OPTIMIZATION OF CONTROLLED MARKOV CHAINS WITH APPLICATION TO DAM MANAGEMENT

A thesis submitted for the degree of

Doctor of Philosophy

By

Daniel McInnes

B.Sc.(Hons)

Supervisors: Dr. Boris Miller (Monash University)

Dr. Kais Hamza (Monash University)

School of Mathematical Sciences,

Monash University.

April 2015

© Copyright 2015

by

Daniel McInnes
B.Sc.(Hons)

Certificate of Originality

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of a university or other institute of higher learning, except where due acknowledgement is made in the text.

I also declare that the intellectual content of this thesis is the product of my own work, even though I may have received assistance from others on style, presentation and language expression.

Daniel McInnes

Copyright Notices

Under the Copyright Act 1968, this thesis must be used only under the normal conditions of scholarly fair dealing. In particular no results or conclusions should be extracted from it, nor should it be copied or closely paraphrased in whole or in part without the written consent of the author. Proper written acknowledgement should be made for any assistance obtained from this thesis.

I certify that I have made all reasonable efforts to secure copyright permissions for third-party content included in this thesis and have not knowingly added copyright content to my work without the owner's permission.

Abstract

Continuous-time Controlled Markov Chains are a useful model for many processes where it is necessary to alter the future behavior of the chain in a probabilistic way based on the current state. This project examines the techniques required to find the set of optimal controls for each state of the chain given a set of performance criteria, in both the unconstrained and constrained cases. We focus on the control of systems where there is a finite control horizon, the dynamics are non-stationary, and there are no apriori stability conditions. This will be demonstrated with a series of increasingly complex models which describe the management of a single dam or a system of arbitrarily connected dams. In these models a variety of controls are used: a price is imposed on water consumption in order to reduce overall water use, controlled transfers between dams are imposed to maintain system balance and controlled releases are allowed to reduce the chance of catastrophic flood. High performance numerical computing techniques are used for the solution of these problems and we demonstrate that implementable optimal control strategies can be computed.

Acknowledgements

I would first like to thank Dr. Boris Miller for his longsuffering and patient instruction and supervision over the previous three-and-a-half years. Boris has ensured that I understood the basics and then stretched me by prompting me to write papers from very early in the PhD. Subsequently I have been able to take the lead with some papers and the joint work between us has taught me many things, not only about the mathematics of stochastic control theory, but about the publication process, its highs and lows, and the need for perseverance. The multiple opportunities to present research at international conferences largely stems from Boris's early insistence that I write and publish, so I am very grateful to him for his foresight and for the funding to do all this.

I also must thank Dr. Kais Hamza, who is the second supervisor for this project. Kais is not an expert in optimal control theory but he is certainly an expert in stochastic processes and much of what I know I owe to his patient and thorough instruction during my undergraduate and honours years. I have had little occasion to bother Kais over the last few years but was always aware of where to go if I needed help with difficult questions and his advice at the regular progress seminars has lead to improvements in what we considered in our models, especially more complex inflow distributions.

The other main contributors to the academic process have been Professor Kate Smith-Miles and Professor Fima Klebaner, whose comments at the regular progress seminars have helped to ensure that the PhD project was on track, properly resourced and supported. Dr. Simon Clark, the post-graduate coordinator, must also be mentioned for his sound advice and help with regards to the general needs of the PhD student.

Many other people within the School of Mathematical Sciences have helped in various ways over the past three years. Gertrude Nayak, the school manager, has been a great help, especially with approving funding for the many conferences that I have attended and general advice on the administrative procedures to follow. Likewise, Rosie Frigo has handled the large file of forms and expense statements post trip and has always ensured that all was in order. Linda Mayer has organized the battery of progress seminars and other paperwork with courtesy and professionalism. I thank them all for their work.

Lastly, but most importantly, I thank my wife Yoko and my daughters Kaila and Rianah for their continuous support and for putting up with a perpetual student for so long. Without their support this project would not have been started, let alone completed.

Contents

Chapter 1

Introduction

> *Not only do dams represent some of the most impressive achievements of engineers over the centuries, but their vital role in supplying water to towns and cities, irrigating dry lands, providing a source of power and controlling floods is more than sufficient to rank dam-building among the most essential aspects of man's attempt to harness, control and improve his environment [53].*

The above quote from Norman Smith's, *A History of Dams*, provides a succinct overview of both the importance of dams and the various uses that they have been put to. Moreover, man's interest in building these structures dates from antiquity. One of the oldest known civil engineering structures discovered is the remains of a dam near Helwan, 20 miles south of Cairo. Known in Arabic as 'Sadd el-Kafara', or 'Dam of the Pagans', it was discovered in 1885 and is thought to date from between 2950 and 2750 B.C. [53]. Its apparent purpose was purely to hold water. Herodotus (in *Herodotus: The History,* Book 2, Chapter 99) [23] recounts a tale of a dam being built across the Nile by King Men, believed to have lived sometime between 3500 and 2850 B.C., to protect the city of Memphis from flood caused by the Nile breaking its banks. According to Smith, the damming of the Nile is most unlikely to have occurred, however, Herodotus no doubt saw a flood levee designed to divert flood flows. Smith also provides a great deal of information about the use of dams for irrigation in many places in ancient Mesopotamia, including Ur, Babylon and Assyria, and of course their construction by the Romans and others [53].

It suffices to say that the control of often unpredictable water resources has been one of the major considerations for the development of human civilization, especially with the building of large cities. While not a history thesis, the historical importance and use of dams along with their continued importance provides excellent motivation to consider the attributes that a model of a dam, or indeed dam system, should have. Ideally we would like to be able to build models of dams which provide water for human consumption, irrigation and power while reducing the risk of flooding. We would like to be able to achieve these aims under diverse and possibly extreme climatic conditions and do so in a way that provides us with the maximum flexibility in management of the resource.

From a mathematical point of view, this thesis deals with the development of tools based on the optimal control of continuous-time controllable Markov chains (CMC) and their application to the control of complex systems, in this case dams. The applicability of the method is wider, as evidenced by its use in internet congestion control [32], but in this thesis the application to dams will provide an interesting environment in which to display the utility of this method. The main question in this thesis is how to optimally control a large dam (or system of dams) with non-stationary inflows and outflows on a finite time horizon. This is approached via controlled Markov chains, as originally presented in [18, 32]. The research takes especially [32] as a base and further develops the tools for arbitrarily connected systems with more complex dynamics and considers the feasibility of constraints.

The body of the thesis will be based around six chapters, three of which present the content of published articles in conference proceedings [37, 38, 39]. The other chapters extend the work presented in these published papers.

The first chapter will provide a concise description of what continuous-time controlled Markov chains are and the methods used in their optimization. This will provide necessary background for all of the thesis, but will not deal with the feasibility of constraints. This will be left to Chapter 6, where this topic is discussed in some detail.

The following five chapters will fully present the methods and results of papers [37, 38, 39] and extend this published work in the direction of control under constraints. The actual control under constraints is not addressed because of the computational skill required to obtain results, but feasibility is addressed. The chapters present the material in the order in which it was completed, except for the details of [39], which deals with computational aspects and will be presented last. The other chapters follow a natural progression in terms of growing complexity and maturity of the models and results presented. Note that overflow of dams was not considered in the early models presented in Chapters 3, 4 and 7, but is dealt with in Chapters 5 and 6. In the early models we just let the dam overflow, being more concerned with using a price on water to maintain water in the dam. You could imagine that if the dam was full then it would be up to customer consumption, evaporation and overflow to deal with this, clearly not a reliable strategy. These models were built when Australia was going through severe drought and the threat of overflow was essentially zero. For that reason we focussed only on maintaining water in the dam via a price control on consumption. We added controlled release mechanisms to the models in Chapters 5 and 6 at about the time that extremely high water levels in the Wivenhoe Dam west of Brisbane, Australia, led to significant loss of life and widespread damage despite a realease program. These models are an attempt to incorporate a more flexible and proactive release regime.

The final chapter will be the conclusion, which will summarize the material and emphasize how this work has filled a gap in the knowledge of the optimal control of complex systems, such as dams.

1.1 Review of the current research

Markov chains have found application in many areas of science and mathematics since they model random processes in which the current state of the process depends only upon the previous state. This current research involves the use of continuous-time controlled Markov chains. These are Markov chains in which the transition probabilities for the transitions between states of the chain can be modified via a control to achieve certain performance criteria.

In general problems of this type are approached as Markov decision processes (MDP) in either discrete or continuous time. Discrete time models make obvious sense in the case of discrete time observations or where there are discrete decision epochs. Piunovskiy [47] provides many interesting examples of these processes on finite time horizons, infinite time horizons with total or discounted loss, average loss on an infinite time horizon, and control under constraints [45, 48]. The work under constraints is of particular relevance to this thesis, although studied here in the discrete time case. The types of problems approached in this way include control of epidemics [46, 61], inventory control [6], supply chain optimization [60], and many others. An interesting example of both discrete and continuous time Markov chains being used for decision support in hospital bed management can be found in [50]. Of course, the discrete-time case for water resource control has received significant attention, two examples of which are Sniedovich [54] and, Karbowski and Magierra [27]. Sniedovich used dynamic programming to solve a variance-constrained problem and Karbowski and Magierra used various techniques on both finite and infinite time horizons to solve a two criteria optimal reservoir management problem. Another example of a discrete time optimal control model for dams was studied by Ozelkan et.al. via linear-quadratic dynamic programming (LQ) under the assumption of stationarity of the inflow process and applied to an existing dam [44].

For continuous-time Markov decision processes (MDP) there is abundant literature available (see for example [7], [11] and [28]). Miller et.al state that these processes have found application in communications engineering, queuing systems and control of epidemics [34]. Of course, dams have not escaped such study and a range of papers have been written where the inflow process is a Wiener process or a compound Poisson process and the dam has either finite or infinite capacity (for examples see [2],[4],[8],[19] and [62].) It has been noted that a Wiener process is not very realistic for modeling inflows but simplifies some calculation [19]. Non-negative Levy processes as a general class have also been tried [3], but in general, compound Poisson processes are used.

The key point of commonality in most of the above examples is that they use very simple threshold control models. They use a long-run average criterion as the principal optimality criterion. Such models generally assume an infinite time horizon for control and stationary inflow data. However, in the case of dams, as with other natural systems that depend on variable inflows, the inflows are non-stationary, varying with seasonal rains. The long time horizon requires that inflows be greater than or equal to outflows or the system will run out of water in finite time, but in the short term usage may well exceed inflows. It is also clear that short term water availability is extremely important and that some account needs to be taken of this in the optimality conditions. Finally, the long-term average criterion does not consider the costs of transient states and the resources required for these transitions on a finite time interval [32].

As already stated, there is a large body of literature on MDP dealing with the case of optimization problems on an infinite time horizon. The research into the area of optimization of controlled Markov chains on a finite time horizon is more limited; as early as 1968 Bruce Miller [36] was researching Markov decision processes on a finite planning horizon. His paper considered a finite state continuous time controlled Markov chain where the returns are generated by maximizing the expected return on a finite time horizon. He was able to find necessary and sufficient conditions for optimality and to show that a piecewise constant policy is optimal in this particular control problem.

More recent work in this area has been done by Robert Elliott [17] and by Elliott et.al [18] on control with hidden Markov models (HMM). HMM deals with Markov chains which are hidden in a noisy observation process. Another example of control in partially observed jump processes is given by Ceci et al. [14]. For example, you may wish to find a signal sent via radio in a noisy channel. The signal itself can be modeled as a Markov chain but the observed signal must be somehow filtered to estimate this original signal. The main tool used is that of reference probability methods. This is a method which takes the noisy observation process, say $\{Y_k\}$ for $k \geq 0$, under the real world probability measure $\mathbb{P}$ on $(\Omega, \mathscr{F})$ and transforms this process via a change of measure to a sequence of i.i.d. random variables under the new ideal measure $\bar{\mathbb{P}}$. Under $\bar{\mathbb{P}}$ the estimation is relatively easy and the result can be transformed back to the real world problem via a reverse measure change.

The most recent research relevant to this project is being done by Boris Miller et.al [32, 33, 34, 35]. This research takes as a basis the works of Robert Elliott et.al [5, 18, 17] and considers problems of optimal control under constraints. Some specific examples of applications to queuing systems have already been solved [32] but this has not been applied to a system with relatively complex dynamics such as a large dam. This project fits into the development of the theory and modeling required to deal with the optimal control of such systems, including extending and building new methods to deal with multidimensional connected systems.

Chapter 2

Inhomogeneous continuous time controlled Markov chains and their optimization methods

In line with the general outline given in the introduction, the following gives a detailed account of how the optimal stochastic control problem is posed and the methods used in its solution.

2.1 Outline of mathematical and research methods

The mathematical methods involved in this project encompass many areas of pure and applied mathematics as well as probability. The problem is given in a stochastic setting, however, many problems can be reduced to deterministic equivalents by taking taking the expectations and considering the solution on average. Thus, a wide range of techniques will be employed.

In the first instance, it is necessary to state the problem in a form that can be solved via dynamic programming. The development of dynamic programming as a systematic optimization tool is largely due to Richard Bellman. The *principle of optimality* owes its name to Bellman and states in the discrete time case (see [10]) that if $\{u_0, ..., u_{N-1}\}$ is the optimal control law for the control problem, then if we start in state x at time i and want to minimize the cost to go from i to N, the control law $\{u_i, ..., u_{N-1}\}$ is optimal for the truncated problem. In continuous time this would mean that if $u(t), t \in [0, T]$ is the optimal control function, then $u(t), t \in [s, T]$ is the optimal control function for the truncated problem. It follows that if we have an optimal terminal state, then we can optimize the function of states and controls step by step back from this terminal state. This is the essence of one of the main dynamic programming techniques.

In this thesis we make extensive use of the martingale description of a continuous Markov chain, with a finite state space in $\mathbb{R}^{N+1}$. Assume that we have a continuous time jump process $\{X(t), t \in [0,T]\}$ with piecewise constant right continuous paths defined on a probability space $\{\Omega, \mathscr{F}, \mathbb{P}\}$. Then define the state space of this process to be the unit vectors in $\mathbb{R}^{N+1}$, e_i such that $X(t) = \{0, ..., 1, ..., 0\}$, with unity in the i^{th} position [18].

2.2 Inhomogeneous continuous time Markov chains

We begin by assuming that we have $N+1 \in \mathbb{Z}^+$ states, such that at any time $t \in [0,T]$, $T < \infty$, that state of the process takes its value from the set, $S = \{0, 1, 2, ..., N-1, N\}$. Further, there is an exact correspondence between the set S and $X(t)$, given that if the process is in state $i \in S$ at time t, then $X(t) = e_i$. For simplicity, we define the inhomogeneous continuous time Markov chain with respect to the set S, keeping in mind the correspondence with the process $X(t)$.

Definition 2.1. *(ICTMC). An inhomogeneous continuous time Markov chain is a tuple* $\mathcal{C} = (S, R)$ *where:* $S = \{0, 1, 2, ..., N\}$ *is a finite set of states, and* $R(t) = [R_{i,j}(t) \geq 0] \in \mathbb{R}^{(N+1)\times(N+1)}$ *is a time dependent rate matrix, where* $R_{i,j}(t)$ *is the rate of transitions from state* i *to* j, $i, j \in S$, *at time* $t \in [0, \infty)$*[22].*

The term "intensity" is used interchangably with "rate" throughout. We define the diagonal matrix $E(t) = diag[E_i(t)] \in \mathbb{R}^{(N+1)\times(N+1)}$, where $E_i(t) = \sum_{j \in S} R_{i,j}(t)$ for $i, j \in S$, $i \neq j$. This is the total exit rate out of state i at time t. For completeness we give the probability measures for waiting time in state i and the transitions from state i to j (Refer to [22] for all proofs of the following properties).

Let $\{Z(t)|t \geq 0\}$ be an inhomogeneous Poisson process with arrival rate $R(t)$. The probability of k arrivals in the interval $[t, t+\Delta t]$ is given by:

$$\mathbb{P}\left\{Z(t+\Delta t) - Z(t) = k\right\} = \frac{\left[\int_t^{t+\Delta t} R(s)\, ds\right]^k}{k!} e^{-\int_t^{t+\Delta t} R(s)\, ds},\ k = 0, 1, ...$$

It follows from this that if $k = 0$,

$$\mathbb{P}\left\{Z(t+\Delta t) - Z(t) = 0\right\} = e^{-\int_t^{t+\Delta t} R(s)\, ds} = e^{-\int_0^{\Delta t} R(t+s)\, ds}, \tag{2.1}$$

which is the probability of no arrivals. Now let $W_{i,j}(t)$ be a random variable representing the waiting time until the transition from state i to state j, with rate $R_{i,j}(t)$ at time t. Then the probability that the waiting time until the transition occurs before Δt is

$$\mathbb{P}\left\{W_{i,j}(t) \leq \Delta t\right\} = 1 - \Pr\left\{Z(t+\Delta t) - Z(t) = 0\right\} = 1 - e^{-\int_0^{\Delta t} R_{i,j}(t+s)\, ds}, \tag{2.2}$$

from equation (2.1). We now can state some fundamental transition probabilities for an ICTMC. The first is the probability that the waiting time in state i will be less than Δt, which is clearly one minus the probability of any exits out of state i in the interval Δt:

$$\mathbb{P}\left\{W_i(t) \leq \Delta t\right\} = 1 - e^{-\int_0^{\Delta t} E_i(t+s)\, ds}, \tag{2.3}$$

where $E_i(t)$ is the total exit rate out of state i at time t as defined above. The second is the probability that the chain will transition from state i to state j, $i \neq j$, at any time in the future after time t, with transition rate $R_{i,j}(t)$:

$$\mathbb{P}_{i,j}(t) = \int_0^{\infty} R_{i,j}(t+\tau) e^{-\int_0^{\tau} E_i(t+s)\, ds}\, d\tau. \tag{2.4}$$

The final one is the probability that the chain will transition from state i to state j, $i \neq j$, in the interval Δt after time t, with rate $R_{i,j}(t)$:

$$\mathbb{P}_{i,j}(t, \Delta t) = \int_0^{\Delta t} R_{i,j}(t+\tau) e^{-\int_0^{\tau} E_i(t+s)\, ds}\, d\tau. \tag{2.5}$$

So, ICTMCs are well defined processes, which are especially useful for modelling systems where the transition rates between states vary through time. Of course in this construction the Markov property holds as for continuous time Markov chains, that is $\mathbb{E}\left[X(t)|\mathscr{F}_{\tau}^{X}\right] = \mathbb{E}\left[X(t)|X(\tau)\right]$ for $\tau < t$, using the representation given by $X(t)$ [26].

2.3 Generator of the ICTMC

Here we use the representation of the ICTMC, $X(t)$, as described in the introduction to Section 2.2, and denote rate of transitions out of state i by the parameter $\alpha_i(t) = E_i(t)$; if the present state is $X(t) = \{0, ..., 1, ..., 0\}$, with unity in the i^{th} position at time t, the next state will be $X(t+h) = \{0, ..., 1, ..., 0\}$, with unity in the j^{th} position at time $t+h$ for $j \neq i$. This occurs with probability $p_{ij}(t+h) = \mathbb{P}_{i,j}(t,h)$ independently of the history of the process and of the time until the next jump. Transitions from i to i have zero probability [12].

Define $q_{ij}(t) = \alpha_i(t)p_{ij}(t)$ as the expected number of jumps from i to j per unit of time spent in i. The $(N+1) \times (N+1)$ matrix of these values gives the $Q(t)$ matrix of the process, with the exception that we define $q_{ii}(t) = -\sum_{j \neq i} q_{ij}(t)$, such that the rows sum to zero,

$$Q(t) = \begin{pmatrix} -\sum\limits_{j \neq 1} q_{1j}(t) & q_{12}(t) & ... & ... & q_{1n}(t) \\ q_{21}(t) & -\sum\limits_{j \neq 2} q_{2j}(t) & ... & ... & q_{2n}(t) \\ ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... \\ q_{n1}(t) & ... & ... & q_{nn-1}(t) & -\sum\limits_{j \neq n} q_{nj}(t) \end{pmatrix}.$$

2.4 Controlled ICTMCs

Given a ICTMC, we add a control parameter $u \in U$, where U is a compact set in a complete metric space, such that the rate of transition out of state i to another,

$\nu_i(t,u)$, now also depends on u. It follows that each element of the matrix Q now also depends on u.

Assumption 2.2. *We assume the following to be true of the entries of the matrix* $A(t,u) = Q(t,u)^T$ *([18] Chapter 12, [17]):*

1. $a_{ji}(t,u) \geq 0, \forall j \neq i$;
2. $a_{jj}(t,u) = -\sum_{i \neq j} a_{ij}(t,u)$;
3. $A(t,u)$ *is continuous on* $[0,T] \times U$; *and*
4. *For a given* $u \in U$ *and an initial probability distribution* $P(0)$ *of* X_0, *the probability column vector* $P(t) = (P_t^1, ..., P_t^n)$, $P_t^i = \mathbb{P}(X_t = e_i)$, *satisfies the forward Kolmogorov equation*

$$\frac{dP(t)}{dt} = A(t,u)P(t). \tag{2.6}$$

$A(t,u)$ is called a time dependent generator of the process $X(t)$ and so for different u we have a family of such generators.

Remark 2.3. *In Markov chain theory, this generator matrix conventionally has rows which sum to zero and is usually termed the* Q *matrix. In control theory the transpose of this matrix is used such that* $A = Q^T$, *and this is the notation used above and throughout the thesis.*

Remark 2.4. *While termed a controllable ICTMC, in general the resulting process is not Markovian. The controls at time* $t < T$ *may depend on the entire history of the process up to time* t.

The process $X(t)$ generates a family of right continuous sigma algebras $\mathscr{F}_t^X = \sigma\{X(s) : s \in [0,t]\}$. We assume that there exists a set of admissible controls, $\bar{U} = \{u(\cdot)\}$, which is the set of $\mathscr{F}_t^X$ predictable processes in U. So, let the jump times of the process $X(t)$ on $[0,t]$ be written as $\tau_k, k = 1, ..., N(t)$, where $N(t)$ is the total number of jumps on $[0,t]$, then the states from time zero until t, X_0^t are

$$X_0^t = \{(X(0),0),(X(\tau_1),\tau_1),...,(X(\tau_{N(t)}),\tau_{N(t)})\}.$$

Now, the assumption means that for $\tau_{N(t)} < t \leq \tau_{N(t)+1}$, $u(t) = u(t, X_0^t)$ is a measurable function of X_0^t and the current time t [18, 32]. This allows the probability measure $\mathbb{P}$ on $\{\Omega, \mathscr{F}\}$ to extend naturally to the controlled chain for each $u \in \bar{U}$, which is based on the theorem of Ionescu Tulcea (see [33]).

To derive the dynamic programming equation we use the semi-martingale representation of a controlled Markov chain, where $u(s)$ is $\mathscr{F}_t^X$ predictable, which has the form

$$X^u(t) = X(0) + \int_0^t A(s, u(s))X^u(s-)ds + \mathscr{M}^u(t) \tag{2.7}$$

where $X(0)$ is a random initial condition with a given distribution and $\mathscr{M}^u(t) := \{\mathscr{M}^1(t), ..., \mathscr{M}^n(t), ...\}$ is a square integrable $(\mathscr{F}_t^X, \mathbb{P}^u)$ martingale with quadratic variations

$$\begin{aligned} \langle \mathscr{M}^u \rangle(t) \quad = \quad & \int_0^t \mathrm{diag}(A(s, u(s))X^u(s-))ds \\ & -\int_0^t [A(s, u(s))(\mathrm{diag}(X^u(s-))) \\ & +(\mathrm{diag}(X^u(s-)))A^T(s, u(s))]ds. \end{aligned} \tag{2.8}$$

The derivation of this representation of an uncontrolled continuous-time Markov chain is given in [18] and depends on representations given in [59], and a result from Pliska [49](Section 3) which gives us that for each admissible control policy, a Markov process exists. It follows that for each admissible control policy, the above representation is valid (see especially Theorem 2 of [33] for a proof of this using the theorem of Ionescu Tulcea).

The method of defining the infinitesimal generator of the controlled Markov chain, $A(t, u(t))$, is of considerable importance in this model. For the models given in Chapters 5 and 6, the derivation of $A(t, u(t))$ is given in each chapter, since its derivation is more complex than in the models given in the preceding chapters. It is instructive, however, to explicitly prove the simpler form of the generator since it brings together the dynamics of the model and informs the proofs given in Chapters 5 and 6. Since the generators of the models in all other chapters are minor extensions of that used in Chapter 3, we now derive the generator used in the next chapter in detail.

2.5 The basic dam model

2.5.1 General definitions

We begin by assuming that we can discretize the level of water in a single large dam into $N+1 \in \mathbb{Z}^+$ levels. So at any time $t \in [0,T]$, $T < \infty$, we let the integer valued random variable, $L(t) \in \{0,1,2,...,N-1,N\}$, describe the state of the dam. At the same time we can represent the dam by a controlled jump Markov process with piecewise right-continuous paths, $X(t) \in \mathbb{R}^{N+1}$, on the probability space $\{\Omega, \mathscr{F}, \mathbb{P}\}$, with state space, $S = \{e_0, e_1, ..., e_{N-1}, e_N\}$, where each of the e_i, $i = 0,...,N$ is a unit vector with 1 in the i^{th} position and zeros elsewhere. So, we have the relation

$$I\{L(t) = i\} = I\{X(t) = e_i\}, \tag{2.9}$$

which will be used in the later proof of the generator.

We also assume that we can alter the probability of jumps in this process through a price on water and make the following assumptions about the control, $p(t, X(t))$.

Assumption 2.5. *Assume that the set of admissible controls, $\bar{P} = p(\cdot)$ is a set of $\mathscr{F}_t^X$-predictable controls taking values in $P = \{p \in [p_{min}, p_{max}]\}$.*

Remark 2.6. *If the history of the jump process from time 0 to t is denoted X_0^t, then assumption 2.5 ensures that our control, $p(t, X_0^t)$ is predictable with respect to t and X_0^t.*

2.5.2 Dynamics

We now consider the inflows and outflows of the simple dam model proposed in Chapter 3. Our first assumption is that there can be no outflows when the dam is at its lowest level, $L(t) = 0$, and no inflows will be counted when at the highest level, $L(t) = N$. Then we assume that the inflows to the dam can be modeled as a simple time-inhomogeneous Poisson process with intensity $\lambda(t)I\{L(t) < N\} \geq 0$, where $I\{\cdot\}$ is an indicator function. Such counting processes have the semi-martingale representation,

$$I(t) = \int_0^t \lambda(s) I\{L(s) < N\}\, ds + M_I(t),$$

where $M_I(t)$ is a square-integrable martingale. Since $I(t)$ is a sub-martingale it follows from Doob's decomposition theorem that $I(t)$ has the unique decomposition $I(t) = M_I(t) + a(t)$, where $M_I(t)$ is a martingale and $a(t)$ is a predictable increasing process [52]. So, assuming that $L(s) < N$, for all $s \leq t$, the process has independent increments and

$$\mathbb{E}\left[I(t)\right] = \int_0^t \lambda(s)\, ds.$$

The outflows from the dam consist of evaporation with intensity, $\mu(t, X(t)) > 0$, which is a time and state dependent, and controlled consumption with intensity, $C(t) = C(t, p(t, X(t))) > 0$, which depends on the time and state dependent price of water, only if $L(t) > 0$. Similarly to the case of the inflows, we assume that this process can be modeled as a time-inhomogeneous Poisson process with intensity $(C(t, p(t, X(t))) + \mu(t, X(t)))I\{L(t) > 0\}$ and so it has the semi-martingale representation,

$$O(t) = \int_0^t (C(s, p(s, X(s))) + \mu(s, X(s)))I\{L(s) > 0\}\, ds + M_O(t),$$

where $M_O(t)$ is a square-integrable martingale. Again this follows from Doob's decomposition theorem since $O(t)$ is clearly a sub-martingale. The proof is as for the representation of the inflows.

Then, with the above dynamics, we can now define $L(t) = I(t) - O(t)$ as the dam level process.

Remark 2.7. *We assume that $I(t)$ and $O(t)$ are processes whose jumps do not occur at the same instant. This implies that the mutual quadratic variation,* $\langle M_I, M_O \rangle_t = 0$.

Remark 2.8. *The semi-martingale representation has a nice probabilistic interpretation which helps to make sense of the model and its inflow and outflow intensities in terms of real processes. The semi-martingale representation gives the transition time distributions of the states in terms of their dependence on the average inflow and outflow intensities. From it we also can obtain the distribution of the transition state (see Theorem 1 in [33]).*

2.5.3 Infinitesimal generator

Proposition 2.9. *Let the intensity of controlled consumption and evaporation in each state $L(t) \in \{0, ..., N\}$ be $C_i(t, p(t, X(t)))$ and $\mu_i(t, X(t))$ respectively. Then the infinitesimal generator of the controlled Markov chain $X(t)$ has the form,*

$$A(t, p(t, X(t))) =$$

$$\begin{pmatrix} -\lambda(t) & \begin{matrix} C_1(t,p(t,e_1))+ \\ \mu_1(t,e_1)) \end{matrix} & ... & 0 & 0 \\ \lambda(t) & \begin{matrix} -(\lambda(t)+C_1(t,p(t,e_1))+ \\ \mu_1(t,e_1)) \end{matrix} & ... & 0 & 0 \\ 0 & \lambda(t) & ... & 0 & 0 \\ ... & ... & ... & ... & ... \\ 0 & 0 & ... & \begin{matrix} C_{N-1}(t,p(t,e_{N-1}))+ \\ \mu_{N-1}(t,e_{N-1}) \end{matrix} & 0 \\ 0 & 0 & ... & \begin{matrix} -(\lambda(t)+C_{N-1}(t,p(t,e_{N-1}))+ \\ \mu_{N-1}(t,e_{N-1})) \end{matrix} & \begin{matrix} C_N(t,p(t,e_N))+ \\ \mu_N(t,e_N) \end{matrix} \\ 0 & 0 & ... & \lambda(t) & \begin{matrix} -(C_N(t,p(t,e_N))+ \\ mu_N(t,e_N)) \end{matrix} \end{pmatrix}. \tag{2.10}$$

Proof. We can define an increment of the dam level process as $\Delta L(t) = \Delta I(t) - \Delta O(t)$, where Δ is an operator defined as $\Delta h(t) = h(t) - h(t-)$. Then if $L(t-) = k$ and $L(t) = k+1$, we define a vector f such that $I\{L(t-) + \Delta I(t) = k+1\} =$

$I\{X(t-)+f=e_{k+1}\}$. Then f has -1 in the k^{th} position, 1 in the $(k+1)^{th}$ position and zeros elsewhere:

$$f=(0,0,0,...,0,\underbrace{-1}_{k},\underbrace{1}_{k+1},0,...,0,0,0)^T.$$

From this we can construct an $(N+1)\times(N+1)$ matrix, A^+, which captures the effect of inflows on an infinitesimal increment of time starting in any state $k\in\{0,...,N-1\}$, such that

$$A^+=\begin{pmatrix} -1 & 0 & 0 & ... & 0 & 0 & 0 \\ 1 & -1 & 0 & ... & 0 & 0 & 0 \\ 0 & 1 & -1 & ... & 0 & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & -1 & 0 & 0 \\ 0 & 0 & 0 & ... & 1 & -1 & 0 \\ 0 & 0 & 0 & ... & 0 & 1 & 0 \end{pmatrix}. \tag{2.11}$$

Likewise, if $L(t-)=k+1$ and $L(t)=k$, then we can construct a vector g such that $I\{L(t-)-\Delta O(t)=k\}=I\{X_{t-}+g=e_k\}$. Then g has 1 in the k^{th} position, -1 in the $(k+1)^{th}$ position and zeros elsewhere. So

$$g=(0,0,0,...,0,\underbrace{1}_{k},\underbrace{-1}_{k+1},0,...,0,0,0)^T.$$

We now construct an $(N+1)\times(N+1)$ matrix, A^-, which describes the effect of outflows on an infinitesimal increment of time starting in any state $k\in\{1,...,N\}$, such that

$$A^{-} = \begin{pmatrix} 0 & 1 & 0 & ... & 0 & 0 & 0 \\ 0 & -1 & 1 & ... & 0 & 0 & 0 \\ 0 & 0 & -1 & ... & 0 & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & -1 & 1 & 0 \\ 0 & 0 & 0 & ... & 0 & -1 & 1 \\ 0 & 0 & 0 & ... & 0 & 0 & -1 \end{pmatrix}. \tag{2.12}$$

Then we have the following expression for $\Delta X(t)$, $\Delta X(t) = A^{+}X(t-)\Delta I(t) + A^{-}X(t-)\Delta O(t)$, recalling that $I(t)$ and $O(t)$ have unit jumps. Since the matrix $A(t, p(t))$ has bounded entries, the number of jumps is finite with probability one (w.p.1), so we can write the following relation for $X(t)$ without doubts about the existence of the sum: $X(t) = X(0) + \sum_{\tau \leq t} \Delta X(\tau)$, so

$$\begin{aligned} X(t) &= X(0) + \textstyle\sum_{\tau \leq t}[A^{+}X(t-)\Delta I(t) + A^{-}X(t-)\Delta O(t)] \\ &= X(0) + \int_0^t A^{+}X(\tau-)\, dI(\tau) + \int_0^t A^{-}X(\tau-)\, dO(\tau) \\ &= X(0) + \int_0^t [\lambda(\tau)A^{+} + (C(\tau, p(\tau)) + \mu(\tau))A^{-}]X(\tau-)\, d\tau + M(t) \\ &= X(0) + \int_0^t A(\lambda(\tau), C(\tau, p(\tau)), \mu(\tau))X(\tau-)\, d\tau + M(t). \end{aligned} \tag{2.13}$$

The matrix A is clearly of the form given in equation (2.10). □

2.5.4 *Boundaries of optimal consumption*

In this model we use a time and dam level dependent price to modify the demands of customers. However, it is difficult to find the optimal price $p(t, X(t))$ directly and so we optimize the price control through a process of optimizing the consumption. This requires some relationship between the price being charged for water and the customers' reaction to that price. So for all of the models presented we have assumed that each customer reacts to the impost of a price on water by minimizing a quadratic utility function, which best describes the desire to limit deviations from their demanded consumption needs. Let there be n customers each with their own seasonal demand intensity, $\bar{x}_i(t)$, $i = 1, ..., n$, and let $r \in [0, 1]$ be a minimum demand intensity reduction target, such that $(1 - r) \times 100\%$ is the maximum percentage of their demand allowed. The reduction target r is imposed prior to any consideration of price control and may be thought of as a reduction target imposed by an external regulator. This means that the customer may receive less than their demanded water even with a zero price if this target is not zero. Also let x_i be the optimal demand intensity for each customer having taken in to account the price on water. Then, each customer minimizes the utility function

$$f(x_i) = [((1-r)\bar{x}_i(t) - x_i)^2 + p(t, X(t))x_i] I\{x_i \geq 0\},$$

such that $x_i = \arg\min_{x_i} f(x_i)$. Solving this gives

$$x_i(t, p(t, X(t))) = \max\left\{(1-r)\bar{x}_i(t) - \frac{p(t, X(t))}{2}, 0\right\},$$

and so the total optimal intensity of demand for all customers is defined as

$$C(t, p(t, X(t))) = \sum_{i=1}^{n} x_i(t). \tag{2.14}$$

Since $p(t, X(t)) \in [p_{min}, p_{max}]$, we can define natural boundaries for the optimal demand intensity by substituting p_{min} and p_{max} into equation (2.14). So

$$C(t, p(t, X(t))) = \begin{cases} C_{min}(t), & \text{if } p(t, X(t)) = p_{max} \\ C(t, p(t, X(t))), & \text{if } p_{min} < p(t, X(t)) < p_{max} \\ C_{max}(t), & \text{if } p(t, X(t)) = p_{min}. \end{cases} \tag{2.15}$$

With this definition in place, the optimal price for water in each state can be recovered after solution of the optimal control problem.

2.5.5 Dynamic programming equation

While the general idea of dynamic programming was introduced in section 2.1, we now consider the procedure in detail. We begin by stating that in practice dynamic programming is a procedure used to minimize a general performance criterion of the Markov chain states and controls, that is

$$\min_{p(\cdot)} J[p(\cdot)],$$

where

$$J[p(\cdot)] = \mathbb{E}^p \left[\phi^0(X(T)) + \int_0^T f_0(s, p(s, X(s)), X(s))\, ds \right], \tag{2.16}$$

$\phi^0(X(T)) = \langle \phi^0, X(T) \rangle$ is the terminal cost,

$$f_0(s, p(s, X(s)), X(s)) = \langle f_0(s, p(s, X(s)), X(s) \rangle$$

is a cost function (the cost of control) when the system is in state $X(s)$ at time $s \in [0, T]$, and $\langle \cdot, \cdot \rangle$ is the standard inner product. If we now set $X(s) = e_i$, for $i = 0, ..., N$, then we obtain the vector

$$\mathbf{f}_0(s, p(s, X(s))) = (f_0(s, p(s, e_0)), ..., f_0(s, p(s, e_N)))). \tag{2.17}$$

Assumption 2.10. *(a) Each of the functions $f_0(\cdot, \cdot, e_i)$ is continuous on $[0, T] \times [p_{min}, p_{max}]$ and bounded below;*
(b) Each element of the set of vectors $\{A^i(t, p, f_0(t, p, e_i)\}$, where A^i is the i^{th} column vector of A and $p \in P$, is convex for any $i = 0, ..., N$ and $t \in [0, T]$. This means that for each of these functions, the infimum exists.

Next we define the value function

$$V(t, x) = \inf_{p(\cdot)} J[p(\cdot)|X(t) = x] \tag{2.18}$$

where

$$J[p(\cdot)|X(t) = x] = \mathbb{E}^p \left[\phi^0(X(T)) + \int_t^T f_0(s, p(s, X(s)), X(s))\, ds | X(t) = x \right]. \tag{2.19}$$

This is interpreted as the cost of transitions from state $X(t) = x$ at time t to state $X(T)$ at the terminal time T. By Assumption 2.10 the performance criterion (2.16) is bounded below, so the infimum in (2.18) exists, and there is a minimizing sequence of controls $\{p_k(\cdot)\}$. Since for each of the controls $p_k(\cdot)$ we have the function

$$J[p_k(\cdot)|X(t)=x] = \langle \hat{\phi}_k(t), x \rangle$$

with continuous $\hat{\phi}_{(k)}(t)$, then we can write the function

$$V(t,x) = \lim_k \langle \hat{\phi}_{(k)}(t), x \rangle = \langle \hat{\phi}(t), x \rangle,$$

with a measurable column vector-valued function $\hat{\phi}(t) = (\hat{\phi}_0(t), ..., \hat{\phi}_N(t)) \in \mathbb{R}^{N+1}$.

Now let $\phi(t) = (\phi_0(t), ..., \phi_N(t)) \in \mathbb{R}^{N+1}$ be a column vector of measurable functions giving the transition costs for each state, then we have the *dynamic programming equation* with respect to this function $\phi(t)$, in conventional form:

$$\langle \phi'(t), x \rangle + \min_{p \in \bar{P}} [\langle \phi(t), A(t,p)x \rangle + \langle f_0(t,p), x \rangle] = 0, \tag{2.20}$$

with terminal condition $\phi(T) = \phi^0$ [18, 10]. Since the function

$$H(\phi, t, p, x) = \langle \phi(t), A(t,p)x \rangle + \langle f_0(t,p), x \rangle$$

is continuous for any $(t,p) \in [0,T] \times \bar{P}$ and affine in ϕ for any $(t,x) \in [0,T] \times S$, then the function

$$\mathscr{H}(\phi, t, x) = \min_{p \in \bar{P}} H(\phi, t, p, x)$$

is Lipschitz in ϕ with the constant $\mathscr{L} = \max_{(t,p,x)} \|A(t,p)x\|$ and continuous in t for any $x \in S$.

Remark 2.11. *By setting $x = e_i$, $i = 0, ..., N$, we get a system of ordinary differential equations*

$$\frac{d\phi_i(t)}{dt} = -\mathscr{H}(t, \phi(t), e_i), \quad i = 0, ..., N, \tag{2.21}$$

with terminal condition $\phi(T) = \phi^0$. The right-hand side of Equation (2.21) is clearly Lipschitz in ϕ.

Proposition 2.12. *With Assumptions 2.2 and 2.10 holding, equation (2.20) has a unique solution on $[0, T]$ [51].*

The optimal control is then characterized as in the following theorem [18, 15, 32].

Remark 2.13. *As long as Proposition 2.12 holds, the following theorem says that $\phi(t) = \hat{\phi}(t)$.*

Theorem 2.14. *Let $\phi(t)$ be the solution of the system of equations (2.21), then for each $(t, x) \in [0, T] \times S$ there exists $p_0(t, x) \in \bar{P}$ such that $H(t, \phi, p, x)$ achieves a minimum at $p_0(t, x)$. Then*

1. *There exists an $\mathscr{F}_t^X$-predictable optimal control, $\hat{p}(t, X_0^t)$ such that $V(t, x) = J[\hat{p}(\cdot)|X(t) = x] = \langle \phi(t), x \rangle$.*
2. *The optimal control can be chosen as Markovian, that is*

$$\hat{p}(t, X_0^t) = p_0(t, X(t-)) = \arg\min_{p \in \bar{P}} H(t, \phi, p, X(t-)).$$

The following proof is given in [32] and is included for completeness of presentation.

Proof. Let $SD_{[0,T]}$ be the space of all piecewise constant functions $X(t)$ such that $X(t) = \{X(t) \in S, t \in [0, T]\}$.

1. Consider the space of $\omega = X(t) \in SD_{[0,T]}$. For each $(t, \omega) = (t, X(t)) \in [0, T] \times SD_{[0,T]}$ there is a $p_0 \in \bar{P}$ such that

$$\mathscr{H}(\phi(t), t, X(t)) = H(\phi(t), t, p_0, X(t)) = \min_{p \in \bar{P}} H(\phi(t), t, p, X(t)).$$

According to Wan and Davis (1979, Theorem 4.2) [57] there exists $\hat{p}(t, X_0^t)$ which belongs to the class of $\mathscr{F}_t^X$ predictable controls and such that for any $(t, X(t)) \in [0, T] \times S$,

$$\hat{p}(t, X_0^t) = \arg\min_{p \in \bar{P}} H(\phi(t), t, p, X(t)).$$

Also, according to the same theorem, the control can be chosen as a Markov type control, $\hat{p}(t, X_0^t) = p_0(t, X(t-))$.

2. It still must be shown that this control, $\hat{p}(t, X_0^t)$ is optimal. Since $\hat{p}(\cdot)$ is a predictable control then for any initial condition $X(0) \in S$ there exists a unique solution of the martingale problems (2.7) and (2.8). That is to say that there exists a process $X^{\hat{p}}(t) \in SD_{[0,T]}$ that satisfies

$$dX^{\hat{p}}(t) = A(t, (X^{\hat{p}})_0^t)X^{\hat{p}}(t-)\,dt + dM^{\hat{p}}(t), \tag{2.22}$$

where $M^{\hat{p}}(t)$ is a square integrable $\mathscr{F}_t^X$ martingale with quadratic variation given by (2.8).

Take some admissible control $p(s, X(s))$ and the corresponding solution $X^p(\cdot)$ of the martingale problems (2.7) and (2.8) such that $X^p(t) = x$. We then apply Ito's formula to the process $\langle \phi(t), X^p(t)\rangle$, where $\phi(t)$ is the solution of (2.20) and add to both sides of the equation

$$\int_t^T \langle f_0(s, p(s)), X^p(s)\rangle\, ds.$$

We then have

$$\begin{aligned}&\langle \phi(T), X^p(T)\rangle - \langle \phi(t), x\rangle + \textstyle\int_t^T \langle f_0(s, p(s)), X^p(s)\rangle\, ds\\ &= \int_t^T [\langle \phi'(s), X^p(s)\rangle + H(\phi(s), s, p(s), X^p(s))]\, ds + \int_t^T \langle \phi(s), dM^p(s)\rangle.\end{aligned}$$

If we now take the expectation of this equation we find that since $\phi(s)$ is a continuous deterministic function, the integral over the martingale is equal to zero and the expectation of the first integral on the right-hand side is nonnegative because of (2.20), which is of the same form. So,

$$\begin{aligned} J[p(\cdot)|X(t) = x] &= \mathbb{E}^p\left[\langle \phi(T), X^p(t)\rangle + \int_t^T \langle f_0(s, p(s, X(s))), X^p(s)\rangle\, ds\right] \\ &\geq \langle \phi(t), x\rangle = V(t, x). \end{aligned} \tag{2.23}$$

Note that the same calculations with the control $\hat{p}(t, X_0^t) = p_0(t, X(t-))$

and Equation (2.22) give the equality $J[\hat{p}(\cdot)|X(t) = x] = V(t, X)$, which completes the proof.

$\square$

The case of finding feasible solutions given some constraints on the performance criteria will be dealt with in detail in Chapter 6, so further discussion will be deferred until then. In the next chapter we present a basic dam model and the solution results based on the methods of this chapter.

Chapter 3

Control of a single dam with simple counting process inflows

The material presented in this chapter was written for and presented at the "18^{th} IFAC World Congress" held in Milan between the 28^{th} of August and the 2^{nd} of September, 2011 [37]. This chapter presents the basic model upon which all the subsequent models have been built and was also the first attempt at the numerical solution of the optimal control problem. The model and its derivation is explained in detail in Chapter 2, so nothing further will be added here; however, a few words of explanation about the numerical solution are in order. After the presentation of the results, a few comments will follow stressing the weaknesses of this model and the need for the enhancements later developed.

3.1 Initial numerical solution

The numerical solution method used for this model differs from that used in subsequent models in that a quite naive programming method was employed. All the numerical work was done using *Mathematica 7* and this model was somewhat of an experiment in how to efficiently write code and solve the problem posed. Having no prior programming experience, the code for this model was written specifically for a controlled Markov model with a fixed number of states and was not capable of being altered except in some details of parameters. In effect this meant that the general definitions of the optimal controls were calculated prior to writing the code and then the differential equation for each state was written. Some regularity in the form of the differential equations made the automation of this process possible but in retrospect it did not allow for experimentation with different numbers of states or of different performance criteria without rewriting much of the code. It was also written as a purely serial program despite there being quite powerful automatic methods for parallelization in *Mathematica*. This did not significantly change the computation speed given the small number of states in this model but was certainly a deficiency when considering extending the model to larger numbers of states or connected systems.

As stated, the model used here is the basic model given in Chapter 2, with all the definitions and assumptions as given there. The first new element that must be defined for the numerical solution is the specific performance criteria to be used.

3.2 Performance criterion

The goal of optimization in the context of this problem is to minimize some cost function of the Markov chain states and the price controls. Such a function should minimize the difference between the actual customer demand, $\bar{C}(t)$, and the optimal demand, $C(t, p(t, X(t)))$. It should also minimize the average probability that the dam falls below a prescribed level during the control interval, and the probability that the dam is below a prescribed level at the terminal time. These are expressed as the perfromance criteria listed below.

The three specific criteria for this dam problem have the same form as in equation (2.16). The first is the mean square deviation of the total customer demand for water and the water actually supplied. Let $\bar{C}(t) = (1-r)\sum_{i=1}^{n} \bar{x}_i(t)$, where n is the number of consumption sectors then:

$$J_1[p(\cdot)] = \mathbb{E}^p \left\{ \int_0^T \left(C(s, p(s, X(s))) - \bar{C}(s)\right)^2 ds \right\}. \tag{3.1}$$

The second gives the average probability that the dam level falls below level $M \leq N$ over the interval $[0, T]$:

$$J_2[p(\cdot)] = \mathbb{E}^p \left\{ \int_0^T \sum_{i=0}^{M} X_i(s) ds \right\}. \tag{3.2}$$

The third criterion gives the probability that the dam is below level $M \leq N$ at time T:

$$J_3[p(\cdot)] = \mathbb{E}^p \left\{ \sum_{i=0}^{M} X_i(T) \right\}. \tag{3.3}$$

The linear combination of J_1 and J_2 gives us an integral cost function

$$J_1[p(\cdot)] + J_2[p(\cdot)] = \mathbb{E}^p \left\{ \int_0^T f_0(t, p(t, X(t)), X(t)) dt \right\}, \tag{3.4}$$

where

$$\begin{aligned} f_0(t, p(t, X(t)), X(t)) &= \textstyle\sum_{i=0}^{N} \langle (C(t, p(t, e_i)) - \bar{C}(t))^2, e_i \rangle \\ &\quad + \textstyle\sum_{i=0}^{M} \langle \mathbf{l}, X(t) \rangle, \end{aligned} \tag{3.5}$$

where $\mathbf{l} = (1, 1, ..., 1, 0, ..., 0)$ with $M + 1$ first units. This integral cost function appears in the dynamic programming equation and the terminal criterion is accounted for as the initial conditions in the solution of the dynamic programming equations.

3.3 The dynamic programming equations and solutions for a large dam

Using the results of the previous sections we can now show how the optimal control problem for a large dam is solved. Let us take equations (2.21) and evaluate these with the matrix $A(t, p(t, X(t)))$ and the function $f_0(t, p(t, X(t)), X(t))$. Doing so, we get the following dynamic programming equations:

$$
\begin{aligned}
0 &= \phi_0'(t) + \min\nolimits_{C_0(\cdot)}\{\lambda(t)(\phi_1(t) - \phi_0(t)) + (C_0(t, p(t, e_0)) - \bar{C}(t))^2\} + 1 \\
0 &= \phi_1'(t) + \min\nolimits_{C_1(\cdot)}\{(C_1(t, p(t, e_1)) + \mu_1(t, e_1))(\phi_0(t) - \phi_1(t)) + \\
&\quad \lambda(t)(\phi_2(t) - \phi_1(t)) + (C_1(t, p(t, e_1)) - \bar{C}(t))^2\} + 1 \\
\cdots &\quad \cdots \\
0 &= \phi_M'(t) + \min\nolimits_{C_M(\cdot)}\{(C_M(t, p(t, e_M)) + \mu_M(t, e_M))(\phi_{M-1}(t) - \phi_M(t)) + \\
&\quad \lambda(t)(\phi_{M+1}(t) - \phi_M(t)) + (C_M(t, p(t, e_M)) - \bar{C}(t))^2\} + 1 \\
\cdots &\quad \cdots \\
0 &= \phi_N'(t) + \min\nolimits_{C_N(\cdot)}\{(C_N(t, p(t, e_N)) + \mu_N(t, e_N))(\phi_{N-1}(t) - \phi_N(t)) + \\
&\quad (C_N(t, p(t, e_N)) - \bar{C}(t))^2\}.
\end{aligned} \tag{3.6}
$$

Here we take $C(t, p(t, X(t)))$ as the control for ease of calculation. Since $C(t, p(t, X(t)))$ depends linearly on the price, $p(t, X(t))$ can be recovered for each state after solution.

As mentioned in Section 3.2, the terminal conditions account for the J_3 criterion by attaching a reasonable but significant cost to all states less than the prescribed state M. That is, for $X_i(T) \leq M$, $i = 0, ..., M$, $\phi_i(T) = K$, where K the cost penalty for ending in this state. For all $X_i(T) > M$, $i = M+1, ..., N$, $\phi_i(T) = 0$. The size of K must be significant enough to ensure that the solution is sensitive to the criterion whilst still being reasonable.

Now, $C(t, p(t, X(t))) = \sum_{i=1}^{n} x_i(t, p(t, X(t)))$, where $x_i(t, p(t, X(t)))$ is defined as in Section 2.5.4. We need the minimum of this function in each of the above equations (3.6) for each t. The absolute maximum of $C(t, p(t, X(t)))$ occurs when the price is the stipulated minimum set by the regulator, that is

$$\begin{aligned} C_{max}(t) &= (\bar{C}(t) - \tfrac{p_{min}}{2})\times \\ & \quad I(\bar{C}(t) - \tfrac{p_{min}}{2} \geq 0), \end{aligned} \tag{3.7}$$

where p_{min} is the minimum price. The absolute minimum occurs with the maximum stipulated price and so is

$$\begin{aligned} C_{min}(t) &= (\bar{C}(t) - \tfrac{p_{max}}{2})\times \\ & \quad I(\bar{C}(t) - \tfrac{p_{max}}{2} \geq 0), \end{aligned} \tag{3.8}$$

where p_{max} is the maximum price and $I(\cdot)$ is an indicator function, which is equal to one if the condition is true or zero if false. If the minimizing function is below or above the absolute minimum or maximum respectively, then we take the absolute minimum or maximum as the minimizing function. So, considering each equation, we minimize it and find the conditions which give us the correct minimizing function.

From the first, differentiating with respect to $C_0(t, p(t, e_0))$ gives us

$$C_0(t, p(t, e_0)) = (1 - r)\bar{C}(t) \tag{3.9}$$

which implies that $C(t, p(t, e_0))$ is equal to the minimum reduced demand target. For $C_i(t, p(t, e_i))$, $i = 1, ..., N$, the minimizing equations are given by

$$C_i(t, p(t, e_i)) = \bar{C}(t) + \frac{\phi_i(t) - \phi_{i-1}(t)}{2}. \tag{3.10}$$

The functions $C_i(t, p(t, e_i))$, $i = 0, ..., N$ can be summarized as follows:

1. $C_0(t, p(t, e_0)) = \bar{C}(t)$ is greater than $C_{max}(t)$ for all t and nonzero price $p(t, e_0)$, so we take $C_{max}(t)$ as the minimizing function.
2. For the remainder of the equations, the minimizing function is given by the following definition:

$$C_i(t, p(t, e_i)) = \begin{cases} C_{max}(t), \quad \text{if} \quad \bar{C}(t) + \frac{\phi_i(t) - \phi_{i-1}(t)}{2} > C_{max}(t), \\ \bar{C}(t) + \frac{\phi_i(t) - \phi_{i-1}(t)}{2}, \\ \text{if} \quad C_{max}(t) \geq \bar{C}(t) + \frac{\phi_i(t) - \phi_{i-1}(t)}{2} \geq C_{min}(t), \\ C_{min}(t), \quad \text{if} \quad \bar{C}(t) + \frac{\phi_i(t) - \phi_{i-1}(t)}{2} < C_{min}(t). \end{cases} \tag{3.11}$$

With these minimizing equations, the system (3.6) is now a system of ordinary differential equations, which can be solved numerically.

3.4 Numerical example

For this example, Mathematica 7 was used to solve the system of ODE's and provide plots which demonstrate the effect of the optimal price control on consumption. The following functions and parameters were used as the basis of the model:

- $N = 20$;
- $M = 10$;
- $T = 1$;
- maximum price, $p_{max}(t) = 2.5$, and minimum price, $p_{min}(t) = 2$;
- inflow function, $\lambda(t) = \sin(2\pi t) + 10$;
- natural loss function at the maximum level, $\mu_L(t) = -\sin(2\pi t) + 2.5$;
- natural loss function at lower levels, $\mu_i(t) = \frac{L-i}{L}\mu_L(t)$ for $i = 1, ..., L-1$;
- demand functions, $\bar{x}_1(t) = \cos(2\pi t) + 4.5$, $\bar{x}_2(t) = 0.3\cos(2\pi t) + 3.5$ and $\bar{x}_3(t) = 0.5\cos(2\pi t) + 5$;
- $r = 0.25$; and
- $K = 100$.

The above parameters give us a one year control period for a dam with twenty levels. The regulator has stipulated that the maximum price to be charged is 2.5 (dollars per kiloliter, say) and the minimum is 2. We have a minimum reduction target of 25% off uncontrolled demand and have limited the water that can be consumed above net natural flows to 20%. Natural losses are mostly due to evaporation and this largely depends on the surface area of the dam. For this simple model we have assumed that the losses decrease linearly, however, for any real dam this would require significant modeling in itself. We also have a terminal cost penalty of 100 if the dam level is at or below level 11 at time $T = 1$. This penalty would be paid by the dam manager to the regulator. Recall that the J_2 criterion added a unit cost to the running transition cost of each level. This was found to be too low a cost and the solution was insensitive to it, so for each state where such a cost applied, it was multiplied by $K = 100$.

Figure 3.1 shows the demand functions and the unweighted mean natural losses along with the inflow function. Clearly total demand and loss exceeds inflows and so the necessity of controlling the demand is well demonstrated, particularly if the dam starts in a low level. Figure 3.2 gives the maximum and minimum consumption curves used to decide the optimal consumption function for each level in this model. Again, it is clear that the uncontrolled demand is well above the maximum consumption.

After the solution of the system of ODE's, the solutions were substituted back into the consumption equations, $C_i(t, p(t, e_i))$, $i = 0, ..., 20$. From these equations the optimal price functions were easily found. Figure 3.3 shows the weighted average of controlled demand and the original demand over the control period, assuming that the dam started in level 11. The weighting is given by the probability that the dam was in state i at time t and was found by solving the forward Kolmogorov equation,

$$\frac{dP(t)}{dt} = A(t, p(t, X(t)))^T P(t) \tag{3.12}$$

with initial condition,

$$P(0) = e_{10}.$$

So, the weighted average of controlled demand is

$$\sum_{i=0}^{N} P_i(t) C_i(t, p(t, e_i)),$$

where $P_i(t)$ is the probability of the dam being in state i at time t. Figure 3.4 shows four of the controlled demand functions and the shape of these functions explains why the weighted average of controlled demand is not a smooth function. The probabilities are smooth but the multiplication by non-smooth functions leads to a rather irregular curve. Even so, it is clear that the control is effective on the control period.

Figure 3.5 gives an indication of the nature of the price functions produced by the optimization. Each function is piecewise continuous with few jumps on the control period. As the state changes one need simply change to the price function for that state. Clearly price does not increase monotonically as the dam level gets lower. The price depends on the state and time in a very complex way due to the different criteria we want dam performance to meet. Figure 3.10 gives the solution curves of the ODE system. Since the solutions depend on the difference between the current state, the state below and the state above, we can see that there will be frequent sign changes because the solution curves are so close together. Figures 3.6, 3.7 and 3.8 show the price structure at various times on the control period. It is clear that the prices reflect this behavior of the ODE system solutions.

Figure 3.9 shows the cumulative controlled consumption in states 1, 5, 10, 15 and 20. They are monotonically increasing so our prices have not altered the general nature of outflows. It is likely that the differing demands of each customer in the dam is causing the price behavior we see. Note that the cumulative consumption in each state shown is almost the same. This is due

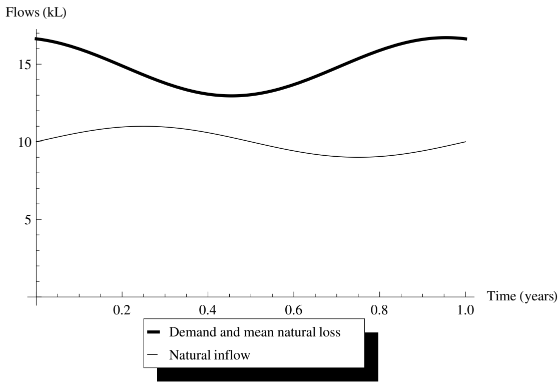

Figure 3.1: Demands, mean natural flows and losses.

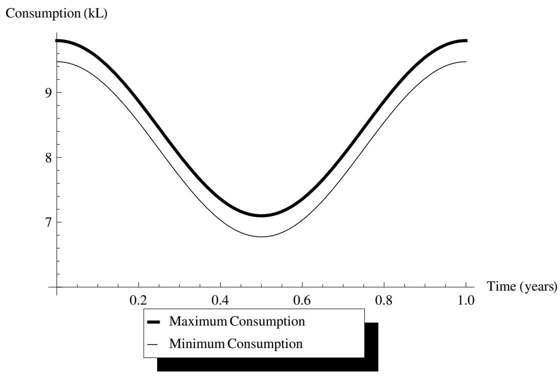

Figure 3.2: Maximum and minimum consumption curves.

to the optimal consumption rate falling within a very narrow band between its maximum and minimum at any given time. This may not be a practical strategy to control the dam level, due to the non-monotonic changes in price, but does demonstrate emphatically the difficulty of finding a practical strategy to control the dam level primarily through price. We must also consider other types of control.

3.5 General comments on the results

At this stage we have shown that it is possible to find optimal price functions for a dam with twenty states while taking into account a number of important performance criteria. This provides a solid framework to model larger, more

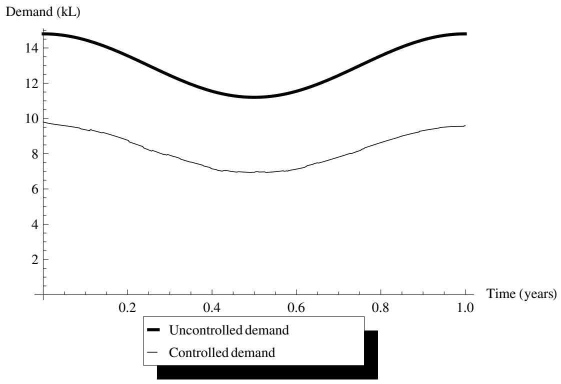

Figure 3.3: Uncontrolled consumption and average controlled consumption.

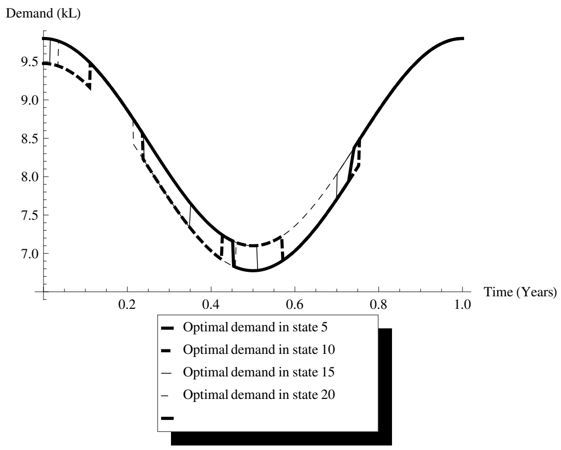

Figure 3.4: Optimal demand in states 5,10,15 and 20.

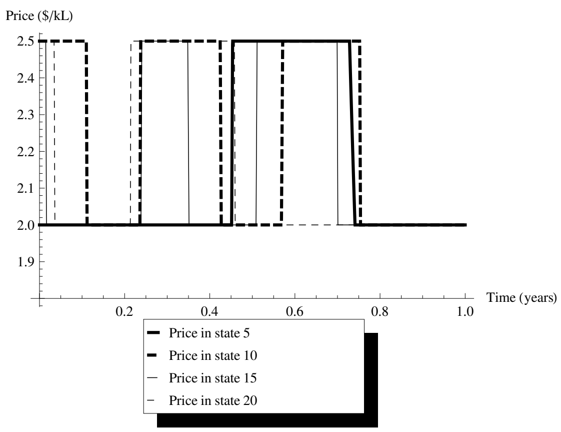

Figure 3.5: Optimal price functions for states 5,10,15,and 20.

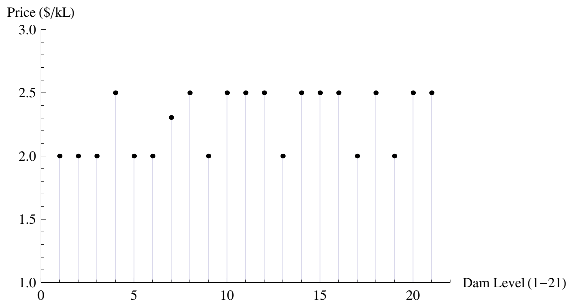

Figure 3.6: Price against dam level at t=0.25.

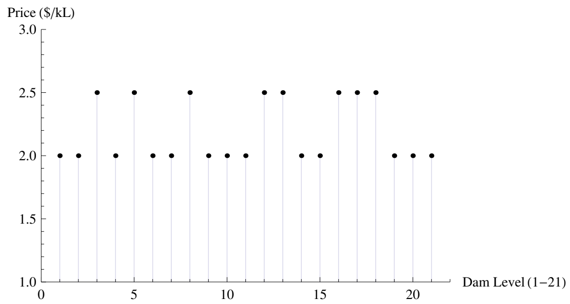

Figure 3.7: Price against dam level at t=0.5.

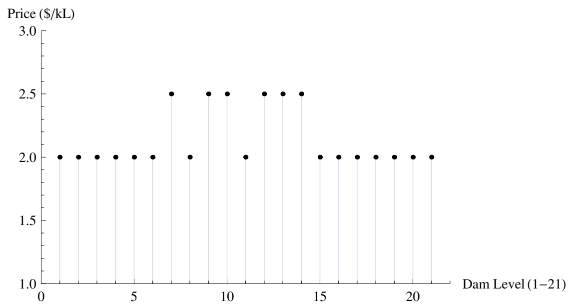

Figure 3.8: Price against dam level at t=0.75.

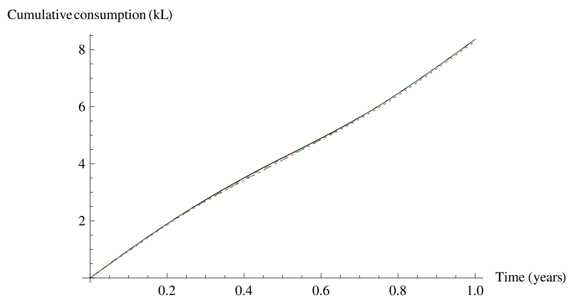

Figure 3.9: Cumulative consumption for states 1,5,10,15,20.

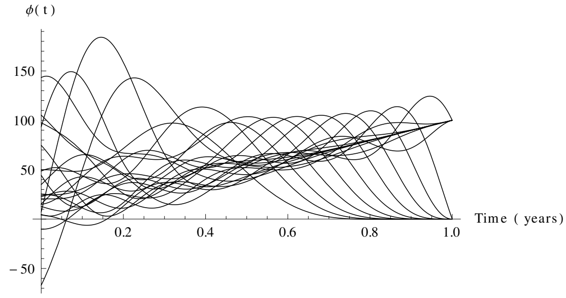

Figure 3.10: Dynamic programming ODE solution curves.

realistic systems. An improvement would be to increase the number of states so as to improve the smoothness of the resulting price function as it moves from state to state. This would make such a strategy more attractive to implement, however, numerically this will be far more computationally intensive and so we must find efficient means of simultaneously solving large systems of differential equations. Another improvement would be to increase the number of dams in the system and have them coupled together such that water can be moved under control between dams. If there were N states in each dam and M dams, this would lead to a system of N^M differential equations which need to be solved simultaneously. To this end we will try to implement this with high performance computing techniques (HPC), such as parallel computing, and Chapter 7 gives detail on the progress of these efforts.

3.6 Further analysis of the results

The development of the model and numerical solution method presented in this chapter provided a good basis for further work but there were some obvious deficiencies considering the results. The main deficiency is clearly seen in Figures 3.6 to 3.8, showing the optimal water price at specific times during the control period. There is no monotonicity in the prices. One would reasonably expect that as the level of water in the dam decreased that the price would monotonically increase but this clearly does not occur in this model. Figure 3.10 along with the definition of the optimal consumption functions given in Equation (3.11) explain this.

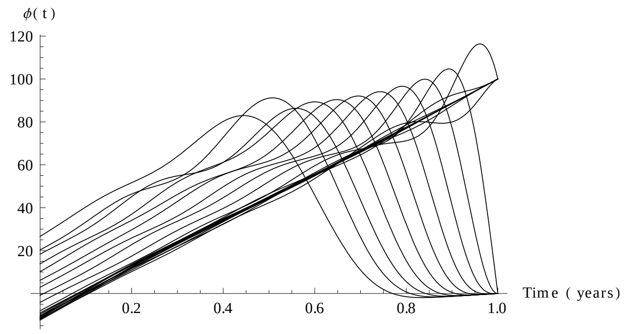

Figure 3.11: ODE system solution curves with no control applied.

The solution curves of the differential equations for the states are quite nonlinear and the solution curve for state i may be crossed at multiple points by the solution curves for states $i-1$ and $i+1$. The optimal consumption definitions for each state depend on the difference between these curves, that is the absolute difference and the sign, and so there is a rapid changing of the prices. It is instructive to compare the ODE system solutions presented above with those found with no control. Figure 3.11 shows solutions with no price control applied. There is a much more linear character to the solutions without controls and this implies that most of the nonlinearity in the controlled case comes from the controls themselves. From a practical point of view this would not be an attractive control solution.

On the other hand, if we consider extreme initial conditions, such as starting with the dam being either almost empty or full, we find that from a probabilistic perspective, the control is effective.

3.6.1 Starting in the lowest state

If we begin in the lowest possible state, which signifies that the dam is near empty, then we find that the probabilities of remaining in this state decline to approximately 25% by the end of the control period, as seen in Figure 3.12. The probability declines rapidly to this level but still remains the most likely outcome, although very closely matched with the level increasing to level 1. The probabilities of increasing above level 1 decrease in order. Compare this with Figure 3.13,

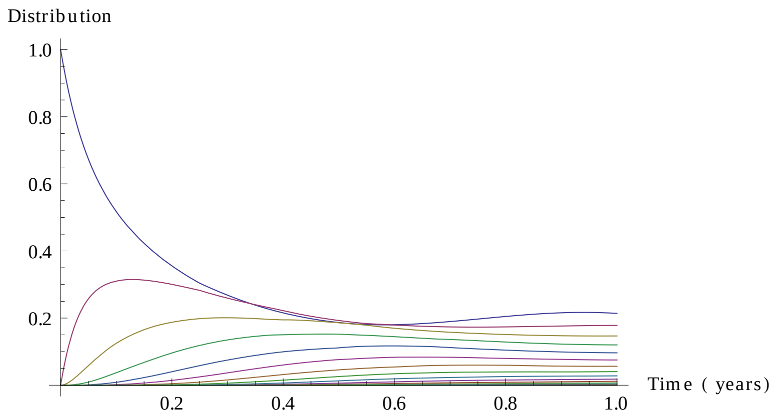

Figure 3.12: State probabilities against time starting in state L(0)=0.

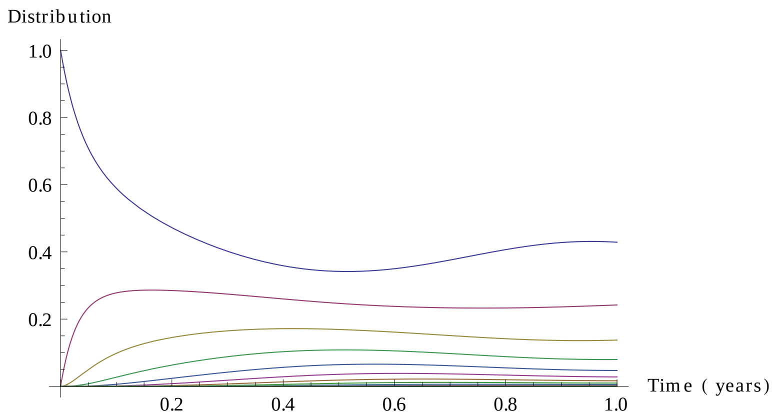

Figure 3.13: State probabilities against time starting in state L(0)=0 with no control.

which gives the transition probabilities with no control applied. There is a significant decrease in the probability of staying in level 0, to approximately 40%, but there is a wide gap from this to the probability of increasing to level 1. Clearly the controlled solution gives a better outcome and the control does affect the transition probabilities.

3.6.2 Starting in the highest state

Likewise, if we begin in the highest state, when the dam is near full, then we find in the controlled case that the probability of remaining in this state falls quite rapidly at first and the more slowly until it is a little under 20% at the end of the control period, as shown in Figure 3.14. It remains the most likely outcome for most of this time. On the other hand, if we compare with the case where no controls are applied, we find that the probability of remaining in the highest

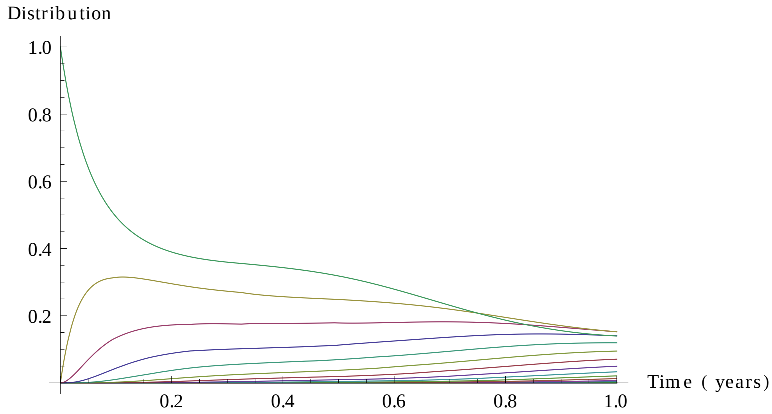

Figure 3.14: State probabilities against time starting in state L(0)=20.

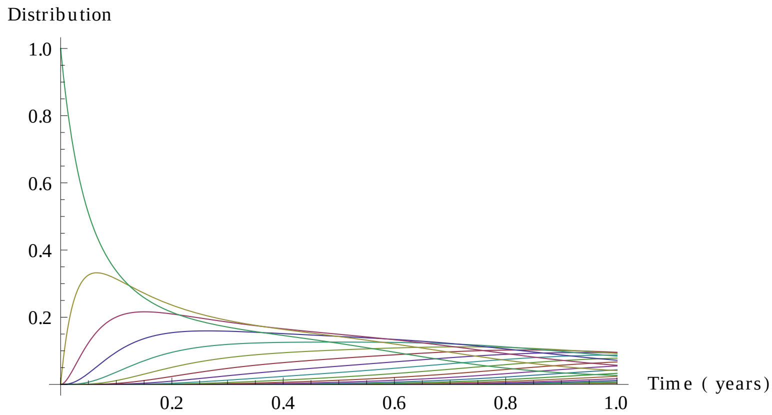

Figure 3.15: State probabilities against time starting in state L(0)=20 with no control.

state drops very rapidly and is about 10% at the end of the control period, as seen in Figure 3.15. This is certainly intelligible considering that in this model there is only a control to reduce water consumption but not to control potential flooding. So here the control has the effect of making the probability of flood more likely and is a significant weakness of this model. This is addressed in later models.

Chapter 4

Control of a system of dams with simple counting process inflows

This chapter is the result of a paper written for and presented at the "International Conference on Computer Science, ICCS 2011", held from the 1^{st} to the 3^{rd} of June, 2011, at Nanyang Technological University, Singapore. It came about as a natural extension of the dam, described by the model in Chapter 3, being connected to another dam, perhaps with different dynamics. In order to connect the two dams a transfer control was introduced. This allowed the state and time dependent transfer of water in both directions, but in only one direction at a time.

One approach to the representation of this system is to construct a single generator for the entire coupled system and treat it as a single controlled Markov chain. This has the disadvantage of resulting in a generator matrix which has dimensions $((N+1) \times (M+1)) \times ((N+1) \times (M+1))$ for two controlled Markov chains with $N+1$ and $M+1$ states respectively. Also, taking the same example, the state space of the coupled chains has dimensions $(N+1) \times (M+1)$ and so the representation of the system of ODE's is cumbersome, especially their setup for numerical work.

The approach that we have taken is to consider the state space first, not the generator of the coupled chain, and use tensors to simplify the representation. The joint state of the connected chains can be viewed as the direct product of the states of the individual chains. When represented in this way, it is a simple operation to take the derivative with respect to time of the direct product of the states and find a system of differential equations in terms of the infinitesimal generators of the individual subcomponents of the system. Computationally both representations give the same result with the same computational challenges, but this representation is more compact and easier to write for computational purposes.

The infinitesimal generator is only slightly modified in this case and the transfer control is simply added to the outflow process, in all states where outflows are possible, and to the inflow process, where inflows come from other dams. However, this is for only one dam in the system so we then have a separate generator for each dam in the system. Here it must be stated that the dependence on the state is more precisely dependence on the joint state of the system. The same applies to price, which for simplicity we took to be a single price on water for the whole system. This joint state dependence of the price could be replaced by separate prices for each dam dependent only on the state of the relevant dam with only minimal changes to this model.

The practical and numerical challenge here was that the size of the resulting system is $(N+1)^d$ for d dams each with $N+1$ states. So, for example, if $N=20$ and $d=2$ then we will have a set of 441 nonlinear ODE's to solve. If $d=3$ then there are 9261 ODE's, and so on. There is little that can be done about this aspect of the numerical solution other than using High Performance Computing (HPC) and parallelization where possible. This is discussed in Chapter 7. The challenge is obtaining the set of ODE's in a systematic way.

In this chapter we consider the optimal management of a system of d dams via a state and time dependent price control and flow controls between dams in the system. The level of each dam in the system is approximated by $N+1$ discrete levels and each is then modeled as a continuous-time controlled Markov chain. The general approach to the solution of this type of problem is to reduce the stochastic problem to a deterministic one with integral and terminal optimality criteria and then solve it via dynamic programming. This type of problem has been solved for server queuing systems by Miller [32] and in general by Miller et. al. [35, 34].

4.0.3 Structure

In Section 4.1 we detail the method of modeling a multi-dam system as linked continuous-time controlled Markov chains. Section 4.2 provides details of the key innovation of this model and explains how state dependent consumption functions are derived for each dam given our price feedback control. In Section 4.4 we consider a general performance criterion and the general solution of this problem as developed in [32, 35, 34]. Section 5 develops the specific performance criteria appropriate to this setting.

Section 6 will discuss some issues dealing with the numerical solution and the application of parallelization to parts of the solution. Section 7 will give a numerical example for a system with only two dams so that the solutions can be readily visualized. The final section will outline future directions for research and enhancement of this model and solution method.

4.1 Model of the controlled dam system

In order to model the dam system, we make some assumptions about the behavior of each dam. We assume that each dam has a natural inflow process, independent of flows into other dams. We likewise assume that natural losses from each dam due to evaporation are independent of evaporative losses in other dams. In terms of consumption, we assume that the consumption in each dam is controlled by a time and state dependent price and therefore depends on the joint state of the system. Likewise, cross-flows between dams are time and state dependent and depend on the joint state of the system.

For each dam, we approximate its level by discretizing it into $N + 1$ states, $N < \infty$, and then let $L_i(t) \in \{0, ..., N\}$, $i = 1, ..., d$, be an integer valued random variable describing the level of dam i at time t. Using the martingale approach [18] we describe the $N + 1$ possible levels in each dam by unit vectors in $\mathbb{R}^{N+1}$, giving us $S_i = \left\{e_0^{(i)}, ..., e_N^{(i)}\right\}$ for each $i = 1, ..., d$.

All processes are defined on the probability space $\{\Omega, \mathscr{F}, \mathbb{P}\}$. Specifically, we define $X_i(t)$, $i = 1, ...d$, where $\{X_i(t) \in S_i, t \in [0, T]\}$ for $T < \infty$, as a controlled jump Markov process with piecewise constant right-continuous paths. Clearly this represents the process of change in the level of each dam on the interval $[0, T]$. Let the joint state be defined as $\mathbf{X}(t) = X_1(t) \otimes X_2(t) \otimes ... \otimes X_d(t)$ and assume that $\mathbf{X}(t)$ generates a family of right continuous σ-algebras, $\mathscr{F}_t^{\mathbf{X}} = \sigma\left\{\mathbf{X}(s) : s \in [0, T]\right\}$. We make the following assumption about the control, $p(t, \mathbf{X}(t))$, and the transfer controls, $u^{(i \to j)}(t, \mathbf{X}(t))$, between dams i and j, for $i, j = 1, ..., d$ and $i \neq j$.

Assumption 4.1. *Assume that the set of admissible controls,* $\bar{P} = p(\cdot)$ *and* $\bar{U} = \left\{u^{(i \to j)}(\cdot) : i, j = 1, ..., d; i \neq j\right\}$ *are sets of* $\mathscr{F}_t^{\mathbf{X}}$*-predictable controls taking values in* $P = \{p \in [p_{min}, p_{max}]\}$ *and* $U = \{u \in [0, 1]\}$ *respectively.*

Remark 4.2. *Assumption 4.1 ensures that if the number of jumps in the* i^{th} *dam up to time* $t \in [0, T]$ *is* $N(t)$*,* τ_k *is the time of the* k^{th} *jump and*

$$X_{i,0}^t = \left\{(X_{i,0}, 0), (X_{i,1}, \tau_1), ..., (X_{i,N(t)}, \tau_{N(t)})\right\}$$

is the set of states and jump times, then for $\tau_{N(t)} \leq t < \tau_{N(t+1)}$ *the controls* $p(t, \mathbf{X}_0^t)$ *and* $u^{(i \to j)}(t, \mathbf{X}_0^t)$ *are measurable with respect to* t *and* $\mathbf{X}_0^t$ *[18, 32].*

4.1.1 Dam system dynamics

In this approach it is supposed that the inflows and outflows of each dam in the system can be approximated by general $\mathscr{F}_t^{\mathbf{X}}$-predictable counting processes with unit jumps and let the inflow into dam i be $Y_{in}^{(i)}(t)$. It is assumed that the natural component of this has a deterministic intensity, $\lambda_i(t) \geq 0$. The intensity of inflow components from other dams are the result of the water transfer controls, $\left\{u^{(j \to i)}(t) : j, i = 1, ..., d; j \neq i\right\}$. So for the i^{th} dam the inflow process has the following form:

$$Y_{in}^{(i)}(t) = \int_0^t (\lambda_i(s) + \sum_{j=1}^{d} u^{(j \to i)}(s)) I\left\{L_i(s) < N\right\} ds + M_{in}^{(i)}(t), \tag{4.1}$$

where $M_{in}^{(i)}(t)$ is a square integrable martingale with quadratic variation

$$\langle M_{in}^{(i)} \rangle_t = \int_0^t (\lambda_i(s) + \sum_{j=1}^{d} u^{(j\to i)}(s)) I\left\{L_i(s) < N\right\} ds.$$

For the outflow process in each dam there is a natural component and components due to consumption and water transfer controls. Let the outflow from dam i be $Y_{out}^{(i)}(t)$ and let the intensity of evaporation from dam i be $\mu_i(t, \mathbf{X}(t))$, such that the intensity depends on the joint state of the process. The intensity of outflows from water transfers are $\left\{u^{(i\to j)}(t, \mathbf{X}(t)) : i, j = 1, ..., d; j \neq i\right\}$ and the intensity of outflow from consumption is the controllable consumption rate $C_i(t, p(t, \mathbf{X}(t)))$, which depends on the current price of water and the intensity of customer demands. This will be derived in the next section. So, in similar form to $Y_{in}^{(i)}(t)$,

$$\begin{aligned} Y_{out}^{(i)}(t) = &\int_0^t (\mu_i(s, \mathbf{X}(s)) + C_i(s, p(s, \mathbf{X}(s))) + \sum_{j=1}^{d} u^{(i\to j)}(s, \mathbf{X}(s))) I\left\{L_i(s) > 0\right\} ds \\ &+ M_{out}^{(i)}(t), \end{aligned} \tag{4.2}$$

where $M_{out}^{(i)}(t)$ is a square integrable martingale with quadratic variation

$$\langle M_{out}^{(i)} \rangle_t = \int_0^t (\mu_i(s, \mathbf{X}(s)) + C_i(s, p(s, \mathbf{X}(s))) + \sum_{j=1}^{d} u^{(i\to j)}(s, \mathbf{X}(s))) I\left\{L_i(s) > 0\right\} ds.$$

It follows that the approximate dynamics for the i^{th} dam are governed by the equation

$$L_i(t) = Y_{in}^{(i)}(t) - Y_{out}^{(i)}(t). \tag{4.3}$$

It should be emphasized that the dynamics of each dam clearly depend on the dynamics of the other dams.

In essence by splitting each dam into N levels we are saying that the mean time between level changes of the continuous flow processes correspond with the mean time between jumps in our counting process approximations. The martingale terms provide the random perturbation about this mean and, importantly, the mean of the martingale terms is zero.

4.1.2 Controlled dam system as a system of controlled Markov chains

The above approximation of the dam system dynamics allow us to make the following proposition with respect to each dam in the system.

Proposition 4.3. *Given the approximate dynamics for the i^{th} dam, as stated in 4.1.1, in a system of d dams, the controlled process for this dam is represented by a controlled Markov chain with $(N+1)^d$ states and, taking into account the representation of the system as tensors, the $(N+1) \times (N+1)$ matrix describing the generator of the i^{th} dam is,*

$$A_i(t, p(t, \mathbf{X}(t)), u^{(i\to j)}(t, \mathbf{X}(t)), u^{(j\to i)}(t, \mathbf{X}(t))) =$$

$$\begin{pmatrix}
\begin{array}{l} -\lambda_i(t) \\ -\sum\limits_{j\neq i} u_i^{(j\to i)}(t,\mathbf{X}(t)) \end{array} & \begin{array}{l} C_i(t,p(t,\mathbf{X}(t))) \\ +\mu_i(t,\mathbf{X}(t)) \\ +\sum\limits_{j\neq i} u_i^{(i\to j)}(t,\mathbf{X}(t)) \end{array} & \cdots & 0 & 0 \\
\begin{array}{l} \lambda_i(t) \\ +\sum\limits_{j\neq i} u_i^{(j\to i)}(t,\mathbf{X}(t)) \end{array} & \begin{array}{l} -(C_i(t,p(t,\mathbf{X}(t))) \\ +\mu_i(t,\mathbf{X}(t)) \\ +\sum\limits_{j\neq i} u_i^{(i\to j)}(t,\mathbf{X}(t)) \\ +\lambda_i(t) \\ +\sum\limits_{j\neq i} u_i^{(j\to i)}(t,\mathbf{X}(t))) \end{array} & \cdots & 0 & 0 \\
0 & \begin{array}{l} \lambda_i(t) \\ +\sum\limits_{j\neq i} u_i^{(j\to i)}(t,\mathbf{X}(t)) \end{array} & \cdots & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & \cdots & \begin{array}{l} C_i(t,p(t,\mathbf{X}(t))) \\ +\mu_i(t,\mathbf{X}(t)) \\ +\sum\limits_{j\neq i} u_i^{(i\to j)}(t,\mathbf{X}(t)) \end{array} & 0 \\
0 & 0 & \cdots & \begin{array}{l} -(C_i(t,p(t,\mathbf{X}(t))) \\ +\mu_i(t,\mathbf{X}(t)) \\ +\sum\limits_{j\neq i} u_i^{(i\to j)}(t,\mathbf{X}(t)) \\ +\lambda_i(t) \\ +\sum\limits_{j\neq i} u_i^{(j\to i)}(t,\mathbf{X}(t))) \end{array} & \begin{array}{l} C_i(t,p(t,\mathbf{X}(t))) \\ +\mu_i(t,\mathbf{X}(t)) \\ +\sum\limits_{j\neq i} u_i^{(i\to j)}(t,\mathbf{X}(t)) \end{array} \\
0 & 0 & \cdots & \begin{array}{l} \lambda_i(t) \\ +\sum\limits_{j\neq i} u_i^{(j\to i)}(t,\mathbf{X}(t)) \end{array} & \begin{array}{l} -(C_i(t,p(t,\mathbf{X}(t))) \\ +\mu_i(t,\mathbf{X}(t)) \\ +\sum\limits_{j\neq i} u_i^{(i\to j)}(t,\mathbf{X}(t))) \end{array}
\end{pmatrix}.$$

The column number corresponds to the current state of the i^{th} dam and the column entries add to zero. This generator is constructed in such a way that if the i^{th} dam is empty, then no outflows are possible, and if the i^{th} dam is full, then no inflows are possible.

Proof. The proof of this proposition is accomplished in the same way as for the generator of a controlled Markov chain for a single queuing system given by Miller [32]. The only difference is that the chains are linked via water transfer controls and a price for the joint states, however, as all controls are $\mathscr{F}_t^{\mathbf{X}}$-predictable, this does not affect the proof. □

4.2 Derivation of controlled demand functions

As already stated, the key innovation of this model is the use of a time and state dependent feedback control, $p(t, \mathbf{X}(t))$, to take into account the active seasonal demands of consumers. It is more intuitive and makes calculation easier to find $p(t, \mathbf{X}(t))$ through the effect it has on consumption in each dam. For the i^{th} dam, the resulting controlled demand is denoted $C_i(t, p(t, \mathbf{X}(t)))$. Here we show how we take the price of water into account through controlled consumption. To be clear, we are looking for a single price structure for all users of the *dam system.*

So, considering the i^{th} dam, let there be n sectors, or consumers, each with their own seasonal demand intensity, $\bar{x}_{i,k}(t)$, for $k = 1, ..., n$. In order to have some control on this demand intensity we want to set an optimal demand intensity for each sector, which we denote $x_{i,k}$, for $k = 1, ..., n$. Now we define in what sense we want this target to be optimal by defining the utility function as $\min_{x_{i,k}} f_{i,k}(x_{i,k})$ where $f_{i,k}(x_{i,k}) = (x_{i,k} - (1-r)\bar{x}_{i,k}(t))^2 + p(t, \mathbf{X}(t))x_{i,k}$, and r is a minimum demand reduction target. To find the minimum we differentiate and solve for $x_{i,k}$, giving

$$x_{i,k}(t, p(t, \mathbf{X}(t))) = ((1-r)\bar{x}_{i,k}(t) - \frac{p(t, \mathbf{X}(t))}{2}) I \left\{ (1-r)\bar{x}_{i,k}(t) - \frac{p(t, \mathbf{X}(t))}{2} \geq 0 \right\}.$$

This is the optimal intensity of demand for the k^{th} sector. It follows that for the i^{th} dam, the total optimal intensity of demand is

$$C_i(t, p(t, \mathbf{X}(t))) = \sum_{k=1}^{n} x_{i,k}(t, p(t, \mathbf{X}(t))). \tag{4.4}$$

We now have a vector of optimal demand intensities for the i^{th} dam.

Since we also know that $p(t, \mathbf{X}(t)) \in [p_{min}, p_{max}]$, we can now also define maximum and minimum optimal demand intensities for the i^{th} dam in the following way:

$$\begin{aligned} C_{i,max}(t) = \sum_{k=1}^{n} x_{i,k}(t, p_{min}) \geq C_i(t, p(t, \mathbf{X}(t))) = \sum_{k=1}^{n} x_{i,k}(t, p(t, \mathbf{X}(t))) \\ \geq C_{i,min}(t) = \sum_{k=1}^{n} x_{i,k}(t, p_{max}). \end{aligned} \tag{4.5}$$

These equations allow us to define piecewise functions for the solution of $C_i(t, p(t, \mathbf{X}(t)))$ in the dynamic programming equations.

4.3 Dynamic programming and optimal control

In Chapter 2 the solution for the optimal control of a single server queuing system was developed via dynamic programming. This method can be extended to systems defined by multiple controlled Markov chains. Since with the extension to d connected dams we are still dealing with essentially one controllable Markov chain, the results from Chapter 2 carry over with only the slight modifications required for the increased number of controls. As these are mostly trivial extensions of the results already given we will focus here on the representation of the controllable Markov chain and how this affects the results already given.

4.3.1 Extension to d dams

There are two ways to extend these results to d dams. The first is to find the infinitesimal generator of the controlled Markov chain for the entire system, writing the entire set of possible joint states as a vector and solving as in section 2.5.5. While this is possible it presents practical problems. Firstly, the generator is of the form

$$G = A_1 \otimes I \otimes I \otimes ... \otimes I + I \otimes A_2 \otimes I \otimes ... \otimes I + ... + I \otimes I \otimes ... \otimes A_{d-1} \otimes I + I \otimes I \otimes ... \otimes I \otimes A_d,$$

which is a matrix of dimensions $(N+1)^d \times (N+1)^d$. With a large number of states and dams this is a rather complex matrix to construct for computational use. The representation used here is much more convenient, since it deals with the generators of each dam separately, and relies on Theorem 2.14.

We consider the value function $V(t, \mathbf{x}) = \langle \phi(t), x_1 \otimes x_2 \otimes ... \otimes x_d \rangle$, where $\phi(t)$ is a tensor of order d. From Theorem 2.14 we know that there exist optimal Markovian controls that satisfy $V(t, \mathbf{x}) = \langle \phi(t), \mathbf{x} \rangle$. So, considering that

$$\frac{d\langle \phi(t), \mathbf{X} \rangle}{dt} = \frac{\langle d\phi(t), \mathbf{X} \rangle}{dt} + \langle \phi(t), A_1 X_1 \otimes X_2 \otimes ... \otimes X_d + X_1 \otimes A_2 X_2 \otimes ... \\ \otimes X_d + ... + X_1 \otimes X_2 \otimes ... \otimes A_d X_d \rangle,$$

we minimize $\langle \phi(t), \mathbf{X} \rangle$, taking into account the order-d tensor or performance criteria, $f_0(t, p(t, \mathbf{X}(t)), u^{(l \to m)}(t, \mathbf{X}(t)))$, and solve the resulting system of ODE's:

$$\begin{aligned} \frac{\langle d\phi(t), \mathbf{X} \rangle}{dt} = -\min_{p(\cdot), u^{(l \to m)}(\cdot)} & [\langle \phi(t), A_1 X_1 \otimes X_2 \otimes ... \otimes X_d \\ & + X_1 \otimes A_2 X_2 \otimes ... \otimes X_d + ... \\ & + X_1 \otimes X_2 \otimes ... \otimes A_d X_d \rangle \\ & + \langle f_0(t, p, u^{(l \to m)}), \mathbf{X} \rangle]. \end{aligned} \tag{4.6}$$

where we use unit vectors $e_i^{(k)}, i = 1, ...N$ for x_k, $k = 1, ..., d$. The actual method of computing this will be considered in Section 4.5.

4.4 Performance criteria

Performance criteria define in what way we want the management policy for the dam system to be optimal. For a problem with many dams we consider four different performance criteria. The first type gives the mean squared difference of the optimal consumption in each joint state and in each dam, and the total customer demand each dam. For a d dam system, we have

$$
\begin{aligned}
J_1(t, p(t, \mathbf{X}(t)), \mathbf{X}(t)) &= \mathbb{E}^p\left[\left(C_1(t, p(t, \mathbf{X}(t)) - \textstyle\sum_{k=1}^n x_{1,k}(t)\right)^2\right] \\
&. \\
&. \\
&. \\
J_d(t, p(t, \mathbf{X}(t)), \mathbf{X}(t)) &= \mathbb{E}^p\left[\left(C_d(t, p(t, \mathbf{X}(t)) - \textstyle\sum_{k=1}^n x_{d,k}(t)\right)^2\right],
\end{aligned} \tag{4.7}
$$

where each of the $J_i, i = 1, ...d$ are tensors of order d.

The second type of performance criteria concerns controlled transfers between dams. We consider the difference squared of the natural inflows and transfers into each dam and the customer demand and evaporation in each dam. For a d dam system the criteria have the form

$$
\begin{aligned}
J_{1+d}(t, u^{(j\to 1)}(t, \mathbf{X}(t)), \mathbf{X}(t)) &= \mathbb{E}^u\Big[\Big(\lambda_1(t) + \textstyle\sum_{j=1}^d u^{(j\to 1)}(t, \mathbf{X}(t)) \\
&\quad - \textstyle\sum_{j=1}^d u^{(1\to j)}(t, \mathbf{X}(t)) - \sum_{k=1}^n x_{1,k}(t) \\
&\quad -\mu_1(t, \mathbf{X}(t))\Big)^2\Big] \\
&. \\
&. \\
&. \\
J_{2d}(t, u^{(j\to d)}(t, \mathbf{X}(t)), \mathbf{X}(t)) &= \mathbb{E}^u\Big[\Big(\lambda_d(t) + \textstyle\sum_{j=1}^d u^{(j\to d)}(t, \mathbf{X}(t)) \\
&\quad - \textstyle\sum_{j=1}^d u^{(d\to j)}(t, \mathbf{X}(t)) - \sum_{k=1}^n x_{d,k}(t) \\
&\quad -\mu_d(t, \mathbf{X}(t))\Big)^2\Big],
\end{aligned} \tag{4.8}
$$

where each of the $J_{i+d}, i = 1, ...d$ is a tensor of order d. The idea here is to maintain balance between inflows and outflows via these transfers. This simple quadratic criteria is just to demonstrate the method but clearly criteria more suited to a particular dam system could be developed. The above criterion is that used in the numerical example. It would have been better to include the controlled consumption in this criterion, however, while using Mathematica to solve this in conjunction with parallel computing techniques, it was not possible to find a method to solve for a criteria with multiple controls. Also, note that we treat water transfer intensities separately in the performance criteria but what we would observe is the difference between these intensities as a single flow intensity. This is explained in section 4.6.

The third type of performance criteria is a single criterion which gives the sum of the average probability that the level of each dam in the system falls below level $M \leq N$ over the interval $[0, T]$:

$$J_{2d+1}(t, p(t, \mathbf{X}(t)), u^{(l \to m)}(t, \mathbf{X}(t)), \mathbf{X}(t)) = \sum_{l=1}^{d} \left(\mathbb{E}^{p,u} \left[\int_0^T \sum_{k=1}^{M} X_{l,k}(s) ds \right] \right). \tag{4.9}$$

As with the other criteria, this is a tensor of order d.

Now we define $f_0(t, p(t, \mathbf{X}(t)), u^{(l \to m)}(t, \mathbf{X}(t)), \mathbf{X}(t)) = \sum_{k=1}^{2d+1} J_k(\cdot)$. This is then used in the dynamic programming equation. A final criterion concerns the terminal state. We consider the sum of the probabilities that the level of each dam is below level $M \leq N$ at time T:

$$J_{2d+2}(t, p(t, \mathbf{X}(t)), u^{(l \to m)}(t, \mathbf{X}(t)), \mathbf{X}(t)) = \sum_{l=1}^{d} \left(\mathbb{E}^{p,u} \left[\sum_{k=1}^{M} X_{l,k}(T) \right] \right). \tag{4.10}$$

This last criterion gives us the terminal conditions for the solution of the system of ODE's given at (4.6).

In this section we consider the control resources of the entire dam system to be unconstrained, however, it is possible to consider the problem with constrained control resources. This type of work has been done by Miller et.al. [35] and uses the Lagrangian approach to find the optimal weighting of each criterion.

4.5 Computational methods

With the results of 4.3.1 and 4.4, we can solve the system numerically. So far, all the numerical work we have done has been using Mathematica 7. Clearly the numerical solution of this problem will be implemented differently in different languages, however, there are some common issues to deal with in any implementation. The first is how to handle expressions like

$$\langle \phi(t), A_1 X_1 \otimes X_2 \otimes ... \otimes X_d + X_1 \otimes A_2 X_2 \otimes ... \otimes X_d + ... + X_1 \otimes X_2 \otimes ... \otimes A_d X_d \rangle,$$

recalling that $\phi(t)$ is a tensor of order d and all of the X_k are unit vectors.

In Mathematica 7 this is handled via the repeated use of a generalized inner product such that

$$\langle \phi(t), A_1 X_1 \otimes X_2 \otimes ... \otimes X_d \rangle = \langle X_d, \langle X_{d-1}, \langle ..., \langle X_2, \langle A_1 X_1, \phi(t) \rangle \rangle ... \rangle \rangle \rangle.$$

This may be an abuse of notation since an inner product gives a scalar, however, the implementation produces the correct result. In general, this calculation would be carried out as follows, where A^T refers to the transpose of a matrix A:

$$X_d \left(X_{d-1} \left(.... \left(X_2 \left(A_1 X_1 \phi(t) \right)^T \right)^T ... \right)^T \right)^T.$$

All other calculations involving the extraction of a the tensor element corresponding to a particular joint state, or operations on a particular state, are handled in the same way.

Another issue is the minimization operation involved in each ODE in the system. In the case we have with the integral performance criteria given, these minimizations can all be carried out prior to solving the ODE system. Essentially we have to minimize over $C_l(t, p(t, \mathbf{X}(t)), \mathbf{X}(t))$ and $u^{(l \to m)}(t, \mathbf{X}(t))$ in (4.6). Due to the particular structure we have, we simply take the partial derivatives with respect to these controls and minimize, since the minimizations will be separable. If the performance criteria were not such nice integral criteria we may have to minimize during the solution of the ODE system, resulting in far more complex calculations.

In the case we have, the actual time of solution of the ODE system is much less than that taken to carry out the minimizations. Since these are done prior to the ODE system solution, they can be done separately in parallel. Mathematica 7 has good support for parallelization and we have parallelized the minimization operations. As yet, this has only been tested on a dual core machine but the results are promising. We have access to a computer cluster and will implement the program on this cluster and report on the results when available. Mathematica has been useful for experimenting and prototyping but in future work we plan to rewrite this program in Fortran or C. The code for the model used in this chapter is included as Appendix A.

4.6 Numerical example

We include a two dam system model as a numerical example of our results so far. Table 4.1 is a list of parameters and functions corresponding to those defined in previous sections and a value K. The values of the performance criteria for the probability of the dams falling below level M during and at the end of the control interval are multiplied by K to make the solutions more sensitive to these criteria. The first subscript refers to the first or second dam as appropriate. The values given for α_1 and α_2 correspond to an allowance of an extra 20% consumption above net natural flows in each dam. The performance criteria (4.7), (4.8) and (4.9) are included in the ODE system definition, while (4.10) is dealt with by the terminal conditions. On a dual-core processor desktop, for three runs the calculations took on average 575.18 CPU seconds in serial and 71.39 CPU seconds with some parallelization, an increase in speed of around eight times.

Figures 4.1 to 4.3 give the type of price structure achieved, noting that this is not a surface but a lattice of prices. It is clear that the slightly higher demand on dam one seems to bias the price structure to behave more responsively to changes in dam one, which is reasonable. The structure is also quite stable, except at the end. With the control objectives largely achieved by $t = 1$, the prices move towards the minimum.

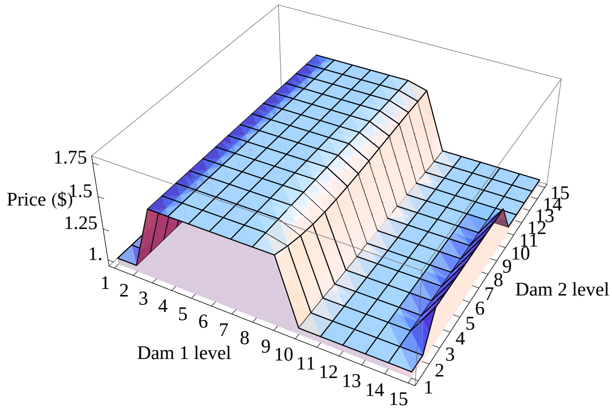

Figure 4.1: Prices at t=0.

A similar result holds for transfers between dams. Figure 4.4 gives the intensity of selected flows between dam one and two. This is defined as $u^{(1\to 2)}(t,\mathbf{x}) - u^{(2\to 1)}(t,\mathbf{x})$ taken at the states specified in figure 4.4. That is, it is the net intensity of flows between the dams. If the intensity is positive, then it is a flow from dam one to two and if negative, from dam two to one. This graph shows that there is a clear bias toward transfers into dam one where the demand is higher. One can also observe the quadratic nature of the performance criteria at work. This may not be realistic but we are simply demonstrating the method at this stage. We will work on better performance criteria in our future research.

Figure 4.5 shows the effects of these controls on the total demand. It shows the total original demand for the system and a weighted average of the total controlled demand. The weighted average is given by

$$\sum_{l=1}^{2}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbb{P}\left\{\text{Dam } l \text{ is in state } [i,j] \text{ at time } t\right\} \times C_l[i,j](t)\right).$$

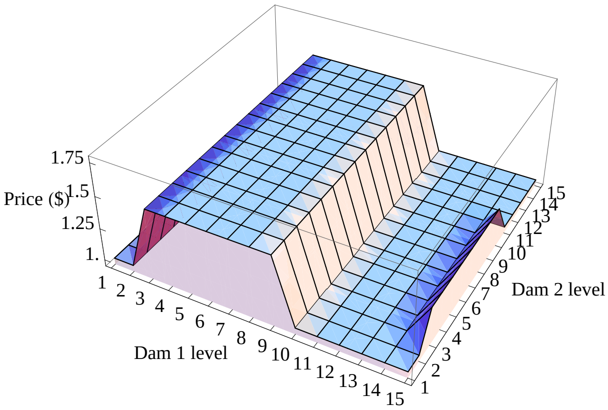

Figure 4.2: Prices at t=0.5.

The probabilities are found by solving the Kolmogorov forward differential equations for each dam given the solution to the system of equations (4.6). It clearly shows that on average there would be a significant reduction in water use in the system using this method of feedback price control.

Table 4.1: Model parameters and functions.

Parameters	Parameters
$N = 15$	$r_1 = r_2 = 0.25$
$M = 5$	$\alpha_1 = 0.91$
$n_1 = 3$	$\alpha_2 = 1.82$
$n_2 = 3$	$K = 150$
$p_{max} = 1.75$	
$p_{min} = 1.00$	
Natural flows	**Demands**
$\lambda_1(t) = \sin(2\pi t) + 10$	$x_{1,1}(t) = \cos(2\pi t) + 4.5$
$\lambda_2(t) = \sin(2\pi t + \frac{\pi}{6}) + 9$	$x_{1,2}(t) = 0.3\cos(2\pi t) + 4.5$
$\mu_{1,N}(t) = -\sin(2\pi t) + 4.5$	$x_{1,3}(t) = 0.5\cos(2\pi t) + 5$
$\mu_{2,N}(t) = -\sin(2\pi t + \frac{\pi}{6}) + 3.5$	$x_{2,1}(t) = \cos(2\pi t) + 5$
	$x_{2,2}(t) = 0.4\cos(2\pi t) + 4$
	$x_{2,3}(t) = 0.3\cos(2\pi t) + 4$

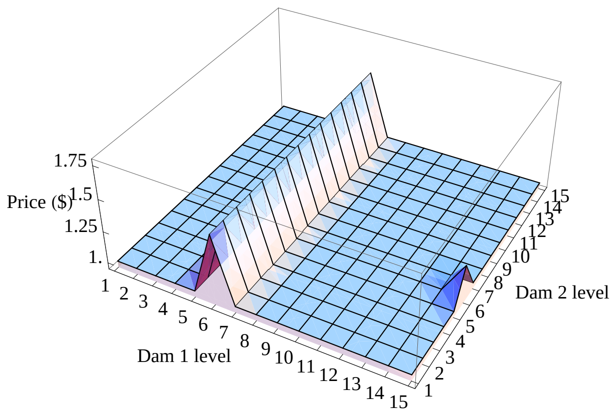

Figure 4.3: Prices at t=1.

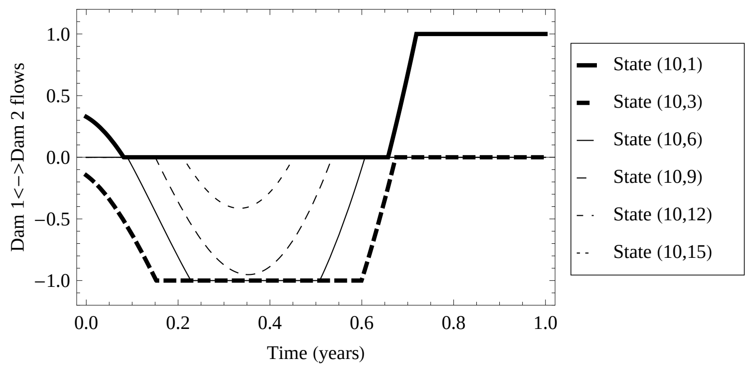

Figure 4.4: Selected flows between dams 1 and 2.

4.7 Further analysis of results

4.7.1 Starting in the lowest joint state

As with the previous models it is instructive to consider how the controls affect the probabilities in extreme states, since these are the states that we are attempting to avoid. So, we take the initial state of the dam system to be essentially empty in both dams, that is, $L(0) = (1, 1)$. If we apply the optimal controls, shown in Figure 4.6, we find that the probability of remaining in this state at the end of the control period falls to approximately 5%. On the other hand, with no controls, as shown in Figure 4.7, the probability falls to a little under 20%. From

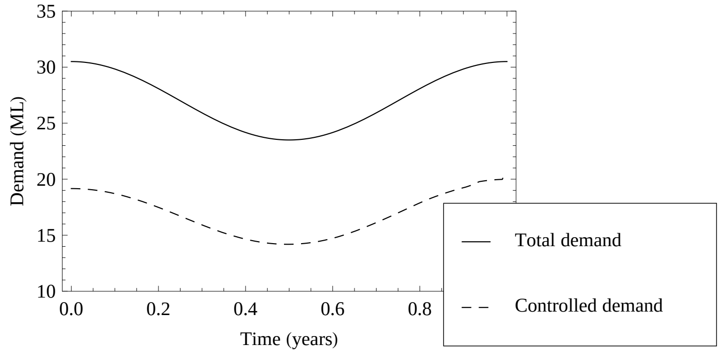

Figure 4.5: Total original demand and controlled demand.

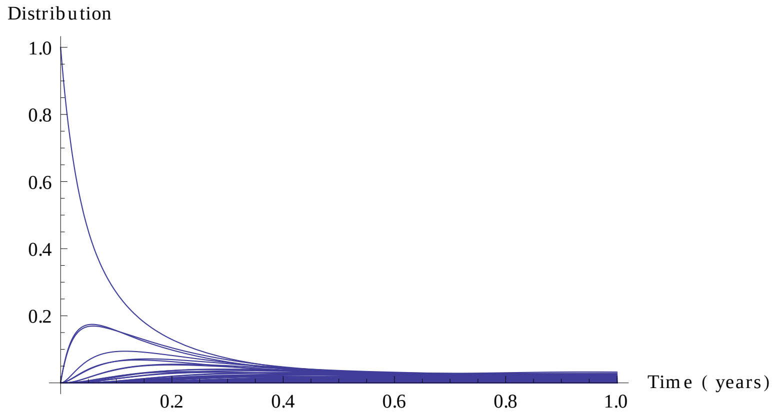

Figure 4.6: State probabilities against time starting in state L(0)=(1,1).

a management perspective there is a clear advantage in applying the optimal control strategy, which for this model is the price control on water.

4.7.2 Starting in the highest joint state

If we start with both dams in the system in the highest state, that is in near overflow conditions, then by applying the optimal controls we can reduce the probability of remaining in this state to under 5%, as shown in Figure 4.8. However, without the price on water, the reduction in the probability is greater, as shown in Figure 4.9. So, from a management point of view it is better not to apply the price control in the highest state, which makes intuitive sense since the customers will be able to use up to their desired level of water. This demonstrates the weakness of this model, in that we did not incorporate any other method to reduce water in the dam apart from actual consumption by customers. This motivated further work on introducing controlled release strategies in conjunction

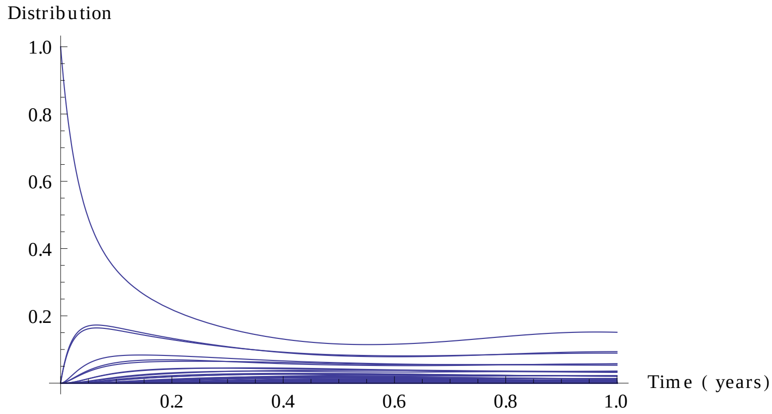

Figure 4.7: State probabilities against time starting in state L(0)=(1,1) with no price control.

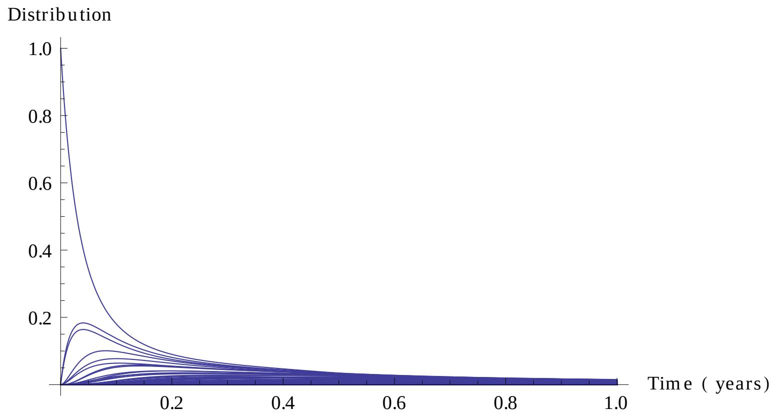

Figure 4.8: State probabilities against time starting in state L(0)=(15,15).

with a price on water so that there would be a balance between water conservation and overflow prevention. This forms part of the subject of the models given in chapters 5 and 6.

4.8 Conclusion

In this chapter we have developed a model for managing water use in a dam system via a dynamic feedback price control. We have shown via a numerical example that the resulting price structure is 'reasonable' in that the prices are generally high when the water level is low and low when high, taking into account the bias toward the dam with greater demand. In this chapter we considered users without connection to a dam suited to their particular needs. More

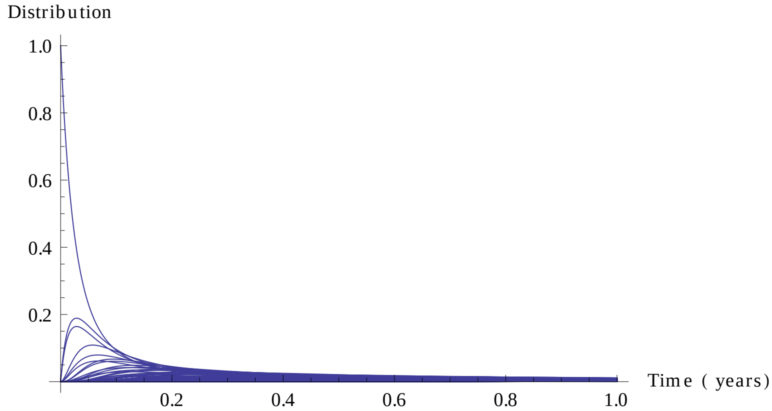

Figure 4.9: State probabilities against time starting in state L(0)=(15,15) with no price control.

realistic is the presence of a relation between a particular customer and a particular dam or even the variable and controllable structure of customer-dam relations.

Following on from this work, we were encouraged to introduce more seasonable variability to the natural inflows and outflows by comments made in conference and seminar presentations. This is the topic of the next chapter.

Chapter 5

Control of a single dam with compound Poisson inflows and provision for flood control

5.1 Introduction

In this chapter we introduce some major innovations to the model outlined in Chapter 2 and [37]. In particular we introduce a time-inhomogeneous inflow process to the dam model as well as a time and state dependent water release control. We also include an overflow state to explicitly take into account the probability of overflow. In section 5.2 we detail the dynamics of the new model and give a proof of the form of the infinitesimal generator of the continuous-time controllable Markov chain (CCMC) which describes the dam process. In section 5.3 we describe how the price control dependent consumption, $C(t, p(t, X))$, is derived. Section 5.4 deals with the solution of the problem via dynamic programming and in Sections 5.5 and 5.6 we give some numerical results.

5.2 Dam model

We assume that we can approximate the level of a large finite dam by discretizing the volume of the dam into $N + 1 \in \mathbb{N}$ levels and denote the level at any time $t \in [0, T]$, $T < \infty$, by an integer valued random variable $L(t) \in \{0, ..., N\}$ [16]. Figure 5.1 gives a stylized depiction of the approximate flow process with the overflow level marked. We designate the level $L(t) = N$ as an overflow or flood state in this model.

If we let each level be represented by the set of unit vectors $S = \{e_0, e_1, ..., e_N\}$ in $\mathbb{R}^{N+1}$, then we can define a random vector $X(t) \in S$ on $[0, T]$ to represent this

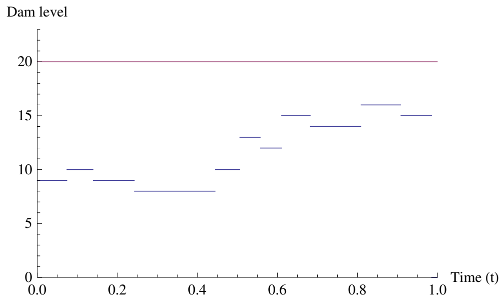

Figure 5.1: Approximate dam level process with overflow level.

level at any time t. Note that this means that $I\{L(t) = i\} = I\{X(t) = e_i\}$ where I is an indicator function. All processes are taken to be defined on the standard probability space, $\{\Omega, \mathscr{F}, \mathbb{P}\}$.

5.2.1 Inflows and outflows

Inflows

We will assume that the process of inflows to the dam can be approximated by a time-inhomogeneous compound Poisson process, $I(t)$. We will further assume that this natural inflow is the result of rain events which arrive randomly according to the time-inhomogeneous counting process $R(t)$ with intensity $\lambda(t)$. The resulting distribution of jumps in the dam level will be given by $Z(t)$. If we designate the maximum jump size as $Z_{max} \in \mathbb{N}$, then $Z(t) \in \{1, 2, 3, ..., Z_{max-1}, Z_{max}\}$, with $\mathbb{P}\{Z(t) = j\} = q_j(t)$ independent of $R(t)$ and the state of the dam. Then, if τ_k is the time of the k^{th} jump, $I(t) = \sum_{k=0}^{R(t)} Z(\tau_k)$.

The semi-martingale representation of $I(t)$ is

$$I(t) = \int_0^t \lambda(s)\mathbb{E}[Z(s)]\, ds + M(t)^{(i)}, \tag{5.1}$$

where $M(t)^{(i)}$ is a square-integrable martingale. The expectation of this process is $\mathbb{E}[I(t)] = \lambda(t)\mathbb{E}[Z(t)]$.

If we consider the jumps of this process, then the size of each jump is given by the random variable $Z(t)$. Relating this to the level of the dam, it is clear that if a jump occurs at time τ when the dam is at level $L(\tau)$ and $L(\tau) + Z(\tau) \geq N - 1$, then the dam is overflowing. It follows that from the perspective of dam dynamics we should include jumps greater than $N-1-L(\tau)$ in an overflow state, N. The problem of how to deal with the overflow will be dealt with in further research because it cannot be dealt with in the probabilistic sense we have here. Applying this we can represent the inflows in the following way, where τ is the jump instant of $I(t)$:

$$I(t) = \sum_{\tau \leq t} Z(\tau). \tag{5.2}$$

Now, $Z(\tau)$ can take the values from $1, ..., Z_{max}$, so we define a sum of indicators of $i \leq Z(\tau)$ and then rewrite (5.2) as

$$I(t) = \sum_{\tau \leq t} \sum_{i=1}^{Z_{max}} I\left\{i = Z(\tau)\right\} i. \tag{5.3}$$

Outflows

Outflows will be controlled explicitly by controlled releases and implicitly via a price on the resource. Outflows from the dam will be assumed to comprise of natural losses due to evaporation, the consumption of the various dam users, controlled water releases as well as overflows if the inflows exceed the dam capacity. We will approximate the natural losses by a general counting process with state dependent intensity, $\mu(t, X(t))$. The consumption will be another counting process which depends on a price control, $p(t, X(t))$, which depends on the current state of the dam, and its intensity will be denoted, $C(t, p(t, X(t)))$. Here $p(t, X(t))$ is taken to be a $\mathscr{F}_t^X$-predictable control in the compact set $[\underline{p}, \overline{p}]$. A state and time dependent controlled counting process with controllable intensity $\nu(t, X(t))$, will represent controlled water releases and is a $\mathscr{F}_t^X$-predictable control in the compact set $[\nu_{min}, \nu_{max}]$, where ν_{min} and ν_{max} are the minimum and maximum release rates respectively. The semi-martingale form of these processes is known and is given by

$$O(t) = \int_0^t (\mu(s, X(s)) + C(s, p(s, X(s))) + \nu(s, X(s)))I\left\{L(s) > 0\right\} ds + M(t)^{(o)},$$

where $M(t)^{(o)}$ is a square-integrable martingale. Here, as with the inflows, we can rewrite the outflow process as

$$O(t) = \sum_{\eta \leq t} I\left\{L(\eta) > 0\right\} \Delta O(\eta), \tag{5.4}$$

where the η are the jump instants for the outflow process and $\Delta O(\eta) = 1$.

5.2.2 Semi-Martingale model of the process $X(t)$

Proposition 5.1. *The infinitesimal generator, $A(t, p(t, X(t)), \nu(t, X(t)))$, of the controllable Markov chain, $X(t)$, has the form*

$$A(t, p(t, X(t)), \nu(t, X(t))) =$$

$$\begin{pmatrix}
-\lambda(t) & \begin{array}{r} C(t,p(t,e_1)) + \mu(t,e_1) \\ +\nu(t,e_1) \end{array} & \ldots & 0 & 0 \\
\lambda(t)q_1(t) & \begin{array}{r} -(\lambda + C(t,p(t,e_1)) \\ +\mu(t,X(t)) + \nu(t,e_1)) \end{array} & \ldots & 0 & 0 \\
\lambda(t)q_2(t) & \lambda(t)q_1(t) & \ldots & 0 & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots \\
\lambda(t)q_{N-2}(t) & \lambda(t)q_{N-3}(t) & \ldots & \begin{array}{r} C(t,p(t,e_N)) + \mu(t,e_N) \\ +\nu(t,e_N) \end{array} & 0 \\
\lambda(t)q_{N-1}(t) & \lambda(t)q_{N-2}(t) & \ldots & \begin{array}{r} -(\lambda(t) + C(t,p(t,e_N)) \\ +\mu(t,e_N) + \nu(t,e_N)) \end{array} & 0 \\
\lambda(t)\sum_{k=N}^{Z_{max}} q_k(t) & \lambda(t)\sum_{k=N-1}^{Z_{max}} q_k(t) & \ldots & \lambda(t) & 0
\end{pmatrix}.$$

Remark 5.2. *In Markov chain theory, this generator matrix conventionally has rows which sum to zero and is usually termed the Q matrix. In control theory the transpose of this matrix is used such that $A = Q^T$, and this is the notation used throughout this paper.*

Remark 5.3. *While termed a controllable Markov chain, in general the resulting process is not Markovian. The controls at time $t < T$ may depend on the entire history of the process up to time t.*

Proof. The following proof an adapted version of a proof given for the form of the infinitesimal generator of a controllable queuing system (see [32] section 3, especially section 3.1 and the proof of Proposition 2). In the original proof the inflows were given as a Poisson type counting process with unit jumps. Here we give the proof for a complex inflow process, given by a time-inhomogeneous compound Poisson process.

Given the inflows and outflows as defined, we can now give the dynamics of the dam level process as $L(t) = I(t) - O(t)$ and the change in the process as $\Delta L(t) = \Delta I(t) - \Delta O(t)$, which simply follows the principle of conservation of mass [44]. Recall that $I\{L(t) = i\} = I\{X(t) = e_i\}$ and then define a vector $f_{k,i}$ such that if $L(t) = k$, then $I\{L(t) + \Delta I(t) = k + i\} = I\{X(t) + f_{k,i} = e_{k+i}\}$. In this case the vector $f_{k,i}$ will have -1 in the k^{th} entry, 1 in the $(k+i)^{th}$ entry and zero for all other entries, that is

$$f_{k,i} = (0, 0, ..., 0, \underbrace{-1}_{k}, 0, ..., 0, \underbrace{1}_{k+i}, 0, ..., 0)^T.$$

We must also consider that $k + i$ may be greater then N. This is the case where an inflow produces an overflow of the dam. In this case the k^{th} entry of $f_{k,i}$ will have a -1 and the $(N+1)^{th}$ entry will have a 1. Now using this $f_{k,i}$, for each i we can define a matrix A_i^+ which captures the effect of the inflow starting in any state k. This matrix is $N + 1 \times N + 1$, the $(N+1)^{th}$ state being the overflow state.

Remark 5.4. *Once in the overflow state the process stops, that is, the $(N+1)^{th}$ state is absorbing. This is shown by every entry of the last column of each of the following matrices being zero. The last column represents the transitions out of the overflow state and it means that there is no possible transition from this state to any other. In this control scheme, control of transition probabilities stops in* *this state and other direct control methods must be used. These are not dealt with in this thesis.*

So for $i = 1$

$$A_1^+ = \begin{pmatrix} -1 & 0 & 0 & ... & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & ... & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & ... & 0 & 0 & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & ... & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & ... & 0 & 0 & 1 & 0 \end{pmatrix}.$$

For $i = 2$

$$A_2^+ = \begin{pmatrix} -1 & 0 & 0 & ... & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & ... & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & ... & 0 & 0 & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & ... & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & ... & 0 & 1 & 1 & 0 \end{pmatrix},$$

for $i = 3$

$$A_3^+ = \begin{pmatrix} -1 & 0 & 0 & ... & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & ... & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & ... & 0 & 0 & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & ... & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & ... & 1 & 1 & 1 & 0 \end{pmatrix},$$

and so on up to $i = Z_{max}$.

Likewise, by similar reasoning and recalling that outflows occur only in unit jumps, we define a matrix A^- which captures the effect of outflows:

$$A^- = \begin{pmatrix} 0 & 1 & 0 & ... & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & ... & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & ... & 0 & 0 & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & ... & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & ... & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The first step is to rewrite the expression $\Delta L(t) = \Delta I(t) - \Delta O(t)$ in terms of the matrices we have defined and the state vector, $X(t)$. The main difference between the proof given in [32] occurs here, since the representation of the compound Poisson process in the form of matrices and state vectors is more complicated. So starting with $\Delta I(t)$, it is clear that if a jump occurs in the process $R(t)$, the rainfall arrival process, then there is a corresponding jump in the inflows, which is random and given by $Z(t)$. Now, taking the family of matrices, A_i^+, $i = 1, ..., Z_{max}$, we define a random matrix, $A_{Z_t}^+$, which takes as its values this family of matrices with $\mathbb{P}\left\{A_{Z(t)}^+ = A_i^+\right\} = q_i(t)$. Let $\Delta \hat{I}(t)$ be the new representation of $\Delta I(t)$, then

$$\Delta \hat{I}(t) = A_{Z(t)}^+ X(t-) \Delta R(t).$$

This essentially captures in matrix form the idea that according to the process $R(t)$, rainfall either occurs or does not, and if it does, then the size of the jump is random and given by $Z(t)$. Likewise, let $\Delta \hat{O}(t)$ be the matrix representation of $\Delta O(t)$, then

$$\Delta \hat{O}(t) = A^- X(t-) \Delta O(t),$$

with a similar interpretation as for $\Delta \hat{I}(t)$.

It follows that $\Delta X(t) = A^+_{Z(t)} X(t-)\Delta R(t) + A^- X(t-)\Delta O(t)$. Now using the relation

$$X(t) = X(0) + \sum_{\tau \le t} \Delta X(\tau),$$

and recalling that $R(t)$ and $O(t)$ are counting processes we obtain

$$\begin{aligned} X(t) &= X(0) + \textstyle\sum_{\tau \le t} [A^+_{Z(\tau)} X(\tau-)\Delta R(\tau) + A^- X(\tau-)\Delta O(\tau)] \\ &= X(0) + \int_0^t A^+_{Z(\tau)} X(\tau-)\, dR(\tau) + \int_0^t A^- X(\tau-)\, dO(\tau). \end{aligned}$$

Now we substitute the semi-martingale representations of the counting processes $R(t)$ and $O(t)$ into the above equation and take the conditional expectation with respect to $\mathscr{F}^X_t$ to obtain

$$\begin{aligned} & X(t) \\ = {} & X(0) + \mathbb{E}\left[\int_0^t A^+_{Z(\tau)} X(\tau-)\lambda(\tau)\, d\tau | \mathscr{F}^X_t\right] + M(t)^R \\ & + \int_0^t A^- X(\tau-)(C(\tau, p(\tau, X(\tau))) + \mu(\tau, X(\tau)) + \nu(\tau, X(\tau)))\, d\tau + M(t)^O \\ = {} & X(0) + \mathbb{E}\left[\mathbb{E}\left[\int_0^t A^+_{Z(\tau)} X(\tau-)\lambda(\tau)\, d\tau | Z(\tau)\right] | \mathscr{F}^X_t\right] + M(t)^R \\ & + \int_0^t A^- X(\tau-)(C(\tau, p(\tau, X(\tau))) + \mu(\tau, X(\tau)) + \nu(\tau, X(\tau)))\, d\tau + M(t)^O \\ = {} & X(0) + \mathbb{E}\left[\int_0^t \sum_{i=1}^{Z_{max}} \mathbb{E}\left[A^+_{Z(\tau)} X(\tau-)\lambda(\tau) | Z(\tau) = i\right] \mathbb{P}\left\{Z(\tau) = i\right\}\, d\tau | \mathscr{F}^X_t\right] + M(t)^R \\ & + \int_0^t A^- X(\tau-)(C(\tau, p(\tau, X(\tau))) + \mu(\tau, X(\tau)) + \nu(\tau, X(\tau)))\, d\tau + M(t)^O \\ = {} & X(0) + \mathbb{E}\left[\int_0^t \sum_{i=1}^{Z_{max}} A^+_i X_{\tau-}\lambda(\tau)\mathbb{P}\left\{Z(\tau) = i\right\}\, d\tau | \mathscr{F}^X_t\right] + M(t)^R \\ & + \int_0^t A^- X(\tau-)(C(\tau, p(\tau, X(\tau))) + \mu(\tau, X(\tau)) + \nu(\tau, X(\tau)))\, d\tau + M(t)^O \\ = {} & X(0) + \int_0^t \left[\sum_{i=1}^{Z_{max}} A^+_i \lambda(\tau)\mathbb{P}\left\{Z(\tau) = i\right\}\right. \\ & \left. + A^-(C(\tau, p(\tau, X(\tau))) + \mu(\tau, X(\tau)) + \nu(\tau, X(\tau)))\right] X(\tau-)\, d\tau + M(t)^X \\ = {} & X(0) + \int_0^t \left[\sum_{i=1}^{Z_{max}} A^+_i \lambda(\tau) q_i(\tau)\right. \\ & \left. + A^-(C(\tau, p(\tau, X(\tau))) + \mu(\tau, X(\tau)) + \nu(\tau, X(\tau)))\right] X(\tau-)\, d\tau + M(t)^X, \end{aligned}$$

where $M(t)^R$ and $M(t)^O$ are the martingales associated with the semi-martingale form of $R(t)$ and $O(t)$ respectively, and $M(t)^X$ is the sum of these. The resulting matrix under the integral given in the last line has exactly the form of that in Proposition 5.1 and this last line is also the known form of the semi-martingale representation of the controlled Markov chain, $X(t)$ [18, 32]. $\square$

5.3 Controlled water use via price control and controllable release

In order to optimally regulate the demand on water from the dam, we want to introduce an optimal price on water for each state of the dam at any time. So the price $p(t, X(t)) \in [p_{min}, p_{max}]$ is given to customers, who then attempt to modify their demand to minimize their costs. Since there will be a different price for each state of the dam, $p(t, X(t))$ is a vector of price functions. It is more convenient to define optimal consumption in terms of the optimal price and then derive equations for optimal consumption and bounds on this consumption based on the optimal price. This was defined in Chapter 2 and is not modified for this model.

5.4 Dynamic programming equation and its solution

Here we outline the method of solution for the optimal controls in a single large dam. This method has been developed in [32] for control of a server router, and in [37], [38] and [39] for single and multiple dams in a system. In this case we will find the optimal solution under control constraints and will give the solution for the single dam. In general, this can be extended to a system of dams by using the methodology developed in [38] and [39] under the same assumptions as for a single dam. Here we restate the method with the inclusion of the flood controls.

We start with the general performance criterion

$$\min_{p(\cdot),\nu(\cdot)} J[p(\cdot), \nu(\cdot)], \tag{5.5}$$

where

$$J[p(\cdot),\nu(\cdot)] = \mathbb{E}^{p,\nu}\left[\phi^0(X(T)) + \int_0^T f_0(s,p(s,X(s)),\nu(s,X(s)),X(s))\,ds\right],$$

and if $\langle\cdot,\cdot\rangle$ is the inner product and $\phi^0 \in \mathbb{R}^{N+1}$, then $\phi^0(X(T)) = \langle\phi^0, X(T)\rangle$, $f_0(s,p(s,X(s)),\nu(s,X(s)),X(s)) = \langle f_0(s,p(s,X(s)),\nu(s,X(s))),X(s)\rangle$.The term $f_0(s,p(s,X(s)),\nu(s,X(s)),X(s))$ is the transition cost of the chain at time s in state $X(s) = e_i$, for $i = 0,...,N$, and so we define

$$f_0^*(s,p(s,X(s)),\nu(s,X(s))) = (f_0(s,p(s,e_0),\nu(s,e_0)),...,f_0(s,p(s,e_N),\nu(s,e_N)))$$

as the vector of transition cost functions of the Markov chain.

Assumption 5.5. *For each i, $i = 0,...,N$ the components of the transition cost function $f_0^*(s,p(s,X(s)),\nu(s,X(s)))$ are continuous on $[0,T]\times[p_{min},p_{max}]\times[0,\nu_{max}]$ and bounded below.*

Next we define the value function

$$V(t,x) = \inf_{p(\cdot),\nu(\cdot)} J[p(\cdot),\nu(\cdot)|X(t)=x] \tag{5.6}$$

where
$J[p(\cdot),\nu(\cdot)|X(t)=x] =$

$$\mathbb{E}^{p,\nu}\left[\phi^0(X(T)) + \int_t^T f_0(s,p(s,X(s)),\nu(s,X(s)),X(s))\,ds|X(t)=x\right]. \tag{5.7}$$

This is interpreted as the cost of transitions from state $X(t) = x$ at time t to state $X(T)$ at the terminal time T. By Assumption 5.5 the performance criterion (5.5) is bounded below, so the infimum in (5.6) exists, and there is a minimizing sequence of controls $\{(p_k(\cdot),\nu_k(\cdot))\}$. Since for each set of the controls $(p_k(\cdot),\nu_k(\cdot))$ we have the function

$$J[p_k(\cdot),\nu_k(\cdot)|X(t)=x] = \langle\hat{\phi}_k(t),x\rangle$$

with continuous $\hat{\phi}_k(t)$, then we can write the function

$$V(t,x) = \lim_k \langle \hat{\phi}_k(t), x \rangle = \langle \hat{\phi}(t), x \rangle,$$

with a measurable column vector-valued function $\hat{\phi}(t) = (\hat{\phi}_0(t), ..., \hat{\phi}_N(t)) \in \mathbb{R}^{N+1}$.

From this assumption we can consider the value function

$$V(t,x) = \inf_{p(\cdot),\nu(\cdot)} J[p(\cdot), \nu(\cdot) | X(t) = x], \tag{5.8}$$

which gives the infimal cost from state $X(t) = x$ at some time $t < T$ to state $X(T)$, and be certain that this infimum exists.

Now, recalling that the state space is made up of the unit vectors in $\mathbb{R}^{N+1}$, where $\phi(t) = (\phi_0(t), \phi_1(t), ..., \phi_N(t))^T \in \mathbb{R}^{N+1}$ is some measurable function giving the cost for each state, then the following is the *dynamic programming equation* with respect to this function $\phi(t)$ in conventional form:

$$\langle \phi'(t), x \rangle + \min_{p,\nu} [\langle \phi(t), A(t,p,\nu)x \rangle + \langle f_0(t,p,\nu), x \rangle] = 0, \tag{5.9}$$

with terminal condition $\phi(T) = \phi^0$ [18, 10]. Since the function

$$H(\phi, t, p, \nu, x) = \langle \phi(t), A(t,p,\nu)x \rangle + \langle f_0(t,p,\nu), x \rangle$$

is continuous for any $(t,p,\nu) \in [0,T] \times \bar{P} \times [0, \nu_{max}]$ and affine in ϕ for any $(t,x) \in [0,T] \times S$, then the function

$$\mathscr{H}(\phi, t, x) = \min_{p,\nu} H(\phi, t, p, \nu, x)$$

is Lipschitz in ϕ with the constant $\mathscr{L} = \max_{(t,p,\nu,x)} \|A(t,p,\nu)x\|$ and continuous in t for any $x \in S$.

Remark 5.6. *By setting $x = e_i$, $i = 0, ..., N$, we get a system of ordinary differential equations*

$$\frac{d\phi^i(t)}{dt} = -\mathscr{H}(t, \phi(t), e_i), \quad i = 0, ..., N, \tag{5.10}$$

with terminal condition $\phi(T) = \phi^0$*. The right-hand side of Equation (5.10) is clearly Lipschitz in* ϕ*.*

Proposition 5.7. *Given Assumption 5.5, equation (5.10) has a unique solution on* $[0, T]$*. This follows from the the fact that the equation is Lipschitz [51].*

The following theorem describes the connection between the value function, $V(t, x)$, and the solution of the system of equations (5.10) as well as some key features of the optimal controls [15, 18, 32].

Remark 5.8. *As long as Proposition 5.7 holds, the following theorem says that* $\phi(t) = \hat{\phi}(t)$*.*

Theorem 5.9. *Let* $\phi(t)$ *be the solution of the system of equations (5.10), then for each* $(t, x) \in [0, T] \times S$ *there exists* $(p_0(t, x), \nu_0(t, x)) \in \bar{P} \times [0, \nu_{max}]$ *such that* $H(t, \phi, p(t, x), \nu(t, x), x)$ *achieves a minimum at* $(p_0(t, x), \nu_0(t, x))$*. Then*

1. *There exist an* $\mathscr{F}_t^X$*-predictable optimal controls,* $(\hat{p}(t, X_0^t), \hat{\nu}(t, X_0^t))$ *such that* $V(t, x) = J[\hat{p}(\cdot), \hat{\nu}(\cdot)|X(t) = x] = \langle \phi(t), x \rangle$*.*
2. *The optimal control can be chosen as Markovian, that is*

$$(\hat{p}(t, X_0^t), \hat{\nu}(t, X_0^t)) = (p_0(t, X(t-), \nu_0(t, X(t-))) =$$
$$\arg\min_{p,\nu} H(t, \phi, p(t, X(t-)), \nu(t, X(t-)), X(t-)).$$

Proof. The proof of the theorem is essentially the same as given in Chapter 2, Section 2.14. □

The system of equations (5.10) can be solved numerically to give the minimal cost to go for each state at any time $t \in [0, T]$, given a chosen terminal state and specific running cost of transitions and controls. From this we can extract the values of the optimal controls for each state and any time on the control interval. While we want to minimize over the control $p(t, X(t))$, in practice we minimize over the controllable consumption, $C(t, p(t, X(t)))$, since we can easily obtain $p(t, X(t))$ from this after solving the system of equations (5.10) explicitly for $C(t, p(t, X(t)))$.

5.4.1 Performance criteria for the dam model

We begin by noting that no controls appear in the generator of this controlled process for states $L(t) = 0$, since the dam is empty in this state, and $L(t) = N$, since the dam is in a state of overflow and we stop control in the probabilistic sense. So we really only have performance criteria for the states $L(t) = 1, ..., N-1$. Our main objective here is to provide customers with their demanded water as far as possible, while at the same time reducing the probability of overflow. This gives two criteria: the first seeks to minimize the square of the difference between what customers demand and what they are actually supplied, and the second seeks to minimize the square of the difference between inflows to the dam and all outflows so as to keep the dam level as stable as possible. Let the customer demand $(1-r)\sum_{i=1}^{n} x_i(t) = \bar{C}(t)$, then the performance criteria are

$$J_1[p(\cdot)] = \mathbb{E}^p \left[\int_0^T \left(C(s, p(s, X(s))) - \bar{C}(s) \right)^2 ds \right]. \tag{5.11}$$

and

$$J_2[p(\cdot), \nu(\cdot)] = \mathbb{E}^{p,\nu} \left[\int_0^T (\lambda(s) - \mu(s, X(s)) - \nu(s, X(s)) - C(s, p(s, X(s))))^2 \, ds \right]. \tag{5.12}$$

and so the vector function of the running cost of control $f_0(t, p(t, X(t)), \nu(t, X(t)), X(t))$ has the following form:

$$\begin{aligned} f_0(t, p(t,x)), \nu(t,x), x \quad = \quad & (\bar{C}(t) - C(t, p(t,x)))^2 \\ & + (\lambda(t) - \mu(t,x) - C(t, p(t,x)) - \nu(t,x))^2, \end{aligned} \tag{5.13}$$

for $x = e_1, ..., e_{N-1}$. The first part of the sum expresses that the difference between what the customer has demanded, less a compulsory reduction of $r \times 100\%$, and what will be supplied should be minimal. The second part of the sum expresses that the overall flows in and out of the dam should be as balanced as possible. We treat the terminal conditions as a performance criteria also by setting the initial conditions of the ODE system such that undesirable states, such as very low or high states, attract a high terminal cost. This is shown in the numerical example.

5.4.2 Form of the optimal controls

To find the form of the optimal controls we find

$$\begin{aligned}\min_{p(\cdot),\nu(\cdot)} H(t,\phi(t),p(t,e_i),\nu(t,e_i),e_i) = \\ &\phi_{i-1}(t)(C(t,p(t,e_i)+\mu(t,e_i)+\nu(t,e_i)) \\ &-\phi_i(t)(\lambda(t)+C(t,p(t,e_i)+\mu(t,e_i)+\nu(t,e_i)) \\ &+\sum_{j=1}^{Z_{max}}\phi_{i+j}(t)q_j(t)+(\bar{C}(t)-C(t,p(t,e_i)))^2 \\ &+(\lambda(t)-\mu(t,e_i)-C(t,p(t,e_i))-\nu(t,e_i))^2\end{aligned} \tag{5.14}$$

for $i = 2, ..., N-1$. In practice we minimize over $C(t,p(t,e_i))$ because we can find the explicit form of this from the solutions of the ODE system and extract $p(t,e_i)$ from this. To minimize we take the partial derivatives of equation (5.14) with respect to $C(t,p(t,e_i))$ and $\nu(t,e_i)$ to get the following system of two linear equations:

$$\begin{aligned} 0 &= \frac{\partial H(t,\hat{\phi}(t),p,\nu,e_i)}{\partial C(t,p(t,e_i))} &&= \phi_{i-1}(t)-\phi_i(t)-2(\bar{C}(t)-C(t,p(t,e_i))) \\ & && -2(\lambda(t)-\mu(t,e_i)-C(t,p(t,e_i))-\nu(t,e_i)) \\ 0 &= \frac{\partial H(t,\hat{\phi}(t),p,\nu,e_i)}{\partial \nu(t,e_i)} &&= \phi_{i-1}(t)-\phi_i(t) \\ & && -2(\lambda(t)-\mu(t,e_i)-C(t,p(t,e_i))-\nu(t,e_i))\end{aligned} \tag{5.15}$$

The equations here come from convex optimization and apply only if the controls that give the stationary points are inside the control set. If so then the partial derivatives are identical except for the term $-2(\bar{C}(t)-C(t,p(t,e_i)))$, which implies that this term must be zero, or $\bar{C}(t) = C(t,p(t,e_i))$. Substituting this into either of the equations in (5.15) gives

$$\nu(t,e_i)) = \frac{\phi_i(t)-\phi_{i-1}(t)}{2}+\lambda(t)-\mu(t,e_i)-\bar{C}(t).$$

Let $C^*(t, e_i)$ and $\nu^*(t, e_i)$ be the form of these minimizing controls, then the solutions are

$$(C^*(t, e_i), \nu^*(t, e_i)) = (\bar{C}(t), \frac{\phi_i(t) - \phi_{i-1}(t)}{2} + \lambda(t) - \mu(t, e_i) - \bar{C}(t)).$$

However, $C(t, p(t, e_i)) \in [C_{min}(t), C_{max}(t)]$ and $\nu(t, e_i) \in [\nu_{min}, \nu_{max}]$, so $C^*(t, e_i)$ and $\nu^*(t, e_i)$ are either within these bounds or they are on the boundary. In the case of $C^*(t, e_i)$, since $\bar{C}(t)$ is known beforehand, this means that for each $t \in [0, T]$ we set $C(t, p(t, e_i))$ as

$$C(t, p(t, e_i)) = \begin{cases} C_{min}(t), & \bar{C}(t) \leq C_{min}(t) \\ \bar{C}(t), & C_{min}(t) < \bar{C}(t) < C_{max}(t) \\ C_{max}(t), & \bar{C}(t) \geq C_{max}(t) \end{cases}$$

The control $\nu^*(t, e_i)$ depends on the solution of the system of ODE's. In this case, when we solve the ODE system numerically, for each $t \in [0, T]$ we set the definition of $\nu(t, e_i)$ to be

$$\nu(t, e_i) = \begin{cases} \nu_{min}, & \nu^*(t, e_i) \leq \nu_{min} \\ \nu^*(t, e_i), & \nu_{min} < \nu^*(t, e_i) < \nu_{max} \\ \nu_{max}, & \nu^*(t, e_i) \geq \nu_{max} \end{cases}$$

In general, this solution is not the solution of the convex optimization problem because the optimal controls may not be inside the control set. The solutions given here will minimize the RHS of equation (5.14). When solving this optimization problem numerically, we solve for

$$(C^*(t, e_i), \nu^*(t, e_i)) = (\bar{C}(t), \frac{\phi_i(t) - \phi_{i-1}(t)}{2} + \lambda(t) - \mu(t, e_i) - \bar{C}(t))$$

for each time $t \in [0, T]$ and then check to see whether or not it is in the control set. If it is not, then we set the control at that time to be the closest boundary point. In the cases we have considered this is straightforward because the control set is rectangular and the edges of the control set are parallel to the axes of the quadratic level set formed by the performance criteria. In more complex cases where the boundary of the control set is convex but curved, all possible combinations of the points on the boundary need to be checked for the minimizing controls. In the numerical example we found that the solutions are often outside the control set and so the realizable solution is taken to be the combination of the controls on the boundary of the control set that minimizes equation (5.14). Clearly this is not the optimal solution but rather the best possible given the set of controls available.

We now give a numerical example with a focus on the probability of overflow.

5.5 Numerical results - flood control only

To demonstrate the form of the minimizing solutions found above we present a simple model of a large dam with properties as described in Table 5.1.

Table 5.1: Model parameters, inflows and demands.

Parameters	Parameters	Inflows & Demands
$N = 20$	$n = 3$	$\lambda(t) = -3\cos(4\pi t) + 7$
$M = 10$	$r = 0$	$\mu_{N+1}(t) = -\sin(2\pi t) + 1$
$p_{max} = 3.00$	$K = 50$	$x_1(t) = \cos(2\pi t) + 5$
$p_{min} = 0$	$\nu_{max} = 20$	$x_2(t) = 0.3\cos(2\pi t) + 4$
$\nu_{min} = 1$		$x_3(t) = 0.5\cos(2\pi t) + 5$

These inflow, evaporation and demand intensities give us asynchronicity of peak supply and demand. The parameter K is a cost placed on the undesirable terminal states and is implemented in the initial conditions of the ODE system in the following way:

$$\phi_i(T) = \begin{cases} \dfrac{M}{(M-1+i)}K & i \leq M \\ 0 & M < i < L-1 \\ K & i = L-1 \\ 2K & i = L \\ 3K & i = L+1 \end{cases}$$

Since the main objective of this chapter is to show how this strategy affects the probability of overflow, we have set the minimum price on water to zero and there is no compulsory reduction in water supplied. This ensures that the customer receives as much water as they demand and that the only difference is the addition of a controlled release to the system. With this scenario we can gain a clear picture of how the controlled release affects the overflow probability.

We now give two cases for a clear demonstration of the effect of the controlled release. The examples are given for the initial state $L(0) = 19$, the state just below the overflow state, which is the state of most interest in this analysis because it is necessary to place the system in an extreme state to test the controls. The probability of overflow is very high in this state. The difference between these cases is the inflow rate. In the first case it is as in Table 5.1, which has inflows being less than outflows across the control period. In the second case the inflows have been increased so that they are much higher than consumption. In this case we would naturally expect the probability of overflow to be higher and for the controlled releases to behave accordingly.

5.5.1 Case 1: inflow rate lower than consumption rate

Figure 5.2(a) gives the optimal release rate compared with the inflow intensity and consumption of dam level 19. The bold line gives the optimal release rate, the faint line the optimal consumption, and the dashed line the inflow. The release rate is roughly proportional to the inflow intensity but the combined effect of inflow and consumption on the rate is clear. Figure 5.2(b)-(d) show the release rate by state for low, medium and high intensity inflows. It is clear that the higher the intensity of inflow the higher the release rate. This is also true of dam level. Both of these observations are consistent with what one would expect and this solution method provides a way to obtain results by this reasonable approximation of the process. Figure 5.3 shows the probability of entering the overflow state, $L(t) = 20$, having started in state $L(0) = 19$, with and without control. With control it is approximately 32% at the end of the control period and almost 44% without control, which shows that the control is effective. The average probability of entering the overflow state is 27% with control versus

35.7% without control.

5.5.2 Case 2: inflow rate higher than consumption rate

For a more extreme example, we changed the inflow intensity $\lambda(t)$ to $\lambda(t) = -3\cos(4\pi t) + 25$, so that the inflow intensity far exceeds that rate of consumption and evaporation. Figure 5.4(a) gives the optimal release rate compared with the inflow intensity and consumption of level 19, with the bold line being the release rate, faint the consumption rate, and dashed the inflow rate as above. Now the release rate is at maximum for almost the entire period, which is the obvious result of such a high inflow rate. Figure 5.4(b)-(d) show the release rate by state for low, medium and high intensity inflows. It is again clear that the higher the intensity of inflow the higher the release rate but in this scenario releases start much earlier and at a higher initial intensity. Figure 5.5 shows the probability of entering the overflow state, $L(t) = 20$, having started in state $L(0) = 19$, with and without control. With control it is approximately 80% at the end of the control period and almost 98% without control, which for such an extreme scenario is highly significant in terms of flood prevention. The average probability of entering the overflow state is 68.2% with control versus 89.7% without control. The 68.2% probability of overflow is still very high, though significantly less than without control. This shows the limitation of a probabilistic control method. More concrete actions are required before reaching such a critical state.

5.6 Numerical results - price and flood control

To demonstrate the form of the optimal price controls we present a simple model of a large dam with properties as described in Table 5.2 and Table 5.3.

The parameters and functions above are for a one year control period for a dam with twenty-one levels. The twenty-first level is the flood level and if the dam actually reaches this level then, as stated before, we allow maximum release and demand. The regulator has set the maximum price as 3.00 (dollars per kilolitre, say) and the minimum is 0.25. Our control strategy requires that we

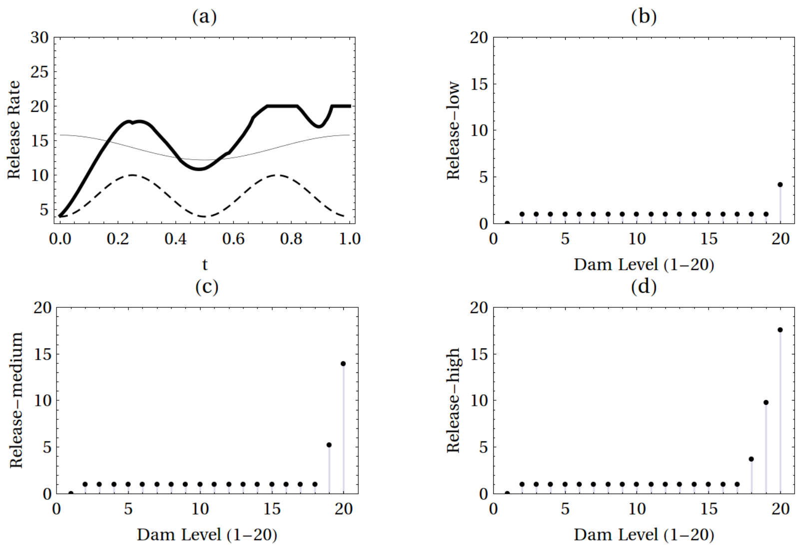

Figure 5.2: (a)Bold-release rate at N=19, thin-consumption at N=19, dashed-inflow, (b) Release rates by state, low intensity inflow (t=0), (c) Release rates by state, medium intensity inflow (t=0.15), (d) Release rates by state, high intensity inflow (t=0.25).

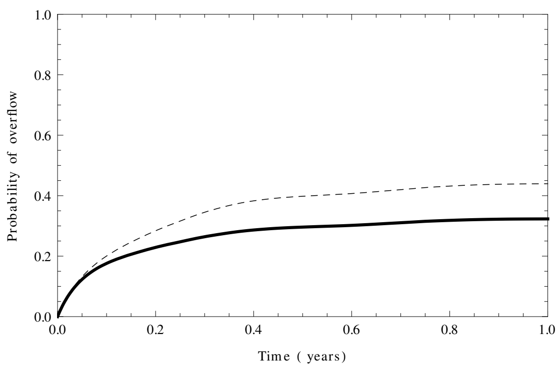

Figure 5.3: Low inflow rate case: probability of entering the overflow state with controlled releases (bold) and without (dashed).

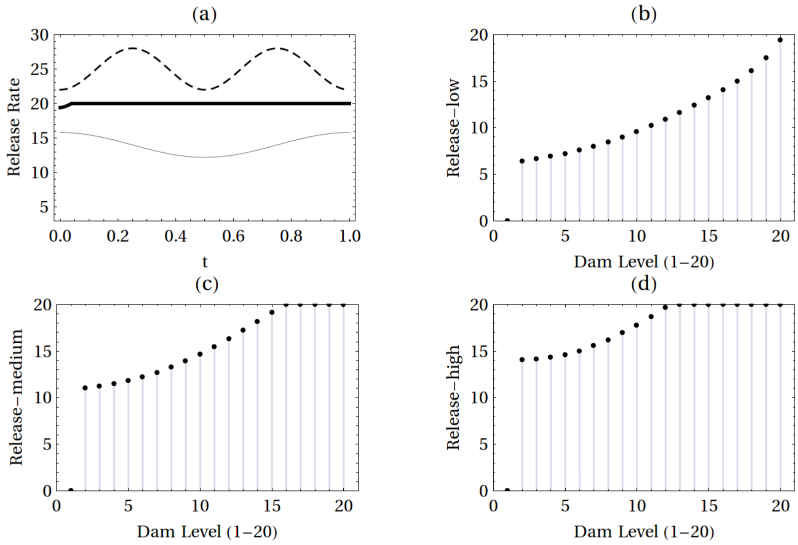

Figure 5.4: (a)Bold-release rate at N=19, thin-consumption at N=19, dashed-inflow, (b) Release rates by state, low intensity inflow (t=0), (c) Release rates by state, medium intensity inflow (t=0.15), (d) Release rates by state, high intensity inflow (t=0.25), with extreme inflows.

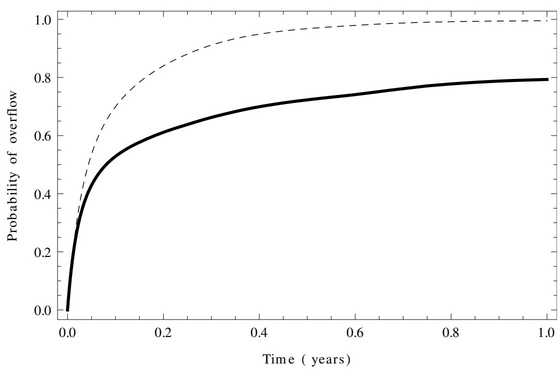

Figure 5.5: High inflow rate case: probability of entering the overflow state with controlled releases (bold) and without (dashed).

Table 5.2: Model parameters.

Parameters	Parameters
$N = 20$	$n = 3$
$M = 7$	$r = 0.1$
$p_{max} = 3.00$	$K = 50$
$p_{min} = 1.5$	$\nu_{max} = 15$
$\nu_{min} = 0$	

Table 5.3: Model functions.

Natural flows & Demands
$\lambda(t) = 3\cos(4\pi t) + 6$
$\mu_{N+1}(t) = -\sin(2\pi t) + 1.5$
$x_1(t) = \cos(2\pi t) + 6.5$
$x_2(t) = 0.3\cos(2\pi t) + 5.5$
$x_3(t) = 0.5\cos(2\pi t) + 5$

reduce consumption by 10% off uncontrolled demand. Natural losses are assumed to be due to evaporation which depends on the level of the dam. For this simple model we have assumed that the evaporation losses decrease linearly as the dam level falls. We have a terminal cost penalty of 100 if the dam level is at or below level $M = 7$ at time $T = 1$ as well as a penalties of 100, 200 and 300 for levels 18, 19 and 20 respectively, taking level 17 as the "safety level". These penalties would be paid by the dam manager to the regulator. We add these costs to the respective states over the entire control period in order to penalize falling into these states, on average. The innovation of this model is the time-inhomogeneous compound Poisson inflow and the probabilities of the different jump sizes, the $q_j(t)$, were derived from rainfall data from the Terang station in Victoria, Australia [43]. There are 104 years of monthly rainfall data for this station (years 1887-2000) and from this a frequency distribution of monthly rainfall was derived. In this model we are allowing a maximum jump of up to 3 levels in the dam due to inflows and we used this rainfall frequency distribution as a proxy for the frequency of jumps sizes in the inflow process. While this is rather crude it serves the purpose of having a seasonal profile for the sizes of jumps in our dam. Figure 5.6 shows the probabilities of the jump sizes by month. It shows clear seasonality and is sufficiently complex to make the inflow process interesting.

The system of ODE's (5.10) was solved using Mathematica 8 on a desktop computer and gave the price structures for the control period, as shown in Figure 5.7. A general feature of this price structure is that it is constant over the control period. This is because our control solution for consumption is to give the customer the maximum possible amount of water to balance the possibility of flood. This leads to the minimum price always being applied. It would be up to the regulator to set this minimum price to give a fixed maximum water usage. This makes sense since we want to maintain the dam at level M but do not want the dam level to rise above about 90% full. This means that a minimum price, set by regulators, is sufficient in the case of there also being a controlled release. The optimal minimum price to maintain the water level is a question for future research.

Heavy penalties were placed on controlled water releases above state $N-2$ since this is wasted water. Figure 5.8 shows the effects of this control for approximately 90% full, 95% full and 100% full compared to inflow intensity. There is a clear relationship between the intensity of inflows and the intensity of controlled releases. In general, as the state of the dam increases, the intensity of controlled releases also rises and has the general shape of the inflow intensity up to ν_{max}. This result is certainly reasonable and is essentially what one would expect to occur.

Finally, Figure 5.9 shows the controlled intensity of demand when the dam is almost empty, 25% full, 50% full, 75% full and 100% full. Again, this shows that prices are affecting consumption mainly when the dam is below the "safety level". In Figure 5.10 we have taken an average of the optimally controlled consumption, weighted by the probability of being in each state, and compared it with the original total demand of the dam users. It clearly shows that the price controls have had the desired effect of reducing overall consumption by moderating the users demands. Moreover, it has the same general shape as the original demand which would be desirable form the customers point of view, since they could adapt to the lower level of supply with roughly the same seasonal characteristics.

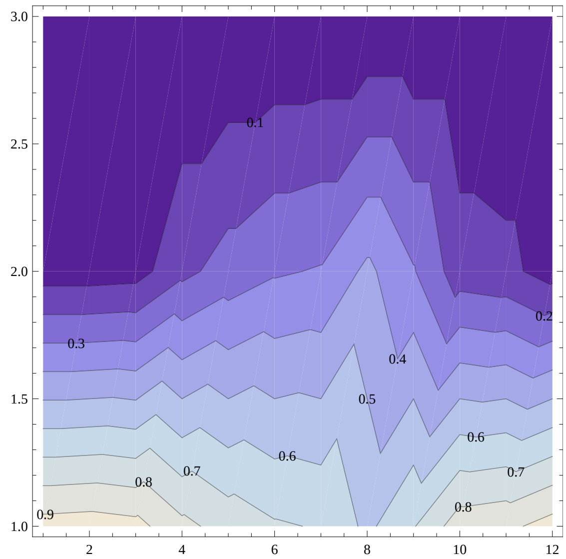

Figure 5.6: Assumed inflow jump density by month and inflow jump size. The horizontal axis gives the month and the vertical the jump size. The contours represent the changing probabilities of the jump sizes through time.

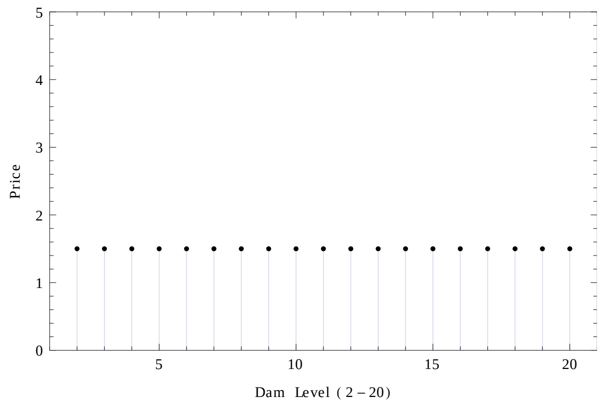

Figure 5.7: Price structure over control interval.

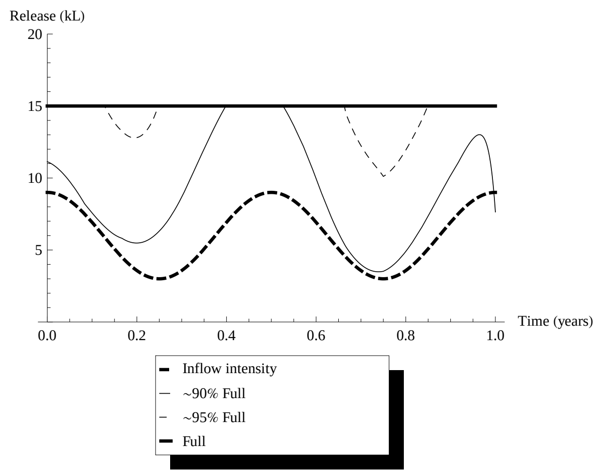

Figure 5.8: Controlled releases at about 90%, 95% and 100% full.

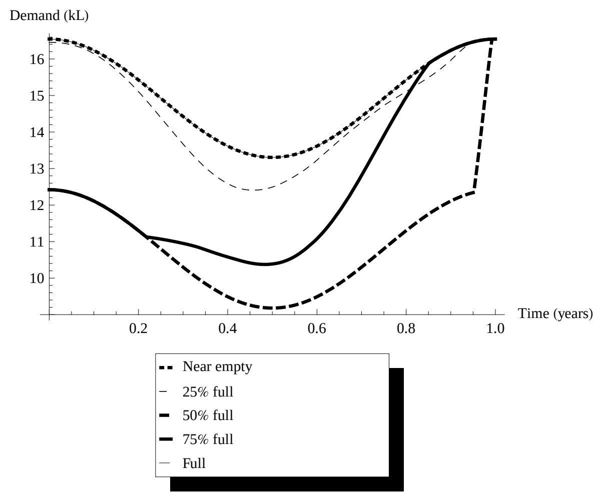

Figure 5.9: Controlled demands for various levels.

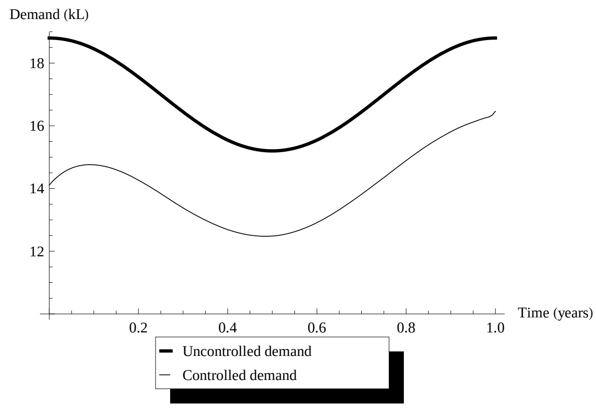

Figure 5.10: Average optimal demand and original demand.

5.7 Further analysis of results

5.7.1 Starting in the lowest state

If we start the system in the second to lowest state, state 1, then we are relying on the price control through moderation of consumption to decrease the probability of remaining in this state. Figure 5.11 shows the result of this control on the state probabilities for states $L(t) = 0$ and $L(t) = 1$. By the end of the control period the probability of remaining in state 1 is of the order of 25% with or without control. On the other hand, the probability of the dam level decreasing to state 0 is of the order of 45% with control and more then 50% without control. Given that this is an indirect control, the result is reasonable, since our performance criteria state that the difference between what the customers demand and what they are supplied should be minimal. It demonstrates that the control works as intended.

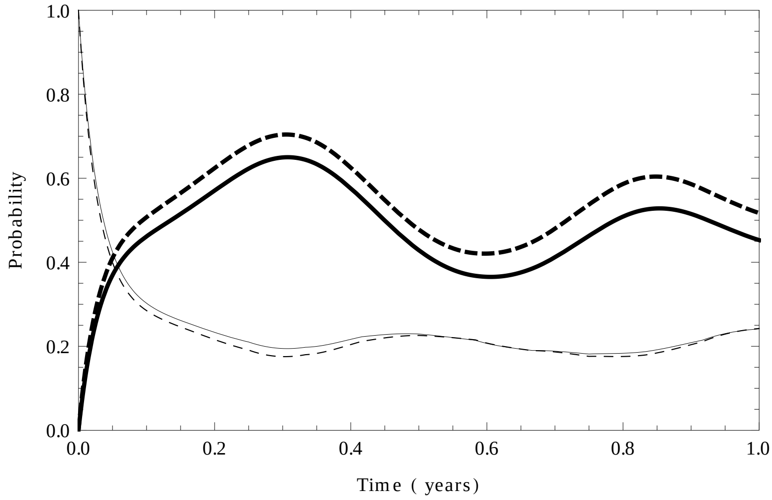

Figure 5.11: Probabilities of states $L(t) = 1$ (with control - thin, without control - thin dashed) and $L(t) = 0$ (with control - bold, without control - bold dashed).

5.7.2 Starting in the highest state

When we start in state 20, the highest state in the numerical model, we find that control is less effective. Figure 5.12 shows that state probabilities for $L(t) = 20$ and $L(t) = 21$. The probability of staying in state 20 drops essentially to zero very quickly, with or without control, while the probability of entering the overflow state heads to 45% without control, but 50% with control. This is readily explained with reference to the performance criteria. In the highest state there is no lack of water but the performance criteria relating to demand, which would try to keep the level high, is still active. This leads to more water being available at a time when we want it to be used. This scenario may actually be better in the case where a price control is used to allow the maximum water allocation to each user. When exactly this should be the case is a question for further research, that is to find the decision point for demand levels that tells us whether to use the price control or not.

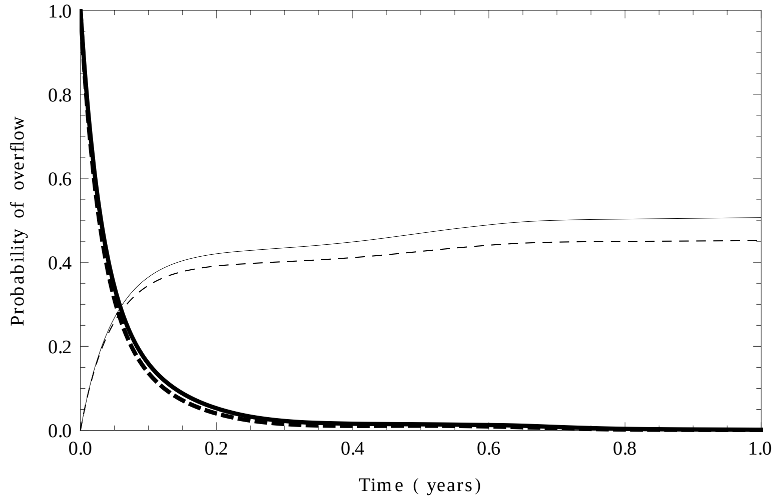

Figure 5.12: Probabilities of states $L(t) = 21$ (with control - thin, without control - thin dashed) and $L(t) = 20$ (with control - bold, without control - bold dashed).

5.8 Conclusion

In this chapter we have given a new model for dam management via optimal price controls and controlled water releases when the dam has a time-inhomogeneous compound Poisson inflow process. We have shown that under some reasonable assumptions about evaporation and jump probabilities we can obtain price structures for the dam at any time and in any state. Furthermore, these price structures result in significant reductions in total demand but do not alter the general seasonal form of the demands on average. We have also shown that a controlled release can help to reduce the risk of reaching a dangerously full state and that we can find both the optimal timing and the optimal amount of water to release. The results that we have obtained give simple forms for the controls and are readily computed even for larger systems with the help of high performance computing (HPC) [39]. They provide a control strategy which is sensible in terms of the controls behaving in a logical way with respect to demand and flood control. In general the controls may be suboptimal depending on the limits of the controls sets. The next innovation will be to use essentially the same model and show how to demonstrate the existence of feasible solutions for the resource constrained problem, which will be done in the next chapter.

Chapter 6

Finding feasible controls for a dam under control resource constraints

In Chapter 5 we considered the control of a dam with time-inhomogeneous inflows without constraints on the performance criteria. This chapter has as its base the model given in chapter 5, however, we introduce constraints on the performance criteria. These constraints result from imposition of controls on resources. Examples of this would include regulated maximum discharge rates or maximum shortfalls in the water delivered to customers, regardless of the physical capacity of the system. The introduction of constraints on the performance criteria leads to the immediate question of whether or not any feasible controls exist, that is controls that meet the constrained performance criteria and are within the bounds of the control set. In other words, we want to know whether a balance can be found between the competing criteria that allows realizable control solutions to be found. This chapter gives the methods to show the consistency of the constraints, which is a criterion for establishing the existence of feasible solutions to the constrained optimal control problem in the case where the problem is a convex optimization problem [9]. Finding the optimal control solution under constraints will not be shown here. To show consistency of the constraints and hence find feasible solutions is a relatively simple numerical procedure, as will be shown, however, the finding of the values for the Lagrange multipliers that give the optimal solution is not trivial. Examples given in [34] and [40] show that convergence to the optimal solution for just two criteria can be problematic and that specialized knowledge of how to apply algorithms in lower level programming languages is needed. However, showing the consistency of constraints is a necessary first step in the constrained optimization procedure and so is given in full detail with respect to dams. The following sections follow the details of [34] and are adapted with specific reference to the dam control problem of Chapter 5.

6.1 The constrained problem as a convex optimization problem

What follows is the general approach to this problem. A specific example is given in section 6.3. As in previous chapters we assume that all processes are defined on the probability space $\{\Omega, \mathscr{F}, \mathbb{P}\}$ and consider the process $\{X(t),\, t \in [0,T]\}$ to be a controlled jump Markov process with piecewise constant right-continuous trajectories, which represents the state of the dam at time t. Now, let the right-continuous set of complete σ-algebras generated by $X(t)$ be denoted

$$\left\{\mathscr{F}_t^X\right\} = \sigma\left\{X(s) :\, s \in [0,T]\right\}.$$

Furthermore, let the controls for this dam be denoted as in Chapter 5 so that $p(t, X(t))$ is the time and state dependent price of water, and $\nu(t, X(t))$ is the time and state dependent water release rate. Let these controls be in some compact metric space $U = [p_{min}, p_{max}] \times [0, \nu_{max}]$. We make the following assumption about the set of admissible controls, which is a convexity condition [34]:

Assumption 6.1. *The set $\bar{U}$ of admissible controls $\{(p(\cdot), \nu(\cdot))\}$ is the set of $\mathscr{F}_t^X$-predictable controls that take values in U. This means that if $N(t)$ is the number of state changes of the process $X(t)$ and X_0^t is the history of the process* *from time zero to time* $t \in [0, T]$ *such that*

$$X_0^t = \{(X(0), 0), (X(\tau_1, \tau_1), ..., (X(\tau_{N_t}), \tau_{N_t})\}$$

is the set of all states and jump times, then for $\tau_{N_t} < t < \tau_{N_t+1}$ *the controls* $(p(t, X_0^t), \nu(t, X_0^t))$ *are functions of the current time* t *and* X_0^t *[18, 13].*

Following the arguments of [34], if Assumption 6.1 holds then the constrained Markov control problem can be written as a convex optimization problem. It follows that for any set of multipliers $\bar{\gamma} = (\gamma_0, \gamma_1, ..., \gamma_M)$ such that ($\gamma_m \geq 0$, $m = 0, ..., M$), where $M + 1$ is the number of criteria, the dynamic programming equation

$$\langle \phi'(t), x \rangle + \min_{p,\nu} [\langle \phi(t), A(t, p, \nu)x \rangle + \langle f_0(t, p, \nu), x \rangle] = 0, \tag{6.1}$$

with terminal condition $\phi(T) = \phi^0$, as given in Chapter 5, can be solved, if only numerically. Certainly for the criteria we have used for dam control, the vector of values of the criteria $\bar{J}(\bar{U}(\cdot)) = (J_0(\bar{U}(\cdot)), ..., J_M(\bar{U}(\cdot)))$ belongs to a bounded convex set in $\mathscr{J} \in \mathbb{R}^{M+1}$, and the solution of (6.1) gives the minimum of the linear form $\mathscr{K}(\bar{\gamma}, \bar{J}) = \langle \bar{\gamma}, \bar{J} \rangle$. So we have

$$\min_{\bar{J} \in \mathscr{J}} \mathscr{K}(\bar{\gamma}, \bar{J}) = \min_{\bar{U}(\cdot)} \langle \bar{\gamma}, \bar{J} \rangle.$$

In this form the constrained optimal control problem can be seen as a finite dimensional convex optimization problem.

6.2 Consistency of constraints

We now consider how to test the consistency of constraints. This means that we need to show that for a system of constraints J_m, for $m = 1, ..., M$, the

$J_m \leq 0$, $\forall m$. Note that γ_0 is the multiplier of the criterion describing the objective function and so is set to zero when considering the consistency of the system of constraints. The following proposition gives a criterion for the consistency of the system of constraints [34].

Proposition 6.2. *Let*

$$\Gamma_0 = \left\{ \gamma : \, \gamma_0 = 0, \, \gamma_1 \geq 0, ..., \gamma_M \geq 0; \, \sum_{m=1}^{M} \gamma_m = 1 \right\}.$$

Then, the system of inequality constraints $J_m(\bar{U}(\cdot)) \leq 0$ is consistent if and only if

$$\max_{\Gamma_0} \min_{\bar{J} \in \mathscr{J}} \mathscr{K}(\bar{\gamma}, \bar{J}) \leq 0. \tag{6.2}$$

Proof. The proof is given here for completeness and closely follows that given in [34]. We begin by assuming that (6.2) holds and so the function

$$\min_{\bar{J} \in \mathscr{J}} \mathscr{K}(\bar{\gamma}, \bar{J})$$

is continuous and concave in $\bar{\gamma}$ [63], and so it achieves a maximum over the compact set Γ_0 at some $\bar{\gamma}^* \in \Gamma_0$. Let $\bar{U}^*$ be the point where the function $\mathscr{K}(\bar{\gamma}^*, \bar{J})$ achieves the minimum over $\mathscr{J}$, then for any $\bar{\gamma} \in \Gamma_0$ we have

$$\langle \bar{\gamma}, \bar{J}(\bar{U}^*) \rangle \leq \langle \bar{\gamma}^*, \bar{J}(\bar{U}^*) \rangle \leq 0,$$

which implies that $J_m(\bar{U}^*) \leq 0$, $\forall \, m = 1, ..., M$.

Now assume that there exists $\bar{U}^*$ such that $J_m(\bar{U}^*) \leq 0$, $\forall \, m = 1, ..., M$, then for any $\bar{\gamma} \in \Gamma_0$

$$\min_{\bar{U}} \langle \bar{\gamma}, \bar{J}(\bar{U}) \rangle \leq \langle \bar{\gamma}, \bar{J}(\bar{U}^*) \rangle \leq 0,$$

and this implies (6.2). □

The next proposition gives a stronger condition for consistency.

Proposition 6.3. *The system of inequality constraints $J_m(\bar{U}(\cdot)) \leq 0$ is strongly consistent and satisfies the Slater condition [9] if and only if*

$$\max_{\Gamma_0} \min_{\bar{J} \in \mathscr{J}} \mathscr{K}(\bar{\gamma}, \bar{J}) < 0. \tag{6.3}$$

This is equivalent to the statement that there exists $\bar{U}^0$ such that $J_m(\bar{U}^0) < 0, \forall\, m = 1, ..., M$ [34, 40].

Proof. Again we follow the proof given in [34]. If (6.3) holds then for any $\bar{\gamma} \in \Gamma_0$ we have

$$\langle \bar{\gamma}, \bar{J}(\bar{U}^*) \rangle \leq \langle \bar{\gamma}^*, \bar{J}(\bar{U}^*) \rangle < 0,$$

which implies that $J_m(\bar{U}^*) < 0, \forall\, m = 1, ..., M$.

Now assume that there exists $\bar{U}^0$ such that $J_m(\bar{U}^0) < 0, \forall\, m = 1, ..., M$, then for any $\bar{\gamma} \in \Gamma_0$

$$\min_{\bar{U}} \langle \bar{\gamma}, \bar{J}(\bar{U}) \rangle \leq \langle \bar{\gamma}, \bar{J}(\bar{U}^0) \rangle \leq \max_{m} \left\{ \bar{J}_m(\bar{U}^0) \right\} \leq C < 0,$$

and this implies (6.3). □

As stated earlier, we do not deal with the procedure for finding the optimal control solutions for the dam problem in Chapter 5 in this thesis, but simply note that the consistency of the constraints implies that an optimal solution exists, based on Section 3.3 of [34] and classical results given in [21] and [29].

6.3 Example showing consistency of constraints

We take as the example the model given in Chapter 5.

6.3.1 Performance criteria

The first criterion is the mean square difference of some fixed proportion of the water demanded by customers and water optimally supplied, representing a measure of how well the customers demanded water needs are being met. This is subject to the square of this difference being less than some amount, say $\alpha > 0$, with α being greater than the minimum possible value of this criterion. Let the customer demand $(1-r)\sum_{i=1}^{n} \bar{x}_i(t) = \bar{C}(t)$, then

$$J_1[p(\cdot)] = \mathbb{E}^p \left\{ \int_0^T \big(C(s, p(s)) - \bar{C}(s)\big)^2 \, ds \right\}, \tag{6.4}$$

with $J_1[p(\cdot)] \leq \alpha$.

The second criterion is the mean square difference between inflows and outflows, and is an expression of the requirement for balance between inflows and outflows in the system. This should be less than some given quantity, say $\beta > 0$, with β greater than the minimum possible value of the criterion:

$J_2[p(\cdot), \nu(\cdot)] =$

$$\mathbb{E}^{p,\nu} \left\{ \int_0^T (\lambda(s) - \mu(s, X(s)) - C(s, p(s, X(s))) - \nu(s, X(s)))^2 ds \right\}, \tag{6.5}$$

with $J_2[p(\cdot), \nu(\cdot)] \leq \beta$.

Clearly the constraints can be rewritten as

$$J_1[p(\cdot)] - \alpha \leq 0, \tag{6.6}$$

and

$$J_2[p(\cdot), \nu(\cdot)] - \beta \leq 0. \tag{6.7}$$

6.3.2 System of ODE's

To find the form of the controls, we set $\gamma_0 = 0$, as in Proposition 6.2 and consider the slightly modified set of differential equations (6.8) from Chapter 5. Here we have only two criteria, so the multiplier for the first will be γ and the second $(1-\gamma)$, such that $\gamma + (1-\gamma) = 1$ as required by Proposition 6.2. We are looking for the value of γ which gives the maximum value of the constrained criteria, so we will set $\gamma = 0$, solve the system and then increment γ by a fixed amount, say $1/m$, and solve again. Continuing this until $\gamma = 1$ will give us $m+1$ solutions of the system for $\gamma \in [0, 1]$. In this way we approximate a continuous function in γ and then look for the maximum. So the required equation is

$$\begin{aligned} \min_{p(\cdot),\nu(\cdot)} H(t, \phi(t), p(t, e_i), \nu(t, e_i), e_i, \gamma) = \\ & \phi_{i-1}(t)(C(t, p(t, e_i) + \mu(t, e_i) + \nu(t, e_i)) \\ & - \phi_i(t)(\lambda(t) + C(t, p(t, e_i) + \mu(t, e_i) \\ & + \nu(t, e_i)) + \sum_{j=1}^{Z_{max}} \phi_{i+j}(t) q_j(t) \\ & + \gamma[(\bar{C}(t) - C(t, p(t, e_i)))^2 - \alpha] \\ & + (1-\gamma)[(\lambda(t) - \mu(t, e_i) - C(t, p(t, e_i)) \\ & - \nu(t, e_i))^2 - \beta] \end{aligned} \tag{6.8}$$

for $i = 1, ..., N-1$. In practice we minimize over $C(t, p(t, e_i))$ because we can find the explicit form of this from the solutions of the ODE system and extract $p(t, e_i)$ from this after. To minimize we take the partial derivatives of equation (6.8) with respect to $C(t, p(t, e_i))$ and $\nu(t, e_i)$ to get the following system of two linear equations:

$$\begin{aligned}
0 &= \frac{\partial H(t,\hat{\phi}(t),p,\nu,e_i,\gamma)}{\partial C(t,p(t,e_i))} &=& \ \phi_{i-1}(t)-\phi_i(t)-2\gamma(\bar{C}(t)-C(t,p(t,e_i))) \\
& & & -2(1-\gamma)(\lambda(t)-\mu(t,e_i)-C(t,p(t,e_i)) \\
& & & -\nu(t,e_i)) \\
0 &= \frac{\partial H(t,\hat{\phi}(t),p,\nu,e_i,\gamma)}{\partial \nu(t,e_i)} &=& \ \phi_{i-1}(t)-\phi_i(t) \\
& & & -2(1-\gamma)(\lambda(t)-\mu(t,e_i)-C(t,p(t,e_i)) \\
& & & -\nu(t,e_i))
\end{aligned} \tag{6.9}$$

As in Chapter 5, these equations are from the convex optimization procedure and apply only if the controls at the stationary points are inside the control set. If they are then the partial derivatives are identical except for the term $-2\gamma(\bar{C}(t)-C(t,p(t,e_i)))$, which implies that this term must be zero, or $\bar{C}(t) = C(t,p(t,e_i))$. Substituting this into either of the above gives

$$\nu(t,e_i)) = \frac{\phi_i(t)-\phi_{i-1}(t)}{2(1-\gamma)} + \lambda(t) - \mu(t,e_i) - \bar{C}(t).$$

Let $C^*(t,e_i)$ and $\nu^*(t,e_i)$ be the form of these minimizing controls, then the solutions are

$$(C^*(t,e_i),\nu^*(t,e_i)) = (\bar{C}(t), \frac{\phi_i(t)-\phi_{i-1}(t)}{2(1-\gamma)} + \lambda(t) - \mu(t,e_i) - \bar{C}(t)).$$

However, $C(t,p(t,e_i)) \in [C_{min}(t), C_{max}(t)]$ and $\nu(t,e_i) \in [\nu_{min}, \nu_{max}]$, so $C^*(t,e_i)$ and $\nu^*(t,e_i)$ are either within these bounds or they are on the boundary. In the case of $C^*(t,e_i)$, since $\bar{C}(t)$ is known beforehand, this means that for each

$t \in [0, T]$ we set $C(t, p(t, e_i))$ as

$$C(t, p(t, e_i)) = \begin{cases} C_{min}(t), & \bar{C}(t) \leq C_{min}(t) \\ \bar{C}(t), & C_{min}(t) < \bar{C}(t) < C_{max}(t) \\ C_{max}(t), & \bar{C}(t) \geq C_{max}(t) \end{cases}$$

The control $\nu^*(t, e_i)$ depends on the solution of the system of ODE's. In this case, when we solve the ODE system numerically, for each $t \in [0, T]$ we set the definition of $\nu(t, e_i)$ to be

$$\nu(t, e_i) = \begin{cases} \nu_{min}, & \nu^*(t, e_i) \leq \nu_{min} \\ \nu^*(t, e_i), & \nu_{min} < \nu^*(t, e_i) < \nu_{max} \\ \nu_{max}, & \nu^*(t, e_i) \geq \nu_{max} \end{cases}$$

As in chapter 5, these solutions are in general not the solutions of the convex optimization problem. They will minimize the RHS of equation (6.8) for the allowable set of controls but will not provide the optimal solution in an absolute sense. As stated above, the convex optimization procedure only applies when the controls at the stationary points are inside the control set. In many cases this is not the case and so we have to check at each time $t \in [0, T]$ to see whether this is the case and, if not, check combinations of controls on the boundary of the control set. Our set of controls is rectangular and the axes of the level set formed by the performace criteria are parallel to its edges, so this procedure is straightforward. At each time t we solve for the controls using the convex optimization procedure, test to see whether the resulting control is inside the control set and, if not, set the control to have the closest boundary value. The above cases give the definitions of the controls for each γ in the numerical solution.

6.3.3 Numerical example

For this example we have taken the increment of γ as 1/100 so that we end up with 101 solutions for the system (6.8). The model has the following equations and parameters as inputs taken directly from Chapter 5:

Table 6.1: Model parameters, inflows and demands.

Parameters	Parameters	Inflows & Demands
$N = 20$	$n = 3$	$\lambda(t) = -3\cos(4\pi t) + 7$
$M = 10$	$r = 0.1$	$\mu_{N+1}(t) = -\sin(2\pi t) + 1$
$p_{max} = 3.00$	$K = 50$	$x_1(t) = \cos(2\pi t) + 5$
$p_{min} = 1.5$	$\nu_{max} = 15$	$x_2(t) = 0.3\cos(2\pi t) + 4$
$\nu_{min} = 1$		$x_3(t) = 0.5\cos(2\pi t) + 5$

Figure 6.1 shows the value of the constrained performance criteria against γ for $\alpha = 4$ and $\beta = 100$ at time $t = 0$. It clearly shows that the values of the criteria are greater than zero for all γ and so the constraints are inconsistent in this case. If we change the value of β to $\beta = 225$, while keeping $\alpha = 4$ constant, we get Figure 6.2. This shows that the values of the criteria are less than zero for all γ and so the constraints are consistent in this case. Because of the relatively small value of α, changing it makes little difference to the overall picture. Figure 6.3 shows that the values of the criteria are still mostly above zero, which means that we should constrict our search for controls to the set of values of γ where the values of the weighted mixed criteria are less than zero.

6.4 Concluding remarks

In this chapter we have given the theoretical background and the numerical procedure for testing a system of inequality constraints for consistency. We have used a very simple case here with just two criteria to demonstrate the method. It would clearly be more difficult with more criteria, requiring more complicated

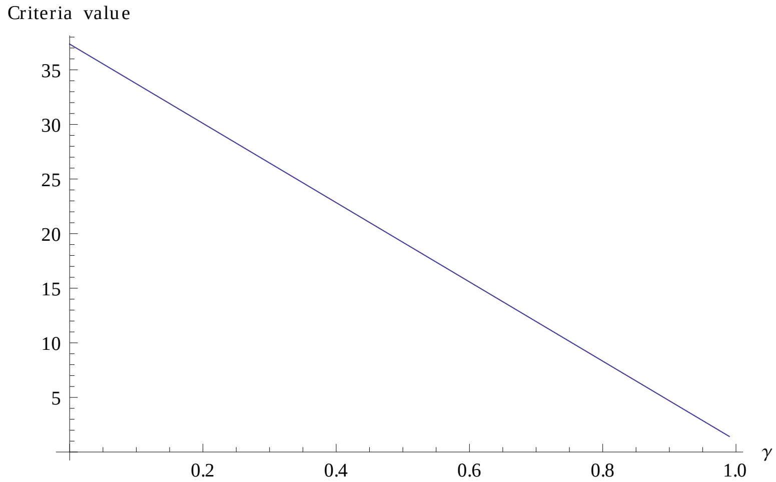

Figure 6.1: Criteria value as a function of γ with $\alpha = 4$ and $\beta = 100$.

procedures to find the maximum than shown here. Also, we have used the example of a single dam with only 21 levels, so the solution for each step of γ was quite fast. With connected systems or large numbers of levels this would be a time consuming calculation. In the next chapter we describe approaches for increasing the speed of computation of the solutions of these models.

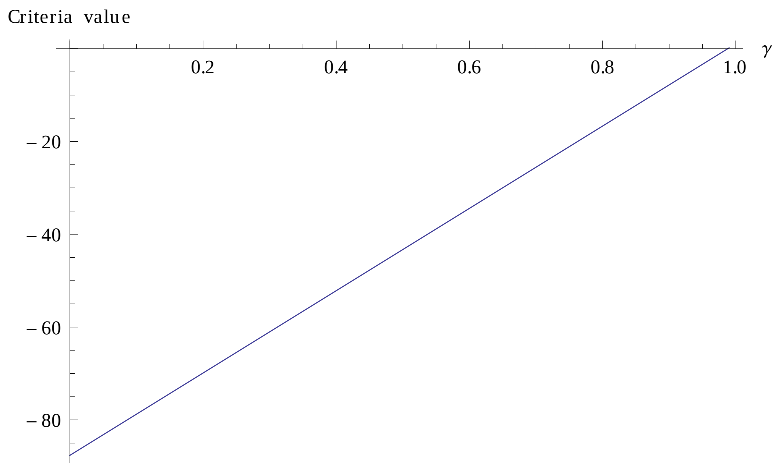

Figure 6.2: Criteria value as a function of γ with $\alpha = 4$ and $\beta = 225$.

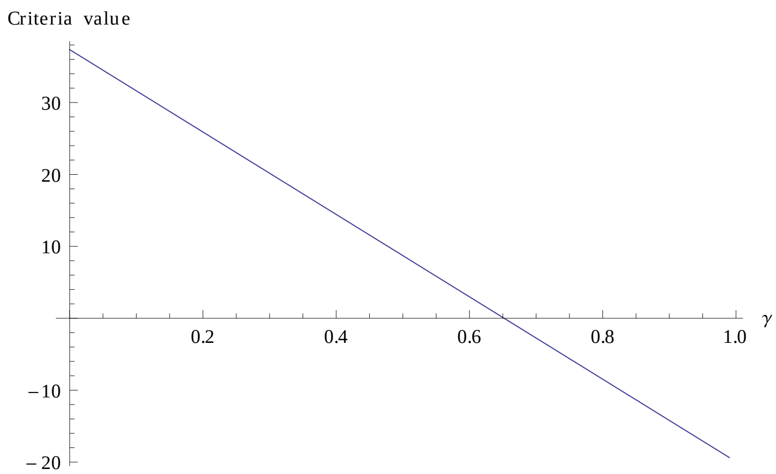

Figure 6.3: Criteria value as a function of γ with $\alpha = 25$ and $\beta = 100$.

Chapter 7

Computational aspects including the use of parallel and high performance computing

The contents of this chapter were presented at the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC) held in Orlando, Florida, USA, between December 12-15,2011. The purpose of this chapter is to outline the numerical procedures which we have used to solve the problem of two connected dams. The reason this model was chosen is that with connected systems the number of differential equations to be simultaneously solved becomes large very quickly, even when the number of states in the individual parts of the system are quite small. In general, if we have N levels in each part of the system and we connect d different parts, then we will generate a system of N^d differential equations. So, whereas for a single dam 100 levels may provide a quite fine discrete approximation, for two connected dams with 100 levels we have 10000 differential equations to solve simultaneously. These differential equations are non-linear in general and the computational resources required to numerically solve the system is high. As the computational difficulty of problems with high dimensionality is a common criticism of the dynamic programming solution method, this is an important area to examine.

7.1 Programming languages

From the beginning of this project all models have been written and solved in Mathematica 7 or 8. As a high level scripting language there is a general consensus that it will not perform as well as lower level languages such as C, C++ or FORTRAN in terms of speed, however, with limited programming skills and the ability to do quite complicated calculations with minimal coding it has provided a good environment in which to test ideas. The performance difference with other programming languages has not been tested and we make no pretensions as to this being the best possible result, indeed the performance we have achieved using Mathematica may likely be improved by writing the code in a lower level language. Even so, the techniques we have employed are available in all of these languages and so our solution methods could be readily implemented in them.

7.2 Computational aspects considered

In this chapter we focus on two aspects in particular that have improved the speed of numerical solutions. The first is parallelization of parts of the code. The motivation for trying this was advice and training received during attendance at Super Computing 2010 (SC10) in New Orleans, USA, between 13-19 November, 2010. At a poster presentation many helpful comments were received about the possibility of using parallel computing and a workshop provided all the basic knowledge to try and implement it. This was in the context of programming with C or C++ but with Mathematica it was discovered that most of the parallelization could be done automatically with only a few changes in the code. This is outlined in this chapter.

The second aspect is the use of High Performance Computing (HPC), which was the special focus of the conference in New Orleans. This area of computing has seen a revolution in terms of the size of problems that can be solved in a reasonable time frame. We have compared solution times with a desktop PC and with the Monash University HPC Center's Monash Sun Grid (MSG). We show that the use of HPC dramatically improves the speed of solution for dam models with a large number of states, especially systems of connected dams. More importantly, these larger problems cannot be solved on PC desktops due to memory problems and so HPC is the only viable platform available. In this chapter we detail some of the experiments we have done with both with and without parallelization on the MSG.

7.3 The 2-dam model

This model is covered in depth in Chapter 4, so here we give a brief account of the model. We begin by making some simplifying assumptions about the dams. First we assume that each dam has independent natural inflow and outflow processes. Secondly we assume that the consumption in each dam depends on a time and joint-state dependent price. We likewise assume that water transfers between dams depend on time and the joint-state of the dams.

We approximate the level in each dam by discretizing it into $N+1$ levels or states, $N<\infty$, and let $L_i(t) \in \{0, ..., N\}$, $i = 1, 2$ be an integer valued random variable describing the state of dam i at time t. The martingale approach in [18] allows us to describe the $N+1$ states by the unit vectors in $\mathbb{R}^{N+1}$, giving $S_i = \{e_0^i, ..., e_N^i\}$ as the set of unit vectors for dam i.

Define $X_i(t)$, $i = 1,2$, where $\{X_i(t) \in S_i, t \in [0,T]\}$ for $T < \infty$ on the probability space $\{\Omega, \mathscr{F}, \mathbb{P}\}$, as a controlled jump Markov process with piecewise right-continuous paths. This process is for the change in level of dam i. We make the following assumptions about the price control, $p(t, \mathbf{X}(t))$ and the transfer control, $u^{(i \to j)}(t, \mathbf{X}(t))$, between dams i and j, $i,j = 1,2$ and $i \neq j$.

Assumption 7.1. *Assume that the set of admissible controls,* $\bar{P} = p(\cdot)$ *and* $\bar{U} = \left\{u^{(i \to j)}(\cdot) : i,j = 1,2; i \neq j\right\}$ *are sets of* $\mathscr{F}_t^{\mathbf{X}}$*-measurable controls taking values in* $P = \{p \in [p_{min}, p_{max}]\}$ *and* $U = \{u \in [0,1]\}$ *respectively, where* $\mathbf{X} = X_1 \otimes X_2$*.*

Remark 7.2. *If the history of the jump process from time 0 to t is denoted* $\mathbf{X}_0^t = (X_1)_0^t \otimes (X_2)_0^t$*, then assumption 7.1 ensures that our controls,* $p(t, \mathbf{X}_0^t)$ *and* $u_{i,j}(t, \mathbf{X}_0^t)$ *are measurable with respect to t and* $\mathbf{X}_0^t$ *(for detail see [38]).*

7.3.1 System dynamics

For this model we assume that we can approximate the inflow and outflow processes of each dam by general $\mathscr{F}_t^{\mathbf{X}}$-measurable counting processes with unit jumps, $Y_{in}^i(t)$ and $Y_{out}^i(t)$, $i, = 1,2$, respectively. The intensity of inflows comes from the deterministic intensity of natural inflows, $\lambda_i(t)$, and the intensity of inflows from the other dam, $u_{j,i}(t)$. The intensity of outflows comes from the deterministic intensity of evaporation, $\mu_i(t)$, the intensity of consumption, $C_i(t) = C_i(t, p(t), \mathbf{X}_t)$ and the intensity of transfers to the other dam, $u_{i,j}(t)$. This gives us two processes,

$$Y_{in}^{(i)}(t) = \int_0^t (\lambda_i(s) + \sum_{j=1}^{d} u^{(j \to i)}(s)) I\left\{L_i(s) < N\right\} ds + M_{in}^{(i)}(t),$$

where $M_{in}^{(i)}(t)$ is a square integrable martingale with quadratic variation

$$\langle M_{in}^{(i)} \rangle_t = \int_0^t (\lambda_i(s) + \sum_{j=1}^{d} u^{(j \to i)}(s)) I \left\{ L_i(s) < N \right\} ds$$

and

$$Y_{out}^{(i)}(t) = \int_0^t (\mu_i(s, \mathbf{X}(s)) + C_i(s, p(s, \mathbf{X}(s))) + \sum_{j=1}^{d} u^{(i \to j)}(s, \mathbf{X}(s))) I \left\{ L_i(s) > 0 \right\} ds$$
$$+ M_{out}^{(i)}(t),$$

where $M_{in}^i(t)$ and $M_{out}^i(t)$ are square integrable martingales. So now the approximate dynamics for each dam in our model are given by

$$L_i(t) = Y_{in}^i(t) - Y_{out}^i(t).$$

For a more detailed explanation of these approximations please see [38].

7.3.2 Dam system as a system of connected controlled Markov chains

With the approximations in 7.3.1, recall that the infinitesimal generator of each component of the dam system has the form given in Chapter 4, Section 4.1.1:

$$A_i(t,p(t,\mathbf{X}(t)),u^{(i\to j)}(t,\mathbf{X}(t)),u^{(j\to i)}(t,\mathbf{X}(t)))=$$

$$\begin{pmatrix}
\begin{array}{l}-\lambda_i(t)\\ -\sum\limits_{j\neq i}u_i^{(j\to i)}(t,\mathbf{X}(t))\end{array} & \begin{array}{l}C_i(t,p(t,\mathbf{X}(t)))\\ +\mu_i(t,\mathbf{X}(t))\\ +\sum\limits_{j\neq i}u_i^{(i\to j)}(t,\mathbf{X}(t))\end{array} & \cdots & 0 & 0\\
\begin{array}{l}\lambda_i(t)\\ +\sum\limits_{j\neq i}u_i^{(j\to i)}(t,\mathbf{X}(t))\end{array} & \begin{array}{l}-(C_i(t,p(t,\mathbf{X}(t)))\\ +\mu_i(t,\mathbf{X}(t))\\ +\sum\limits_{j\neq i}u_i^{(i\to j)}(t,\mathbf{X}(t))\\ +\lambda_i(t)\\ +\sum\limits_{j\neq i}u_i^{(j\to i)}(t,\mathbf{X}(t)))\end{array} & \cdots & 0 & 0\\
0 & \begin{array}{l}\lambda_i(t)\\ +\sum\limits_{j\neq i}u_i^{(j\to i)}(t,\mathbf{X}(t))\end{array} & \cdots & 0 & 0\\
\cdots & \cdots & \cdots & \cdots & \cdots\\
0 & 0 & \cdots & \begin{array}{l}C_i(t,p(t,\mathbf{X}(t)))\\ +\mu_i(t,\mathbf{X}(t))\\ +\sum\limits_{j\neq i}u_i^{(i\to j)}(t,\mathbf{X}(t))\end{array} & 0\\
0 & 0 & \cdots & \begin{array}{l}-(C_i(t,p(t,\mathbf{X}(t)))\\ +\mu_i(t,\mathbf{X}(t))\\ +\sum\limits_{j\neq i}u_i^{(i\to j)}(t,\mathbf{X}(t))\\ +\lambda_i(t)\\ +\sum\limits_{j\neq i}u_i^{(j\to i)}(t,\mathbf{X}(t)))\end{array} & \begin{array}{l}C_i(t,p(t,\mathbf{X}(t)))\\ +\mu_i(t,\mathbf{X}(t))\\ +\sum\limits_{j\neq i}u_i^{(i\to j)}(t,\mathbf{X}(t))\end{array}\\
0 & 0 & \cdots & \begin{array}{l}\lambda_i(t)\\ +\sum\limits_{j\neq i}u_i^{(j\to i)}(t,\mathbf{X}(t))\end{array} & \begin{array}{l}-(C_i(t,p(t,\mathbf{X}(t)))\\ +\mu_i(t,\mathbf{X}(t))\\ +\sum\limits_{j\neq i}u_i^{(i\to j)}(t,\mathbf{X}(t)))\end{array}
\end{pmatrix}.$$

The column number corresponds to the current state of the i^{th} dam and the column entries add to zero. The two generators for the system proposed here are clearly connected through transfers and the definitions of the controls being dependent on the joint state of the system.

7.4 Derivation of controlled demand functions

The key innovation of this model is the use of a time and state dependent feedback control, $p(t, \mathbf{X}(t))$, to take into account the active seasonal demands of consumers. It easier to find $p(t, \mathbf{X}(t))$ through the effect it has on consumption in each dam. For the i^{th} dam, the resulting controlled demand is denoted $C_i(t, p(t, \mathbf{X}(t)))$. Here we show how we take the price of water into account through controlled consumption. We are looking for a single price structure for all users of the *dam system*. This is not a necessary element of the model and different pricing structures could apply to different components of the system or even different sectors. In this case the number of price controls would need to be increased.

So, considering the i^{th} dam, let there be n sectors, or consumers, each with their own seasonal demand intensity, $\bar{x}_{i,k}(t)$, for $k = 1, ..., n$, recall that the optimal consumption for each dam is defined as in Section 4.2, such that for the i^{th} dam, the total optimal intensity of demand is

$$C_i(t, p(t, \mathbf{X}(t))) = \sum_{k=1}^{n} x_{i,k}(t, p(t, \mathbf{X}(t))). \tag{7.1}$$

and, since we also know that $p(t, \mathbf{X}(t)) \in [p_{min}, p_{max}]$:

$$\begin{aligned} C_{i,max}(t) = \sum_{k=1}^{n} x_{i,k}(t, p_{min}) \geq C_i(t, p(t, \mathbf{X}(t))) = \sum_{k=1}^{n} x_{i,k}(t, p(t, \mathbf{X}(t))) \\ \geq C_{i,min}(t) = \sum_{k=1}^{n} x_{i,k}(t, p_{max}). \end{aligned} \tag{7.2}$$

These equations allow us to define piecewise functions for the solution of $C_i(t, p(t, \mathbf{X}(t)))$ in the dynamic programming equations.

7.5 Dynamic programming and optimal control

The method of solution follows that given in Chapter 4, Section 4.3.1, with the following performance criteria:

7.5.1 Performance criteria

For this example with two dams we have four types of performance criteria. The first type seeks to minimize the difference between the customer's seasonal demand intensity and the optimal demand intensity:

$$J_1(t, p(t, \mathbf{X}(t)), \mathbf{X}(t)) = \mathbb{E}^p \left[\left(C_1(t, p(t, \mathbf{X}(t)) - \sum_{k=1}^{n} x_{1,k}(t) \right)^2 \right],$$

and

$$J_2(t, p(t, \mathbf{X}(t)), \mathbf{X}(t)) = \mathbb{E}^p \left[\left(C_2(t, p(t, \mathbf{X}(t)) - \sum_{k=1}^{n} x_{2,k}(t) \right)^2 \right].$$

The second type considers the difference squared of the natural inflows and transfers into each dam and the customer demand and evaporation in each dam:

$$\begin{aligned} J_3(t, u^{(i \to j)}(t, \mathbf{X}(t)), \mathbf{X}(t)) = &\mathbb{E}^u \big[\big(\lambda_1(t) + u^{(2 \to 1)}(t, \mathbf{X}(t)) \\ &- u^{(1 \to 2)}(t, \mathbf{X}(t)) - \sum_{k=1}^{n} x_{1,k}(t) \\ &- \mu_1(t, \mathbf{X}(t)) \big)^2 \big] \end{aligned}$$

and

$$\begin{aligned} J_4(t, u^{(i \to j)}(t, \mathbf{X}(t)), \mathbf{X}(t)) =& \mathbb{E}^u \left[\left(\lambda_2(t) + u^{(1 \to 2)}(t, \mathbf{X}(t)) \right. \right. \\ & -u^{(2 \to 1)}(t, \mathbf{X}(t)) - \sum_{k=1}^{n} x_{2,k}(t) \\ & \left. \left. -\mu_2(t, \mathbf{X}(t)) \right)^2 \right]. \end{aligned}$$

The third type is to minimize the probability that on average either dam falls below level M:

$$J_5(t, \mathbf{X}(t)) = \sum_{l=1}^{2} \left(\mathbb{E}^{p,u} \left[\int_0^T \sum_{k=1}^{M} X_{l,k}(s) ds \right] \right).$$

The fourth type is to minimize the probability the either dam is below level M at the terminal time T:

$$J_6(t, \mathbf{X}(t)) = \sum_{l=1}^{2} \left(\mathbb{E}^{p,u} \left[\sum_{k=1}^{M} X_{l,k}(T) \right] \right).$$

Here, for the J_5 and J_6 criteria, the expectation is taken under the probability measure induced by the set of optimal controls.

In the current example, the control resources are unconstrained so $f_0(t, p(t, \mathbf{X}(t)), u^{(1 \to 2)}(t, \mathbf{X}(t)), u^{(2 \to 1)}(t, \mathbf{X}(t)), \mathbf{X}(t))$ is simply the sum of the first four criteria. The fifth criterion is added as a running cost on each state of the chain below level M on the control period and the sixth as a cost on the terminal states below level M.

7.6 Computational aspects

Given the system of differential equations (4.6), we must write efficient code to solve the system and obtain the optimal controls. In the first instance we wrote the code to solve the system in a serial fashion and the results of this have been included as a numerical example in [38]. In serial, however, the time taken to solve the system and calculate the controls can be considerable for a two dam system with a large number of states in each. For example, it took approximately 45 minutes to solve for a two dam system with 20 states in each dam on the office desktop computer. It was desirable to reduce this substantially. Parallel computing was the obvious way forward because there were parts of the code that could clearly be parallelized. In this section we discuss the main points of how our code was parallelized.

The ideas presented here are taken from a tutorial on parallel computing given at Supercomputing 2010 (SC10) [56]. Since this is not intended as a technical exposition of parallel computing in general, we will limit discussion to the parallelization of our dam system code. The first step was to identify parts of the code that could be easily run in parallel. This was comprised of two different types of code in general. The first type was where code simply constructed a definition. For example, the following code defines the $J_1(t, p(t, \mathbf{X}(t)), \mathbf{X}(t))$ performance criteria. It seeks to minimize the difference between the demand for water in dam one, the sum of the $x_k(t)$, and the optimally supplied water, taking into account the level of both dams, $C_1(i, j)(t)$, that is dam one is in level i and dam two in level j. The Mathematica code for the serial definition is

```
J_1=Table[(c_1[i][j][t]-Sum[x[k][t],
    {k,1,nsector[1]}])^2,{i,1,L},{j,1,L}].
```

The key point to note is that since we are simply defining this criterion and no calculation is taking place, we can safely define this criterion for each joint state in parallel. The following code is for the same definition but parallelized:

```
J_1=WaitAll[Table[ParallelSubmit[{i,j},
```

```
    (c_1[i][j][t]-Sum[x[k][t],
    {k,1,nsector[1]}])^2],{i,1,L},{j,1,L}]].
```

The command `ParallelSubmit` submits each element of the definition to the next available Mathematica kernel and `WaitAll` ensures that no code after this command is executed until all the definitions have been made. This is not so important here but becomes important when the value of a definition is changed as a result of this calculation. If code is executed before all new values are assigned, errors result. So this command is for safety and reliability of execution.

The second type of code that could be parallelized involved calculations which included variables that would not be evaluated until a later point. For example, in our problem we must minimize over each control individually. The solution is a function that involves some combinations of $\Phi(i,j)(t)$, which are the running costs of the joint states. However, these $\Phi(i,j)(t)$ are to be solved for as the solution to a system of ODE's at a later time and so we can do these minimizations in parallel without affecting the values of $\Phi(i,j)(t)$. The following code finds the minimizing function for consumption in the first dam, $C_1(i,j)(t)$, in serial:

```
S_1=Table[s_1=Solve[D[EQ[[i,j]],
    c_1[i][j][t]]==0,c_1[i][j][t]];
    c_1[i][j][t]/.s_1,{i,1,L},{j,1,L}].
```

Compare this with the parallelized code:

```
S_1=WaitAll[Table[ParallelSubmit[{i,j},
    s_1=Solve[D[EQ[[i,j]],c_1[i][j][t]]==0,
    c_1[i][j][t]];c_1[i][j][t]/.s_1],
    {i,1,L},{j,1,L}]]
```

You will notice that the commands are the same and used in the same way as for simple parallel definitions.

The last element which must be considered is the distribution of variable and function definitions to all available processors, in our case Mathematica kernels. This is accomplished by executing the command `DistributeDefinitions[...,...,...]`, where the arguments are all of the variable and function definitions which must be available to each kernel in the subsequent calculation, separated by commas. At least in Mathematica, the process of parallelizing code is relatively simple, although experimentation is required. One aspect that needs to be considered is the computational overhead of parallelization. Every call to a different kernel and transfer of data between kernels takes time and some parts of the code may not be worth parallelizing if they already execute very quickly. In our example there were a number of very simple definitions that could be parallelized but the resulting performance was either a very minimal increase in speed or it was slightly slower. We found that it pays to focus on the areas of code that seem to take the most time when executing in serial.

All of this can be accomplished in various other programming languages, but the ease of implementation may differ considerably. We have used Mathematica 7.0 due to familiarity with this package and its relative ease of use. Implementing this in C, for example, would be more difficult but the execution would very likely be faster. For the purpose of preliminary experimentation Mathematica has been sufficient, however, we may need to write our problem in a lower level language like C at a later point depending on the size of the dam system we want to model.

7.7 Numerical results

The results of this section were obtained on a desktop computer with an Intel®Core™2 Duo CPU E8600 3.33Ghz processor and 3.49GB of RAM. The operating system was Microsoft Windows XP Professional version 2002, Service Pack 3 and the numerical software was Wolfram Mathematica version 7.0.0, Microsoft Windows 32-bit. Having a dual-core processor results in Mathematica being able to run two computational kernels in parallel. This was an ideal environment to perform some numerical experiments with parallel computing to see what the performance gains are like by parallelizing sections of appropriate code. After this, the code was run on the Monash Sun Grid (MSG), a high performance computer.

The basic idea was to see the difference in CPU seconds used between the Mathematica code executed in serial and in parallel. The code used solved for the optimal controls in a two dam system, found the probability of the dam system being in a particular state at any time $t \in [0, 1]$ and compared the original demand with an average optimal consumption weighted by these probabilities. From these results one can obtain all of the pertinent performance characteristics of the optimal system. This code was executed in series and parallel three times each for dams with $3, 4, ..., 9, 10$ states in each, and the mean time calculated. The results of this are shown in Table 7.1.

From Table 7.1 we can see that when the dams have three levels each, the performance is marginally worse with the parallel code since there is a lot of computational overhead for a very small number of equations to solve. However, it is clear from the column showing 'Speedup' that there is a rapid increase in performance with the increase in states. For interest we also ran the two versions

Table 7.1: Mean CPU seconds for serial and parallel execution of Mathematica code.

States per dam	Serial	Parallel	Speedup
3	1.74	1.97	0.88
4	3.47	3.46	1.00
5	6.97	5.93	1.18
6	12.34	8.83	1.40
7	20.08	11.00	1.83
8	33.85	15.29	2.21
9	54.05	19.13	2.83
10	84.36	23.92	3.53

of code once at $15, 20$ and 25 levels. For 15 levels the approximate speedup was 8.69 times, for 20 levels it was approximately 16.24 times, and at 25 levels Mathematica ran out of memory.

For a very small amount of change in the original serial code there is clearly a significant gain in execution speed. As stated, we have also executed these two versions of our code on the head node of the MSG, the university's grid computer. These results for one run are for dams with 10, 20,...,50 levels in each and shown in Table 7.2, although serial calculations were aborted once the calculation time exceeded that of the largest parallel calculation. This node consists of a Sun X4600 chassis with 8 Opteron quad-core CPUs for a total of 32 cores. Each core has 2GB of RAM [1]. These calculations were carried out on one CPU (4 cores with 8GB RAM). You will note that for 10 levels the parallel calculations on the desktop desktop are faster. This is apparently due to hardware differences, especially processor speed. While not shown, the calculations for 20 levels were also approximately twice as fast on the desktop, so for a number of levels that a desktop can handle, desktop performance is

Table 7.2: CPU seconds for serial and parallel execution of Mathematica code on the MSG.

States per dam	Serial	Parallel	Speedup
10	308.19	48.79	6.32
20	4639.55	316.76	14.65
30	> 21636.5	1882.44	-
40	-	6235.81	-
50	-	21636.5	-

comparatively good.

Of course, the MSG can handle much larger systems. In this application we want to control water consumption in a two dam system with an annual storage fluctuation of up to 20%. Since dam levels are usually quoted in 0.1% increments, this would require up to 200 levels per dam (a system of 40000 ODEs), an extremely demanding task computationally. Probably 0.5% changes in level are fine enough for our purposes and this would give up to 40 levels in each dam (a system of 1600 ODEs), which took under two hours for a single CPU on the MSG.

7.8 Final remarks

In this chapter we have considered a model for the optimal control of a system of two dams through the agency of state and time dependent price and water transfer controls and some of the computational aspects of its solution. We have explained the use of parallel computing in our work as a means for reducing calculation speed and making larger systems more tractable. Parallel computing presents real opportunities to attempt to solve problems with a larger number of states in each dam and more dams. This chapter was the result of work done in 2011, however, the methods shown in this are the backbone of the solution for all subsequent models produced. Over time the code for each model has become more or less standardized in format with parallelization used in much the same places. This does not mean that attempts at parallelization have always been successful. Mathematica provides an excellent environment to try things with relatively limited programming skills. The downside of this is that much of what it is doing is hidden and cannot be readily altered. As a result, where parallelization has worked for certain calculations in one model, it has failed in a subsequent model with no clear reason as to why. This is definitely a point in favor of trying to solve the problem in another language such as C where there is far greater control over how exactly the parallelization is being implemented. This must be balanced with the much larger investment in time required for such an implementation.

Conclusion

The optimal control of resources, such as water, is a major challenge in an increasingly resource constrained world. It is more challenging when the supply of the resource is difficult to predict and demands are changing seasonally. This thesis has demonstrated some optimal control strategies and solution methods that may be of use in such settings. For this purpose we have chosen dams because of the complexity of the dynamics and interactions between supply and customer demands. We have deliberately avoided trying to model any specific dam so as to make the methods found as general as possible while still providing reasonable solutions when viewed from a common sense point of view. It would be misleading to provide examples where these methods perform very well but fail to show where they do not, and such cases must also exist. Each situation needs to be evaluated on its own merits as to the applicability of these methods.

We stated in the introduction that ideally we would like to be able to build models of dams which provide water for human consumption, irrigation and power while reducing the risk of flooding. We would like to be able to achieve these aims under diverse and possibly extreme climatic conditions and do so in a way that provides us with the maximum flexibility in management of the resource. The papers presented in this thesis have shown that such models can be built and that in general they give results that 'make sense', at least in terms of the specific performance criteria chosen.

Mathematically, this thesis has dealt with the development of tools based on the optimal control of continuous-time controllable Markov chains (CMC) and their application to the control of complex systems. As stated, the applicability of these methods is much wider than shown here, as evidenced by its use in internet congestion control [32]. The main question was how to optimally control a large dam (or system of dams) with non-stationary inflows and outflows on a finite time horizon, under constraints. This was approached via controlled Markov chains, as originally presented in [18, 32]. We have shown that it is possible, subject to computing power, to model an arbitrarily sized and configured system under control resource constraints and obtain optimal pricing and water release strategies which are reasonable in terms of the application.

In Chapter 2 the basic solution method for the unconstrained case was given in full detail. Essentially we solve the stochastic optimal control problem by taking expectations and using dynamic programming. This gives a system of ordinary differential equations that have solutions at all times during the control period. The great advantage of this method is that given a set of system parameters, unique optimal solutions can be found using well established numerical procedures.

Chapter 3 gave the first model of an abstract dam based on these methods. It had only a time-inhomogeneous simple counting process for inputs and outputs and no provision for water release to control flooding. The optimal prices found were reasonable, in that prices were generally higher in the lowest states, but this was not uniform and made the solution forms difficult to interpret. The

ODE solutions showed that there was a significant number of points where the solutions for each state crossed, leading to irregularity in prices.

The next attempt, given in chapter 4 was to try to join two abstract dams together. This was successfully achieved and the price solutions were more regular than in the single dam case described in Chapter 3. This was because the system allowed for the transfer of water between dams and so there was more regularity to the ODE solutions. This model did not deal with flood prevention.

Chapter 5 detailed a significant increase in the complexity of the dynamics of the model. It introduced a time-inhomogeneous compound Poisson process for inflows and a controllable release for flood prevention. The time-inhomogeneous compound Poisson process was required for the possibility of a level jump distribution to give the probabilities of jumping more than one level in the event of extreme inflows. This is an increase in the flexibility of the model and no particular jump distribution is assumed. We have used some rainfall data to build a distribution but this could be fitted for any location. Likewise, the possibility of controlled release improves the flexibility of the model, especially the control of transition probabilities in the highest states. This was not done under constraints and the task of finding feasible solutions to the optimal control problem under constraints was dealt with in the next chapter.

In Chapter 6 we considered the question of how to demonstrate the existence of optimal control solutions in the constrained optimal control problem. The model used had essentially the same dynamics of that given in the previous chapter but the performance criteria were a system of inequality constraints. The results given in that chapter provide a numerical method for establishing the existence of optimal solutions by showing that the system of inequality constraints is consistent. Finding the optimal controls under constraints was not done here due to the increased complexity of the numerical methods needed to solve the problem. This requires some programming ability in languages such as C++ or FORTRAN due to the difficulty in getting the solutions to converge near the points of optimality.

The final chapter, Chapter 7, was dedicated more to the computational aspects of this project. It took the model presented in Chapter 4 and explored the use of high performance computing (HPC) and of parallelization of computation to enable the solution of large systems, especially connected systems. It was shown that HPC and parallelization provide significant benefits in computational time compared with desktop computing and that there is scope to improve the efficiency by rewriting the Mathematica code in a lower level language.

In summary, the main achievements of this work have been:

- The use of a semimartingale model for dam management. This model allows for control in a non-stationary environment on finite time frames and for the ability to build models with connected dams, each with their own active users. The method of building models of connected dams is quite intuitive and simplifies the construction of the numerical system to be solved.

- The optimal control problems considered are numerically tractable, especially with the use of high performance computing. It is well known that the curse of dimensionality is a major problem for discrete MDP problems and for continuous MDP problems. The method shown here is also computationally intensive but has been reduced to the solution of a system of ODEs, the size of which may be large, and can be controlled by judicious discretization of the dam levels.

- The introduction of the compound Poisson process for modeling inflows. This has allowed a significant increase in the flexibility of modeling because it is possible to include the seasonality of inflows. It fits very well with the semimartingale model used and the numerical methods developed.

- Finding feasible solutions for the optimal control problem. In many of the models developed it is not possible to find the optimal solution in any global sense because such a solution is not able to be realized with the given set of controls. It is very important to be able to find the best solution with a given set of controls, since this is what is required in a practical sense. The method of solution demonstrated clearly allows us to find feasible solutions.

- The development of the methodology for the solution of the optimal control problem for dams with constraints. This cannot be achieved analytically and needs effective numerical methods for its realization. This work shows how to numerically determine whether or not feasible solutions exists in the constrained case and explains the general method for finding the optimal solution in the case where feasible solutions exist. The actual solution requires significant skill in programming.

We have shown through this series of chapters that it is possible to find reasonable control solutions for systems with complex dynamics, such as dams, at least in an abstract setting. We have also shown that while these methods are computationally intensive, HPC and parallel computing offer a way forward with considerable promise. Further research in this field should begin to focus on how to control complex systems under constraints when there is incomplete information. In all the work presented here we have assumed that we at least know the average intensities of flows for all inputs of the system. It is a more difficult problem when these are only partly known or known through some other indirect measurement which introduces ambiguity. The field of the optimal control of continuous-time Markov chains is still very young and offers many opportunities for further research into basic methods and applications.

Appendix A

Mathematica code for model in Chapter 4

```
SetSharedVariable[L];
L = 15;
M = 5;
nsector[1] = 3;
nsector[2] = 3;
Array[x, nsector[1]];
x[1][t_] = Cos[2 * Pi * t] + 4.5;
x[2][t_] = 0.3 * Cos[2 * Pi * t] + 4.5;
x[3][t_] = 0.5 * Cos[2 * Pi * t] + 5;
Array[y, nsector[2]];
y[1][t_] = Cos[2 * Pi * t] + 5;
y[2][t_] = 0.4 * Cos[2 * Pi * t] + 4;
y[3][t_] = 0.3 * Cos[2 * Pi * t] + 4;
Array[λ, 2];
λ[1][t_] = Sin[2 * Pi * t] + 10;
λ[2][t_] = Sin[2 * Pi * t + Pi / 6] + 9;
μ[L][t_] = -Sin[2 * Pi * t] + 4.5;
For[i = 1, i ≤ L - 1, i++, μ[L - i][t_] = (L - i)/L * μ[L][t]]
ν[L][t_] = -Sin[2 * Pi * t + Pi / 6] + 3.5;
For[i = 1, i ≤ L - 1, i++, ν[L - i][t_] = (L - i)/L * ν[L][t]]
pmax = 1.75;
pmin = 1;
K = 150;
r = 0.25;
```

$$\text{cmax}[1][\text{t\_}] = \sum_{i=1}^{\text{nsector}[1]} \text{Max}[(1 - r) * x[i][t] - \text{pmin} / (2 * \alpha), 0];$$

$$\text{cmin}[1][\text{t\_}] = \sum_{i=1}^{\text{nsector}[1]} \text{Max}[(1 - r) * x[i][t] - \text{pmax} / (2 * \alpha), 0];$$

```
α = 3 * pmin /
    (2 * (1 - r) * Sum[x[i][0], {i, 1, nsector[1]}] - 2.4 * (λ[1][0] - μ[M][0]));
```

$$\text{cmax}[2][\text{t\_}] = \sum_{i=1}^{\text{nsector}[2]} \text{Max}[(1 - r) * y[i][t] - \text{pmin} / (2 * \beta), 0];$$

```
cmin[2][t_] = Sum[Max[(1 - r) * y[i][t] - pmax / (2 * β), 0], {i, 1, nsector[2]}];

β = 3 * pmin /
    (2 * (1 - r) * Sum[y[i][0], {i, 1, nsector[2]}] - 2.4 * (λ[2][0] - ν[M][0]));

DistributeDefinitions[L, M, nsector[1], nsector[2], pmax, pmin, α, β, K, r];

SetSharedFunction[x[1], x[2], x[3], y[1], y[2], y[3],
  λ[1], λ[2], cmax[1], cmin[1], cmax[2], cmin[2], μ[L], ν[L]];

A1 = SparseArray[{{1, 1} -> -λ[1][t] - u2[1][j][t],
   {1, 2} → c1[2][j][t] + u1[2][j][t] + μ[2][t]}, {L, L}];

Table[Al = SparseArray[{{l, l - 1} -> λ[1][t] + u2[l - 1][j][t],
    {l, l} → -(c1[l][j][t] + u1[l][j][t] + μ[l][t] + λ[1][t] + u2[l][j][t]),
    {l, l + 1} → c1[l + 1][j][t] + u1[l + 1][j][t] + μ[l + 1][t]}, {L, L}], {l, 2, L - 1}];

AL = SparseArray[{{L, L - 1} -> λ[1][t] + u2[L - 1][j][t],
   {L, L} → -(c1[L][j][t] + u1[L][j][t] + μ[L][t])}, {L, L}];

A = Sum[Ak, {k, 1, L}];

B1 = SparseArray[{{1, 1} -> -λ[2][t] - u1[i][1][t],
   {1, 2} → c2[i][2][t] + u2[i][2][t] + ν[2][t]}, {L, L}];

Table[Bl = SparseArray[{{l, l - 1} -> λ[2][t] + u1[i][l - 1][t],
    {l, l} → -(c2[i][l][t] + u2[i][l][t] + ν[l][t] + λ[2][t] + u1[i][l][t]),
    {l, l + 1} → c2[i][l + 1][t] + u2[i][l + 1][t] + ν[l + 1][t]}, {L, L}], {l, 2, L - 1}];

BL = SparseArray[{{L, L - 1} -> λ[2][t] + u1[i][L - 1][t],
   {L, L} → -(c2[i][L][t] + u2[i][L][t] + ν[L][t])}, {L, L}];

B = Sum[Bk, {k, 1, L}];

Ψ = WaitAll[Table[ParallelSubmit[{i, j}, ψ[i][j][t]], {i, 1, L}, {j, 1, L}]];

J1 = WaitAll[Table[ParallelSubmit[{i, j},
    (c1[i][j][t] - Sum[x[k][t], {k, 1, nsector[1]}]) ^ 2], {i, 1, L}, {j, 1, L}]];

J2 = WaitAll[Table[ParallelSubmit[{i, j},
    (c2[i][j][t] - Sum[y[k][t], {k, 1, nsector[2]}]) ^ 2], {i, 1, L}, {j, 1, L}]];

J3 = WaitAll[Table[ParallelSubmit[{i, j},
    (λ[2][t] + u1[i][j][t] - Sum[y[k][t], {k, 1, nsector[2]}] - ν[j][t]) ^ 2], {i,
    1, L}, {j, 1, L}]];

J4 = WaitAll[Table[ParallelSubmit[{i, j},
    (λ[1][t] + u2[i][j][t] - Sum[x[k][t], {k, 1, nsector[1]}] - μ[i][t]) ^ 2], {i,
    1, L}, {j, 1, L}]];

J5 = WaitAll[
  Table[ParallelSubmit[{i, j}, K Boole[i ≤ M] + K Boole[j ≤ M]], {i, 1, L}, {j, 1, L}]];

f0 = Sum[Jl, {l, 1, 5}];

EQ =
 Partition[Flatten[Table[SparseArray[{1, i} → 1, {1, L}].(Transpose[A].Ψ + Ψ.B + f0).
     SparseArray[{j, 1} → 1, {L, 1}], {i, 1, L}, {j, 1, L}]], L];

DistributeDefinitions[EQ];
```

126

```
S1 = WaitAll[Table[
   ParallelSubmit[{i, j}, s1 = Solve[D[EQ[[i, j]], c1[i][j][t]] == 0, c1[i][j][t]];
    c1[i][j][t] /. s1], {i, 1, L}, {j, 1, L}]];

DistributeDefinitions[S1];

Table[c1[i][j][t_] = Piecewise[{{cmin[1][t], S1[[i, j, 1]] ≤ cmin[1][t]},
    {S1[[i, j, 1]], cmin[1][t] < S1[[i, j, 1]] < cmax[1][t]},
    {cmax[1][t], S1[[i, j, 1]] ≥ cmax[1][t]}}], {i, L}, {j, L}];

S2 = WaitAll[Table[
   ParallelSubmit[{i, j}, s2 = Solve[D[EQ[[i, j]], c2[i][j][t]] == 0, c2[i][j][t]];
    c2[i][j][t] /. s2], {i, 1, L}, {j, 1, L}]];

Table[c2[i][j][t_] = Piecewise[{{cmin[2][t], S2[[i, j, 1]] ≤ cmin[2][t]},
    {S2[[i, j, 1]], cmin[2][t] < S2[[i, j, 1]] < cmax[2][t]},
    {cmax[2][t], S2[[i, j, 1]] ≥ cmax[2][t]}}], {i, L}, {j, L}];

S3 = WaitAll[Table[
   ParallelSubmit[{i, j}, s3 = Solve[D[EQ[[i, j]], u1[i][j][t]] == 0, u1[i][j][t]];
    u1[i][j][t] /. s3], {i, L}, {j, L}]];

S3 = S3 /. Join[Table[S3[[1, j, 1]] → 0, {j, 1, L}], {S3[[L, L, 1]] → 0}];

Table[u1[i][j][t_] = Piecewise[{{0, S3[[i, j, 1]] ≤ 0},
    {S3[[i, j, 1]], 0 < S3[[i, j, 1]] < 1}, {1, S3[[i, j, 1]] ≥ 1}}], {i, L}, {j, L}];

S4 = WaitAll[Table[
   ParallelSubmit[{i, j}, s4 = Solve[D[EQ[[i, j]], u2[i][j][t]] == 0, u2[i][j][t]];
    u2[i][j][t] /. s4], {i, 1, L}, {j, 1, L}]];

S4 = S4 /. Join[Table[S4[[i, 1, 1]] → 0, {i, 1, L}], {S4[[L, L, 1]] → 0}];

Table[u2[i][j][t_] =
  Piecewise[{{0, S4[[i, j, 1]] ≤ 0}, {S4[[i, j, 1]], 0 < S4[[i, j, 1]] < 1},
    {1, S4[[i, j, 1]] ≥ 1}}], {i, 1, L}, {j, 1, L}];

INIT = Flatten[WaitAll[Table[ParallelSubmit[{i, j}, ψ[i][j][1] ==
       K Boole[i <= M] + K Boole[j <= M]], {i, 1, L}, {j, 1, L}]]];

DESYS =
  Flatten[WaitAll[Table[ParallelSubmit[{i, j}, ψ[i][j]'[t] == -EQ[[i, j]]],
     {i, 1, L}, {j, 1, L}]]];

SOL1 =
  NDSolve[Join[DESYS, INIT], Flatten[Table[ψ[i][j][t], {i, 1, L}, {j, 1, L}]],
   {t, 0, 1}, Method -> ExplicitRungeKutta];

DistributeDefinitions[SOL1];

Price = Table[pr[i][j][t], {i, 1, L}, {j, 1, L}];

Table[pr[i][j][t_] = 4 α β / (2 β nsector[1] + 2 α nsector[2])
    (Sum[x[k][t] (1 - r), {k, 1, nsector[1]}] + Sum[y[k][t] (1 - r),
       {k, 1, nsector[2]}] - c1[i][j][t] - c2[i][j][t]) /. SOL1, {i, L}, {j, L}];

Partition[Flatten[Table[{i, j, pr[i][j][t]}, {i, 1, L}, {j, 1, L}]], L];

PR = Table[prob[i][j][t], {i, 1, L}, {j, 1, L}];
```

```
EQ2 = Partition[
   Flatten[Table[SparseArray[{1, i} -> 1, {1, L}].(A.PR + PR.Transpose[B]).
      SparseArray[{j, 1} -> 1, {L, 1}], {i, 1, L}, {j, 1, L}]], L];

DistributeDefinitions[EQ2, PR];

DESYS2 = Flatten[WaitAll[Table[ParallelSubmit[{i, j},
      D[PR[[i, j]], t] == EQ2[[i, j]] /. SOL1], {i, 1, L}, {j, 1, L}]]];

INIT2 = Flatten[WaitAll[Table[ParallelSubmit[{i, j},
      prob[i][j][0] == Boole[{i, j} == {L, L}]], {i, 1, L}, {j, 1, L}]]];

SOL2 = NDSolve[Join[DESYS2, INIT2], Flatten[PR], {t, 0, 1}, PrecisionGoal -> 3];
```

References

[1] Monash Sun Grid Overview. Monash University e-Research Centre, September 2010. https://confluence-vre.its.monash.edu.au/display/mcgwiki/Monash+Sun+Grid+Overview, last accessed 30 March 2015.

[2] M. Abdel-Hameed. Optimal control of a dam using $p^M_{\lambda,\tau}$ policies and penalty cost when the input process is a compound Poisson process with positive drift. *Journal of Applied Probability*, 37(2):408–416, 2000.

[3] Mohamed Abdel-Hameed. Control of dams using $p^M_{\lambda,\tau}$ policies when the input process is a nonnegative Lévy process. *International Journal of Stochastic Analysis*, 2011:17, 2011.

[4] V. Abramov. Optimal control of a large dam. *Journal of Applied Probability*, 44(1):249–258, 2007.

[5] L. Aggoun and R. Elliott. *Measure Theory and Filtering: Introduction with Applications*. Cambridge University Press, Cambridge, 2004.

[6] S. Sebnem Ahiska, Samyuktha R. Appaji, Russell E. King, and Donald P. Warsing Jr. A Markov decision process-based policy characterization approach for a stochastic inventory control problem with unreliable sourcing. *International Journal of Production Economics*, 144(2):485–496, 2013.

[7] E. Altman. *Constrained Markov Decision Processes*. Chapman & Hall/CRC, Boca Raton, FL, 1999.

[8] J. Bae, S. Kim, and E.Y. Lee. Average cost under the $p^M_{\lambda,\tau}$ policy in a finite dam with compound Poisson inputs. *Journal of Applied Probability*, 40(2):519–526, 2003.

[9] L. Berkovitz. *Convexity and Optimization in* $\mathbb{R}^n$. John Wiley and Sons, New York, 2002.

[10] D. Bertsekas. *Dynamic Programming and Stochastic Control*. Academic Press, London, 1976.

[11] D. Bertsekas and S. Shreve. *Stochastic Optimal Control: The Discrete-Time Case*. Athena Scientific, Belmont, MA, 1996.

[12] D. Bertsekas and J. Tsitsiklis. *Introduction to Probability*. Athena Scientific, Belmont, Mass., second edition edition, 2008.

[13] P. Bremaud. *Point Processes and Queues, Martingale Dynamics*. Springer-Verlag, Berlin, 1981.

[14] C. Ceci, A. Gerardi, and P. Tardelli. Existence of optimal controls for partially observed jump processes. *Acta Applicandae Mathematicae*, 74:155–175, 2002.

[15] M.H.A. Davis. *Markov models and optimization*. Chapman & Hall, 1993.

[16] F. Delebecque and J. Quadrat. Optimal control of Markov chains admitting strong and weak interactions. *Automatica*, 17(2):281–296, 1981.

[17] R. Elliott. A partially observed control problem for Markov chains. *Applied Mathematics and Optimization*, 25:151–169, 1992.

[18] R. Elliott, L. Aggoun, and J. Moore. *Hidden Markov Models: Estimation and Control*. Springer-Verlag, New York, 1995.

[19] M. Faddy. Optimal control of finite dams: Discrete (2-stage) output procedure. *Journal of Applied Probability*, 11(1):111–121, 1974.

[20] J.A. Filar and K. Vrieze. *Competitive Markov Decision Processes - Theory, Algorithms and Applications*. Springer, New York, USA, 1997.

[21] W. H. Fleming and R. W. Rishel. *Deterministic and Stochastic Optimal Control*. Springer-Verlag, Berlin, Heidelberg, New York, 1975.

[22] Tingting Han, Joost-Pieter Katoen, and Alexandru Mereacre. Compositional modeling and minimization of time-inhomogeneous markov chains. In Magnus Egerstedt and Bud Mishra, editors, *Hybrid Systems: Computation and Control*, volume 4981 of *Lecture Notes in Computer Science*, pages 244–258. Springer Berlin Heidelberg, 2008.

[23] Herodotus. *Herodotus: The History*. University Of Chicago Press, reprint edition edition, 1988. Translated by David Grene.

[24] R. Howard. *Dynamic Programming and Markov Processes*. John Wiley and Sons, New York, 1960.

[25] Qiying Hu and Wuyi Yue. *Markov Decision Processes With Their Applications*. Springer US, New York, 2008.

[26] Martin Jacobsen. *Point Process Theory and Applications: Marked Point and Piecewise Deterministic Processes*. Birkhäuser, Boston, 2006.

[27] Andrzej Karbowski and PrzemysÅaw Magiera. Optimal control of a water reservoir with expected value-variance criteria. *Optimal Control Applications and Methods*, 28(1):3–20, 2007.

[28] M. Kitaev and V. Rykov. *Controlled Queueing Systems*. CRC, Boca Raton, FL, 1995.

[29] E. B. Lee and L. Marcus. *Foundation of Optimal Control Theory*. John Wiley & Sons, New York, London, Sydney, 1967.

[30] Daniel McInnes. Optimal Management of Dam Systems via Controlled Markov Chains. Poster presentation at Super Computing 2010 (SC10), 13-19 November, New Orleans., 2010.

[31] Alexander Miller and Boris Miller. Control of connected Markov chains. application to congestion avoidance in the Internet. In *Decision and Control*

and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on, pages 7242–7248, 2011.

[32] B. Miller. Optimization of queuing system via stochastic control. *Automatica*, 45:1423–1430, 2009.

[33] B. Miller, G. Miller, and K. Semenikhin. Methods to design optimal control of Markov process with finite state set in the presence of constraints. *Automation and Remote Control*, 72(2):323–341, 2011.

[34] B. Miller, G. Miller, and K. Siemenikhin. Control of Markov chains with constraints. In *Proceedings of the VIII International Conference "System Identification and Control Problems" SICPRO '09*, Moscow, 26-30 January 2009.

[35] B. Miller, G. Miller, and K. Siemenikhin. Towards the optimal control of Markov chains with constraints. *Automatica*, 46:1495–1502, 2010.

[36] B.L. Miller. Finite state continuous time Markov decision processes with a finite planning horizon. *SIAM Journal of Control*, 6(2):266–280, 1968.

[37] B.M. Miller and D.J. McInnes. Management of a large dam via optimal price control. In *Proceedings of the 18th IFAC World Congress*, volume 18, pages 12432–12438. The International Federation of Automatic Control (IFAC), IFAC-PapersOnLine, August 2011.

[38] B.M. Miller and D.J. McInnes. Management of dam systems via optimal price control. In *Proceedings of the International Conference on Computational Science, ICCS 2011*, volume 4, pages 1373–1382. International Conference on Computational Science (ICCS), Procedia Computer Science,Elsevier, June 2011.

[39] B.M. Miller and D.J. McInnes. Optimal management of a two dam system via stochastic control: Parallel computing approach. In *Decision and Control*

and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on, pages 1417–1423. IEEE, IEEEXplore, December 2011.

[40] Boris Miller, Gregory Miller, and Konstantin Siemenikhin. Optimal control of Markov chains with constraints. In *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 512–518, Shanghai, December 16-18 2009. IEEE.

[41] P.C.D. Milly. An analytic solution of the stochastic storage problem applicable to soil water. *Water Resources Research*, 29(11):3755–3758, 1993.

[42] P.A.P. Moran. *The Theory of Storage.* Methuen's Monographs on Applied Probability and Statistics. Methuen & Co., Ltd., London; Wiley & Sons, Inc., New York, 1959.

[43] Masaki Ota. A Stochastic Rainfall Generator for the Hopkins River Catchment. Master Thesis, RMIT University, September 2009. Raw data made available from Minor Thesis.

[44] E.C. Ozelkan, A. Galambosi, E. Fernández-Gaucherand, and L. Duckstein. Linear quadratic dynamic programming for water reservoir management. *Applied Mathematical Modelling*, 21:591–598, 1997.

[45] A. B. Piunovskiy. *Optimal Control of Random Sequences in Problems with Constraints.* Kluwer Academic Publishers, Dordrecht-Boston-London, 1997.

[46] A. B. Piunovskiy. Multicriteria impulsive control of jump Markov processes. *Mathematical Methods of Operations Research*, 60:125–144, 2004.

[47] A. B. Piunovskiy. *Examples in Markov Decision Processes.* Imperial College Optimization Series, Vol. 2. Imperial College Press, London, 2013.

[48] A. B. Piunovskiy and X. Mao. Constrained Markovian decision processes: the dynamic programming approach. *Operations Research Letters*, 27:119–126, 2000.

[49] Stanley R. Pliska. Controlled jump processes. *Stochastic Processes and their Applications*, 3:259–282, 1975.

[50] M. Ramakrishnan, D. Sier, and P. G. Taylor. A two-time-scale model for hospital patient flow. *IMA Journal of Management Mathematics*, 16:197–215, 2005.

[51] Georgi E. Shilov. *Elementary functional analysis*. Dover, 1996.

[52] A.N. Shiryayev. *Probability*. Springer-Verlag, New York, 1984.

[53] Norman Smith. *A History of Dams*. Peter Davies, London, 1971.

[54] Moshe Sniedovich. A variance-constrained reservoir control problem. *Water Resources Research*, 16(2):271–274, 1980.

[55] South East Queensland Water (seqwater). January 2011 flood event: Report on the operation of sommerset dam and wivenhoe dam. Online, 2 March 2011.

[56] Q.F. Stout and C. Jablonowski. Parallel Computing 101, 14 November 2010. Tutorial notes from Supercomputing 2010 (SC10) - electronic copy given to participants.

[57] C.B. Wan and M.H.A. Davis. Existance of optimal control for stochastic jump processes. *SIAM Journal on Control and Optimization*, 17(4):511–524, 1979.

[58] Byron K. Williams. Markov decision processes in natural resources management: Observability and uncertainty. *Ecological Modelling*, 220(6):830–840, 2009.

[59] E. Wong and B. Hajek. *Stochastic Processes in Engineering Systems*. Springer-Verlag, New York, 1985.

[60] Wuthichai Wongthatsanekorn, Matthew J. Realff, and Jane C. Ammons. Multi-time scale Markov decision process approach to strategic network growth of reverse supply chains. *Omega*, 38(1-2):20–32, 2010.

[61] Reza Yaesoubi and Ted Cohen. Generalized Markov models of infectious disease spread: A novel framework for developing dynamic health policies. *European Journal of Operational Research*, 215(3):679–687, 2011.

[62] L. Yeh and L.J. Hua. Optimal control of a finite dam: Wiener process input. *Journal of Applied Probability*, 24(1):186–199, 1987.

[63] W. Zangwill. *Nonlinear Programming: A Unified Approach*. Prentice-Hall, Englewood Cliffs, 1969.