        putc('?');
    }
} /* End display_CTRLstate */


/*************************************************************************/
/*  NAME:        get_variables()                                         */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Asks operator for various parameters before starting up */
/*  NOTES:                                                               */
/*  HISTORY:     XX/XX/2001  MN  XXX                                     */
/*************************************************************************/
void get_variables(void)
{
    char str[10];
    char is_PCR, is_sync, is_mag;

    sw_freq = 5000;             /* Switching frequency is currently fixed */

    puts("\nEnter parameters\n");

    dead_band_comp = MAX_DEADBAND + 1;
    while (dead_band_comp>MAX_DEADBAND)
    {
        puts("Enter deadband compensation in nanoseconds :");
        safe_gets(str,10);
        dead_band_comp = atoi(str);
    }
    is_PCR = 7;
    while ((is_PCR!=TRUE)&&(is_PCR!=FALSE))
    {
        puts("Run as PCR (p) or VSI (v)?\n");
        switch(getc())
        {
            case 'p' : is_PCR = TRUE; puts("Running as PCR\n"); break;
            case 'v' : is_PCR = FALSE; puts("Running as VSI\n"); break;
        }
    }
    is_sync = 7;
    while ((is_sync!=TRUE)&&(is_sync!=FALSE))
    {
        puts("Synchronize to external voltage signal? (y/n)\n");
        switch(getc())
        {
            case 'y' : is_sync = TRUE; puts("Synchronizing\n"); break;
            case 'n' : is_sync = FALSE; puts("Running free\n"); break;
        }
    }
    is_mag = 7;
    if (is_PCR&&is_sync)
    {
        while ((is_mag!=TRUE)&&(is_mag!=FALSE))
        {
            puts("Current control (c) or Vdc control (v)?\n");
            switch(getc())
```



```
            {
                case 'c' : is_mag = TRUE; puts("Using current control\n"); break;
                case 'v' : is_mag = FALSE; puts("Using Vdc control\n"); break;
            }
        }
    }
    else if (is_PCR)
    {
        is_mag = TRUE;
        puts("Using current control\n");
    }
    else if (is_sync)
    {
        while ((is_mag!=TRUE)&&(is_mag!=FALSE))
        {
            puts("Voltage magnitude control (v) or match external signal (m)?\n");
            switch(getc())
            {
                case 'v' : is_mag = TRUE; puts("Using voltage control\n"); break;
                case 'm' : is_mag = FALSE; puts("Matching signal\n"); break;
            }
        }
    }
    else
    {
        is_mag = TRUE;
        puts("Using voltage control\n");
    }
    op_mode.is_mag = is_mag;
    op_mode.is_sync = is_sync;
    op_mode.is_PCR = is_PCR;
} /* End get_variables */


/*************************************************************************/
/*  NAME:        set_variables()                                         */
/*  RETURNS:     void                                                    */
/*  DESCRIPTION: Sets the relevant parameters using default values       */
/*  NOTES:                                                               */
/*  HISTORY:     XX/XX/2001  MN  XXX                                     */
/*************************************************************************/
void set_variables(void)
{
    puts("\nAutostarting with default values\n");

    sw_freq = 5000;
    dead_band_comp = DEADBAND;
    switch (get_dips()&(DIP_PCR|DIP_SYNC|DIP_MAG))
    {
        case 0:                     /* DIP(0)=OFF  DIP(1)=OFF  DIP(2)=OFF  */
        case 1:                     /* DIP(0)=ON   DIP(1)=OFF  DIP(2)=OFF  */
        case 2:                     /* DIP(0)=OFF  DIP(1)=ON   DIP(2)=OFF  */
            op_mode.is_mag  = 0;
            op_mode.is_sync = 1;
            op_mode.is_PCR  = 0;
            puts("Running as VSI matching external voltage signal\n");
```

```
        op_mode.is_mag  = 1;
        puts("Not implemented at present. Control your own mag\n");
        break;
    case 3:                     /* DIP(0)=ON   DIP(1)=ON   DIP(2)=OFF  */
        op_mode.is_mag  = 0;
        op_mode.is_sync = 1;
        op_mode.is_PCR  = 1;
        puts("Running as PCR rectifier with Vdc control\n");
        break;
    case 4:                     /* DIP(0)=OFF  DIP(1)=OFF  DIP(2)=ON   */
        op_mode.is_mag  = 1;
        op_mode.is_sync = 0;
        op_mode.is_PCR  = 0;
        puts("Running as VSI with magnitude and frequency control\n");
        break;
    case 5:                     /* DIP(0)=ON   DIP(1)=OFF  DIP(2)=ON   */
        op_mode.is_mag  = 1;
        op_mode.is_sync = 0;
        op_mode.is_PCR  = 1;
        puts("Running as PCR with magnitude and frequency control\n");
        break;
    case 6:                     /* DIP(0)=OFF  DIP(1)=ON   DIP(2)=ON   */
        op_mode.is_mag  = 1;
        op_mode.is_sync = 1;
        op_mode.is_PCR  = 0;
        puts("Running as VSI synced to external voltage\n");
        break;
    case 7:                     /* DIP(0)=ON   DIP(1)=ON   DIP(2)=ON   */
        op_mode.is_mag  = 1;
        op_mode.is_sync = 1;
        op_mode.is_PCR  = 1;
        puts("Running as PCR rectifier with current magnitude control\n");
        break;
    }
} /* End set_variables */
```


E.3 filters.asm: Assembler Digital Filter Code

```
;***************************************************************************
;  Application:     Assembly IIR filter functions for use with C code
;  Developed By:    Monash University
;  Author:          M.Newman
;  File:            filters.asm
;  History:
;       07/08/2001 MN - Initial Creation
;       10/08/2001 MN - IIR_1stx2 added
;       10/08/2001 MN - IIR_2nd and IIR_2ndx2 added
;       24/09/2001 MN - 2nd order Delta now working OK, other parts deleted.
;       13/11/2001 MN - StatRF Cross coupled delta filter added
;       15/11/2001 MN - IIR_1stx2 added
;
;***************************************************************************
;
; 1. void delta_2nd(signed int* arg1)    arg1 --> [ y   x   x2  x1  x0  * ]
;
;***************************************************************************
;
; Typical calling process   r = IIR_Xxx(arg1,..,argx);
;   SACL    * ,AR1      <- arp always equals 1 entering into a function
;   BLKD    _arg2+0,*+
;   BLKD    _arg1+0,*+  <- arguments pushed on stack in reverse order
;   CALL    _IIR
;   SBRK    2
;   LDPK    _r
;   SACL    _r          <- result returned in low acc.
;
;***************************************************************************
                                ;
    .text                       ; For macd one coefficient table must be
                                ; defined in program space (ie. Tables A & B)
                                ;
B_1st                           ;
    .word   -32747,    32747    ; B_1st = [ b1 b0 ] HP Z: 10kHz/xHz/2^15
A_1st                           ;
    .word    32727              ; A_1st = [ -a1 ]   HP Z: 0kHz/xHz/2^15
                                ;
                                ;
B_2nd                           ;
    .word    16363, 517,  16534 ; B_1st=[b0 b1 b2] NB D: 10kHz/50Hz/2Hz/2^14/5
A_2nd                           ;
    .word    -16534, -1832      ; A_2nd=[-a2 -a1 ] NB D: 10kHz/50Hz/2Hz/2^14/5
                                ;
                                ;
B_2nd_11                        ;
    .word    16374, 1186, 16915 ; B_1st=[b0 b1 b2] StatRF D: 10kHz/50Hz/2Hz/2^14/5
A_2nd_11                        ;
    .word    -16560, -1833      ; A_2nd=[-a2 -a1]  StatRF D: 10kHz/50Hz/2Hz/2^14/5
B_2nd_22                        ;
    .word    16374, 1163, 16152 ; B_1st=[b0 b1 b2] StatRF D: 10kHz/50Hz/2Hz/2^14/5
A_2nd_22                        ;
    .word    -16560, -1833      ; A_2nd=[-a2 -a1] StatRF D: 10kHz/50Hz/2Hz/2^14/5
B_2nd_12                        ;
    .word    0, 24,  764        ; B_1st=[b0 b1 b2] StatRF D: 10kHz/50Hz/2Hz/2^14/5
```

```
A_2nd_12                        ;
    .word    -16560, -1833      ; A_2nd=[-a2 -a1] StatRF D: 10kHz/50Hz/2Hz/2^14/5
                                ;
                                ;
    .globl  _IIR_1stx2          ; Double first order shift based IIR filter
    .globl  _Delta_2nd          ; Single second order Delta Filter
    .globl  _Delta_2nd_11       ; Single second order Delta Filter for term 11
    .globl  _Delta_2nd_22       ; Single second order Delta Filter for term 22
    .globl  _Delta_2nd_12       ; Single second order Delta Filter for term 12
    .globl  _Delta_2nd_x4       ; 4 unit cross coupled Delta Filters
                                ;
                                ;
;*******************************************************************************
                                ;
_IIR_1stx2:                     ; Note: arp = 1 from calling process
                                ;
                                ;**** Setup Input Args and Other Params. ****
    SAR   AR1, *-               ;*ar1 =  ar1, ar1-- :Put ar1 onto stack
    LAR   AR2, *+, AR2          ; ar2 = *ar1, ar1++ :Load arg1 value into ar2
    ADRK  #6                    ; Point arg2 to the end of the table -1
    SPM   3                     ; Set PREG Shift mode to right shift by 6
                                ;
                                ;****          Perform Filters          ****
                                ; 1st order Filter number  - 1 -
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MACD  B_1st+0, *-           ; Positive pma inc. & negative dma increment
    APAC                        ; Add result to Accumulator
    MAR   *+                    ; Point ar2 to the dma address of yd0
    SACH  *-, 7                 ; Put final result back into integer array
                                ;
                                ; 1st order Filter number  - 2 -
    MAR   *-                    ; Point ar2 to the dma address of yq0
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MACD  B_1st+0, *-           ; Positive pma inc. & negative dma increment
    APAC                        ; Add result to Accumulator
    MAR   *+                    ; Point ar2 to the dma address of yq0
    SACH  *, 7, AR1             ; Put final result back into integer array
                                ;
                                ;**** Restore Setup then Ret. from Funct.****
    SPM   0                     ; Reset PREG Shift mode to no shift
    RET                         ; Return from function back to C code
                                ;
;*******************************************************************************
                                ;
_Delta_2nd:                     ; Note: arp = 1 from calling process
                                ;
                                ;****       Setup Input Arguments        ****
    SAR   AR1, *-               ;*ar1 =  ar1, ar1-- :Put ar1 onto stack
    LAR   AR2, *+, AR2          ; ar2 = *ar1, ar1++ :Load arg1 value into ar2
    ADRK  #1                    ; Point arg2 to the second element
                                ;
                                ;****      Perform Multiplications       ****
```



```
    SPM   3                     ; Set PREG Shift mode to right shift by 6
    LACC  *+, 8                 ; Load 'x' into 32bit Accum. with shift 8 left - 14bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_2nd+0, *+           ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MAC   B_2nd+0, *-           ; Positive pma inc. & negative dma increment
    APAC                        ; Add final result to Accumulator
    MAR   *-                    ; Point ar2 to the dma address of y
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7, AR1             ; Store result back into arg1[0]
    SPM   0                     ; Reset PREG Shift mode to no shift
    RET                         ; Return from function back to C code
                                ;
;*******************************************************************************
                                ;
_Delta_2nd_11:                  ; Note: arp = 1 from calling process
                                ;
                                ;****       Setup Input Arguments        ****
    SAR   AR1, *-               ;*ar1 =  ar1, ar1-- :Put ar1 onto stack
    LAR   AR2, *+, AR2          ; ar2 = *ar1, ar1++ :Load arg1 value into ar2
    ADRK  #1                    ; Point arg2 to the second element
                                ;
                                ;****      Perform Multiplications       ****
    SPM   3                     ; Set PREG Shift mode to right shift by 6
    LACC  *+, 8                 ; Load 'x' into 32bit Accum. with shift 8 left - 14bit
;    LACC  *+, 9                ; Load 'x' into 32bit Accum. with shift 9 left - 15bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_2nd_11+0, *+        ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MAC   B_2nd_11+0, *-        ; Positive pma inc. & negative dma increment
    APAC                        ; Add final result to Accumulator
    MAR   *-                    ; Point ar2 to the dma address of y
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7, AR1             ; Store result back into arg1[0]
    SPM   0                     ; Reset PREG Shift mode to no shift
    RET                         ; Return from function back to C code
                                ;
;*******************************************************************************
                                ;
_Delta_2nd_22:                  ; Note: arp = 1 from calling process
                                ;
                                ;****       Setup Input Arguments        ****
    SAR   AR1, *-               ;*ar1 =  ar1, ar1-- :Put ar1 onto stack
    LAR   AR2, *+, AR2          ; ar2 = *ar1, ar1++ :Load arg1 value into ar2
```

```
    ADRK  #1                    ; Point arg2 to the second element
                                ;
                                ;****      Perform Multiplications       ****
    SPM   3                     ; Set PREG Shift mode to right shift by 6
    LACC  *+, 8                 ; Load 'x' into 32bit Accum. with shift 8 left - 14bit
;    LACC  *+, 9                ; Load 'x' into 32bit Accum. with shift 9 left - 15bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_2nd_22+0, *+        ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MAC   B_2nd_22+0, *-        ; Positive pma inc. & negative dma increment
    APAC                        ; Add final result to Accumulator
    MAR   *-                    ; Point ar2 to the dma address of y
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7, AR1             ; Store result back into arg1[0]
    SPM   0                     ; Reset PREG Shift mode to no shift
    RET                         ; Return from function back to C code
                                ;
;*******************************************************************************
                                ;
_Delta_2nd_12:                  ; Note: arp = 1 from calling process
                                ;
                                ;****      Setup Input Arguments         ****
    SAR   AR1, *-               ;*ar1 =  ar1, ar1-- :Put ar1 onto stack
    LAR   AR2, *+, AR2          ; ar2 = *ar1, ar1++ :Load arg1 value into ar2
    ADRK  #1                    ; Point arg2 to the second element
                                ;
                                ;****      Perform Multiplications       ****
    SPM   3                     ; Set PREG Shift mode to right shift by 6
    LACC  *+, 8                 ; Load 'x' into 32bit Accum. with shift 8 left - 14bit
;    LACC  *+, 9                ; Load 'x' into 32bit Accum. with shift 9 left - 15bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_2nd_12+0, *+        ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MAC   B_2nd_12+0, *-        ; Positive pma inc. & negative dma increment
    APAC                        ; Add final result to Accumulator
    MAR   *-                    ; Point ar2 to the dma address of y
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7, AR1             ; Store result back into arg1[0]
    SPM   0                     ; Reset PREG Shift mode to no shift
    RET                         ; Return from function back to C code
                                ;
;*******************************************************************************
                                ;
_Delta_2nd_x4:                  ; Note: arp = 1 from calling process
```



```
                                ;
                                ;****      Setup Input Arguments         ****
    SAR   AR1, *-               ;*ar1 =  ar1, ar1-- :Put ar1 onto stack
    LAR   AR2, *+, AR2          ; ar2 = *ar1, ar1++ :Load arg1 value into ar2
    ADRK  #1                    ; Point arg2 to the second element
                                ;
                                ;****      Perform Multiplications       ****
    SPM   3                     ; Set PREG Shift mode to right shift by 6
                                ; Filter   --  1  --
    LACC  *+, 8                 ; Load 'x' into 32bit Accum. with shift 8 left - 14bit
;    LACC  *+, 9                ; Load 'x' into 32bit Accum. with shift 9 left - 15bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_2nd_11+0, *+        ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MAC   B_2nd_11+0, *-        ; Positive pma inc. & negative dma increment
    APAC                        ; Add final result to Accumulator
    MAR   *-                    ; Point ar2 to the dma address of y
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7                  ; Store result back into arg1[0]
    ADRK  #7                    ; Move arg2 to point to 2nd element in next filter
                                ; Filter   --  2  --
    LACC  *+, 8                 ; Load 'x' into 32bit Accum. with shift 8 left - 14bit
;    LACC  *+, 9                ; Load 'x' into 32bit Accum. with shift 9 left - 15bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_2nd_12+0, *+        ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MAC   B_2nd_12+0, *-        ; Positive pma inc. & negative dma increment
    APAC                        ; Add final result to Accumulator
    MAR   *-                    ; Point ar2 to the dma address of y
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7                  ; Store result back into arg1[0]
    ADRK  #7                    ; Move arg2 to point to 2nd element in next filter
                                ; Filter   --  3  --
    LACC  *+, 8                 ; Load 'x' into 32bit Accum. with shift 8 left - 14bit
;    LACC  *+, 9                ; Load 'x' into 32bit Accum. with shift 9 left - 15bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_2nd_12+0, *+        ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
```

```
    MAC   B_2nd_12+0, *-        ; Positive pma inc. & negative dma increment
    APAC                        ; Add final result to Accumulator
    MAR   *-                    ; Point ar2 to the dma address of y
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7                  ; Store result back into arg1[0]
    ADRK  #7                    ; Move arg2 to point to 2nd element in next filter
                                ; Filter   --  4  --
    LACC  *+, 8                 ; Load 'x' into 32bit Accum. with shift 8 left - 14bit
;    LACC  *+, 9                ; Load 'x' into 32bit Accum. with shift 9 left - 15bit
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #2-1                  ; Repeat the Multiply & Data Move X times
    MAC   A_2nd_22+0, *+        ; Positive pma inc. & positive dma increment
    APAC                        ; Add final result to Accumulator
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7                  ; Store result x0 for use in delta shifts
    LACL  #0                    ; Set 32bit Accumulator to 0
    MPY   #0                    ; Multiply TREG by 0 to clear PREG
    RPT   #3-1                  ; Repeat the Multiply & Data Move X times
    MAC   B_2nd_22+0, *-        ; Positive pma inc. & negative dma increment
    APAC                        ; Add final result to Accumulator
    MAR   *-                    ; Point ar2 to the dma address of y
    ROL                         ; Use this for 14bit scaling
    SACH  *, 7, AR1             ; Store result back into arg1[0]
    SPM   0                     ; Reset PREG Shift mode to no shift
    RET                         ; Return from function back to C code
                                ;
;*******************************************************************************
    .end
```

Bibliography

[1] M. J. Newman and D. G. Holmes, "An integrated approach for the protection of series injection inverters," *IEEE Trans. on Ind. Applicat.*, vol. 38, no. 3, pp. 679–687, 2002.

[2] M. J. Newman, D. N. Zmood, and D. G. Holmes, "Stationary frame harmonic reference generation for active filter systems," *IEEE Trans. on Ind. Applicat.*, vol. 38, no. 6, pp. 1591–1599, 2002.

[3] M. J. Newman and D. G. Holmes, "Delta operator digital filters for high performance inverter applications," *IEEE Trans. on Power Electronics*, vol. 18, no. 1, pp. 447–454, 2003.

[4] M. J. Newman, D. G. Holmes, J. G. Nielsen, and F. Blaabjerg, "A dynamic voltage restorer (dvr) with selective harmonic compensation at medium voltage level," *IEEE Trans. on Ind. Applicat.*, (In Review).

[5] P. C. Loh, M. J. Newman, D. N. Zmood, and D. G. Holmes, "Improved transient and steady state voltage regulation for single and three phase uninterruptible power supplies," *IEEE Trans. on Power Electronics*, 2003 (In Print).

[6] J. G. Nielsen, M. J. Newman, H. N. Nielsen, and F. Blaabjerg, "Control and testing of a dynamic voltage restorer at medium voltage level," *IEEE Trans. on Power Electronics*, (In Review).

[7] M. J. Newman and D. G. Holmes, "An integrated approach for the protection of series injection inverters," in *Conf. Rec. IEEE/IAS Annual Meeting*, pp. 871–878, 2001.

[8] M. J. Newman, D. N. Zmood, and D. G. Holmes, "Stationary frame harmonic reference generation for active filter systems," in *Conf. Rec. IEEE/APEC*, pp. 1054–1060, 2002.

[9] M. J. Newman and D. G. Holmes, "Delta operator digital filters for high performance inverter applications," in *Conf. Rec. IEEE/PESC Annual Meeting*, pp. 1407–1412, 2002.

[10] M. J. Newman and D. G. Holmes, "A universal custom power conditioner (ucpc) with selective harmonic voltage compensation," in *Conf. Rec IEEE/IECON*, pp. 1261–1266, 2002.

[11] M. J. Newman, D. G. Holmes, J. G. Nielsen, and F. Blaabjerg, "A dynamic voltage restorer (dvr) with selective harmonic compensation at medium voltage level," in *Conf. Rec. IEEE/IAS Annual Meeting*, 2003.

[12] P. C. Loh, M. J. Newman, D. N. Zmood, and D. G. Holmes, "Improved transient and steady state voltage regulation for single and three phase uninterruptible power supplies," in *Conf. Rec. IEEE/PESC Annual Meeting*, pp. 498–503, 2001.

[13] J. G. Nielsen, M. J. Newman, H. N. Nielsen, and F. Blaabjerg, "Control and testing of a dynamic voltage restorer at medium voltage level," in *Conf. Rec. IEEE/PESC Annual Meeting*, 2003.

[14] G. H. Bode, M. J. Newman, and D. G. Holmes, "Design and analysis of robust predictive current regulated algorithms," in *Conf. Rec. IEEE/PEDS*, 2003.

[15] E. Twining, M. J. Newman, P. C. Loh, and D. G. Holmes, "Voltage compensation in weak distribution networks using a d-statcom," in *Conf. Rec. IEEE/PEDS*, 2003.

[16] D. N. Zmood, *A Systematic Development of Improved Linear Regulation for Sinusoidal Power Converters, PhD Thesis*. Monash University, 2002.

[17] P. Mattavelli, "Synchronous-frame harmonic control for high-performance ac power supplies," *IEEE Trans. on Ind. Applicat.*, vol. 37, no. 3, pp. 864–872, 2001.

[18] R. Dugan, M. McGranaghan, and W. Beaty, *Electrical Power Systems Quality*. McGraw Hill, 1st ed., 1996.

[19] IEEE, "Ieee guide for service to equipment sensitive to momentary voltage disturbances," *IEEE Std 1250-1995*, 1995.

[20] IEEE, "Ieee recommended practice for evaluating electric power system compatibility with electronic process equipment," *IEEE Std 1346-1998*, 1998.

[21] IEEE, "Ieee recommended practices and requirements for harmonic control in electrical power systems," *IEEE Std 519-1992*, 1992.

[22] M. H. Bollen, *Understanding Power Quality Problems: Voltage Sags and Interruptions*. IEEE Press, 1st ed., 2000.

[23] J. de Abreu and A. Emanuel, "Induction motor thermal aging caused by voltage distortion and imbalance: loss of useful life and its estimated cost," *IEEE Trans. on Ind. Applicat.*, vol. 38, no. 1, pp. 12-20, 2002.

[24] P. Santos Valois, C. Vieira Tahan, N. Kagan, and H. Arango, "Voltage unbalance in low voltage distribution networks," in *in Conf. Rec. IEE/CIGRED Int'l Conf. on Elect. Dist.*, pp. 155-155, 2001.

[25] S. Salon, "Finite element analysis of electric machinery," *IEEE Computer Applications in Power*, vol. 3, no. 2, pp. 29–32, 1990.

[26] O.R.G., "Electricity distribution code," *Office of the Regulator-General, Victoria*, 2002.

[27] J. Collins, E. Randolph and A. Mansoor, "Effects of voltage sags on ac motor drives," in *Proceedings of the IEEE Annual Textile, Fiber and Film Industry Technical Conference*, pp. 1–7, 1997.

[28] IEEE, "Ieee standard for shunt power capacitors," *IEEE Std 18-1992*, 1992.

[29] J. Brozek, "The effects of harmonics on overcurrent protection devices," in *Conf. Rec. IEEE/IAS Annual Meeting*, pp. 1965–1967, 1990.

[30] L. Tolbert, H. Hollis, and P. J. Hale, "Survey of harmonics measurements in electrical distribution systems," in *Conf. Rec. IEEE/IAS Annual Meeting*, pp. 2333–2339, 1996.

[31] J. Gutierrez, J. Montano, A. Lopez, and M. Castilla, "Effects of harmonic distortion of the supply voltage on the optimum performance of a thyristor controlled reactor-type compensator," *IEE Proceedings-A-Science Measurement and Technology*, vol. 141, no. 1, pp. 15–19, 1994.

[32] W. Jewell, W. Miller, and T. Casey, "Audible telephone noise from small harmonic sources on single-phase distribution," in *Conf. Rec. IEEE/Harmonics and Quality of Power*, pp. 890–895, 1998.

[33] M. Kuussaari, "Statistical evaluation of telephone noise interference caused by ac power line harmonic currents," *IEEE Trans. on Power Delivery*, vol. 8, no. 2, pp. 524–530, 1993.

[34] IEC, "Electromagnetic compatibility (emc), part 3: Limits-section 3:limitation of voltage fluctuations and flicker in low-voltage supply systems for equipment with rated current less than or equal to 16a per phase," *IEC 61000-3-3:1995*, 1995.

[35] N. Hingorani, "High power electronics and flexible ac transmission system," *IEEE Power Engineering Review*, 1988.

[36] N. Hingorani, "Power electronics in electric utilities: role of power electronics in future power systems," *Proc. IEEE*, vol. 76, no. 4, pp. 481–482, 1988.

[37] N. Hingorani, "Introducing custom power," *IEEE Spectrum*, vol. 32, no. 6, pp. 41–48, 1995.

[38] N. Hingorani, "Future role of power electronics in power systems," in *Conf. Rec. Intl. Symp. on Pow. Semi. Devices and ICs(ISPSD)*, pp. 13–15, 1995.

[39] H. Akagi, "Trends in active power line conditioners," *IEEE Trans. on Power Electronics*, vol. 9, no. 3, pp. 263–268, 1994.

[40] H. Akagi, "New trends in active filters for power conditioning," *IEEE Trans. on Ind. Applicat.*, vol. 32, no. 6, pp. 1312–1322, 1996.

[41] H. Yunus and R. Bass, "Comparison of vsi and csi topologies for single-phase active power filters," in *Conf. Rec. IEEE/PESC Annual Meeting*, pp. 1892–1898, 1996.

[42] L. Benchaita, S. Saadate, and A. Salem nla, "Comparison of voltage source and current source shunt active filter by simulation and experimentation," *IEEE Trans. on Power Systems*, vol. 14, no. 2, pp. 642–647, 1999.

[43] D. G. Holmes and T. Lipo, *Pulse Width Modulation for Power Converters: Principles and Practice*. John Wiley and Sons, 1st ed., 2003.

[44] H. Sasaki and T. Machida, "A new method to eliminate ac harmonic currents by magnetic flux compensation-considerations on basic design," *IEEE Trans. on Power Apparatus and Sys.*, vol. PAS-90, no. 5, pp. 2009–2019, 1971.

[45] A. Ametani, "Generalised method of harmonic reduction in a.c.-d.c. convertors by harmonic current injection," *Proc. IEE*, vol. 119, no. 7, pp. 857–864, 1972.

[46] L. Gyugyi and E. Strycula, "Active ac power filters," in *Conf. Rec. IEEE/IAS Annual Meeting*, pp. 529–535, 1976.

[47] M. El-Habrouk, M. Darwish, and P. Mehta, "Active power filters: a review," *IEE Proc. Electric Power Applicat.*, vol. 147, no. 5, pp. 403–413, 2000.

[48] B. Singh, K. Al-Haddad, and A. Chandra, "Review of active filters for power quality improvement," *IEEE Trans. on Ind. Electronics*, vol. 46, no. 5, pp. 960–971, 1999.

[49] F. Z. Peng, H. Akagi, and A. Nabae, "A new approach to harmonic compensation in power systems," in *Conf. Rec. IEEE/IAS Annual Meeting*, pp. 874-880, 1988.

[50] F. Z. Peng, H. Akagi, and A. Nabae, "A new approach to harmonic compensation in power systems -- a combined system of shunt passive and series active filters," *IEEE Trans. on Ind. Applicat.*, vol. 26, no. 6, pp. 983–990, 1990.

[51] F. Z. Peng and D. J. Adams, "Harmonic sources and filtering approaches-series/parallel, active/passive, and their combined power filters," in *Conf. Rec. IEEE/IAS Annual Meeting*, pp. 448–455, 1999.

[52] S. Bhattacharya, D. M. Divan, and B. B. Banerjee, "Control and reduction of terminal voltage total harmonic distortion (thd) in a hybrid series active and parallel passive filter system," in *Conf. Rec. IEEE/PESC Annual Meeting*, pp. 779–786, 1993.

[53] J. W. Dixon, G. Venegas, and L. A. Moran, "A series active power filter based on a sinusoidal current-controlled voltage-source inverter," *IEEE Trans. on Ind. Electronics*, vol. 44, no. 5, pp. 612–619, 1997.

[54] S. Senini and P. Wolfs, "Systematic identification and review of hybrid active filter topologies," in *Conf. Rec. IEEE/PESC Annual Meeting*, pp. 394–399, 2002.

[55] G. Blajszczak, "Direct method for voltage distortion compensation in power networks by series converter filter," *IEE Proc. Electric Power Applicat.*, vol. 142, no. 5, pp. 308–312, 1995.

[56] M. Aredes and E. H. Watanabe, "New control algorithms for series and shunt three-phase four-wire active power filters," *IEEE Trans. on Power Delivery*, vol. 10, no. 3, pp. 1649–1656, 1995.

[57] N. H. Woodley, L. Morgan, and A. Sundaram, "Experience with an inverter-based dynamic voltage restorer," *IEEE Trans. on Power Delivery*, vol. 14, no. 3, pp. 1181–1186, 1999.

[58] J. G. Nielsen and F. Blaabjerg, "Comparison of system topologies for dynamic voltage restorers," in *Conf. Rec. IEEE/IAS Annual Meeting*, pp. 2397–2403, 2001.

[59] J. G. Nielsen, F. Blaabjerg, and N. Mohan, "Control strategies for dynamic voltage restorer compensating voltage sags with phase jump," in *Conf. Rec. IEEE/APEC*, pp. 1267–1273, 2001.

[60] J. G. Nielsen, *Design and Control of a Dynamic Voltage Restorer, PhD Thesis*. Aalborg University, 2002.

[61] M. D. Stump, G. J. Keane, and F. K. S. Leong, "Role of custom power products in enhancing power quality at industrial facilities," in *Conf. Rec. IEEE/EMPD*, pp. 507–517, 1998.

[62] M. Fang, A. I. Gardiner, A. MacDougall, and G. A. Mathieson, "A novel series dynamic voltage restorer for distribution systems," in *Conf. Rec. IEEE/PowerCon*, pp. 38–42, 1998.

[63] M. Vilathgamuwa, A. A. D. Ranjith Perera, S. S. Choi, and K. J. Tseng, "Control of energy optimized dynamic voltage restorer," in *Conf. Rec IEEE/IECON*, pp. 873–878, 1999.

[64] R. Dugan, M. McGranaghan, and W. Beaty, *Electrical Power Systems Quality*. McGraw Hill, 2nd ed., 2002.

[65] M. Bollen, P. Wang, and N. Jenkins, "Analysis and consequences of the phase jump associated with a voltage sag," in *Conf Rec. Power Sys. Comput. (PSCC)*, pp. 316–322, 1996.

[66] S. Walton, *Commercial DVR Discussion - Personal Communication*. Vectek Electronics, 25th June 2002.

[67] J. Svensson and A. Sannino, "Active filtering of supply voltage with series-connected voltage source converter," *EPE Journal*, vol. 12, no. 1, pp. 19–25, 2001.

[68] A. Sannino and J. Svensson, "Static series compensator for voltage sag mitigation supplying nonlinear loads," in *Conf. Rec. IEEE Power Eng. Soc. Winter Meeting*, pp. 1147–1152, 2002.

[69] M. Aredes, K. Heumann, and E. H. Watanabe, "An universal active power line conditioner," *IEEE Trans. on Power Delivery*, vol. 13, no. 2, pp. 545–551, 1998.

[70] S. Moran, "Line voltage regulator/conditioner for harmonic-sensitive load isolation," in *Conf. Rec. IEEE/IAS Annual Meeting*, pp. 947–951, 1989.

[71] L. M. Tolbert, F. Z. Peng, and T. G. Habetler, "Multilevel converter-based universal power conditioner," *IEEE Trans. on Ind. Applicat.*, vol. 36, no. 2, pp. 596–603, 2000.

[72] H. Akagi and H. Fujita, "New power line conditioner for harmonic compensation in power systems," *IEEE Trans. on Power Delivery*, vol. 10, no. 3, pp. 1570–1575, 1995.

[73] C. Zhan, M. Wong, Z. Wang, and Y. Hao, "Dsp control of power conditioner for improving power quality," in *Conf. Rec. IEEE Power Eng. Soc. Winter Meeting*, pp. 2556–2561, 2000.

[74] C. Zhan, V. Ramachandaramurthy, A. Arulampalam, C. Fitzer, M. Barnes, and N. Jenkins, "Universal custom power conditioner (ucpc) with integrated control," in *Conf. Rec. IEEE Power Eng. Soc. Winter Meeting*, pp. 1039–1044, 2001.

[75] R. Li, A. Johns, and M. Elkateb, "Control concept of unified power line conditioner," in *Conf. Rec. IEEE Power Eng. Soc. Winter Meeting*, pp. 2594 - 2599, 2000.

[76] H. Fujita and H. Akagi, "Unified power quality conditioner: the integration of series active filters and shunt active filters," in *Conf. Rec. IEEE/PESC Annual Meeting*, pp. 494–501, 1996.

[77] H. Fujita and H. Akagi, "Unified power quality conditioner: The integration of series- and shunt-active filters," *IEEE Trans. on Power Electronics*, vol. 13, no. 2, pp. 315–322, 1998.

[78] B. Singh, H. Chandra, K. Al-Haddad, and B. Singh, "Fuzzy control algorithm for universal active filter," in *Conf. Rec. IEEE/Harmonics and Quality of Power*, pp. 73–80, 1998.

[79] M. Vilathgamuwa, Y. Zhang, and S. Choi, "Modelling, analysis and control of unified power quality conditioner," in *Conf. Rec. IEEE/Harmonics and Quality of Power*, pp. 1035–1040, 1998.

[80] Y. Chen, X. Zha, J. Wang, H. Liu, J. Sun, and H. Tang, "Unified power quality conditioner (upqc): the theory, modeling and application," in *Conf. Rec. IEEE/PowerCon*, pp. 1329–1333, 2000.

[81] A. Elmitwally, M. Kandil, and M. Elkateb, "A fuzzy-controlled versatile system for harmonics, unbalance and voltage sag compensation," in *Conf. Rec. IEEE Power Eng. Soc. Summer Meeting*, pp. 1439–1444, 2000.

[82] M. Basu, S. Das, and G. Dubey, "Experimental investigation of performance of a single phase upqc for voltage sensitive and non-linear loads," in *Conf. Rec. IEEE/PEDS*, pp. 218–222, 2001.

[83] S. Chen and G. Joos, "Rating issues of unified power quality conditioners for load bus voltage control in distribution systems," in *Conf. Rec. IEEE Power Eng. Soc. Winter Meeting*, pp. 944–949, 2001.

[84] S. Chen and G. Joos, "A unified series-parallel deadbeat control technique for an active power quality conditioner with full digital implementation," in *Conf. Rec. IEEE/IAS Annual Meeting*, pp. 172–178, 2001.

[85] D. Graovac, V. Katic, A. Rufer, and J. Knecevic, "Unified power quality conditioner based on current source converter topology," in *Conf. Rec. European Pow. Elect. and Applicat. (EPE)*, 2001.

[86] A. Elnady and M. Salama, "New functionalities of the unified power quality conditioner," in *Conf. Rec. IEEE/Power Eng. Soc.Transmission and Dist.*, pp. 415–420, 2001.

[87] A. Elnady and M. Salama, "New functionalities of an adaptive unified power quality conditioner," in *Proceedings of Power Engineering Society Summer Meeting*, pp. 295–300, 2001.

[88] G. Jianjun, X. Dianguo, L. Hankui, and G. Maozhong, "Unified power quality conditioner (upqc): the principle, control and application," in *Conf. Rec. Pow. Conversion Conf. (PCC)*, pp. 80–85, 2002.

[89] A. Elmitwally, S. Abdelkader, and M. Elkateb, "Universal power quality manager with a new control scheme," *IEE Proc. Generation, Transmission and Distribution*, vol. 147, no. 3, pp. 183–189, 2000.

[90] S.-J. Jeon and G.-H. Cho, "A series-parallel compensated uninterruptible power supply with sinusoidal input current and sinusoidal output voltage," in *Conf. Rec. IEEE/PESC Annual Meeting*, pp. 297–303, 1997.

[91] F. Kamran and T. G. Habetler, "Combined deadbeat control of a series-parallel converter combination used as a universal power filter," *IEEE Trans. on Power Electronics*, vol. 13, no. 1, pp. 160–168, 1998.

[92] F. Kamran and T. G. Habetler, "Novel on-line ups with universal filtering capabilities," *IEEE Trans. on Power Electronics*, vol. 13, no. 3, pp. 410–418, 1998.

[93] S. A. O. d. Silva, P. F. Donoso-Garcia, and P. C. Cortizo, "Three-phase series-parallel compensated line-interactive ups system with sinusoidal input current and sinusoidal output voltage," in *Conf. Rec. IEEE/IAS Annual Meeting*, pp. 826–832, 1999.

[94] E. Silva, P. F. Donoso-Garcia, P. C. Cortizo, and P. Seixas, "A comparative analysis of control algorithms for three-phase line-interactive ups systems with series-parallel active power-line conditioning using srf method," in *Conf. Rec. IEEE/PESC Annual Meeting*, pp. 1023–1027, 2000.

[95] J. H. R. Enslin, "Unified approach to power quality mitigation," in *Conf. Rec. IEEE/ISIE*, pp. 8–20, 1998.

[96] S. S. Choi, B. H. Li, and D. M. Vilathgamuwa, "Dynamic voltage restoration with minimum energy injection," *IEEE Trans. on Power Systems*, vol. 15, no. 1, pp. 51–57, 2000.

[97] H. Akagi, Y. Kanazawa, and A. Nabae, "Instantaneous reactive power compensators comprising switching devices without energy storage components," *IEEE Trans. on Ind. Applicat.*, vol. IA-20, no. 3, pp. 625–630, 1984.

[98] S. Bhattacharya and D. Divan, "Design and implementation of a hybrid series active filter system," in *Conf. Rec. IEEE/PESC Annual Meeting*, pp. 189–195, 1995.

[99] S. Middlekauff and E. Collins, "System and customer impact: considerations for series custom power devices," *IEEE Trans. on Power Delivery*, vol. 13, no. 1, pp. 278–282, 1998.

[100] L. A. Moran, R. Oyarzun, I. Pastorini, J. W. Dixon, and R. Wallace, "Fault protection scheme for series active power filters," in *Conf. Rec. IEEE/PESC Annual Meeting*, pp. 489–493, 1996.

[101] L. A. Moran, I. Pastorini, J. W. Dixon, and R. Wallace, "Fault protection scheme for series active power filters," *IEEE Trans. on Power Electronics*, vol. 14, no. 5, pp. 928–938, 1999.

[102] M. H. J. Bollen, "Voltage, power and current ratings of series voltage controllers," in *Conf. Rec. IEEE Power Eng. Soc. Winter Meeting*, pp. 2910–2915, 2000.

[103] H. Akagi, "Control strategy and site selection of a shunt active filter damping of harmonic propagation in power distribution system," *IEEE Trans. on Power Delivery*, vol. 12, no. 1, pp. 354–363, 1997.

[104] G. Wan, *Internal Work Discussion - Personal Communication.* Australian Gas Light (AGL), Jan. 1999.

[105] P. Singh, J. Pacas, and C. Bhatia, "An improved unified power quality conditioner," in *Conf. Rec. European Pow. Elect. and Applicat. (EPE)*, 2001.

[106] C. Zhan, V. K. Ramachandaramurthy, A. Arulampalam, C. Fitzer, S. Kromlidis, M. Barnes, and N. Jenkins, "Dynamic voltage restorer based on voltage space vector pwm control," *IEEE Trans. on Ind. Applicat.*, vol. 36, no. 6, pp. 1855–1863, 2001.

[107] Y. Zhang, D. Vilathgamuwa, and S. Choi, "An experimental investigation of dynamic voltage restorer (dvr)," in *Conf. Rec. IEEE Power Eng. Soc. Winter Meeting*, pp. 2745–2750, 2000.

[108] D. Brod and D. Novotny, "Current control of vsi-pwm inverters," *IEEE Trans. on Ind. Applicat.*, vol. IA-21, no. 3, pp. 562–570, 1985.

[109] L. Malesani, P. Mattavelli, and S. Buso, "Robust dead-beat current control for pwm rectifiers and active filters," *IEEE Trans. on Ind. Applicat.*, vol. 35, no. 3, pp. 613–620, 1999.

[110] P. Mattavelli, "Closed-loop selective harmonic compensation for active filters," *IEEE Trans. on Ind. Applicat.*, vol. 37, no. 1, pp. 81–89, 2001.

[111] D. G. Holmes and D. A. Martin, "Implementation of a direct digital predictive current controller for single and three phase voltage source inverters," in *Conf. Rec. IEEE/IAS Annual Meeting*, pp. 906–913, 1996.

[112] D. N. Zmood, D. G. Holmes, and G. H. Bode, "Frequency-domain analysis of three-phase linear current regulators," *IEEE Trans. on Ind. Applicat.*, vol. 37, no. 2, pp. 601–610, 2001.

[113] D. N. Zmood and D. G. Holmes, "Stationary frame current regulation of pwm inverters with zero steady state error," *IEEE Trans. on Power Electronics*, vol. 18, no. 3, pp. 814–822, 2003.

[114] D. N. Zmood, *A Systematic Development of Improved Linear Regulation for Sinusoidal Power Converters, PhD Thesis.* Monash University, 2002.

[115] R. H. Park, "Two reaction theory of synchronous machines - generalized method of analysis - part i," *AIEE Trans.*, vol. 48, pp. 716–727, 1929.

[116] D. N. Zmood and D. G. Holmes, "Stationary frame current regulation of pwm inverters with zero steady state error," in *Conf. Rec. IEEE/PESC Annual Meeting*, pp. 1185–1190, 1999.

[117] Y. Sato, T. Ishizuka, K. Nezu, and T. Kataoka, "A new control strategy for voltage-type pwm rectifiers to realize zero steady-state control error in input current," *IEEE Trans. on Ind. Applicat.*, vol. 34, no. 3, pp. 480–486, 1998.

[118] P. A. Dahono, P. Ismutadi, Y. Sato, and T. Kataoka, "A control method for single-phase pwm inverters," in *Conf. Rec. IEEE/PEDS*, pp. 282–285, 2001.

[119] T. Kataoka, T. Ishizuka, K. Nezu, Y. Sato, and H. Yamaguchi, "Current control of voltage-type pwm rectifier introducing resonance-based controller," in *Conf. Rec. IEE/Power Electronics and Variable Speed Drives Conference*, pp. pp.519–524., 1996.

[120] P.-T. Cheng, S. Bhattacharya, and D. Divan, "Experimental verification of dominant harmonic active filter for high-power applications," *IEEE Trans. on Ind. Applicat.*, vol. 36, no. 2, pp. 567–577, 2000.

[121] P.-T. Cheng, S. Bhattacharya, and D. D. Divan, "Line harmonics reduction in high-power systems using square-wave inverters-based dominant harmonic active filter," *IEEE Trans. on Power Electronics*, vol. 14, no. 2, pp. 265–272, 1999.

[122] J. Svensson and R. Ottersten, "Shunt active filtering of vector-current controlled vsc at a moderate switching-frequency," *IEEE Trans. on Ind. Applicat.*, vol. 35, no. 5, pp. 1083–1090, 1999.

[123] M. Machmoum and N. Bruyant, "Dsp based control of shunt active power filters for global or selective harmonics compensation," in *Conf. Rec. IEEE/Harmonics and Quality of Power*, pp. 661–666, 2000.

[124] A. von Jouanne, P. Enjeti, and D. Lucas, "Dsp control of high-power ups systems feeding nonlinear loads," *IEEE Trans. on Ind. Electronics*, vol. 43, no. 1, pp. 121–125, 1996.

[125] Y. Y. Tzou, R. S. Ou, S. L. Jung, and M. Y. Chang, "High-performance programmable ac power source with low harmonic distortion using dsp-based repetitive control technique," *IEEE Trans. on Power Electronics*, vol. 12, no. 4, pp. 715–725, 1997.

[126] U. Borup, P. N. Enjeti, and F. Blaabjerg, "A new space-vector-based control method for ups systems powering nonlinear and unbalanced loads," *IEEE Trans. on Ind. Applicat.*, vol. 37, no. 6, pp. 1864–1870, 2001.

[127] A. Sannino and J. Svensson, "A series-connected voltage source converter for voltage sag mitigation using vector control and a filter compensation algorithm," in *Conference Record of the 2000 IEEE Industry Applications Conference. Thirty-Fifth IAS Annual Meeting and World Conference on Industrial Applications of Electrical Energy*, pp. 2476–2481, 2000.

[128] S. Srianthumrong and S. Sangwongwanich, "An active power filter with harmonic detection method based on recursive dft," in *Proc. of the International Conference on Harmonics and Quality of Power*, pp. 127–132, 1998.

[129] R. Elshatshat, M. Kazerani, and M. Salama, "Modular approach to active power-line harmonic filtering," in *Conf. Rec. IEEE/PESC Annual Meeting*, pp. 223–228, 1998.

[130] Y. Sato and T. Kataoka, "A current-type pwm rectifier with active damping function," *IEEE Trans. on Ind. Applicat.*, vol. 32, no. 3, pp. 533–541, 1996.

[131] P. A. Dahono, Y. R. Bahar, Y. Sato, and T. Kataoka, "Damping of transient oscillations on the output lc filter of pwm inverters by using a virtual resistor," in *Conf. Rec. IEEE/PEDS*, pp. 403–407, 2001.

[132] H. Awad, *Vector Control of Static Series Compensator for Mitigation of Voltage Dips, Licentiate Thesis.* Chalmers University of Technology, 2002.

[133] A. Ghosh and G. Ledwich, "Structures and control of a dynamic voltage regulator (dvr)," in *Conf. Rec. IEEE Power Eng. Soc. Winter Meeting*, pp. 1027–1032, 2001.

[134] A. Ghosh and G. Ledwich, "Compensation of distribution system voltage using dvr," *IEEE Trans. on Power Delivery*, vol. 17, no. 4, pp. 1030–1036, 2002.

[135] B. Li, S. Choi, and D. Vilathgamuwa, "A new control strategy for energy-saving dynamic voltage restoration," in *Conf. Rec. IEEE Power Eng. Soc. Summer Meeting*, pp. 1103–1108, 2000.

[136] K. Chan, A. Kara, and D. Westermann, "Integrated gate commutated thyristor based dynamic voltage restorer," in *Conf. Rec. IEEE/PowerCon*, pp. 635–638, 1998.

[137] P. Mattavelli and S. Fasolo, "A closed-loop selective harmonic compensation for active filters," in *Conf. Rec. IEEE/APEC*, pp. 399–405, 2000.

[138] R. H. Middleton and G. C. Goodwin, "Improved finite word length characteristics in digital control using delta operators," *IEEE Trans. on Automatic Control*, vol. 31, no. 11, pp. 1015–1021, 1986.

[139] E. C. Ifeachor and B. W. Jervis, *Digital Signal Processing: A Practical Approach.* Addison-Wesley Publishers Ltd., 1st ed., 1993.

[140] C. Rorabaugh, *Digital Filter Designer's Handbook: with C++ algorithms.* McGraw Hill, 2nd ed., 1997.

[141] C.-T. Chen, *Digital Signal Processing: Spectral Computation and Filter Design.* Oxford University Press, 2nd ed., 2001.

[142] T. Parks and C. Burrus, *Digital Filter Design.* John Wiley and Sons, 1st ed., 1987.

[143] A. Feuer and G. C. Goodwin, *Sampling in Digital Signal Processing and Control.* Birkhauser, 1st ed., 1996.

[144] D. B. Keogh and K. K. Pang, *Signal Processing.* Mi-tec Publishing, 1997.

[145] L. A. d. S. Ribeiro, C. B. Jacobina, A. M. N. Lima, and A. Oliveira, "Real-time estimation of the electrical parameters of an induction machine using sinusoidal pwm voltage waveforms," *IEEE Trans. on Ind. Applicat.*, vol. 36, no. 3, pp. 743–754, 2000.

[146] A. Khodabkhshian, V. J. Gosbell, and F. Coowar, "Discretization of power system transfer functions," *IEEE Trans. on Power Systems*, vol. 9, no. 1, pp. 255–261, 1994.

[147] G. C. Goodwin, R. H. Middleton, and H. V. Poor, "High-speed digital signal processing and control," *Proc. IEEE*, vol. 80, no. 2, pp. 240–259, 1992.

[148] M. B. Naumovic and M. R. Stojic, "Comparative study of finite word length effects in digital filter design via the shift and delta transforms," in *Conf. Rec. IEEE/TELSIKS*, pp. 141–144, 1999.

[149] J. Kauraniemi, T. I. Laakso, I. Hartimo, and S. J. Ovaska, "Delta operator realizations of direct-form iir filters," *IEEE Trans. on Circuits and Sys. II - Analog and Dig. Sig. Processing*, vol. 45, no. 1, pp. 41–52, 1998.

[150] X. Yuan, W. Merk, H. Stemmler, and J. Allmeling, "Stationary-frame generalized integrators for current control of active power filters with zero steady-state error for current harmonics of concern under unbalanced and distorted operating conditions," *IEEE Trans. on Ind. Applicat.*, vol. 38, no. 2, p. March/April, 2002.

[151] S. Bhattacharya and D. Divan, "Synchronous frame based controller implementation for a hybrid series active filter system," in *Conf. Rec. IEEE/IAS Annual Meeting*, pp. 2531–2540, 1995.

[152] P. K. Dash, A. K. Pradhan, G. Panda, and A. C. Liew, "Adaptive relay setting for flexible ac transmission systems (facts)," *IEEE Trans. on Power Delivery*, vol. 15, no. 1, pp. 38–43, 2000.

[153] C. Chaoying, C. Yong, D. Xuefei, and C. Liyi, "Study of protective relay's behaviors in unified power flow controller of facts," in *Conf. Rec. IEEE/APSCOM*, pp. 632–637, 1997.

[154] MathWorks, *Using Simulink - Version 3.* The MathWorks Inc., 3 ed., 1999.

[155] MathWorks, *Power System Blockset for use with Simulink - Users Guide Version 2.* The MathWorks Inc., 2000.

[156] P. McGoldrick, M. J. Newman, and D. G. Holmes, *CS-IIB Integrated Inverter - Technical Manual.* Monash University, Power Electronics Group, 1.0 ed., 2001.

[157] P. McGoldrick, A. McIver, and D. G. Holmes, *TMS320-F240 CS-MiniDSP - Technical Manual.* Monash University, Power Electronics Group, 1.5 ed., 2001.

[158] M. J. Newman, *3 Phase Active Filter Design and Construction, Undergraduate Thesis.* Monash University, 1999.

[159] Safco, *FELSIC 85 - Electrolytic Capacitor, Datasheet.* FELSIC Safco, 1996.

[160] Siemens, *BSM 50 GB 170 DN2 IGBT Datasheet.* Siemens, 1996.

[161] H. Akagi, A. Nabae, and S. Atoh, "Control strategy of active power filters using multiple voltage-source pwm converters," *IEEE Trans. on Ind. Applicat.*, vol. 22, no. 3, pp. 460–465, 1986.

[162] H. Akagi, Y. Tsukamoto, and A. Nabae, "Analysis and design of an active power filter using quad-series voltage source pwm converters," *IEEE Trans. on Ind. Applicat.*, vol. 26, no. 1, 1990.

[163] Y. Liang and C. O. Nwankpa, "Power-line conditioner based on flying-capacitor multilevel voltage-source converter with phase-shift spwm," *IEEE Trans. on Ind. Applicat.*, vol. 36, no. 4, pp. 965–971, 2000.

[164] IEC, "Electromagnetic compatibility (emc), part 3: Limits–section 2:limits for harmonic current emissions (equipment input current less than or equal to 16a per phase)," *IEC 61000-3-2:1995*, 1995.

[165] IEEE, "Ieee recommended practice for emergency and standby power systems for industrial and commercial applications," *IEEE Std 446-1995 (Orange Book)*, 1995.

[166] IEC, "Electromagnetic compatibility (emc) - part 2: Environment - section 1: Description of the environment - electromagnetic environment for low-frequency conducted disturbances and signalling in public power supply systems," *IEC 61000-2-1:1990*, 1990.

[167] IEC, "Electromagnetic compatibility (emc), part 3: Limits–section 5:limitation of voltage fluctuations and flicker in low-voltage supply systems for equipment with rated current greater than 16a," *IEC 61000-3-5:1994*, 1994.

[168] IEEE, "Ieee recommended practice for powering and grounding electronic equipment," *IEEE Std 1100-1999 (Emerald Book)*, 1999.

[169] SEMI, "Specification for semiconductor processing equipment voltage sag immunity," *SEMI F47-0200*, 2000.

[170] SEMI, "Test method for semiconductor processing equipment voltage sag immunity," *SEMI F42-0600*, 2000.

[171] D. S. Dorr, A. Mansoor, A. G. Morinec, and J. C. Worley, "Effects of power line voltage variations on different types of 400-w high-pressure sodium ballasts," *IEEE Trans. on Ind. Applicat.*, vol. 33, no. 2, pp. 472–476, 1997.

[172] L. M. Anderson and K. B. Bowes, "Effects of power-line disturbances on consumer electronic equipment," *IEEE Trans. on Power Delivery*, vol. 5, no. 2, pp. 1062–1065, 1990.

[173] I. T. Force, V. Wagner, J. Balda, D. Griffith, A. McEachern, T. Barnes, D. Hartmann, D. Phileggi, A. Emannuel, W. Horton, W. Reid, R. Ferraro, and W. Jewell, "Effects of harmonics on equipment," *IEEE Trans. on Power Delivery*, vol. 8, no. 2, pp. 672–680, 1993.

[174] IEEE, "Ieee recommended practice for emergency and standby power systems for industrial and commercial applications," *IEEE Std 446-1987*, 1987.

[175] J. G. Webster, *Wiley encyclopedia of electrical and electronics engineering.* John Wiley and Sons, Inc., 1999.

[176] A. de Lima and R. Ellis, "Discussion of 'harmonic analysis of industrial power systems'," *IEEE Trans. on Ind. Applicat.*, vol. 32, no. 5, pp. 1220–1221, 1996.