Self-Diagnosis and Self-Compensation in Autonomous Walking Robots

by

Gustavo Adolfo Schleyer Daza

A thesis submitted in fulfilment of
the requirements for the degree of
Doctor of Philosophy

Supervisor: Associate Professor R. Andrew Russell

Intelligent Robotics Research Centre
Department of Electrical and Computer Systems Engineering
Monash University
Clayton, Victoria 3800, Australia
November 2014

To my wife and little family, my partners in this journey.

Declaration

I hereby declare that no part of my dissertation has already been or is being concurrently submitted for any such degree or other qualification. Except where explicit references are made to the work of others, this dissertation is the result of my own work.

[redacted]

November 7th, 2014

Acknowledgements

First I would like to express my sincere and deep gratitude to my supervisor, R. Andrew Russell, for his unconditional support, kindness and patience. I could never thank him enough for all of the guidance, teaching and valuable advice he has provided during the development of this research. I feel very fortunate to have had the opportunity to work with him and I will value this experience during my lifetime.

I also would like to thank the support I received from the staff in the Department of Electrical and Computer Systems Engineering in Monash University. I specially thank Jack who helped me with the construction of the robot's mechanical components at an early stage of this project. I also thank Martin and Andrew Linzner in the mechanical workshop, for the construction of some of the metallic robot components and for being enthusiastic about teaching me how to use the tools and equipment in the workshop. I am grateful to Mr. Ray Cooper for providing everything I needed at such a short notice; and Daryl Gaspero and Ian Reynolds for their technical and cycling advice. I would also like to thank Emily Simic for the help and personal advice she offered me from the first day of my PhD studies.

In Chile I received support from a number of people who helped me to turn this dream into reality. I specially thank Edmundo López, Gastón Lefranc, José Ceroni, Enrique Piraino, Jorge Zazópulus and Carla Trincado in the Pontifical Catholic University of Valparaíso and Felipe Leighton for all of their support during the process of application to Monash University.

In addition, it is with great pleasure that I thank my parents Gustavo and Angelina, step-parents Marcelo and Susy; and my and my wife's family in Chile for all of their love, teachings and support.

Special thanks to the great friends I made in Melbourne. Caro, Felipe, Javy, Kathy, Carlos, Fabiola, Pili, Lucho, Emilio, Dani, Anh Tuan Phan and Ben Border, you guys made me feel at home.

I am forever in-debt with my wonderful wife Chany, who has sacrificed everything in the name of love and followed me to distant lands. This thesis would not have been possible without her endless love, caring and support.

This research was sponsored by Conicyt (National Commission of Scientific and Technological Research) through the Chilean Government Bicentennial Scholarship Program.

Publications

The work in this thesis has been presented in the following conference proceedings:

(1) Schleyer, G., Russell, A., Disturbance and Failure Classification in Walking Robots. Presented at the Australasian Conference on Robotics and Automation (ACRA), Melbourne, Australia, 2011.

(2) Schleyer, G., Russell, A.*, Adaptable Gait Generation for Autotomised Legged Robots*. Presented at the Australasian Conference on Robotics and Automation (ACRA), Brisbane, Australia, 2010.

Abstract

This thesis proposes and implements methods for the autonomous identification and classification of disturbances that have negative effects on a robot's performance (self-diagnosis), and the autonomous selection of suitable compensatory actions (self-compensation).

The proposed methods have been implemented in a walking hexapod robot provided with a number of sensors. Both, the robot's sensorial information and a quantitative measure of the robot's performance are obtained. This information is used for detecting, identifying and classifying obstructive conditions that have a strong impact on the robot's performance. Once the cause of a lack of progress in the robot's mission has been identified, suitable compensatory actions are found, executed and recorded. Then, when previously experienced detrimental situations arise, the associated compensatory measures are immediately taken without involving a searching process. As a result, the recovery from abnormal conditions is accelerated and the robot can promptly continue with its mission. In order to evaluate the performance of the proposed methods, different sets of experiments addressing the robot's hardware faults, abnormal situations generated in the robot's environment and a combination of both, were conducted. Results were evaluated by means of two indicators: the number of attempts before a correct identification of the robot's hardware fault was achieved, and a discrepancy measure. The latter indicates the Euclidean distance between the centroid of an abnormal situation experienced by the robot and the centroid of abnormal situations incorporated into the robot's database of anomalies. Results showed a good identification rate inside the repertoire of considered abnormal situations.

Among the compensatory measures addressed in this thesis, an adaptable gait generation algorithm which allows legged robots to walk in a stable fashion after they have shed a variable number of legs, and a compact leg release mechanism that provides legged robots with a method for the autonomous physical ejection of damaged legs without requiring extra motors, are proposed.

Two self-compensating methods (Autonomous Generated Compensatory Measures and Learned Compensating Measures) not implemented into the experimental robot are proposed in this thesis. In theory, these methods will allow robots to autonomously generate compensatory measures and learn from previously encountered abnormal situations.

The methods developed in this research are fundamental for the autonomous detection of a robot’s failures and adaptability to unforeseen features of the robot’s environment. By using these techniques, it is expected to increase resilience to damage, extend lifespan and improve autonomy in robotic missions where human intervention is difficult or impossible, such as in extra-terrestrial exploration or other remote hostile environments.

List of Acronyms

3D	Three-Dimensional
ABS	Acrylonitrile Butadiene Styrene
ADC	Analogous/Digital Converter
AIS	Abnormality Index Set
AL	Abnormality Label
ASC	Abnormal Situation Centroid
CAD	Computer-Aided Design
CAL	Cumulative Abnormality Label
I^{2}C	Inter-Integrated Circuit
ICM	Innate Compensatory Measure
LCM	Learned Compensatory Measure
LED	Light Emitting Diode
PC	Personal Computer
PCB	Printed Circuit Board
PWM	Pulse Width Modulation
RAL	Robot Anatomy Label
RMPSS	Robot’s Moving Part Selection Sequence
SMD	Surface Mount Device
SRS	Sensor Reading Set
TASC	Target Abnormal Situation Centroid

Contents

List of Figures

List of Tables

Chapter 1
Introduction

This chapter presents the underlying motivation and objectives of this investigation. Identification of the original contribution of this work and an overview of the thesis structure are also provided.

1.1. Motivation and Objective

The research work presented in this thesis has been strongly inspired by behavioural patterns of biological creatures when dealing with unstructured environments. According to Konrad Lorenz [1] biological behavioural patterns consist of innate and learned or experienced components. John W. Kimball [2] introduced taxes, instincts and reflexes as three examples of innate behaviour. These are described as follows.

- Taxes: are responses where organisms automatically move directly toward, away from, or at some defined angle to a stimulus. An example of this behaviour is phototaxis, which is commonly displayed by photosynthetic microorganisms. A positive phototaxis is displayed by microorganisms moving towards a light source, whereas a negative phototaxis is displayed by microorganisms moving away from a light.

- Instincts: according to Charles Darwin “An action, which we ourselves should require experience to enable us to perform, when performed by an animal, more especially by a very young one, without any experience, and when performed by many individuals in the same way, without their knowing for what purpose it is performed, is usually said to be instinctive" [3]. Examples of instinctive behaviour are foraging, mating, homing and safety related actions.

- Reflexes: are behavioural patterns that, like instincts, are inborn, rather inflexible and valuable for allowing an animal to respond quickly to its environment. They differ from instincts in complexity. Instincts are more complex, they may involve the whole body and an elaborate set of actions. On the other hand, reflexes are simpler and faster involuntary responses to a stimulus. Most reflexes are not mediated by the brain but involve a very simple nervous pathway called a reflex arc [4]. Examples of reflex behaviour are the withdrawal behaviour (i.e. occurs when a part of the body makes contact with a hot object and is quickly moved away from it) and autonomic reflexes (which regulate body functions such as digestion or blood pressure).

In [3], Darwin claimed that inborn behaviour and current physical features of species have been adjusted by biological evolution and natural selection. Therefore, complex instinctive behaviour and physical capabilities are the result of slow and gradual accumulation of numerous, slight, yet beneficial, variations. One physical capability that has motivated this thesis is autotomy, which is described in [5] as "an extreme response to predators, whereby an organism voluntarily sheds a limb or appendage to aid escape".

The second type of biological behavioural pattern introduced by Lorenz, learned behaviour, is produced as the result of an experience affecting an individual. This kind of behaviour has been classified into habituation, sensitization, imprinting, conditioned response and instrumental conditioning. Of all of these categories, instrumental conditioning is most related to this research. This learned behaviour is also known as trial-and-error learning, because in this scenario biological creatures are free to try various responses before finding the one that is rewarded.

Biological creatures are able to identify and compensate a range of detrimental abnormal situations by using innate behaviour provided by evolution. In addition, learning capabilities furnish a living organism with a certain degree of adaptation to changing conditions of their body and environment. Autonomous robots could benefit from such behavioural patterns, both innate and learned, when performing their missions in unstructured environments.

This thesis presents an investigation of techniques for generation of both types of artificial behavioural patterns in robotic systems. These patterns are used for the identification and compensation of detrimental disturbances originating in the robots or their environment. The research considers how robot designers can simulate biological evolution as the provider of innate behaviour and how empirical tests of compensatory measures can substitute for natural selection. Furthermore, this investigation addresses how robots can be provided with learning capabilities in order to benefit from experience and generate suitable learned behaviour for the compensation of a broader range of anomalies.

The overall aim of this work is to propose and implement methods for the autonomous identification and classification of disturbances that have negative effects on a robot's performance (self-diagnosis), and the autonomous selection of suitable compensatory actions (self-compensation). Applications of this work would help to increase the autonomy of robotic systems. For environments such as planetary exploration where human intervention is difficult or impossible, the capability for robots to diagnose and accommodate malfunctions would increase the probability of completing a mission successfully.

1.2. Contributions

Firstly, this investigation has proposed self-diagnosis and self-compensation methods that use both, innate and learned behavioural patterns.

This thesis provides a series of novel contributions to the areas of self-diagnosis, self-compensation and mechatronics in robotics.

In the area of self-diagnosis this research provides new methods and algorithms whose purpose is:

- Detection and identification of abnormal situations that negatively affect a robot's performance.
- Labelling and classification of anomalies generated internally within a robot or through interaction with the environment.
- Use of research actions and information about a robot's anatomy as tools for the rejection or corroboration of a robot's theories about sources of detrimental disturbances.

In the area of self-compensation this investigation presents novel techniques and algorithms whose aim is:

- Association of detrimental disturbances with corresponding compensatory measures.
- Adaptable gait generation that allows legged robots to walk in a stable fashion after they have shed a variable number of legs.
- Learning from experience and compensation of detrimental disturbances not included as part of the robot's innate behavioural patterns.

In the area of mechatronics this research has lead to the development of the following robotic sensor and mechanism.

- A 3D printed plastic optomechanical robot leg tip force sensor that generates a voltage derived from an opto-interrupter, which is related to the pressure or traction applied to the leg's tip.
- A compact leg release mechanism that provides legged robots with a method for the autonomous physical ejection of damaged legs without requiring extra motors.

Although contributions have been classified into different areas of knowledge, they can also be organised into primary and secondary contributions.

Primary contributions of this thesis are:

- New algorithms for detection and identification of abnormal situations that negatively affect a robot's performance.
- Novel methods for labelling and classification of anomalies generated internally within a robot or through interaction with the environment.
- Techniques that facilitate the association of detrimental disturbances with corresponding compensatory measures.
- Methods for Learning from experience and compensation of detrimental disturbances not included as part of the robot's innate behavioural patterns.

Secondary contributions of this thesis are:

- Techniques that use research actions and information about a robot's anatomy as tools for the rejection or corroboration of a robot's theories about sources of detrimental disturbances.
- An adaptable gait generation that allows legged robots to walk in a stable fashion after they have shed a variable number of legs.
- Development of a novel 3D printed plastic optomechanical robot leg tip force sensor that produces a voltage derived from an opto-interrupter, which is related to the pressure or traction applied to the leg's tip.
- Development of a compact leg release mechanism that provides legged robots with a method for the autonomous physical ejection of damaged legs without requiring extra motors.

1.3. Thesis Structure

This thesis is organised as follows:

Chapter 1 (this chapter) outlines the motivation, aims, contribution and structure of this work.

Chapter 2 provides a review of literature mainly dealing with biologically inspired walking robots, gait optimisation and adaptability, detection of robots anomalies and fault tolerant robots able to identify and compensate failures.

Chapter 3 presents the experimental hexapod robot designed, built and used during this investigation. This chapter is subdivided into the robot's design, control, sensors and robot's leg release mechanism.

Chapter 4 describes methods for the detection of abnormal situations that negatively affect a robot's performance. Techniques for the identification of the sources and the classification of these detrimental disturbances are also provided.

Chapter 5 provides techniques that allow the matching of detrimental disturbances with their corresponding compensatory measures included as patterns of the robot's innate behaviour. The chapter continues by presenting examples of compensatory measures used in the experimental robot. In addition, methods for the generation of compensatory measures not included in the robot's innate behaviour are provided. The chapter finally describes how learned compensatory measures are included into the robot's set of compensatory measures. Interactions among the algorithms that will be introduced in chapters 4 and 5 are shown in Fig. 1.1.

Chapter 6 concludes the thesis and recommends avenues for future research.

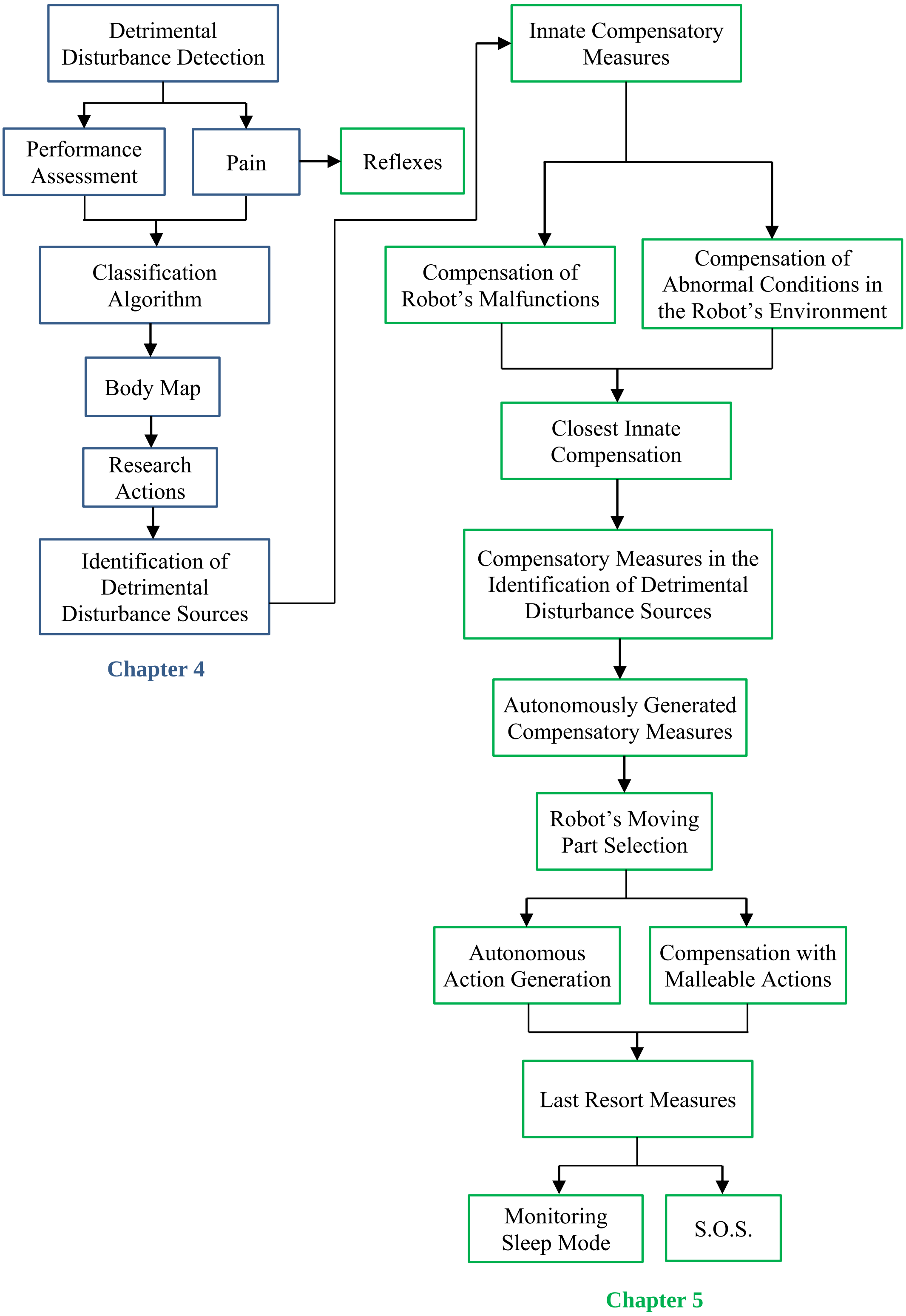

Detrimental Disturbance Detection
Performance Assessment
Pain
Reflexes
Classification Algorithm
Body Map
Research Actions
Identification of Detrimental Disturbance Sources
Chapter 4
Innate Compensatory Measures
Compensation of Robot's Malfunctions
Compensation of Abnormal Conditions in the Robot's Environment
Closest Innate Compensation
Compensatory Measures in the Identification of Detrimental Disturbance Sources
Autonomously Generated Compensatory Measures
Robot's Moving Part Selection
Autonomous Action Generation
Compensation with Malleable Actions
Last Resort Measures
Monitoring Sleep Mode
S.O.S.
Chapter 5

Figure 1.1. Flow Chart of Detrimental Condition Detection and Compensation Described in Chapters 4 and 5.

Chapter 2
Literature Review

This chapter provides background information for this research by introducing a selection of relevant publications. The review begins with a number of approaches to anomaly detection in walking robots. Then, the report is focused on works dealing with self-healing and fault tolerance, and in robots able to identify and compensate failures. Here, particular attention is given to biological inspired methods such as autotomy, force detection and reflex actions in robots. The report continues with different approaches for the use of sensorial information, provided by adaptable walking robots, in gait optimisation and adaptability to changing conditions associated with both robots and their environment. As more complex data sets are available, the study is extended to algorithms employed in robotics for the analysis of this information. This is followed by the presentation of studies dealing with the best ways of mechanical locomotion over rough terrain. Finally, biological approaches to gait generation in walking robots are reviewed.

2.1. Anomaly Detection in Walking Robots

A number of publications deal with adaptability for changes in the robot itself. They intend to detect faults or anomalies that affect a robot’s performance. For instance, a sensor could be sending wrong information, a motor could be broken or part of the mechanical structure of the robot could have changed. The detection of the anomaly is a first step before a compensatory measure can be executed.

A few related works have been published. For instance, the paper in [6] presents a method for classifying sensing failures in autonomous mobile robots. Here, the error classification is performed by a variation of the Generate and Test method [7]. This technique basically generates all of the possible causes of an error based on the symptoms, orders a list of associated tests and executes tests to confirm any of these causes. The classification is terminated when all tests have been performed or an environmental change has been confirmed. Subsequent papers of the authors [8] [9] also describe works following this line. However, the presented methods seem to fail when they face different kinds of unexpected situations. The methods in [6], [8], and [9] present some similarities with the self-diagnosis techniques proposed in this thesis. Basically, both approaches generate theories about the source of an anomaly and then execute tests to validate or reject these theories. Nevertheless, the way that these theories are generated is different. In the considered publications, the generation of theories about the source of an anomaly is based on the Generate and Test method, presented as a part of the DENDRAL Project. In comparison, the algorithms that will be proposed in this thesis are simpler, which facilitate their implementation in small robots. In addition, the authors in [6], [8], and [9] identify anomalies by using 4 symptoms: missing data, highly uncertain data, and highly conflicting observations and below minimum certainty in the percept. This does not allow the robot to identify gradual changes in the sensor readings generated by a fault (e.g. a servo motor gradually increasing its temperature until it is overheated). In addition, tests are executed when a symptom is detected even if this is not preventing the robot from making progress in its mission, which in some cases could be inefficient. As will be discussed in Chapter 4, these issues are addressed in this thesis by continuously monitoring the performance of the robot and incorporating pain into the robot. As a result, the self-diagnosis methods only interrupt the normal operation of the robot when it is necessary. Therefore, in this thesis an anomaly is understood as extreme sensor values or a lack of progress in the robot's mission.

A biologically inspired fault detection method referred to as an Artificial Immune System and based on immune theory is reported in [10], [11] and [12]. There are a number of techniques associated with the Artificial Immune System. For instance, Negative Selection, a mechanism based on the ability of the immune system to learn to distinguish between non-self- cells and self-cells is employed in [10]. On the other hand, the authors in [11], [12] and [13] prefer to use Clonal Selection. As stated in [12], the Clonal Selection method is based on a mechanism employed by antibody cells, which upon discovery of an antigen; start to proliferate by cloning themselves. They also memorise the antigenic attack (immune memory) so they have higher responsiveness to the particular antigenic attack over time. Researchers in [11], [12] and [13] combined the Clonal Selection principle and Fuzzy Logic with weight factors for detecting robot anomalies.

A different approach is adopted in [14] where the robot EduBot is presented. This walking machine is able to identify disturbances by means of finite state machines. The proposed technique is employed for the identification of two abnormal states, corresponding to missing ground contact (interpreted as a leg fault) or unexpected contact (interpreted as contact with a wall). EduBot is provided with two reactive behaviours, each one targeting one of the two considered disturbances. Finite state machines for detecting malfunctions evidenced by a drastic difference between the expected and the actual value of a sensor reading are also reported in [15] and [16]. Here, those faults that manifest themselves through a combination of abnormal sensor readings are classified by means of neural networks. The investigation presented in [17], [18] and [19] use a back-propagation neural network for identifying simulated hardware faults. Here, the network is trained to identify changes in the relation between sensor readings and the control program behaviour, which arise when the faults are introduced. A disadvantage of methods based on neural networks is that they require large amounts of training data before they can provide a good rate of correct classification. This intrinsic training requirement seems unnatural when we consider living creatures that successfully compensate anomalies without the need of training or a learning process. Indeed, it is widely accepted that a combination of innate behaviour provided by evolution and learning capabilities is what has allowed living creatures to adapt to the changing conditions of their environment. The self-diagnosis and self-compensation methods to be proposed later on in this thesis utilise innate information given to robots about possible anomalies they may find. In addition, these methods also provide robots with the ability to learn ways to compensate anomalies not originally considered.

Abnormality detection has also been applied in multi-agents systems. For example, in [20] a cross-regulation model is utilised for detecting an agent having an abnormal behaviour inside a group of 19 agents having normal swarm behaviour.

In [21] a situation analysis technique is proposed. This method uses the sensorial information and context data provided by a target robot to describe the current situation. Then, this information is matched with a set of pre-defined situations by means of matching algorithms such as Bayesian networks.

An example of fault detection and identification in the wheeled robot Pioneer I is reported in [22]. This work considers two abnormal conditions, a flat tire and an object stuck to a tire, which are identified by utilising Kalman filters. Another approach is adopted in [23], where the robot LAURON IVc is presented. This robot is able to detect seven different types of faults, ranging from mechanical coupling problems to the total loss of leg controller units. The walking machine status is a combination of many different internal states representing the hardware and software components of the robot. Each internal state has a maximum of three levels. Depending on one or more sensor values representing the condition of its hardware or software component, the internal states are classified into one of these levels. The range of each level is determined by using thresholds. According to the authors, these thresholds have evolved from many experiments over several years and from expert knowledge. A similar approach will be proposed in section 4.2.3 for the classification of abnormality levels of sensor readings. The difference here is that thresholds are used for determining differences between obtained and expected sensor values and not only for dividing the range of the sensor into a certain number of levels.

Refer to [24] for a structured and comprehensive overview of existing research on anomaly detection.

2.2. Self-Healing and Fault-Tolerant Robots.

As stated by [25] and [26] a self-healing system is one that is able to recover from the abnormal (or “unhealthy”) state and return to the normal (“healthy”) state, and function as it was prior to disruption. A few robots with self-healing capabilities are reported in the literature. For instance, the authors in [27] present a ROS (Robot Operative System) based robot that is able to restart its hardware modules after a fault has been detected. In this case, it is assumed that the abnormal state is generated by a software error. Therefore, the module will probably return to its normal state after being restarted. However, there are many situations where, although robots are not able to completely recover after a failure, they can compensate it and minimise its effects. This sort of robot can be more accurately referred as fault-tolerant systems, where faults are compensated in such a way that they do not lead to system failures [28]. As proposed by [29], the interactions among different components of fault-tolerant robots are depicted in Fig. 2.1.

A number of fault-tolerant robots can be found in the literature. For instance, the research in [30] considers the case where one of the 18 joints of a hexapod robot is locked. In this case, if the angle of the locked joint is known, then the new leg workspace is determined. Therefore, a stable gait is generated after calculating the appropriate coordinates for the new leg’s condition. Another example, reported by [31], is a four-legged robot that is able to automatically synthesise a predictive model of its own topology (where and how its body parts are connected) through interaction with its environment. This model is used to synthesise successful new locomotive

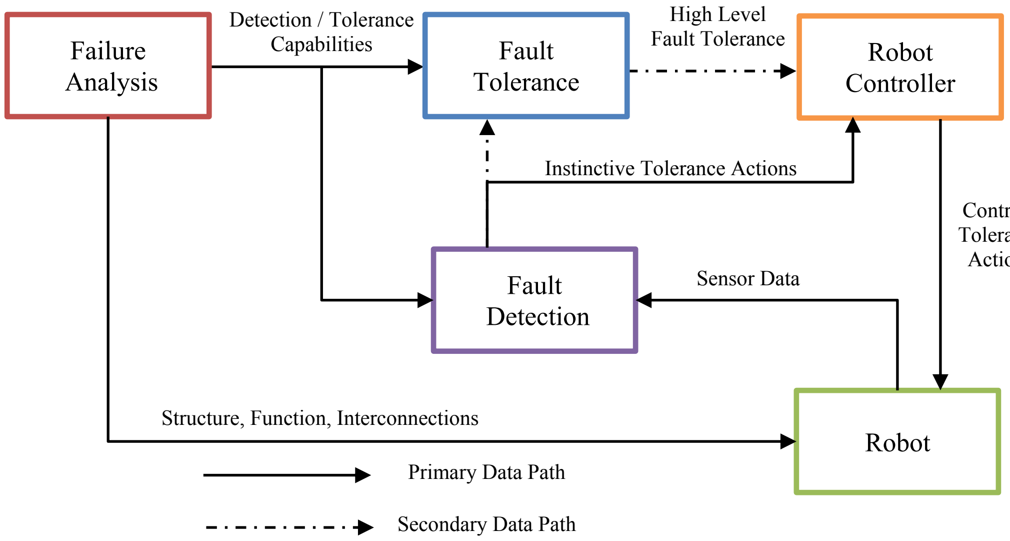

Failure Analysis
Detection / Tolerance Capabilities
Fault Tolerance
High Level Fault Tolerance
Robot Controller
Instinctive Tolerance Actions
Control / Tolerance Actions
Fault Detection
Sensor Data
Structure, Function, Interconnections
Robot
Primary Data Path
Secondary Data Path

Figure 2.1. Algorithm Interaction in Fault Tolerant Robotics.

behaviour before and after damage. The same self-modelling approach is used in [32]. The algorithm proposed here does not identify the damaged parts but it implicitly searches for efficient behaviours that do not use them. This technique was evaluated on a hexapod robot that must compensate for leg removal, broken legs and motor failures.

Another example of self-modelling oriented to self-compensation is reported in [33]. Here, the authors propose a self-modelling algorithm which is used for the generation of forward locomotion in a four-legged robot. When a part of one of the robot's legs is removed, the robot adapts its self-models in order to generate alternative gaits.

Although self-modelling is not a subject of this thesis, methods for the autonomous generation of compensatory measures are proposed. These methods are similar in the sense that both generate patterns of robot movement or behaviour until some goal is achieved. A difference with the work found in the literature is that in the self-compensation methods proposed in this thesis, abnormal sensorial information is utilised for guiding or constraining the range of movements of the robot. Therefore, the time necessary for achieving the compensatory behaviour is potentially reduced.

In this thesis, the generation of random behavioural patterns is only utilised as a last resort.

A different fault-tolerant system is presented in [34]. In the method proposed here, a correct configuration of the system (which leads to correct behaviour) is defined by constraining the system. When a failure arises, one or more of the constraints are violated. As a result, the system calculates a new configuration, such that the constraints hold again.

Researchers in [35] used a model-based failure detection and isolation (FDI) method for fault detection in robotic manipulators. As stated in [36], the FDI problem is addressed firstly by processing input/output data sets in order to detect the presence of various faults and to isolate them. Then, the system control is reconfigured in order to compensate for the negative effects of the fault and restore performance. The authors in [35] proposed a FDI filter based on a smooth velocity observer. This kind of observer-based methods aim at generating some residual signals such that each residual is sensitive to a group of faults. Then, logical combination of residuals can ultimately lead to localisation of faults.

A bond graph model used to represent a quadruped robot’s locomotion is presented in [37]. This model of the system is used to generate the analytical redundancy relations, which are then evaluated with actual measurements to generate residuals. These residuals are used to perform structural fault isolation. Once the fault list is updated in the equipment availability database, an automaton selects the best option to reconfigure the system such that the given control objectives are achieved. The fault-tolerance here is obtained by redundancy. This means that the robot is provided with spare modules that are not used during normal operation. When a fault is detected, the system is reconfigured such that faulty modules are replaced by their corresponding spare modules. This redundancy approach of fault-tolerant system is commonly found in modular and self-configurable robots [38].

The methods that will be proposed in this thesis also consider the isolation of faulty components of the target robot. For instance, once a faulty sensor has been identified, its readings are no longer considered. In Chapter 5, a labelling system that identifies faulty components of the target robot will be proposed. Although the proposed self-compensation methods were not tested in a robot with hardware redundancy, no difficulty for this kind of implementation is anticipated. Because no redundancy was considered, the experimental robot loses part of its functionality once a fault is generated in one or more of its components. However, the methods to be proposed in this thesis consider function redundancy for the compensation of abnormal situations. These functions can indicate to the robot the course of action to be taken when failures anticipated by the robot's designer are faced. This is similar to information provided by evolution to living creatures, which can immediately compensate dramatic events (e.g. the loss of a leg) without requiring a learning process. As a result, the target robot adapts its behaviour instead of replacing the faulty hardware module; and continues with its mission although its functionality and, in most cases, its performance has been degraded.

2.2.1. Autotomy in Robots

There are many examples in nature of organisms that shed parts of their body in order to escape from a predator or a fouled moulting event. This process is called autotomy [39] and [40]. Once an organism performs autotomy it is able to immediately adapt its behaviour to the new configuration of its body [41]. For instance, an arthropod which has autotomised a leg can immediately adapt its gait (without a learning process) in order to maintain stability when it is walking. The literature contains a few examples of fault-tolerant robots, which emulate this capability.

In [42], an eight-legged robot SCORPION suffering leg loss is presented. Its fault tolerance was tested by removing different combinations of leg pairs. Then, an arachnid gait and three different insect gaits were utilised to evaluate, in terms of speed and accuracy, the performance of the robot walking in a straight line. The results showed the importance of stability for the accuracy and even the mobility of the robot. On average, when the four gaits previously mentioned were utilised, only 15 of the 28 possible combinations of removed leg pairs were stable. These combinations allowed SCORPION to walk with some deviation from a straight line. As a result, the most stable gaits were obtained when the middle leg pairs were removed. By using this configuration, the best performance was exhibited by the tripod gait. However, the remaining tested gaits showed the best performance when different legs pairs were removed. Hence, the authors stated that more experimentation is necessary in order to determine the best gait for their robot.

Inagaki [43] has considered the case of a hexapod with one leg disabled and the generation of a gait for the five remaining legs. This work proposed an asymmetrical gait, composed of two combinations of leg phases, with a high stability margin and a duty factor in the range 0.6 to 1. A problem with this gait is that it has a fixed leg configuration and leg phase. Therefore, it is not adaptable to unexpected situations that can arise when walking in rough terrain. An analogous publication [44] considers a hexapod robot from which a leg has failed and been removed. The authors here propose a sliding system for removing the damaged leg and rearrange the remaining legs in order to improve the static stability of the robot. Then, an alternative gait suitable for the new robot's configuration is presented. Unfortunately, this approach was not tested in a real robot and only the general idea of the sliding system was presented. Following this publication, in [45] the authors present a survey of usable alternative gaits for hexapod and octopod robots with one or more severed legs. Another model of a hexapod robot suffering a leg loss and an octopod robot undergoing two leg losses is reported in [46]. In this work, gait parameters such as speed and pitch angles are heuristically modified until a stable gait is achieved. As occurred in [44] and [45], only simulation results are provided.

The work most closely related to the aims of this research has been developed by the Institute of Computer Engineering at the University of Luebeck, Germany. This group has presented its robot OSCAR (Organic Self-Configuring and Adapting Robot) [47], a six-legged robot that is able to adapt its gait after one or two legs are manually deactivated by the operator (simulating leg amputation). OSCAR detects the amputation thanks to missing feedback, such as ground contact or servo feedback. The robot's gait is generated moving the legs between their Anterior Extreme Position (AEP) and Posterior Extreme Position (PEP) during the stance phase. Once the legs reach their PEP, the swing phase begins; the legs are lifted and moved to their AEP. Then they are lowered and the stance phase starts again. The leg coordination is controlled by only one rule: a leg is only allowed to swing, when its neighbouring legs perceive ground contact. In the case of amputation, if the neighbouring leg is amputated and sends no ground contact signal the next leg will compensate for the missing one. After using this scheme, the walking robot exhibited instability even when only one leg was deactivated. In the case where two legs were lost, the robot was only able to walk with greatly reduced efficiency when two opposite legs were deactivated. When the relative position of these legs was not opposite each other, the robot was not able to walk at all.

After the first version of the robot OSCAR, a number of publications show the progress in this project. In [48] the authors included turning and curved walking. The turning movement is a rotation on the spot. It is accomplished setting the AEP and PEP of each leg to the same value and moving all the legs in the same direction (clockwise or counterclockwise). On the other hand, the curving walking is achieved by shortening the stance trajectories of the legs at the inner side of the curve and enlarging the stance trajectories of the legs at the outer side. This is accomplished by changing the respective AEP and PEP values. In the case of leg amputation, the AEP and PEP are altered in order to fill the gap left by the missing leg. This approach effectively reduced the time required by the robot for reaching a goal position. Unfortunately, the analysis of the results in [48] only considers the deactivation of one leg at the time. It is not clear how the stability of the robot is affected during the turning movement when one or more legs are amputated. In [49], a series of rules is presented for a uniform distribution of legs once one or up to three legs have been removed. The paper in [11] describes the anomaly detection system of the robot OSCAR. This is based on fuzzy logic and clonal selection [13]. The system adapts the weights of its fuzzy rules, which provides a kind of short-term memory. It would allow the robot to perform better when facing previously learnt situations. However, this learning process has not been implemented yet and the authors in [11] only present a comparison between fuzzy rules with static and dynamic weights in their experimental results. A summary of the current state of this project is reported in [50].

In this thesis, an adaptable gait generation method will be proposed. This technique will be utilised by the experimental robot for compensating the loss of one or more legs. The approach to be proposed selects and calculates the final coordinates of a robot's stance and swing legs by maximising the stable mobility of the robot in the direction of locomotion. More details about this technique will be presented in Appendix B. In comparison, the gait generation method that will be proposed in this thesis seems to be more complex than the one incorporated into the robot OSCAR. While the former deals with combinations of shapes created by the robot legs' tip and geometrical intersections generated as the legs move, the latter uses simple rules that modify the AEP and PEP of the legs and generate a new gait after a leg loss. Nevertheless, the gait generation method that will be proposed in this thesis shows better results than the one achieved by OSCAR in terms of stability. In cases were OSCAR is unable to maintain stability (e.g. in some cases where only one leg or when two non-opposite-legs are lost), the proposed adaptable gait generation algorithm shows no stability issues.

The Robot Leg Amputation Mechanism (R-LEGAM) is presented in [51] and [52]. This mechanism is used when the robot experiences problems in some of its legs and there is no other solution but to amputate a leg. Depending on the circumstances of the amputation, this action could give the robot an extra chance of reaching its goals or simply extend its functional life. As well as in the biological counterpart [53], the disadvantages are that the walking speed is reduced and the energy requirements are larger in comparison with the intact robot. R-LEGAM is integrated with springs in each amputation mechanism, which are used for ejecting the amputated legs from the robot's body. Each compressed spring is released by a small mechanical latch that is controlled by a small additional servo motor.

In section 3.4, a novel leg release mechanism will be presented. The proposed system is more compact and lighter than R-LEGAM because no extra servos are necessary for detaching a leg. On the other hand, the leg ejection process is slower in the proposed system, which might be a disadvantage in some applications.

2.2.2. Force Detection and Reflex Reaction

The control of walking and posture in insects is affected by the integration of information about external forces applied to the insect's legs. These forces are detected by receptors called campaniform sensilla, which also adjust the motor output to changes in load. Since terrain features are perceived by the locomotor system as a pattern of load distribution among the insect's legs, this information is essential for the adaptability of a walking insect to rough terrain. A deeper analysis of how far load detection is related to flexibility in adaptation of posture and locomotion can be found in [54].

Load detection is another capability emulated by biologically inspired robots. El Sayed Auf, et al. [55], as a part of the OSCAR project, introduced a decentralised controller approach. This involves the measuring of external forces affecting each of a robot´s joints and combines active compliance with a step-performing reflex. The external force is calculated using the existing correlation between the force applied to the servo and its electrical current consumption. When an external force is perceived by the servo, its actual and goal positions start to differ. If this difference is larger than a determined threshold, then the active compliance moves the servo in the direction of the applied force. This principle is used in the proposed step reflex. Here, if a robot's leg is pushed backwards or forwards, to the limit of its working space, then the robot steps in the direction of the applied force. Therefore, the movement of the pushed leg relative to the robot is in the opposite direction to the applied force. The reflex behaviour has been proposed in previous papers as well. For instance, in [56] Espenschied et al. proposed an elevator and a searching reflex. The elevator reflex acts when a swinging leg encounters an obstacle, reversing the motion of the leg and lifting it higher before continuing to swing forward. In this case, the external force is the reaction force opposite to the force with which the leg pushes the obstacle. The searching reflex is applied when the terrain support is missing (e.g. if there is a hole) or lost (e.g. if part of the terrain slides away from under a leg). It consists in a circular foot movement that progressively increments its radius a fixed number of times or until the foot finds an acceptable foothold.

In this thesis, reflex actions are understood as responses to pain (or extreme sensorial values). This is discussed in subsection 4.1.2, where an experiment with extreme values of a servo motor force sensor is presented. In particular, the elevator and the searching reflexes presented in [56] were not implemented into the experimental robot. However, similar tasks are utilised as research actions for validating or rejecting the robot's theories about the source of an anomaly.

2.3. Sensorial Information in Walking Robots

2.3.1. Gait Adapatability

A number of walking machines intended for rough terrain have been equipped with sensors in their legs. This information provides feedback that has improved the gait stability of these robots. For instance, the authors in [57] have embedded joint angle and joint torque sensors in the legs of the hexapod walking robot DLR Crawler. This information is used by a 3D odometry algorithm that allows the estimation of pitch and roll angles. Similarly, the robot HITCR-II, presented in [58], is equipped with joint torque sensors. An extra feature of this robot is the omni-directional force sensors located in the robot's feet. In addition, a force-position controller is employed in the balance stabilisation method presented in [59]. This technique considers the impact dynamics of walking robots. As a result, external perturbations are compensated by estimating impulsive forces in real-time.

Some authors have considered problems that robots face when walking on rough terrain. For instance, the authors in [60] equipped their hexapod robot ASTERIX with touch sensors on the tip of its legs. The robot utilises these sensors to ensure that all of its stance legs are in contact with the irregular terrain. Then, the elevation of the robot's swing legs is calculated by considering that the plane formed by the tip of these legs must be parallel to the stance legs' plane. In addition, problems related to foothold selection for reducing slippage and improving static stability when walking on rough terrain are addressed in [61]. Furthermore, an adaptable gait has been reported in [62]. This gait allows robots to maintain stability when walking over 25-degree slopes or when an external force of up to 58% the robot weight is applied.

The authors in [63] use the software architecture ORCA (Organic Robot Control Architecture) and Learning Classifier Systems (LCS) as methods for the autonomous tuning of an algorithms' parameter. The LCS analyse the output of a module and apply changes to algorithm parameters or exchange parts of the algorithm. In operation, the LCS further employ unsupervised machine learning to determine, which actions have most improved the algorithm performance. When this approach is utilised in gait generation algorithms, the resulting gait is able to adapt to changes in the robot's environment.

Similarly to the works presented in [57] , [58], [59] and [60], the experimental robot utilised in this thesis has been provided with a number of sensors. These sensors supply information about torque and position of the robot’s servo motors, and force applied on the tip of the robot’s legs. In particular, data provided by the leg tip force sensors is utilised for calculating the position of the robot’s centre of mass. This is a parameter of the gait generation algorithm that will be proposed in this thesis and presented in Appendix B. The calculation of the robot’s centre of mass allows the gait generation algorithm to improve gait stability when the robot’s centre of mass is shifted (e.g. because an object has been deposited on top of the robot). A larger degree of adaptability is provided by the ability of the gait generation algorithm for to generate a stable gait in robots after a series of leg losses. This feature differentiates the gait generation algorithm that will be proposed in this thesis, from other adaptable gait generation algorithms found in the literature.

2.3.2. Gait Optimisation

Researchers have also considered the optimisation of energy consumption in walking robots. An example of this can be found in [64], where the hexapod robot SILO6 is used for comparing the energy consumption of insect and mammal leg configurations. Another hexapod robot is utilised in [65] for evaluating the energy required by a set of wave-gaits with different duty cycles. This work is extended in [66], where the energy requirements during turning are analysed. In addition, Jin et al. [67] propose a torque distribution algorithm to minimise the energy consumption of hexapod robots. Furthermore, a report about the use of power efficiency degradation of robots for health monitoring and fault detection purposes can be found in [68].

In this thesis, two conclusions commonly found in the literature regarding gait efficiency have been corroborated. The gait generation algorithm that will be proposed in this thesis was designed to maximise stability and distance travelled. As a result, a tripod gait was generated for hexapod robots, which is commonly considered as the fastest gait in the literature. In addition, a wave gait was generated for quadruped robots, which is considered as the most stable gait in literature. Both gaits have been favoured by natural evolution and are found in many species of living creatures. The second corroborated conclusion is that the speed of locomotion is drastically reduced in hexapod living creatures after a leg loss. This speed reduction was also experienced by the experimental robot. Results in Table B.20 show a reduction of 56% in the average distance per gate cycle covered by the experimental robot after a leg loss.

2.4. Clustering

Robots are commonly provided with sensors that have different resolutions and ranges; they also measure different physical phenomena. As a result, the sensorial information provided by robots typically generates a dynamic multivariate space. Some authors have classified these complex data by using classical clustering methods. For instance, [69] use the fuzzy c-means clustering algorithm [70] in the classification of sensors readings provided by a robot team. The authors in [71] report the use of classical Bayesian clustering as a part of their method for online classification of robot sensor data. In the two previous examples, the authors have used techniques for reducing the data dimension.

A difficulty commonly found when using the original fuzzy c-means or later optimised versions [72] is that these algorithms require initial values that are often unknown. This problem has been addressed by a group of researchers [73] who have modified the original algorithm in order to automatically calculate some of these values. As a result, better classification is obtained, but sacrificing simplicity and speed.

Several researchers have preferred to develop their own clustering methods. For instance, [74] have proposed a clustering algorithm that exploits the time-dependency between sensor readings samples. In [75], robotic sensorial information has been clustered by using non-parametric statistics.

A disadvantage of classic clustering methods commonly found in the literature is the large number of calculations they require. This number is often incremented as the robot's sensors collect more data points. In comparison, the self-diagnosis methods that will be proposed in Chapter 4 utilises a simple and fast approach for data classification. New sets of data are associated with an existing group after establishing that the Euclidean distance between their centroids is less than with other groups. In addition, a new group is only created after a correct diagnosis is not possible with the existing groups. The classification methods that will be proposed in this thesis only store the centroid of each group and the number of data points. This drastically reduces the memory requirements of the system and, because the system deals with less data during the classification process, processing time is also reduced.

2.5. Robot Locomotion over Rough Terrain

The performance of robots in the real world is currently compromised by their limited degree of adaptability when dealing with a complex and uncontrolled environment. In the locomotion area, how robots can travel autonomously over rough terrain is an unsolved issue that has focused the attention of an increasing number of researchers. Since an early stage, different methods of robot locomotion, such as wheels, tracks, legs, etc., have been employed. Wheels and tracks are easy to control and they are also inexpensive solutions. However, it has been estimated that vehicles with this kind of locomotion cannot access about half the earth's land surface [76]. On the other hand, walking machines offer the following advantages:

- Adaptive to uneven terrain.
- Use isolated footholds.
- Provide active suspension.
- Environmental effects of legged vehicles are less than wheeled or tracked vehicles.

The disadvantages of legged locomotion are the complex control and mechanical design that makes legs slower and more prone to failure than tracks or wheels [77]. However, walking machines have a wider range of movements. This is necessary to overcome different terrain irregularities that they could find in their path. For instance, legged robots can step over, climb or jump obstacles comparable with their own size ([78], [79] and [80]). They are also able to place carefully each leg on the ground and to decide whether to back up or advance when walking on intermittent terrain [81]. All of these features make legs more suitable for locomotion over rough terrain.

A number of publications analyse the different types of robot locomotion. For instance, [77] assesses the locomotion performance of all-terrain rovers whereas [82] compares wheels and tracks from the traction perspective. Another analysis [83], including biological creatures and robots, compares the specific power (the vehicle power normalised by a product of the gross weight and the speed), with locomotion speed. Furthermore, a survey including tracked, wheeled and legged locomotion can be found in [84]. Two additional surveys, one including walking machines with two, four and six-legged propulsion; and the other involving compliant legged robots are presented in [85] and [86], respectively. In addition, an amphibious robot able to walk and swim is introduced in [87]. Examples of other modes of robot locomotion, such as jumping, rolling and crawling are reported in [88].

2.6. Biologically Inspired Walking Robots

Observing the many examples of successful legged locomotion through rough terrain found in nature, robotics engineers have included some biological principles in the mechanics and control of their designs. However, due to the high complexity of biological walking organisms and the impracticality of emulating nature’s actuators and sensors, only some of the ideas have been currently implemented in the control architecture of real robots [89]. A number of biologically inspired approaches to robot locomotion are found in the literature, for instance, Quinn et al. ([90] and [91]) have developed a series of four robots (named Robot I, II, III and IV) each progressively more exact copies of cockroaches. Robot I, has demonstrated insect-like gait generation walking over flat surfaces using a neural network gait controller.

Robot II has a distributed control system which includes a central posture controller, leg reflexes and a gait generator based on biological studies on leg coordination mechanisms in the stick insect [92] [93]. This robot is capable of walking, turning and walking sideways over uneven terrain. Robot III is self-contained and utilises air cylinders as actuator, which have a larger force to weight ratio than electric motors. It is capable of walking, running, turning and climbing over small obstacles with an agility that resembles a real cockroach. Finally, Robot IV employs braided pneumatic actuators which optimise the energy consumption while adding passive compliance to the robot's legs.

Another series of robots, Lauron (Legged Autonomous Robot, Neural Controlled) I, II and III have been developed using the stick insect configuration and 18 dc motors. Lauron I [94] uses a hierarchical neural control architecture, which selects a gait from a discrete set. It utilises an insect model that provides training samples for a supervised leg control learning algorithm. Lauron II [95] includes a stereo vision system and additional sensors for obstacle detection. The third robot, Lauron III [96], incorporates leg reflexes like collision reaction and searching for ground contact. It also controls its body height and considers gravity influence in the robot's gait. Lauron III has a hierarchical control; a central unit controls the leg coordination whilst individual controllers on the legs manage the leg reflex behaviour. This robot is capable of walking in real world environments such as forests and mountains.

Gregor I [79] is another biologically inspired robot. It attempts to emulate some cockroach attributes such as self-stabilising posture and specialised functions of the legs. Its gait is generated using neural networks based on the theory of Central Pattern Generators (CPG). These are neural circuits found in both invertebrate and vertebrate animals that can produce rhythmic patterns of neural activity without receiving rhythmic inputs [97]. Refer to [97], [98], [99], [100], [101], [102], [103], [104] and [105] for a deeper explanation and additional examples of CPGs applied to locomotion control of legged robots. Thanks to its sprawled posture and piston-like actuation for rear legs, Gregor I is able to walk at 0.1 body length per second and climb obstacles up to 170% of the height of its mass centre. The third version of this robot, Gregor III [106], is controlled using an extension of Walknet ([107] and [108]), a decentralized network based on studies of the stick insect which by means of local reflexes produces the control signals utilized to generate the gait in hexapod robots. The performance of Gregor III was tested in simulation and using the actual robot. In the simulated environment, the robot generates a gait that maintains its stability at low speed. When the locomotion speed reaches its maximum value, the robot adopts a tripod gait. In the real robot, differences between the expected and the real joint angles arose. Those were attributed to the effect of friction on the robot's low torque motors. A more recent example of gait generation using Walknet is the robot HECTOR introduced in [109].

Dante II [110] is a robot designed for exploring inside volcanic craters. Its gait was programmed with 24 asynchronous processes or behaviours, which allow the robot to stand, step, walk and also control its posture. The behaviours are inhibited or activated according to the leg position, leg phase and additional information supplied by the sensors. Dante II descended into the crater of Mount Spurr and was able to perform exploration being tele-operated during the whole process. However, during the exploration it tipped over and had to be rescued.

Porta et al. [111] have proposed an adaptable gait for hexapod robots. Gait generation uses a simple rule: "Never have two neighbouring legs raised from the ground at the same time". By using this rule and considering the relative position of the legs with respect to their posterior extreme position, the robot central controller calculates the order in which the legs must be raised, generating the gait. The experimental results showed that the robot adapts its gait according to the terrain over which it is walking, generating a tripod gait when it is traversing flat surfaces.

Gait pattern self-synchronisation for walking robots is presented in [112]. This self-synchronisation allows the robot to adapt its gait in certain situations, for instance, when a leg must modify the length of its stance/swing phase in order to overcome an obstacle. The method uses two rules. First, when one leg shortens or lengthens its own gait, its neighbouring legs imitate this change, adapting their gait as well. The second rule establishes that the prolongation occurs only during the stance phase whereas the shortening takes place only during the swing phase. As a result, the gait of all the robot´s legs adapts to the changes of their neighbouring legs generating a synchronized robot gait pattern without using global coordination. The synchronisation time depends on the final gait length to which the legs synchronise their swing and stance phases. Gait stability during the synchronisation process depends on how often such synchronisation is initiated by external influences during a short period of time.

A more complex model of a biologically inspired walking robot with 22 degrees of freedom is presented in [113]. Here, the gait is generated by means of a hierarchical recurrent neural network. Whilst the simulation results proved to be satisfactory, the model was not applied to a real robot. Another example where hierarchical neural networks are involved can be found in [114]. In this work, the gait generation of the walking hexapod robot AMOS II is achieved by means of a CPG and an adaptive neural control. Furthermore, the authors in [115] propose to improve the long-term stability of the CPG by introducing a new type of neuron called a single unit pattern generator.

Additional approaches for both gait generation and optimisation are genetic algorithms ([116], [117] and [118]) and reinforcement learning ([119], [120] and [121]). These methods intend to mimic the biological processes of evolution and learning, respectively. In many cases, a combination of methods is utilised. For instance, [122] uses a CPG for the gait generation and a mixture of genetic algorithms and reinforcement learning for optimisation of the gait. The works presented in [123] and [124] utilise evolutionary algorithms to autonomously determine and optimise gait parameters in a Sony entertainment four-legged robot AIBO. Researchers in both publications report that the resulting evolved dynamic gaits are faster than those obtained by hand-tuning the gait parameters.

A literature review considering genetic algorithms for gait optimisation can be found in [125].

A five-legged robot, inspired by the sea star, has been reported in [126]. Here a set of fuzzy Q-learning modules is utilised to generate different actions. The resulting behaviour is employed for obstacle avoidance in a cluttered environment.

In some publications, fuzzy logic has been involved in the gait generation. For instance, in [127] it is reported that different hexapod wave gaits have been generated by means of appropriate fuzzy rules. Similarly, in [128] the swing legs of a hexapod robot are selected by using this technique.

Similarly to other gait generation algorithms previously presented in this section, the one that will be proposed in Appendix B of this thesis is able to generate a gait that resembles gaits commonly found in living creatures. In cases where 6 legs are available, the proposed algorithm generates a tripod gait. This gait can be observed in hexapod creatures such as stick insects, flies, etc. Furthermore, in cases where 4 legs are available, the proposed algorithm generates a wave gait. This gait can be observed in a large number of quadrupeds (e.g. elephants). As in other approaches (e.g. [92] [93]), the proposed gait generation method is able to produce turning movement, rotation on the spot and allow the robot to walk sideways. An important difference between the proposed gait generation method and the ones presented in this section is that the latter fail to produce a stable gait after one or more leg losses.

2.7. Summary

This chapter has presented a selection of relevant publications addressing areas related to the investigation reported in this thesis.

First, a number of approaches to anomaly detection in walking robots have been presented in section 2.1. In some cases, these approaches failed to identify gradual changes in the sensor readings generated by a fault. In other works, fault detection was over-sensitive, diverting the robot from performing its mission in order to corroborate the existence of anomalies not originally affecting the robot's performance.

In Section 2.2, a number of approaches to self-healing and fault-tolerance have been introduced. Robots using these methods have been able to increase resilience to damage, increase lifespan and improve autonomy. Here, particular attention has been given to self-healing and fault-tolerant walking robots. In particular, works that provide walking robots with autotomy capabilities and reflex reactions have been described and briefly compared with the research reported in this thesis. Furthermore, the robot OSCAR has been presented in detail as the research most closely related with the one reported in thesis. Researchers working on this project are the only group (in the reviewed literature) that have developed a mechanical system able to physically release a robot's leg. A succinct comparison with the leg release mechanism to be presented in Chapter 3 has been presented here.

Publications dealing with gait adaptability and gait optimisation have been presented in section 2.3. A number of similarities and differences with the gait generation algorithm that will be proposed in Appendix B have been indicated. This is followed by section 2.4 where a selection of works addressing the subject of clustering has been discussed. The interest here has been focused on applications for the classification of robot sensor information that leads to the identification of anomalies. In addition, disadvantages of such methods regarding their high requirements of memory and processing time for large quantities of data have been pointed out and concisely compared with the classification methods to be proposed in this thesis. Next, studies dealing with the best ways of mechanical locomotion over rough terrain have been presented in section 2.5. Finally, biological approaches to gait generation in walking robots have been reviewed in section 2.6. These publications have addressed ways to adapt a robot's gait to different terrain features. In most of the cases, obtained gaits are similar to the ones found in living creatures. This coincides with the gait produced by the gait generation algorithm to be proposed in Appendix B. However, most of these publications did not considered the adaptability of the generated gait to extreme disturbances in the robot itself, such as the loss of one or more legs.

The next chapter will introduce the experimental robot where the self-diagnosis methods proposed in chapter 4 and self-compensating methods presented in sections 5.1 and 5.2 have been implemented.

Chapter 3
Experimental Equipment

This chapter introduces the experimental hexapod robot where the algorithms and techniques developed during this investigation have been implemented and tested. Firstly, aspects related to the design and physical construction of the experimental robot are addressed in section 3.1. Then, the distributed control of the robot is discussed in section 3.2. Next, the robot's sensors and their operating principles are presented in section 3.3. This is followed by section 3.4, where a number of gait parameters of the experimental robot are empirically determined. In section 3.5, a leg release mechanism, that emulates the autotomy capability of living creatures, is described. Finally, a summary of the chapter is available in section 3.6.

3.1. Experimental Robot Design

The methods and algorithms introduced in this thesis have been applied to the hexapod robot depicted in Fig. 3.1. This has generated the experimental results necessary for the validation of the proposed theory and techniques developed during this research. The experimental robot has been especially designed and built for this research at Monash University. The hexapod's body has been mainly built of ABS (Acrylonitrile Butadiene Styrene) plastic by means of a 3-D printer. This method facilitates the construction of more complex parts necessary for the development of some sensors and mechanisms of the experimental robot. At the same time, it accelerates the prototyping process in comparison with other tools such as milling machines. However, a few parts, those under intense stresses, have been made of brass or aluminium. In total, the robot is composed by more than 100 printed parts,

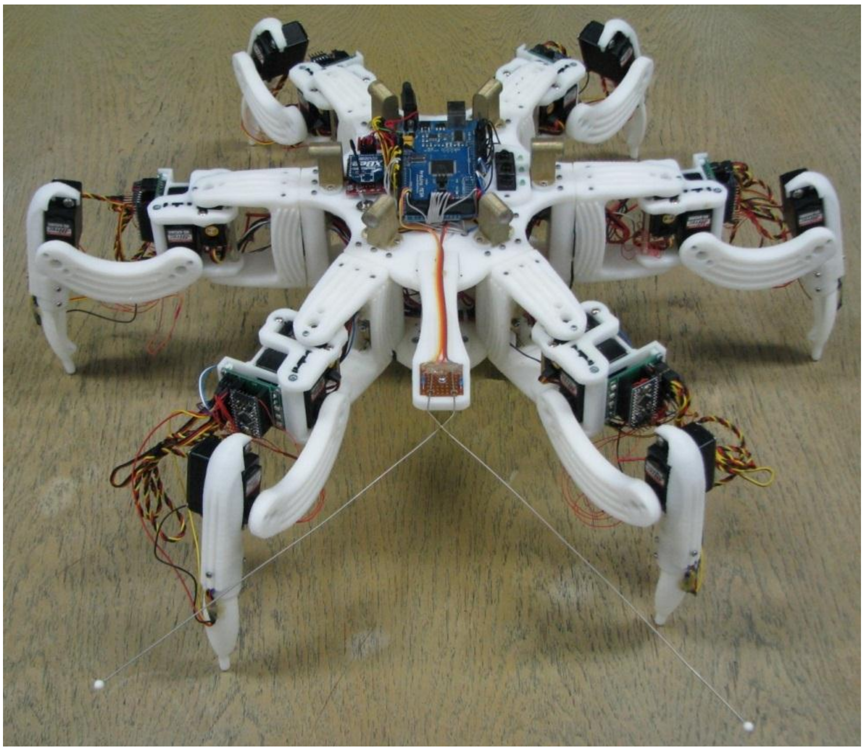


Figure 3.1. Experimental Hexapod Robot.

32 parts made of brass and 6 made of aluminium. These parts comprising the experimental robot are better represented by the exploded view of the robot's CAD (Computer-Aided Design) model shown in Fig. 3.2.

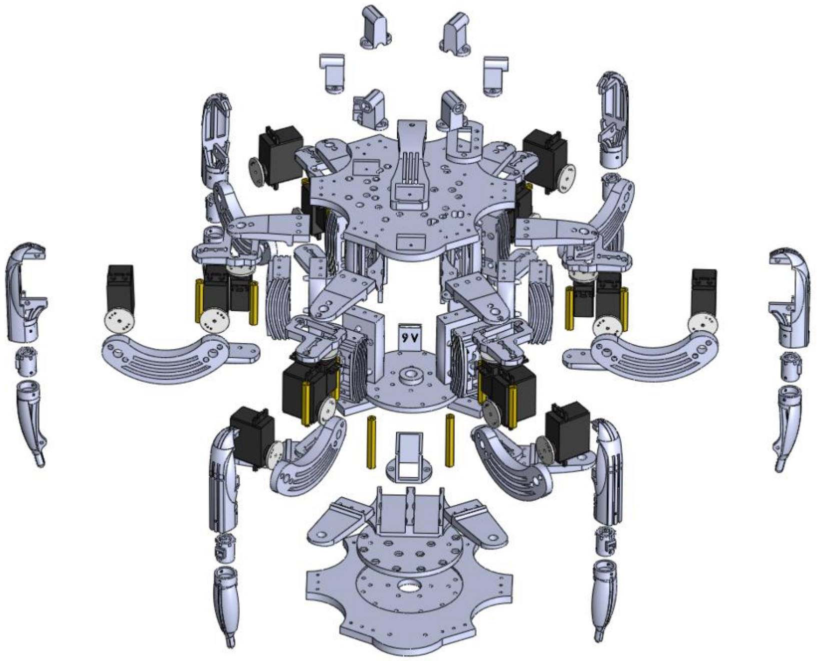


Figure 3.2. Exploded View of the Experimental Robot.

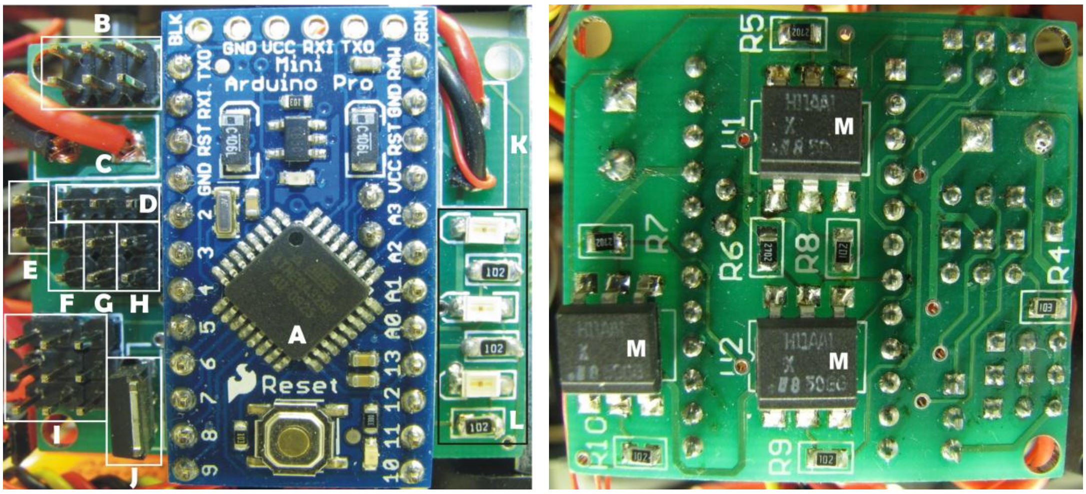
The robot's legs are attached to the robot core (i.e. the central part of the robot's body) at 6 points distributed in a circular fashion around the core. In this way, there is an equal separation of 60 degrees between any adjacent pair of legs. As each leg has 3 degrees-of-freedom, the hexapod has 18 degrees-of-freedom in total.

3.2. Robot Control

Control of the experimental robot is distributed among the robot's legs, the robot's core and an external computer. Both, the tasks performed by each control component and the interactions among them are explained in the following subsections.

3.2.1. Leg Control

The robot's legs are equipped with sensors that provide information about the force applied to the tip of the leg and also the position and force applied by the leg's servo motors. In addition, each leg controls its own servo motors, the release mechanism associated with that leg and communication with the robot's core. Legs are also responsible for local reflexes such as temporary disconnection of motors when potentially harmful stress or temperature levels are detected. The control of each leg is performed by means of an Arduino Pro Mini microcontroller, which is connected to an associated PCB (printed circuit board) specially designed for the project. A detailed front and rear view of the robot leg PCB is shown in Fig. 3.3. Refer to Appendix C for an electronic schematic of these boards. Basically, each leg PCB integrates the operation of the leg's microcontroller, the leg's sensors and communication with the robot's core control. As is shown in Fig. 3.4, the outputs of each leg microcontroller are the PWM (Pulse Width Modulation) control signals of the leg's servo motors, on/off signals for 3 multi-purpose LEDs and a signal that activates the leg release system (which will be introduced in section 3.4). These microcontrollers also interchange information with the robot's core microprocessor utilising an I^2C (Inter-Integrated Circuit) communication bus.



A: Leg Microcontroller. B: Force Sensors Inputs. C: Servos and Alloy Heating System Power Input.
D: Position Sensor Inputs. E: I^{2}C Communication. F: Leg Tip Force Sensor Input.
G: Alloy Heating Output. H: Additional Inputs. I: Servos Output.
J: FET (Activates the Alloy Heating System). K: PCB Power Input.
L: LEDs. M: Opto-isolators.

Figure 3.3. Leg Printed Circuit Board.

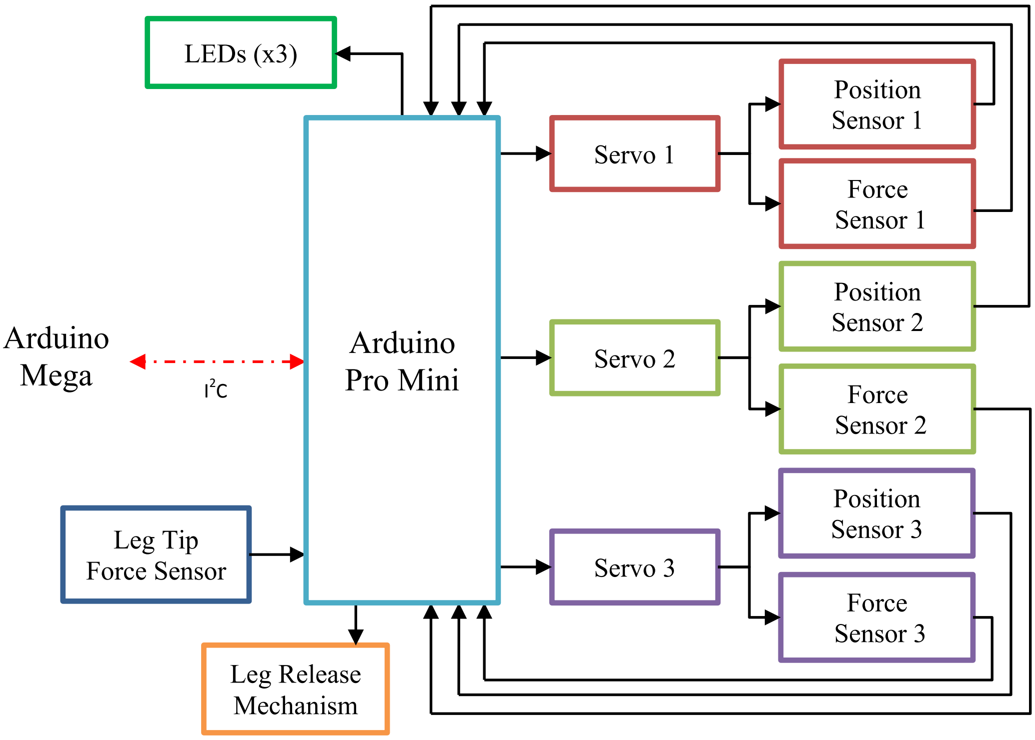


Figure 3.4. Block Diagram of Robot's Leg Control Signals.

3.2.2. Core Control

The robot core control is performed by an Arduino Mega microcontroller (located on top of the robot in Fig. 3.1). This microprocessor manages communication with the robot's legs and an external computer. As was mentioned in the previous subsection, communication with the leg's microcontrollers is achieved by using an I^{2}C bus. In addition, the robot wirelessly communicates with the external computer by means of a pair of XBee modules. These devices use the IEEE 802.15.4 standard for communicating with each other and the RS-232 standard for the communication with the external computer and the core microcontroller. As a result, a WLAN (Wireless Local Area Network) is established in a transparent way between the computer and the robot. Therefore, both machines are able to establish a wireless link using simple serial communication.

The core control receives commands from the external PC (Personal Computer) and coordinates their transmission to the leg microcontrollers. In addition, it receives the sensorial information provided by the robot's legs. This data is grouped with information provided by the robot core sensors and sent back to the external computer. A diagram representing signals connected to the robot core processor is depicted in Fig. 3.5. As can be seen, the robot core sensors consist of an

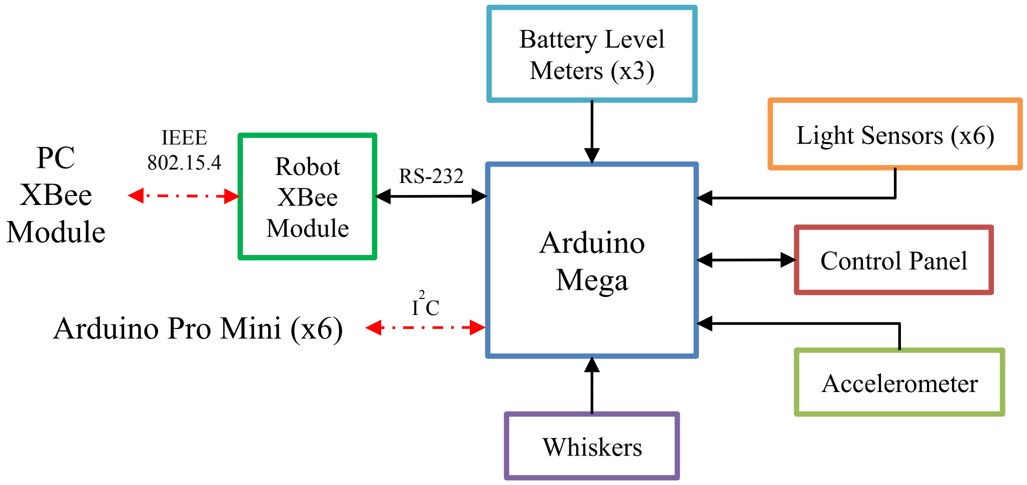


Figure 3.5. Block Diagram of Robot Core Control Signals.

accelerometer, 6 light sensors, 3 battery level meters and a pair of whiskers. There is also a simple control panel, which has proved to be useful for interacting with the robot during experiments.

3.2.3. External Computer

Due to the limited amount of flash memory available in the robot's microcontrollers, most of the algorithms developed during this research are executed in an external computer. Communication between the PC and the robot's core control is represented by Fig. 3.6. Here, it can be seen that the external computer is connected to one of the XBee modules and communication with the robot is established as explained in the previous subsection. The external computer is responsible for sending orders to the robot based on information provided by the core control. Among the tasks executed by the external computer are the robot's gait generation, evaluation of the robot's progress towards the completion of its task, detection of abnormal situations and selection of measures able to compensate for these situations.

3.3. Robot Sensors

This section describes the leg and core sensors, which provide information about the robot's status and environment.

3.3.1. Leg Sensors

As was mentioned in section 3.2.1, each leg of the experimental hexapod robot is equipped with 3 types of sensors that provide information about applied force and leg

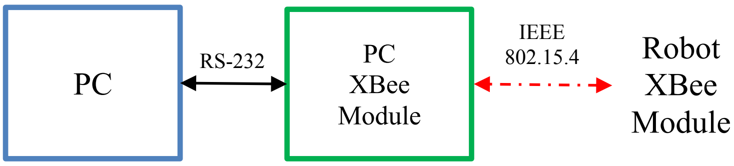


Figure 3.6. Block Diagram of External Computer Control Signals.

positions. The operating principles of these sensors are explained as follows.

- Force sensor: measures the pulse width of the PWM voltage signal applied to each of the leg's servo motors. The pulse width is directly proportional to error between the current and target servo positions. Therefore, if an external force prevents a servo from reaching its target position, then the pulse width associated with the resulting error can be used to calculate the external force magnitude. The microcontroller processing the force sensor output has been electrically isolated from this signal by means of an opto-isolator ('M' in Fig. 3.3).

- Position sensor: measures the voltage from the servo's internal potentiometer, which is directly proportional to the servo position. This is the most accurate feedback provided by the sensing system about the servo's actual position. When this information is used together with the kinematic model of the robot (provided in Appendix A), it is possible to determine the xyz coordinates of the leg tip.

- Leg tip force sensor: measures a voltage derived from an opto-interrupter, which is non-linearly proportional to the pressure, and inversely proportional to the traction, applied to the leg's tip. A CAD model of the opto-interrupter employed in this sensor and its infrared light beam is shown in Fig. 3.7. The

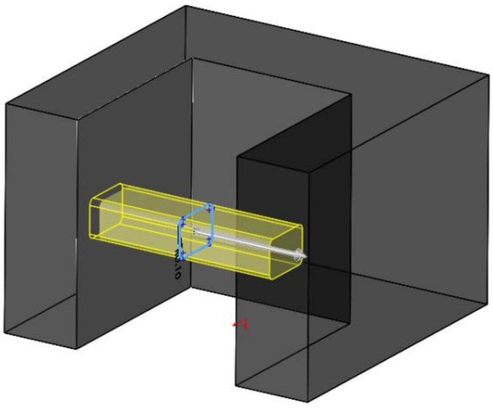


Figure 3.7. Opto-Interrupter CAD Model.

entire leg tip force sensor shown in Fig. 3.8 is 3D printed fully assembled as a single item. The voltage from the opto-interrupter reaches its maximum value when the whole light beam is sensed by the opto-interrupter's phototransistor. This voltage decreases when part of the light beam is obstructed. In the design of the leg tip force sensor this property has been exploited. Initially, a pin blocks around half the beam of the opto-interrupter. Movement of this pin in one direction (pointing upwards in Fig. 3.8) allows a larger part of the infrared beam to be sensed, while the movement in the opposite direction (pointing downwards in Fig. 3.8) blocks a larger part of the infrared beam. The relation between the force applied and the distance the pin is displaced depends on the amount of elastic deformation the force produces in six supporting beams connecting the internal and external cylinders depicted in Fig. 3.8. The elastic deformation can be calculated by means of Hooke's law expressed in Eq. 3.1.

$$\varepsilon = \frac{\sigma}{\mathrm{E}} \tag{3.1}$$

Where,

- σ is the applied stress.
- E is the Young's modulus, a material constant (2.3 GPa for ABS plastics).
- ε is the resulting strain.

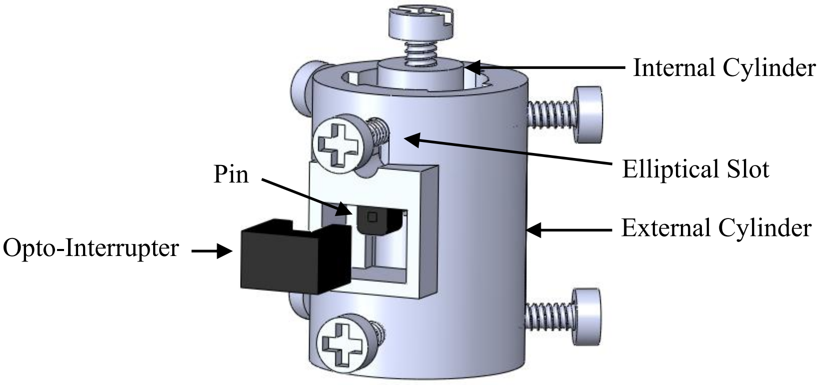


Figure 3.8. Leg Tip Force Sensor.

However, Eq. 3.1 is a linear approximation that is valid for a certain range of applied forces. The relation between the force applied to one of the robot's legs and the output voltage of the associated leg tip force sensor is shown in Fig. 3.9. Here it can be seen that this relation is non-linear and it can be better approximated by a second-order curve.

The leg tip force sensor shown in Fig. 3.8 joins the upper and lower parts of the robot's leg. This is more clearly represented by Fig. 3.10, where the exploded view of one of the robot's legs shows the position of the leg tip force sensor inside the leg together with the parts involved in the leg construction.

Whilst the lower part of the leg is fixed to the external cylinder, the upper part is allowed to move in the direction of the cylinder axis. The distance the upper part can be moved in relation to the lower part is limited by the major axis of elliptical slots in the upper part of the external cylinder. Because the

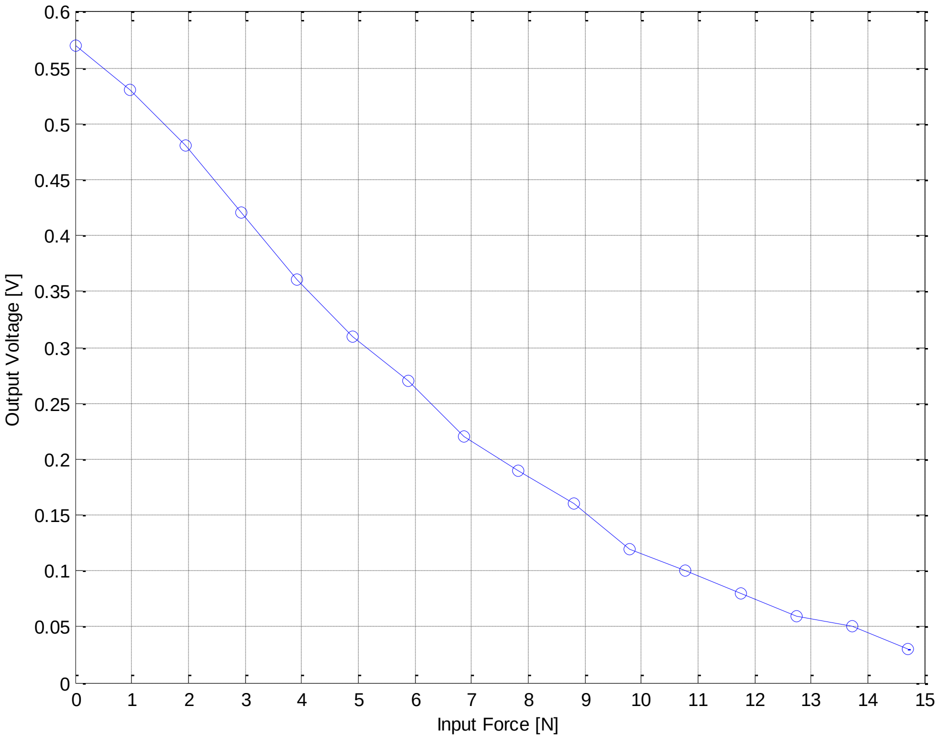


Figure 3.9. Non-linear Relation between Input Force and Output Voltage of the Leg Tip Force Sensor.

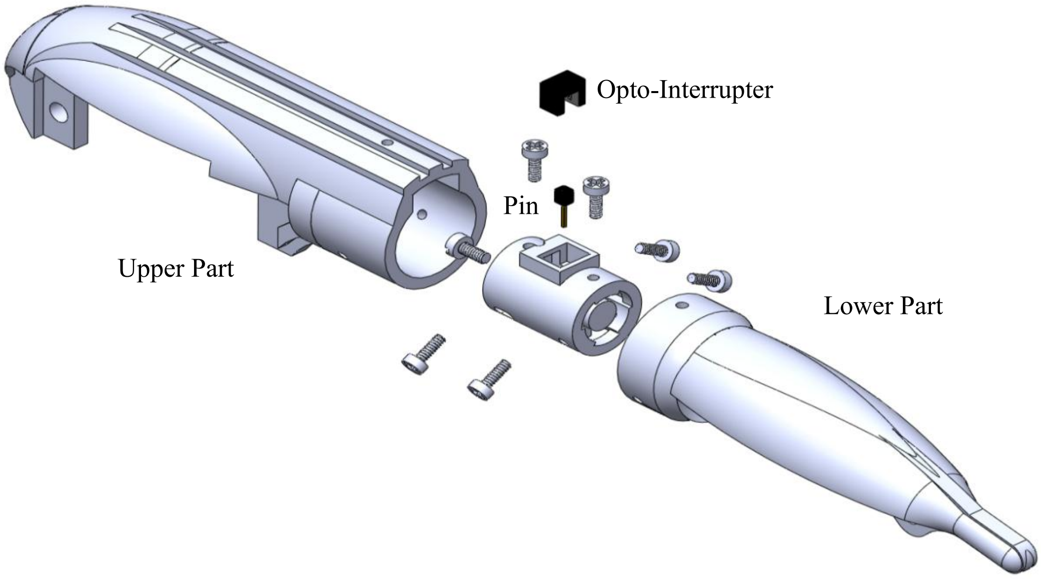


Figure 3.10. Exploded View of the Robot's Leg.

internal cylinder is bolted to the upper part of the leg, its relative position with respect to the external cylinder changes when a force is applied. As a result, the relative position of the pin (attached to the internal cylinder as shown in Fig. 3.11) with respect to the position of the opto-interrupter (attached to the cylinder as shown in Fig. 3.11) also changes, which alters the amount of light sensed by the opto-interrupter’s receptor.

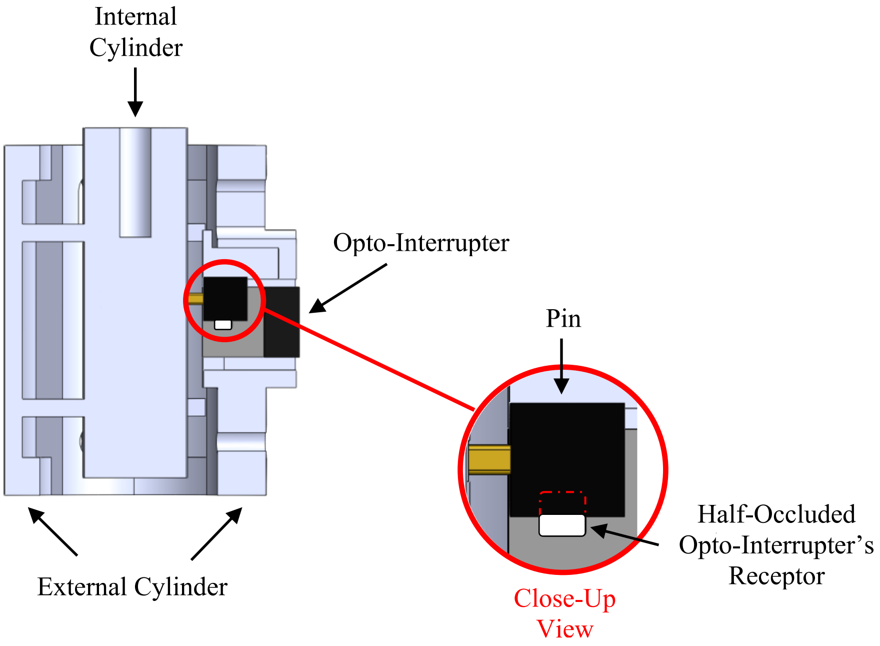


Figure 3.11. Cross-Section and Close-Up Views of the Leg Tip Force Sensor.

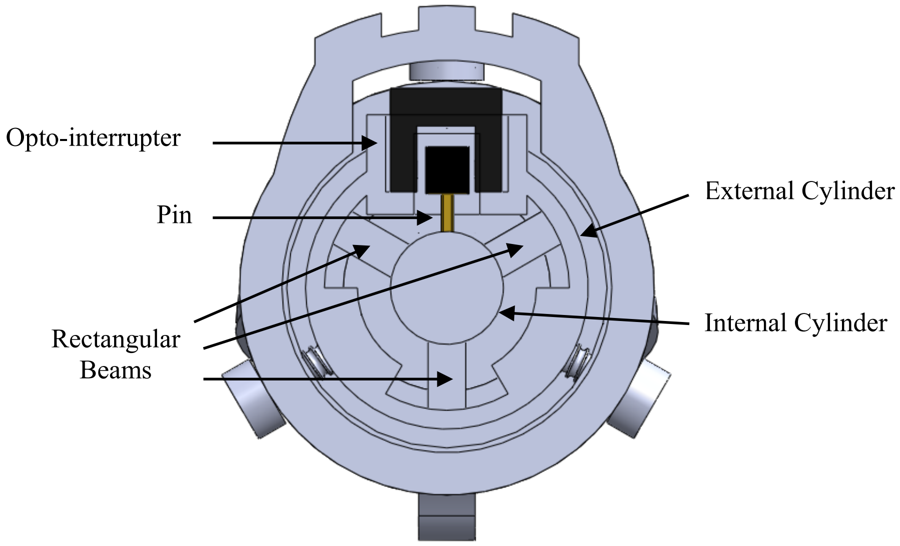


Figure 3.12. Cross-Section View of the Robot Leg and Leg Tip Force Sensor.

The external and internal cylinders are connected by means of 6 rectangular cross-section beams depicted in Fig. 3.12. These are elastically deformed when a force is applied in the direction of the common axis of the cylinders. Once the force is no longer applied, the beams return to their original shape. As a result, the cylinders together with the attached pin and opto-interrupter, return to their original position. Refer to Appendix C for an electronic schematic of the leg tip force sensor.

3.3.2. Core Sensors

The robot's core sensors are described as follows.

- Light sensors: These sensors are used by the experimental robot for finding a single light source. This simple task is the mission the robot has been assigned to perform and motivates the robot's locomotion. The light sensor output is the voltage on a potential divider consisting of a normal resistor and a photo-resistor (refer to Appendix C for the circuit schematics). This voltage is proportional to the amount of light received by the photo-resistor. There are 6 light sensors mounted on top of the robot. They are distributed in a circular fashion with a separation of 60 degrees between any pair of adjacent sensors. As a result, the robot is able to detect the magnitude and direction of a single light source.

The light sensor distribution, which matches the leg distribution, can be seen in Fig. 3.13. In addition, Fig. 3.14 shows a close-up view of a single light

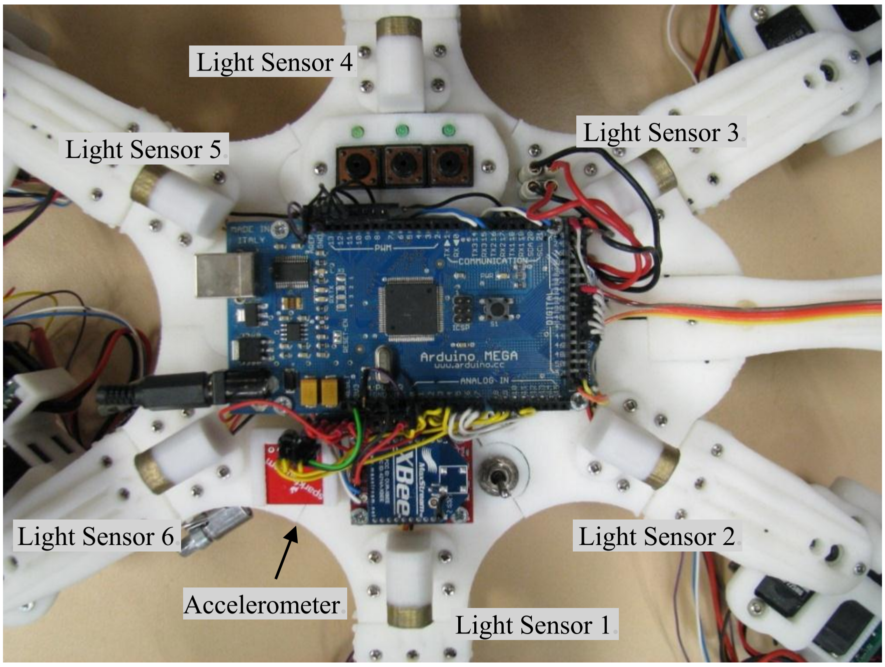


Figure 3.13. Robot Core Top View.

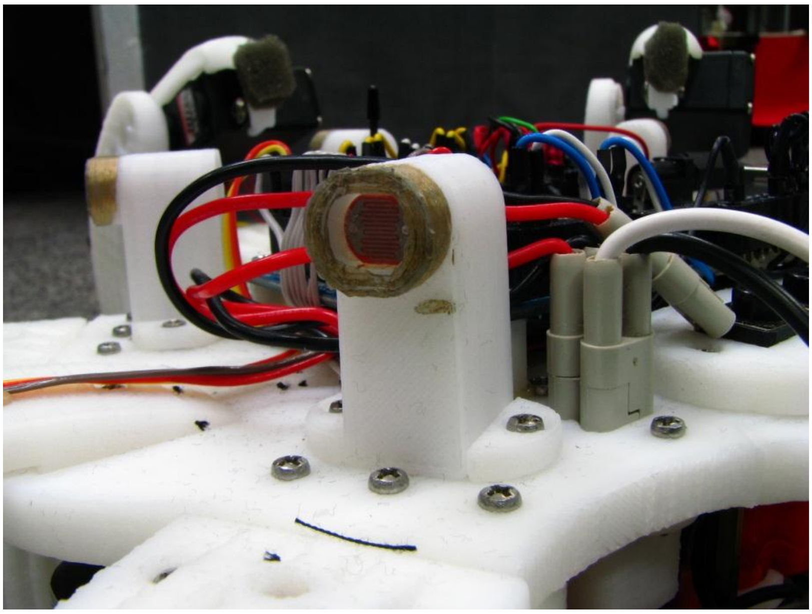


Figure 3.14. Light Sensor.

sensor. Here it can be observed how the photo-resistor is located at the end of a short tube, whose function is to partially collimate light falling on the phototransistor and thus reduce coupling effects between adjacent light sensors.

- Accelerometer: in the steady state this measures acceleration due to gravity. The robot's two-axis accelerometer shown in Fig. 3.13 is used to measure robot tilt and terrain inclination.
- Battery Level Indicators: these sensors measure the voltage on each of the robot's batteries. The robot is powered by three batteries. Firstly, a 6 volt battery is used for powering the leg's microcontroller and associated PCB. Then, another 6 volt battery powers the servo motors and the leg release mechanism that will be introduced in section 3.4. Finally, a 9 volt battery energises the electronics in the robot's core.
- Whiskers: these contact sensors (shown in Fig. 3.15) are used for detecting obstacles in front of the robot. Each whisker acts like a switch that makes contact whenever the whisker encounters an obstacle. Therefore, the whiskers

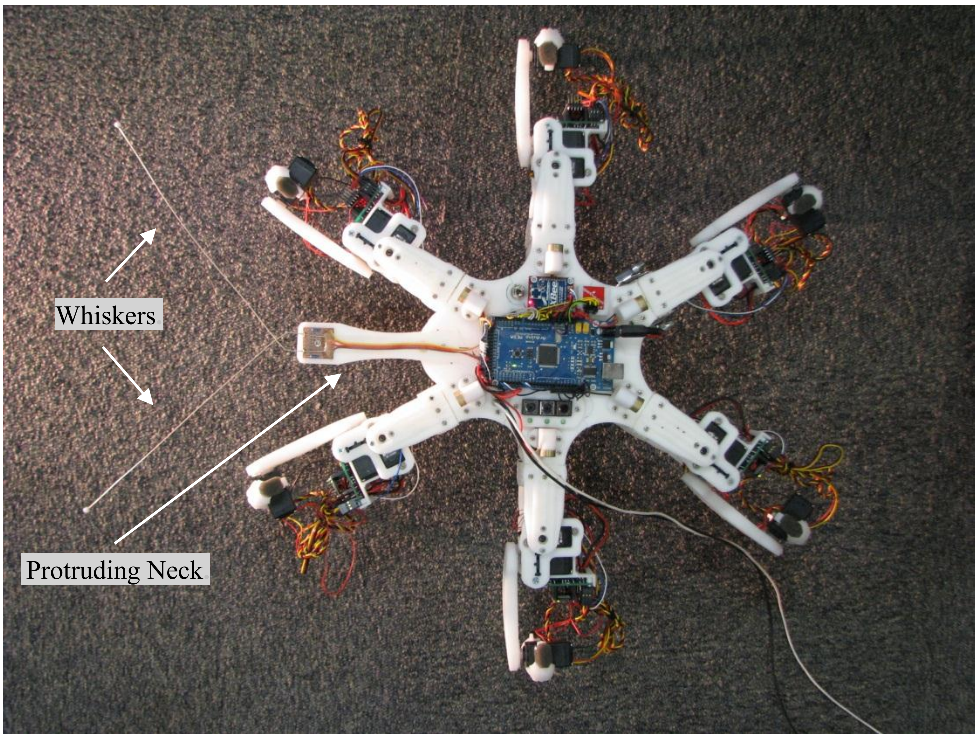


Figure 3.15. Robot Top View.

output is a two bits binary signal indicating the status (‘on’ or ‘off’) of each whisker. In order to reduce the probability of collision with the legs, the whiskers have been extended from the robot’s body. This solution was inspired by observing certain species of beetles such as the Trachelophorus Giraffa and the Colliuris Pensylvanica. These insects have a long neck that (among other possible functions) avoids collisions between their legs and antennae.

3.4. Empirical Determination of Gait Parameters

In order to determine gait parameters such as rotation speed and turning speed, the time required by the robot for performing a preset rotation has been measure while the robot walked on a wooden surface. After 10 experiments where the robot exclusively performed one type of rotation, values shown in Table 3.1 were obtained.

Table 3.1. Rotation on the Spot Speed and Turning Movement Speed for Experimental Robot.

Type of Movement	Speed
Rotation on the Spot	7.8º / sec
Turning	2.8º / sec

In all of the experiments, results were consistent with the ones shown in Table 3.1. Videos showing the experimental robot performing a rotation on the spot and turning movement, which involves simultaneous translation and rotation, can be found by following links D01 and D02 in Appendix D, respectively.

Additional experiments were conducted in order to measure the robot’s walking speed on different kinds of terrain. In all of these experiments, the robot walks with its 6 legs and the slope of the terrain is negligible. Results are shown in Table 3.2.

Table 3.2. Robot Walking Speed for Different Types of Terrain.

Terrain	Average Speed [mm/sec]
Black Rubber Tiles	70.3
Carpet Tiles	70.3
Ceramic Tiles	69.9
Concrete	70.0
Grass	63.1
Gravel	69.1
Wooden Boards	70.0

Average speeds present little variation when the robot walked on black rubber tiles, carpet tiles, ceramic tiles, concrete or wooden boards. These surfaces in conjunction with the robot's rubber leg tip provided good traction for walking. When the robot walked on gravel, there was a little slip due to the terrain's instability. As a result, walking speed was slightly reduced. In order to allow the robot to walk on grass, the lifting height of the swing legs was increased. As a consequence, the time required by each gait cycle was also incremented, which in turn reduced the average walking speed of the experimental robot.

Another set of experiments was conducted in order to determine how the robot's walking speed is affected by the loss of one or more legs. This set of experiments was conducted while the experimental robot walked or dragged itself on carpet tiles. Results are shown in Table 3.3.

Table 3.3. Average Walking Speed as a Function of the Robot's Number of Legs.

Number of Legs	Average Speed [mm/sec]
1	18.8
2	20.3
3	20.9
4	33.4
5	30.9
6	70.3

As expected, the fastest gait of the series turned out to be the tripod gait. This coincides with what is commonly accepted by researchers interested in walking robots or insects. When 5 legs were available, the robot generated an asymmetrical gait which was slower than the ones generated for 6 or even 4 legs. In the latter case, a wave gait was generated. When 3 legs were available, the robot generated an alternated tripod gait. Here, all of the robot's legs were in the same stance/swing phase during half of the gait cycle. Because the robot's core body had to be lowered to the ground during the swing phase and lifted during the stance phase, more time (and energy) was required during each gait cycle. As a result, walking speed was much less than half (and third) the walking speed of the normal tripod gait. In the final two cases, the robot dragged its body during locomotion. In comparison with the alternated tripod, a significant speed reduction was observed here.

In general, the time during each gait cycle is attributed to the following 4 factors:

- Servo Movement Time: is the time the servo motors actually move and is determined by the distance between starting and target servo motor angles.

- Servo Delay: A delay of 8ms per each 1° of servo rotation has been introduced in order to generate a smoother and more natural leg movement.

- Processing Time: is the time the system requires for collecting all of the robot's sensorial information, evaluating the robot's performance, calculating the light source orientation and determining the leg coordinates for the next gait cycle.

- Communication Time: is the time required for data communication among the robot's microcontrollers and between the robot and the external computer.

Figure 3.16 shows an example of time expenditure during a tripod gait cycle of the experimental robot. Here it can be observed that the processing time of the self-diagnosis methods to be presented in the next chapter are normally less than a quarter of the gait cycle period. There is a direct proportion between the number of sensors

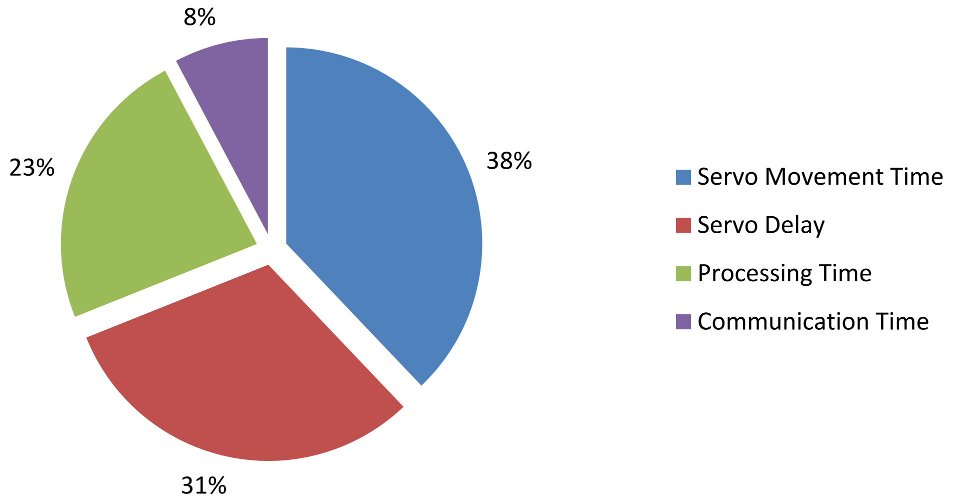


Figure 3.16. Example of Time expenditure During Gait Cycle.

in the robot and the processing time required by the system. In fact, most of the processing time is spent in collecting sensor data.

The walking speed of the experimental robot can be increased by altering one or more of the 4 factors previously discussed. For instance, the robot walking speed could be increased by:

- Providing the robot with faster servo motors in order to reduce servo movement time.

- Reducing the servo delay.

- Reducing the robot’s number of sensors in order to decrease the processing time.

A reduction of the number of legs in the robot would reduce communication time (as there would be fewer microcontrollers communicating with each other). However, the speed increment due to the communication time reduction would be cancelled by the larger speed decrement associated with the leg loss.

3.5. Leg Release Mechanism

The leg release mechanism allows the robot to automatically detach any of its legs as an extreme compensatory measure. For instance, it may be necessary to shed a malfunctioning leg when it interferes with the robot’s gait preventing it from walking. This mechanism emulates the autotomy capability found in a number of living creatures, where it is used as a self-defence mechanism designed to elude a predator's grasp.

The authors in [52] have presented the R-LEGAM leg release mechanism. In this system, each autotomy capable leg has an extra servo motor exclusively utilised for leg detachment. The leg release mechanism proposed in this thesis is more compact and lighter than R-LEGAM because this extra servo is not required.

Each leg of the experimental robot utilised in this research is joined to the robot’s core by means of the attachment module depicted in Figs. 3.17 and 3.18. The core section of each attachment module is bolted to the robot’s core whilst the leg section is bolted to the robot’s leg. Both sections are held together by means of a hook and slotted pin. In order to prevent any undesirable rotation of the leg around the pin, the

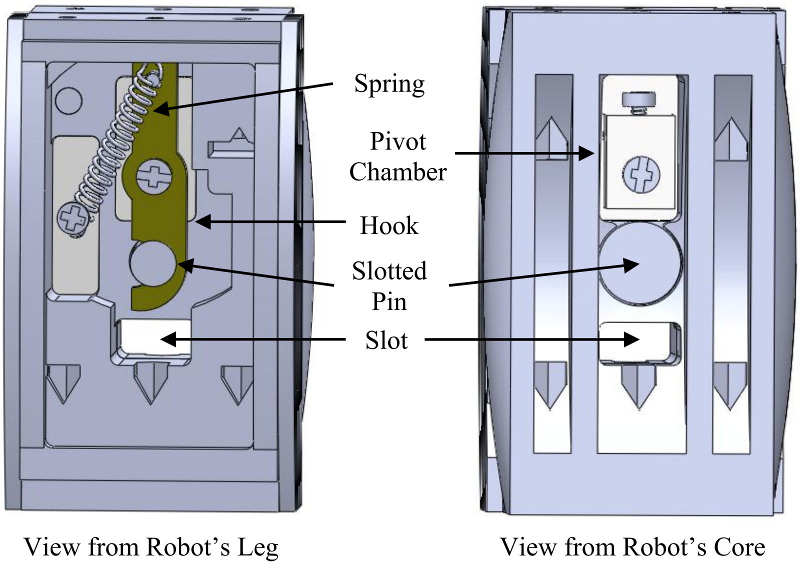


Figure 3.17. Leg Attachment Module.

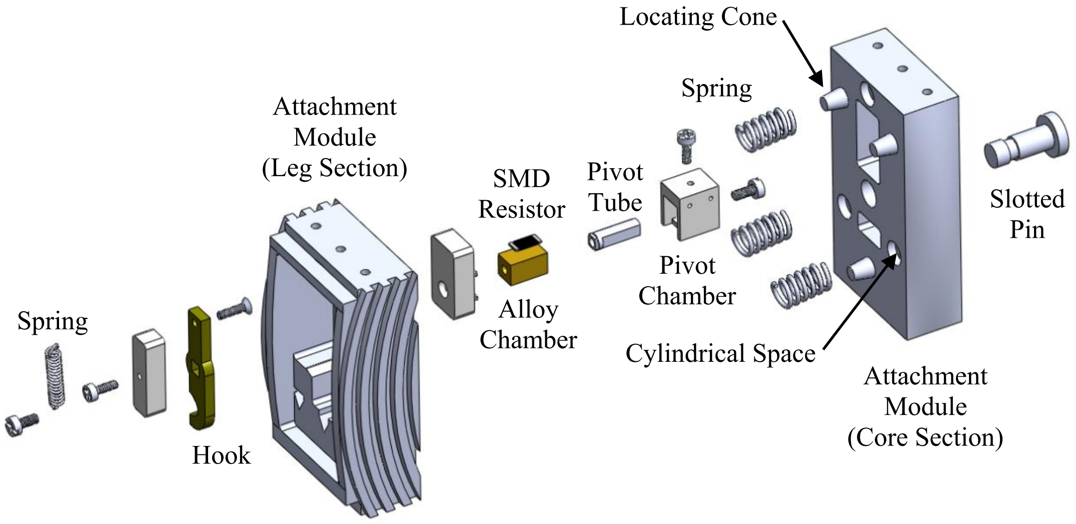


Figure 3.18. Exploded View of the Leg Release Mechanism.

attachment modules were designed with three fixing cones (in the core section) and their respective conical holes (in the leg section). In addition, both sections incorporate cylindrical cavities for three compression springs that storage the energy necessary for leg ejection. The electrical connections between the robot's leg and core are established by means matching pins and headers located in the designated slots shown in Fig. 3.17.

The leg release system utilises a low melting point alloy that fills the empty space left by the pivot tube in the alloy chamber. When the leg is attached to the robot's core, the alloy is cold and solid. This prevents the pivot tube and the hook from rotating. When the robot determines that it is necessary to shed a leg, it applies power to an SMD resistor physically connected (but electrically isolated) to the alloy chamber. Then, the resistor generates heat that is transferred to the low melting point alloy. As this is heated, it loses strength until it is unable to resist the energy stored in the tension spring attached to the hook. As a result, the pivot tube and hook rotate liberating the slotted pin. When this occurs, the three compressed springs between the two sections of the attachment modules eject the leg, which is then electrically and mechanically disconnected from the robot's core.

Table 3.4 shows the results of 10 experiments that were conducted for evaluating the performance of the leg release mechanism. Before each experiment, the alloy inside the leg release mechanism was at room temperature (~20 ºC). Then, the leg release mechanism was activated and the time required for melting the alloy and ejecting the leg was measured. In addition, the electric charge consumed during each experiment was calculated.

Table 3.4. Execution Time and Electric Charge Consumption of Leg Release Mechanism.

Experiment	Execution Time [secs]	Electric Charge Consumption [mAh]
1	164	10.1
2	174	10.7
3	169	10.4
4	208	12.8
5	199	12.3
6	190	11.7
7	200	12.3
8	207	12.8
9	208	12.8
10	195	12.0

Results in Table 3.4 show an average execution time of 191 seconds and an associated average electric charge consumption of 11.8 mAh for the 10 conducted experiments. A video showing the leg release mechanism in action can be found by following link D03 in Appendix D.

3.6. Summary

This chapter has introduced the experimental hexapod robot on which the algorithms and techniques presented in the following chapters have been implemented and tested. A number of topics related to the robot design and construction have been discussed in section 3.1. Then, section 3.2 has presented an outline of the distributed robot control. There are two reasons behind the distributed nature of the robot control. Firstly, the memory constraints of the robot’s core microcontroller necessitated using an external computer where most of the developed algorithms are executed. Secondly, in order to reduce the number of electrical connections between the robot’s core and legs, the leg’s microcontrollers were introduced. Therefore, the information provided by the leg’s sensors is processed by these microcontrollers and then transmitted to the core control. Because only power and communication connections are required between any leg and the robot’s core, the leg release mechanism is simplified.

The robot’s sensors and some of their operation principles have been presented in section 3.3. In particular, a leg tip force sensor developed during this research is introduced. The set of sensors available in the experimental robot has proved to be sufficient for testing the methods and techniques proposed in this thesis. However, these methods are not constrained to these specific types and number of sensors. It is expected that these methods can be applied to most robot and sensor topologies.

In Section 3.4, a number of the robot’s gait parameters related to walking speed have been empirically determined. Here, it was discussed how walking speed of the experimental robot was affected by factors such as type of terrain, leg losses, number of sensors, etc.

Finally, section 3.5 has introduced a leg release mechanism that emulates the autotomy capability of some living creatures. Experiments showed the time and energy required by this mechanism for ejecting a leg. Due to the detrimental effects that shedding a leg has on the robot’s walking speed and energy consumption, this system will be only used as a last resort.

A number of compensatory measures will be considered in chapter 5. However, this will be preceded by the next chapter, which addresses the identification of abnormal situations in robots and their environment.

Chapter 4
Autonomous Identification of Detrimental Disturbance Sources

This chapter introduces techniques for the detection, identification and classification of situations that degrade robots' performance or prevent them from accomplishing their missions. The identification and classification of abnormal conditions developed in this research is fundamental for the autonomous detection of a robot's failures and adaptability to unforeseen features of the robot's environment. By using these techniques, it is expected to increase resilience to damage, extend lifespan and improve autonomy in robotic missions where human intervention is difficult or impossible, such as in extra-terrestrial exploration or other remote hostile environments.

Chapter 4 is structured as follows. The first section addresses the detection of detrimental disturbances. An abnormal situation classification algorithm and techniques for the identification of detrimental disturbance sources are presented in the next two sections. Then, the methods previously introduced in this chapter are applied and the experimental results are discussed. Lastly, a summary of the chapter is provided.

4.1. Detrimental Disturbance Detection

Detrimental disturbances are generated by abnormal situations that negatively impact the robot's performance. From a robot's perspective a detrimental disturbance could be perceived as one or more sensor readings which are out of the expected range. Alternatively, detrimental disturbances can be evidenced by normal sensor values whose combined information is unexpected or generates contradictions. Both cases are generated under two general scenarios. In the first one, there could be a malfunction in the robot’s hardware. For instance, a robot could develop defective sensors, communication problems among its different control modules, faulty controllers or damaged parts of its body. On the other hand, unexpected sensor readings could be generated by certain features of a robot’s environment that have not been previously experienced by the robot. Consequently, detrimental disturbances produced by a robot’s malfunction and those generated by the robot’s environment will be referred to as internal and external disturbances, respectively. Both kinds of disturbances can be detected by the negative impact they have on a robot’s performance or by extreme values in the robot’s sensor readings.

4.1.1. Performance Assessment

A way of detecting detrimental disturbances is by evaluating the progress the robot is making towards the completion of its mission. Then, a detrimental disturbance arises when the robot is not making progress after a certain amount of time. In this case, a quantitative measure of the robot’s progress is necessary. Depending on the nature of the robot’s mission, this calculation could involve different levels of complexity. Because this is closely related to the kind of task each particular robot is undertaking, a general solution to this problem will not be addressed. However, the method employed in this work for quantitatively measuring the progress of the experimental robot will be presented as an example.

In this research, the experimental robot has been assigned a simple task, from which quantitative measurements of the robot’s progress are easy to acquire. This task consists in finding a light source, whose direction is determined by the array of light sensors introduced in the previous chapter. In the experimental environment it is assumed that there is a single light source. Under normal conditions, despite the non-linear relation between light intensity detected by the photo-resistors and the corresponding light sensor readings, an approximate calculation of the light source direction (γ_r) can be determined by means of Eq. 4.1.

$$\gamma_r = \operatorname{arctan2}\left(\frac{\sum_{k=0}^{N-1} L_{k+1}\sin(k\pi/3)}{\sum_{k=0}^{N-1} L_{k+1}\cos(k\pi/3)}\right) \tag{4.1}$$

Where

- L_k is the reading of the *k-th* light sensor. These sensors are enumerated in counterclockwise order starting at the one located on the positive x-axis as depicted in Fig. 4.1.
- N is the number of functional light sensors (6 under normal conditions).

Because γ_r is measured from the positive x-axis, a heading of $\pi/2$ rad indicates that the robot is walking towards the light source. Therefore, when different values of γ_r are obtained, the robot is rotated $\gamma_r - \pi/2$ rad. This assures that the maximum light intensity is detected by one of the two front light sensors $(L_2 \text{ or } L_3)$. Then, a

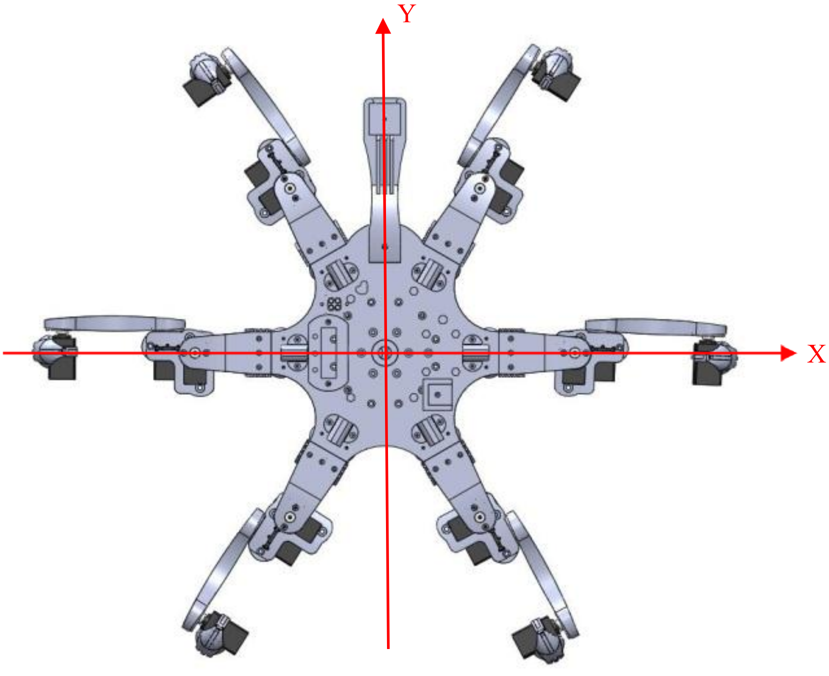


Figure 4.1. Experimental Robot X-Y Coordinate System.

more exact light direction angle (γ) is calculated as follows. The diagram in Fig. 4.2 illustrates a light source located between light sensors 2 and 3. After applying the law of cosines to the red triangle in Fig. 4.2, Eq. 4.2 is obtained.

$$d_{L3}^2 = d_{L2}^2 + d_{23}^2 - 2d_{L2}d_{23}\cos(\alpha) \tag{4.2}$$

Where

- d_{L2} is the distance between light sensor 2 and the light source.
- d_{L3} is the distance between light sensor 3 and the light source.
- d_{23} is the distance between light sensors 2 and 3 (75 mm in the experimental robot).
- α is the angle between the d_{23} and the d_{L2} lines.

The magnitude of d_{L2} and d_{L3} can be calculated from the readings of light sensors 2 and 3, respectively. In this case, the light source intensity and the relation between light sensor readings and distance from the light source must be known. If this information is not available, the readings from light sensors 2 and 3 can be used as alternative values of d_{L2} and d_{L3}, respectively. However, in this case the distance between the robot and the light source cannot be calculated with this method.

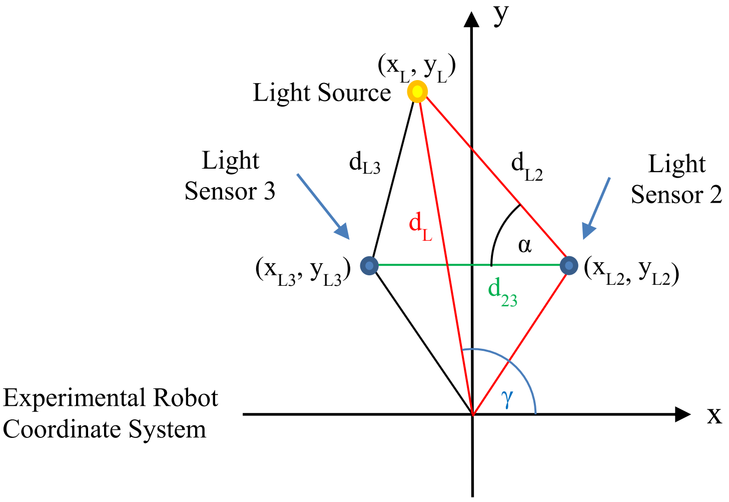


Figure 4.2. Parameters Used in the Calculation of the Light Source Location.

The α angle obtained from Eq. 4.2 is calculated using expression 4.3.

$$\alpha = \arccos\left(\frac{d_{L2}^2 + d_{23}^2 - d_{L3}^2}{2d_{L2}d_{23}}\right) \tag{4.3}$$

Then, the α angle is used for calculating the (x_L, y_L) coordinates of the light source by means of Eq. 4.4.

$$(x_L, y_L) = (x_{L2}\cos(\pi - \alpha), y_{L2}\sin(\pi - \alpha)) \tag{4.4}$$

Where (x_{L2}, y_{L2}) are the (x, y) coordinates of light sensor 2 ((38,65) mm in the experimental robot).

Finally, the distance and the angle between the light source and the origin of the experimental robot coordinate system can be determined by means of Eqs. 4.5 and 4.6, respectively.

$$d_L = \sqrt{x_L^2 + y_L^2} \tag{4.5}$$

$$\gamma = \text{arctan2}(x_L, y_L) \tag{4.6}$$

It must be noted that erroneous results will be obtained from Eq. 4.5 if the correct d_{L2} and d_{L3} values are not provided. On the other hand, Eq. 4.6 utilises the relation between the angle of the light source in relation to the experimental robot and the difference between readings of light sensors 2 and 3. Therefore, if these readings are used as d_{L2} and d_{L3} values, it is still feasible to obtain correct results from Eq. 4.6.

A quantitative measure of the experimental robot's progress is given by the light sensors' maximum value (ℓ), which is determined by Eq. 4.7.

$$\ell = \max(L_1, L_2, ..., L_N) \tag{4.7}$$

It is taken that when $\ell = 1000$, the robot has found the light source and its mission has been accomplished.

After each step, the walking robot transmits its current sensorial information to the external computer. Among this information, the light sensors readings are utilised for the calculation of the ℓ value associated with that step. If ℓ_s represents the ℓ value associated with the *s-th* step, the progress or lack of progress between two consecutive steps in the robot's mission can be represented by P_2 in Eq. 4.8.

$$\mathrm{P}_2 = \ell_s - \ell_{s-1} \tag{4.8}$$

Where

- $\mathrm{P}_2 > 0 \Rightarrow$ Progress
- $\mathrm{P}_2 \leq 0 \Rightarrow$ Lack of Progress

The lack of progress between two consecutive steps by itself does not show an abnormal situation. It is common for the robot to take a few steps without getting closer to the light source (e.g. when adjusting its walking direction) before it can make further progress. However, in this investigation the following two cases, where a lack of progress is manifested, have been related to the identification of abnormal situations.

- Sustained lack of progress: occurs when the robot is unable to make any progress after a number of steps. In other words, $\mathrm{P}_2 \leq 0$ for the considered number of steps.

- Cyclic lack of progress: in this case the robot enters into a loop where after some progress is made, the robot suffers a setback that reduces its progress until it is equal or less than that at the beginning of the loop. Here P_2 may be positive or negative during the cycle, but ℓ at the beginning of the loop is greater or equal to ℓ at the end of the loop.

Both, sustained and cyclic lack of progress are identified when expression 4.9 is satisfied.

$$\ell_s \le \ell_{s-1} \wedge \ell_s \le \ell_{s-2} \wedge \cdots \wedge \ell_s \le \ell_{s-n} \tag{4.9}$$

Where n is the number of previous steps taken into account during the robot's performance evaluation.

For instance, if we consider $n=10$ and the following simulated data set containing ℓ values associated with the first 15 steps taken by the robot.

$$\ell \text{ set} = \{300, 312, \underline{320}, 299, 285, 276, 290, 293, 301, 308, 297, 280, \underline{286}, 290, 300\}$$

This ℓ set has been graphed as shown in Fig. 4.3. In this case, the robot has alternating periods of progress and lack of progress. As $n=10$ and $\ell_{13} - \ell_3 = 286 - 320 = -34 < 0$, an abnormal situation alarm will be triggered at step 13. In other words, since the robot is unable to exceed the $\ell_3 = 320$ value after

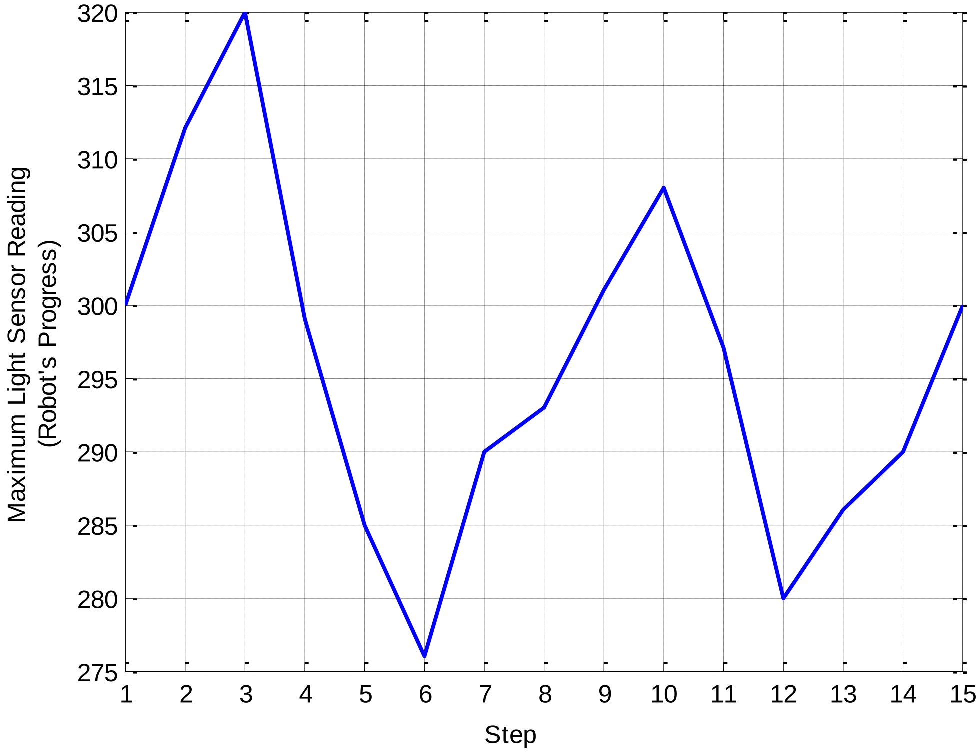


Figure 4.3. Cyclic Lack of Progress (Simulated Data).

$n = 10$ steps, a cyclic lack of progress is detected.

The selection of the value n determines the sensitivity of the lack of progress detection. A very small value of n (e.g. $n = 2$) will trigger false alarms, while a large value of n will make the detection slow. As in the previous example, $n = 10$ has empirically demonstrated to be a good compromise between speed and sensitivity. Therefore, this value is utilised during this research. This means that the robot must make progress within every set of 10 consecutive steps; otherwise, an abnormal situation alarm is triggered.

The performance assessment method has been tested using the experimental robot. In this work it is considered that the robot has accomplished its mission when the maximum light sensor reading is greater than or equal to 1000 $(\ell \geq 1000)$. Initially, a single light source was located on the floor and the robot started walking towards it as shown in Fig. 4.4. Then, after the robot had taken 11 steps under normal conditions, light sensor 2 was covered as can be seen in Fig. 4.5.

The robot's performance, corresponding to the maximum value of all the robot's light sensors, was monitoring during the experiment. This data is plotted on the graph in Fig. 4.6, where the abscissa represents the number of steps taken by the robot and the ordinate is the output of the ADC (Analogous/Digital Converter) digitalising the maximum light sensor reading. In this research, 10 bit ADCs, with an output in the range 0~1023, were utilised.

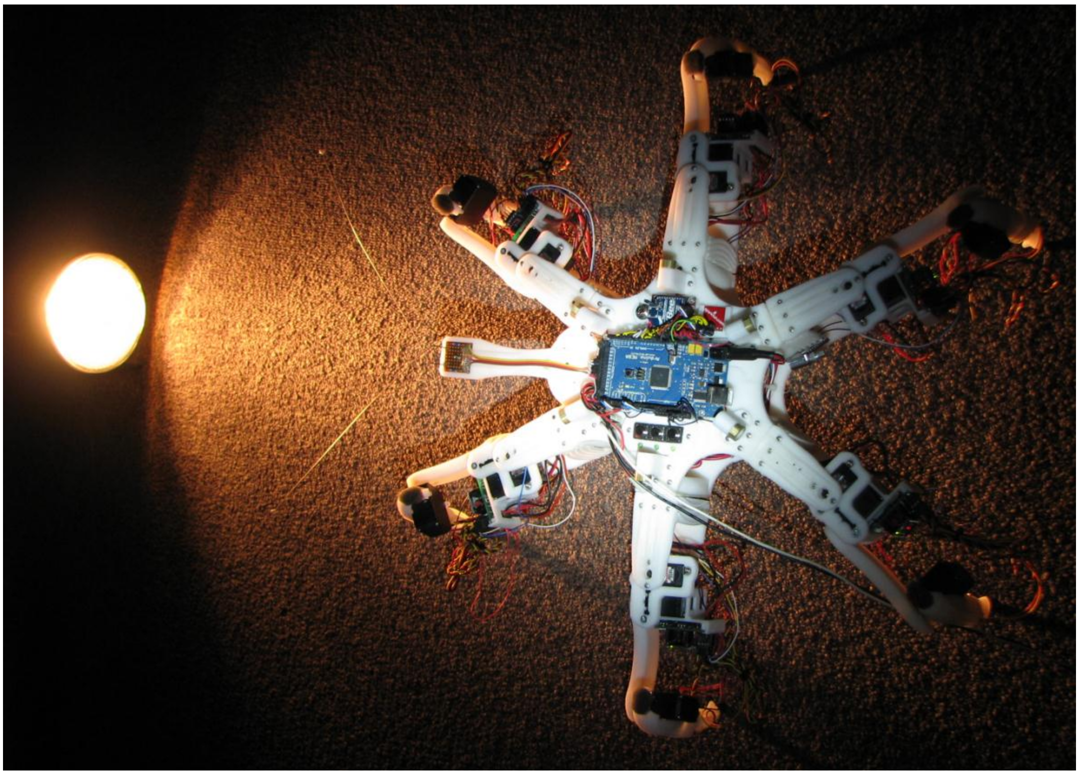


Figure 4.4. Experimental Robot Walking Towards a Light Source.

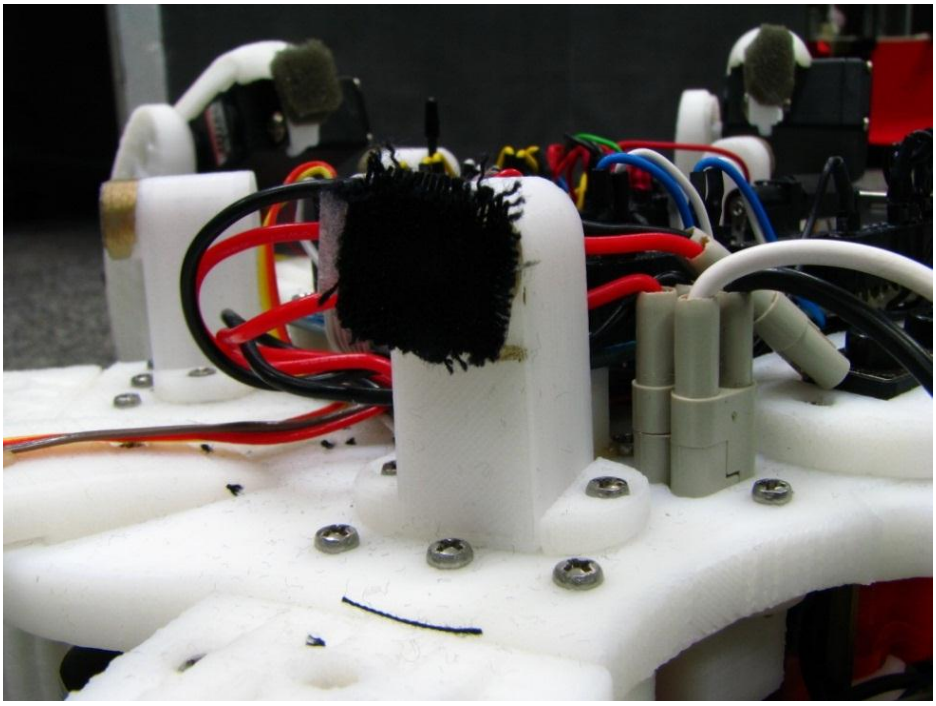


Figure 4.5. Covered Light Sensor.

Figure 4.6 shows that under normal conditions (before step 11) the experimental robot achieved good performance characterised by continuous progress. Once light sensor 2 was covered, the robot continued progressing in its mission until step 13. Then, between steps 13 and 21, although the robot was alternately closer and farther from the light source, it was still able to make progress overall. At step 21, the robot hit a peak of 961 in its light sensor readings. However, after 10 steps, (even when it was close at steps 26 and 27 with a sensed value of 960) the robot was unable to exceed this reading. Therefore, an abnormal situation alarm was triggered at step 30.

Figure 4.7 illustrates the robot's behaviour during the experiment, before and after light sensor 2 was covered. Here, the yellow circle represents the light source, the red sector(s) represent(s) the light sensor(s) where the maximum amount of light is received and the grey sector represents the blind zone associated with the covered sensor. Then, robot operation during the experiment can be described by considering the robot initial state, represented by Fig. 4.7(a), and the three robot orientations

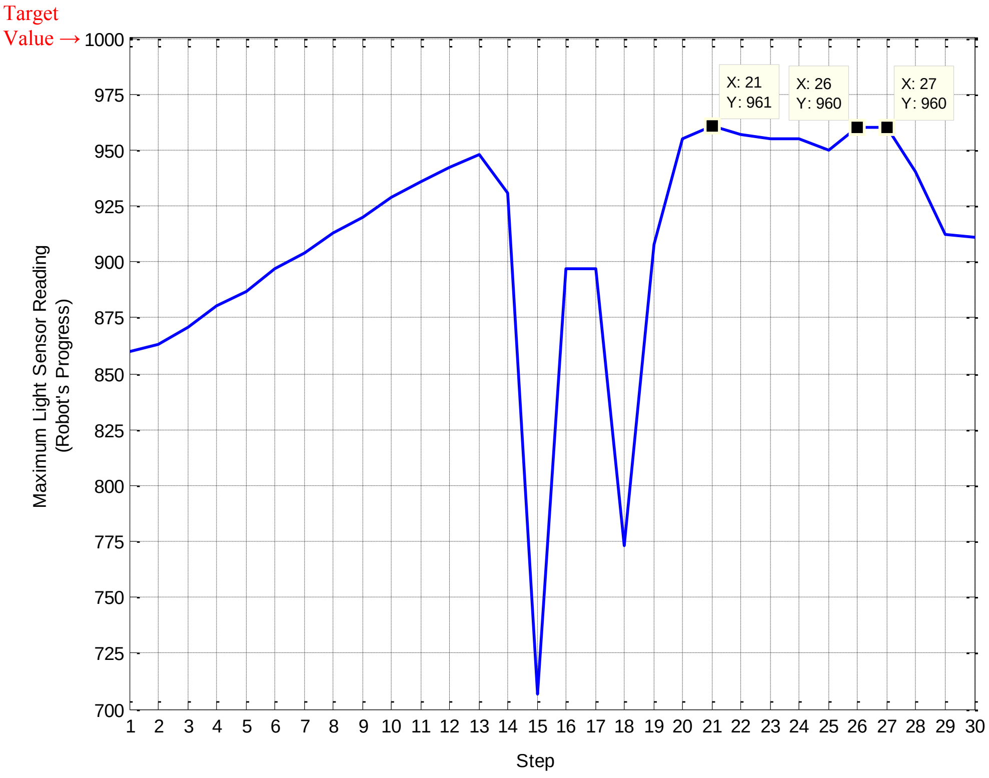


Figure 4.6. Robot's Performance during an Experiment with Covered Light Sensor.

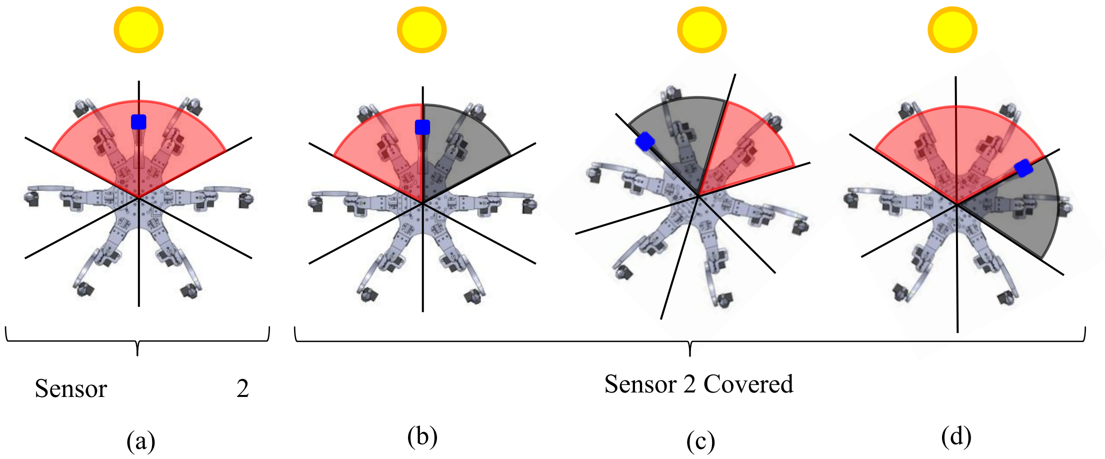


Figure 4.7. Abnormal Robot Behaviour after One of Its Light Sensors Is Covered.
(Note that the Blue Square Indicates the Front of the Robot).

depicted in Fig. 4.7(b-d), as follows.

- Initially, all the light sensors operate normally and the robot makes sustained progress in its task. As can be seen in Fig. 4.7(a), under these conditions the maximum amount of light is received by either light sensor 2 or 3.
- Once light sensor 2 is covered, Fig. 4.7(b) shows that the maximum amount of light is received by sensor 3. Consequently, the robot performs a counterclockwise rotation on the spot towards the expected light source direction.
- Once the robot reaches orientation (c), the maximum amount of light is then received from the sensor 1 direction. This time the robot performs a clockwise rotation on the spot towards the expected light source direction. In this case, the robot rotates until it reaches orientation (d). Here, the maximum amount of light is received by either light sensor 3 or 4.
- Following this, the robot performs a new counterclockwise rotation towards the expected light source direction. In this ocassion, the robot rotates until it, once again, reaches orientation (b).

Therefore, the robot enters into an infinite loop of rotations on the spot resulting from the covered light sensor and associated miscalculation of the light source direction.

This infinite loop is only interrupted when the system detects the resulting cyclic lack of progress in the robot's mission.

The periods of increase in the robot's performance shown in Fig. 4.6 coincide with the periods where uncovered light sensors face the light source. Those periods where the covered light sensor was facing the light source coincide with a decrease in the robot's performance.

The graph in Fig. 4.8 displays the output of all the robot's light sensors during the experiment. As it can be seen here, the maximum sensor reading values are provided by light sensor 2 between steps 1 and 11, light sensor 1 between steps 15 and 18, light sensor 4 between steps 23 and 25; and light sensor 3 otherwise. The output of these sensors, during the corresponding interval where they are the greatest, represents the robot's performance illustrated by the graph in Fig. 4.6.

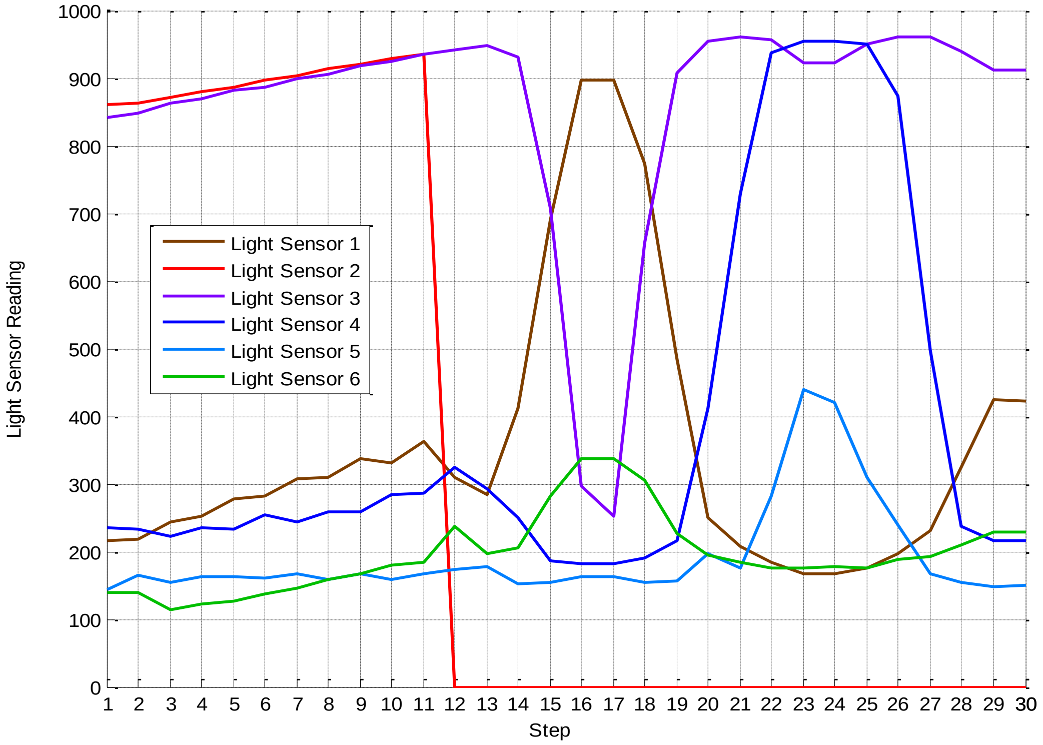


Figure 4.8. Light Sensor Outputs during an Experiment with a Covered Light Sensor (Sensor 2 Covered after Step 11).

The response of the robot's light sensors during the experiment can be seen in Fig. 4.8. The robot operation corresponding to this experiment is described as follows.

- Between steps 1 and 11 the robot walks with all light sensors functioning correctly and light sensor 2 receives the maximum amount of light. This means that the light source is in front of the robot but slightly to the right.
- At step 11, light sensor 2 is covered. As a result, its readings drop to zero, but the robot's performance is not immediately compromised. This is due to the fact that the robot is positioning its legs to perform a rotation on the spot and light sensor 3 is still facing the light source.
- The robot operation after step 11 is more clearly explained by considering the three orientations of the experimental robot shown in Fig. 4.7(b-d). As soon the robot begins to rotate to orientation (c), there is an increment in the readings of light sensor 1 and a decrease in the readings of light sensor 3.
- At step 16, the robot reaches orientation (c) and makes a new estimation of the light source direction. This time a counterclockwise rotation is performed towards orientation (d). As the robot rotates, light sensor 3 and then light sensor 4 start to receive more light, whereas the readings of light sensor 1 drop.
- At step 23, the robot reaches orientation (d) and makes a new estimation of the light source direction. This time a clockwise rotation is performed towards orientation (b). As the robot rotates, light sensor 4 and then light sensor 3 start to receive less light, whereas the readings of light sensor 1 are increased.
- This loop is interrupted when a cyclic lack of progress is detected after step 30, before the robot is able to reach orientation (b).

4.1.2. Pain

In some circumstances, regardless of the progress that the robot is making in its task, one or more of the robot's sensor readings may be very different from what is expected. This kind of detrimental disturbance will be referred to as pain. When this is not compensated, it could have a devastating impact on the robot’s performance in the short or long term. For instance, a robot’s motor could become overheated even when the robot is performing its task correctly. However, if this situation is not addressed, that motor will become permanently damaged, which in time will affect the robot’s performance.

In comparison with the lack of progress previously introduced, this kind of disturbance is simpler to detect. In order to provide the experimental robot with the faculty of experiencing pain, those sensors that can evidence a potentially damaging situation for the robot must be identified. Then, a threshold must be established for each one of these sensors. Therefore, sensor readings above this threshold value will be identified by the robot as pain. Because pain is “built into” living creatures rather than learned, the threshold values are not calculated by any learning method. They are included as part of the initial information the robot possess.

In this research, the experimental robot’s sensors able to identify a damaging situation are the servo motor and leg tip force sensors. A high value of these sensors during a prolonged period of time can burn out a servo motor or break a robot’s leg, respectively. In order to determine the pain threshold and the period of time the robot can experience pain without being damaged, an empirical approach was adopted. The servos where manually forced until they became hot and the legs were stressed by applying a reasonable amount of force. As a response to a painful experience the robot has a reflex reaction. This is comprised of shutting down the servo motors of the leg(s) in pain, moving the rest of the legs to home position (in which the legs are retracted and stop supporting the robot’s weight) and triggering an abnormal situation alarm.

In order to illustrate the effect of the pain mechanism, a force has been applied to one of the leg servo motors and the output of its force sensor has been displayed in the

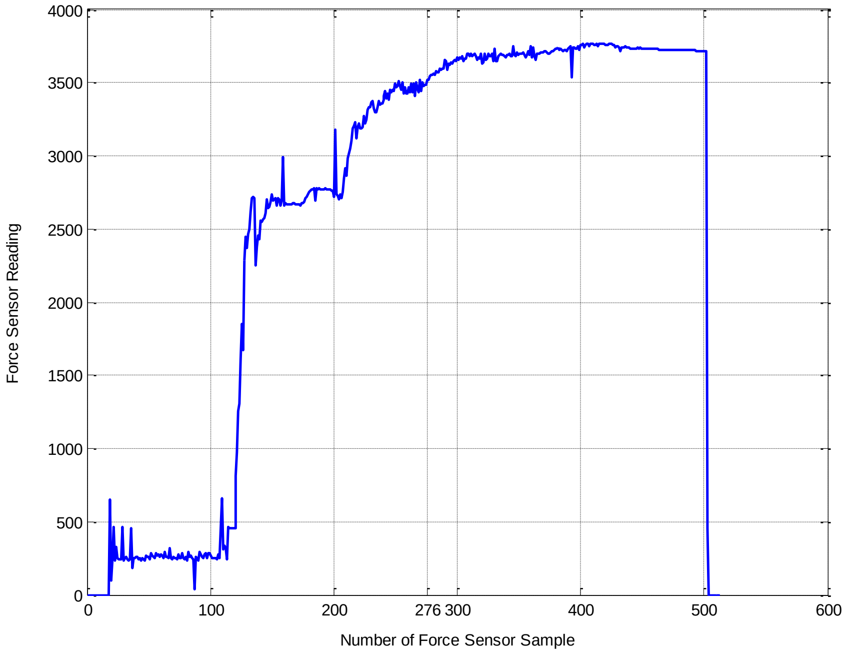


Figure 4.9. Force Sensor Output of a Servo Motor Experiencing Pain.

graph of Fig. 4.9. As was mentioned in chapter 3, the force sensor's output is the width of the PWM signal applied to the servo motor. This is measured by an algorithm that continuously increments a counter while the PWM signal is in a low voltage state. The larger the magnitude of the force applied by the servo motor to stay in its commanded position, the larger the value of the counter. The graph in Fig. 4.9 shows the value of this variable for 550 samples of the servo's force sensor. As can be seen in the graph, a force was detected by the sensor around sample 20. Then, the applied force had an upward trend until the force sensor gave an output of 3500 at sample 276. This is the pain threshold established for the force sensor. From this point, the time while the force sensor output is greater than the pain threshold is measured by the system. Once 10 seconds of continuous pain have passed, which corresponds to the period between sample 276 and 500, the system disconnects the control signal of the motor. As a result, overheating the servo and the associated impact on the robot's performance are avoided.

4.2. Classification Algorithm

In the simplest scenario a detrimental disturbance source can be related to a sensor providing readings that are very different from its expected value (pain). However, as it is common to find tens of sensors in a robot, different combinations of sensors may present unexpected readings. This obscures the relationship between the sensorial information evidencing the detrimental disturbance, and the detrimental disturbance source. The same situation occurs with detrimental disturbances identified by means of a performance assessment. In this case, the sensorial information may present only slight differences from expected values. Here, more important than the magnitude of this variation is the particular combination of sensors presenting the difference. A first step to addressing these problems, and to be able to link an identified detrimental disturbance with its source, is anomaly classification. Therefore, once a detrimental disturbance has been properly classified and labelled it can be quickly identified in future recurrences. As a result, an anomaly can be directly linked with its associated compensatory measures if these have been previously determined. The classification algorithm developed in this research is presented as follows.

4.2.1. Sensor Reading Set

In order to report status and mission performance, the experimental walking robot transmits all of its sensor readings to the external computer after every step. These data grouped in a vector constitute the SRS (Sensor Reading Set) shown in Table 4.1.

Table 4.1. Experimental Robot's Sensor Reading Set.

Ls_k	Ps_{ij}	$Fs_{i\,j}$	TFs_i	Acc_x	Acc_y	Bl_k	Ws

Where

- Ls_k represents the reading of the *k-th* light sensor, with $k = 1, 2, ..., 6$.
- Ps_{ij} represents the position sensor reading of the servo motor *j* located in the experimental robot’s leg *i*.

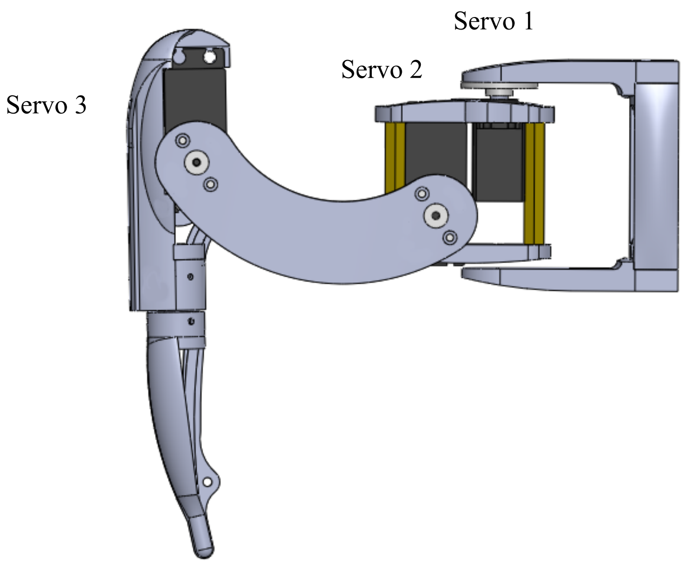


Figure 4.10. Distribution of Leg Servo Motors.

- Fs_{ij} represents the force sensor reading of servo motor j located in the experimental robot's leg i.
- $i = 1,2,...,6$ represents each robot's leg. These have been enumerated in counterclockwise order starting from the right middle leg.
- $j = 1,2,3$ indicates each leg's servo motors. As illustrated in Fig. 4.1, Servos 1, 2 and 3 provide the robot's leg with coxa, femur and tibia articulations, respectively.
- TFs_i represents the tip leg force sensor reading of leg i.
- Acc_x and Acc_y represent the x-axis and y-axis accelerometer readings, respectively.
- Bl_k represents the reading of the *k-th* battery level indicator, with $k = 1,2,3$ as illustrated in Table 4.2.
- Ws represents the whiskers' status as shown in Table 4.3.

Table 4.2. Battery Level Indicator Symbology.

Symbol	**Voltage**	**Battery Power Application**
Bl_1	6 volts	Leg PCB and leg microcontroller
Bl_2	6 volts	Servo motors and leg release mechanism
Bl_3	9 volts	Core sensors and core microcontroller

Table 4.3. *Ws* Code for Whiskers' Status.

W_s	**Whiskers' status**
0	Both whiskers deactivated
1	Only right whisker activated
2	Only left whisker activated
3	Both whiskers activated

4.2.2. Expected Sensor Readings

The information contained in a SRS is received and processed by the external computer to be translated into an AIS (Abnormality Index Set), which is the input for the proposed classification algorithm. The AIS is a vector of normalised differences between expected and actual sensor values. This is calculated by means of expression 4.10.

$$AIS_K = Nr\frac{|SRS_k - E_k|}{R_k} \tag{4.10}$$

Where

- SRS_k is the reading of the *k-th* sensor.
- E_k is the expected value of the *k-th* sensor reading.
- R_k is the range of the *k-th* sensor.
- *Nr* is the normalisation value of AIS (in other words, $0 \le AIS_k \le Nr$).
- $k = 1, 2, ..., N$ (number of the robot's sensors).

It must be noted that the $|SRS_k - E_k|$ operation, in expression 4.10, describes a general process whereby a sensor reading is compared with its corresponding expected normal value. The expected value of a sensor reading depends on a number of factors such as type of robot, application and the robot's environment. Therefore, a general solution for the calculation of this value is not provided here. However, the determination of the expected sensor readings, under normal conditions, for the experimental robot is presented as an example as follows.

- Calculation of expected robot light sensor readings.

The lighting conditions the robot may find during its mission are unknown. Under normal operation they may span from complete darkness to sufficient light to saturate the light sensors, which will cover the whole of the light sensors' range. In addition, considerable differences between readings of light sensors facing the light source and those facing shadows are anticipated. However, under normal operation, similar patterns are expected for all of the robot's light sensor readings. On the other hand, faulty light sensors most commonly present two abnormal conditions. They may produce noise (a low amplitude random signal) or a constant signal (corresponding to a short-circuited or open-circuited light sensor). Therefore, an abnormal light sensor reading has a constant value, while the rest of the light sensors provide readings that change with time, or an abnormal light sensor has a variable value, when the rest of the light sensors present constant readings. Because, this is a complex case where there is an expected pattern rather than an expected sensor reading, the AIS will be determined by means of Eq. 4.11.

$$AIS_k = \begin{cases} \dfrac{Nr}{n}\displaystyle\sum_{i=1}^{n} c_i & , \quad if\ Ls_k^0 = Ls_k^{-1} \\ \dfrac{Nr}{R_k}\left\| Ls_k^0 - Ls_k^{-1} \right| - median(D) \right| & , \quad otherwise \end{cases} \tag{4.11}$$

Where

- Ls_k^0 and Ls_k^{-1} are the current and the previous readings of the *k-th* light sensor, respectively. In general, Ls_k^{-i} is the *i-th* previous reading of the *k-th* light sensor.
- $c_i = \begin{cases} 0 & , \quad if\ Ls_k^0 \neq Ls_k^{-i} \\ 1 & , \quad otherwise \end{cases}$
- n is the number of previous steps taken into account during the robot's performance evaluation.

- $D=\left\{\left|Ls_i^0 - Ls_i^{-1}\right|, \forall i \in \left\{\{1,2,3,4,5,6\} - \{k\}\right\}\right\}$
- A normalisation value (Nr) of 100 has been utilised for all of the robot's sensors. Therefore, the abnormality index is in the range 0 to 100.
- The range (R) of the light sensor depends on the maximum value of the ADC digitalising the sensor reading. Therefore, $R_k = 1023$ was utilised in Eq. 4.11.

It must be noted that while expression 4.11 has been successfully utilised for the identification of faulty light sensors, abnormal situations affecting functional light sensors might not be detected by using this method. For instance, a partially covered functional light sensor will not be identified by using expression 4.11. Nevertheless, this kind of abnormality will be identified by the research actions discussed in subsection 4.3.2.

- Calculation of expected position sensor readings.

The outputs of the position sensors are the angles of the leg servo motors. These sensor readings are compared with the expected angles of the servo motors, which are the output of the inverse kinematic model of the robot when the commanded xyz coordinates of the legs are used as inputs. Therefore, by using the range of the position sensors (180 degrees corresponds to the servo motor rotation range), the indexes of abnormality for the position sensors can be determined by means of Eq. 4.10.

- Calculation of expected force sensor readings.

The output of these sensors is directly proportional to the error signal, that is to say, the difference between the current and the target servo positions. Because it is required that the servo motors reach their commanded positions, an expected value of 0 is utilised for the force sensor outputs. Therefore, by considering a range of 0 to 5000, the indexes of abnormality for the force sensors are calculated by means of Eq. 4.10.

- Calculation of expected leg tip force sensor readings.

The leg tip force sensors measure compression and tension of the robot’s legs. When no force is applied to the leg’s tip, the reading of the sensor is around the middle of the scale (initial value). Then, tension and compression of the leg decreases or increases the sensor reading value, respectively. Because, a high magnitude of any of these forces can damage the robot’s legs, it is expected that the sensor readings will be close to their initial values. Consequently, the abnormality index for this sensor is determined by using Eq. 4.12.

$$AIS_k = \begin{cases} Nr\dfrac{TFs_k - TFs\_initial_k}{TFs\_\max_k - TFs\_initial_k} & , \ if\ TFs_k > TFs\_initial_k \\ Nr\dfrac{TFs\_initial_k - TFs_k}{TFs\_initial_k - TFs\_\min_k} & , \ otherwise \end{cases} \tag{4.12}$$

Where

- $TFs\_initial_k$ is the unloaded middle scale value of the *k-th* leg tip force sensor reading.
- $TFs\_\max_k$ is the maximum possible value of the *k-th* leg tip force sensor reading.
- $TFs\_\min_k$ is the minimum possible value of the *k-th* leg tip force sensor reading.

- Calculation of expected accelerometer readings.

The output of this sensor is two angles that indicate the inclination of the robot with respect to the xy-plane (See Fig. 4.1). In this case, the expected sensor readings are the angles corresponding to the direction cosines of the plane formed by the tips of the robot’s supporting legs. Given the non-collinear xyz coordinates of three supporting legs: $P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$ and $P_3(x_3, y_3, z_3)$. The angles of the plane formed by these points with respect to the xy axes of the robot’s coordinate system can be determined by means of Eqs. 4.13 and 4.14.

$$\alpha = \arccos\left(\frac{a}{\sqrt{a^2+b^2+c^2}}\right) \tag{4.13}$$

$$\beta = \arccos\left(\frac{b}{\sqrt{a^2+b^2+c^2}}\right) \tag{4.14}$$

Where

- α is the angle between the supporting plane and the x axis.
- β is the angle between the supporting plane and the y axis.
- $a = (y_3 - y_1)(z_3 - z_2) - (y_3 - y_2)(z_3 - z_1)$
- $b = -(x_3 - x_1)(z_3 - z_2) - (x_3 - x_2)(z_3 - z_1)$
- $c = (x_3 - x_1)(y_3 - y_2) - (x_3 - x_2)(y_3 - y_1)$

The α and β angles are the expected values for Acc_x and Acc_y, respectively. Then, by considering a possible range of 360 degrees, the indexes of abnormality for the accelerometer readings are calculated by means of Eq. 4.10.

- Calculation of expected battery level indicators.

Initially, it is expected that batteries are fully charged (considering a maximum over-voltage of 20% of the battery's nominal voltage). Therefore, the expected sensor value is in the range [818 1023] whereas the sensor range is [0 1023] (1023 being the maximum possible output of the sensor's ADC).

- Calculation of expected whisker readings.

Under normal conditions, the whiskers are expected to provide readings with value 0 (indicating no collision detected). Any larger output indicates the presence of an anomaly. Consequently, the abnormality index for this sensor can be calculated by means of Eq. 4.15.

$$AIS_k = \begin{cases} 0 & , \ if \ Ws = 0 \\ Nr & , \ otherwise \end{cases} \tag{4.15}$$

4.2.3. Classification of Abnormality Levels

The range and resolution of current robot sensors allow the detection of very small changes of the corresponding measured property. In comparison, human beings do not seem to have as many levels of perception. For instance, in [129] six methods for the measurement of clinical pain intensity have been compared. Here patients are asked to identify their level of pain by using different scales. The authors comment that some patients struggle when discriminating between 6 pain levels. If our biological sense of abnormality is extrapolated to robots, then it is necessary to classify the differences between actual and expected robot sensor readings, or indexes of abnormality, into only a few manageable groups. The number of groups utilised for the classification of each index of abnormality will depend on the range of output of the corresponding sensor. This classification can be performed by using Eq. 4.16, where each AIS is labelled by means of an Abnormality Label (AL).

$$AL = \left(AL_1,\ AL_2,\ \dots\ ,\ AL_n\right) \tag{4.16}$$

Where

- $AL_k = \left| \dfrac{Se_k AIS_k}{Nr} \right|$
- Se_k is the sensitivity value for the *k-th* sensor. This is the number of groups into which the abnormality index of the *k-th* sensor is classified $\left(Se_k \in \mathbb{N}\right)$.
- $\lceil Se_k AIS_k / Nr \rceil$ is the nearest integer value higher than $Se_k AIS_k / Nr$.

For instance, if it is necessary to classify the abnormality index of a force sensor into one of three levels of abnormality (low, moderate or extreme), an *Se* value of 3 must be set. Consequently, if the abnormality index of this sensor is in the range [0,100], then the 3 groups formed and their respective range of values will be as indicated by Table 4.4.

Table 4.4. Classification Example.

Abnormality Label (AL)	1	2	3
Level of Abnormality	Low	Moderate	Extreme
Index of Abnormality	[0,33.3]	(33.3,66.6]	(66.6,100]

The previous example corresponds to only one abnormality index. However, this can be expanded to as many dimensions as necessary. Then, because there is an Se value assigned to each sensor, it is possible to specify the number of groups into which each sensor's abnormality index will be classified.

The graph shown in Fig. 4.11 has been obtained after applying the proposed classification algorithm to a simulated data set of abnormality indexes associated with force and position sensor readings. The data, represented by circles, have been normalised to the range [0 50]. Then, this range has been divided into 5 groups for both sensors ($Se_1 = 5$ and $Se_2 = 5$). As a result, each square in Fig. 4.11 encloses data points that represent a different group. For instance, the red rectangle in Fig. 4.11 represents a group of two elements, with abnormality label $AL = (2,2)$.

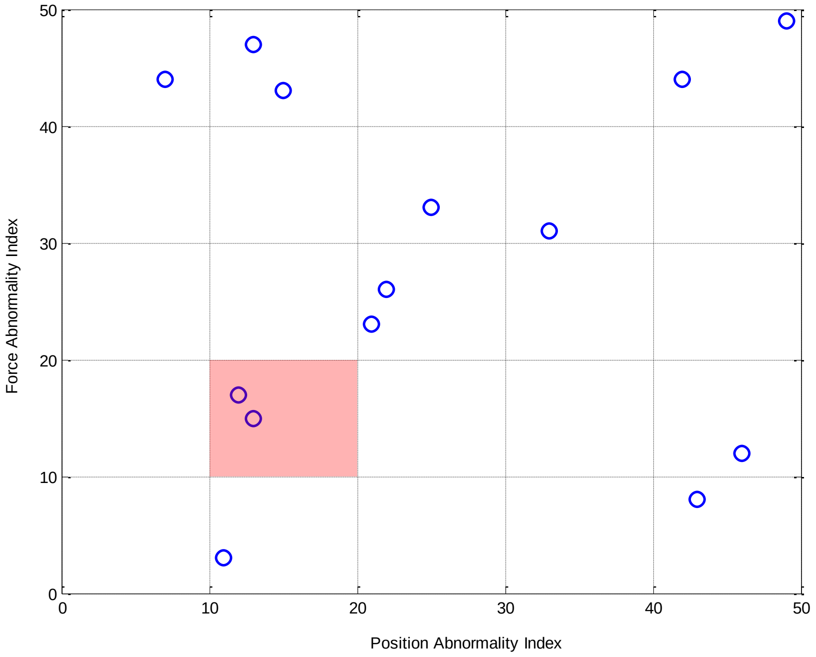


Figure 4.11. Data Clustered with Proposed Classification Method.

The maximum number of groups the classification algorithm can create depends on the target robot's number of sensors and the values of the sensitivity parameter associated with each sensor. For instance, if the abnormality indexes of a robot with 15 sensors are divided in 3 groups, then the classification algorithm will be able to create a maximum of up to 3^{15} (or $1.43x10^{7}$) groups. Ideally, when the sensitivity parameter values are optimally set, the robot will classify each type of anomaly into a different group. In this case, the number of created groups will exclusively depend on the number of different anomalies experienced by the robot and not on the number of sensors the robot has. In practice, it is very unlikely that a robot will face as many as $1.43x10^{7}$ different anomalies. Therefore, they will not all be created. However, if this number of different anomalies did exist and provided that a suitable amount of memory and processing power has been incorporated into the robot, then the proposed algorithm will be still capable of classifying them.

4.2.4. Classification of Abnormal Situations

The abnormality indexes classified during the identification of a detrimental disturbance can be utilised for the classification of the associated abnormal situation. In the case where the anomaly is identified by means of a performance assessment, the abnormal situation can be classified as the centroid of the n corresponding abnormality labels (as indicated for Eq. 4.9, n is the number of previous steps taken into account during the robot's performance evaluation). This can be expressed by means of Eq. 4.17.

$$ASC = \frac{\sum_{k=s}^{s-n} AL^{k}}{n+1} \qquad (4.17)$$

Where

- ASC is the Abnormal Situation Centroid.
- AL^{k} is the Abnormality Label of the AIS associated with the *k-th* step.

If the abnormal situation is identified by means of the pain mechanism, the ASC is simply determined by using $n = 0$ in Eq. 4.17. This is because only the current sensor reading set is utilised in pain identification. The abnormal situation centroid will be used by the method presented in chapter 5, which aims to provide robots with a technique for recognising previously experienced abnormal situations.

The abnormality labels associated with the last 10 steps taken by the robot during the experiment with the covered light sensor have been added to form the Cumulative Abnormality Label (CAL) illustrated in the graph of Fig. 4.12. This is the numerator of expression 4.17. As expected, the covered light sensor (Ls2) shows the maximum abnormality levels. However, the leg tip force sensors, and particularly TFs2, also show high cumulative abnormality labels. This is because the rotations on the spot performed by the robot as depicted in Fig. 4.7 generate extra pressure in the robot's legs. If the cumulative abnormality labels are divided by the number of steps considered for the performance evaluation of the robot (10 in this case), the abnormal

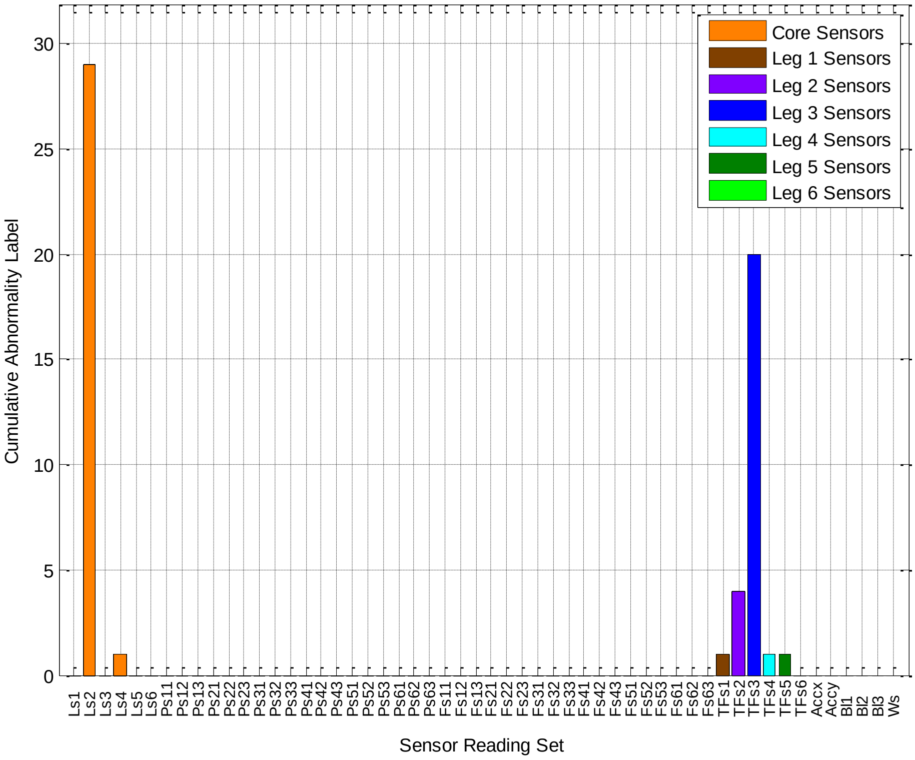


Figure 4.12. Cumulative Abnormality Label of the Experiment with the Covered Light Sensor.

situation centroid is obtained. As a result, the abnormal situation where Ls2 is covered can be represented by the following abnormal situation centroid.

$$
\begin{aligned}
ASC = \{&0\ 2.9\ 0\ 0.1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \ldots \\
&\ldots 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0.1\ 0.4\ 2\ 0.1\ 0.1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\}
\end{aligned}
$$

Similarly, an ASC can be calculated and used for representing any abnormal situation experienced by the experimental robot.

4.3. Identification of Detrimental Disturbance Sources

In this section two techniques for the identification of detrimental disturbance sources are presented. The first method uses information provided by the physical location of a robot's sensors and the robot's moving parts in the creation of a hierarchical list of possible anomaly sources. These items are confirmed or rejected by the second process, which uses a series of robot actions especially developed for this purpose.

4.3.1. Body Map

A starting point, from which a robot could list possible causes of a lack of progress, is to consider the robot's anatomy. The physical distribution of sensors and moving parts on the robot's body could provide useful information regarding the location of a hardware malfunction or a feature of the robot's environment that degrades the robot's performance. This seems to correspond with what we humans do when experiencing pain or when some part of our body collides with our physical environment. That is to say, we focus our attention on that part of our body sensing pain or contact with our environment.

In some cases a malfunction has stronger effects on sensors that are physically further away from the hardware experiencing it. However, as this is not the most common situation, it still seems reasonable for a robot to first look for the origin of a malfunction in the area surrounding the sensors detecting it. This could accelerate the search for the cause of a detrimental disturbance. If we consider a robot with *i* moving parts and *k* sensors, it is possible to utilise expression 4.18 in order to determine the moving parts that are closer to those sensors showing high levels of abnormality.

$$w_{MPi} = \sum_{k=1}^{N} \frac{CAL_k}{\left\| P_{Sk} - P_{MPi} \right\|} \tag{4.18}$$

Where,

- w_{MPi} is the weight associated with the *i-th* moving part.
- $\left\| P_{Sk} - P_{Ai} \right\|$ is the Euclidean distance between the physical location of the *k-th* sensor $\left(P_{Sk} \right)$ and the *i-th* moving part $\left(P_{MPi} \right)$.
- If $\left\| P_{Sk} - P_{MPi} \right\| = 0$ then $w_{MPi} = 1$.
- N is the number of sensors in the robot.

A fact that is important to note before using expression 4.18 is that not all of a robot’s sensors provide information that is often related to a specific part of the robot’s body. Some examples of this kind of sensors are: accelerometers, gyroscopes and battery level indicators. These sensors should not be included in expression 4.18.

As a result of using expression 4.18, the weights associated with each moving part are determined. Then, those moving parts with the higher weights should be considered first as possible causes of a detrimental disturbance. These assumptions must be rejected or validated by the robot via a series of research movements. Therefore, those moving parts with higher associated weights are the first to be considered for executing the research actions. Examples of using the body map will be presented in section 4.4.

4.3.2. Research Actions

In order to reject or validate a robot's theories about causes of a lack of progress in its mission, the robot must be able to perform a set of research actions. Although this set could be different depending on the type of the robot's moving parts, environment and application; some examples are available in the literature. For instance, the authors in [56] incorporate two reflexes for walking robots, the elevator and the searching reflex. The elevator reflex is used when a swinging leg encounters an obstacle. Then the leg is moved backwards and lifted before swinging forward. If the reflex is triggered again, then the leg is lifted higher the next time. This reflex could be used as a research action to reject or validate a robot's theory about the existence of an obstacle obstructing the leg. On the other hand, a searching reflex is utilised when the robot's leg loses ground contact. Here, the leg's tip is moved in circles of increasing radius for a fixed number of times. This reflex could be used to determine the validity of the robot's assumptions about the existence of a hole in its path.

There is no need to create a research action for every possible situation. But, the research action set should be as general as possible. By combining different research actions a large range of the possible robot actions, rather than the much larger spectrum of possible situations that could affect the robot, should be covered.

After the set of research actions has been established, the robot must select the subset of actions that is suitable for the moving parts involved in the research process. These are the moving parts with the largest associated weights (calculated by means of expression 4.18). The research actions do not aim to solve the problem that is generating the poor performance of the robot. At this stage, the goal is just to identify its source. So, after executing these actions and analysing their effect on the sensor readings of the robot, the list of possible causes of the problem should be reduced. As an example, some of the research actions utilised by the experimental robot are presented next. In general, before executing these actions, the robot is sent to its home position with legs raised off the ground. This allows decoupling the robot's legs, so the robot's actions only have localised effects.

- Servo motor research actions.

The servo motor research action consists of rotating the motor from the minimum to the maximum possible rotation angle considering that the legs should not collide with the rest of the robot's body. Each time the servo reaches one of these extreme positions, the abnormality indexes of force and position sensor readings are analysed. As a result, a high abnormality index value in one of the sensors will indicate a high probability that the sensor is faulty, there is something abnormal in the robot's environment or the robot structure is damaged.

- Leg tip force sensor research actions.

When executing this action, the robot's leg with the suspect leg tip sensor is progressively lowered. As this occurs, the system waits for changes in two consecutive leg tip force sensors readings. Then, if the sensor works as expected, these changes should be generated as soon as the leg touches the floor. However, if no changes are detected after the leg is lowered below a threshold position, the system assumes that either the sensor is faulty or it is the interaction between the robot and its environment what is causing the problem.

- Accelerometer research action.

The accelerometer is one of the sensors which are not included in the body map presented in section 4.3.1. In this case all the robot's legs execute the research actions, which consist of tilting the robot. Thus, the robot is tilted forward, backward, to the right and to the left. After each tilt, the abnormality index is calculated. As a result, a high abnormality index is associated with a high probability of accelerometer malfunction. A video showing the experimental robot executing accelerometer research action can be found by following the link D04 in Appendix D.

- Light sensor research action.

The light sensor research actions consist of a series of robot rotations. After each rotation all of the light sensor readings are recorded. As illustrated in Fig. 4.13, the rotation angle is equal to the angle between adjacent light sensors. The number of light sensors in the robot determines the number of research rotations, the number of points (represented by orange circles in Fig. 4.13 for the initial configuration of 6 operational light sensors) where the light measures are taken; and the number of light sensor readings recorded at each point. Once the robot has finished executing the research rotations and recording light sensor readings at each point, abnormal light sensor conditions are identified by means of expression 4.19.

$$Ls_k \text{ status} = \begin{cases} normal & , \; if \; \sum_{p=1}^{N} c_k^p \leq N/2 \\ abnormal & , \; otherwise \end{cases} \tag{4.19}$$

Where,

- $c_k^p = \begin{cases} 0 & , \; if \; \overline{Ls^p} \text{-} 1.5\sigma^p \leq Ls_k^p \leq \overline{Ls^p} + 1.5\sigma^p \\ 1 & , \; otherwise \end{cases}$ is a binary variable associated with the *k-th* light sensor when this is measuring the light at point *p* (illustrated in Fig. 4.13).
- N is both, the number of operational light sensors and the number of *p* points.
- $\overline{Ls^p}$ is the mean value of the *k* light sensor readings registered at point *p*.
- σ^p is the standard deviation of the *k* light sensors readings registered at point *p*.
- Ls_k^p is the sensor reading registered at point *p* by the *k-th* light sensor.

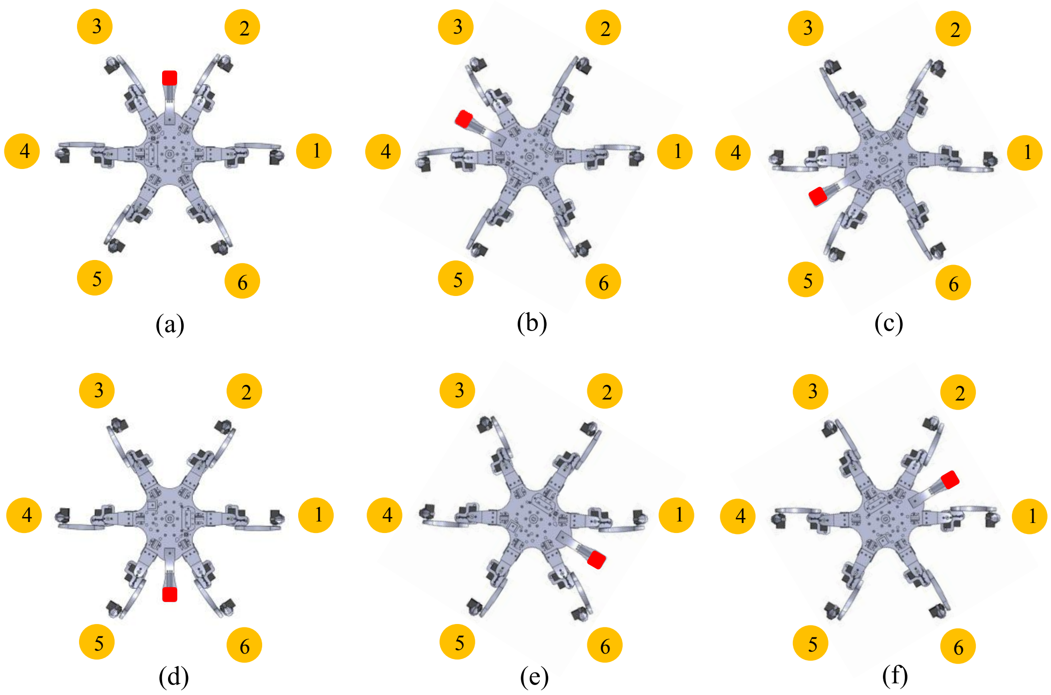


Figure 4.13. Light Sensor Research Rotations.
(Note that the Red Square Indicates the Front of the Robot).

4.4. Illustrative Examples

This section illustrates the application of the classification method presented in this chapter by means of 2 experiments. As before, the robot's mission is to find a single light source. Initially, the robot takes a few steps under normal conditions, but then a different detrimental disturbance is introduced in each test. These experiments and associated results are presented next.

4.4.1. Disconnection of Servo Motor Control Signal

In this experiment, the PWM control signal of Servo 1, located in leg 1, was disconnected after the robot had taken 10 steps. This motor will be referred as the target servo in this subsection. The robot's performance during this experiment is shown in Fig. 4.14. Once the servo control signal is disconnected, the robot is still able to make progress until step 33. Then, because the robot is unable to make further progress after 10 steps; a cyclic lack of progress is detected at step 42.

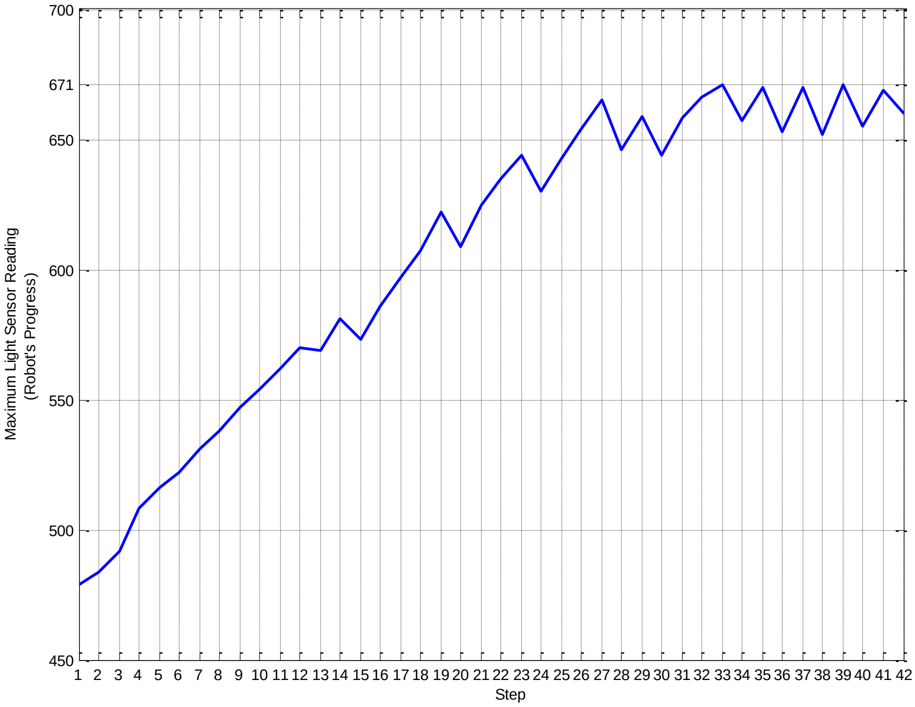


Figure 4.14. Robot's Performance during the Experiment with a Disconnected Servo Motor Control Signal.

The abnormality index set of the robot’s sensors during steps 33 and 42 has been labelled. As a result of adding all these labels, the cumulative abnormality label depicted in the graph of Fig. 4.15 has been obtained. In this case, the largest cumulative abnormality label corresponds to abnormal readings of Ps11, the position sensor associated with the target servo.

However, the displacement of leg 1 spreads its effects to other parts of the robot. This is mainly detected by leg tip force sensors and position sensors of the legs located on the same side of the robot as the target servo (legs 1 and 6). In addition, the leg tip force sensor associated with leg 3 also shows a high cumulative abnormality label. This occurs as a result of the displacement of leg 1 when legs 1, 3 and 5 are the robot’s stance legs. The abnormal situation examined during this experiment can be classified by using expression 4.17 into the following abnormal situation centroid.

$$\begin{aligned} ASC = \{&0\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 0\ 0\ 0\ 0.5\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0.5\ 0\ 0\ 0\ 0 \ldots \\ &\ldots 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0.1\ 0\ 0.5\ 0.1\ 0.5\ 0\ 0\ 0.7\ 0\ 0\ 0\ 0\ 0\ 0\} \end{aligned}$$

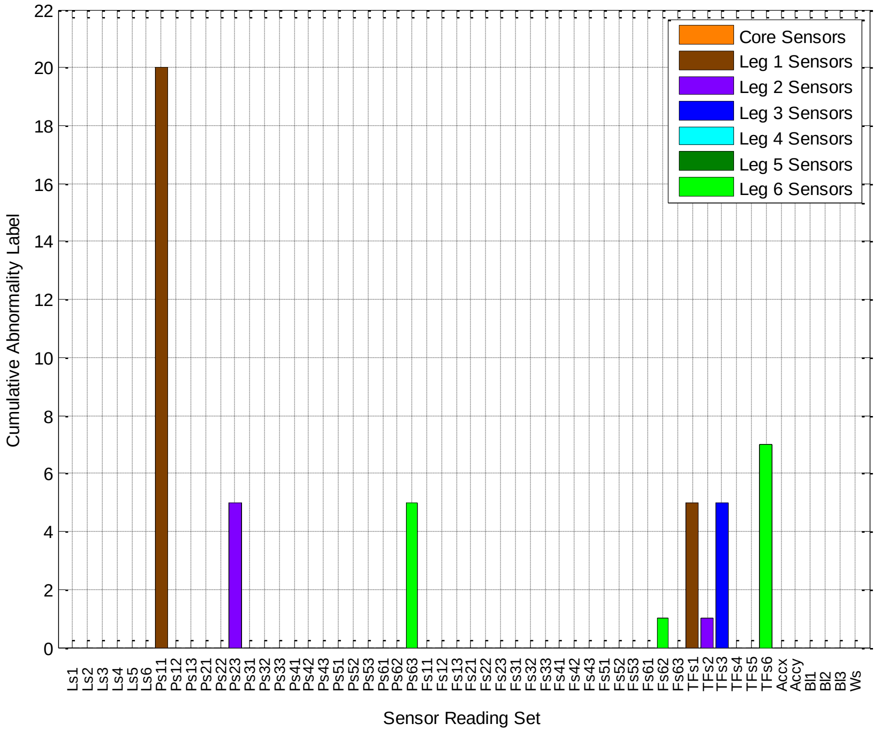


Figure 4.15. Cumulative Abnormality Label during the Experiment with the Disabled Servo Motor Control Signal.

The weights associated with each one of the 18 servo motors in the experimental robot are shown in Table 4.5, where S_{xy} represents the servo motor *y* located in leg *x*. These weights indicate the order in which the robot's moving parts must execute the research actions.

Table 4.5. Moving Part Weights during Experiment with Disabled Servo Motor Control Signal.

S_{11}	S_{12}	S_{13}	S_{21}	S_{22}	S_{23}	S_{31}	S_{32}	S_{33}
20.12	0.92	0.32	0.26	0.27	5.16	0.18	0.17	0.15
S_{41}	S_{42}	S_{43}	S_{51}	S_{52}	S_{53}	S_{61}	S_{62}	S_{63}
0.15	0.14	0.12	0.17	0.16	0.14	0.31	1.28	5.19

The relation between location and weight of the robot's moving parts is more clearly illustrated in Fig. 4.16. Here, the size of the circles representing the leg servo motors is related to the associated weight magnitude. As can be seen in the figure, servo motor 1 of leg 1 is the moving part with the largest weight. This is followed by servo 3 of legs 2 and 6. Therefore, these parts are the first to be selected for executing the corresponding research actions. Because the largest cumulative abnormality label corresponds to a position sensor, a servo motor research action is executed by S_{11}.

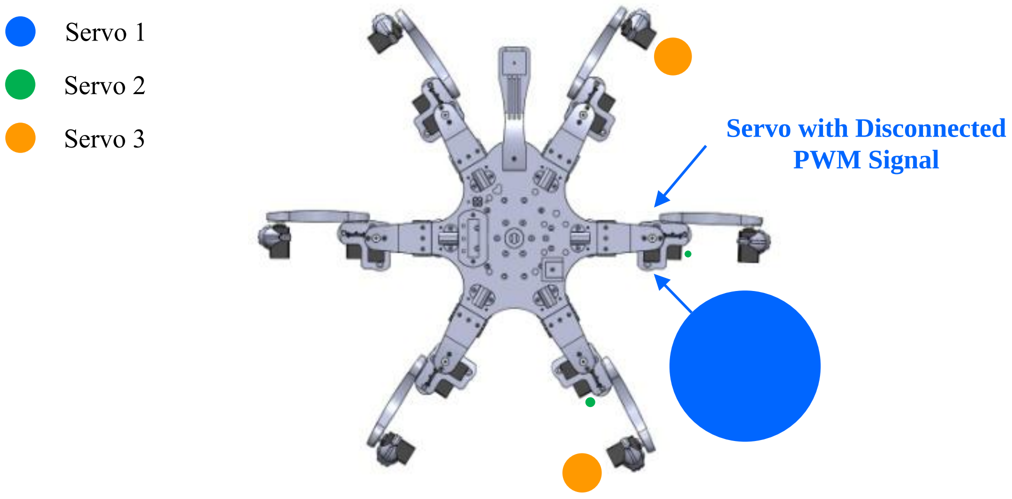


Figure 4.16. Weight Distribution among Robot's Moving Parts during Experiment with Disabled Servo Motor Control Signal.

Then, the system is able to identify inconsistencies between the commanded coordinates of the servo and the position sensor readings. As a result, the system determines that the source of the detrimental disturbance experienced by the robot is a fault in S_{11} or its position sensor, something in the robot’s environment or damage in the robot’s structure that is affecting S_{11} normal operation.

4.4.2. Robot’s Legs over Soft Terrain

In this experiment, the robot walked under normal conditions until step 11. Then, leg 3 (at step 12) and leg 4 (at step 21) walked onto a foam sponge sheet as shown in Fig. 4.17. The idea of the experiment is to simulate an environment where some of the robot’s legs experience resistance when walking. For instance, a robot could walk over a soft terrain such as grass, sand, mud or snow. Because the robot’s legs could sink into the terrain, extra force will be necessary to perform the gait. In fact, the resistance offered by the sponge utilised during the experiment was enough to prevent the robot from getting closer to the light source. The robot’s performance during this experiment is shown in Fig. 4.18.

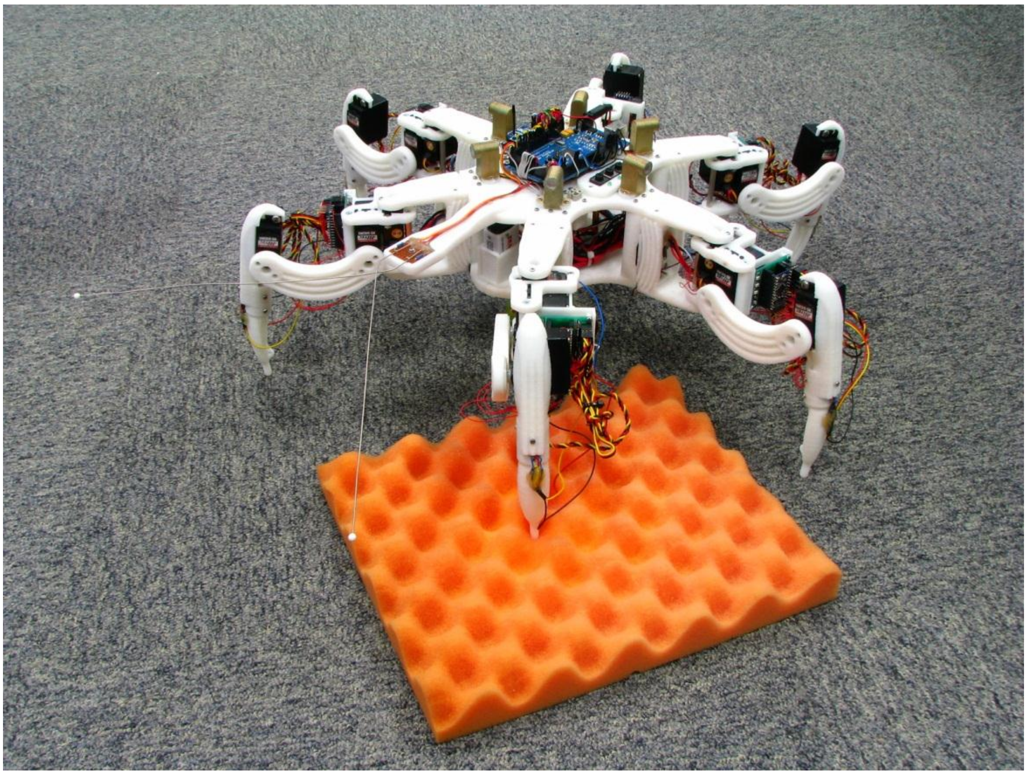


Figure 4.17. Experimental Robot Walking over Soft Terrain.

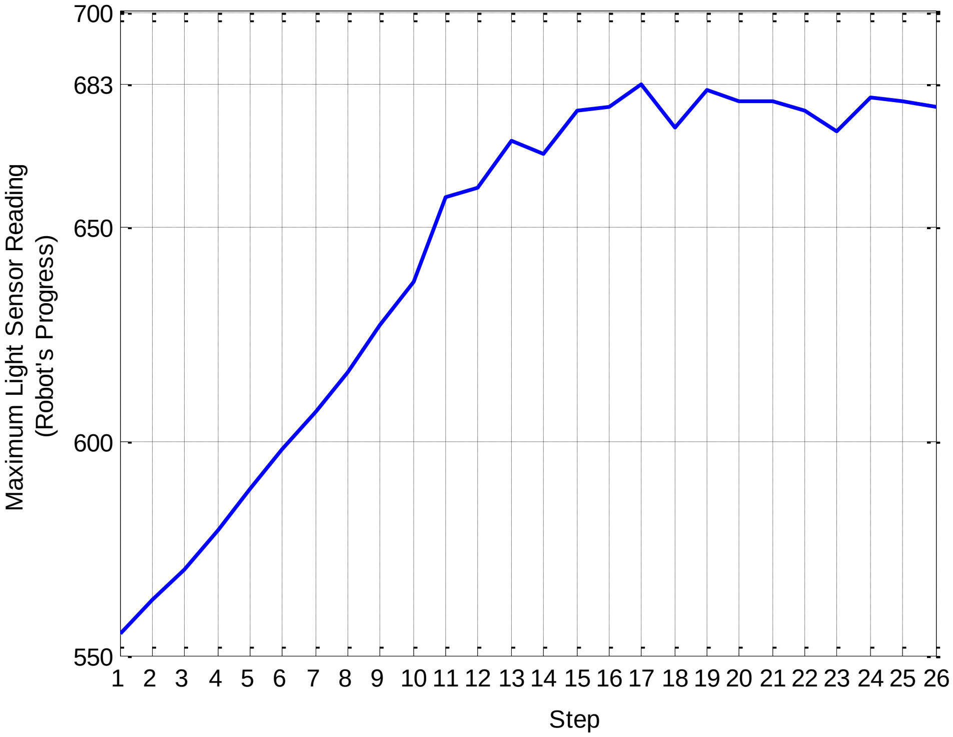


Figure 4.18. Robot's Performance when Encountering Soft Terrain.

Once leg 3 steps onto the sponge, the robot is still able to make progress until step 17. Then, because the robot is unable to make further progress after 10 steps; a cyclic lack of progress is detected at step 26.

The abnormality index set for the robot's sensors during steps 17 and 26 has been labelled. As a result of adding all these labels, the cumulative abnormality label depicted in the graph of Fig. 4.19 has been obtained. In this case, the largest cumulative abnormality label corresponds to abnormal readings of TFs3, the leg tip force sensor associated with leg 3. However, the extra force applied by the legs while they try to move the robot forwards and the displacement of the legs caught in the sponge spreads abnormal readings to other parts of the robot. This is mainly detected by servo force and leg tip force sensors of legs 1 and 5. That is to say, an opposite leg and an adjacent leg to one of the legs caught in the sponge (leg 1 is opposite and leg 5 is adjacent to leg 4). The abnormal situation examined during this experiment can be classified by using expression 4.17 into the following abnormal situation centroid.

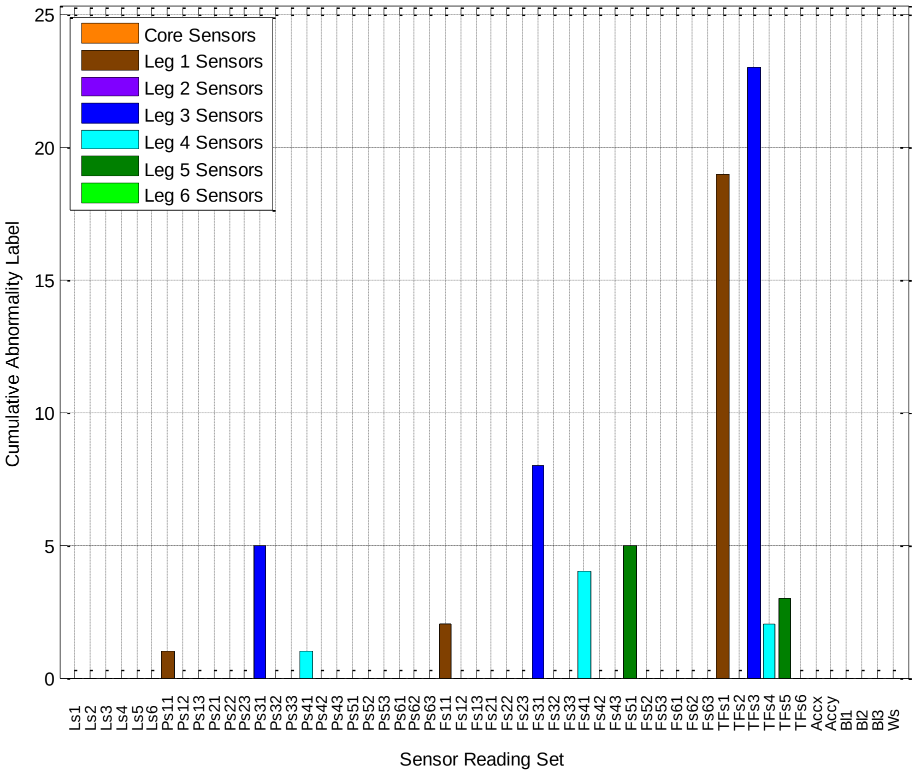


Figure 4.19. Cumulative Abnormality Label when Encountering Soft Terrain.

$$ASC = \{0\ 0\ 0\ 0\ 0\ 0\ 0.1\ 0\ 0\ 0\ 0\ 0\ 0.5\ 0\ 0\ 0.1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0.2\ 0\ 0\ 0 \ldots$$
$$\ldots 0\ 0\ 0.8\ 0\ 0\ 0.4\ 0\ 0\ 0.5\ 0\ 0\ 0\ 0\ 0\ 1.9\ 0\ 2.3\ 0.2\ 0.3\ 0\ 0\ 0\ 0\ 0\ 0\ 0\}$$

As in subsection 4.4.1, the weights associated with each one of the 18 servo motors in the experimental robot have been determined for establishing the order of the research actions. These weights are shown in Table 4.6.

Table 4.6. Moving Part Weights during Robot's Legs Over Soft Terrain Experiment.

S_{11}	S_{12}	S_{13}	S_{21}	S_{22}	S_{23}	S_{31}	S_{32}	S_{33}
3.27	0.39	0.29	0.32	0.30	0.26	13.26	0.77	0.39
S_{41}	S_{42}	S_{43}	S_{51}	S_{52}	S_{53}	S_{61}	S_{62}	S_{63}
5.29	0.47	0.28	5.23	0.41	0.24	0.26	0.25	0.21

The relation between location and weight of the robot's moving parts for this experiment is illustrated in Fig. 4.20. As can be seen in the figure, only servo motor 1 of legs 1, 3, 4 and 5 have visible representing circles. This is a consequence of the robot's inability to move forward. In this experiment, servo motor 1 of leg 3 is the moving part with the largest weight. This is followed by servo 1 of legs 4 and 5. Therefore, these parts are the first in executing the corresponding research actions. Because the largest cumulative abnormality label corresponds to a leg tip force sensor, a leg tip force sensor research action is executed by leg 3. However, this test

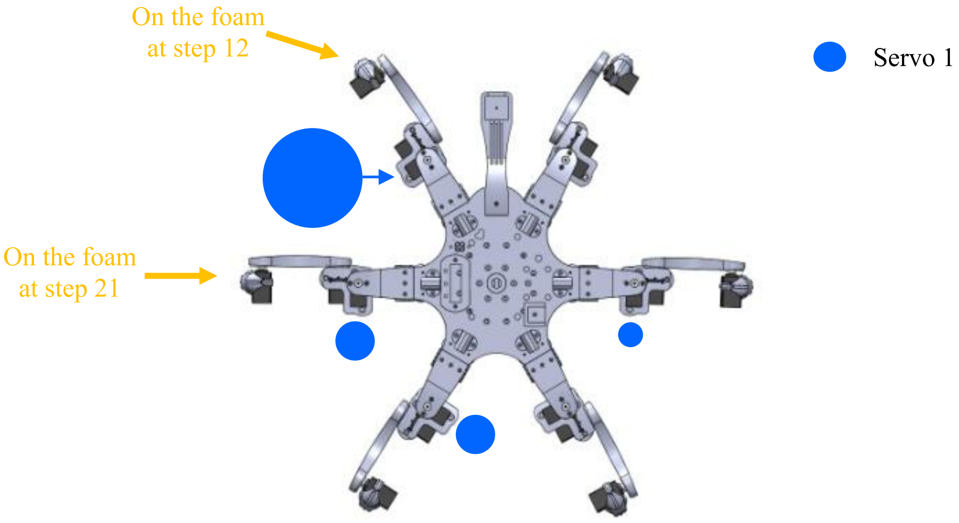


Figure 4.20. Weight Distribution among Robot's Moving Parts during Robot's Legs over Soft Terrain Experiment.

does not reveal abnormal conditions. Consequently, the second test on the list is performed. The following two largest cumulative abnormality labels shown in Fig. 4.19 correspond to force and position sensors of servo S_{31}, respectively. Therefore, servo motor research actions are executed by S_{31}. Then, the system identifies inconsistencies between the commanded coordinates of the servo and the position sensor readings; and abnormal values of force sensors readings. As a result, the system is able to identify that the source of the detrimental disturbance experienced by the robot is either a fault in S_{31} or its associated position and force sensors; or something in the robot's environment that is affecting S_{31} normal operation.

4.5. Additional Experimental Results

Experimental validation of self-diagnosis methods presented in the thesis will be addressed by conducting three groups of experiments. In the first series of experiments, failures will be introduced into the robot's hardware and, under normal environmental conditions; the performance of the methods presented in this chapter will be analysed. Another group of experiments will be used for examining the performance of the proposed methods when the robot's hardware is fully functional but abnormal environmental conditions arise. The last set of experiments will allow the evaluation of the proposed self-diagnosis methods in the presence of both robot's hardware failures and abnormal environmental conditions occurring at the same time.

Because it is unfeasible to expose the robot to every possible situation or even consider all of the possible combinations of faults that the robot may undergo on each situation, only a small subset has been selected. The experiments as a whole consider faults that affect all of the sensors of the robot, moving parts, robot's body, low battery levels and even a simulated software bug. Due to time constraints and considering the scope of the thesis, the self-diagnosis methods are only tested in the experimental robot and not on other robotic platforms. However, the experimental robot is equipped with an array of sensors and compensation capabilities that are superior to the ones commonly found on other robotic systems. In any case, the experimental robot is able to provide a large amount of suitable data for the evaluation of the proposed methods. Once again, the robot's mission is to find a single light source. During the conducted experiments, the robot was set to identify a sustained or cyclic lack of progress by using sets of 7 steps ($n = 7$ in Eq. 4.9). These experiments and their associated results are presented next.

4.5.1. Experiments with Anomalies in the Robot's Hardware.

These experiments were designed to evaluate the self-diagnosis methods presented in this chapter when the robot deals with faults in its hardware. When a lack of progress in the robot mission is detected, the robot generates a table of Cumulative Abnormality Labels which are shown as bar graphs. The legend of these graphs is the same as previously utilised in section 4.4 and it is also shown in Fig. 4.21.

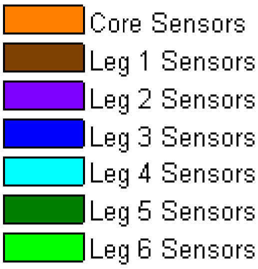


Figure 4.21. Legend of Cumulative Abnormality Label Bar Graphs.

CAL values are analysed in order to evaluate if the system is able to identify the considered fault in the robot's hardware. Each one of the robot's theories about the source of the anomaly is analysed according to the order established by the system and they are verified by the execution of the corresponding research action. If the system determines that a theory is not valid, the next theory is tested and so on. Then, the system continues in this fashion until the anomaly is identified or the robot runs out of theories. Each diagnosis attempt demands energy and time from the robot and divert it from performing its mission. Therefore, the best possible result occurs when the system is able to diagnose the problem correctly on its first attempt. Two criteria will be utilised in this subsection for the evaluation of the self-diagnosis:

i) Whether or not the system is able to diagnose the anomaly correctly.

ii) Number of attempts required when the correct diagnosis was accomplished.

4.5.1.1 Disconnected Control Signal of Leg Servo Motors.

As the experimental robot has 18 servo motors which are prone to overheating and physical damage, it is important to evaluate if the system is able to detect and diagnose a faulty servo. This situation has been represented by disconnecting the PWM control signal of the motor simulating the fault. The conducted experiments consider combinations of number and position of faulty motors. However, due to the huge number of possible combinations, it is not feasible to consider all of them. Therefore, only a small subset of experiments, which includes faulty motors located in up to 3 legs, has been considered.

In a first set of experiments only one servo motor was disconnected at a time. Figures 4.15 (where the CAL associated with the disconnection of servo 1, in leg 1, is depicted) and 4.22 to 4.24 show the results of these experiments in the form of Cumulative Abnormality Labels (CALs). As illustrated in Fig. 4.1, servos 1, 2 and 3 provide the robot's leg with coxa, femur and tibia articulations, respectively.

Ideally, the effects of the disconnection of one or more of the robot's servo motors should only increase the CAL value of the position sensor corresponding to the disconnected motor(s). However, in practice these effects are spread to other sensors. As a result of the disconnection, the robot's legs can reach awkward positions which increase CALs associated with force and pressure sensors. In addition, the robot may lose stability, which is register by CALs corresponding to accelerometer readings.

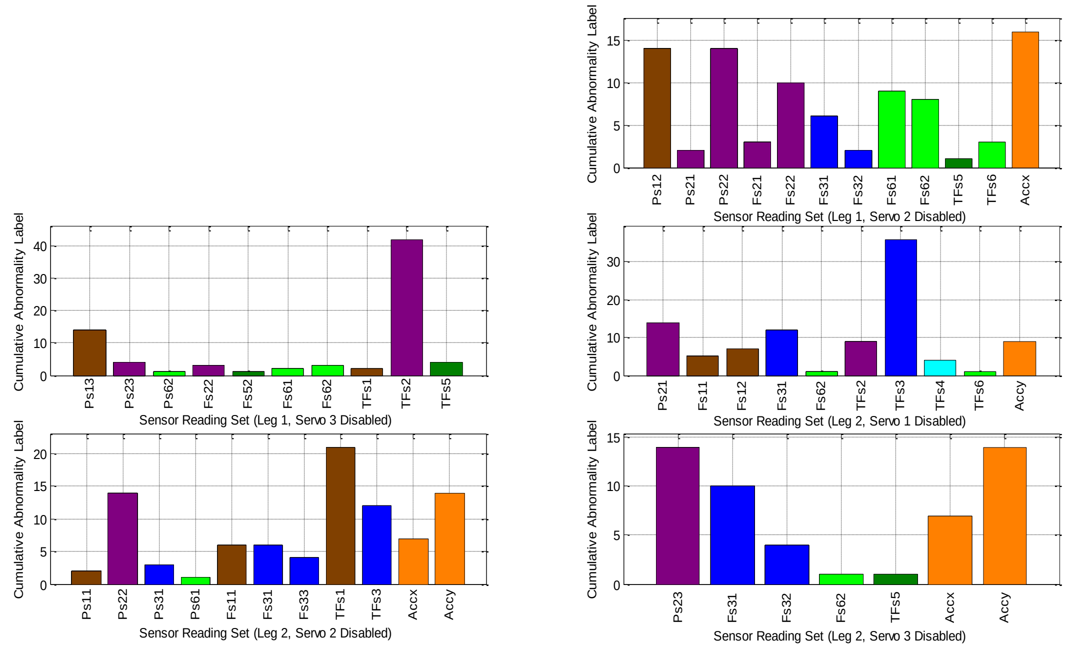


Figure 4.22.Cumulative Abnormality Labels from Experiments with One Disabled Servo Motor in Legs 1 and 2.

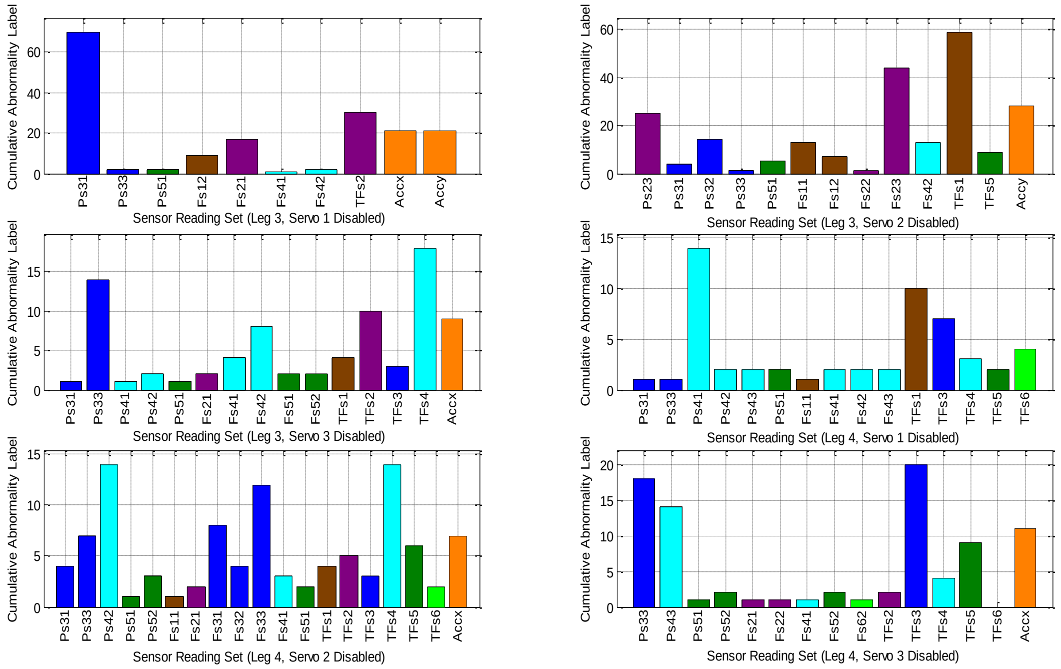


Figure 4.23. Cumulative Abnormality Labels from Experiments with One Disabled Servo Motor in Legs 3 and 4.

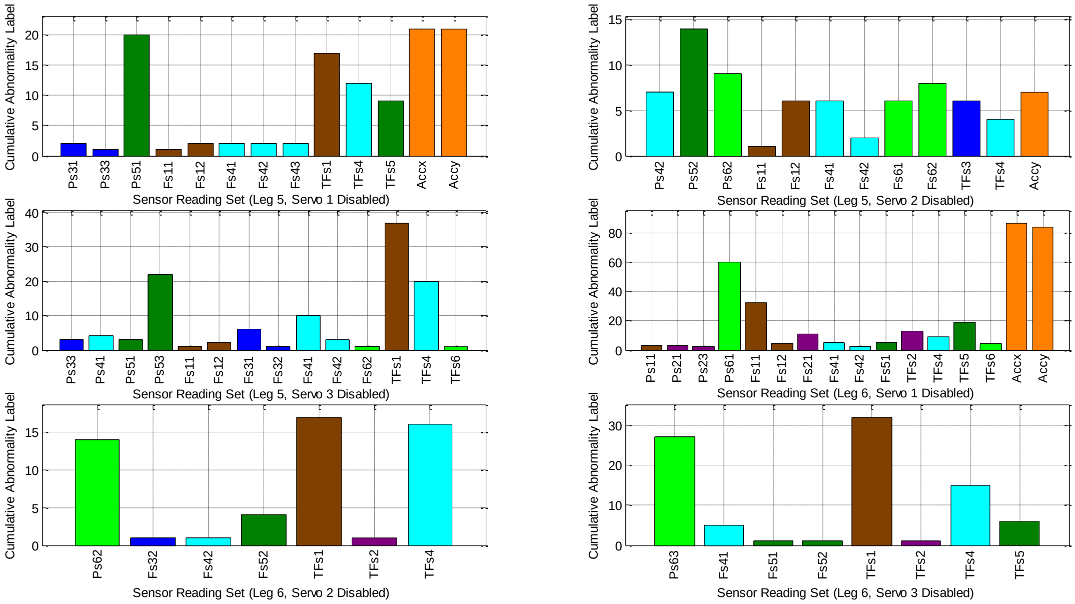


Figure 4.24. Cumulative Abnormality Labels from Experiments with One Disabled Servo Motor in Legs 5 and 6.

The best results for this set of experiments were obtained when the following servo motors were independently disconnected.

- Servo 1 of Leg 1
- Servo 3 of Leg 2
- Servo 1 of Leg 3
- Servo 1 of Leg 4
- Servo 2 of Leg 4
- Servo 2 of Leg 5

In all of these cases, the system was able to immediately identify the source of the anomaly. Then, in experiments where the following servos were independently disconnected, the system was able to identify the source of the abnormal situation on a second attempt.

- Servo 2 of Leg 1
- Servo 3 of Leg 1

- Servo 1 of Leg 2
- Servo 2 of Leg 2
- Servo 3 of Leg 2
- Servo 3 of Leg 3
- Servo 3 of Leg 4
- Servo 3 of Leg 5
- Servo 3 of Leg 6

For instance, during the experiment where servo motor 3 of leg 1 was disconnected, the robot executed accelerometer research actions. Consequently, it was corroborated by the robot that this sensor was functional. Then, the robot continued executing servo motor research actions for servo motor 3 of leg 1. As a result, the source of the anomaly was found.

When other leg servo motors were independently disconnected, the robot was able to identify the source of the anomaly on its third attempt.

Another set of experiments was conducted in order to test the capability of the self-diagnosis methods to identify the simultaneous disconnection of two servo motors (servos 2 and 3 of the corresponding robot's leg). The cumulative abnormality labels of these experiments are shown in Fig. 4.25. Here, the best results were obtained when servo motors 2 and 3 were disconnected in legs 5 and 6 (hind legs). In these cases, the system had no difficulty identifying the source of the anomaly on its first attempt. When servo motors 2 and 3 on legs 1, 2 and 3 were disconnected, the system was able to immediately identify one of the disconnected servo motors. The other one was identified only after performing servo motor research actions. Most difficulties arose when servo motors 2 and 3 were disconnected in leg 4. In this case, there were high CAL values associated with leg 3 (in blue). Therefore, only after the robot had checked that everything was normal with this leg, were the disconnected

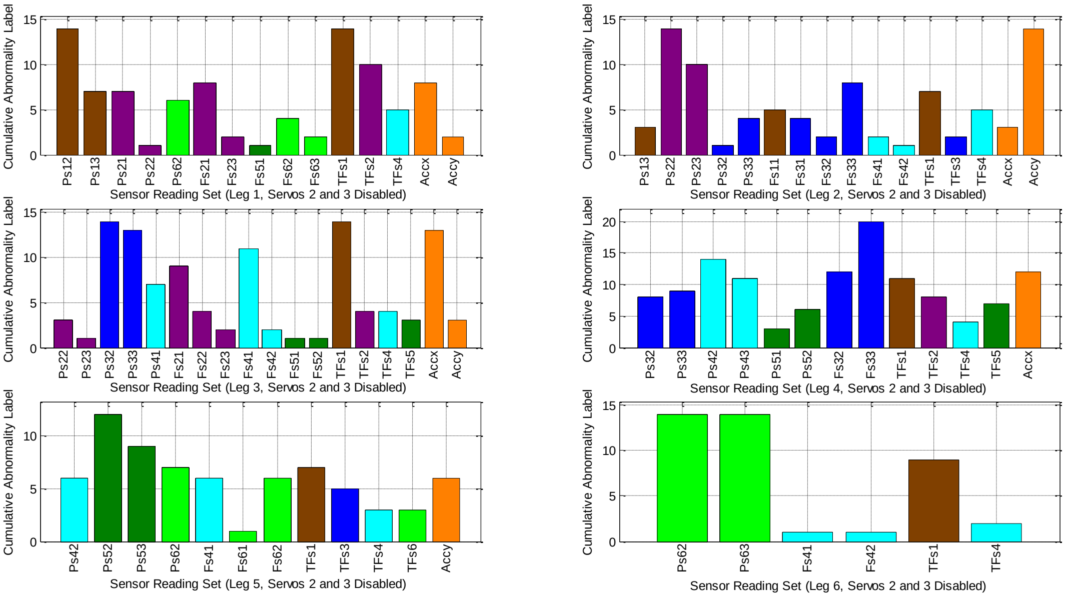


Figure 4.25. Cumulative Abnormality Labels from Experiments with Disabled Servo Motors 2 and 3.

servos in leg 4 identified. Table 4.7 shows the number of attempts necessary for the identification of the disconnected servos during the conducted experiments.

Table 4.7. Number of Attempts for Identification of Disabled Servo Motors 2 and 3.

Leg	Servo	Number of Attempts
1	2	1
	3	6
2	2	1
	3	3
3	2	1
	3	3
4	2	2
	3	5
5	2	1
	3	2
6	2	1
	3	2

Experiments were also conducted by disconnecting servo motors located in different legs. Fig. 4.26 shows the obtained CAL values after disconnecting servo motor 1 in different pairs of legs. The effects of the disconnection of servo 1 are notoriously propagated to other legs. This is a result of the awkward position the not fully controlled legs adopt, which increase the force required from other servos in order to

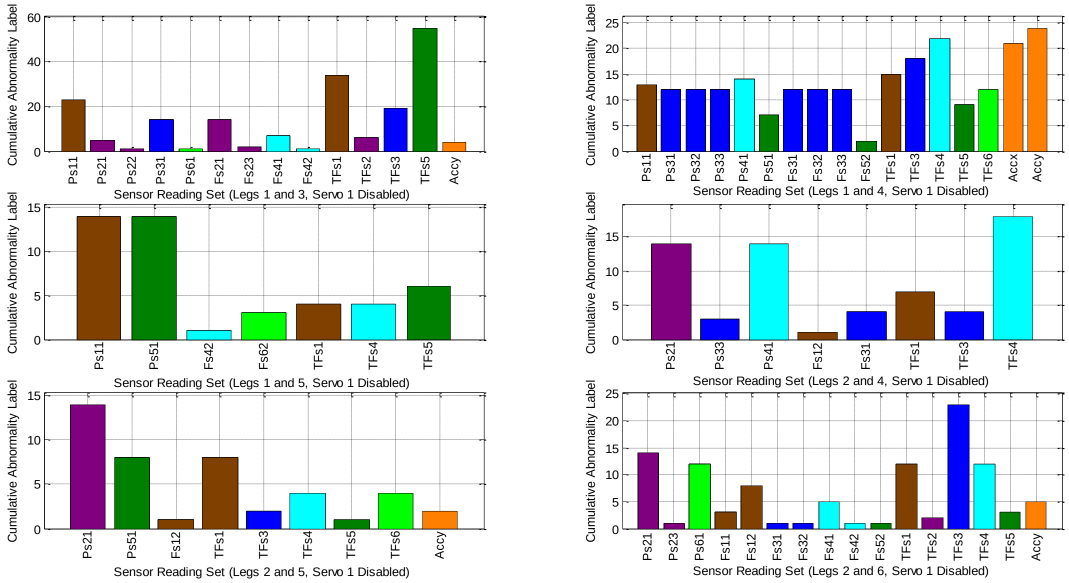


Figure 4.26. Cumulative Abnormality Labels from Experiments with Disabled Servo Motor 1 in Different Pairs of Legs.

maintain their commanded positions. In addition, the robot may lose stability, which increases CAL values corresponding to accelerometer readings.

The best results for these experiments were obtained when servo motor 1 of legs 1 and 5; and legs 2 and 5 were disconnected. Here the system was able to recognise the disconnected servos at its first attempt. Then, 2 attempts were necessary for the identification of the disconnected servo in legs 2 and 4, 3 attempts for the servo in legs 2 and 6; 4 attempts for the one in legs 1 and 3; and 6 attempts for the one in legs 1 and 4. Table 4.8 shows the number of attempts necessary for the identification of the disconnected servos during this set of experiments.

Table 4.8. Number of Attempts for Identification of Disabled Servo Motors 1 in Different Pairs of Legs.

Experiment	Leg	Servo	Number of Attempts
1	1	1	3
	3	1	5
2	1	1	7
	4	1	6
3	1	1	1
	5	1	2
4	2	1	2
	4	1	3
5	2	1	1
	5	1	2
6	2	1	2
	6	1	3

4.5.1.2 Disconnected Light Sensors.

During these experiments the robot performed its mission with one or a combination of disconnected light sensors. These disconnections may be produced by a cut wire. Initially, each one of the robot's light sensors was disconnected in a separate experiment. The results in the form of cumulative abnormality labels are shown in Fig. 4.27. In general and for simplicity purposes, the graphs of this section will not show negligible CAL values (≤ 1). As a result, in this case only CALs associated with light sensors are shown. As can be observed in the figure, in this series of experiments 100% of the disconnected light sensors were identified by the system at the first attempt.

In order to test the methods described in the chapter with more challenging situations a new set of 6 experiments was conducted. Here, the capability of the described methods for detecting combinations of different numbers of disconnected light sensors (between 2 and 5) was tested. Results of these experiments are shown in the graphs of Fig. 4.28.

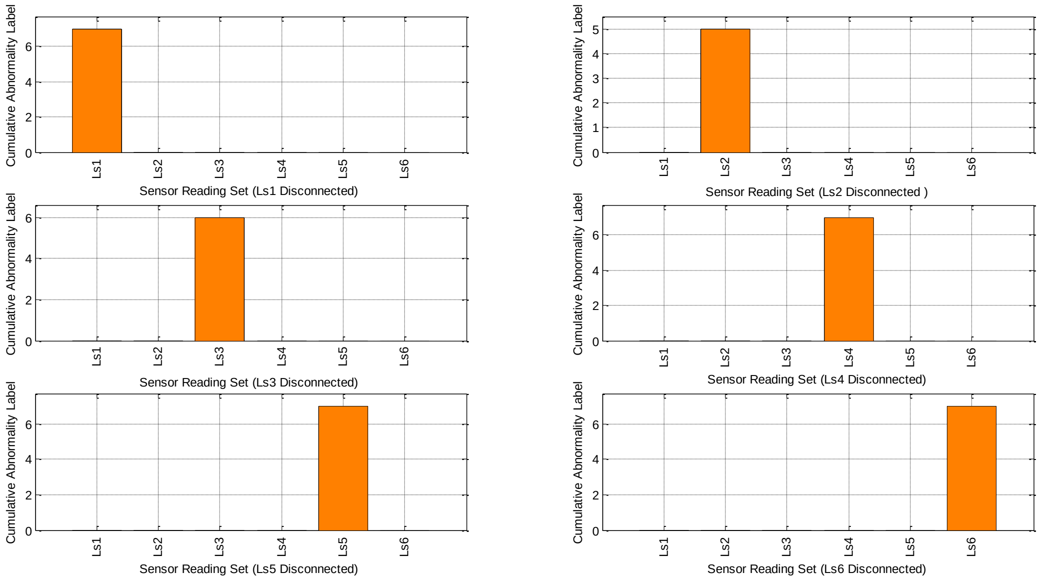


Figure 4.27. Cumulative Abnormality Labels from Experiments with One Disconnected light Sensor.

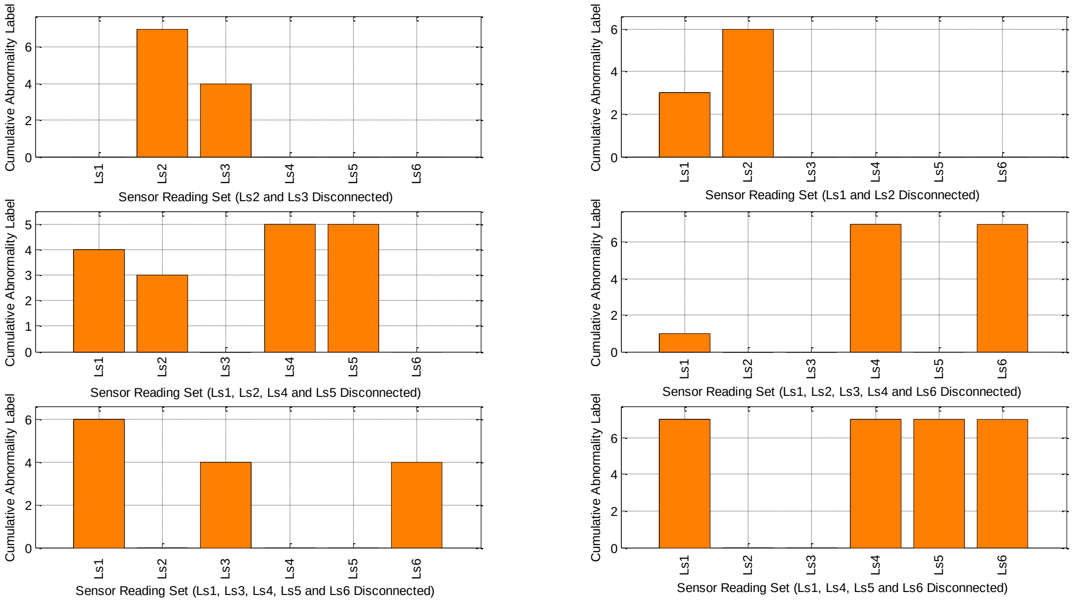


Figure 4.28. Cumulative Abnormality Labels from Experiments with Different Combinations of Disconnected Light Sensor.

Here, it can be seen that after calculating the corresponding CALs, the system had no difficulty detecting adjacent pairs of disconnected light sensors. In general, this method was capable of detecting up to 4 disconnected light sensors. When 5 light sensors were disconnected, the system was still able to identify that an anomalous situation was affecting these sensors. However, the calculation of the corresponding CAL only identifies 3 out of the 5 disconnected light sensors. This is an extreme case where the robot is unable to identify all of the disconnected sensors even after executing the light sensor research action described in subsection 4.3.2. This is because there are more sensors giving incorrect information than sensors working correctly. Another extreme case occurs when all of the robot's light sensors are disconnected. In this situation the robot tries to clean all of its light sensors. After realising that this does not improve the situation the robot assumes that there is no light source and goes to its home position. Here the robot indefinitely waits for a light source to appear.

4.5.1.3 Broken Legs

These experiments were designed to evaluate the performance of the self-diagnosis methods when all of the robot’s sensors and moving parts were fully functional but a part of the robot’s body had been damaged. A broken leg was simulated in these experiments by unscrewing the lower part of the leg. The result of this operation is shown in Fig. 4.29.

A first set of experiments was conducted by simulating a single broken leg. The obtained cumulative abnormality labels are shown in Fig. 4.30. In these graphs it can be observed that the effects of the broken leg are quickly propagated and measured by sensors located in other parts of the robot’s body. In this case, the calculation of the respective CALs is not enough for the identification of the broken leg. Because legs that are adjacent to the broken leg show larger CAL values, the robot’s body map can be used to identify in which area the broken leg should be looked for first. However, the system is only able to identify the anomalous leg after this executes leg tip force sensor research actions.

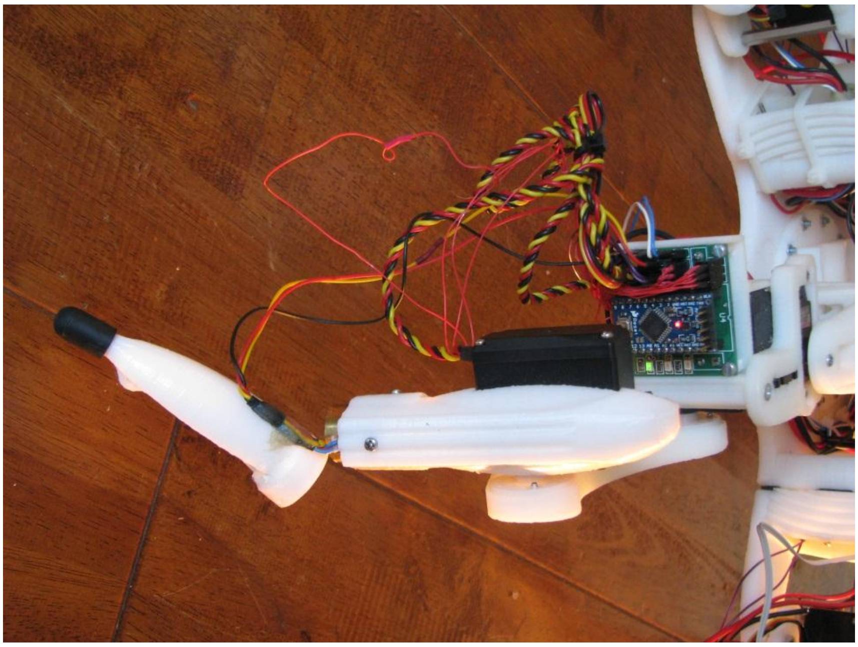


Figure 4.29. Simulated Broken Leg.

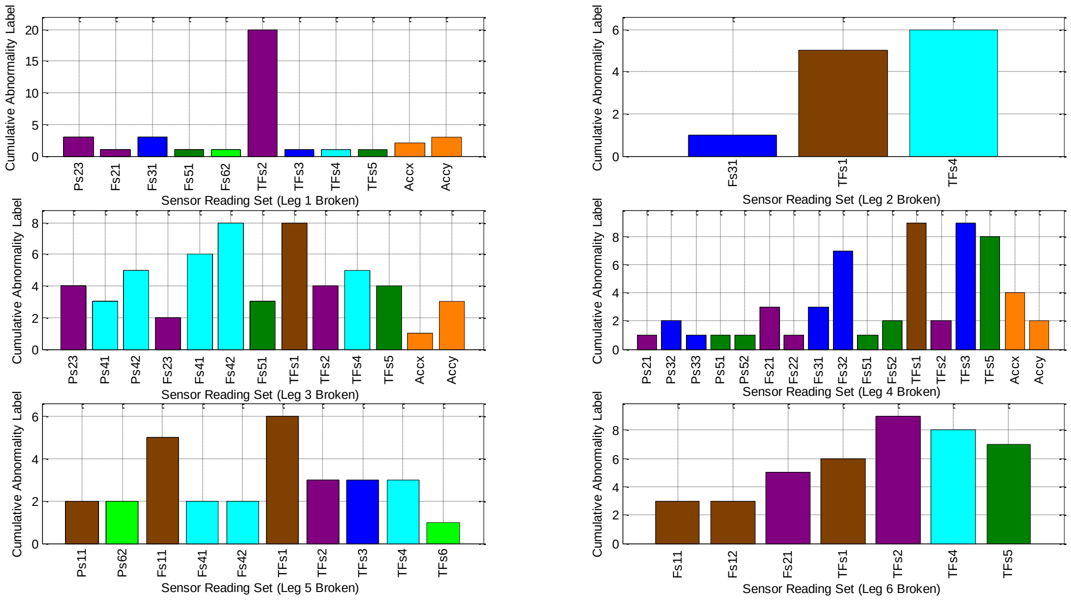


Figure 4.30. Cumulative Abnormality Labels from Experiments with a Single Broken Leg.

As could be expected, when more than one leg is broken simultaneously, the calculation of the CALs is not enough for the identification of broken legs, either. Results of experiments with pairs and trios of broken legs are shown in Figs. 4.31 and 4.32, respectively. The only case when the anomalous legs were detected exclusively by the use of CALs was when legs 1 and 4 were broken (Fig. 4.31). However, in most of the cases, the effects of the broken legs are propagated to other sensors. In contrast to experiments with only one broken leg, here the robot is not able to maintain a stable gate and the broken legs support a part of the robot's weight. As a result, the values of the leg tip force sensor readings of the broken legs are incremented and registered in the respective CAL. In this case, the leg tip force sensors measure the force applied in the middle of the leg instead of the force applied in the leg's tip as the sensor's name indicates. Similarly to the experiment with one broken leg, the methods discussed in the chapter can only indicate the area of the robot most affected by the abnormal situation by means of a body map. The exact legs which present the anomaly are only detected once the leg tip force sensor research actions are executed.

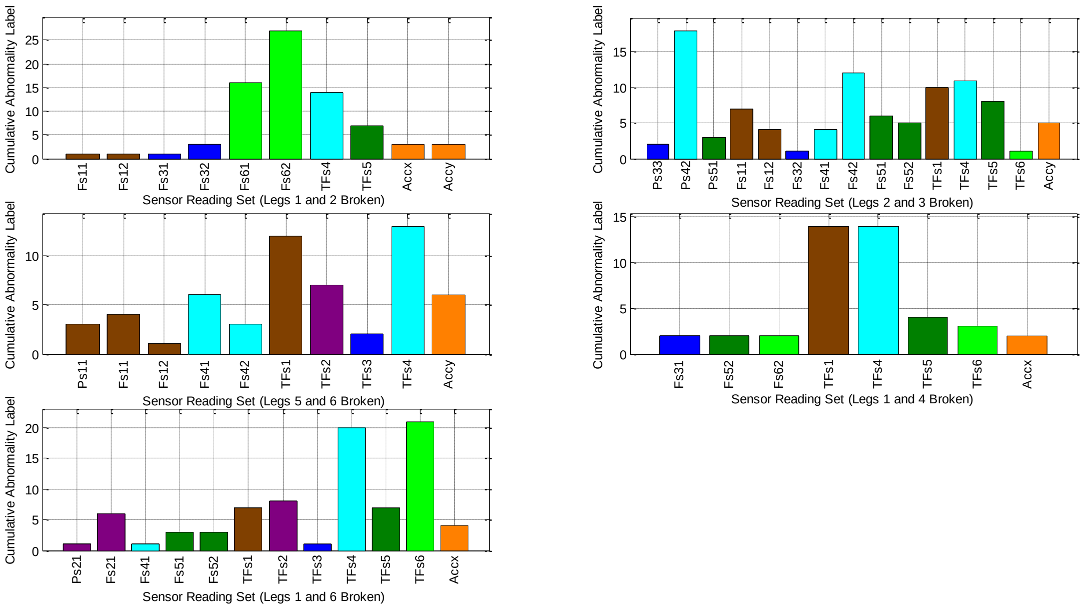


Figure 4.31. Cumulative Abnormality Labels from Experiments with Pairs of Broken Legs.

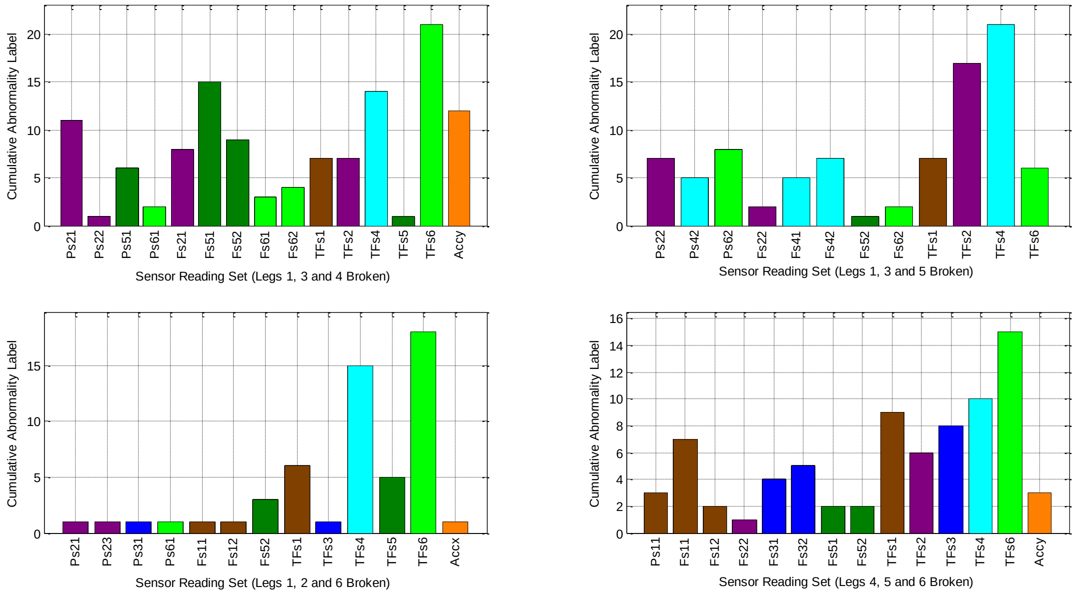


Figure 4.32. Cumulative Abnormality Labels from Experiments with Trios of Broken Legs.

Additional experiments with broken legs were conducted, this time the leg tip force sensors of the broken legs were disconnected. The results are similar to the ones shown on Figs. 4.30 to 4.32, but in this case the CAL value associated with the leg tip force sensor of the broken leg is 0. As a consequence, the robot was unable to find the anomalous leg with the research actions considered in this thesis.

4.5.1.4 Low Battery Levels.

In this experiment there are no faulty sensors and the robot's body is intact. However, the robot starts performing its mission with little charge in battery Bl_2. This battery provides the energy utilised by the servo motors and the leg release mechanism. Initially, the robot walks towards the light source without any apparent problem. However, when the energy level is too low, the lack of energy prevents the robot from making progress in its mission. This abnormal situation is detected by the system and the CAL showed in Fig. 4.33 is generated. Here it can be seen that the effects of the considered abnormal situation are propagated and detected by other sensors. This is because when the energy levels are too low the robot's legs are unable to reach the commanded coordinates. As a result, before the robot falls over, the legs reach awkward positions which increment the torque required from the servos and the pressure applied in the legs' tip. This, in turn, increases even more the error in the commanded position of the servo motors. However, because the low level of Bl_2 arises before other effects in the robot behaviour, the CAL value associated with Bl_2 is the greatest in the group. As a result, the system attributes correctly the anomaly to the low level of Bl_2 on its first attempt.

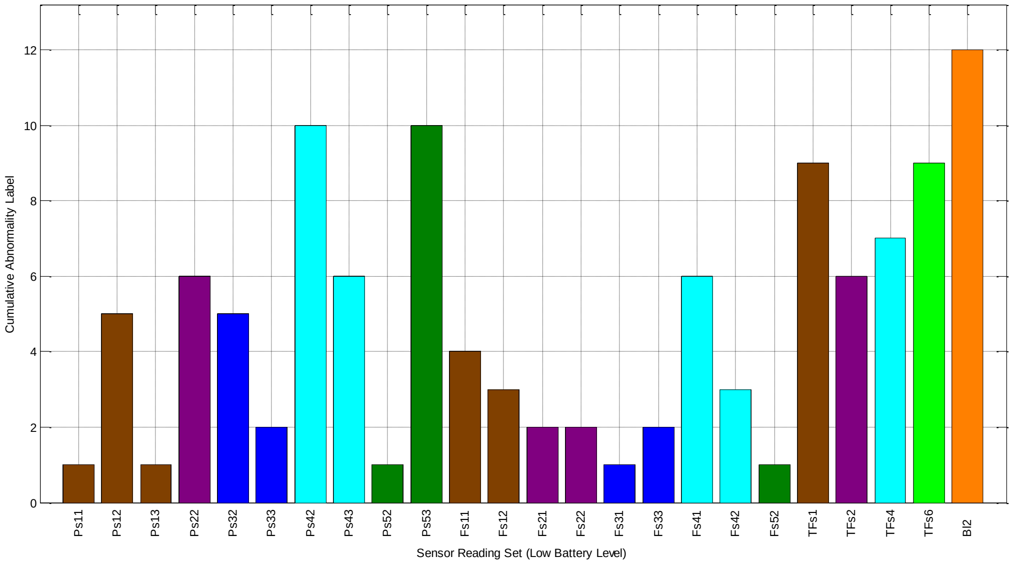


Figure 4.33. Cumulative Abnormality Labels from Experiments Low Battery Level.

Experiments with low levels of batteries Bl_1 and Bl_3 were conducted. These batteries power the microcontrollers and electronics located on the legs and the robot's core, respectively. As expected, when the level of any of these batteries is too low, the corresponding microcontroller stops working properly and the control of the robot is lost. This occurs even before the robot is able to detect a lack of progress. However, this situation is unlikely to happen because more power is required from the servo motors and normally Bl_2 discharges first. Nevertheless, a method which would allow the system to detect low battery levels of Bl_1 and Bl_3 is to treat low battery levels as pain. So, when battery levels go below a determined threshold, compensatory actions could be executed immediately.

4.5.1.5 Software Bug.

Abnormal situations generated by software bugs can be more difficult to detect. In this experiment a software bug with detrimental effects on the robot's performance was introduced into the robot's code. The considered software bug disabled the turning and the rotation on the spot movements, which prevented the robot from reaching the light source. During the experiment, the robot detected a lack of progress but in this case all of the CAL values were 0. Therefore, the robot was unable to identify the source of the abnormality even after all the repertoire of research actions was executed. In this case, an alternative for the robot is to execute the methods presented in sections 5.4 and 5.5.

4.5.2. Experiments with Anomalies in the Robot's Environment

4.5.2.1 Robot Walking Uphill

The experimental robot has been provided with information regarding expected sensor readings when walking uphill. The slope of the terrain changes the weight distribution of the robot among its legs. As a result, CAL values corresponding to leg tip force sensor readings of hind legs are greater than those of front legs. Simultaneously, CAL values of Acc_y are slightly less than the number of steps used for identifying a cyclic lack of progress (7 in these experiments). The mean value of the ASCs obtained during experiments with the robot walking uphill has been utilised for desribing the abnormal situation. As a result, the abnormal situation centroid shown in Table 4.9 has been obtained.

Table 4.9. Abnormal Situation Centroid for Robot Walking Uphill.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0.06	0	0	0.07	0	0	0.07	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	0	0	0	0.10	0.06	0.06	0	0.01	0	0.23	0	0	0.01
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0.01	0	0	0	0	0	0	0	0.04	0	0	0	0	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
1.43	1.03	1.16	1.16	2.07	2.13	0	0.81	0	0	0	0		

Experiments have been conducted in order to evaluate the performance of the self-diagnosis methods when the robot is walking over a steep terrain. In this case, the slope of the terrain is so steep $(\sim 40°)$ that it prevents the robot from making further progress in its mission. Figure 4.34 shows one of the CAL sets obtained during these experiments. In order to verify that the abnormal CAL values are not generated by faults in sensors or other of the robot's components, the corresponding research actions must be executed. In this case, the executed research actions are those corresponding to position of servo motors, leg tip force sensors and accelerometer. Once the robot has verified that the abnormal CAL values were not generated due to

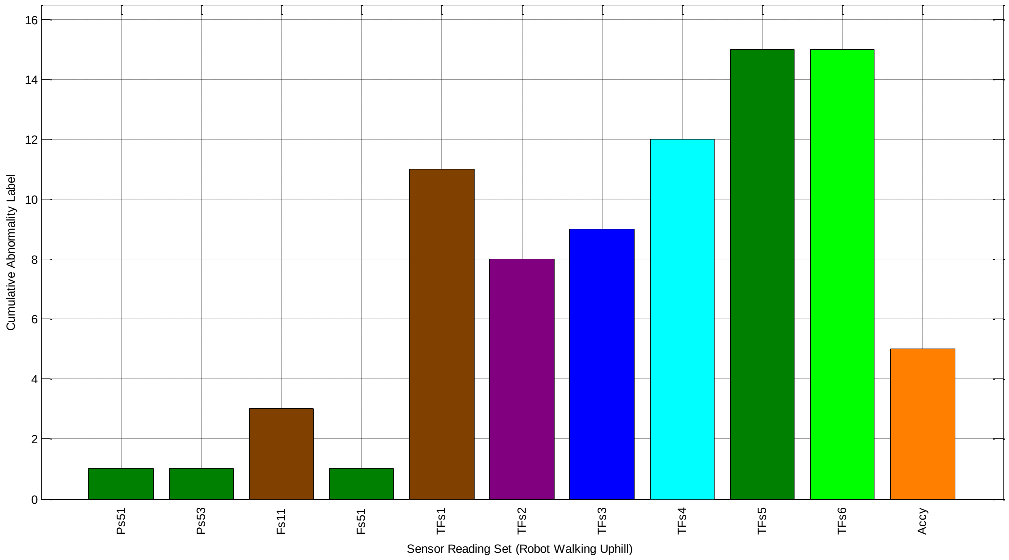


Figure 4.34. Cumulative Abnormality Labels from Experiment with Robot Walking Uphill.

the robot's malfunction or damage; the corresponding ASC is calculated by means of expression 4.17. This ASC is compared with the one previously incorporated into the robot for describing this abnormal situation. Then, the Euclidean distance between both ASCs was calculated as a discrepancy measure. Table 4.10 contains the discrepancy obtained from 10 experiments where the robot walks uphill. For comparison purposes, discrepancy values with ASC corresponding to other abnormal situations are also shown in the table.

Table 4.10. Discrepancy Values from Experiments with Robot Walking Uphill.

Experiment	1	2	3	4	5	6	7	8	9	10
Robot Walking Uphill	**0.40**	**1.04**	**1.60**	**0.34**	**0.63**	**0.43**	**0.45**	**0.57**	**0.39**	**0.44**
Slippery Surface	2.19	2.98	3.25	2.30	2.19	2.29	2.06	2.19	2.30	2.13
Air Walking	4.28	3.62	3.21	4.25	4.36	4.23	4.36	4.34	4.16	4.28
Collision with Small Obstacles	1.23	1.42	1.78	1.35	1.39	1.45	1.28	1.45	1.13	1.18
Leg 1 on Soft Terrain	4.18	3.51	3.34	4.28	4.45	4.25	4.30	4.41	4.10	4.19
Leg 2 on Soft Terrain	4.18	3.53	3.04	4.19	4.32	4.15	4.25	4.27	4.11	4.19
Leg 3 on Soft Terrain	3.85	3.37	2.85	3.80	3.91	3.80	3.90	3.97	3.75	3.90
Leg 4 on Soft Terrain	3.67	3.24	2.84	3.63	3.84	3.58	3.69	3.76	3.60	3.73
Leg 5 on Soft Terrain	3.60	3.27	2.95	3.70	3.72	3.72	3.72	3.66	3.64	3.68
Leg 6 on Soft Terrain	3.89	3.65	3.28	3.96	3.94	3.93	3.96	3.87	3.88	3.89
Legs 1 and 2 on Soft Terrain	4.19	3.88	3.37	4.32	4.36	4.30	4.30	4.27	4.17	4.20
Legs 2 and 3 on Soft Terrain	3.79	3.68	2.82	3.78	3.84	3.76	3.82	3.77	3.73	3.86
Legs 5 and 6 on Soft Terrain	3.66	3.43	3.36	3.78	3.79	3.80	3.76	3.77	3.79	3.76
Legs 1 and 4 on Soft Terrain	3.93	3.70	3.42	4.06	4.10	4.02	4.03	4.00	3.92	3.94
Legs 1 and 6 on Soft Terrain	4.04	3.40	3.30	4.09	4.22	4.06	4.11	4.19	4.00	4.04
Legs 1, 3 and 4 on Soft Terrain	4.42	3.89	3.58	4.48	4.61	4.46	4.50	4.60	4.36	4.44
Legs 1, 3 and 5 on Soft Terrain	4.26	3.61	3.53	4.37	4.50	4.39	4.38	4.52	4.31	4.33
Legs 1, 2 and 6 on Soft Terrain	4.22	3.61	3.35	4.29	4.44	4.25	4.30	4.38	4.17	4.22
Legs 4, 5 and 6 on Soft Terrain	4.30	3.80	3.91	4.35	4.41	4.39	4.39	4.49	4.38	4.38

This data shows an average discrepancy value of 0.42, which was small enough to allow the correct identification of the abnormal situation during the experiments.

4.5.2.2 Slippery Surface

When walking over a slippery surface the experimental robot's legs tend to spread out. As a result, more torque is required from the servo motors and servo position errors arise. Eventually, legs get to awkward positions, which increases readings in leg tip force sensors. The effects may escalate until the robot falls over and is unable to stand up and continue with its mission. A number of experiments have been conducted in order to evaluate the proposed self-diagnosis methods when dealing with this situation. In this case, a slippery surface has been simulated by removing the robot's rubber feet while walking on a wood surface. This is shown in Fig. 4.35. As before, the robot has been provided with an ASC for that situation obtained empirically. In this case, the ASC shown in Table 4.11 has been calculated.

Table 4.11. Abnormal Situation Centroid for Robot Walking Over a Slippery Surface.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0.03	0.63	0	0	0.04	0	0	0.03
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	0	0.6	0	0	0.79	0	0	0.84	0	0	0.5	0	0
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0	0	0	0	0	0.04	0.49	0	0	0.03	0	0	0	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
2.33	1.30	1.46	2.56	2.33	2.49	0	0	0	0	0	0		

In general, when the robot walks on slippery surfaces high CAL values of leg tip force sensors are expected. In addition, noticeable CAL values corresponding to the position sensor of servo 2 are expected to be present. This can be seen in Fig. 4.36, where a CAL set obtained during experiments with a slippery surface is displayed.

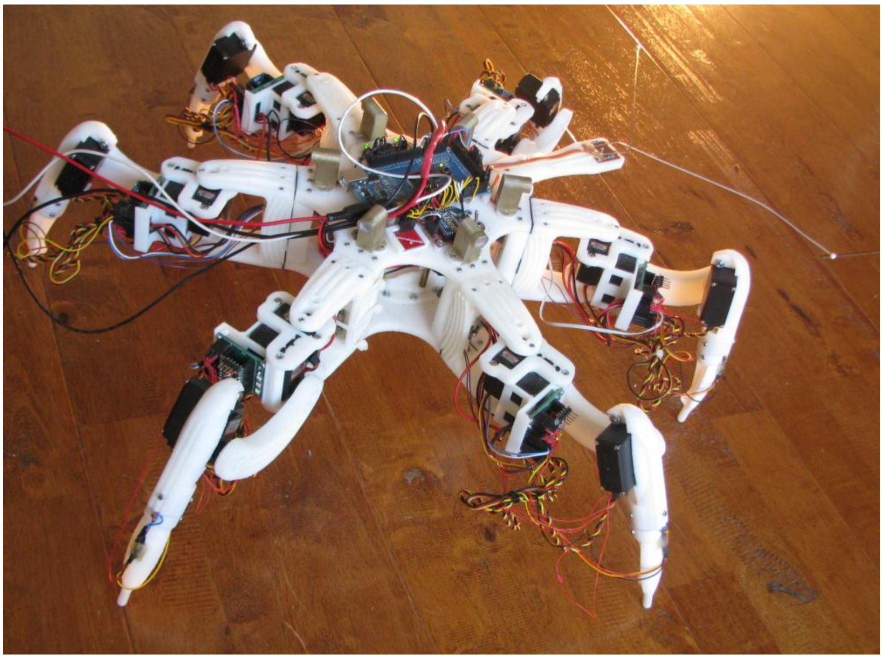


Figure 4.35. Robot Walking Over a Slippery Surface (Rubber Feet Removed).

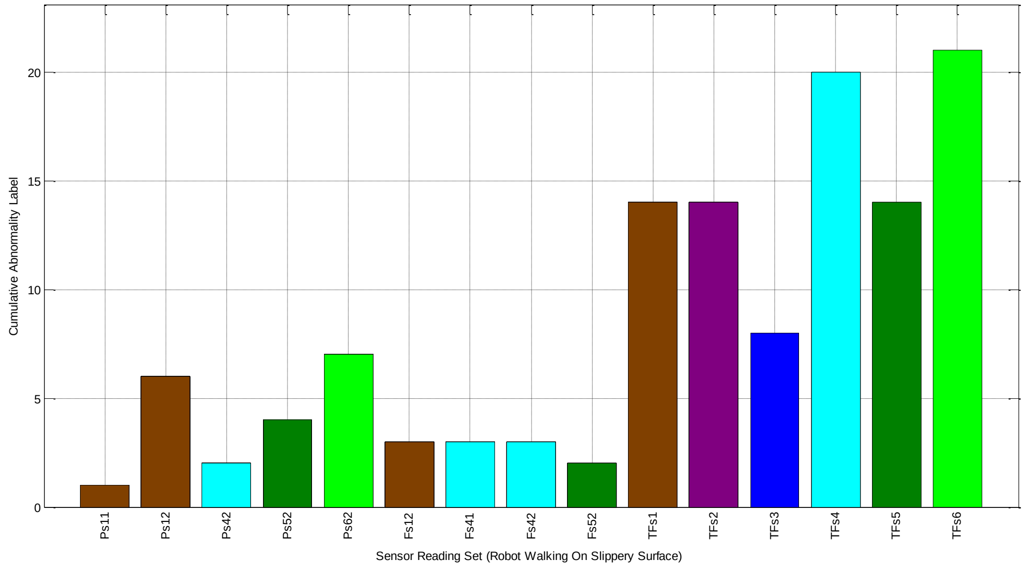


Figure 4.36. Cumulative Abnormality Labels from Experiment with Robot Walking Over a Slippery Surface.

Once again the robot must verify that the abnormal CAL values are not generated by faults in sensors or other of the robot's components. Therefore, the corresponding research actions must be executed.

The discrepancy measure for 10 experiments where the robot walks over a slippery surface is shown in Table 4.12.

Table 4.12. Discrepancy Values from Experiments with Robot Walking Over a Slippery Surface.

Experiment	1	2	3	4	5	6	7	8	9	10
Robot Walking Uphill	2.67	2.26	2.31	2.49	2.58	2.43	2.14	2.61	2.78	2.55
Slippery Surface	**1.27**	**0.81**	**0.81**	**1.06**	**0.98**	**0.88**	**0.87**	**1.08**	**0.88**	**0.64**
Air Walking	5.77	5.37	5.49	5.05	5.71	5.33	5.10	5.74	6.10	5.75
Collision with Small Obstacles	2.77	2.50	2.42	2.63	2.76	2.68	2.37	2.80	3.04	2.74
Leg 1 on Soft Terrain	5.65	5.30	5.44	4.97	5.61	5.30	5.04	5.72	6.02	5.68
Leg 2 on Soft Terrain	5.54	5.17	5.31	4.82	5.48	5.18	4.90	5.60	5.90	5.57
Leg 3 on Soft Terrain	5.36	4.95	5.12	4.73	5.42	4.93	4.63	5.35	5.55	5.32
Leg 4 on Soft Terrain	5.08	4.60	4.81	4.34	5.10	4.76	4.31	5.25	5.30	5.10
Leg 5 on Soft Terrain	5.00	4.58	4.65	4.31	4.97	4.55	4.33	4.91	5.19	4.82
Leg 6 on Soft Terrain	5.15	4.73	4.83	4.41	5.11	4.69	4.47	5.07	5.34	4.98
Legs 1 and 2 on Soft Terrain	5.50	5.05	5.16	4.78	5.42	5.02	4.84	5.41	5.66	5.29
Legs 2 and 3 on Soft Terrain	5.26	4.58	4.73	4.44	5.18	4.68	4.33	5.13	5.14	4.92
Legs 5 and 6 on Soft Terrain	4.80	4.49	4.66	4.18	4.84	4.42	4.23	4.78	5.03	4.73
Legs 1 and 4 on Soft Terrain	5.14	4.71	4.85	4.44	5.10	4.73	4.51	5.14	5.35	4.98
Legs 1 and 6 on Soft Terrain	5.27	5.03	5.19	4.67	5.32	5.02	4.75	5.41	5.71	5.38
Legs 1, 3 and 4 on Soft Terrain	5.63	5.42	5.56	5.15	5.74	5.42	5.16	5.78	6.03	5.73
Legs 1, 3 and 5 on Soft Terrain	5.46	5.33	5.49	5.00	5.59	5.27	5.03	5.61	5.91	5.62
Legs 1, 2 and 6 on Soft Terrain	5.54	5.20	5.36	4.86	5.50	5.23	4.94	5.63	5.90	5.58
Legs 4, 5 and 6 on Soft Terrain	5.26	5.29	5.46	4.98	5.53	5.16	4.98	5.46	5.82	5.55

In this case, the data shows an average discrepancy value of 0.93 and the abnormal situation was identified in all of the considered experiments.

4.5.2.3 Air Walking

While walking over unstructured terrain, the robot could eventually get to a position where all of its legs lose contact with the ground and the only point of contact between the robot and the irregular terrain is the robot's underside. Then, the robot could perform its normal gait without moving forward. These experiments aim to evaluate the performance of the self-diagnosis methods when the robot is facing this kind of situation. During experiments dealing with air walking the robot has been located on a plastic container with enough height to prevent the robot's legs from establishing contact with the ground while walking. This is shown in Fig. 4.37. As the robot is unable to make any progress in its mission, an abnormal situation is detected. However, in this case nothing abnormal is detected by the robot's sensors (all of the CAL values are 0). This situation could be described by the following abnormal situation centroid.

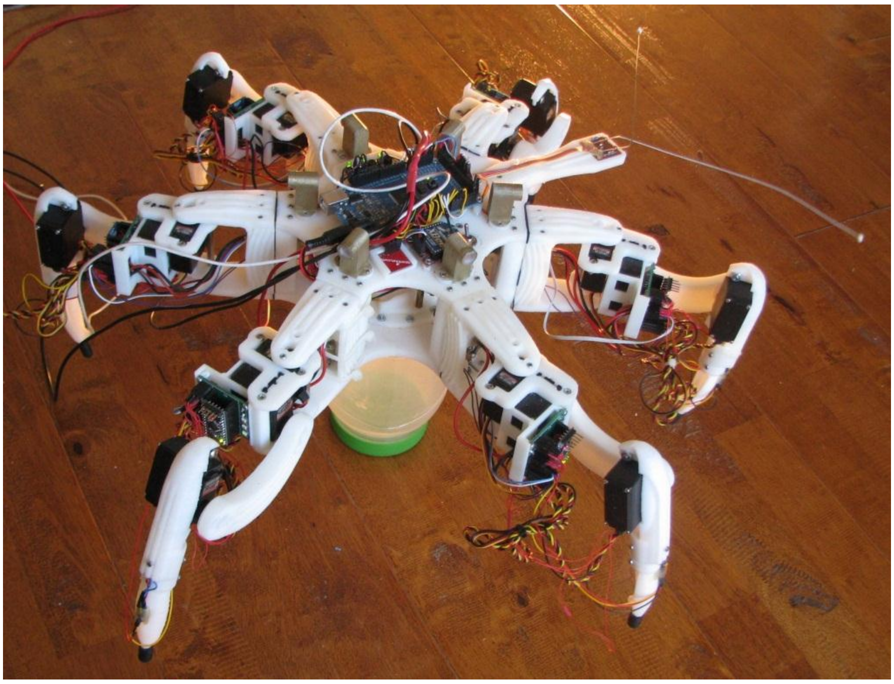


Figure 4.37. Robot Air Walking.

$$ASC = \{0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \ldots$$
$$\ldots 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\}$$

Because the resulting CAL values are always the same for this experiment, the discrepancy measure is always 0. However, there are different abnormal situations that can present exactly the same CAL values (e.g. partially covered light sensors). In these cases, the robot must execute compensatory measures in turn for each of the situations with the same ASC until one is able to compensate the abnormal situation.

4.5.2.4 Partially Covered Light Sensors

During this experiment, fully functional light sensors were partially covered. In general, the robot found no difficulty in performing its mission with partially covered light sensors. However, the calculation of the direction of the light source was strongly affected when one of the two light sensors facing the light source was partially covered. Then, the robot was unable to progressively approach the light source and, after 7 steps without making any progress in its mission, a cyclic lack of progress alarm was triggered. Similarly to the experiment in subsection 4.5.2.3, all of the CAL values registered by the robot with partially covered light sensors were 0. As a result, the system was unable to identify the anomaly with this method. However, partially covered light sensors can be identified by the incorporated light sensor research actions if the number of uncovered and functional light sensors is greater than the number of partially covered light sensors. Therefore, a non-optimal solution for the correct identification of the anomaly is to execute all of the repertoire of research actions until the abnormal situation is identified.

4.5.2.5 Collision with Small Obstacles.

Small obstacles, as the ones shown in Fig. 4.38, could pass under the robot's whiskers and collide with the robot's legs, preventing the robot from making further progress in its mission. Exposing the robot to this situation will allow additional evaluation of the performance of the self-diagnosis methods.

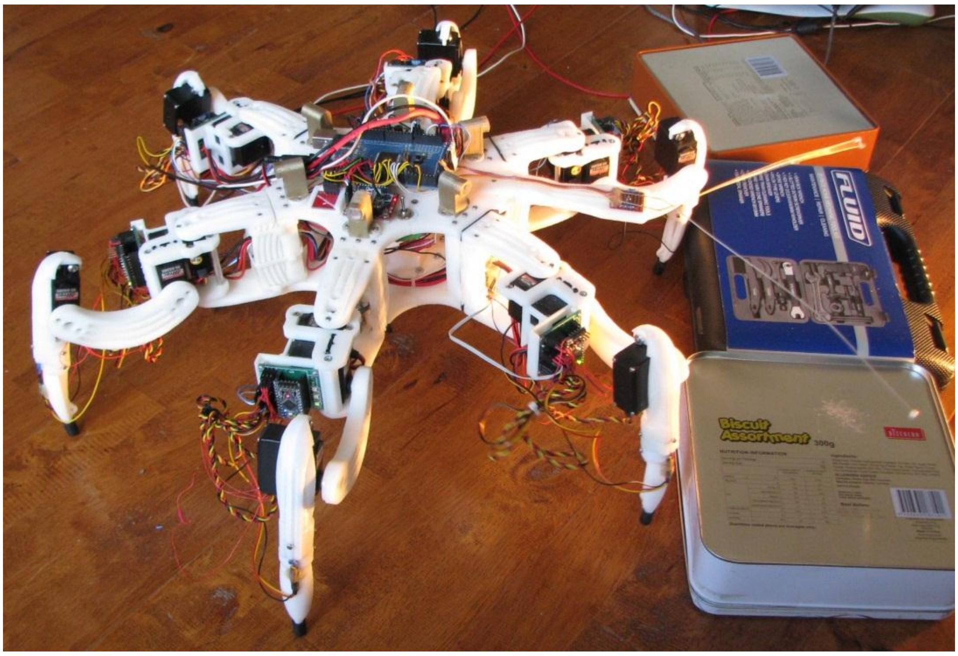


Figure 4.38. Robot Colliding with Small Obstacles.

Empirical data has been utilised for describing this situation. Here, high CAL values of leg tip force sensors have been obtained. In addition, small CAL values corresponding to the force sensor of servo 1 and accelerometer y-coordinates are present. An example of the CAL values obtained during experiments with small obstacles is shown in Fig. 4.39.

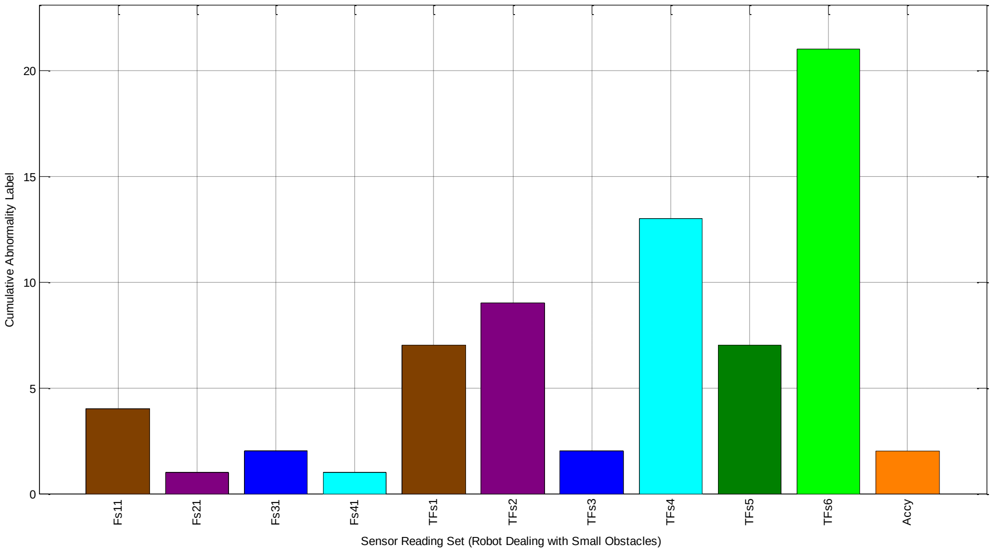


Figure 4.39. Cumulative Abnormality Labels from Experiment with Robot Colliding with Small Obstacles.

Considering the obtained data, the ASC shown in Table 4.13 has been incorporated into the robot for identifying this kind of abnormal situation.

Table 4.13. Abnormal Situation Centroid for Robot Colliding with Small Obstacles.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	0	0	0	0	0	0	0	0	0	0.43	0	0	0.5
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0	0	0.41	0	0	0.39	0	0	0	0	0	0	0	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
1.43	0.71	0.94	1.39	1.90	2.14	0	0.09	0	0	0	0		

As usual, the corresponding research actions must be executed in order to discard CAL generated due to the robot’s malfunction or damage.

The discrepancy measures obtained from 10 experiments where the robot deals with small obstacles are shown in Table 4.14.

Table 4.14. Discrepancy Values from Experiments with Robot Colliding with Small Obstacles.

Experiment	1	2	3	4	5	6	7	8	9	10
Robot Walking Uphill	1.88	1.68	1.36	1.54	1.43	1.73	1.81	1.39	1.89	1.17
Slippery Surface	2.95	2.88	2.28	2.63	2.52	2.82	3.19	2.44	3.13	2.40
Air Walking	4.09	3.85	4.59	4.18	4.14	3.26	2.89	4.11	3.75	4.22
Collision with Small Obstacles	**1.48**	**0.88**	**0.93**	**0.77**	**0.99**	**1.32**	**1.47**	**0.73**	**1.16**	**1.02**
Leg 1 on Soft Terrain	3.94	3.81	4.43	4.01	4.15	3.23	2.94	3.94	3.74	4.09
Leg 2 on Soft Terrain	3.95	3.65	4.37	3.97	3.92	3.11	2.73	3.87	3.65	4.13
Leg 3 on Soft Terrain	3.91	3.58	3.96	3.68	3.57	2.90	2.56	3.62	3.65	3.70
Leg 4 on Soft Terrain	3.83	3.37	3.79	3.54	3.30	2.71	2.37	3.48	3.47	3.49
Leg 5 on Soft Terrain	3.70	3.43	3.96	3.75	3.61	2.64	2.69	3.39	3.37	3.53
Leg 6 on Soft Terrain	3.89	3.68	4.19	3.98	3.84	2.76	2.87	3.59	3.64	3.78
Legs 1 and 2 on Soft Terrain	4.21	3.74	4.30	4.14	4.12	3.14	3.06	3.68	3.85	3.97
Legs 2 and 3 on Soft Terrain	4.16	3.35	3.69	3.57	3.45	2.84	2.57	3.17	3.66	3.30
Legs 5 and 6 on Soft Terrain	3.83	3.78	4.06	3.97	3.74	2.73	2.96	3.60	3.78	3.81
Legs 1 and 4 on Soft Terrain	4.02	3.65	4.12	3.99	3.91	2.84	2.96	3.53	3.72	3.73
Legs 1 and 6 on Soft Terrain	3.78	3.77	4.30	3.94	3.92	3.04	2.88	3.86	3.70	4.07
Legs 1, 3 and 4 on Soft Terrain	4.22	4.03	4.46	4.17	4.19	3.41	3.33	4.07	4.02	4.27
Legs 1, 3 and 5 on Soft Terrain	4.02	4.13	4.50	4.16	4.21	3.33	3.33	4.09	4.06	4.35
Legs 1, 2 and 6 on Soft Terrain	4.00	3.79	4.38	4.03	4.05	3.26	2.95	3.91	3.81	4.16
Legs 4, 5 and 6 on Soft Terrain	4.17	4.32	4.60	4.49	4.13	3.34	3.49	4.38	4.20	4.60

In this case, the data shows an average discrepancy value of 1.08 and the abnormal situation was identified in all of the considered experiments.

4.5.2.6 Robot Walking over Soft Terrain.

This group of experiments considers situations that are potentially challenging for the system to diagnose. When the experimental robot walks over soft terrain, its legs sink into the terrain and the robot is unable to move forward. As a result, more torque is demanded from both the servo motors of legs walking over hard terrain and those walking over soft terrain. The propagation of the effects of the anomaly over components of the robot not directly connected with the abnormal situation also extends to the robot's leg tip force sensors. Because of this propagation it is more difficult for the system to diagnose which parts of the robot are directly connected with the abnormal situation.

A series of experiments was conducted in order to test the capability of the self-diagnosis methods presented in this chapter when one of the six robot legs is walking on soft terrain. As in the experiment presented in subsection 0, the soft terrain was simulated by using a sponge. In general, high CAL values for the position and force sensors of servo 1, corresponding to the leg walking on soft terrain, were obtained. In addition, different leg tip force sensors presented high CAL values depending on the leg walking on soft terrain. An example of the CAL values registered by the robot during the experiments is shown in Fig. 4.40.

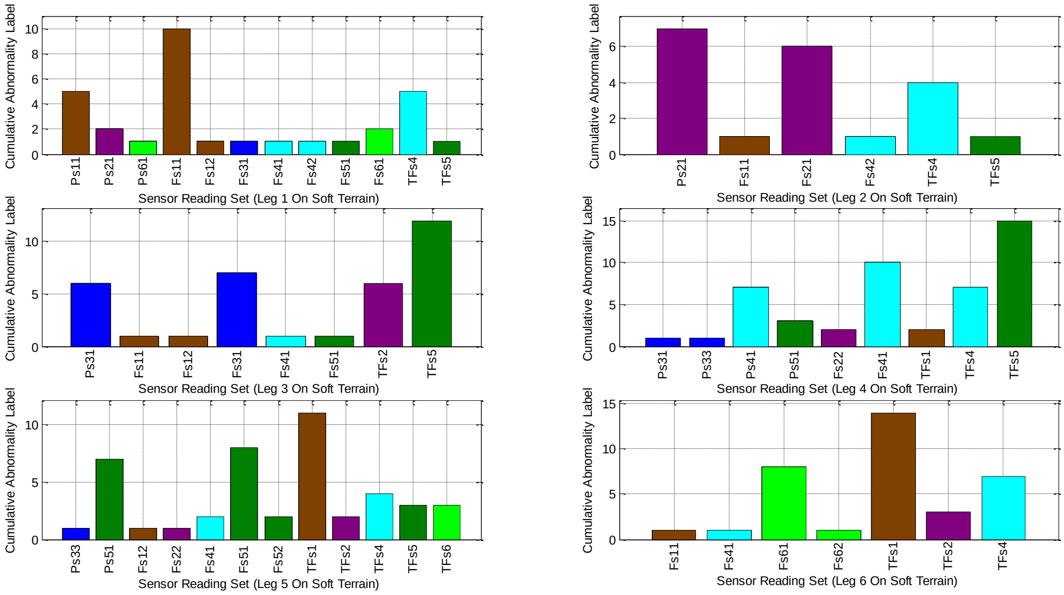


Figure 4.40. Cumulative Abnormality Labels from Experiments with One Leg Walking Over Soft Terrain.

Here it can be seen that when legs 1 or 2 walked over soft terrain, they were identified by the experimental robot on its first attempt. This was possible due to high values of CALs corresponding to abnormal position and force sensor readings of the leg walking over soft terrain. In the remaining cases, there is a high CAL associated with leg tip force sensors TFs_1 or TFs_5. Therefore, the robot had to execute leg tip force sensor research actions first in order to check that the anomaly was not associated with these sensors. Then, the robot was able to identify that the leg was walking over soft terrain.

In order to provide the robot with a representation of each situation, the ASCs shown in Tables 4.15 to 4.20 were incorporated into the robot's database of abnormal situations. These ASCs were obtained from empirical data collected by the robot while walking on soft terrain.

Table 4.15. Abnormal Situation Centroid for Leg 1 Walking on Soft Terrain.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0.79	0	0	0.16	0	0	0	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	0	0	0	0	0	0	0.11	0	0	1.54	0.03	0	0.07
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0.01	0	0.17	0.01	0	0.14	0.06	0	0.16	0.06	0	0.17	0.01	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
0.03	0.03	0.1	0.59	0.09	0.04	0	0	0	0	0	0		

Table 4.16. Abnormal Situation Centroid for Leg 2 Walking on Soft Terrain.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0.16	0	0	1.01	0	0	0	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	0	0	0	0	0	0	0.2	0	0	0.11	0.01	0	1.04
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0	0	0.06	0.04	0	0.11	0.07	0	0.16	0.01	0	0.13	0.06	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
0.04	0.03	0.07	0.69	0.09	0.07	0	0	0	0	0	0		

Table 4.17. Abnormal Situation Centroid for Leg 3 Walking on Soft Terrain.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0	0	0	0	0	0	1.09	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	0	0	0	0	0	0	0	0	0	0.19	0.01	0	0.21
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0.04	0	1.07	0.01	0	0.09	0.06	0	0.11	0.04	0	0.09	0.06	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
0.07	0.73	0.09	0.06	1.24	0.04	0	0	0	0	0	0		

Table 4.18. Abnormal Situation Centroid for Leg 4 Walking on Soft Terrain..

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0	0	0	0	0	0	0.01	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0.01	1.04	0	0	0.04	0	0	0	0	0	0.09	0.03	0	0.13
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0.09	0	0.17	0.03	0	1.11	0.03	0	0.16	0.04	0	0.13	0.03	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
0.1	0.01	0.07	0.83	1.44	0.09	0	0	0	0	0	0		

Table 4.19. Abnormal Situation Centroid for Leg 5 Walking on Soft Terrain.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0.07	0	0	0	0.96	0	0	0	0	0	0.14	0.03	0	0.07
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0.05	0	0.07	0.01	0	0.17	0.04	0	1.1	0.17	0	0.06	0.01	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
1.69	0.33	0.06	0.36	0.27	0.36	0	0	0	0	0	0		

Table 4.20. Abnormal Situation Centroid for Leg 6 Walking on Soft Terrain.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	0	0	0	0	0	0	0.96	0	0	0.13	0.03	0	0.14
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0.06	0	0.14	0.02	0	0.11	0.04	0	0.04	0.06	0	1.2	0.06	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
1.74	0.39	0.04	0.56	0.06	0.09	0	0	0	0	0	0		

These ASCs were compared with results obtained from 5 experiments conducted for each case and the discrepancy measures shown in Table 4.21 were obtained.

Table 4.21. Discrepancy Values from Experiments with One Leg Walking Over Soft Terrain.

Experiment	1	2	3	4	5
Discrepancy (Leg 1 on Soft Terrain)	0.34	0.41	0.48	0.42	0.42
Discrepancy (Leg 2 on Soft Terrain)	0.44	0.55	0.52	0.59	0.51
Discrepancy (Leg 3 on Soft Terrain)	0.60	0.43	0.51	0.86	0.54
Discrepancy (Leg 4 on Soft Terrain)	0.99	0.68	0.64	0.82	0.89
Discrepancy (Leg 5 on Soft Terrain)	0.43	0.63	0.63	0.63	0.62
Discrepancy (Leg 6 on Soft Terrain)	1.12	0.77	0.89	0.84	0.66

A comparison between the maximum discrepancy values for each row in Table 4.21 and the discrepancy values corresponding to other type of experiments can be done by considering the data shown in Table 4.22.

Table 4.22. Discrepancy Values Between Obtained ASCs in Experiments with One Leg Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	Discrepancy (Leg 1 on Soft Terrain)	Discrepancy (Leg 2 on Soft Terrain)	Discrepancy (Leg 3 on Soft Terrain)	Discrepancy (Leg 4 on Soft Terrain)	Discrepancy (Leg 5 on Soft Terrain)	Discrepancy (Leg 6 on Soft Terrain)
Robot Walking Uphill	4.30	4.47	4.08	3.69	3.82	3.74
Slippery Surface	5.35	5.49	5.25	4.70	4.75	4.54
Air Walking	2.12	2.07	1.57	3.00	2.47	2.56
Collision with Small Obstacles	3.70	3.85	3.55	3.32	3.45	3.39
Leg 1 on Soft Terrain	**0.48**	2.34	2.13	3.25	2.71	2.81
Leg 2 on Soft Terrain	2.21	**0.59**	2.19	3.11	2.64	2.71
Leg 3 on Soft Terrain	2.75	2.83	**0.86**	2.74	2.87	3.13
Leg 4 on Soft Terrain	2.67	2.84	2.34	**0.99**	2.82	3.01
Leg 5 on Soft Terrain	2.74	2.99	2.45	3.21	**0.63**	2.02
Leg 6 on Soft Terrain	2.76	2.87	2.67	3.50	2.19	**1.12**
Legs 1 and 2 on Soft Terrain	2.46	2.34	3.06	3.90	2.68	2.56
Legs 2 and 3 on Soft Terrain	3.51	2.81	2.47	3.13	2.96	3.04
Legs 5 and 6 on Soft Terrain	3.16	3.39	3.12	3.61	1.63	2.08
Legs 1 and 4 on Soft Terrain	2.44	3.30	3.17	3.15	2.64	2.02
Legs 1 and 6 on Soft Terrain	1.69	2.34	2.48	3.33	2.82	2.46
Legs 1, 3 and 4 on Soft Terrain	2.43	3.35	2.49	3.06	3.44	3.24
Legs 1, 3 and 5 on Soft Terrain	2.33	3.26	2.27	3.72	2.58	3.30
Legs 1, 2 and 6 on Soft Terrain	1.86	1.68	2.74	3.50	2.99	2.63
Legs 4, 5 and 6 on Soft Terrain	3.30	3.56	3.31	3.24	2.70	3.18

Results show 100% of correct first attempt identification for the considered experiments. Even in the worst case, with the largest discrepancy between the target ASC and the obtained ASC, first attempt identification was achieved.

Figure 4.41 shows the CALs obtained when the robot performed its mission with 2 legs walking over soft terrain. In the considered experiments, high values of CALs associated with leg tip force sensors arose. Therefore, once again the robot had to execute leg tip force sensor research actions and discard the possibility that the abnormal situation was associated with these sensors. Only then the robot was able to identify that the legs were walking over soft terrain. In these cases, a body map has proved to be useful for the identification of these legs.

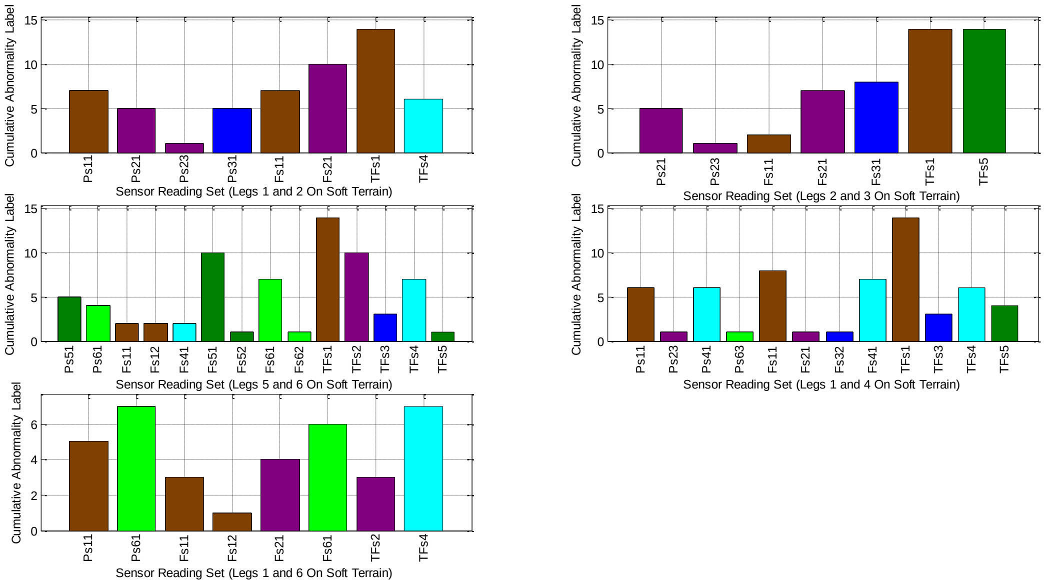


Figure 4.41. Cumulative Abnormality Labels from Experiments with Different Pairs of Legs Walking Over Soft Terrain.

The ASCs shown in Tables 4.23 to 4.27 were incorporated into the robot's database of abnormal situations in order to provide the robot with a representation of each anomaly.

Table 4.23. Abnormal Situation Centroid for Legs 1 and 2 Walking On Soft Terrain.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	1	0	0	0	0	0.07	0	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	0	0	0	0	0	0	1.01	0	0.1	0.71	0.14	0	0.31
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0.04	0	0.13	0.03	0	0.11	0.06	0	0	0.11	0	0.96	0.04	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
0.14	0.49	0.14	1.01	0.10	0.13	0	0	0	0	0	0		

Table 4.24. Abnormal Situation Centroid for Legs 2 and 3 Walking On Soft Terrain.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0	0	0	1.01	0	0.09	0.67	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0.04	0	0	0	0	0	0	0	0	0	0.11	0.03	0	1.19
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0.06	0	1.21	0.06	0	0.16	0.03	0	0.14	0.04	0	0.14	0.01	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
1.59	0.14	0.09	0.14	1.69	0.10	0	0	0	0	0	0		

Table 4.25. Abnormal Situation Centroid for Legs 5 and 6 Walking On Soft Terrain.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0	0	0	0	0	0.06	0	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0.06	0	0	0	1.16	0	0	1.01	0	0	0.11	0.21	0	0.17
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0.10	0	0.13	0.06	0	0.14	0.03	0	1.07	0.04	0	1.04	0.06	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
1.64	1.03	0.36	1.10	0.30	0.14	0	0	0	0	0	0		

Table 4.26. Abnormal Situation Centroid for Legs 1 and 4 Walking On Soft Terrain.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	1.14	0	0	0	0	0.06	0	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	1.01	0	0	0	0	0	0	0	0.04	1.00	0.16	0	0.19
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0.07	0	0.09	0.03	0	0.89	0.03	0	0	0.03	0	0.96	0.01	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
1.84	0.14	0.33	0.77	0.29	0.11	0	0	0	0	0	0		

Table 4.27. Abnormal Situation Centroid for Legs 1 and 6 Walking On Soft Terrain.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	1.00	0	0	0	0	0.07	0	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	0	0	0	0	0	0	1.01	0	0.10	0.71	0.14	0	0.31
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0.04	0	0.13	0.03	0	0.11	0.06	0	0	0.11	0	0.96	0.04	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
0.14	0.49	0.14	1.01	0.10	0.13	0	0	0	0	0	0		

Once again, these ASCs were compared with results obtained from the 5 experiments conducted for each case. Then, the discrepancy measures shown in Table 4.28 were obtained.

Table 4.28. Discrepancy Values from Experiments with Different Pairs of Legs Walking Over Soft Terrain.

Experiment	1	2	3	4	5
Discrepancy (Legs 1 and 2 on Soft Terrain)	0.88	0.65	0.71	0.76	0.67
Discrepancy (Legs 2 and 3 on Soft Terrain)	1.01	1.01	0.86	1.05	0.82
Discrepancy (Legs 5 and 6 on Soft Terrain)	1.00	0.70	0.77	0.67	0.67
Discrepancy (Legs 1 and 4 on Soft Terrain)	1.13	0.82	0.88	0.68	0.77
Discrepancy (Legs 1 and 6 on Soft Terrain)	0.61	0.74	0.60	0.73	0.67

Table 4.29 shows discrepancy values between the target ASC of the repertoire of conducted experiments and the ASC with the maximum discrepancy value from each row in Table 4.28.

Table 4.29. Discrepancy Values Between Obtained ASCs in Experiments with Different Pairs of Legs Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	Discrepancy (Legs 1 and 2 on Soft Terrain)	Discrepancy (Legs 2 and 3 on Soft Terrain)	Discrepancy (Legs 5 and 6 on Soft Terrain)	Discrepancy (Legs 1 and 4 on Soft Terrain)	Discrepancy (Legs 1 and 6 on Soft Terrain)
Robot Walking Uphill	4.29	3.73	4.12	3.75	4.03
Slippery Surface	5.07	4.68	4.93	4.58	5.08
Air Walking	3.13	3.30	3.17	3.02	1.79
Collision with Small Obstacles	3.73	3.25	3.79	3.29	3.55
Leg 1 on Soft Terrain	2.67	3.51	3.37	2.48	1.62
Leg 2 on Soft Terrain	2.50	3.07	3.13	3.05	1.92
Leg 3 on Soft Terrain	3.38	2.69	3.46	3.33	2.48
Leg 4 on Soft Terrain	3.56	3.05	3.52	2.56	2.55
Leg 5 on Soft Terrain	2.74	2.87	1.79	2.49	2.60
Leg 6 on Soft Terrain	2.68	3.00	1.80	2.54	1.99
Legs 1 and 2 on Soft Terrain	**0.88**	2.70	3.23	2.15	2.74
Legs 2 and 3 on Soft Terrain	2.66	**1.01**	3.48	3.07	3.38
Legs 5 and 6 on Soft Terrain	3.25	3.51	**0.77**	3.05	2.52
Legs 1 and 4 on Soft Terrain	2.37	3.25	2.96	**1.12**	2.47
Legs 1 and 6 on Soft Terrain	2.85	3.68	2.72	2.79	**0.73**
Legs 1, 3 and 4 on Soft Terrain	3.30	3.78	3.69	2.74	2.45
Legs 1, 3 and 5 on Soft Terrain	3.27	3.95	2.95	3.37	2.44
Legs 1, 2 and 6 on Soft Terrain	2.42	3.43	2.96	2.91	1.54
Legs 4, 5 and 6 on Soft Terrain	4.22	4.49	2.33	3.54	2.56

In this set of experiments, 100% of correct first attempt identification was achieved.

Another series of experiments considering 3 legs walking over soft terrain was conducted to test the capability of the self-diagnosis methods to identify this abnormal situation. Figure 4.42 shows the CAL values obtained during this set of experiments. As in the previous experiments with legs walking on soft terrain, the robot was able to identify these legs after executing leg tip force sensor research actions in order to discard high CAL values associated with these sensors.

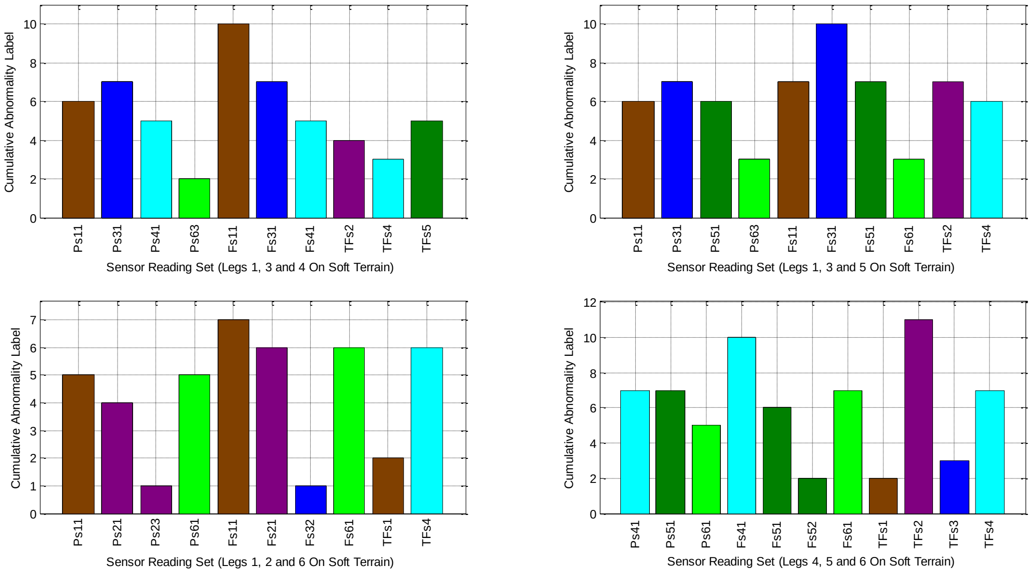


Figure 4.42. Cumulative Abnormality Labels from Experiments with Different Trios of Legs Walking Over Soft Terrain.

In this case, a representation of each abnormal situation was incorporated into the robot. This information was provided by means of the ASCs shown in Tables 4.30 to 4.33, which were obtained from empirical data collected by the robot while walking on soft terrain.

Table 4.30. Abnormal Situation Centroid for Legs 1, 3 and 4 Walking On Soft Terrain.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	1.16	0	0	0	0	0	1.13	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	1.09	0	0	0	0	0	0	0	0.19	1.12	0	0	0.14
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0.07	0	1.04	0	0	1.03	0	0	0	0.07	0	1.09	0	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
0.17	0.50	0.14	0.64	0.53	0.17	0	0	0	0	0	0		

Table 4.31. Abnormal Situation Centroid for Legs 1, 3 and 5 Walking On Soft Terrain.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0.94	0	0	0	0	0	1.10	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	0	0	0	0.93	0	0	0	0	0.34	0.97	0	0	0.17
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0.06	0	1.09	0	0	0.11	0	0	1.19	0.09	0	0.40	0	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
0.11	1.00	0.16	1.06	0.13	0.11	0	0	0	0	0	0		

Table 4.32. Abnormal Situation Centroid for Legs 1, 2 and 6 Walking On Soft Terrain.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0.73	0	0	0.69	0	0.16	0	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	0	0	0	0	0	0	0.74	0	0	0.97	0	0	1.07
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0	0	0	0.14	0	0	0	0	0	0.06	0	1.10	0	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
0.19	0.14	0.14	0.95	0.16	0.09	0	0	0	0	0	0		

Table 4.33. Abnormal Situation Centroid for Legs 4, 5 and 6 Walking On Soft Terrain.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	1.04	0	0	1.10	0	0	1.06	0	0	0	0	0	0
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0	0	0	0	0	1.14	0	0	1.00	0.13	0	1.06	0	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
0.20	1.56	0.33	1.06	0.13	0.13	0	0	0	0	0	0		

These ASCs were compared with results obtained from 5 experiments conducted for each case and the following discrepancy measures were obtained.

Table 4.34. Discrepancy Values from Experiments with Different Trios of Legs Walking Over Soft Terrain.

Experiment	1	2	3	4	5
Discrepancy (Legs 1, 3 and 4 on Soft Terrain)	1.35	0.82	0.49	0.84	0.84
Discrepancy (Legs 1, 3 and 5 on Soft Terrain)	0.59	0.73	0.80	0.76	0.61
Discrepancy (Legs 1, 2 and 6 on Soft Terrain)	0.46	0.67	0.71	0.77	0.63
Discrepancy (Legs 4, 5 and 6 on Soft Terrain)	0.56	0.69	0.71	0.79	0.87

Table 4.35 shows discrepancy values between the target ASC of the repertoire of conducted experiments and the ASC with the maximum discrepancy value from each row in Table 4.34.

Table 4.35. Discrepancy Values Between Obtained ASCs in Experiments with Different Trios of Legs Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	Discrepancy (Legs 1, 3 and 4 on Soft Terrain)	Discrepancy (Legs 1, 3 and 5 on Soft Terrain)	Discrepancy (Legs 1, 2 and 6 on Soft Terrain)	Discrepancy (Legs 4, 5 and 6 on Soft Terrain)
Robot Walking Uphill	4.33	4.39	4.41	4.43
Slippery Surface	5.43	5.29	5.42	5.29
Air Walking	2.63	3.10	2.36	3.52
Collision with Small Obstacles	3.73	3.86	3.89	4.14
Leg 1 on Soft Terrain	1.86	2.59	1.99	3.67
Leg 2 on Soft Terrain	2.83	2.97	1.90	3.53
Leg 3 on Soft Terrain	1.95	2.41	2.90	3.70
Leg 4 on Soft Terrain	2.38	3.27	2.88	3.24
Leg 5 on Soft Terrain	3.25	3.03	3.12	2.99
Leg 6 on Soft Terrain	3.32	3.24	2.14	2.95
Legs 1 and 2 on Soft Terrain	3.03	3.29	2.78	4.14
Legs 2 and 3 on Soft Terrain	3.12	3.40	3.38	4.32
Legs 5 and 6 on Soft Terrain	3.67	2.93	2.73	2.12
Legs 1 and 4 on Soft Terrain	2.69	3.20	2.76	3.36
Legs 1 and 6 on Soft Terrain	2.46	2.45	1.02	2.97
Legs 1, 3 and 4 on Soft Terrain	**1.35**	2.10	2.70	3.45
Legs 1, 3 and 5 on Soft Terrain	2.13	**0.80**	2.88	3.24
Legs 1, 2 and 6 on Soft Terrain	2.76	2.76	**0.77**	3.42
Legs 4, 5 and 6 on Soft Terrain	3.39	3.07	2.88	**0.87**

In general, for the 3 series of experiments analysed in this subsection, results show that when the robot has been previously provided with the ASC corresponding to the abnormal situation that must be identified, a correct first attempt diagnosis is performed in 100% of the cases. That is to say, the system is able to identify the anomaly affecting the robot in a database of 19 different abnormal situations.

4.5.3. Experiments with Anomalies in the Robot's Hardware and in the Robot's Environment

4.5.3.1 Robot Walking Uphill with Faulty Accelerometer

This experiment is similar to the one presented in subsection 4.5.2.1, but in this case the effects of the introduction of a faulty accelerometer on the performance of the self-diagnosis methods when the robot walks uphill will be analysed. Here, the fault is simulated by disconnecting the accelerometer. An example of the CAL values registered by the robot during this kind of experiments is shown in Fig. 4.43.

The high CAL values associated with the accelerometer give a clear indication that the anomaly is connected to this sensor. Then, the robot proceeds to execute accelerometer research actions and the faulty accelerometer is identified. In order to proceed with the diagnosis, the system ignores the accelerometer readings (by setting accelerometer CAL values to 0). This procedure was followed by the robot in 10 experiments. Results are presented in the form of discrepancy measures between the ASC of these experiments and the target ASC of experiments with other abnormal situations. The results of the 5 smallest discrepancy measures are shown in Table 4.36. Here, the smallest discrepancy measures for each experiment are displayed in bold numbers.

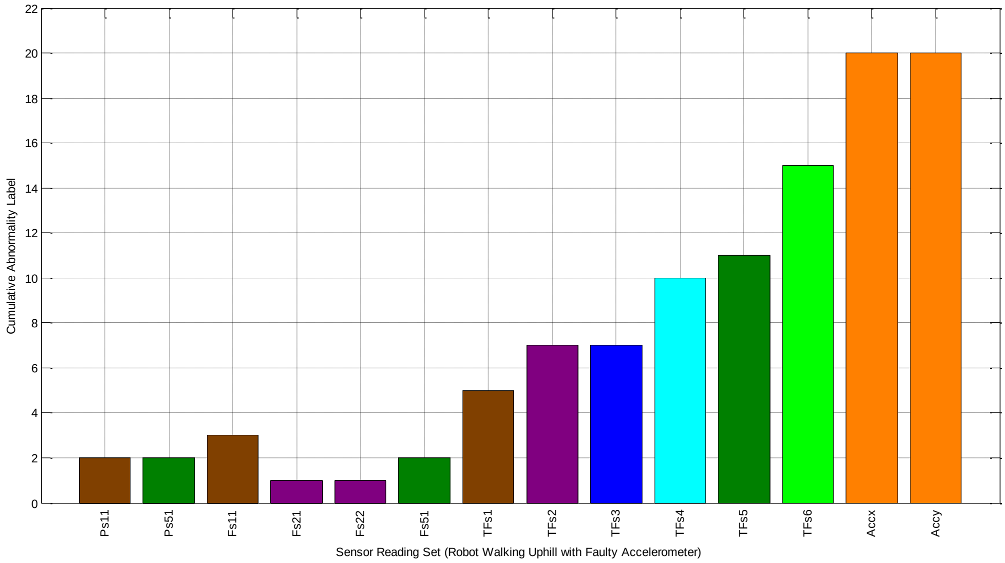


Figure 4.43. Cumulative Abnormality Labels from Experiment with Robot Walking Uphill and Faulty

Table 4.36. Discrepancy Values Between Obtained ASCs in Experiments with Robot Walking Uphill and Faulty Accelerometer, and Target ASCs of the Repertoire of Conducted Experiments (Target ASC for Experiments with Robot Walking Uphill and Faulty Accelerometer Not Incorporated).

Experiment	1	2	3	4	5	6	7	8	9	10
Robot Walking Uphill	1.74	**0.94**	**0.94**	**1.04**	**0.92**	**1.15**	**1.03**	**0.95**	**0.96**	**1.03**
Collision with Small Obstacles	**1.38**	1.13	1.05	1.16	1.00	1.40	1.16	1.12	1.15	1.25
Slippery Surface	3.13	2.13	2.06	1.94	2.02	1.83	2.05	1.97	1.94	1.89
Trapped legs 5 & 6, Faulty P. S.	2.21	2.22	2.34	2.53	2.47	2.67	2.56	2.43	2.56	2.60
Trapped leg 6, Faulty P. S.	2.80	2.23	2.25	2.52	2.48	2.51	2.65	2.32	2.47	2.52

The discrepancy measures show that in 9 of the 10 experiments the system was able to identify correctly the abnormal situation on its first attempt. In the other experiment, the robot first believed to be in the presence of a collision with a small obstacle. Only, on its second attempt was it able to identify the abnormal situation correctly.

Discrepancy measures could be reduced if a new ASC corresponding to robot walking uphill with faulty accelerometer is included in the robot's database. The ASC shown in Table 4.37 was calculated by using data obtained from the 10 previous experiments.

Table 4.37. Abnormal Situation Centroid for Robot Walking Uphill with Faulty Accelerometer.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0.03	0	0	0	0	0	0	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	0	0	0	0.06	0.03	0.04	0.03	0.03	0.01	0.17	0	0	0.01
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0.01	0	0	0	0	0.04	0	0	0.03	0	0	0	0	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
1.63	1.10	1.14	1.17	2.10	2.14	0	0	0	0	0	0		

Then, discrepancy measures in Table 4.38 are obtained if this ASC is included into the robot's database of abnormal situations.

Table 4.38. Discrepancy Values Between Obtained ASCs in Experiments with Robot Walking Uphill with Faulty Accelerometer and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5	6	7	8	9	10
Walking Uphill, Faulty Acc.	1.67	**0.43**	**0.50**	**0.48**	**0.24**	**0.61**	**0.44**	**0.45**	**0.34**	**0.42**
Robot Walking Uphill	1.74	0.94	0.94	1.04	0.92	1.15	1.03	0.95	0.96	1.03
Collision with Small Obstacles	**1.38**	1.13	1.05	1.16	1.00	1.40	1.16	1.12	1.15	1.25
Slippery Surface	3.13	2.13	2.06	1.94	2.02	1.83	2.05	1.97	1.94	1.89
Trapped legs 5 & 6, Faulty P. S.	2.21	2.22	2.34	2.53	2.47	2.67	2.56	2.43	2.56	2.60

Results show how discrepancy values were reduced. However, this reduction was insufficient for the correct identification of the abnormal situation on the first attempt during experiment 1.

4.5.3.2 Obstacle Collision with Faulty Whiskers.

In this experiment the robot hits a block of transparent material. Because the robot is able to sense light through the material, the course of locomotion is not altered significantly when the robot approaches the obstacle. As a result, the robot hits the obstacle directly. Normally, collisions with objects are detected by the robot's whiskers. However, in this set of 10 experiments, the whiskers have been disconnected in order to simulate a fault. Typical CAL values for this type of experiments are shown in Fig. 4.44. In general, the main difference between these CAL values and those representing obstacle collision with functional whiskers is the Ws label (0 in this case and 7 with functional whiskers). The corresponding ASC of obstacle collision with functional whiskers was included into the robot's database of abnormal situations. Then, a correct diagnosis of the system should indicate that an obstacle collision took place. This should be evidenced by a minimum discrepancy value between the ASC of the collision with faulty whiskers and the one corresponding to the obstacle collision with functional whiskers. Discrepancy

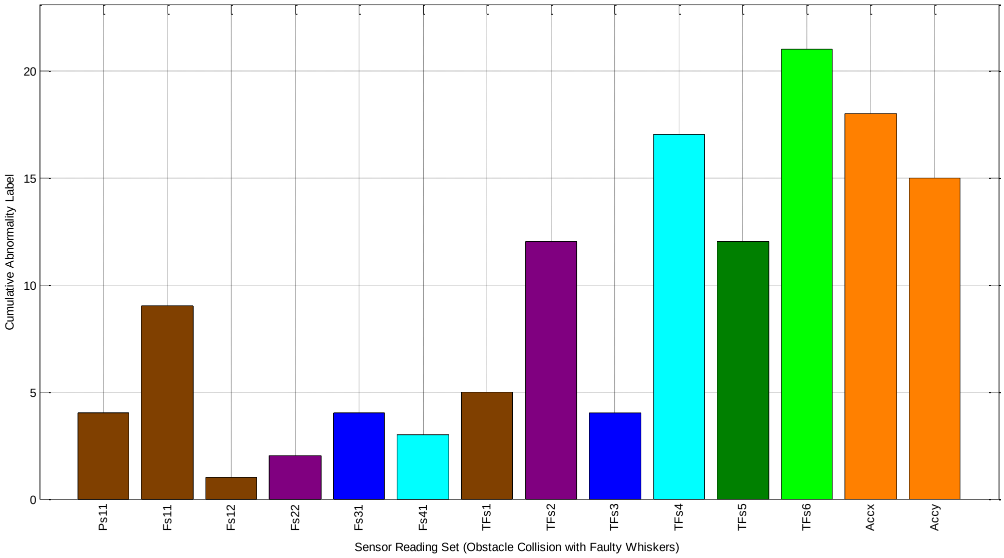


Figure 4.44. Cumulative Abnormality Labels from Experiment with Obstacle Collision and Faulty Whiskers.

measures corresponding to this set of 10 experiments considering obstacle collision with faulty whiskers are shown in Table 4.39.

Table 4.39. Discrepancy Values Between Obtained ASCs in Experiments with Obstacle Collision and Faulty Whiskers, and Target ASCs of the Repertoire of Conducted Experiments (Target ASC of Experiments with Obstacle Collision and Faulty Whiskers Not Incorporated).

Experiment	1	2	3	4	5	6	7	8	9	10
Obstacle Collision	**2.15**	**2.01**	**2.72**	**2.43**	2.96	**2.14**	2.16	**2.71**	2.39	2.31
Walking Uphill, Faulty Acc.	2.37	2.38	3.37	3.28	3.59	2.86	**1.95**	3.35	**2.35**	2.89
Robot Walking Uphill	3.66	3.55	3.94	2.78	**2.60**	3.52	3.62	3.71	4.13	**2.07**
Obstacle Collision, Faulty Front L. S.	2.45	3.55	3.68	4.19	3.27	3.83	3.11	3.98	3.56	3.07
Slippery Surface	4.49	3.92	4.32	3.11	3.38	4.33	4.62	4.43	4.67	2.91

Results show that in 6 of the 10 conducted experiments the robot was able to identify the abnormal situation on its first attempt, while in the rest of the experiments a correct diagnosis were performed on the second attempt.

As before, discrepancy measures could be reduced if a new ASC corresponding to obstacle collision with faulty whiskers is included in the robot's database of abnormal situations. The ASC shown in Table 4.40 were calculated by using data obtained from the 10 previous experiments.

Table 4.40. Abnormal Situation Centroid for Experiments with Obstacle Collision with Faulty Whiskers.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0.23	0	0	0.09	0	0	0	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0.06	0	0	0	0	0	0	0	0	0	0.74	0.16	0	0
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0.23	0	0.51	0	0	0.3	0.24	0	0	0	0	0.04	0	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
1.94	1.83	1.71	1.89	1.95	1.53	2.09	1.73	0	0	0	0		

Once this ASC is included into the robot's database of abnormal situations, discrepancy measures shown in Table 4.41 can be calculated.

Table 4.41. Discrepancy Values Between Obtained ASCs in Experiments with Obstacle Collision and Faulty Whiskers, and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5	6	7	8	9	10
Obstacle Collision, Faulty Ws	2.49	**1.65**	**2.19**	**1.72**	**2.52**	**2.07**	**1.54**	**1.98**	**1.76**	**1.76**
Obstacle Collision	**2.15**	2.01	2.72	2.43	2.96	2.14	2.16	2.71	2.39	2.31
Walking Uphill, Faulty Acc.	2.37	2.38	3.37	3.28	3.59	2.86	1.95	3.35	2.35	2.89
Robot Walking Uphill	3.66	3.55	3.94	2.78	2.6	3.52	3.62	3.71	4.13	2.07
Obstacle Collision, Faulty Front L. S.	2.45	3.55	3.68	4.19	3.27	3.83	3.11	3.98	3.56	3.07

In this case discrepancy values were reduced and in 9 out of 10 experiments, the robot was able to diagnose the abnormal situation correctly on its first attempt. In experiment 1, the robot identified the anomaly as an obstacle collision. Although this assumption was correct, this did not imply the presence of faulty whiskers. Because no research actions for whiskers were considered, the system is unable to verify if whisker readings are correct. In spite of this, experiments showed that obstacle collisions with faulty whiskers can still be identified if the ASC corresponding to this anomaly is incorporated into the robot.

4.5.3.3 Obstacle Collision with Faulty Front Light Sensors.

This experiment introduces a situation not covered by the experiments previously presented. In this case, the robot has faulty front light sensors which were already compensated. As a result, the robot walks backwards towards the light source. However, the robot commonly uses whiskers to detect obstacles and these do not protect the robot's back. Therefore, this could pose an issue for the correct diagnosis of the situation. A new set of 10 experiments has been conducted and compared with the previous set (obstacle collision with faulty whiskers) in order to look for differences in the performance of the self-diagnosis methods when dealing with these abnormal situations.

A typical set of CAL values for obstacle collision with faulty light sensors is shown in Fig. 4.45. Here, there are no CAL values for abnormal front light sensors because during these experiments it was assumed that this anomaly was already compensated.

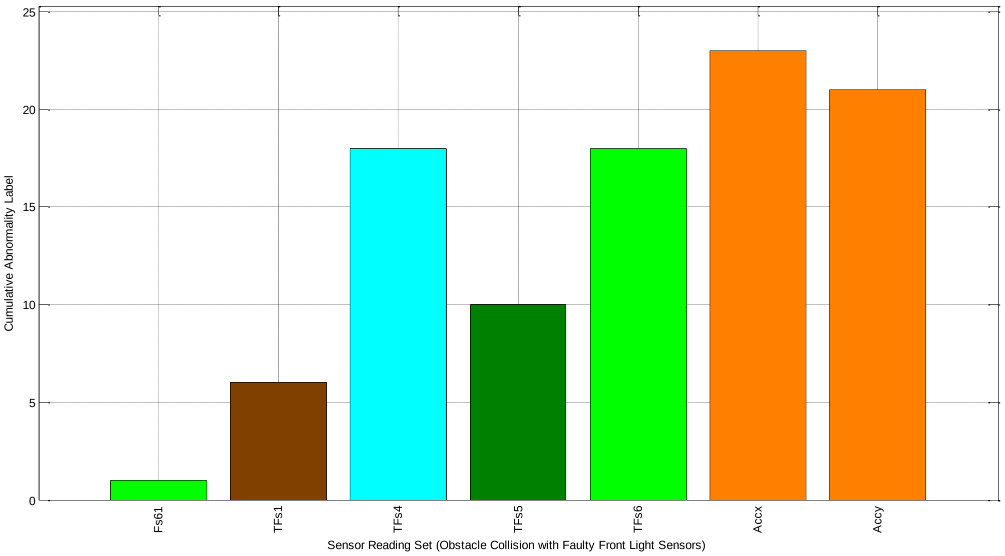


Figure 4.45. Cumulative Abnormality Labels from Experiment with Obstacle Collision and Faulty Front Light Sensors.

A correct diagnosis in these experiments could be indicated by minimum discrepancy levels between the ASC registered during the experiments and those corresponding to obstacle collisions. The obtained discrepancy measures are shown in Table 4.42.

Table 4.42. Discrepancy Values Between Obtained ASCs in Experiments with Obstacle Collision and Faulty Front Light Sensors, and Target ASCs of the Repertoire of Conducted Experiments (Target ASC of Experiments with Obstacle Collision and Faulty Front Light Sensors Not Incorporated).

Experiment	1	2	3	4	5	6	7	8	9	10
Obstacle Collision	3.44	3.09	3.21	3.01	2.92	3.64	3.36	3.14	2.69	3.48
Walking Uphill, Faulty Acc.	**2.15**	**2.31**	**2.57**	**2.34**	**1.60**	**2.72**	**1.93**	**2.73**	**1.73**	**2.24**
Obstacle Collision, Faulty Ws	3.67	3.38	2.97	2.99	3.06	3.54	3.39	3.17	3.00	3.35
Robot Walking Uphill	4.47	4.20	2.84	3.69	3.87	3.20	4.41	2.75	3.59	4.63
Collision with Small Obstacles	4.80	4.35	3.19	3.83	4.16	3.74	4.85	2.98	3.97	5.03
Trapped leg 2, Faulty P. S.	4.77	4.59	3.35	4.26	3.98	3.29	4.76	3.22	4.07	5.18

Results show that in 0 of the 10 conducted experiments the robot was able to identify the abnormal situation on its first attempt. This is mainly due to the orientation of the robot as it approaches the obstacle, which produces very different sensor reading in comparison with those normally generated during frontal obstacle collision. In these cases, the robot was able to diagnose the abnormal situation correctly with a second attempt in experiments 1,2,5,7 and 9. A third attempt was necessary in experiments 4 and 10. A fourth attempt was required in experiment 8. Finally, the robot was able to diagnose the abnormal situation on its fifth attempt in experiments 3 and 6.

Once again, discrepancy measures could be reduced if a new ASC corresponding to obstacle collision with faulty front light sensors is included in the robot's database of abnormal situations. The following ASC was calculated after data obtained from the 10 previous experiments was considered.

Table 4.43. Abnormal Situation Centroid for Experiments with Obstacle Collision and Faulty Front Light Sensors.

Ls1	Ls2	Ls3	Ls4	Ls5	Ls6	Ps11	Ps12	Ps13	Ps21	Ps22	Ps23	Ps31	Ps32
0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ps33	Ps41	Ps42	Ps43	Ps51	Ps52	Ps53	Ps61	Ps62	Ps63	Fs11	Fs12	Fs13	Fs21
0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fs22	Fs23	Fs31	Fs32	Fs33	Fs41	Fs42	Fs43	Fs51	Fs52	Fs53	Fs61	Fs62	Fs63
0	0	0	0	0	0	0	0	0	0	0	0.06	0	0
TFs1	TFs2	TFs3	TFs4	TFs5	TFs6	Accx	Accy	Bl1	Bl2	Bl3	Ws		
0.79	0.36	0.19	2.09	1.46	1.99	2.64	2.46	0	0	0	0		

After this ASC was included into the robot's database of abnormal situations, discrepancy measures shown in Table 4.44 were obtained.

Table 4.44. Discrepancy Values Between Obtained ASCs in Experiments with Obstacle Collision and Faulty Front Light Sensors, and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5	6	7	8	9	10
Obstacle Collision, Faulty Front L. S.	**1.21**	**1.68**	**1.67**	**1.37**	**1.23**	**2.23**	**1.17**	**1.67**	**1.07**	**1.67**
Walking Uphill, Faulty Acc.	2.15	2.31	2.57	2.34	1.6	2.72	1.93	2.73	1.73	2.24
Obstacle Collision	3.44	3.09	3.21	3.01	2.92	3.64	3.36	3.14	2.69	3.48
Obstacle Collision, Faulty Ws	3.67	3.38	2.97	2.99	3.06	3.54	3.39	3.17	3	3.35
Robot Walking Uphill	4.47	4.20	2.84	3.69	3.87	3.20	4.41	2.75	3.59	4.63
Collision with Small Obstacles	4.80	4.35	3.19	3.83	4.16	3.74	4.85	2.98	3.97	5.03

In this case discrepancy values were significantly reduced and in 10 out of 10 experiments the robot was able to diagnose the abnormal situation correctly on its first attempt.

4.5.3.4 Robot Walking Over Soft Terrain with Faulty Force Sensors.

Experiments conducted in subsection 4.5.2.6 showed that force sensors provide useful information for the diagnosis of abnormalities produced when the robot walks over soft terrain. In this new group of experiments, the same situation is faced by the robot but this time the robot's force sensors are not functional. Results obtained from these experiments will be used to evaluate the performance of the self-diagnosis algorithms when diagnosing these combined anomalies.

A first set of 5 experiments, for each one of the robot's leg walking over soft terrain while the rest of the legs walk over hard terrain, will be conducted. Typical CAL values for all of the experiments of this subsection are the same shown in Figs. 4.40 to 4.42, with the exception that in this case, CAL values corresponding to force sensors are 0.

A correct diagnosis of the consider abnormality is indicated by minimum discrepancy levels between the ASC registered during the experiments and the one, corresponding to the same leg walking over soft terrain, introduced in subsection 4.5.2.6. The obtained discrepancy measures are shown in Tables 4.45 to 4.50. Here, the first row of disparities is the one that is obtained after including the ASC of the experiment into the robot's database of abnormal situations. In addition, disparities in bold numbers represent the first attempt of classification if the ASC is not incorporated into the database.

Table 4.45. Discrepancy Values Between Obtained ASCs in Experiments with Leg 1 Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Leg 1 on Soft Terrain, Faulty F. S.	0.41	0.49	0.40	0.38	0.40
Leg 1 on Soft Terrain	1.61	1.61	1.60	1.60	1.60
Air Walking	**1.03**	1.35	1.30	**1.01**	1.25
Leg 2 on Soft Terrain, Faulty F. S.	1.11	1.37	1.33	1.09	1.24
Legs 1, 2 & 6 on Soft Terrain, Faulty F. S.	1.11	**1.28**	**1.22**	1.31	**1.02**
Legs 1 & 6 on Soft Terrain, Faulty F. S.	1.34	1.30	1.25	1.45	1.12
Leg 2 on Soft Terrain	1.47	1.60	1.47	1.40	1.57

Table 4.46. Discrepancy Values Between Obtained ASCs in Experiments with Leg 2 Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Leg 2 on Soft Terrain, Faulty F. S.	0.52	0.42	0.51	0.49	0.48
Leg 2 on Soft Terrain	**1.22**	**1.16**	**1.16**	1.23	1.15
Legs 1, 2 & 6 on Soft Terrain, Faulty F. S.	1.47	1.29	1.20	1.35	1.27
Air Walking	1.55	1.37	1.61	1.14	**1.03**
Leg 1 on Soft Terrain, Faulty F. S.	1.57	1.33	1.39	**1.01**	1.04

Table 4.47. Discrepancy Values Between Obtained ASCs in Experiments with Leg 3 Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Leg 3 on Soft Terrain, Faulty F. S.	1.11	0.55	1.02	0.77	0.44
Leg 3 on Soft Terrain	1.82	**1.28**	**1.37**	**1.19**	**1.21**
Air Walking	**0.88**	1.68	2.03	2.10	1.59
Leg 1 on Soft Terrain, Faulty F. S.	1.22	1.83	2.17	2.25	1.82
Leg 4 on Soft Terrain, Faulty F. S.	1.58	1.90	1.78	1.90	1.62
Legs 1, 3 & 4 on Soft Terrain, Faulty F. S.	2.29	1.91	2.26	2.25	2.03
Leg 2 on Soft Terrain, Faulty F. S.	1.37	1.93	2.24	2.32	1.90
Legs 1, 3 & 5 on Soft Terrain, Faulty F. S.	2.19	2.00	2.60	2.41	2.22
Leg 2 on Soft Terrain	1.67	2.15	2.36	2.52	2.20

Table 4.48. Discrepancy Values Between Obtained ASCs in Experiments with Leg 4 Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Leg 4 on Soft Terrain, Faulty F. S.	0.94	0.63	0.55	0.63	0.62
Leg 4 on Soft Terrain	**1.28**	**1.26**	1.70	**1.29**	**1.20**
Air Walking	1.83	1.95	**1.18**	1.87	2.11
Leg 1 on Soft Terrain, Faulty F. S.	1.94	1.84	1.28	2.02	2.01
Leg 2 on Soft Terrain, Faulty F. S.	2.04	1.94	1.47	2.11	2.09
Leg 3 on Soft Terrain, Faulty F. S.	1.58	1.82	1.61	1.65	1.88
Leg 2 on Soft Terrain	2.25	2.14	1.64	2.29	2.30

Table 4.49. Discrepancy Values Between Obtained ASCs in Experiments with Leg 5 Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Leg 6 on Soft Terrain, Faulty F. S.	0.56	0.75	0.53	0.68	0.69
Leg 5 on Soft Terrain	**1.27**	**1.29**	**1.25**	**1.30**	**1.29**
Leg 6 on Soft Terrain, Faulty F. S.	1.68	1.78	1.76	1.53	1.70
Legs 5 & 6 on Soft Terrain, Faulty F. S.	1.71	1.67	1.72	1.81	1.49
Air Walking	1.89	2.14	2.04	1.74	2.19
Leg 6 on Soft Terrain	1.91	1.96	1.90	1.63	1.97

Table 4.50. Discrepancy Values Between Obtained ASCs in Experiments with Leg 6 Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Leg 6 on Soft Terrain, Faulty F. S.	0.56	0.75	0.53	0.68	0.69
Leg 6 on Soft Terrain	**1.37**	**1.50**	**1.37**	**1.39**	**1.38**
Leg 5 on Soft Terrain, Faulty F. S.	1.69	2.09	1.89	1.91	1.72
Legs 5 & 6 on Soft Terrain, Faulty F. S.	1.84	1.52	1.66	1.44	1.70
Leg 5 on Soft Terrain	1.92	2.30	1.95	2.10	2.02
Legs 1 & 2 on Soft Terrain, Faulty F. S.	2.08	2.29	2.11	2.13	2.13

Experiments show different results depending on the leg walking on soft terrain. Worst results without providing the robot with the ASC of the abnormality were obtained when leg 1 walked on soft terrain. Here, the system was able to diagnose the abnormal situation correctly only after 5 failed attempts. On the other hand, in 4 out of 5 experiments, the robot was able to identify the abnormality on its first attempt when legs 3 or 4 walked on soft terrain. In addition, the robot did not require the ASC of the abnormal situation for diagnosing correctly at the first attempt when legs 5 or 6 walked on soft terrain. Once the ASC of the corresponding experiment was incorporated into the robot's database of abnormalities, the robot was able to identify the anomaly on its first attempt in almost all of the experiments. The only exception occurred in experiment 1, with leg 3 walking on soft terrain, where a second attempt was necessary.

Another set of 5 experiments for combinations of 2 legs walking on soft terrain was conducted. The obtained discrepancy values are shown in Tables 4.51 to 4.55.

Table 4.51. Discrepancy Values Between Obtained ASCs in Experiments with Legs 1 and 2 Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Legs 1 & 2 on Soft Terrain, Faulty F. S.	0.68	0.58	0.46	0.94	0.71
Legs 1 and 2 on Soft Terrain	**1.89**	1.93	1.86	2.07	2.00
Legs 2 & 3 on Soft Terrain, Faulty F. S.	1.93	**1.89**	1.94	2.32	1.87
Legs 1, 2 & 6 on Soft Terrain, Faulty F. S.	1.95	1.79	1.81	2.19	**1.65**
Leg 2 on Soft Terrain, Faulty F. S.	1.99	1.90	2.02	2.53	1.78
Leg 1 on Soft Terrain, Faulty F. S.	2.15	2.10	2.09	2.37	1.95
Legs 1 & 4 on Soft Terrain, Faulty F. S.	2.27	2.08	**1.80**	**1.73**	2.00

Table 4.52. Discrepancy Values Between Obtained ASCs in Experiments with Legs 2 and 3 Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Legs 2 & 3 on Soft Terrain, Faulty F. S.	0.75	0.99	0.65	0.50	0.93
Legs 2 and 3 on Soft Terrain	**1.84**	**1.95**	1.84	**1.79**	2.02
Legs 1 & 2 on Soft Terrain, Faulty F. S.	2.13	2.35	**1.70**	1.95	**1.38**
Leg 5 on Soft Terrain, Faulty F. S.	2.27	2.80	2.01	2.40	1.69
Leg 5 on Soft Terrain	2.37	2.85	2.23	2.48	1.94
Leg 3 on Soft Terrain, Faulty F. S.	2.45	2.08	**1.70**	2.24	2.20
Leg 6 on Soft Terrain, Faulty F. S.	2.50	3.01	2.22	2.58	1.87
Leg 6 on Soft Terrain	2.57	3.04	2.34	2.76	1.94

Table 4.53. Discrepancy Values Between Obtained ASCs in Experiments with Legs 5 and 6 Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Legs 5 & 6 on Soft Terrain, Faulty F. S.	0.50	0.60	0.74	0.54	0.66
Legs 5 and 6 on Soft Terrain	**1.56**	**1.63**	1.74	**1.67**	1.61
Leg 5 on Soft Terrain, Faulty F. S.	1.66	1.74	**1.34**	1.81	**1.33**
Leg 6 on Soft Terrain, Faulty F. S.	1.57	1.77	1.84	1.68	1.40
Leg 5 on Soft Terrain	1.98	2.04	1.78	2.09	1.71
Leg 6 on Soft Terrain	1.93	2.14	2.13	2.04	1.77

Table 4.54. Discrepancy Values Between Obtained ASCs in Experiments with Legs 1 and 4 Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Legs 1 & 4 on Soft Terrain, Faulty F. S.	0.51	0.66	0.85	0.64	0.56
Legs 1 and 4 on Soft Terrain	**1.56**	**1.52**	**1.52**	**1.54**	**1.51**
Leg 6 on Soft Terrain	1.84	1.91	1.61	2.24	2.07
Legs 1, 3 & 4 on Soft Terrain, Faulty F. S.	2.01	2.03	2.45	1.96	2.29
Legs 1 & 2 on Soft Terrain, Faulty F. S.	2.03	1.67	1.80	2.17	2.08
Legs 1 and 6 on Soft Terrain	2.20	2.08	2.55	2.39	2.51

Table 4.55. Discrepancy Values Between Obtained ASCs in Experiments with Legs 1 and 6 Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Legs 1 & 6 on Soft Terrain, Faulty F. S.	0.72	0.48	0.41	0.68	0.63
Legs 1 and 6 on Soft Terrain	1.45	1.36	1.33	1.31	1.44
Legs 1, 2 & 6 on Soft Terrain, Faulty F. S.	**1.08**	**1.24**	**1.23**	1.28	**1.35**
Leg 1 on Soft Terrain, Faulty F. S.	1.17	1.26	1.34	**1.10**	1.48
Leg 2 on Soft Terrain	1.72	1.98	1.97	1.68	2.18
Leg 2 on Soft Terrain, Faulty F. S.	1.78	2.03	2.02	1.78	2.16

Results of this series of experiments showed that the robot was able to identify all of the abnormal situations on the first attempt if the corresponding ASC representing the anomaly was provided. When this was not the case, results vary from experiment to experiment. Best results were obtained when legs 1 and 4 walked on soft terrain. Here, the robot identified the abnormal situation at the first attempt. In contrast, legs 1 and 6 walking on soft terrain presented the worst results. In this case, the robot failed on its first attempt of diagnosis in all of the 5 experiments of the series.

Sets of 5 experiments considering 3 legs walking on soft terrain were also conducted. The obtained discrepancy measures are shown in Tables 4.56 to 4.59.

Table 4.56. Discrepancy Values Between Obtained ASCs in Experiments with Legs 1, 3 and 4 Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Legs 1, 3 & 4 on Soft Terrain, Faulty F. S.	0.46	0.70	0.67	0.78	0.68
Legs 1, 3 and 4 on Soft Terrain	**1.94**	1.99	1.97	**2.01**	2.02
Legs 1, 3 & 5 on Soft Terrain, Faulty F. S.	2.00	1.76	**1.66**	2.08	2.39
Leg 3 on Soft Terrain, Faulty F. S.	2.05	2.17	1.78	**2.01**	2.13
Leg 4 on Soft Terrain, Faulty F. S.	2.10	2.15	1.94	2.16	**1.93**
Legs 1 & 4 on Soft Terrain, Faulty F. S.	2.15	1.94	2.22	2.20	2.32
Legs 1 and 6 on Soft Terrain	2.25	**1.86**	2.01	2.04	2.49
Leg 1 on Soft Terrain, Faulty F. S.	2.34	2.03	1.80	2.07	2.47

Table 4.57. Discrepancy Values Between Obtained ASCs in Experiments with Legs 1, 3 and 5 Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Legs 1, 3 & 5 on Soft Terrain, Faulty F. S.	0.61	0.70	0.61	0.75	0.77
Legs 1, 3 and 5 on Soft Terrain	1.97	2.04	1.97	**2.06**	**2.07**
Legs 1 & 6 on Soft Terrain, Faulty F. S.	**1.65**	**1.64**	1.99	2.18	2.49
Leg 1 on Soft Terrain, Faulty F. S.	1.66	1.71	1.97	2.14	2.47
Legs 1, 3 & 4 on Soft Terrain, Faulty F. S.	1.81	2.14	**1.87**	2.27	2.13
Legs 1 and 6 on Soft Terrain	1.82	2.05	2.06	2.36	2.53
Legs 1, 2 & 6 on Soft Terrain, Faulty F. S.	1.85	1.84	2.05	2.30	2.66
Leg 3 on Soft Terrain, Faulty F. S.	1.96	2.22	2.13	2.13	2.45

Table 4.58. Discrepancy Values Between Obtained ASCs in Experiments with Legs 1, 2 and 6 Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Legs 1, 2 & 6 on Soft Terrain, Faulty F. S.	0.61	0.57	0.50	0.57	0.57
Legs 1, 2 and 6 on Soft Terrain	1.93	1.97	1.90	1.96	1.93
Legs 1 & 6 on Soft Terrain, Faulty F. S.	**1.25**	**1.37**	1.47	**1.19**	**1.14**
Leg 1 on Soft Terrain, Faulty F. S.	1.48	1.52	1.26	1.30	1.50
Leg 2 on Soft Terrain, Faulty F. S.	1.50	1.47	**0.91**	1.40	1.51
Leg 2 on Soft Terrain	1.71	1.70	1.31	1.67	1.70
Legs 1 and 6 on Soft Terrain	1.74	1.85	1.82	1.70	1.64
Legs 1 & 2 on Soft Terrain, Faulty F. S.	2.12	1.66	1.90	1.71	2.11
Air Walking	2.14	2.20	1.67	1.99	2.14

Table 4.59. Discrepancy Values Between Obtained ASCs in Experiments with Legs 4, 5 and 6 Walking Over Soft Terrain and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Legs 4, 5 & 6 on Soft Terrain, Faulty F. S.	0.78	0.46	0.68	0.63	0.73
Legs 4, 5 and 6 on Soft Terrain	2.05	1.90	2.00	1.99	1.98
Trapped leg 1, Faulty P. S..	**0.78**	**0.46**	**0.68**	**0.63**	**0.73**
Legs 5 & 6 on Soft Terrain, Faulty F. S.	2.15	1.91	2.12	2.06	2.10
Legs 5 and 6 on Soft Terrain	2.56	2.33	2.50	2.47	2.49
Legs 1, 3 & 5 on Soft Terrain, Faulty F. S.	2.77	2.12	2.22	2.65	2.35

In general, results show that more attempts are necessary for the correct diagnosis of the anomaly without incorporating the corresponding ASC into the robot's database of abnormal situations. Only in a maximum of 2 out of 5 experiments, when legs 1, 3 and 4 or 1, 3 and 5 walked over soft terrain, was it possible to identify the anomaly at the first attempt. Once the ASC of the abnormal situation was incorporated into the robot, the system was able to diagnose correctly on its first attempt. An exception could occur when legs 4, 5 and 6 walked on soft terrain. Here, discrepancy measures are similar to the abnormal situation where leg 1 is trapped. In this case the system will correctly diagnose the abnormality in a maximum of 2 attempts.

4.5.3.5 Trapped Legs with Faulty Position Sensors.

In this series of experiments one or two of the robot's legs are firmly held in order to simulate a trapped leg. In addition, the position sensors of the held leg have been disconnected in order to simulate an internal fault. In most cases, a trapped leg was unable to get to its target position, which would generate large CAL values corresponding to the leg's position sensors. Therefore, the disconnection of these sensors imposes a greater challenge on the self-diagnosis methods. Then, the aim of this new set of experiments is to evaluate the behaviour of the self-diagnosis algorithms when dealing with this situation.

In this first series of experiments only one of the robot's legs is held at the time. For each leg, 5 experiments were conducted. Typical CAL values for these experiments are shown in Fig. 4.46.

The resulting discrepancy measures, for experiments where one of the robot's legs is held, are shown in Tables 4.60 to 4.65. Disparities showed in the second row of these tables are the ones that would be obtained if the ASC of the analysed anomaly was incorporated into the robot's database of abnormal situations. Otherwise, if this ASC is not incorporated, correct first attempt identifications will be represented by a minimum discrepancy value (written in bod numbers) in the third row of the tables. A minimum discrepancy value in any other row would indicate that more than one attempt was necessary for the correct identification of the abnormal situation.

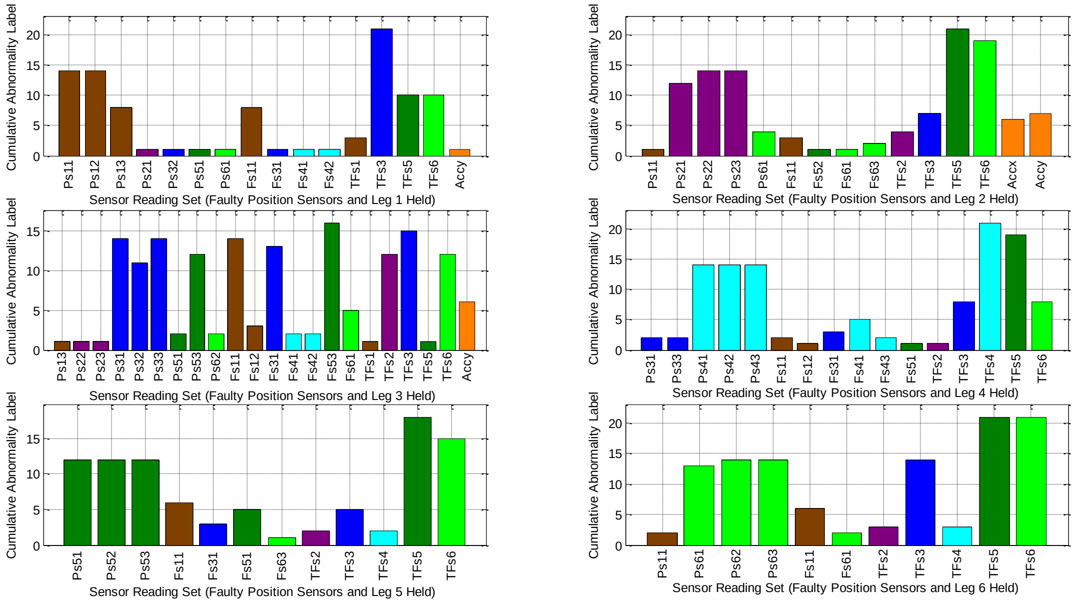


Figure 4.46. Cumulative Abnormality Labels from Experiments with Faulty Position Sensors and One Leg Held.

Table 4.60. Discrepancy Values Between Obtained ASCs in Experiments with Leg 1 Held and Faulty Position Sensors, and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Trapped leg 1, Faulty P. S..	0.61	0.65	0.71	0.72	0.58
Trapped leg 1	5.26	5.41	5.50	5.25	5.34
:	:	:	:	:	:
Legs 4, 5 & 6 on Soft Terrain, Faulty F. S.	**0.61**	**0.65**	**0.71**	**0.72**	**0.58**
Legs 4, 5 and 6 on Soft Terrain	1.94	1.97	2.01	2.04	1.96
Legs 5 & 6 on Soft Terrain, Faulty F. S.	2.15	2.30	2.05	2.29	2.43
Legs 1, 3 & 5 on Soft Terrain, Faulty F. S.	2.20	2.40	2.59	2.71	2.56
Legs 1 & 6 on Soft Terrain, Faulty F. S.	2.46	2.64	2.67	2.69	2.41

Table 4.61. Discrepancy Values Between Obtained ASCs in Experiments with Leg 2 Held and Faulty Position Sensors, and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Trapped leg 2, Faulty P. S.	0.50	1.33	0.87	1.30	1.04
Trapped leg 2	3.36	3.58	3.46	3.62	3.55
:	:	:	:	:	:
Trapped legs 5 & 6, Faulty P. S.	**1.73**	2.17	**1.80**	**1.35**	1.77
Trapped leg 6, Faulty P. S.	1.80	1.94	1.93	2.04	**1.71**
Trapped legs 1 & 6, Faulty P. S.	1.89	**1.79**	2.17	1.99	1.85
Trapped leg 5, Faulty P. S.	1.97	2.05	2.00	1.56	2.20
Robot Walking Uphill	2.69	3.05	2.77	2.15	2.27

Table 4.62. Discrepancy Values Between Obtained ASCs in Experiments with Leg 3 Held and Faulty Position Sensors, and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Trapped leg 3, Faulty P. S.	0.77	1.41	1.63	1.58	1.41
Trapped leg 3	**3.10**	**3.31**	**3.41**	**3.39**	**3.31**
Trapped legs 5 & 6, Faulty P. S.	4.77	4.51	4.86	4.90	5.04
Trapped legs 2 & 3, Faulty P. S.	4.79	4.47	5.12	5.25	4.95
Legs 1, 3 and 5 on Soft Terrain	4.83	4.45	4.58	5.28	5.35
Collision with Small Obstacles	4.85	4.23	4.67	5.05	4.97

Table 4.63. Discrepancy Values Between Obtained ASCs in Experiments with Leg 4 Held and Faulty Position Sensors, and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Trapped leg 4, Faulty P. S.	0.37	0.89	0.70	1.15	0.96
Trapped leg 4	3.48	3.57	3.53	3.65	3.59
:	:	:	:	:	:
Trapped legs 1 & 4, Faulty P. S.	**2.40**	**2.81**	**2.42**	**2.31**	**2.87**
Collision with Small Obstacles	2.74	3.22	2.69	2.78	3.12
Robot Walking Uphill	2.81	3.21	2.64	2.90	3.02
Leg 4 on Soft Terrain	3.20	3.46	3.47	3.33	3.57
Trapped legs 5 & 6, Faulty P. S.	3.31	3.63	3.19	3.73	3.49

Table 4.64. Discrepancy Values Between Obtained ASCs in Experiments with Leg 5 Held and Faulty Position Sensors, and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Trapped leg 5, Faulty P. S.	0.85	0.72	1.11	1.08	0.72
Trapped leg 5	3.42	3.38	3.49	3.48	3.38
:	:	:	:	:	:
Trapped legs 5 & 6, Faulty P. S.	**1.26**	1.74	**1.72**	1.93	**2.03**
Trapped leg 2, Faulty P. S.	1.72	**1.56**	1.96	**1.92**	2.13
Trapped leg 6, Faulty P. S.	1.90	1.74	1.99	2.12	2.20
Trapped legs 1 & 6, Faulty P. S.	1.92	1.78	2.30	1.96	2.36
Collision with Small Obstacles	2.29	2.41	2.15	2.41	2.38

Table 4.65. Discrepancy Values Between Obtained ASCs in Experiments with Leg 6 Held and Faulty Position Sensors, and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Trapped leg 6, Faulty P. S.	0.64	1.10	1.07	0.91	0.99
Trapped leg 6	3.82	3.92	3.91	3.87	3.89
:	:	:	:	:	:
Trapped legs 5 & 6, Faulty P. S.	**1.58**	**1.87**	1.93	1.80	**1.62**
Trapped leg 2, Faulty P. S.	1.67	2.09	1.78	**1.57**	1.86
Trapped legs 1 & 6, Faulty P. S.	1.87	2.37	**1.66**	1.89	1.95
Trapped leg 5, Faulty P. S.	1.89	2.22	2.22	2.08	2.29
Robot Walking Uphill	2.69	2.56	2.58	2.77	2.57

Discrepancy values in this set of experiments show that a larger number of diagnosis attempts are necessary when no information about the abnormal situation is provided to the robot. For instance, when leg 6 is held, in experiment 1 of Table 4.65, the robot must perform 14 attempts at diagnosis before identifying correctly the abnormal situation. This situation changes when the corresponding ASC is included into the robot's database of abnormalities. In those cases, the robot was able to identify the abnormal situation on its first attempt. The only exception could occur in experiments where leg 1 was held. Here, the discrepancy values are the same as those obtained during experiments with legs 4, 5 and 6 on soft terrain with faulty force sensors. Therefore, here the robot correctly identifies the abnormal situation in a maximum of 2 attempts.

A second series of experiments were conducted where combinations of 2 legs are held at the time. Once again, 5 experiments were conducted for each combination of legs. Typical CAL values registered during these experiments are shown in Fig. 4.47. In addition, resulting discrepancy values are shown in Tables 4.66 to 4.70.

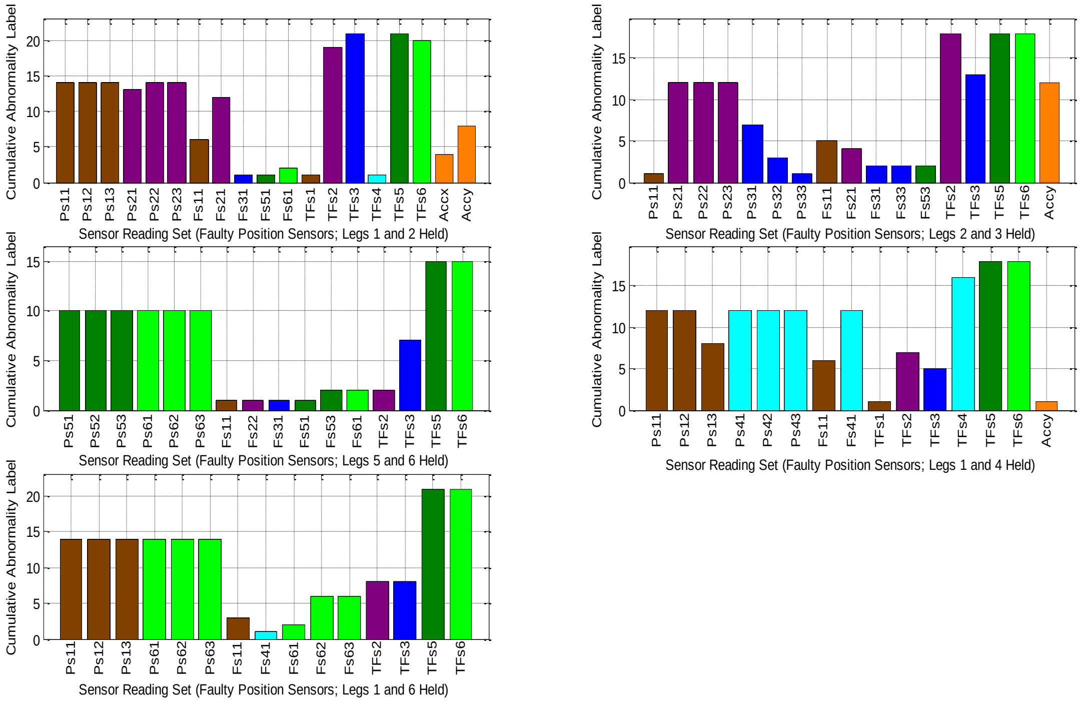


Figure 4.47. Cumulative Abnormality Labels from Experiments with Faulty Position Sensors and Pairs of Legs Held.

Table 4.66. Discrepancy Values Between Obtained ASCs in Experiments with Faulty Position Sensors and Legs 1 and 2 Held, and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Trapped legs 1 & 2, Faulty P. S.	0.54	1.41	1.49	1.16	1.08
Trapped legs 1 & 2	5.16	5.32	5.35	5.26	5.24
:	:	:	:	:	:
Trapped legs 2 & 3, Faulty P. S.	**1.74**	**1.84**	**2.75**	**2.29**	**1.74**
Trapped leg 6, Faulty P. S.	3.41	3.47	4.37	3.75	3.13
Trapped leg 2, Faulty P. S.	3.54	3.59	4.51	3.97	3.14
Trapped legs 1 & 6, Faulty P. S.	3.57	3.69	4.44	4.01	3.36
Trapped legs 2 & 3	3.77	3.82	4.33	4.05	3.77

Table 4.67. Discrepancy Values Between Obtained ASCs in Experiments with Faulty Position Sensors and Legs 2 and 3 Held, and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Trapped legs 2 & 3, Faulty P. S.	0.70	1.42	1.02	1.23	1.37
Trapped legs 2 & 3	3.42	3.63	3.49	3.56	3.61
:	:	:	:	:	:
Trapped legs 1 & 2, Faulty P. S.	**2.24**	**2.31**	**1.72**	**2.45**	**2.27**
Trapped leg 2, Faulty P. S.	2.69	3.67	2.76	2.68	3.39
Trapped legs 1 & 6, Faulty P. S.	2.89	3.73	2.82	2.94	3.21
Trapped leg 6, Faulty P. S.	3.01	3.84	2.85	3.09	3.51
Trapped legs 5 & 6, Faulty P. S.	3.05	4.26	3.21	3.11	3.72

Table 4.68. Discrepancy Values Between Obtained ASCs in Experiments with Faulty Position Sensors and Legs 5 and 6 Held, and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Trapped legs 5 & 6, Faulty P. S.	0.42	0.82	0.69	0.66	0.91
Trapped legs 5 & 6	3.68	3.75	3.72	3.72	3.77
:	:	:	:	:	:
Trapped leg 2, Faulty P. S.	**1.62**	**1.85**	1.69	**1.72**	**1.28**
Trapped leg 5, Faulty P. S.	1.72	2.09	2.01	1.97	1.46
Trapped leg 6, Faulty P. S.	1.76	1.96	**1.57**	1.74	1.38
Trapped legs 1 & 6, Faulty P. S.	1.88	1.94	1.64	1.97	1.54
Collision with Small Obstacles	2.27	2.28	2.38	2.26	2.51

Table 4.69. Discrepancy Values Between Obtained ASCs in Experiments with Faulty Position Sensors and Legs 1 and 4 Held, and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Trapped legs 1 & 4, Faulty P. S.	0.44	1.21	1.54	1.47	0.84
Trapped legs 1 & 4	4.10	4.25	4.36	4.33	4.16
:	:	:	:	:	:
Trapped leg 4, Faulty P. S.	**2.31**	3.22	3.11	2.45	2.14
Collision with Small Obstacles	2.33	2.58	**2.57**	**2.41**	**1.98**
Robot Walking Uphill	2.57	2.83	2.91	2.52	2.15
Trapped leg 5, Faulty P. S.	2.84	2.68	3.25	3.53	2.82
Trapped legs 1 & 6, Faulty P. S.	2.85	**2.49**	3.46	3.18	2.59

Table 4.70. Discrepancy Values Between Obtained ASCs in Experiments with Faulty Position Sensors and Legs 1 and 6 Held, and Target ASCs of the Repertoire of Conducted Experiments.

Experiment	1	2	3	4	5
Trapped legs 1 & 6, Faulty P. S.	0.71	0.87	1.40	1.09	1.26
Trapped legs 1 & 6	5.22	5.25	5.36	5.29	5.32
:	:	:	:	:	:
Trapped leg 2, Faulty P. S.	**1.66**	1.92	1.62	2.27	**1.80**
Trapped leg 6, Faulty P. S.	1.70	**1.89**	**1.52**	**2.12**	2.00
Trapped legs 5 & 6, Faulty P. S.	1.78	1.93	1.67	2.28	1.97
Trapped leg 5, Faulty P. S.	2.03	2.18	2.08	2.55	2.02
Collision with Small Obstacles	2.81	2.47	2.78	2.57	3.04

In this last series of experiments, discrepancy values show that a considerable number of diagnosis attempts are necessary when no information about the abnormal situation is provided to the robot. For example, when legs 1 and 6 are held, in experiment 1 of Table 4.70, the robot must perform 54 attempts at diagnosis before identifying correctly the abnormal situation. This number varies from experiment to experiment even when the same combination of legs is considered. However, discrepancy values are larger than in experiments where only on leg was held. Results improve greatly when the corresponding ASC is included into the robot's database of abnormal situations. In all of those cases, the robot was able to identify the abnormal situation on its first attempt.

4.5.4. Analysis of Experimental Results

Experiments with anomalies in the robot's hardware have shown that faults with isolated effects detected by sensors directly linked to the fault are normally identified correctly by the self-diagnosis methods on the first attempt. As these effects propagate to other sensors, the system needs more attempts before a correct identification is achieved. This is especially true when the effects on sensors directly linked to the fault are smaller than for other sensors. On the other hand, there are cases when a fault may not produce any abnormal sensor reading at all. Although, this situation could be tackled by improving the methods for the calculation of expected sensor readings, in some cases it may be necessary to increase the number and/or type of sensors in the robot. However, the effects of faults among the increased number of robot sensors could propagate and increase the complexity of arriving at a correct diagnosis in certain situations.

The experiments that were conducted with anomalies in the robot's environment showed the good performance of the self-diagnosis methods when identifying situations that generated abnormal sensor readings. However, here the system struggles once again to identify those situations where abnormal sensor readings are not generated. This occurs, for instance, while the robot is air walking or when one of the two light sensors facing the light source is partially covered.

A disadvantage of the proposed self-diagnosis methods is the time they require for the identification of abnormal situations generated in the robot's environment when a large number of abnormal sensor readings are generated. In these cases, the robot generates many theories about possible sources of the anomaly. Then, the corresponding research actions must be executed and these theories are rejected one by one. Only once the possibility of a fault in the robot's hardware is discarded, does the system turn to the identification of abnormal situations in the robot's environments. Additional research could be conducted in order to implement a faster method for differentiating the two types of anomaly sources.

The self-diagnosis methods also showed a good identification rate in experiments where both anomalies in the robot's hardware and in the robot's environments were considered. Best results were achieved in those cases where the ASC corresponding to the target abnormal situation was previously incorporated into the robot's database of abnormalities. In some cases, the robot was still able to identify the abnormal situation correctly at the first attempt when this information was not available. However, in other cases with unknown target ASC the performance of the system dropped drastically. For instance, the robot had to perform more than 54 attempts of diagnosis before identifying correctly the abnormal situation when legs 1 and 6 were held, in experiment 1 of Table 4.70.

In general, the rate of first attempt identification of abnormal situations in the robot's environment will depend on the number and types of the ASCs located in the robot's database of abnormalities. Discrepancy values are usually larger for linearly independent ASCs. Therefore, better diagnosis is performed for databases incorporating this type of ASCs. In the presence of linearly dependent ASCs, the classification of the abnormal situation will depend more on the magnitude of the ASC components. This dependence between the performance of the self-diagnosis algorithms and the nature and size of the database of abnormalities incorporated into the robot increases the complexity of the evaluation of the system. The database used during the evaluation of the self-diagnosis methods incorporated data gathered in experiments presented in sections 4.4 and 4.5. The creation of a larger database of experiments may allow a more exhaustive evaluation of the self-diagnosis methods. However, a practical implementation of this database goes beyond the scope of this thesis and it is proposed as future work.

4.6. Summary

This chapter has introduced a method for autonomous identification and classification of detrimental disturbance sources as a self-diagnosis technique for robots facing abnormal situations.

Initially, section 4.1 has presented two methods for the detection of detrimental abnormal conditions. Here, poor robot performance or extreme abnormal sensor readings (pain) are used as inputs for the disturbance detection algorithm. Then, a classification algorithm has been introduced in section 4.2. As a result of the application of this algorithm, abnormal situations have been classified into centroids of the sensor reading space generated by the robot experiencing the abnormal conditions. The output of the classification algorithm will be utilised by the robot self-compensation techniques presented in chapter 5. Furthermore, use of information relating to the robot's anatomy for accelerating the search for possible disturbance sources has been proposed in section 4.3. This search is executed using the robot's actuators and consists of a series of specialised research actions. In general, each abnormal symptom has an associated research action. These movements allow the validation or rejection of the robot's assumptions about the source of an abnormal condition. The method presented here will be complemented by techniques introduced in section 5.3 of the next chapter. In addition, the algorithms and techniques discussed in this chapter have been illustrated by means of two experiments in section 4.4. Finally, experiments had been conducted in section 4.5 in order to evaluate the self-diagnosis methods discussed in the chapter. The experimental results showed the capability of the proposed method for detection of abnormal conditions and identification of their source. In addition, it was shown that the performance of the self-diagnosis algorithms was affected by the propagation of the effects of abnormal conditions to sensors not directed linked to the anomaly. In most of the considered cases, the proposed method was able to find those moving parts responsible for or able to compensate the introduced disturbance. However, results may be affected by the nature and size of the robot's database of abnormalities.

This chapter has dealt with the detection, identification and classification of detrimental disturbances. Once the abnormal condition source is identified, or the list of possible sources is reduced, it is necessary to execute compensating actions to overcome or at least reduce the problem. This topic is the subject of the next chapter.

Chapter 5
Compensation of Detrimental Disturbances

Techniques for robotic self-compensation of detrimental disturbances are presented in this chapter. These methods are biologically inspired by evolved and instinctive behaviours. The proposed techniques allow robots to find suitable compensatory actions when experiencing known abnormal situations. This chapter also addresses the autonomous generation of actions able to compensate detrimental disturbance where the source is unknown and the selection of the robot's moving parts executing those actions.

Chapter 5 is structured as follows. Initially, innate compensatory measures and application examples using the experimental robot are presented in sections 5.1 and 5.2, respectively. Then, in section 5.3, the use of compensatory measures in the final determination of detrimental disturbance sources is addressed. This is followed by section 5.4, where autonomous generation of compensatory actions and selection of the robot's moving parts for executing those actions is discussed. In addition, section 5.5 introduces "last resort" measures, which are used if the techniques previously discussed fail to compensate a robot's disturbance. Finally, a summary of the chapter is presented in section 5.8.

5.1. Innate Compensatory Measures

As discussed in chapter 4, a robot's performance may be strongly degraded when facing an abnormal situation. In these cases, a robot can generally execute a number of actions in order to improve its performance and continue making progress in its mission. In this thesis, they will be referred to as *compensatory actions* or *compensatory measures*. A special case of compensatory measures inspired by biological instinctive behaviour is introduced in the next subsection. Both, when referring to living creatures or robots, these actions will be presented as *innate compensatory measures* (ICMs) or *innate compensatory actions*.

5.1.1. Biological Inspiration

Biological creatures deal with the same problems a robot may find in unstructured environments. They are successfully able to compensate several different detrimental disturbances due to a repertoire of compensatory measures developed via evolution. Biological ICMs are not learned during a creature's life, but are an innate part of the heritage of its species. The ability to compensate abnormal situations goes beyond purely behavioural aspects, by incorporating physical adaptation. One example of this is the autotomy capability present in over 200 species of animals [53] [5]. These creatures can shed parts of their body in order to escape from a predator. In cases where a leg is severed, they can promptly adapt their gait to the new number of legs. Another example of biological ICM is provided by the work of Walter Hangartner at the University of Zurich [130]. He made experiments with ants following a trail of pheromone. In one experiment, Hangartner removed the left antenna of an ant. Although the ant invariably overcorrected to the right, it had no problem following the trial. This shows that this ant species (Lasius fuliginosus) is able to compensate the loss of an antenna. The immediacy of the creatures' adaptation in the two previous examples suggests that there is no learning process involved.

When talking about ICMs, it may be worthwhile to indicate that anomalies or abnormal situations are faced by members of a species. They are uncommon situations for individuals. However, these may be common or normal situations in terms of the corresponding species. It is precisely because these situations are faced by the species over generations that the slow evolution process is able to compensate them. For instance, it may be common for a species of insects to be hunted by lizards. Consequently, evolution may develop a range of defence techniques in this species so that insects can avoid being eaten by lizards. However, before evolving a defence strategy, repeated encounters between a particular insect of the species and lizards would be uncommon because the insect would probably be eaten on the first encounter.

5.1.2. Innate Compensatory Measure Set Aspects

Biological ICMs can be emulated and incorporated into robots, as a number of action sequences intended for compensating abnormal situations. As the case with biological instinctive behaviour, ICMs are a part of the initial information provided to robots. An ICM set should be as comprehensive as possible. Compensation for general abnormal situations must be incorporated into the set first. Then, compensatory measures for more specific cases can be gradually incorporated when necessary. Important aspects that have to be considered when creating an ICM set are:

- Robots' components that are prone to failure. As more fragile components have a higher probability of being damaged, it is advisable to include suitable ICMs, able to compensate this kind of event, into the ICM set. For example, if a robot's whiskers were thin and not very flexible, it is likely that they are permanently bent or broken. Therefore, a suitable ICM should be considered.

- Mission demands on robots. Depending on the mission a robot is performing, some of the robot's components may be exposed to higher demands. Because these parts are more likely to worn out, appropriate ICMs should be incorporated. For instance, the leg servo motors of the experimental robot are heavily used by the robot when walking. Therefore, the failure of a servo motor should be incorporated into the ICM set.

- Potential hazards of the robots' environment. The elements of an ICM set should include compensation for potential detrimental disturbances robots may encounter in the specific environment where they perform their mission. Here, factors such as terrain features, temperature, humidity, pressure and obstacles should be considered.

5.1.3. A Taxonomy of Innate Compensatory Measures

The classification and labelling of ICMs facilitates the identification and search for suitable ICMs to compensate for a particular disturbance. The following five categories, presented as mutually exclusive attribute pairs (or triads), are utilised during ICM classification:

- Enabled, disabled and in-execution ICMs. Initially, all of the robot's ICMs are enabled and available for use. However, as the robot is damaged or some abnormal situations arise, the requirements demanded by some ICMs may not be satisfied. For instance, the execution of an ICM may require that a hexapod robot has all of its legs. Therefore, if the robot has lost one leg, this ICM cannot be executed. In these cases, the ICM is disabled. A third category, in-execution, corresponds to those ICMs that are currently being executed. These ICMs are not considered by the system when new detrimental disturbances need to be compensated.

- Versatile, specific and component-oriented ICMs. While versatile ICMs can compensate more than one kind of detrimental disturbances, specific ICMs are only able to deal with one type. The use of versatile ICMs reduces the number of elements of the ICM set. As a result, the searching process for suitable compensatory measures is accelerated. Specific ICMs should be included into the ICM set only as a last resort. In addition, component-oriented ICMs are intended to deal with faulty components rather than with other kinds of disturbance.

- Mobile, static and disturbance-oriented ICMs. When compensating one kind of detrimental disturbance, mobile ICMs can be applied to more than one robot component whereas static ICMs can be applied to only one part. This category is more related to the robot's structure than the ICM itself. For instance, the same ICMs could be utilised for releasing a walking robot's trapped leg regardless of which leg is experiencing the disturbance. Then, because the compensatory measure can be applied to more than one of a robot's components experiencing the same disturbance, this is a mobile ICM. However, if the robot only had one remaining leg, the ICM would lose its mobile status and it would become a static ICM. As common components (e.g. legs, light sensors, etc.) usually appear more than once in a robot's structure, the use of mobile ICMs is another strategy for reducing the ICM set. In addition, disturbance-oriented ICMs are intended to deal with abnormal situations arising from the robot's environment rather than with the robot's malfunctions.

- Reversible and irreversible ICMs. Depending on whether or not it is feasible to return to the original state after the application of an ICM, allows a classification into reversible or irreversible, respectively. Commonly, the execution of an irreversible ICM may affect the robot's capabilities necessary for executing other compensatory measures or performing the robot's mission. Therefore, reversible ICMs should be applied first whereas irreversible ICMs should only be used as a last resort. An example of a reversible ICM could be the movement of a walking robot's leg, as it should be feasible for the robot to return its leg to the original position. On the other hand, an ICM could be the shedding of a robot's leg. In this case, the robot may be unable to reattach its leg in an autonomous fashion. Therefore, this ICM would be irreversible.

- Temporary and permanent ICMs. According to the duration of an ICM, it can be classified as temporary or permanent. In general, those ICMs dealing with a robot's hardware failures or a robot's structure damage are permanent. On the other, hand, ICMs compensating for changing features of a robot's environment are temporary. This kind of ICM is assigned with a limited duration, so robots stop executing them once they expire. This is useful for avoiding the continued execution of demanding ICMs in order to compensate for past situations. Once a temporary ICM expires, the robot can evaluate its performance without executing the ICM. If the condition that triggered the ICM in the first place is still present, the robot's performance will be degraded. In this case, the temporary ICM is enabled for a new period. Otherwise, the robot discards it and the ICM is returned to the ICM set.

All ICMs must be classified into one of the previous categories, selecting one attribute by category (as they are mutually exclusive). All of the possible combinations are allowed, except temporary-irreversible or component-oriented – disturbance-oriented pairs. As a result, ICMs can be classified into 69 (or 3^3 2^2 – 3x3x3x1x1 - 3x1x1x2x2) different groups. The ICM taxonomy and allowed combinations of categories are represented by the flow diagram in Fig. 5.1.

5.1.4. Detrimental Disturbance Identification

Before selecting a particular ICM for compensating a detrimental disturbance, the abnormal situation must be correctly identified. In chapter 4 abnormal situations were labelled by means of abnormal situation centroids (ASCs). However, it is unlikely that each time the abnormal situation arises, an exact copy of the corresponding ASC is generated. Therefore, it is necessary to establish what range of

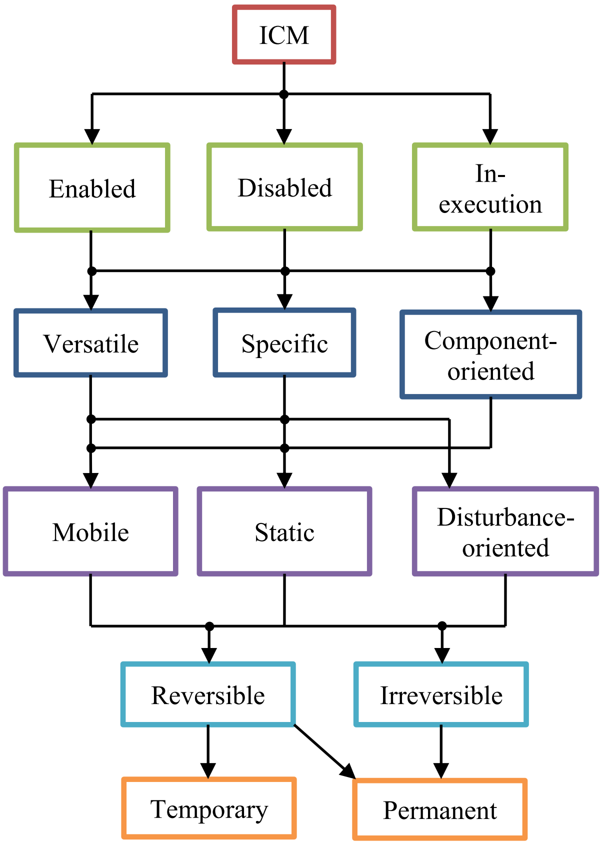


Figure 5.1. ICM Taxonomy Flow Diagram.

relatively similar ASCs will represent the same abnormal situation. This can be determined by means of Eqs. 5.1 and 5.2.

$$ASC^* = \left(\frac{\sum_{k=1}^{E} ASC_1^k}{E}, \frac{\sum_{k=1}^{E} ASC_2^k}{E}, \dots, \frac{\sum_{k=1}^{E} ASC_n^k}{E} \right) \tag{5.1}$$

$$T_{ASC} = \max\left(\sqrt{\left(ASC_1^k - ASC_1^*\right)^2 + \left(ASC_2^k - ASC_2^*\right)^2 + \dots + \left(ASC_n^k - ASC_n^*\right)^2} \right) \tag{5.2}$$

Where

- ASC^* is the centroid of the ASCs corresponding to the E times the robot has experienced the detrimental disturbance.

- T_{ASC} is a threshold that specifies the maximum Euclidean distance an ASC can have and still belong to the detrimental disturbance represented by ASC^*.
- ASC^k is the ASC corresponding to the *k-th* time the robot experienced the detrimental disturbance..
- n is the number of sensors in the robot.

Because only once the detrimental disturbance has been experienced by the robot is it possible to obtain the corresponding ASC, robots must experience the abnormal situation before its corresponding ICM can be incorporated into the system. The ASC value is corrected each time the robot experiences the detrimental disturbance. As a result, the ASC^* calculated by means of Eq. 5.1 is obtained. Then, using Eq. 5.2, the maximum Euclidean distance between ASC^* and all of the ASCs, associated with the same kind of disturbance, is determined. The result is the threshold value T_{ASC}, which will be included into the ICM labels and will be used for matching pairs of ICM-disturbance. As the robot gains more experience with a particular disturbance and both ASC^* and T_{ASC} are updated accordingly, it is more likely that the disturbance will be correctly identified by the system.

5.1.5. Robot Anatomy Labelling

In those situations where, by means of research actions discussed in chapter 4, a detrimental disturbance has been associated with specific parts of the robot, innate compensatory measures (ICMs) corresponding to those components must be selected. This can be done by labelling all of the robot's components. This is equivalent to the labelling system people use for referring to specific parts of their body. However, a numeric system of labelling seems more convenient than words like “lung” or “heart” in this case. In the proposed labelling method, a *robot anatomy label* (RAL), consisting of four elements, is used. The first element indicates if the component is a part of the robot's structure $(RAL_1 = 1)$, the robot's sensors $(RAL_1 = 2)$ or the robot's moving parts $(RAL_1 = 3)$. Then, the second element establishes if the component appears more than once in the robot. Examples of such components are legs, light sensors, etc. Each set of similar components of this kind is identified by using a different RAL_2 value whereas all of the components in the set are labelled with the same RAL_2 value. For instance, all of the robot's legs are labelled with the same RAL_2 value, but the group of legs and the group of light sensors are labelled with different RAL_2 values. Moreover, $RAL_2 = 0$ is the label assigned to all of those components that appear only once in the robot's structure. The third element identifies the component itself and the fourth element indicates the status of this component (operational: $RAL_4 = 1$, temporarily disabled: $RAL_4 = 2$ or permanently disabled: $RAL_4 = 3$). A summary of RAL notation is shown in Table 5.1.

Table 5.1. RAL notation.

RAL Element	Code	Description
Type	RAL_1	1: structure 2: sensors 3: moving parts
Component Set's Tag	RAL_2	0: component appears only once in the robot's structure. A number between 1 and the number of sets of components that appear more than once in the robot's structure. A different number is used for each set.
Component's Tag	RAL_3	If $RAL_2 = 0$: A number between 1 and the number of components with $RAL_2 = 0$ in the robot's structure. A different RAL_3 value is assigned to each component with $RAL_2 = 0$. Otherwise: A number between 1 and the number of components in the set. A different RAL_3 value is assigned to each component in the set.
Status	RAL_4	1: operational 2: temporarily disabled 3: permanently disabled

Tables 5.2 to 5.4 show, as an example, the labelling system utilised with the experimental hexapod robot. The Code column in the tables utilises the same notation introduced in chapter 4 for symbolising the robot's sensors and servo motors. Table 5.2 shows the RAL associated with a robot's leg, which is represented

by X. For example, if X was replaced by 2, then Table 5.2 would contain the RAL corresponding to leg 2.

Table 5.2. Example of RAL for Robot's Components that Appear more than once in the Robot's Structure (Legs).

Component	Code	RAL_1	RAL_2	RAL_3	RAL_4
Leg	Leg_x	1	X	1	1
Position Sensor	Ps_{x1}	2	X	1	1
	Ps_{x2}	2	X	2	1
	Ps_{x3}	2	X	3	1
Force Sensor	Fs_{x1}	2	X	4	1
	Fs_{x2}	2	X	5	1
	Fs_{x3}	2	X	6	1
Tip Leg Force Sensor	TFs_x	2	X	7	1
Servo Motor	S_{x1}	3	X	1	1
	S_{x2}	3	X	2	1
	S_{x3}	3	X	3	1

Table 5.3 shows the labelling of the robot's light sensors and battery level indicators. These sensors appear more than once in the robot's structure but they are not part of a leg. Therefore, they have different RAL_2 values (7 and 8, respectively). The k letter has been used for labelling each light sensor in Table 5.3. Therefore $k = 2$, for instance, is used for labelling the light sensor Ls_2. Similarly, j has been used for labelling each battery level indicator. For example, $j = 3$ is used for labelling the battery level indicator Bl_3.

Table 5.3. Example of RAL for Robot's Components that Appear more than once in the Robot's Structure (Light Sensors and Battery Level Indicator).

Component	Code	RAL_1	RAL_2	RAL_3	RAL_4
Light Sensor	Ls_j	2	7	j	1
Battery Level Indicator	Bl_k	2	8	k	1

Table 5.4 shows the labelling of the robot's components that appear only once in the robot's structure. This is indicated by $RAL_2 = 0$. In addition, the values of RAL_4

have been changed to illustrate the use of this element. In this case, the RAL_4 indicates that the accelerometer has been temporarily disabled $(RAL_4 = 2)$ and the whiskers have been permanently disabled $(RAL_4 = 3)$.

Table 5.4. Example of RAL for Robot's Components that Appear Only Once in the Robot's Structure.

Component	Code	RAL_1	RAL_2	RAL_3	RAL_4
Accelerometer	Acc_x	2	0	4	2
	Acc_y	2	0	5	2
Whiskers	W_s	2	0	6	3

5.1.6. ICM Labelling

In order to identify ICMs and be able to execute them when the corresponding detrimental disturbance arises, an ICM labelling method is utilised. The number of elements of an ICM label depends on the number of abnormal situations and robot's component malfunctions that the ICM is intended to compensate. Table 5.5 shows the notation utilised with ICM label, where the terms introduced in subsection 5.1.3 are utilised.

Table 5.5. ICM Label Notation.

ICM Label Component	Description
Status (s)	0: if the ICM is disabled. 1: if the ICM is enabled. 2: if the ICM is being executed.
Disturbances (d)	0: if the ICM is component-oriented. 1: if the ICM is specific The number of abnormal situations the ICM is intended to compensate if the ICM is versatile.
Components (c)	0: if the ICM is disturbance-oriented. 1: if the ICM is static. The number of faulty robot components the ICM is intended to compensate if the ICM is mobile.
Effects (e)	0: if the effect of the ICM is reversible 1: if the effect of the ICM is irreversible
Duration (t)	0: if the ICM is permanent. The ICM duration in seconds if the ICM is temporary.
RAL_{A1} to RAL_{A4} (Compensation)	The four elements of an ICM associated robot anatomy label. This RAL corresponds to the c robot's components the ICM is intended to compensate.

RAL_{B1} to RAL_{B4} (Requirements)	The four elements of an ICM associated robot anatomy label. This RAL corresponds to the r enabled and functional robot's components required by the ICM in order to be executed.
ASC_1^* to ASC_n^*	The n elements of an abnormal situation centroid representing the d disturbances the ICM is intended to compensate.

The ICM label data frame is shown in Table 5.6.

Table 5.6. ICM Label Data Frame.

Status	Disturbances	Components	Effects	Duration	Requirements
s	*d*	*c*	*e*	*t*	*r*
	↓	↓			↓
T_{ASC}^1	ASC_1^{*1} to ASC_n^{*1}	RAL_{A1}^1 to RAL_{A4}^1			RAL_{B1}^1 to RAL_{B4}^1
T_{ASC}^2	ASC_1^{*2} to ASC_n^{*2}	RAL_{A1}^2 to RAL_{A4}^2			RAL_{B1}^2 to RAL_{B4}^2
⋮	⋮	⋮			⋮
T_{ASC}^d	ASC_1^{*d} to ASC_n^{*d}	⋮			⋮
		RAL_{A1}^c to RAL_{A4}^c			⋮
					RAL_{B1}^R to RAL_{B4}^R

All of the ICM types presented in subsection 5.1.3 can be represented by the ICM label shown in Table 5.6.

5.1.7. Compensation of the Robot's Malfunctions

In cases where abnormal situations are produced by a malfunction in the robot, component-oriented, static or mobile ICMs associated with the faulty parts may be selected. The robot's malfunctions are identified by means of the techniques discussed in chapter 4. Once the faulty part is determined by the system, a suitable ICM must be selected from the ICM set. ICM selection involves an elimination process where four subsets are utilised. This procedure is performed as follows.

- Subset 1 is created by selecting ICMs labels with sections $status = 1$ and $components > 0$, from the ICM set.

- Subset 2 is created by searching for RAL_1 to RAL_3 of the faulty part in the corresponding section of the ICM labels contained in subset 1.
- ICMs in subset 2 are classified into subset 3 and subset 4 according to the *Effects* section of the ICM labels. As a result, reversible ICMs are classified into subset 3 whereas irreversible ICMs are classified into subset 4.
- If subset 3 ≠ Ø, then the ICM with the minimum number of requirements is selected from subset 3. This is the ICM used by the system for compensating the robot's faulty component.
- If subset 3 = Ø, then the ICM with the minimum number of requirements is selected from subset 4. This is the ICM used by the system for compensating the robot's faulty component.

5.1.8. Compensation of Abnormal Situation of the Robot's Environment

Detrimental disturbances generated by the robot's environment can be compensated by disturbance-oriented, specific and versatile ICMs associated with the disturbance. In order to select a suitable ICM from the ICM set, the following procedure is executed.

- Subset 1 is created by selecting ICM labels with sections $status = 1$ and $disturbances > 0$, from the ICM set.
- Subset 2 is created by considering the ASC^* of the disturbance and the list of $ASCs^*$ in the ICM label. Those $ASCs^*$ that are labelling an ICM will be referred to as *target abnormal situation centroid* (TASC). Then, those $ASCs^* - TASC$ pairs that satisfy relation 5.3 will be considered as matching pairs.

$$\sqrt{\left(ASC_1^* - TASC_1\right)^2 + \left(ASC_2^* - TASC_2\right)^2 + \cdots + \left(ASC_n^* - TASC_n\right)^2} \leq T_{ASC} \qquad (5.3)$$

As a result, all of the ICMs that have at least one $ASCs^* - TASC$ matching pair will become an element of subset 2.

- ICMs in subset 2 are classified into subset 3 and subset 4 according to the *Effects* section of the ICM labels. As a result, reversible ICMs are classified into subset 3 whereas irreversible ICMs are classified into subset 4.
- If subset 3 ≠ Ø, then the ICM with the minimum number of requirements is selected from subset 3. This is the ICM used by the system for compensating the robot's environment detrimental disturbance.
- If subset 3 = Ø, then the ICM with the minimum number of requirements is selected from subset 4. This is the ICM used by the system for compensating the robot's environment detrimental disturbance.

5.1.9. Closest ICM Compensation

An alternative to the previous innate compensation methods is the use of closest ICM compensation. This method can be utilised when none of the available ICMs in the ICM set has been selected by the system for compensating a particular detrimental disturbance. The closest ICM compensation consist of finding the minimum distance between the detrimental disturbance experienced by the robot, and the target abnormal situation disturbance-oriented, specific and versatile ICMs available in the ICM set. Furthermore, the distance between any pair of detrimental disturbances can be calculated by means of Eq. 5.4.

$$d_{ASC} = \sqrt{\left(ASC_1^* - TASC_1^S\right)^2 + \left(ASC_2^* - TASC_2^S\right)^2 + \cdots + \left(ASC_n^* - TASC_n^S\right)^2} \qquad (5.4)$$

Where

- ASC^* is the centroid of the detrimental disturbance experienced by the robot.
- $TASC^S$ is the target abnormal situation centroid associated with the *S-th* ICM of the ICM set.

Therefore, the ICM, whose TASC has the minimum distance to the detrimental disturbance experienced by the robot, is selected and executed. If the detrimental disturbance is compensated, both ASC^* and T_{ASC} associated with the abnormal situation are updated. Otherwise, this process can be repeated so the next ICM with minimum d_{ASC} distance is selected and executed. The process can be stopped when a fixed number of ICMs have been tried or the d_{ASC} distance has exceeded a certain threshold.

In general, irreversible ICMs should be avoided when using closest ICM compensation. Otherwise, unsuitable compensatory measures, with associated permanent adverse effects on the robot's capabilities, could be executed.

5.1.10. Post-Compensation Tasks

Once an ICM has been selected and the corresponding detrimental disturbance has been compensated, it is necessary to modify the ICM label accordingly. To begin with, the *status* value of the ICM label must be set to 2. This indicates that the ICM is currently being executed. In cases where the selected ICM is temporary, a monitor process must be executed. Therefore, once the period established in the *duration* section of the ICM label has passed, the ICM execution is stopped. Next, the robot's performance is evaluated. If this is degraded, the compensatory measure is restored for a new period. Otherwise, the ICM is deactivated and the *status* value of its label is set to 1.

The RALs must also be modified if the selected ICM has changed the robot's status or if a faulty component has been identified. Therefore, the RAL_4 elements, associated with the robot's components that the ICM has temporarily disabled, are set to 2. In addition, the RAL_4 elements, corresponding to recently identified faulty robot components, are set to 3.

5.1.11. Reflexes

This kind of compensatory measure is used when a quick response is required. Two main features differentiate reflexes from other types of compensatory actions. Firstly, reflexes are generated by low-level control that has faster communication with the robot's moving parts executing the compensatory actions. Therefore, the robot's central control is not involved in reflex action generation. This is only informed once the reflex action has been executed. Secondly, reflexes aim to reduce the effects of highly detrimental disturbances (pain) in order to avoid robot damage. Consequently, improvement of a robot's performance is not a goal of reflexes as is the case with other kinds of compensatory actions. Nevertheless, by preventing a robot from being damaged, reflexes actions may indirectly improve a robot's performance in the long term. This is true in all of those cases where the damage being avoided by the reflex action has a potential impact on the robot's performance.

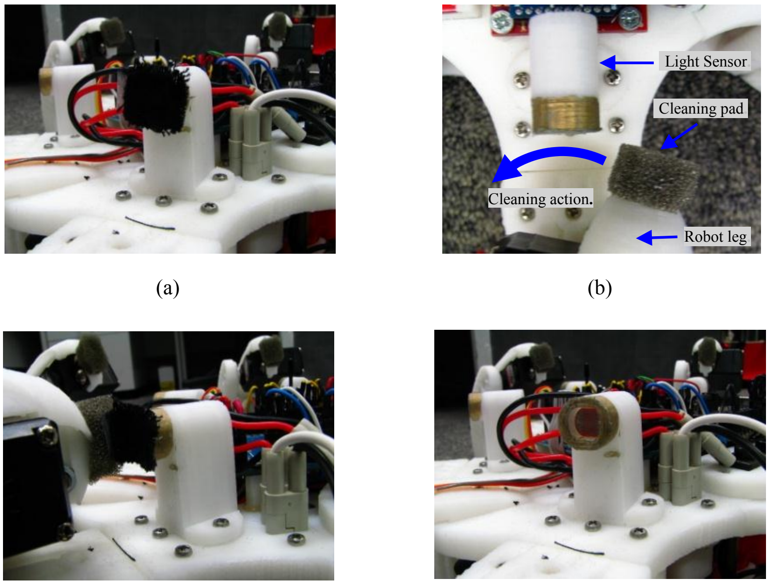
Reflexes are included here as an ICM. However, these are not included into the ICM set, as this will be limited to higher level ICMs. A reflex action must be linked to pain in the robot, so reflex-pain pairs are established. Therefore, every time the robot experiences pain, which will be triggered when the robot's pain threshold is exceeded, the corresponding reflex action will be executed. An example of this was presented in section 4.1.2. Here, the painful experience was an overloaded servo motor. The pain threshold was determined by the amount of time the servo could be overloaded and the reflex action was shutting down the servo and moving the robot to its home position.

5.2. ICM Examples

This section provides examples of innate compensatory measures (ICMs) implemented in the experimental robot. ICMs described in Subsections 5.2.1 to 5.2.9 were programmed into the experimental robot in order to test the correct compensation of their associated disturbances. A complete set of ICMs is not given here, as that would be specifically related to only one robot. Therefore, rather than providing particular solutions to every possible disturbance a robot could experience, this section aims to present a few disturbances and the corresponding ICMs. Some of the ICMs presented here involve the shedding of a leg. This is accomplished by using the leg release mechanism introduced in section 3.4. In addition, more information regarding ICMs adapting the robot's gait to different numbers of legs or changing the direction of the robot's locomotion is available in Appendix B.

5.2.1. Covered Light Sensor

When one or more light sensors are covered by dirt or other obstruction, the experimental robot makes an incorrect estimate of the light source direction. In order to compensate this situation, the robot executes a cleaning action (ICM 1). As illustrated by Fig. 5.2, this consists of a sweeping movement that intends to clean or remove objects covering light sensors. A video showing the robot compensating a covered light sensor by executing a sweeping movement can be watched by following link D05 in Appendix D. If this compensation strategy does not work, then the robot changes its direction of locomotion (ICM 2). This course of action ensures that two adjacent working sensor face the light source. As a result, the light source direction is more accurately calculated. This process is depicted by the flow diagram in Fig. 5.3.



(a) (b)

(c) (d)

Figure 5.2. (a). Covered Light Sensor. (b) Robot' Leg Sweeping the Covered Light Sensor. (c). Close Up of Robot's Leg Sweeping the Covered Light Sensor. (d) Uncovered Light Sensor.

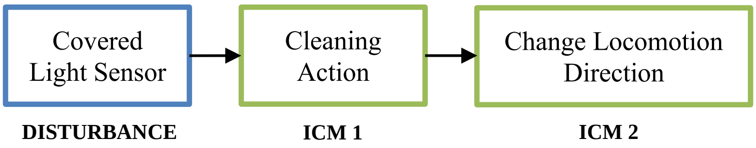


Figure 5.3. Covered Light Sensor ICMs.

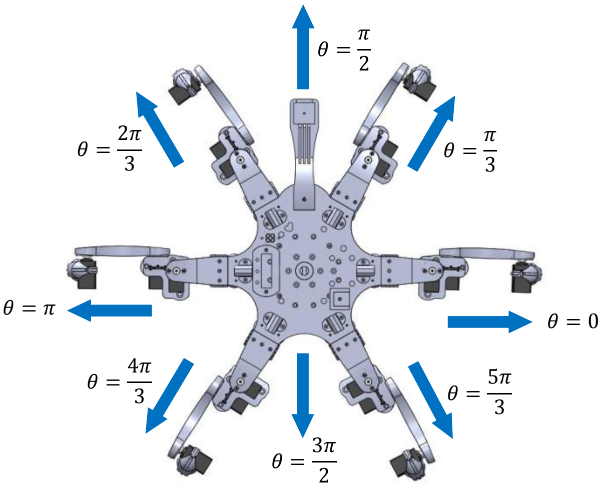

$\theta = \frac{\pi}{2}$
$\theta = \frac{2\pi}{3}$
$\theta = \frac{\pi}{3}$
$\theta = \pi$
$\theta = 0$
$\theta = \frac{4\pi}{3}$
$\theta = \frac{5\pi}{3}$
$\theta = \frac{3\pi}{2}$

Figure 5.4. Robot Heading Direction as a Function of the θ Angle.

The $0 \leq \theta < 2\pi$ angle in Fig. 5.4 represents the heading direction with respect to the body of the robot. For instance, $\theta = 0$ and $\theta = \pi$ mean that the robot is walking sideways, to the right and left, respectively. $\theta = \pi / 2$ indicates the robot is walking forward and $\theta = 3\pi / 2$ the robot is walking backwards. Therefore, when a light sensor is not working properly, the robot can adjust the θ angle, so the heading direction is the middle point between two adjacent working light sensors.

Two videos showing the robot changing its direction of locomotion for compensating one or more covered light sensor can be found by following links D06 and D07 in Appendix D. The video in D06 shows the robot walking backwards $(\theta = 3\pi / 2)$ whereas the video in D07 shows the robot walking diagonally $(\theta = 2\pi / 3)$.

5.2.2. Soft Terrain

The hexapod robot’s legs may sink in when walking over soft terrain such as snow, mud, grass or sand. As a result, the swing legs may not be able to move forwards, which may impede the robot’s locomotion. The ICM utilised by the experimental robot in this situation is to increase the elevation of the legs performing the swing phase. This ICM is depicted by the flow diagram in Fig. 5.5.

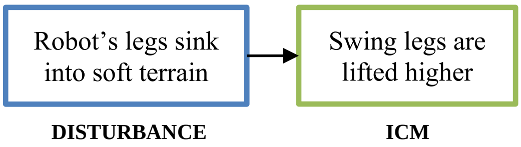

Robot’s legs sink into soft terrain
Swing legs are lifted higher
DISTURBANCE
ICM

Figure 5.5. Soft Terrain ICM.

A video showing the experimental robot walking over soft terrain and compensating this abnormal situation can be found by following link D08 in Appendix D.

5.2.3. Low Power

The experimental robot used in this investigation does not incorporate self-recharging capabilities. Therefore, when the robot detects that its power source is low, it sends a distress signal asking for help. This ICM is illustrated by the flow diagram in Fig. 5.6.

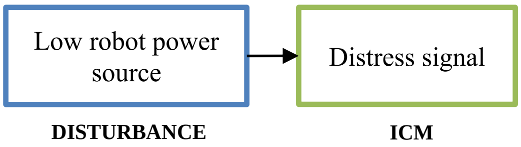


Figure 5.6. Low Power ICM.

A video showing the experimental robot performing the low power innate compensatory measure can be found by following link D09 in Appendix D.

5.2.4. Disconnected Servo Motor PWM Signal

When damage results in a servo motor PWM signal being interrupted, it is not possible to control that motor anymore. This situation would have the same effect as other damage to a servo including burnt out electronics, seized motor, stripped gears, etc. As a result, the robot's gait is not correctly executed and the robot may fall over. When servo 1 or 2 is disconnected, the leg with the disconnected motor may interfere with adjacent legs. This situation further degrades the robot's gait performance. There are two ICMs responsible for compensating this detrimental disturbance. When the PWM signal of servo 3 is disconnected, the robot may limp. Otherwise, if this does not work or the PWM signal of servos 1 or 2 is disconnected, the robot must shed the affected leg and change the gait according to the new leg configuration. This process is illustrated by the diagram in Fig. 5.7.

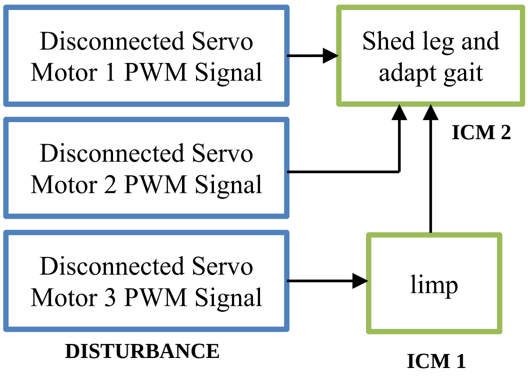

Disconnected Servo Motor 1 PWM Signal
Disconnected Servo Motor 2 PWM Signal
Disconnected Servo Motor 3 PWM Signal
DISTURBANCE
Shed leg and adapt gait
ICM 2
limp
ICM 1

Figure 5.7. Disconnected Servo Motor PWM Signal.

5.2.5. Trapped Leg

If a leg has become entangled, trapped or is held by some object in the robot's environment, the experimental robot would be unable to make further progress towards the light source. In this case, the robot will try to free the leg by making a series of movements. If this does not work, the robot will shed the trapped leg and change its gait accordingly. This situation is represented by the diagram in Fig. 5.8.

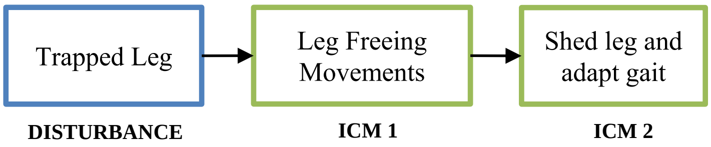


Figure 5.8. Trapped Leg ICM.

A video showing the shedding of a leg can be watched by following link D03 in Appendix D. In addition, a video showing the experimental robot walking with 5 legs can be found by following link D10 in Appendix D.

5.2.6. Frontal Collision

A frontal collision is detected by abnormal readings from the robot’s whiskers. In this case, the robot moves backwards, turns and advances. This ICM is illustrated by the diagram in Fig. 5.9.

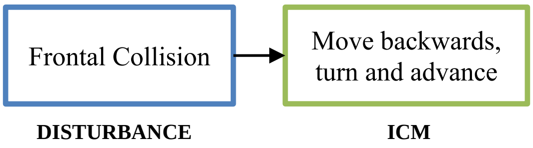


Figure 5.9. Frontal Collision ICM.

A video showing the experimental robot performing the frontal collision innate compensatory measure can be found by following link D11 in Appendix D.

5.2.7. Broken Leg

Depending on where the robot’s leg has been broken, the experimental robot may be able to walk by lowering the broken leg further during the stance phase. If the broken leg is able to make contact with the ground, then it is not necessary to make additional modifications to the robot’s gait. Otherwise, if the robot cannot make contact with the ground and the robot loses stability when walking, the robot must shed the leg and adapt its gait to the new leg configuration. This process is depicted by means of the diagram in Fig. 5.10.

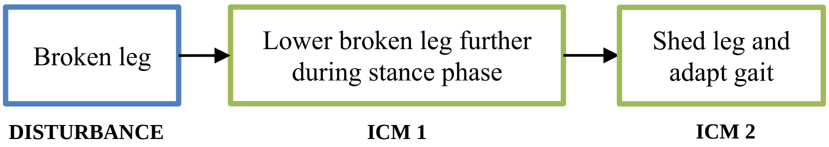


Figure 5.10. Broken Leg ICM.

5.2.8. Lights Off

This disturbance occurs when the robot does not have any light source to find, so it cannot make progress in its mission. In this case, the robot assumes that its light sensors are covered and proceeds with the cleaning actions shown in Fig. 5.3b. If this does not work, the robot waits until a light source is switched on. This ICM is illustrated by means of the flow diagram in Fig. 5.11.

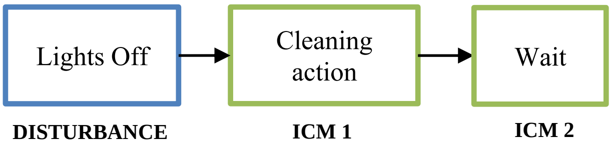


Figure 5.11. Lights Off ICM.

A video showing the experimental robot performing the lights off innate compensatory measure can be found by following link D12 in Appendix D.

5.2.9. Abnormal Accelerometer Readings

This ICM considers two causes of abnormal accelerometer readings in cases where the robot’s performance has been degraded. First, the robot could be walking over a steep slope that prevents the robot from approaching the light source. On the other hand, there could be a mass added to the robot or a shift in the location of the robot’s battery or other cargo. This could unbalance the robot and prevent it from walking. The former disturbance can be compensated by returning the robot to the location where the terrain’s slope was normal and taking a different path from there towards the light source. The compensation of the latter disturbance can be achieved by tilting the robot in order to equalise its leg tip force sensor readings. This compensation approach is represented by the diagram in Fig. 5.12

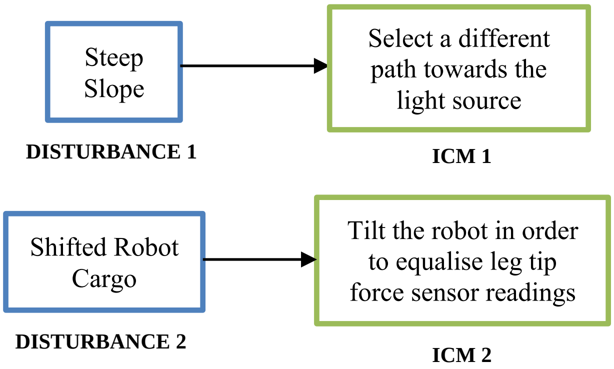

Steep Slope
Select a different path towards the light source
DISTURBANCE 1
ICM 1
Shifted Robot Cargo
Tilt the robot in order to equalise leg tip force sensor readings
DISTURBANCE 2
ICM 2

Figure 5.12. Steep Slope and Shifted Robot Cargo ICMs.

5.3. Compensatory Measures in the Identification of Detrimental Disturbance Sources

Chapter 4 introduced techniques for the identification of detrimental disturbance sources. Once these methods have been applied, a hierarchical list of possible abnormal situation causes is generated. However, the system can only ensure that the source of an anomaly has been found once the associated compensatory actions have been executed successfully. Consequently, if this compensatory measure is able to restore the robot's normal performance, it means that the abnormal situation source was correctly determined. If this is not the case, there are two possible options.

1. The detrimental disturbance source is not on top of the hierarchical list of possible abnormal situation causes.
2. The detrimental disturbance source is on top of the hierarchical list of possible abnormal situation causes, but the associated compensatory measure was unable to restore the normal robot's performance.

Initially, the system assumes that the first alternative will correct the problem. Consequently, if compensatory actions associated with the first detrimental disturbance source on the hierarchical list fail, then the robot executes compensatory measures for the second possible detrimental disturbance on the list. This process continues until the disturbance is compensated or there are no more elements on the list of detrimental disturbance source candidates. This is illustrated by Fig. 5.13.

If the second alternative is true, it means that the method has been unable to compensate the abnormal situation. In this case, the system can recourse to closest ICM compensation. If this also fails, then compensatory measure can be found by using one of the methods discussed in sections 5.4 and 5.5.

5.4. Autonomously Generated Compensatory Measures

Although biological innate compensatory measures cover a broad range of detrimental disturbances, they do not address every possible situation a living creature may find during its life. In general, they fail to compensate relatively new

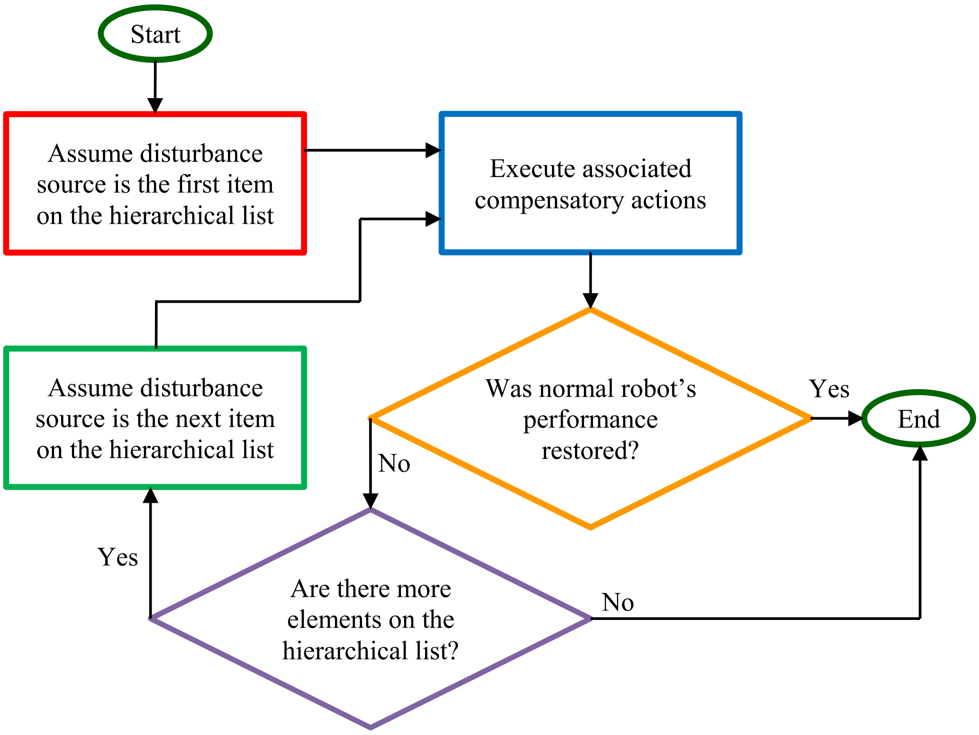

Start
Assume disturbance source is the first item on the hierarchical list
Execute associated compensatory actions
Assume disturbance source is the next item on the hierarchical list
Was normal robot's performance restored?
Yes
End
No
Yes
Are there more elements on the hierarchical list?
No

Figure 5.13. Flow Diagram of Compensatory Measure Selection

detrimental situations or anomalies that affect only a few members of the species. For instance, innate compensatory measures (ICMs) may be unable to compensate abnormal situations that do not belong to a species' habitat. That is the reason why living creatures may die when they are brought into different environments. ICMs may also be quite inefficient when compensating abnormal situations generated by accidents or birth defects.

In order to compensate those detrimental disturbances not covered by ICMs, robots must be provided with techniques for the autonomous generation of compensatory measures. The method presented in this thesis uses available information about the robot's status, robot's anatomy and the ASC^* describing the abnormal situation. These data together with a set of significant actions are utilised for the autonomous generation of compensatory actions.

The theory behind the autonomous generation of compensatory measures will be introduced next. However, these techniques were not implemented into the experimental robot as it was considered to be beyond the scope of the thesis. Therefore, they will be proposed as future work.

5.4.1. Robot's Moving Part Selection

In order to determine which of the robot's moving parts will execute the compensatory measures different criteria may be utilised. The following methods have been considered in this research.

- Hierarchical weight list

Those of the robot's moving parts that are physically closer to the sensors providing abnormal readings, can be determined and labelled by means of Eq. 4.14. As a result, each moving part is assigned a weight. These weights are directly proportional to the physical distance between the robot's moving parts and the sensors providing abnormal readings. Therefore, a hierarchical weight list can be generated by ordering the robot's moving parts according to their associated weight. This list can be utilised for the generation of the robot's moving part selection sequences (RMPSSs). Eq. 5.5 shows a simple RMPSS where the robot's moving parts are selected individually according to their position in the weight list.

$$seq_1 = \{M_1, M_2, \ldots, M_L\} \tag{5.5}$$

Where

- $M_1, M_2, \ldots, M_L$ are the robot's L moving parts ordered from the greatest to the least associated weight.

When selecting a robot's moving parts according to seq_1, only one part executes a compensatory action at any particular time. The selection process continues until the disturbance has been compensated, all of the robot's moving parts have executed their full range of compensatory measures without compensating the disturbance, or a threshold has been exceeded. This threshold (T_W) can be calculated by means of Eq. 5.6.

$$T_W = \delta M_1 \tag{5.6}$$

Where

- $0 \le \delta \le 1$ is the normalised percentage of M_1.
- M_1 is the greatest weight associated with the robot's moving parts.

If T_W is used during the robot's moving part selection process, only parts whose associated weight is greater than T_W execute the compensatory measures.

The robot's moving parts could also execute compensatory measures in pairs. seq_2 in Eq. 5.7 is a sequence of this type.

$$seq_2 = \{(M_1, M_2), (M_2, M_3), \ldots, (M_{L-1}, M_L)\} \tag{5.7}$$

On the other hand, seq_3 in Eq. 5.8 shows a different association of the selected robot's moving part pairs.

$$seq_3 = \{(M_1, M_2), (M_3, M_4), \ldots, (M_{L-1}, M_L)\} \tag{5.8}$$

In general, different associations and numbers of elements considered in each combination of the robot's moving parts can be used. These sequences can be generated by using three parameters, as represented in Eq. 5.9.

$$seq_W = \{N, D, L\} \tag{5.9}$$

Where

- N is the number of selected robot moving parts in each combination.
- D is the distance between the maximum element indexes of two adjacent combinations.
- L is the number of robot moving parts.

Examples of the notation used in Eq. 5.9 are provided in Table 5.7.

Table 5.7. Notation Example of Robot Moving Part Selection Sequences (Hierarchical Weight List).

N	D	L	seq_W
1	1	7	$\{M_1, M_2, M_3, M_4, M_5, M_6, M_7\}$
2	1	6	$\{(M_1, M_2), (M_2, M_3), (M_3, M_4), (M_4, M_5), (M_5, M_6)\}$
2	2	8	$\{(M_1, M_2), (M_3, M_4), (M_5, M_6), (M_7, M_8)\}$
3	1	6	$\{(M_1, M_2, M_3), (M_2, M_3, M_4), (M_3, M_4, M_5), (M_4, M_5, M_6)\}$
3	3	12	$\{(M_1, M_2, M_3), (M_4, M_5, M_6), (M_7, M_8, M_9), (M_{10}, M_{11}, M_{12})\}$
4	2	8	$\{(M_1, M_2, M_3, M_4), (M_3, M_4, M_5, M_6), (M_5, M_6, M_7, M_8)\}$
4	4	12	$\{(M_1, M_2, M_3, M_4), (M_5, M_6, M_7, M_8), (M_9, M_{10}, M_{11}, M_{12})\}$

Therefore, by modifying N , D and δ , the system is able to autonomously generate a set of RMPSSs.

- Hierarchical weight list and the robot's anatomy

Information about the robot's anatomy can be used in combination with the hierarchical weight list in order to determine which of the robot's moving parts will execute the compensatory measures. This kind of RMPSS is represented by means of expression 5.10.

$$seq_{WA} = \begin{Bmatrix} (M_1, M_{11}, M_{12}, \ldots, M_{1D1}), (M_2, M_{21}, M_{22}, \ldots, M_{2D2}), \ldots \\ \ldots, (M_L, M_{L1}, M_{L2}, \ldots, M_{LDL}) \end{Bmatrix} \tag{5.10}$$

Where

- $(M_{i1}, M_{i2}, \ldots, M_{iDi})$ is an ordered list of the Di closest robot moving parts to M_i.
- Di is the number of the robot's moving parts whose distance from M_i is less than or equal to a threshold value (T_{WA}).

The T_{WA} value establishes the maximum distance that two moving parts can have and still be associated together. This value depends on the specific anatomy of each robot.

RMPSSs that consider both hierarchical weight lists and the robot's anatomy can be generated by using $L+1$ parameters. This is represented by means of expression 5.11.

$$seq_{WA} = \{D_1, D_2, \ldots, D_L, L\} \tag{5.11}$$

If we consider that the $D_1, D_2, \ldots, D_L$ parameters are calculated by the system as a function of T_{WA}, then seq_{WA} can be represented by means the following expression:

$$seq_{WA} = \{T_{WA}, L\} \tag{5.12}$$

Examples of the notation used in Eq. 5.11 are provided in Table 5.8.

Table 5.8. Notation Example of Robot Moving Part Selection Sequences (Hierarchical Weight List and Robot's Anatomy).

$D_1, D_2, \ldots, D_L$	L	seq_{WA}
$1,2,3$	3	$\{(M_1,M_{11}),(M_2,M_{21},M_{22}),(M_3,M_{31},M_{32},M_{33})\}$
$2,2,2$	3	$\{(M_1,M_{11},M_{12}),(M_2,M_{21},M_{22}),(M_3,M_{31},M_{32})\}$
$1,1,1,1,1$	5	$\{(M_1,M_{11}),(M_2,M_{21}),(M_3,M_{31}),(M_4,M_{41}),(M_5,M_{51})\}$
$5,2$	2	$\{(M_1,M_{11},M_{12},M_{13},M_{14},M_{15}),(M_2,M_{21},M_{22})\}$
$3,2,1,0$	4	$\{(M_1,M_{11},M_{12},M_{13}),(M_2,M_{21},M_{22}),(M_3,M_{31}),M_4\}$
$1,0,3,0,2,0$	6	$\{(M_1,M_{11}),M_2,(M_3,M_{31},M_{32},M_{33}),M_4,(M_5,M_{51},M_{52}),M_6\}$
$0,0,3,3,1$	5	$\{M_1,M_2,(M_3,M_{31},M_{32},M_{33}),(M_4,M_{41},M_{42},M_{43}),(M_5,M_{51})\}$

The T_W threshold introduced in the hierarchical weight list can also be used in this selection method. As a result, only M_i parts whose corresponding weight is greater than T_W, and their associated close parts $(M_{i1}, M_{i2}, \ldots, M_{iDi})$, execute the compensatory measures. The modification of T_W and T_{WA} parameters allows the system to autonomously generate new RMPSSs.

5.4.2. Autonomous Action Generation

Robotic actions are defined by a series of parameters. For instance, a robot's tasks are commonly executed by motors whose angle of rotation and speed can be modified. The entire repertoire of robot actions can be controlled by changing these values. At the lowest control level, the compensation method proposed in this subsection generates different combinations of parameter values. Then, the actions associated with these parameters are executed and their effects are monitored. Initially, the only restriction imposed on the generated actions is that they must not damage the robot. Therefore, movements are constrained by the robot's workspace and "painful" experiences are avoided.

If a robot's mission is divided into a number of steps, then the robot's moving part parameter values can be used for describing the robot's action status at each step. This can be represented by means of the parameter matrix P_t in expression 5.13.

$$P_t = \begin{bmatrix} p_{11}^t & p_{12}^t & \cdots & p_{1F1}^t \\ p_{21}^t & p_{22}^t & \cdots & p_{2F2}^t \\ \vdots & \vdots & \vdots & \vdots \\ p_{L1}^t & p_{L2}^t & \cdots & p_{LFL}^t \end{bmatrix} \tag{5.13}$$

Where

- p_{xy}^t is the value of the *y-th* parameter associated with the robot's moving part M_x at the instant t.
- L is the number of the robot's moving parts.
- Fx is the number of parameters associated with the robot's moving part M_x. Because the Fx value varies for different values of x, not all of the columns in P_t have the same number of columns. Therefore, those columns with fewer elements are filled with a number that is forbidden for the rest of the P_t elements.

In order to generate different actions, the elements of the parameter matrix must be changed. These changes can be generated by applying increments to each parameter. The increments used can be different for each parameter, and can be expressed by means of the step size matrix in 5.14.

$$S_p = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1F1} \\ S_{21} & S_{22} & \cdots & S_{2F2} \\ \vdots & \vdots & \vdots & \vdots \\ S_{L1} & S_{L2} & \cdots & S_{LFL} \end{bmatrix} \tag{5.14}$$

In addition, the range of each parameter can be represented by means of the range matrix in expression 5.15.

$$R_p = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1F1} \\ R_{21} & R_{22} & \cdots & R_{2F2} \\ \vdots & \vdots & \vdots & \vdots \\ R_{L1} & R_{L2} & \cdots & R_{LFL} \end{bmatrix} \tag{5.15}$$

A robotic action can be defined as the series of parameter matrices corresponding to that action. This is expressed by means of Eq. 5.16.

$$action = \{P_1, P_2, \ldots, P_m\} \tag{5.16}$$

Here, m is the minimum number of parameter matrices required for describing the action. As the m value is incremented, it is possible to represent more complex actions by means of expression 5.16.

Therefore, the full range of robotic actions can be generated by means of the following algorithm.

```
1. Set $m = 1$
2. Initialise $P_1, P_2, \ldots, P_m$.
3. Set $x = 1$, $y = 1$ and $z = 1$.
4. If the $P_1, P_2, \ldots, P_m$ action is harmless for the robot, then execute the action.
5. If abnormal readings were compensated, then stop.
6. Increment $p_{xy}^{z}$ by $S_{xy}$.
7. If $p_{xy}^{z} > R_{xy}$ then initialise $p_{xy}^{z}$ and increment $z$ by 1. Otherwise, go to step 3.
8. If $z > m$ then set $z = 1$ and increment $y$ by 1.
9. If $y > Fx$ then set $y = 1$ and increment x by 1.
10. If $x > L$ then increment $m$ by 1 and go to step 2.
11. Go to step 6.
```

A maximum iteration time can be assigned to the previous algorithm in order to stop it when no compensatory measure is found.

5.4.3. Compensation with Malleable Actions

An alternative to the method introduced in subsection 5.4.2 is to provide robots with a set of malleable actions and parameters. The kind of action to be generated is very dependent on the particular robot executing it. However, the same restriction imposed on the generated actions applies here. That is to say, executed actions must be non-damaging for the robot.

Malleable actions can be distinguished by the number of moving parts executing the actions and the associated parameters. For instance, a malleable action could involve a leg of a walking robot and the associated parameters could be the angles of the leg servo motors. Another malleable action could involve a circular movement of the leg, and the associated parameters could be the radius and the (x, y) centre coordinates of the circle described by the leg. Higher-level malleable actions could involve robot locomotion. For instance, a malleable action could be robot walking and the associated parameters could be direction of locomotion, type of gait, and the height reached by the legs during their swing phase.

When using this technique, robots can modify the parameters associated with malleable actions in order to generate a larger action set. Then, by executing these tasks, the robots could eventually compensate a detrimental disturbance. In order to determine if the abnormal situation has been compensated, robots should monitor the effects of the executed actions on the abnormal robot sensor readings.

Malleable actions can be described by expressions 5.13 to 5.16 and generated by the algorithm introduced in the previous subsection. However, in this case not all of the robot's moving parts are necessarily utilised and the action parameters have a different interpretation. In addition, as malleable actions are generated by means of a constant number of steps, the m value in the algorithm is not incremented.

The use of malleable actions seems to correspond with the instinctive behaviour of biological organisms. Here, innate actions are adapted and utilised in the compensation of abnormal situations. For instance, as Konrad Lorenz observed [131], the jackdaw has an instinctive nest-building behaviour that consists of selecting twigs (malleable action) and incorporating them into a foundation. Initially the bird selects any kind of twigs and, as a result of the selection of weak twigs, its nest collapses (disturbance). Then, the jackdaw tries different types of twigs until it discovers a suitable type, one that lodges firmly and does not break. From that moment on, the bird selects only that type of twig for the construction of its nest (compensatory measure).

5.5. Last Resort Measures

5.5.1. Monitoring Sleep Mode

When other compensatory measures have been unable to compensate a detrimental disturbance, robots can still take a nap and wait for better times. During the nap period, a robot enters into sleep mode. As a result, the robot's energy consumption is minimised and a periodic evaluation of the detrimental disturbance is established. For instance, as was mentioned before, the mission of the experimental robot used in this research is to localise a single light source. Therefore, when the robot is in complete darkness a detrimental disturbance is detected. The robot initially assumes that something is covering its light sensors. However, if after a reasonable period none of the compensation methods previously discussed in this chapter work, the robot stops trying. Instead of wasting energy with random action sequences, the robot takes a compensatory nap. The robot enters into sleep mode in order to minimise its energy consumption and evaluates its light sensors regularly. Then, when a light source is detected, the robot still has energy to continue making progress in its mission.

5.5.2. SOS

This is a last resort measure that is only used when all of the previous compensatory methods have failed. The SOS technique simply consists of sending a distress signal. As a part of this signal, relevant information should be transmitted. For instance, a robot could send its current location and an ASC^* indicating its status.

5.6. Learned Compensating Measures

Autonomously generated actions that are able to compensate a detrimental disturbance are included into the ICM set. Therefore, these learned behaviours can be used in possible future occurrences of the abnormal situation. As a result, more experienced robots will be able to compensate more quickly to a broader range of anomalies. Although the theory behind learned compensating measures is presented next, these methods have not been incorporated into the experimental robot.

In order to include a learned compensatory measure (LCM) into the ICM set, this must be autonomously labelled by the robot. The label structure associated with autonomously generated compensatory measures is shown in Table 5.9.

Table 5.9. LCM Label.

Status	Disturbances	Components	Effects	Duration	Requirements
2	1	0	0	*t*	*r*
	↓				↓
T^1_{ASC}	ASC^{*1}_1 to ASC^{*1}_n				RAL^1_{B1} to RAL^1_{B4}
					RAL^2_{B1} to RAL^2_{B4}
					⋮
					⋮
					⋮
					RAL^R_{B1} to RAL^R_{B4}

The values assigned by the robot to each label category are explained as follows.

Status: Learned compensatory measures are generated once autonomously generated actions have been able to compensate a disturbance. Only then, the robot detects that a suitable compensatory measure has been generated and the system begins the labelling of the compensatory measure. Because the compensatory measure will be in execution when labelled, the label status value is 2.

Disturbances: In this work, it will be assumed that compensatory measures associated with faulty components will be included in the original ICM set. Hence, LCM will be classified as disturbance-oriented. Initially, the LCM will be able to compensate only one disturbance. Consequently, the disturbance value in the label is set to 1 and the ASC associated with the compensated disturbance is attached to the label. The T_{ASC} threshold value is set to a predetermined initial value. An alternative is to use the mean value of T_{ASC} thresholds corresponding to other disturbances in the ICM set.

Components: the value of this category is set to 0 because it is assumed that the LCM is disturbance-oriented.

Effects: Because autonomously generated compensatory measures must not damage the robot, all of them have reversible effects. Hence, the value of this category is set to 0.

Duration: LCMs are intended for compensating abnormal situations generated in a robot's environment. In order to check if the anomaly is no longer affecting the robot's performance, the LCM execution should be interrupted from time to time. Therefore, if the robot's performance is degraded when the LCM is interrupted, the system executes the LCM for a new period. Otherwise, the status LCM label category is set to 1 and the LCM is not executed until the disturbance arises again. Because the LCM is temporary, the duration category in the LCM label is set to *t* (number of seconds the LCM should be in execution during each period).

Requirements: This category is completed with the anatomy label corresponding to each moving part involved in the LCM execution. In addition, the *r* value in the category is set to the number of participating moving parts.

5.7. Overview of Identification and Compensation Methods

The flow diagram in Fig. 5.14 illustrates the detection, identification and compensation of the detrimental disturbances introduced in section 4.4. In that section, the experimental robot was exposed to two abnormal situations: the

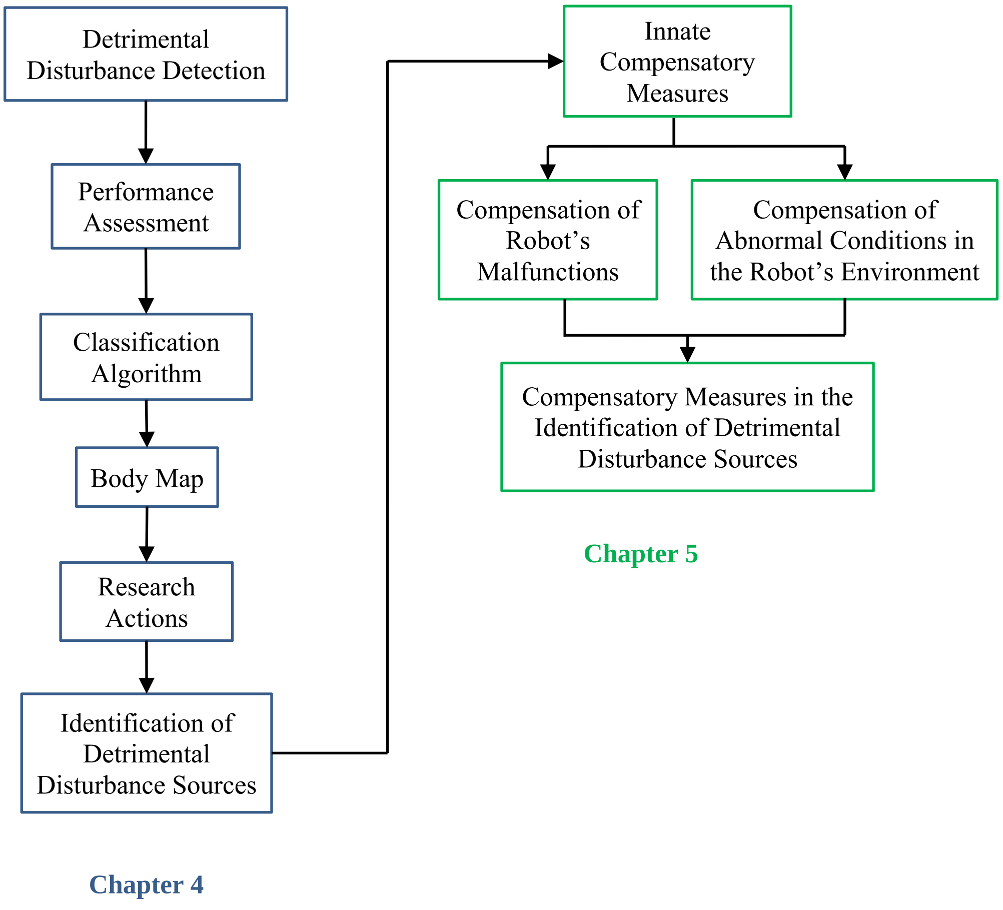

Detrimental Disturbance Detection
Performance Assessment
Classification Algorithm
Body Map
Research Actions
Identification of Detrimental Disturbance Sources
Chapter 4
Innate Compensatory Measures
Compensation of Robot's Malfunctions
Compensation of Abnormal Conditions in the Robot's Environment
Compensatory Measures in the Identification of Detrimental Disturbance Sources
Chapter 5

Figure 5.14. Flow Diagram of Detrimental Condition Detection and Compensation of Disconnected PWM Signal from Servo Motor and Robot Walking over Soft Terrain.

disconnection of the PWM signal of a leg servo motor and walking onto a sponge that simulated soft terrain.

The blue blocks of the flow diagram show the steps described in chapter 4 for the detection and identification of the abnormal situations. First, the robot performance was monitored. Then, an abnormal situation was detected due to a lack of progress in the robot's mission. In the experiment where the PWM servo control signal was disconnected, the affected leg was unable to move to the position required by the robot's gait. This produced problems due to lack of stability and interference between the dysfunctional leg and the fully operational legs. In the case where the robot walked onto soft terrain, the two legs that step onto the sponge were unable to move during their swing phase. As a result, those legs acted like a pivot, which prevented the robot's straight line locomotion. In both experiments, the robot was unable to consistently walk towards the light source and the corresponding abnormal situation was detected as a lack of progress in the robot's mission.

After an anomaly was detected, this was classified as the centroid of the sensors readings provided by the robot during the detection of the detrimental disturbance. Then, the robot used information about the robot's anatomy in order to generate a hierarchical list of possible sources of the detrimental disturbance. These theories were corroborated or rejected by means of a series of research actions. In the experiment described in subsection 4.4.1, the system identified the faulty servo motor with the first research action on the list. By contrast, a possible source of the disturbance affecting the experimental robot in subsection 4.4.2 was identified by the second research action on the list.

Both detrimental disturbances described in section 4.4 were considered as being in the Innate Compensatory Measure Set. Therefore, no autonomously generated compensatory measures were required. The system used the information provided by research actions and cumulative abnormality labels in order to find a suitable compensatory measure. In the case where the PWM servo motor control signal was disconnected, the system identified the disconnected servo motor as a possible source of the disturbance. Therefore, the component-oriented compensatory measure described in subsection 5.2.4 was executed. After executing the corresponding compensatory measure, the robot performance was evaluated. The progress in the robot mission was satisfactorily improved. Hence, the system assumed that the abnormal situation was compensated.

In the experiment where the robot walks over soft terrain, the first theory about the possible source of the disturbance was servo 1 of leg 3. In this case, the system searched for suitable compensatory measures corresponding to this robot component. However, after executing the corresponding component-oriented compensatory measure the robot's performance was not improved. In this case, the system could either consider a second theory about the detrimental disturbance source or verify if the abnormal situation was generated by conditions in the robot's environment. Because the robot moving part weight associated with servo 1 of leg 3 was considerable larger than the weights corresponding to other parts of the robot, the system opted for the latter. Therefore, the cumulative abnormality label was used for searching for corresponding disturbance-oriented compensatory measures. This time a suitable compensatory measure (described in subsection 5.2.2) was found and the robot performance was improved. As a result, the system assumed that the abnormal situation source was correctly identified.

5.8. Summary

The aim of this chapter has been the presentation of techniques that allow robots to emulate biological capabilities such as the use of instinctive and learned behaviour in the compensation of detrimental disturbances. Chapter 5 has presented a number of self-compensation techniques for robots experiencing detrimental disturbances.

Firstly, biological inspired innate compensatory measures (ICMs) were discussed in section 5.1. This method utilises an ICM set, where each element is represented by means of a label. The ICM set is part of the initial information provided to robots and consists of a series of measures intended to compensate expected disturbances that robots may experience. The ICM technique matches abnormal situation centroids (ASC), introduced in chapter 4, with their corresponding compensatory measures in the ICM set. In addition, by labelling components such as sensors, moving parts and pieces of the robot's structure, this method was able to associate a robot's failures with their corresponding ICM. In cases where no suitable ICMs were found, the closest ICM compensation technique was utilised. This method searched for ICMs that are not intended for compensating exactly the same disturbance a robot may be currently experiencing, but that were designed for similar situations.

Examples of ICMs implemented in the experimental robot used in this investigation were presented in section 5.2. Here, external links with videos of the experimental robot performing some of the compensatory measure were provided. Following this, the identification of detrimental disturbances presented in chapter 4 was completed in section 5.3. Here, the theories developed by a robot about the possible source of a detrimental disturbance were rejected or corroborated. The robot executed ICMs corresponding to each disturbance source candidate. As a result, when the disturbance was compensated, the robot assumed that the correct detrimental disturbance source was found.

Section 5.4 introduced a method for the autonomous selection of moving parts and generation of actions that can eventually compensate detrimental disturbances. A general method of action generation was presented first. Then, a more robot-related technique that could accelerate the search for suitable compensatory measures was proposed. Techniques discussed in this section were not implemented in the experimental robot. This implementation will be proposed as future work.

Section 5.5 provided robots with last resort measures. These can be used when the previous methods introduced in this chapter are not able to compensate a particular detrimental disturbance. Techniques discussed in this section were not implemented in the experimental robot. This implementation will be proposed as future work.

Finally, section 5.6 presented methods for the autonomous generation of compensatory measures. These techniques intend to emulate capabilities of biological creatures, which are able to modify their instinctive behaviour as a result of a rewarding experience or punishment. This new behaviour is also referred to as learned behaviour.

Chapter 6
Conclusions

This chapter summarises the main findings of this investigation and provides recommendations for further work.

6.1. Summary of Work

The contributions of this research can be classified into the areas of self-diagnosis, self-compensation and mechatronics in robotics. Findings related to each of these areas are discussed as follows.

6.1.1. Self-Diagnosis in Robotics

This research has developed a method where abnormal situations are detected either as a lack of progress in the robot's mission or pain. The latter is identified as a large difference between expected and current sensor readings, which can often generate robot malfunction if not promptly compensated. In order to detect degraded robot performance, a quantitative measure of the robot's progress is required. Therefore, the simplicity of this technique strongly depends on how difficult it is to calculate this value, which is related to the type of robot mission and sensors. When applied to the experimental robot used in this research, abnormal situations were consistently detected in real-time.

Abnormal situations were classified by using a predetermined number of samples consisting of all of the sensorial information provided by the robot. The centroid of these sensor readings was then utilised for labelling the corresponding abnormal situation. This method has been demonstrated to be effective in the classification of anomalies evidenced by either a single sensor or a combination. Although this technique has been applied to a walking robot, as most (if not all) robots are equipped with sensors, it can be used for a wide range of robots.

In order to determine the source of a detrimental disturbance in a robot, information regarding the robot's anatomy and a set of research actions have been utilised. The robot's anatomy provides information about which of a robot's moving parts are closer to the sensors evidencing an abnormal situation. These components are more likely to be faulty or provide extra information about the source of the anomaly. Therefore, these components are the first to be used for executing research actions. The set of research actions must be specifically designed for each kind of robot in order to help identify faulty components. This is analogous to instinctive behaviour being specific for each individual species. Once research actions are executed and their effects monitored, unexpected sensor readings can be associated with faulty components or a feature of the robot's environment that is affecting that component. In general, it is simpler to identify faulty components than unexpected features of the robot's environment. This is especially true when the effects of the anomaly are spread among the robot's sensors. When no faulty robot components are found, the system assumes that the abnormal situation was generated via the robot's environment. Assumptions or the robot's internally generated theories about the source of a detrimental disturbance are corroborated or rejected after the corresponding compensatory measure is executed. Therefore, if the robot's performance is improved after this compensatory measure has been executed; the system considers that the anomaly source was correctly determined. Otherwise, the next moving part close to the sensors providing abnormal readings is selected to perform the corresponding research actions and a new iteration of the identification-compensation process is executed.

The self-diagnosis method proposed in this research provided correct identification of a series of faults induced in the experimental robot. Although no other kinds of robots were utilised during the testing stage, it is expected that these results can be extrapolated to most robot types.

Experiments with anomalies in the robot's hardware showed that faults with isolated effects detected by sensors directly linked to the fault were normally identified correctly by the self-diagnosis methods on the first attempt. As these effects propagated to other sensors, the system needed more attempts before a correct identification was achieved. This was especially true when the effects on sensors directly linked to the fault were smaller than for other sensors. On the other hand, there were cases when a fault did not produce any abnormal sensor reading. Although, this situation could be tackled by improving the methods for the calculation of expected sensor readings, in some cases it may be necessary to increase the number and/or type of sensors in the robot. However, the effects of faults among the increased number of robot sensors could propagate and increase the complexity of arriving at a correct diagnosis in certain situations.

The experiments that were conducted with anomalies in the robot's environment showed the good performance of the self-diagnosis methods when identifying situations that generated abnormal sensor readings. However, here the system struggled to identify those situations where abnormal sensor readings were not generated. This occurred, for instance, while the robot was air walking or when one of the two light sensors facing the light source was partially covered.

The self-diagnosis methods also showed a good identification rate in experiments where both anomalies in the robot's hardware and in the robot's environments were considered. Best results were achieved in those cases where the ASC corresponding to the target abnormal situation was previously incorporated into the robot's database of abnormalities. In some cases, the robot was still able to identify the abnormal situation correctly at the first attempt when this information was not available. However, in other cases with unknown target ASC the performance of the system dropped drastically. For instance, the robot had to perform more than 54 attempts of diagnosis before identifying correctly the abnormal situation when legs 1 and 6 were held, in experiment 1 of Table 4.70.

In general, the rate of first attempt identification of abnormal situations in the robot's environment will depend on the number and types of the ASCs located in the robot's database of abnormalities. Discrepancy values (Euclidean distance between the centroid of an abnormal situation experienced by the robot and the centroid of abnormal situations incorporated into the robot's database of anomalies) are usually larger for linearly independent ASCs. Therefore, better diagnosis is performed for databases incorporating this type of ASCs. In the presence of linearly dependent ASCs, the classification of the abnormal situation will depend more on the magnitude of the ASC components.

As mentioned before, self-diagnosis methods proposed in this thesis require the target robot to autonomously assess its performance. In addition, they require techniques for the calculation of expected values for all of the types of sensors on the robot. However, this information may be difficult or impossible to obtain in certain robotic systems or missions. Therefore, a limitation of the proposed self-diagnosis methods is that their application is constrained to those robots that are able to provide this information. However, in spite of this limitation, there are many robotic systems where the proposed methods are applicable, which is an advantage. In addition, the sensitivity of each sensor on the target robot can be regulated by a parameter of the classification algorithm. This is an advantage that allows the system to work with different sensor resolutions and better adjust the self-diagnosis methods to different robotic systems and environments.

A disadvantage of the self-diagnosis methods proposed in this thesis is the time they require for the identification of abnormal situations generated in the robot's environment when a large number of abnormal sensor readings are generated. In these cases, the robot generates many theories about possible sources of the anomaly. Then, the corresponding research actions must be executed and these theories are rejected one by one. Only once the possibility of a fault in the robot's hardware is discarded, does the system turn to the identification of abnormal situations in the robot's environments.

Another disadvantage of the proposed methods is that robots utilising them may require more time to complete their mission. This is because these robots must be constantly monitoring their performance and providing all of their available sensorial information in order to communicate their status. In addition, depending on the number of anomalies these robots are expected to detect, they may be provided with more sensors. These sensors consume more energy, increase the robot complexity and introduce more components prone to failure.

6.1.2. Self-Compensation in Robotics

A number of techniques for the compensation of detrimental disturbances faced by robots have been developed during this investigation. Firstly, a group of compensatory measures designed into robots has been considered. This emulates the instinctive behaviour that living creatures have since they were born. In the artificial case, these compensatory measures are programmed by the robot designer.

This thesis has proposed a labelling system that allows matching abnormal situations with their corresponding compensatory measures. The centroid of the abnormal situation that the compensatory measure is intended to compensate is included into the label. Then, the Euclidean distance between the centroid of the current detrimental disturbance experienced by the robot and the centroids included into the compensatory measure labels, is used for finding a suitable compensatory measure.

Here, a threshold value determining the maximum allowed distance between corresponding pairs of disturbance-compensatory measure is utilised. Matching errors may arise due to variations in the centroid associated with different occurrences of an abnormal situation, or an inappropriate initial threshold value. The method proposed in this research adjusts both parameters. As a result, more accurate centroid and threshold values are obtained as the robot accumulates more experience with a particular detrimental disturbance. Hence, it is recommended to expose the robot a number of times to the same detrimental disturbance before including these parameters into the associated compensatory measure label.

This work has also investigated techniques for the association of compensatory measures and a robot's faulty components. These algorithms use a labelling system for the robot's parts. Then, corresponding compensatory measures include this information into their own labels. Therefore, once a faulty component is detected, the system searches for the minimum Euclidean distance between the faulty component label and target components in compensatory measure labels.

The compensation of every single abnormal situation affecting every single type of robot is far beyond the scope of this research. However, this thesis has provided several examples of detrimental disturbances induced in the experimental robot and their corresponding compensatory measures. This work also presents techniques for the automatic generation of compensatory actions. Although these techniques were proposed, they were not implemented into the experimental robot. Because these methods require more processing time and the robot's energy, they should only be used when the previously discussed techniques are unable to compensate an abnormal situation.

Although it was not incorporated into the real robot, the proposed methods integrate autonomously generated actions, which have been able to compensate an abnormal situation, into the robot’s compensatory measure database. This type of compensatory measure is part of the robot’s learned behaviour.

Two last resort measures for robots that have been unable to compensate a detrimental disturbance have been included into this research. In the first technique, the robot enters into a monitoring sleep mode, saving energy and waiting until the abnormal situation rectifies itself. The second measure consists of transmitting a distress signal with information about the robot’s position and status.

An advantage of the self-compensation methods proposed in this thesis is that in theory they can be applied to different robotic systems performing different types of missions. Although compensatory measures built into the target robot must be adjusted to each specific case, the general methods proposed in this thesis are not constrained to the experimental robot utilised in this research. In addition, this thesis has proposed self-compensation methods for different degrees of information about possible anomalies found in a robotic mission. Consequently, the target robot may be provided with a large number of innate compensatory measures for those missions where the kinds of anomalies the robot may experience are known. On the other hand, autonomously generated compensatory measures could be used when there is little information about the environment where the robot will perform its mission.

A disadvantage of the proposed self-compensation methods is that they require computer memory in order to create a database of compensatory measures. This database starts with the innate compensatory measures provided by the robot designer and it may be later extended with autonomously generated compensatory measures. As this database grows, the time required for finding suitable compensatory measures is increased. However, this may only be an issue for complex systems which are required to compensate a very large number of anomalies.

6.1.3. Mechatronics

This research has lead to the development of an opto-mechanical leg tip force sensor. This sensor measures a voltage derived from an opto-interrupter, which is related to pressure or traction applied to the leg's tip. In comparison with the other type of force sensor present in the robot, this opto-mechanical sensor is more sensitive and accurate. Previously in this research, the force applied by each leg was determined by combining information about the applied force and the rotation angle of the leg's servo motors. The leg tip force sensor is more convenient because this information can be directly calculated from the sensor output and the sensor requires less wiring and fewer microcontroller inputs.

This investigation has also led to the development of a leg release mechanism, which is used in extreme circumstances as a last resort corrective action. For instance, a broken or faulty leg could interfere with or prevent a stable gait, or a leg could be entangled in some feature of the robot’s environment. In these cases, the leg release mechanism allows the robot to continue making progress in its mission. The compactness of this novel mechanism allows its incorporation into highly integrated robotic systems. Furthermore, because the operation of the leg release mechanism involves only a few simple components, it may be more reliable than complicated systems including a large number of failure-prone elements.

A disadvantage of the proposed leg release mechanism is that its activation requires more energy that the R-LEGAM system. The proposed system heats up a resistor, which must dissipate heat during a certain period of time (around 3 minutes average) in order to melt the alloy. On the other hand, R-LEGAM only uses energy for rotating a servo motor so its activation is more energy efficient. However, this system utilises more components which add more weight and take up more space on the robot. This means that the R-LEGAM mechanism utilises more energy for its transportation than the proposed method.

6.2. Further Research

Although a number of methods targeting self-diagnosis and self-compensation in robotics have been proposed in this work, it is clear that more research must be conducted in these areas before robots can perform their missions as effectively as living creatures in unstructured environments.

The techniques proposed in this thesis can be further developed by implementing them in a wider range of robots. In these cases, the robot innate behaviour must be modified according to the anatomy and kind of mission performed by the specific robot. Moreover, the range of abnormal situations that robots are able to compensate can be extended by increasing the number of robot research actions and innate compensatory measures. This range can also be extended by exposing robots to a wider range of abnormal situations and allowing them to generate and learn suitable compensatory measures. In addition, the autonomous generation of compensatory measures can be improved by increasing the number of malleable actions incorporated into the robot. This method can be further developed, so parameters associated with every single task a robot is able to perform can be modified and used in the generation of malleable actions.

Self-diagnosis could be accelerated by proposing a better method that allows to distinguish between intrinsic and extrinsic robot anomalies. Currently with the proposed methods, the robot must execute research actions in order to test if every abnormal sensor reading is generated by a hardware fault. In some cases, when a large number of abnormal readings are generated by an anomaly in the robot's environment, this may be very time and energy consuming and is clearly not an optimal solution.

Methods proposed in section 5.4: "Autonomously Generated Compensatory Measures: and section 5.6: "Learned Compensating Measures" were not implemented in a real robot. Experimental validation of these techniques is proposed as future work.

Finally, additional robotic subsystems can be investigated in order to extend robot capabilities to identify and compensate abnormal situations. The leg tip force sensor and the leg release mechanism presented in this thesis are examples of such systems.

6.3. Closure

This investigation has developed and implemented a number of biologically inspired methods for the autonomous identification and classification of disturbances that have negative effects on a robot's performance (self-diagnosis), and the autonomous selection and generation of suitable compensatory actions (self-compensation).

The work developed in this research has reduced the gap between legged locomotion currently found in robots and the fully autonomous locomotion observed in living organisms. Moreover, it is expected that results obtained from this research will benefit most kinds of robots. This is especially true when it is required to increase resilience to damage, extend lifespan and improve autonomy in robotic missions where human intervention is difficult or impossible, such as in extra-terrestrial exploration or deployment in other remote hostile environments.

Appendix A
Robot Kinematics

This appendix presents direct and inverse kinematics equations of the experimental robot. The kinematic model allows us to relate the (x, y, z) *coordinates of a robot leg tip with the corresponding angles of the three leg servo motors, and vice versa.*

Appendix A is organised as follows. The direct kinematics of the robot is presented in section A.1. Finally, the robot's inverse kinematics are provided in section A.2.

A.1. Direct Kinematics

The direct kinematics of the experimental robot allows us to transform the rotation angles of a leg's servo motors into $\left(x_k, y_k, z_k\right)$ coordinates of the leg tip. Because the robot's legs are mirrored on either side of the y-axis, there are slight differences in the calculation of the direct kinematics of legs located on different sides of this axis. Nevertheless, the direct kinematics of any leg of the experimental robot can be determined by means of Eqs. A.1 to A.4.

$$cst = d_2 \sin\left(\theta_{2\_k} + \pi/4\right) + d_3 \sin\left(\theta_{2\_k} - \theta_{3\_k}\right) + d_1 \quad \text{(A.1)}$$

$$x_k = round\left(x_{0\_k} + u \times cst \times \cos\left(\theta_{1\_k} - \alpha_k\right)\right) \quad \text{(A.2)}$$

$$y_k = round\left(y_{0\_k} + cst \times \sin\left(\theta_{1\_k} - \alpha_k\right)\right) \quad \text{(A.3)}$$

$$z_k = round\left(-d_2 \cos\left(\theta_{2\_k} + \pi/4\right) - d_3 \cos\left(\theta_{2\_k} - \theta_{3\_k}\right) - z_{0\_k}\right) \quad \text{(A.4)}$$

Where

- k represents the robot leg whose direct kinematics is being calculated.
- If k is 1, 2 or 6 then θ_{1_k} is the rotation angle of servo 1 belonging to leg k. Otherwise, ϕ_{1_k} is the rotation angle of servo 1 belonging to leg k and $\theta_{1\_k} = \pi - \phi_{1\_k}$.
- If k is 1, 2 or 6 then θ_{2_k} is the rotation angle of servo 2 belonging to leg k. Otherwise, ϕ_{2_k} is the rotation angle of servo 2 belonging to leg k and $\theta_{2\_k} = \pi - \phi_{2\_k}$.
- If k is 1, 2 or 6 then θ_{3_k} is the rotation angle of servo 3 belonging to leg k. Otherwise, ϕ_{3_k} is the rotation angle of servo 3 belonging to leg k and $\theta_{3\_k} = \pi - \phi_{3\_k}$.
- $d_1 = 20\ mm$ is the distance between servo 1 and 2, which is measured as shown in Fig. A.1.
- $d_2 = 80\ mm$ is the distance between servo 2 and 3, which is measured as shown in Fig. A.1.

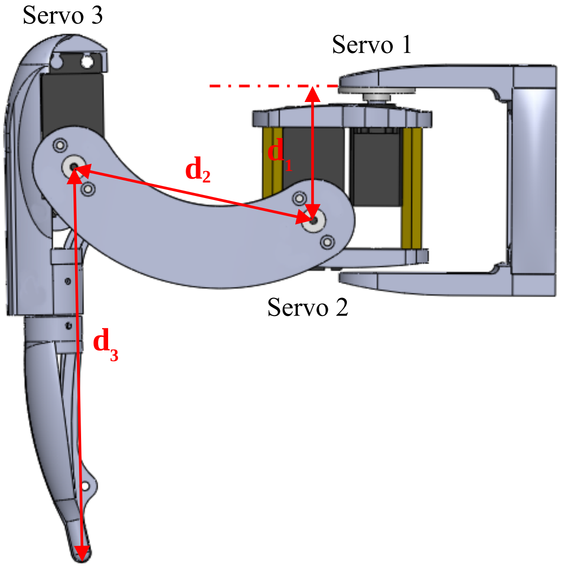


Figure A.1. Distances Associated with the Experimental Robot Kinematics.

- $d_3 = 130\ mm$ is the distance between servo 3 and the leg tip, which is measured as shown in Fig. A.1.
- x_{0_k} represents the x coordinate of servo 1 in leg k. This value is shown in Figure A.2 and Table A.1.
- y_{0_k} represents the y coordinate of servo 1 in leg k. This value is shown in Figure A.2 and Table A.1.
- z_{0_k} represents the z coordinate of servo 1 in leg k. This value is 12 mm for all of the robot's legs.
- α_k is used for adjusting the rotation angle of servo 1. This angle gives consistency to the rotation angle of servo 1 regardless of the leg distribution in the robot. For instance, by using this angle any of the robot's legs will be in a horizontal position when the rotation angle of servo 1 is 0 rad. The α_k angle value is shown in Table A.1.
- If k is 1, 2 or 6 then $u = 1$. Otherwise, $u = -1$.

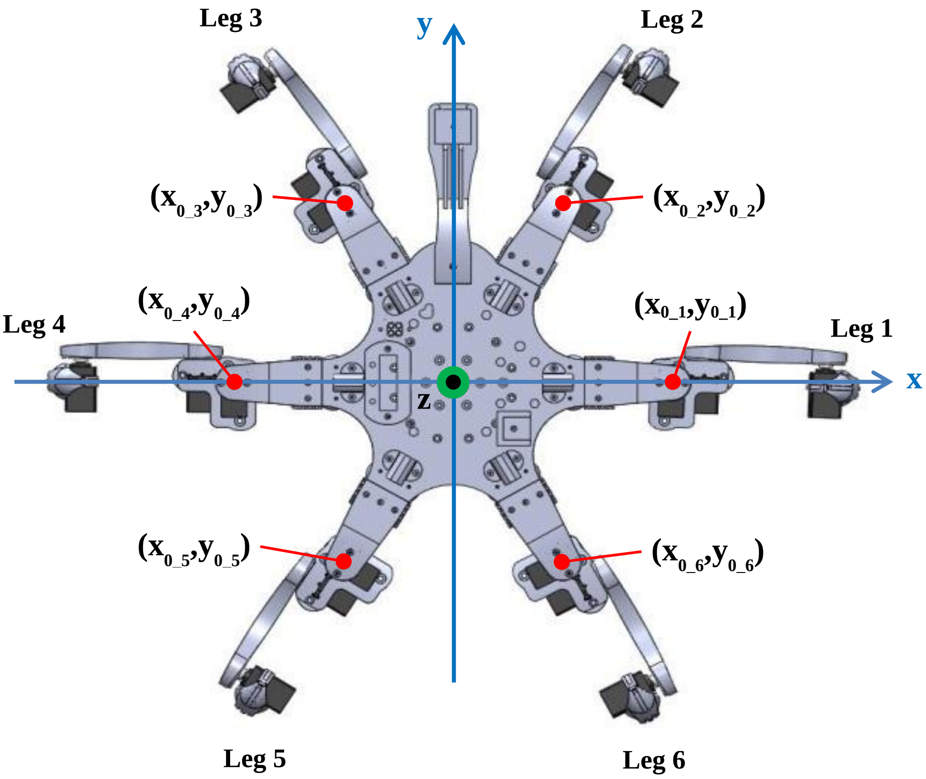


Figure A.2. Parameters Associated with the Experimental Robot Kinematics.

Table A.1. Values of Parameters Associated with Robot Kinematics.

Leg	1	2	3	4	5	6
x_{0_Leg} (mm)	150	75	-75	-150	-75	75
y_{0_Leg} (mm)	0	127	127	0	-127	-127
α_{Leg} (rad)	$\pi/2$	$\pi/6$	$\pi/6$	$\pi/2$	$5\pi/6$	$5\pi/6$

A.2. Inverse Kinematics

The inverse kinematics of the experimental robot allows us to transform the $\left(x_k, y_k, z_k\right)$ coordinates of a leg tip into rotation angles of the corresponding leg servo motors. A first step towards the calculation of these angles is to find the $\left(\theta_{1\_k}, \theta_{2\_k}, \theta_{3\_k}\right)$ angles from the system of equations Eqs. A.1 to A.4. After some algebraic manipulation of this system of equations, the following equations are obtained.

A.2.1. Calculation of θ_{1_k}.

$$\theta_{1\_k} = \alpha_k + \arctan 2\left(\frac{y_k - y_{0\_k}}{x_k - x_{0\_k}}\right) \tag{A.5}$$

Because the servo motor angles are in the range 0 to π, the θ_{1_k} angle is converted to the first or second quadrant.

A.2.2. Calculation of θ_{2_k}.

- If $R > 0$,

$$S_1 = \frac{-B - \sqrt{R}}{2A} \tag{A.6}$$

$$S_2 = \frac{-B + \sqrt{R}}{2A} \tag{A.7}$$

If $S_2 > 1$,

$$S_1 = \arcsin\left(\frac{-B - \sqrt{R}}{2A}\right) - \pi / 4 \tag{A.8}$$

$$S_2 = \pi / 4 \tag{A.9}$$

- If $R \leq 0$,

$$S_1 = \frac{-B}{2A} \tag{A.10}$$

If $S_1 < -1$,

$$S_1 = S_2 = -3\pi / 4 \tag{A.11}$$

If $S_1 > 1$,

$$S_1 = S_2 = \pi / 4 \tag{A.12}$$

Where

- $A = K_1^2 + K_4^2$, $B = -2K_3K_4$, $C = K_3^2 - K_1^2$ and $R = B^2 - 4AC$.
- $K_1 = -2d_2\left(z_k + z_{0\_k}\right)$

- $K_2 = \begin{cases} \dfrac{y_{0\_k} - y_k}{\sin\left(\theta_{1\_k} - \alpha_k\right)} - d_1 & , \quad if \;\; \theta_1 \neq \pi / 2 \\ \dfrac{x_{0\_k} - x_k}{\cos\left(\theta_{1\_k} - \alpha_k\right)} - d_1 & , \quad otherwise \end{cases}$

- $K_3 = \left(-z_k - z_{0\_k}\right)^2 + d_2^2 - d_3^2 + K_2^2$

- $K_4 = 2d_2K_2$

After solving the equation system resulting from Eqs. A.1 to A.4, four solutions for θ_{2_k} in the first four quadrants are determined. These solutions are shown in expression A.13.

$$\theta_{2\_k} = \left\{ S_1, \frac{\pi}{2} - S_1, S_2, \frac{\pi}{2} - S_2 \right\} \tag{A.13}$$

Because the servo motor angles are limited to the range 0 to π, θ_{2_k} angles outside this range are discarded.

A.2.3. Calculation of θ_{3_k}.

After θ_{1_k} was calculated and possible solutions of θ_{2_k} determined, the obtained angles were used for solving for θ_{3_k} from the equation system. Then, the four solutions for θ_{3_k} in expression A.14 were considered.

$$\theta_{3\_k} = \left\{ \theta_{3\_k}^1, \theta_{3\_k}^2, \theta_{3\_k}^3, \theta_{3\_k}^4 \right\} \tag{A.14}$$

Where,

- $\theta_{3\_k}^1 = \begin{cases} \theta_{2\_k} - \arcsin\left(\dfrac{\dfrac{x_{0\_k} - x_k}{\cos\left(\theta_{1\_k} - \alpha_k\right)} - d_1 - d_2 \sin\left(\theta_{2\_k} + \pi / 4\right)}{d_3} \right) & , \quad if \;\; \theta_{1\_k} = \alpha_k \\ \theta_{2\_k} - \arcsin\left(\dfrac{\dfrac{y_{0\_k} - y_k}{\sin\left(\theta_{1\_k} - \alpha_k\right)} - d_1 - d_2 \sin\left(\theta_{2\_k} + \pi / 4\right)}{d_3} \right) & , \quad otherwise \end{cases}$

- $\theta_{3\_K}^{2} = \pi - \theta_{3\_K}^{1}$
- $\theta_{3\_K}^{3} = \theta_{2\_k} - \arccos\left(\dfrac{-z_k - z_{0\_k} - d_2\cos\left(\theta_{2\_k} + \pi/4\right)}{d_3}\right)$
- $\theta_{3\_K}^{4} = \theta_{2\_k} + \arccos\left(\dfrac{-z_k - z_{0\_k} - d_2\cos\left(\theta_{2\_k} + \pi/4\right)}{d_3}\right)$

The final $\left(\theta_{1\_k}, \theta_{2\_k}, \theta_{3\_k}\right)$ solution was determined by considering all of the possible combinations of solutions in expressions A.5, A.13 and A.14. Then, the combination that satisfied the equations in the system was selected as the final $\left(\theta_{1\_k}, \theta_{2\_k}, \theta_{3\_k}\right)$ angles.

A.2.4. Calculation of Leg Servo Motor Angles.

The leg servo motor angles are the $\left(\theta_{1\_k}, \theta_{2\_k}, \theta_{3\_k}\right)$ angles obtained in the previous subsection if k is 1, 2 or 6. Otherwise, the leg servo motor angles are the $\left(\phi_{1\_k}, \phi_{2\_k}, \phi_{3\_k}\right)$ angles calculated by means of Eqs. A.15 to A.17.

$$\phi_{1\_k} = \pi - \theta_{1\_k} \tag{A.15}$$

$$\phi_{2\_k} = \pi - \theta_{2\_k} \tag{A.16}$$

$$\phi_{3\_k} = \pi - \theta_{3\_k} \tag{A.17}$$

Finally, the $\left(x_k, y_k, z_k\right)$ Cartesian coordinates of leg k 's tip can be transformed into the $\left(\theta_{1\_k}, \theta_{2\_k}, \theta_{3\_k}\right)$ or $\left(\phi_{1\_k}, \phi_{2\_k}, \phi_{3\_k}\right)$ corresponding angles of the leg servo motors by following the procedure described in this section.

Appendix B
Adaptable Gait Generation for Autotomised Legged Robots

This appendix presents an adaptable gait generation method, which allows legged robots to walk in a stable fashion after they have shed a number of legs. By using this technique, robots will be able to continue with their mission even after one or more legs have been damaged. The method selects and calculates the final coordinates of a robot's stance and swing legs by maximising the stable mobility of the robot in the direction of locomotion. As a result, stable straight line locomotion in any desired direction can be generated. The proposed technique has been tested in a hexapod robot, but results can be extended to robots with any number of legs. This method intends to reduce the current gap between biological and robot locomotion, extending robot resilience to damage and improving autonomy.

Appendix B is organised as follows. First of all, an adaptable gait generation method is presented in section B.1. Next, a method where the robot's tilt is modified in order to improve the robot's stability is proposed. This is followed by sections B.3 and B.4, where techniques that allow the robot to change its direction of locomotion are explained. Experimental results of the introduced gait generation method are then presented and discussed in section B.5. A summary of this appendix is provided in the final section.

B.1. Gait Generation

In general, gait generation in a robot capable of shedding its legs must be suitable for all possible leg configurations. Once the robot has autotomised one or more of its legs, it must continue walking with its remaining hardware resources. In addition, the gait must be flexible enough to be able to deal with complex terrain and unexpected obstacles. Another two major aspects that should be considered in any proposed gait are stability and mobility. The robot’s relative stability is ensured only if the vertical projection of its centre of mass on the ground plane (from now on simply referred to as robot´s centre of mass) is inside the convex hull formed by the vertical projection of the points of contact of the supporting legs on the ground plane. Therefore, stability can only be maintained if there are at least three legs on the ground.

Initially the intact robot has six legs, which have been named as shown in Fig. B.1. The legs’ positions are established by using the Cartesian coordinate system located on the robot, with its positive z axis pointing out of the paper plane and with its positive y axis pointing in the direction of the front of the robot, as is also shown in

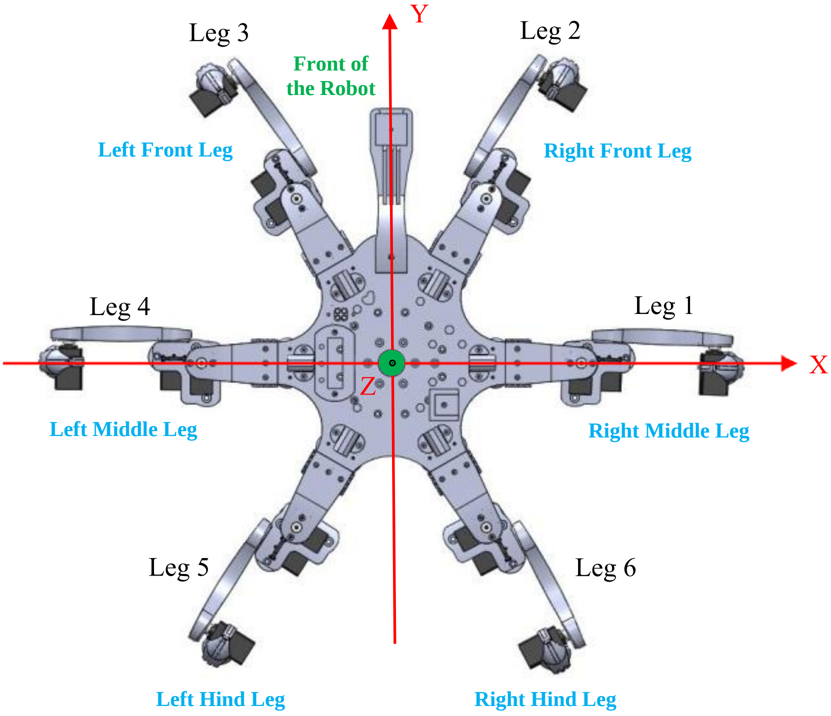


Figure B.1. The Robot’s Cartesian Coordinate System and Initial Leg Configuration.

the figure.

Considering the case when one leg is missing, then there are 10 possible combinations of supporting triangles of legs that can be formed from those still attached to the robot. Overall, the number of possible triangles is determined by performing combinatorial calculations between n_{legs}, the total number of available robot legs, and 3, the number of supporting legs that form the triangle. This is expressed by Eq. B.1.

$$number\ of\ triangles = n_{\Delta} = {}^{n_{legs}}C_3 = \frac{n_{legs}!}{3!\left(n_{legs}-3\right)!} \tag{B.1}$$

In general, the set of available robot legs can be defined as $L_{available} = \{l_1, l_2, ..., l_n\}$, where each l is a number that represents the corresponding available leg. For instance, if $L_{available} = \{2,3,5,6\}$, in this case the value $l_1 = 2$ indicates that Leg 2 is one of the available legs. Then, the set S of all the possible supporting tripods contained in $L_{available}$ is represented by Eq. B.2.

$$S = \left\{s_1, s_2, ..., s_{n_{\Delta}}\right\} \tag{B.2}$$

Where

- $s_i = \left\{l_{k1}, l_{k2}, l_{k3}\right\}$.
- $i = 1, 2, ..., n_{\Delta}$.
- $k_1 = 1, 2, ..., n_{legs} - 2$.
- $k_2 = k_1 + 1, k_1 + 2, ..., n_{legs} - 1$.
- $k_3 = k_2 + 1, k_2 + 2, ..., n_{legs}$.

However, not all of the elements of S represent leg tripods which provide a stable support for the robot. Therefore, only the stable tripods, those which enclose the robot's centre of mass, are selected. Then, the set of stable leg triangles can be defined by means of expression B.3.

$$\Delta = \{ s \mid s \in S \wedge s \ \textit{is a stable tripod} \} \tag{B.3}$$

Considering that no collinear points can enclose the robot's centre of mass, each element of Δ represents a set of three legs that form a stable supporting triangle. The position of the tips of these legs corresponds with the location of the triangle's vertexes in the robot's Cartesian coordinate system. Finding the most suitable triangle for supporting the robot and calculating the final coordinates of both stance and swing legs in order to maximise stable robot mobility are the main tasks of the proposed gait generation approach.

B.1.1. Six Legs Gait Generation

The first step in the proposed six legs gait generation algorithm is to find all of the possible triangles formed with the available robot legs and verify if the robot's centre of mass is located inside these triangles. Once all the unstable triangles are discarded, relative stability is guaranteed if any of the remaining triangles is selected as the supporting legs. Then, a measure of relative stability of each triangle is calculated. This is simply the distance d_{Δ_stable} between the robot's centre of mass and the respective triangle's centroid.

In the second step, the distance d_{stance_stable} between the robot's centre of mass and the side of each stable triangle in the direction of locomotion is calculated. During the stance phase, the maximum distance that the triangle formed by the supporting legs can be moved whilst it still contains the robot's centre of mass is determined by d_{stance_stable}. However, this distance might be larger than the available backward mobility of the stance legs. Hence, it is necessary to calculate the backward mobility of each triangle, which is the minimum distance d_{stance_mob} between the current position and the posterior extreme position (PEP) of its legs. Then, the stance legs triangle Δ^* can be established by means of expression B.4 and the Δ' set, which is composed of all the elements of Δ whose d_{stance_stable} and d_{stance_mob} are different from zero.

$$stance\_legs = \Delta^* \mid \Delta^* \in \Delta' \wedge \forall \Delta_i \in \Delta',$$

$$\frac{d^*_{\Delta\_stable}}{\min\left(d^*_{stance\_stable}, d^*_{stance\_mob}\right)} \le \frac{d_{\Delta\_stable_i}}{\min\left(d_{stance\_stable_i}, d_{stance\_mob_i}\right)} \quad \text{(B.4)}$$

Where

- $i = 1,2,...,$ number of elements of Δ'.

The final (x, y) coordinates of the stance legs, the position of the legs once their stance cycle has concluded, are determined by means of expression B.5.

$$\begin{bmatrix} x_{final\_stance\_coord} \\ y_{final\_stance\_coord} \end{bmatrix} = \cdots$$

$$\cdots = \begin{bmatrix} x_{initial\_stance\_coord} \\ y_{initial\_stance\_coord} \end{bmatrix} + \begin{bmatrix} \min\left(d^*_{stance\_stable} - \varepsilon, d^*_{stance\_mob}\right)\cos\left(\theta + \pi\right) \\ \min\left(d^*_{stance\_stable} - \varepsilon, d^*_{stance\_mob}\right)\sin\left(\theta + \pi\right) \end{bmatrix} \quad \text{(B.5)}$$

- Where θ is the robot movement heading, $\pi/2$ for straight ahead walking. The ε parameter is utilised in order to avoid the robot's centre of mass lying exactly on the edge of the stance legs triangle, which could lead to an unstable condition. This would be the case if $\varepsilon = 0$ when $d^*_{stance\_stable}$ is less than $d^*_{stance\_mob}$ and the stance triangle is moved to the limit of stable mobility. In this work $\varepsilon = 5$ has been adopted, meaning that in the worst case the stance legs will be 5 mm inside the outer limit of the stance triangle.

The θ angle allows the robot to change its direction of locomotion without rotating. This may be useful when the robot is walking in constrained spaces. In comparison with the rotation methods introduced in sections B.3 and B.4, the technique presented here is the faster way to change the robot's direction of locomotion. However, this method presents two drawbacks. The first one is that if some of the robot's sensors are intended for forward locomotion, then some sensorial information may be lost. The second is that the legs' mobility may be reduced when the robot is not walking in the direction for which was designed. As a result, the robot's locomotion could be slower.

A top view of a selected stance legs triangle and the respective swing legs triangle are shown in Figs. B.2 and B.3. Some of the parameters utilised by Eqs. B.4 to B.6 are also illustrated in these figures. In both of them the θ value is around $4\pi/9$ (or 80^{o}). Therefore, in this case, the robot is heading forwards but with a slight inclination to the right.

The position of the robot’s centre of mass depends on the hardware structure, in this example for simplicity it coincides with the origin of the robot’s coordinate system.

The swing legs are simply determined as the remaining available legs once the stance legs are known. When the robot still has six functional limbs, the final coordinates of

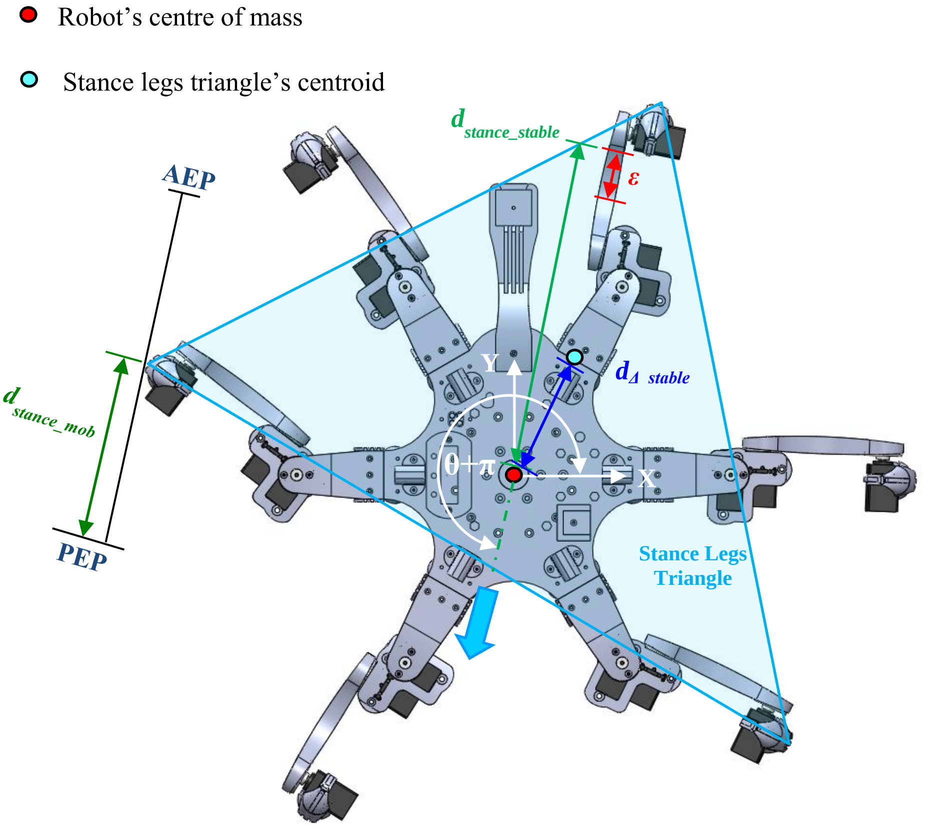


Figure B.2. Top View of a Stance Legs Triangle Including Related Parameters.

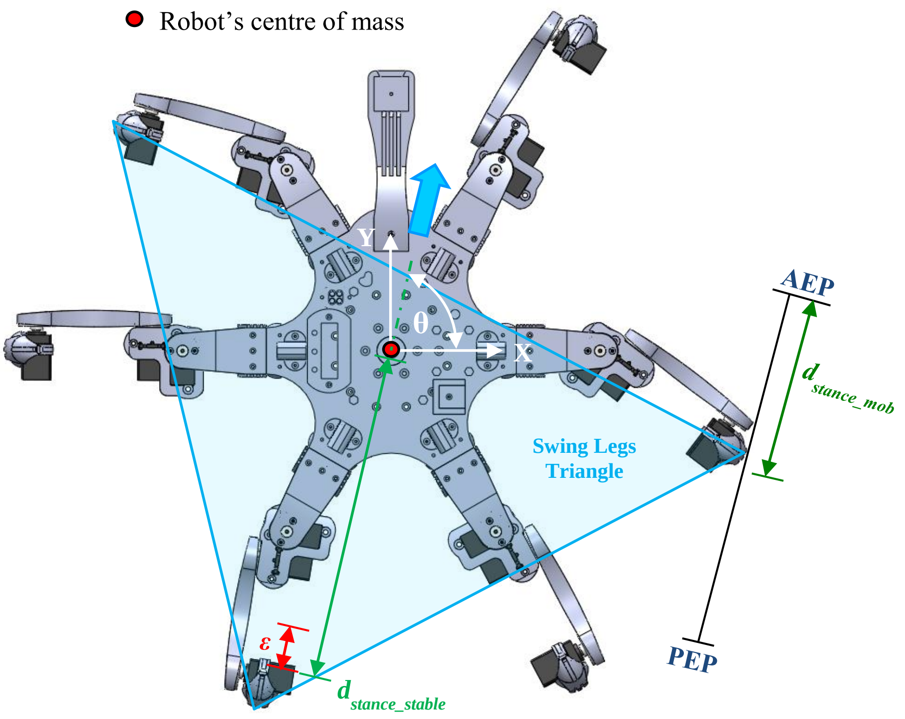


Figure B.3.Top View of a Swing Legs Triangle Including Related Parameters.

the swing legs are established by considering the minimum d_{swing_mob} and the maximum d_{swing_stable} distance. The former is the distance between the current position of the swing legs and their respective anterior extreme position (AEP), and the latter is the distance that the triangle formed by the swing legs can be moved towards the opposite direction of locomotion whilst still containing the robot's centre of mass. This assures the existence of a stable triangle when the swing legs finish their cycle and become stance legs.

The final (x,y) coordinates of the swing legs, the position of the legs once their swing cycle has concluded, are calculated as follows.

$$\begin{bmatrix} x_{final\_swing\_coord} \\ y_{final\_swing\_coord} \end{bmatrix} = \begin{bmatrix} x_{initial\_swing\_coord} \\ y_{initial\_swing\_coord} \end{bmatrix} + \begin{bmatrix} \min\left(d^{*}_{swing\_stable} - \varepsilon, d^{*}_{swing\_mob}\right)\cos(\theta) \\ \min\left(d^{*}_{swing\_stable} - \varepsilon, d^{*}_{swing\_mob}\right)\sin(\theta) \end{bmatrix} \quad \text{(B.6)}$$

B.1.2. Five Legs Gait Generation

The basic strategy employed in the five legs gait generation is analogous to the one described in the previous subsection. However, once three stance legs have been selected, there are only two remaining swing legs. Thus, it is unfeasible to immediately form the triangle of supporting legs for the next cycle. In this case, one of the stance legs must be selected in order to calculate the final (x, y) coordinates of the swing legs triangle by means of Eq. B.6. This is achieved by evaluating all of the possible stable triangles, constituted by the two swing legs and each one of the stance legs, in Eq. B.4. Determining the future supporting legs in this way means that once again stability and mobility are the employed criteria. The resulting gait has two stages. First, the three initial stance legs are moved against the direction of locomotion, towards their final coordinates calculated by means of Eq. B.5. At the same time, the two swing legs are moved in the opposite direction, towards the coordinates determined by Eq. B.6. In this calculation, the information about the position of the selected third swing leg, at the beginning of its stance phase has been employed, but the current movement of the leg is still governed by its stance phase. In the next stage, once all the legs have reached their target coordinates, the stance leg selected as the third swing leg is moved in the same direction as that previously followed by the two initial swing legs. In this stage there is no forward motion, only the third swing leg is moved whilst there are four legs supporting the robot. Once this leg reaches the coordinates specified by Eq. B.6, the cycle starts again with stance legs becoming swing legs and vice versa.

B.1.3. Four Legs Gait Generation

A similar approach to the one previously discussed for five legs can be adopted for gait generation when there are only four functional legs. In this case, the stance legs are also calculated by Eq. B.4. Nonetheless, there is a slight difference with respect to how the final coordinates of both stance and swing legs are established. If these coordinates were determined by means of Eqs. B.5 and B.6, respectively, there would be a point where the only stable triangle that could be selected as stance legs had no mobility at all. In order to avoid this situation, observations of quadruped gaits, particularly in the elephant, have been conducted. The main elephant gait feature incorporated in the four legs robot gait is that each elephant leg stays in stance phase until the remaining three legs have experienced the swing phase, moving the stance legs on each step only about one third of their available backward mobility. Consequently, the backwards mobility of the robot stance legs has been restricted to be equal or less than a third of max_{mob}, the maximum backwards mobility of all the robot's legs. This value is calculated at the beginning of the locomotion or every time the four legs have experienced the swing phase. Once max_{mob} has been determined, the target stance legs and swing leg coordinates are established by means of Eqs. B.7 and B.8, respectively.

$$\begin{bmatrix} x_{final\_stance\_coord} \\ y_{final\_stance\_coord} \end{bmatrix} = \cdots$$

$$\cdots = \begin{bmatrix} x_{initial\_stance\_coord} \\ y_{initial\_stance\_coord} \end{bmatrix} + \begin{bmatrix} \min\left(d^{*}_{stance\_stable} - \varepsilon, d^{*}_{stance\_mob}, \frac{max_{mob}}{3} \right) \cos\left(\theta + \pi\right) \\ \min\left(d^{*}_{stance\_stable} - \varepsilon, d^{*}_{stance\_mob}, \frac{max_{mob}}{3} \right) \sin\left(\theta + \pi\right) \end{bmatrix} \quad \text{(B.7)}$$

$$\begin{bmatrix} x_{final\_swing\_coord} \\ y_{final\_swing\_coord} \end{bmatrix} = \begin{bmatrix} x_{home\_swing\_coord} \\ y_{home\_swing\_coord} \end{bmatrix} + \begin{bmatrix} \frac{max_{mob}}{2} \cos\left(\theta\right) \\ \frac{max_{mob}}{2} \sin\left(\theta\right) \end{bmatrix} \quad \text{(B.8)}$$

Where $x_{home\_swing\_coord}$ and $y_{home\_swing\_coord}$ represent the home position coordinates of the robot legs, which correspond with the middle point between the respective AEP and PEP.

One difference between expressions B.6 and B.8 in the determination of the final coordinates of the swing legs is that in Eq. B.8, the stability of possible triangles formed with a given swing leg is not considered in the calculation of its coordinates. This is unnecessary because, in the proposed four legs gait, sending the swing legs to their AEP results in the creation of stable stance triangles. However, this is not absolutely true at the start of walking, when the four legs are in their home position and must initiate the gait cycle. In order to maintain gait stability each step must be short at the beginning, and then becoming gradually larger until the regular gait has been reached. This is also accomplished by means of the max_{mob} value, which maintains an equilibrium between the backward and forward movement of stance and swing legs, respectively.

B.1.4. Gait Generation with Fewer Legs

The three legs gait generation is analogous to the six legs gait generation. The differences are that here there is only one possible triangle, which constitutes the stance and swing legs alternately. When the legs are in the stance phase the robot is lifted and the legs are moved backwards according to Eq. B.5. Then, the robot is lowered to the ground and the legs start their swing phase. In this stage the legs are moved forwards by using Eq. B.6. Finally, the legs start their stance phase again, lifting the robot and beginning a new stance-swing cycle.

With two or one leg, there is no stable feasible purely legged gait. However, if a point on the bottom plate of the robot is considered as a third leg, it is possible to generate a gait similar to the one for three legs. The difference here is that this kind of locomotion, which resembles a paddling movement, involves dragging the body of the robot. Hence, the robot's bottom plate must be designed accordingly, minimising friction with the terrain and being robust enough to undergo dragging. Once the vertexes of the stance triangle have been identified, the robot rotates until its centroid is closer than the dragging point to the target location, whilst both are located on the straight line defined by the direction of locomotion. Finally, Eqs. B.5 and B.6; and the same sequence of movements described for the three legs gait generation can be employed to generate the gait.

When there is only one leg remaining, the robot is rotated until the dragging point is closer than the robot's centre of mass to the target location, whilst both are located on the straight line defined by the direction of locomotion. In this case, the whole bottom plate of the robot's body is dragged. During the swing phase, the leg is moved forwards to its AEP aligned with the direction of locomotion. Then, during the swing phase, the leg is moved towards the robot's body until it reaches its PEP.

B.2. Z Coordinate Calculations and Improvement of Stability

The initial robot legs z coordinates that are used when it is standing up have been calculated by considering the minimisation of the torque in the joints' motors and maximisation of the legs' mobility. The latter can be determined by calculating the mean value of the distance between each possible leg position and all of the remaining reachable points. Once the minimisation has been performed, it is possible to find the z coordinates which require less energy consumption and allow a wide range of movement. By using this method, the same z coordinate value has resulted for all of the legs, which is represented by z_0.

In some circumstances, changing the z coordinates of the legs in order to modify the robot inclination, and therefore its centre of mass, can improve stability during locomotion. This is true especially when the four or five legs gait is employed and the robot's centre of mass is close to one side of the stance legs triangle. Then, new z coordinates can be calculated, as a function of the desired movement (δ) of the robot's centre of mass towards the stance legs triangle's centroid, by means of expression B.9.

$$z_L = (x_L - x_c)\tan\alpha + (y_L - y_c)\tan\beta - z_0 \tag{B.9}$$

Where

- (x_L, y_L) are the xy coordinates of the leg L. For instance, (x_5, y_5) represents the xy coordinates of the Leg 5.

- (x_c, y_c) are the coordinates of the robot's centre of mass.
- $\alpha = \arccos\left(\dfrac{d_c - \delta\cos(\varphi)}{d_c}\right)$ is the rotation angle of the robot's XZ plane around the y axis.
- $\beta = \arccos\left(\dfrac{d_c - \delta\sin(\varphi)}{d_c}\right)$ is the rotation angle of the robot's YZ plane around the x axis.
- d_c is the distance between the robot's centre of mass and the stance legs convex hull's centroid.
- φ is the angle in the range $[0, \pi/2]$, between the horizontal (parallel to x axis) and the line described by d_c.
- z_L is the z coordinates of the leg L.

Figs. B.4 and B.5 illustrate the parameters α and β involved in Eq. B.9. In Fig. B.4, the robot's centre of mass lies on the positive x axis. In this case, a positive value of α moves the robot's centre of mass towards –x by rotating about the y-axis. If the

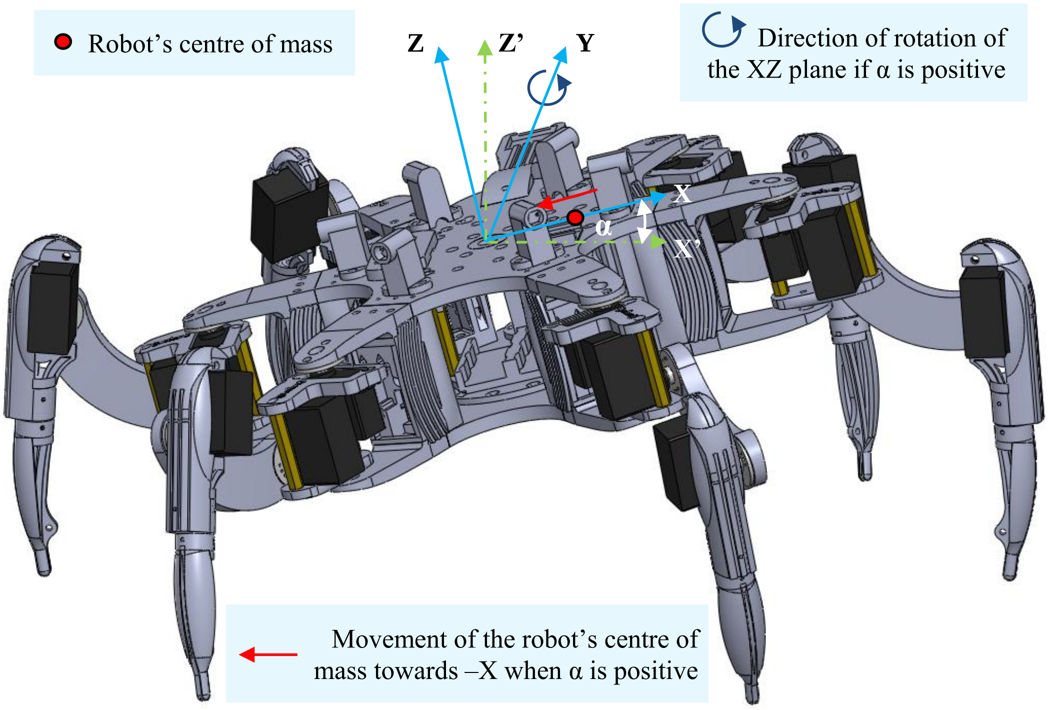

● Robot's centre of mass
Z
Z'
Y
Direction of rotation of the XZ plane if α is positive
X
α
X'
Movement of the robot's centre of mass towards –X when α is positive

Figure B.4. Robot Tilt for a Positive α Angle.

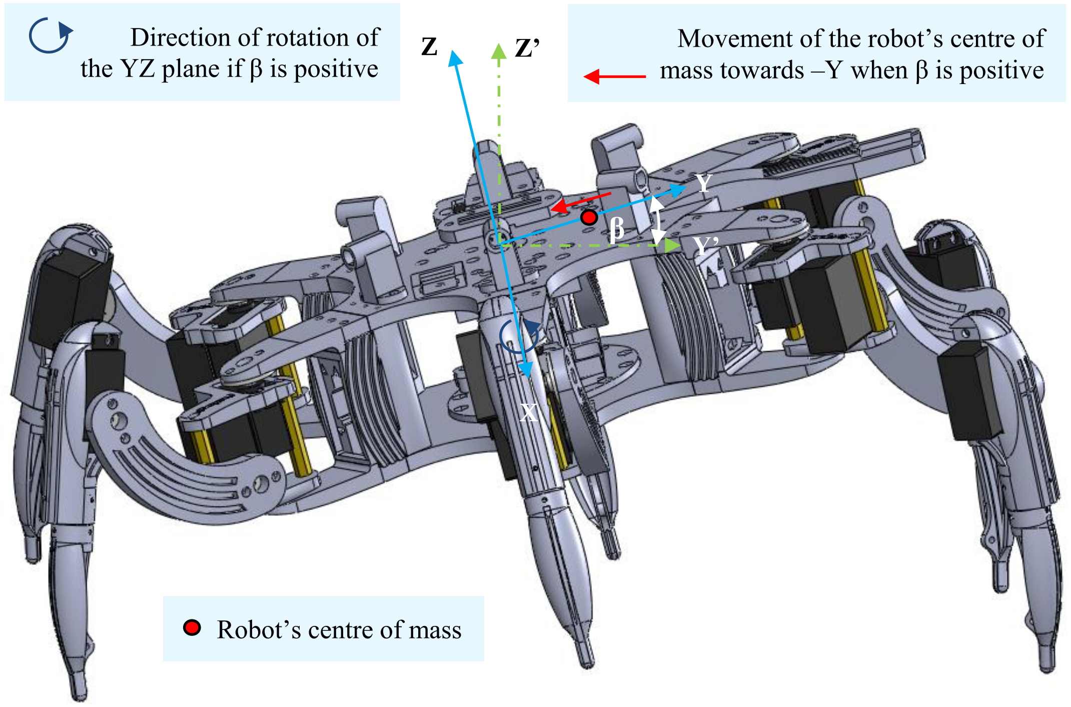

Direction of rotation of the YZ plane if β is positive
Z
Z'
Movement of the robot's centre of mass towards –Y when β is positive
Y
β
Y'
X
● Robot's centre of mass

Figure B.5. Robot Tilt for a Positive β Angle.

robot's centre of mass was lying on the negative x axis, then a negative value of α would produce the same effect. Clearly, in both cases the opposite sign of α moves the robot's centre of mass in the opposite direction. The same principle is illustrated in Fig. B.5, but referring to β instead of α. In this case the robot's centre of mass lies on the y axis and it is moved towards -y with a positive value of β. The superposition of the tilt specified by α and β allows the movement of the robot's centre of mass in any direction on the ground plane.

Figure B.6 shows the remaining parameters of Eq. B.9. In this case, the tilt is necessary for moving the robot's centre of mass inside the stance legs triangle. The δ parameter is a value between 0 and d_c, which is also constrained by the values of α and β that the robot can reach. In practice, these values are limited by the legs' workspace and the relation between z_0 and the robot's body dimensions. In some cases, it may be necessary to decrease z_0 (lowering the legs and lifting the robot's body) in order to avoid dragging the robot's body during the tilt.

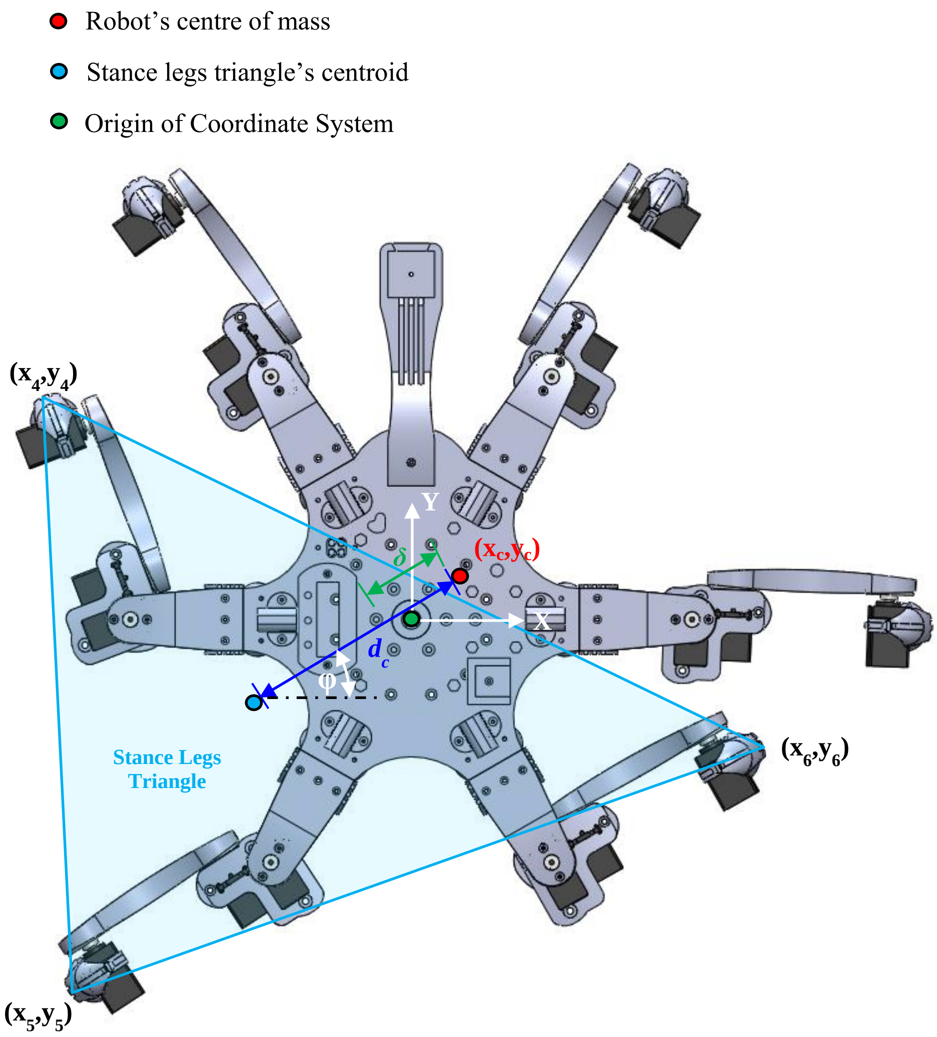


Figure B.6. Parameters Associated with Robot Tilt and Stability Improvement.

When the proposed method is included in the four legs gait, the result is a swaying movement, which improves stability during locomotion. This kind of movement is characteristic of the elephant gait and it can also be observed in many species of quadrupeds.

B.3. Turning Movement

The turning movement is performed by the stance legs, rotating the robot whilst it is walking and mixing rotational and forward locomotion. This generates a smooth robot rotation that resembles the one exhibited by biological creatures. The turning movement can be utilised when it is not desired to stop the robot in order to perform the rotation.

Considering that all the stance legs selected by using expression B.4 have one single leg located on one side of the robot and the remaining two on the opposite side, the single one is chosen as the pivot leg. Then, the turning movement is achieved by maintaining the pivot leg immobile whilst the remaining two legs are moved describing an arc. As a result, a rotation towards the side of the pivot leg is obtained. Hence, the set of stable supporting triangles of legs that allow a turning movement can be defined by expression B.10.

$$\Delta_{turn} = \{s \mid s \in \Delta \wedge s \text{ } has \text{ } the \text{ } pivot \text{ } leg \text{ } on \text{ } the \text{ } side \text{ } of \text{ } the \text{ } desired \text{ } rotation\} \tag{B.10}$$

The same principles of stability and mobility have been applied in the robot turning movement. The difference here is that relative stability is determined by the angle (θ_{stable}) that the robot's centre of mass can be rotated whilst it is still enclosed by the stance legs triangle. On the other hand, mobility is defined by the allowed robot's legs rotation from their current position (θ_{mob}). The θ_{stable} angle is calculated by means of Eq. B.11.

$$\theta_{stable_i} = 2\arcsin\left(\frac{\sqrt{\left(x_c - x_{inters_i}\right)^2 + \left(y_c - y_{inters_i}\right)^2}}{2R_{p_i}}\right) \tag{B.11}$$

Where

- $i = 1,2,...,n_{turn}$ and n_{turn} is the number of elements of Δ_{turn}.
- (x_c, y_c) are the coordinates of the robot's centre of mass.
- (x_{p_i}, y_{p_i}) are the coordinates of the pivot leg.
- $R_{p_i} = \sqrt{\left(x_c - x_{\mathrm{p}_i}\right)^2 + \left(y_c - y_{\mathrm{p}_i}\right)^2}$ is the distance between the robot's centre of mass and the current coordinates of the pivot leg p_i.

- $\left(x_{inters_i}, y_{inters_i}\right)$ are the coordinates of the intersection point between the circle of radius R_{p_i} centred on the pivot leg, and the side of the stance legs triangle first intersected by the robot's centre of mass.

The θ_{mob} angle is determined using Eq. B.12

$$\theta_{mob_i} = \min\left(2\arcsin\left(\frac{1}{2}\sqrt{\frac{\left(x_k - x_{PEPk}\right)^2 + \left(y_k - y_{PEPk}\right)^2}{\left(x_k - x_{p_i}\right)^2 + \left(y_k - y_{p_i}\right)^2}}\right)\right) \tag{B.12}$$

Where

- $k = a,b \mid \{a,b\} = \Delta_{turn_i} - \{\mathrm{p_i}\}$ represents the legs specified by the i_{th} element of Δ_{turn} minus its respective pivot leg.
- (x_k, y_k) are the coordinates of the legs represented by k.
- $\left(x_{PEPk}, y_{PEPk}\right)$ are the leg k PEP coordinates, when this leg is moved over the circumference of radius $\sqrt{\left(x_k - x_{\mathrm{p}_i}\right)^2 + \left(y_k - y_{\mathrm{p}_i}\right)^2}$ and centred on the pivot leg.
- i and $\left(x_{p_i}, y_{p_i}\right)$ are the same as Eq. B.11.

Then, it is possible to determine the stance legs which allow maximum stable rotation by means of Eq. B.13.

$$stance\_legs = \Delta^* \mid \Delta^* \in \Delta_{turn} \wedge \forall \Delta_i \in \Delta_{turn}, \min\left(\theta^*_{stable}, \theta^*_{mob}\right) \geq \min\left(\theta_{stable_i}, \theta_{mob_i}\right) \tag{B.13}$$

Where $i = 1,2,\ldots,n_{turn}$.

A top view of a stance legs triangle performing a counterclockwise rotation is shown in Fig. B.7. The diagram also illustrates a number of parameters utilised by the equations discussed in this subsection. In this example, $k = 2,6$; the legs different from the pivot leg (leg 4). Besides, the θ_{mob} angle has been calculated using the

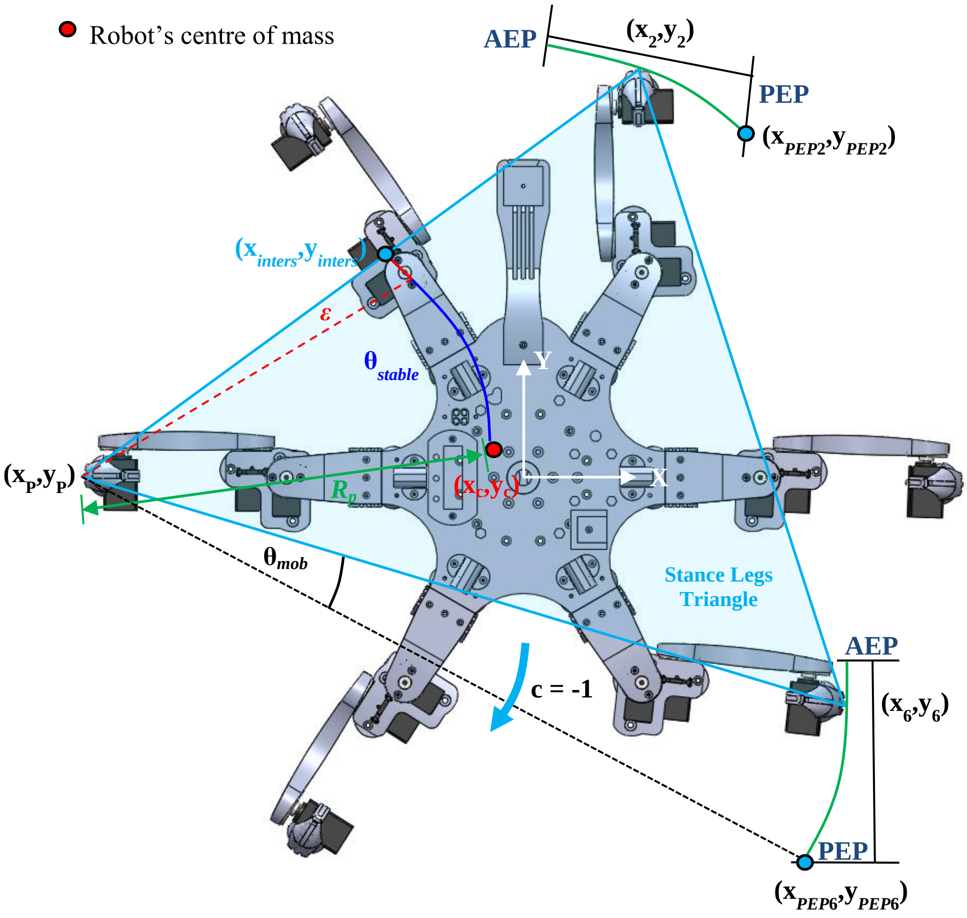


Figure B.7. Top View of a Stance Legs Triangle Performing a Turning Movement (Related Parameters are Included).

smallest leg mobility between the legs specified by k (the mobility of leg 6 in this case).

The final coordinates of the stance legs selected by means of expression B.13 and different from the pivot leg, which remains immobile, are determined utilising Eq. B.14.

$$\begin{bmatrix} x_{final\_stance\_coord_k} \\ y_{final\_stance\_coord_k} \end{bmatrix} = \cdots$$

$$\cdots = \begin{bmatrix} x_p^* + \sqrt{\left(x_k - x_p^*\right)^2 + \left(y_k - y_p^*\right)^2} \cos\left(\arctan\left(\frac{y_k}{x_k}\right) + c\left(\min\left(\theta_{stable}^*, \theta_{mob}^*\right) - \varepsilon\right)\right) \\ y_p^* + \sqrt{\left(x_k - x_p^*\right)^2 + \left(y_k - y_p^*\right)^2} \sin\left(\arctan\left(\frac{y_k}{x_k}\right) + c\left(\min\left(\theta_{stable}^*, \theta_{mob}^*\right) - \varepsilon\right)\right) \end{bmatrix} \text{(B.14)}$$

Where

- c is -1 if the rotation is clockwise, or 1 otherwise.
- ε has the same function as in Eqs. B.5 to B.7, but in this case it is an angle. $\varepsilon = 1^o$ has been used in this research.
- $\left(x_p^*, y_p^*\right)$ are the xy coordinates associated with the pivot leg of the Δ^* triangle.
- The desired angle of rotation should be used instead of $\min\left(\theta_{stable}^*, \theta_{mob}^*\right)$ when the former is less than the latter.

Finally, depending on the number of available legs, the swing leg's final coordinates are calculated by means of one of the expressions previously discussed in section B.1.

B.4. Rotation on the Spot

This type of movement may be utilised when it is necessary to rotate the robot a large angle in a reduced space or when a pure rotation movement is desired. In comparison with the turning movement, rotation on the spot allows a larger rotation but it lacks a straight line locomotion component.

Rotation on the spot is performed rotating the robot around its centre of mass. Once a stable stance legs polygon has been determined, this kind of rotation ensures stability independent of the rotation angle. Hence, the rotation is only limited by the leg's workspace.

B.4.1. Two Tripods Rotation

When there are six available legs and the angle of rotation is larger than the one achievable by rotating the current stance legs, a sequence of two or more tripod rotation could be employed. The suitability of this approach depends on the existence of the two stable stance triangles necessary for completing the movement. A parameter utilised in order to determine this is the angle θ_{rot_mob} that a stance legs triangle can be rotated only considering the legs' workspace. This angle is calculated for each element of Δ by means of expression B.15.

$$\theta_{rot\_mob\_\Delta_i} = \min\left(2\arcsin\left(\frac{\sqrt{\left(x_k - x_{EPk}\right)^2 + \left(y_k - y_{EPk}\right)^2}}{2R_k} \right)\right) \tag{B.15}$$

Where

- $\theta_{rot\_mob\_\Delta_i}$ is the θ_{rot_mob} angle associated with the i_{th} triangle belonging to Δ.
- $k = a,b,c \mid \{a,b,c\} = stance\_legs$ represents each one of the stance legs.
- (x_k, y_k) are the coordinates of the legs represented by k.
- $R_k = \sqrt{(x_k - x_c)^2 + (y_k - y_c)^2}$ is the distance between the robot's centre of mass and the current coordinates of the leg k.
- (x_{EPk}, y_{EPk}) are the leg k PEP or AEP coordinates when this leg is moved over the circumference of radius R_k and centred on the robot's centre of mass. Whether the PEP or AEP coordinates should be used depends on the direction in which the leg is rotating.

Then, one of the two tripods (if they exist) which allow the maximum rotation is the triangle formed by the stance legs. These are determined by means of expression B.16. On the other hand, the second tripod is the triangle formed by $L_{available}$ - $stance\_legs$.

$$\begin{aligned} &stance\_legs = \cdots \\ &\cdots = \Delta^* \mid \Delta^* \in \Delta \wedge s_*^c \theta_{rot\_mob\_\Delta^*} \neq 0 \wedge \forall \Delta_i \in \Delta, s_*^c \theta_{rot\_mob\_\Delta^*} \geq s_i^c \theta_{rot\_mob\_\Delta_i} \end{aligned} \tag{B.16}$$

Where

- $\theta_{rot\_mob\_\Delta^*}$ is the θ_{rot_mob} angle associated with the Δ^* triangle.
- s_i^c is 1 if $L_{available} - \Delta_i \in \Delta$, or 0 otherwise.
- s_*^c is 1 if $L_{available} - \Delta^* \in \Delta$, or 0 otherwise.

If none of the elements of Δ satisfies the conditions in Eq. B.16, this means that it is not possible to perform a rotation on the spot from the current position of the legs. When Eq. B.16 succeeds in selecting the stance legs triangle, the final stance legs coordinates are established by means of expression B.17.

$$\begin{bmatrix} x_{stance\_coord_k} \\ y_{stance\_coord_k} \end{bmatrix} = \begin{bmatrix} x_c + R_k \cos\left(\arctan\left(\frac{y_k}{x_k} \right) + c\theta_{rot\_mob\_\Delta^*} \right) \\ y_c + R_k \sin\left(\arctan\left(\frac{y_k}{x_k} \right) + c\theta_{rot\_mob\_\Delta^*} \right) \end{bmatrix} \tag{B.17}$$

Next, the final swing legs coordinates are their respective PEP or AEP when these legs are moved over the circumference of radius R_k and centred on the robot's centre of mass. If the desired robot rotation is clockwise, then AEP should be used for right legs and PEP for left legs. On the other hand, PEP for right legs and AEP for left legs should be used when the robot rotation is counterclockwise.

During the next stage of the two tripods rotation, swing and stance legs interchange roles. The legs that are on their PEP are moved towards their AEP, and vice versa. Always moving them over the circumference of radius R_k and centred on the robot's centre of mass. This cycle can continue until the desired angle of rotation has been reached or when this is small enough for using the rotation described in the next subsection instead.

B.4.2. Rotation with Fewer Legs and Small Angle Rotation

This subsection describes a mode of rotation on the spot, which does not require two tripods. Therefore, it is employed when fewer than six legs (and more than three) are available. The method is also suitable when the robot has six functional legs, if a rather small rotation, which can be achieved lifting only some of the robot's legs, is required. In some cases, the rotation can be performed maintaining all the legs on the terrain. Depending on the current position of the robot's legs and the desired rotation, this may also require one or several alternating sequences of rotational stance and swing phases. The order in which the stance and swing legs are selected affects the rotation stability, angle and energy consumption. Hence, the goal of the proposed method is to find the best order of stance and swing legs optimising these parameters.

The first step in the proposed algorithm is to calculate the available rotation of each leg ($\theta_{rot\_mob\_k}$) in the direction that produces the desired robot rotation (θ_d). This can be achieved if k in the "min" function argument of Eq. B.15 is replaced by each one of the robot's legs. Then, all the legs whose $\theta_{rot\_mob\_k}$ is less than the desired angle of rotation must be lifted to initiate their swing phase. When the rotation can be performed maintaining all the legs on the ground, the final coordinates of the legs are calculated by using Eq. B.17, after replacing $\theta_{rot\_mob\_\Delta^*}$ by θ_d. The final stance legs coordinates are established in the same way when only some of the robot's legs must be lifted and the remaining legs form a stable stance triangle. If this triangle is unstable or the total number of the robot's legs minus the number of legs that must begin their swing phase is less than three, additional processing is necessary. First of all, the order in which the legs are lifted is determined. For instance, if the robot has six legs and four of them must be lifted, considering that always three or more must be on the terrain, the operation would require at least two stance-swing phases. This could be done by means of a 2-2 sequence, meaning that two legs are lifted during a first stance-swing phase and the remaining two during a second one. However, 1-3 and 3-1 are also valid sequences. In addition, each sequence could be performed using different leg orders. For example, if the chosen sequence is 2-2 and the legs 1, 2, 3 and 4 must be lifted, this could be done by selecting legs 1 and 2, 1 and 3, 1 and 4, 2 and 3, 2 and 4 or 3 and 4 as swing legs during the first stance-swing phase and the two remaining legs during the second one. In general, the different orders in which the legs could initiate their swing phase during the first stance-swing phase are determined by means of expression B.18.

$$leg\_order = \left\{ l_{order\_1}, l_{order\_2}, ..., l_{order\_n} \right\} \qquad \text{(B.18)}$$

Where

- $l_{order\_i} = \{swing\_legs_{k_1}, swing\_legs_{k_2}, ..., swing\_legs_{seq1}\}$
- seq_1 indicates the number of legs that initiate their swing phase during the first stance-swing phase. (e.g. 3 if the sequence is 3-1).
- $i = 1, 2, ..., \frac{n_{swing}!}{\left(n_{swing} - seq_1\right)! seq_1!}$.
- $k_1 = 1, 2, ..., n_{swing}$.
- $k_2 = k_1 + 1, k_1 + 2, ..., n_{swing}$.
- $k_3 = k_2 + 1, k_2 + 2, ..., n_{swing}$.
- n_{swing} is the total number of swing legs (considering all the stance-swing phases).

All the swing legs not included in l_{order_i} initiate their swing phase during the second stance-swing phase.

Once all the possible orders in which the swing legs could initiate their swing phase has been established, the best leg order for each sequence must be determined considering stability, feasible rotation and energy consumption. Eq. B.19 can be used for determining which leg order has the better relation between stability and feasible rotation. The leg order with the maximum associated ms_{leg_order} value requires the fewest steps for performing a stable robot rotation and hence minimises energy consumption.

$$ms_{leg\_order_i} = \max\left(\frac{d_{stable\_poly_{A_i}}}{s_{A_i}\left(1 + \theta_{rot\_mob\_poly_{A_i}}\right)}, \frac{d_{stable\_poly_{B_i}}}{s_{B_i}\left(1 + \theta_{rot\_mob\_poly_{B_i}}\right)}\right) \tag{B.19}$$

Where

- i, as in Eq. B.18, represents each of the possible orders in which the swing legs can initiate their swing phase during a given sequence.

- The A_i (B_i) subscript indicates that the associated parameter must be calculated with respect to the first (second) stance-swing phase and the i_{th} element of leg_orders.
- d_{stable_poly} is the distance between the robot's centre of mass and the centroid of the polygon formed by the stance legs.
- s_i is 1 if the robot's centre of mass is enclosed by the polygon formed by the stance legs, or 0 otherwise.
- $\theta_{rot\_mob\_poly}$ is the available rotation of the polygon formed by the stance legs.

The possible sequences in which the swing legs initiate their swing phase are determined by means of expression B.20.

$$seqs = \left\{ seq_1, seq_2, ..., seq_{n_{seq}} \right\} \tag{B.20}$$

Where

- $seq_k = \{k_1, k_2\}$.
- $k_1 = n_{swing} - n_{legs} + 3, n_{swing} - n_{legs} + 2, ..., n_{legs} - 3$.
- $k_2 = n_{swing} - k_1$.
- $k = 1, 2, ..., n_{seq}$, with $n_{seq} = 2n - n_{swing} - 5$.

Then, once the set of possible leg orders is calculated for each sequence by means of Eq. B.18, the leg order that maximises the stability and the angle of rotation for each sequence is given by expression B.21.

$$\begin{aligned} best\_leg\_order_k = l_{order*} \mid l_{order*} \in leg\_order_k \wedge \forall l_{order\_i} \in leg\_order_k, \\ ms_{leg\_order*} \leq ms_{leg\_order_i} \end{aligned} \tag{B.21}$$

Where

- leg_order_k is the leg_order set associated with the k_{th} element of $seqs$.
- $ms^*_{leg\_order}$ is the ms_{leg_order} value associated with l_{order*}.

The criterion established by Eq. B.21 must be modified in order to consider energy consumption in the selection of the sequence employed to perform the robot's rotation. The energy required is related to the difference in the number of stance legs employed during both parts of the sequence. For instance, a 2-2 sequence is more energy efficient than a 3-1 sequence, even when in both cases four legs experienced a swing phase. Hence, the sequence that performs the robot rotation is finally determined by means of expression B.22.

$$\begin{aligned} &seq = s^* \mid s^* \in seqs \wedge \forall seq_k \in seqs, \\ &\quad ms_{best\_leg\_order*} + ms_{best\_leg\_order*} std\left(A_{swing\_legs*}, B_{swing\_legs*}\right) \leq \\ &\qquad ms_{best\_leg\_order_k} + ms_{best\_leg\_order_k} std\left(A_{swing\_legs_k}, B_{swing\_legs k}\right) \end{aligned} \tag{B.22}$$

Where

- $k = 1, 2, ..., n_{seq}$.
- $ms_{best\_leg\_order_k}$ is the ms_{leg_order} value associated with $best\_leg\_order_k$.
- $ms_{best\_leg\_order*}$ is the value associated with s^*.
- $std\left(A_{swing\_legs_k}, B_{swing\_legs_k}\right)$ is the standard deviation between the number of swing legs in the first and second stance-swing phase of seq_k, respectively.
- $std\left(A_{swing\_legs*}, B_{swing\_legs*}\right)$ is the standard deviation between the number of swing legs in the first and second stance-swing phase of s^*, respectively.

The stance-swing phase in which each swing leg initiate its swing phase is determined by the $best\_leg\_order$ set that corresponds with seq. Next, the final stance legs coordinates are calculated replacing $\theta^*_{rot\_mob\_\Delta}$ by $\theta^*_{rot\_mob\_poly}$, the available rotation angle of the polygon formed by the stance legs, in Eq. B.17. In addition, the swing legs are moved using the same strategy discussed in subsection B.4.1. Finally, all the steps described in this subsection can be repeated until the desired angle of rotation has been reached.

B.5. Experimental Results

This section shows experimental results of the gait generation described in subsections B.1.1 to B.1.4. The results are presented in tables, which specify the robot's stance legs (in black) and swing legs (in white) in each step (for the correspondence between leg number and leg position in the robot, please refer to Fig. B.1). All of the generated gaits are periodic. Therefore, the tables show all the steps belonging to one period. The generated gait has been obtained by considering that the robot's centre of mass is on the origin of the robot's coordinate system. Besides, it has been assumed that all the robot's legs are in their home position at the beginning of gait generation. Figures B.8 to B.13 show the experimental robot walking with different numbers of legs. In these figures, the position of the robot centroid after each gait cycle is represented by means of yellow asterisks and arrows.

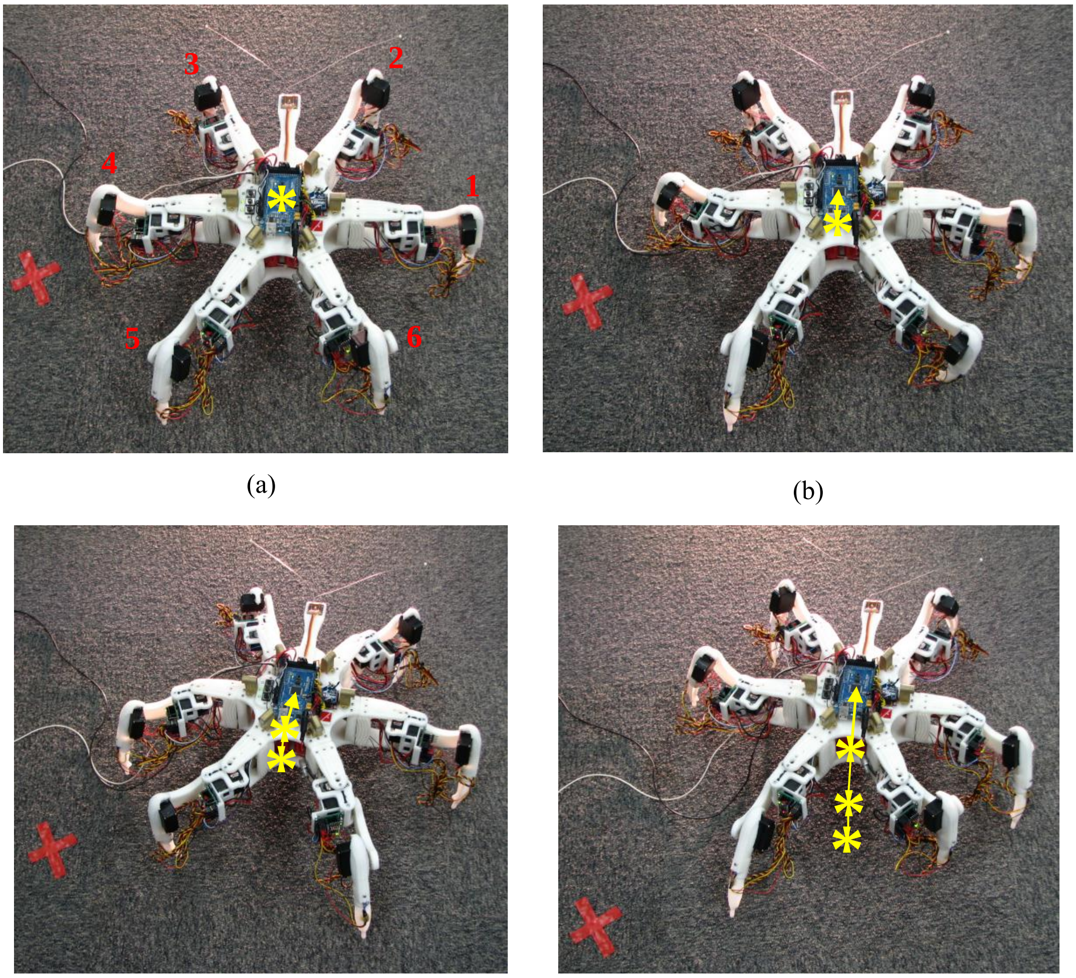
B.5.1. Six Legs Gait Generation Results

Table B.1 shows the generated gait for a six legged robot. The results show that a tripod gait has been obtained, which is commonly utilised by hexapod creatures.

Table B.1. Six Legs Gait Generation.

Step		1	2
Leg	1	■	□
	2	□	■
	3	■	□
	4	□	■
	5	■	□
	6	□	■

Figure B.8 shows the experimental robot walking with the tripod gait. In Fig. B.8(a), the robot is in its initial stand up position. Then, in Fig. B.8(b and d), legs 1, 3 and 5 act as the stance legs and are moved backwards whilst the rest of the legs, or swing legs, are moved forward. Conversely, in Fig. B.8(c), legs 2, 4 and 6 act as stance legs whilst legs 1, 3 and 5 perform the swing cycle.



(a)

(b)

(c)

(d)

Figure B.8. Experimental Robot and Six Legs Gait Generation.

B.5.2. Five Legs Gait Generation Results

Tables B.2 to B.7 contain the generated gait for the robot after it has lost one of its legs. Each table corresponds to the generated gait after a different robot leg has been shed.

Table B.2. Five Legs Gait Generation (Leg 1 Lost).

Step		1	2	3	4
Leg	2	■	□	■	□
	3	■	■	□	■
	4	□	■	■	■
	5	■	■	□	■
	6	□	■	■	■

Table B.3. Five Legs Gait Generation (Leg 2 Lost).

Step		1	2	3	4
Leg	1	■	□	■	□
	3	■	■	□	■
	4	□	■	■	■
	5	■	■	□	■
	6	□	■	■	■

Table B.4. Five Legs Gait Generation (Leg 3 Lost).

Step		1	2	1	2	3	4
Leg	1	▒	▒	■	□	■	□
	2	□	□	■	■	□	■
	4	▒	▒	□	■	■	■
	5	▒	▒	■	■	□	■
	6	□	▒	□	■	■	■

Table B.5. Five Legs Gait Generation (Leg 4 Lost).

Step		1	2	1	2	3	4
Leg	1	▒	▒	■	□	■	□
	2	□	□	■	■	□	■
	3	▒	▒	□	■	■	■
	5	▒	▒	■	■	□	■
	6	□	▒	□	■	■	■

Table B.6. Five Legs Gait Generation (Leg 5 Lost).

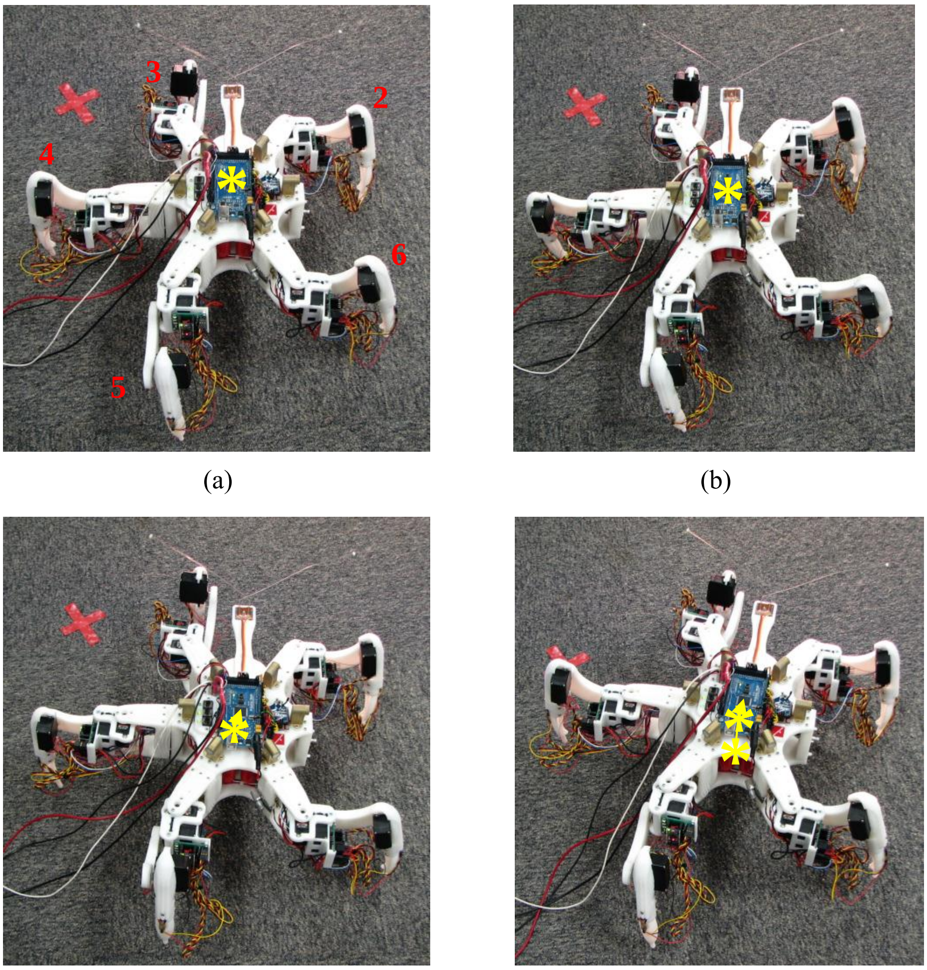
Step		1	2	1	2	3	4
Leg	1	▒	▒	■	□	■	□
	2	□	□	■	■	□	■
	3	▒	▒	□	■	■	■
	4	▒	▒	■	■	□	■
	6	□	▒	□	■	■	■

Table B.7. Five Legs Gait Generation (Leg 6 Lost).

Step		1	2	3	4
Leg	1	■	□	■	□
	2	□	■	■	■
	3	■	■	□	■
	4	□	■	■	■
	5	■	■	□	■

The results for the five legs gait generation show a periodic gait composed of 4 steps. During steps 2 and 4 only one swing leg is moved, the four stance legs stay still supporting the robot. Therefore, the robot is supported alternately by 3 and 4 legs. Tables B.4 to B.6 show cases where the robot takes two steps before actually starting the periodic gait. These are called *Initial Steps* due to the fact that they are taken only at beginning of the sequence of steps.

Figure B.9 shows the experimental robot walking with five legs, after leg 1 has been severed. In this case, the generated gait is contained in Table B.2. Those cases where there is only one swing leg, which correspond to steps 2 and 4 in Table B.2, are shown in Fig. B.9 (a and b). Here, leg 2 is shown during its swing cycle. In Fig. B.9



(a)

(b)

(c)

(d)

Figure B.9. Experimental Robot and Five Legs Gait Generation.

(c), legs 2, 3 and 5 act as the stance legs and are moved backwards whilst the rest of the legs, or swing legs, are moved forward. On the other hand, Fig. B.9 (d) shows the case where legs 2, 4 and 6 act as stance legs whilst legs 3 and 5 perform the swing cycle. Figure B.9 (c and d) correspond to steps 1 and 3 in Table B.2, respectively.

B.5.3. Four Legs Gait Generation Results

Tables B.8 to B.16 contain the generated gait for the robot after it has shed two of its legs. Each table corresponds to the generated gait for a different combination of 4 legs. The combinations that consider the loss of two legs on the same side of the robot are not included. This is because it is not possible to generate a stable gait with these leg configurations if the restrictions imposed by the leg workspace of the experimental robot are considered. A solution to this problem is to use the three legs gait generation, but with four legs instead of three.

Table B.8. Four Legs Gait Generation (Leg 1 and 3 Lost).

Step		1	2	3	4	1	2	3	4
Leg	2	▨	▨	▨	□	■	□	■	■
	4	▨	▨	□	▨	■	■	■	□
	5	▨	□	▨	▨	■	■	□	■
	6	□	▨	▨	▨	□	■	■	■

Table B.9. Four Legs Gait Generation (Leg 1 and 4 Lost).

Step		1	2	3	4	1	2	3	4
Leg	2	▨	▨	□	▨	■	□	■	■
	3	▨	▨	▨	□	■	■	■	□
	5	▨	□	▨	▨	■	■	□	■
	6	□	▨	▨	▨	□	■	■	■

Table B.10. Four Legs Gait Generation (Leg 1 and 5 Lost).

Step		1	2	3	4	1	2	3	4	1	2	3	4
Leg	2	▨	▨	□	▨	▨	▨	▨	▨	□	■	■	■
	3	▨	▨	▨	□	▨	▨	▨	▨	■	□	■	■
	4	▨	□	▨	▨	▨	□	▨	□	■	■	□	■
	6	□	▨	▨	▨	□	▨	□	▨	■	■	■	□
		Repeated 4 times											

Table B.11. Four Legs Gait Generation (Leg 2 and 3 Lost).

Step		1	2	3	4	1	2	3	4
Leg	1	▒	▒	▒	□	■	■	■	□
	4	▒	▒	□	▒	■	□	■	■
	5	▒	□	▒	▒	□	■	■	■
	6	□	▒	▒	▒	■	■	□	■

Table B.12. Four Legs Gait Generation (Leg 2 and 4 Lost).

Step		1	2	3	4	1	2	1	2	3	4
Leg	1	▒	▒	□	▒	▒	▒	■	□	■	■
	3	▒	▒	▒	□	▒	▒	■	■	□	■
	5	▒	□	▒	▒	▒	□	■	■	■	□
	6	□	▒	▒	▒	□	▒	□	■	■	■
		Repeated 4 times									

Table B.13. Four Legs Gait Generation (Leg 2 and 5 Lost).

Step		1	2	3	4	1	2	1	2	3	4
Leg	1	▒	▒	□	▒	▒	▒	■	□	■	■
	3	▒	▒	▒	□	▒	▒	■	■	□	■
	4	▒	□	▒	▒	▒	□	■	■	■	□
	6	□	▒	▒	▒	□	▒	□	■	■	■
		Repeated 4 times									

Table B.14. Four Legs Gait Generation (Leg 3 and 6 Lost).

Step		1	2	3	4	1	2	1	2	3	4
Leg	1	▒	□	▒	▒	▒	□	■	■	■	□
	2	▒	▒	▒	□	▒	▒	■	■	□	■
	4	▒	▒	□	▒	▒	▒	■	□	■	■
	5	□	▒	▒	▒	□	▒	□	■	■	■
		Repeated 4 times									

Table B.15. Four Legs Gait Generation (Leg 4 and 6 Lost).

Step		1	2	3	4	1	2	1	2	3	4
Leg	1	▒	□	▒	▒	▒	□	■	■	■	□
	2	▒	▒	▒	□	▒	▒	■	■	□	■
	3	▒	▒	□	▒	▒	▒	■	□	■	■
	5	□	▒	▒	▒	□	▒	□	■	■	■
		Repeated 4 times									

Table B.16. Four Legs Gait Generation (Leg 5 and 6 Lost).

Step		1	2	3	1	2	3	4
Leg	1	▨	□	▨	■	□	■	■
	2	▨	▨	□	■	■	□	■
	3	▨	▨	▨	□	■	■	■
	4	□	▨	▨	■	■	■	□

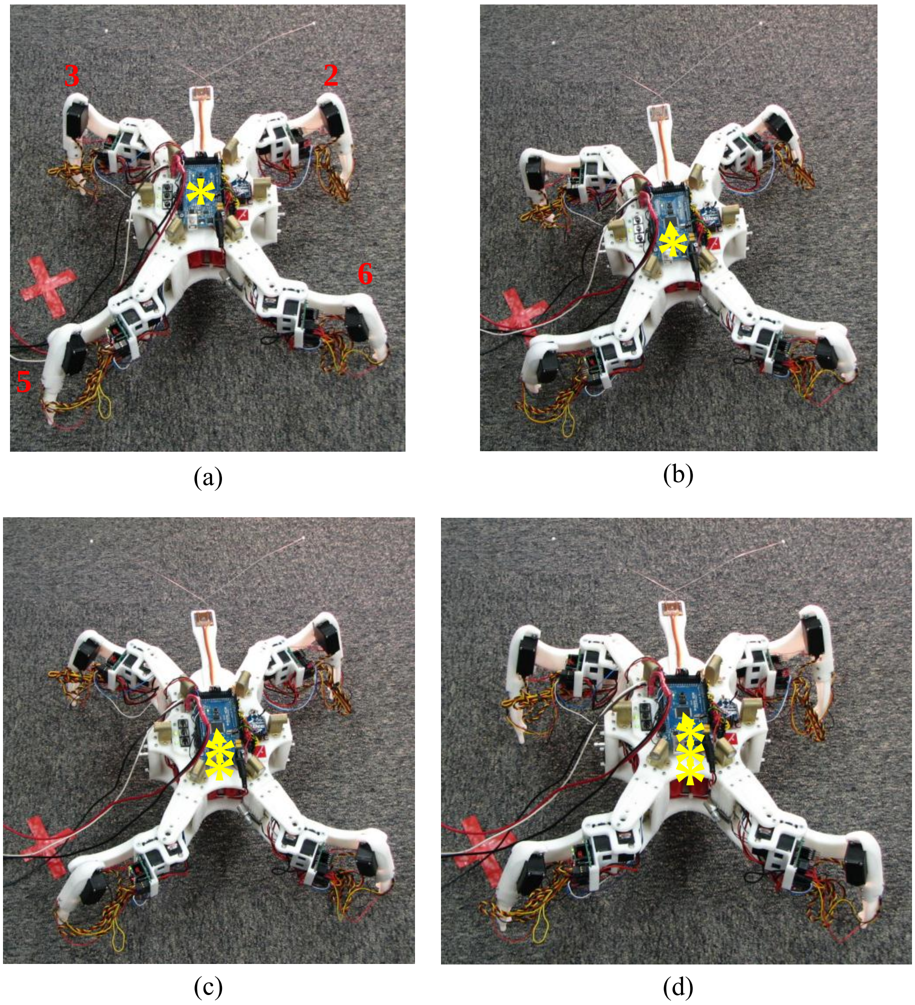
The results for the four legs gait generation show a periodic gait composed of 4 steps. In this case, the *initial steps* are always generated at the beginning of the gait, regardless of the combination of severed legs. They allow the stable transition from the home position of the robot's legs to the position established by the periodic gait. The number of *initial steps* varies with the robot leg configuration. The following four numbers of *initial steps* were present during the experiment: 3 (Table B.16), 4 (Tables B.8, B.9 and B.11), 18 (Tables B.12, B.13, B.14 and B.15) and 20 (Table B.10). Figure B.10 shows the experimental robot walking with four legs, after legs 1 and 4 have been severed. In this case, the generated gait is contained in Table B.2. This gait has four associated initial steps, which are shown in Figs. B.10(a-d).

B.5.4. Three Legs Gait Generation Results

The results for the three legs gait generation when the experimental robot walks with different combinations of three legs are shown in Table B.17. As expected, the generated gait is periodic and it is composed of 2 steps.

Table B.17. Three Legs Gait Generation
(Experimental Robot Walking with Legs L_A, L_B and L_C).

Step		1	2
Leg	L_A	■	□
	L_B	■	□
	L_C	■	□



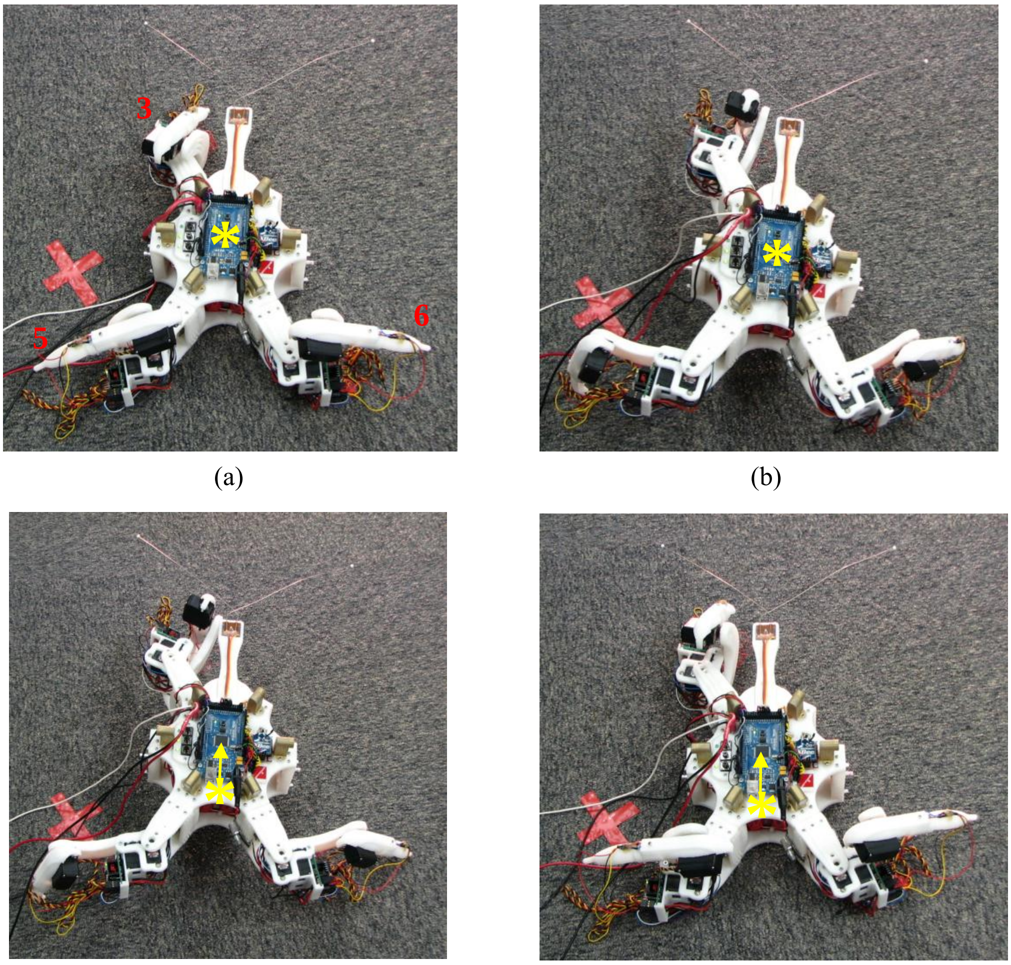
(a) (b)

(c) (d)

Figure B.10. Experimental Robot and Four Legs Gait Generation.

The combinations that consider the loss of three adjacent legs are not included. This is similar to the four legs gait generation, when two legs were lost on the same side of the robot. In these cases, it is not possible to generate a stable gait with these leg configurations if the restrictions imposed by the leg workspace of the experimental robot are considered. A solution to this problem is to use the two legs gait generation, but with three legs instead of two.

Figure B.11 shows the experimental robot walking with three legs, after legs 1, 2 and 4 have been severed. Therefore, $\{L_A, L_B, L_C\} = \{3,5,6\}$ in Table B.2. Figure B.11(a) shows the corresponding home position for this leg configuration. In addition, Figs. B.11(b and c) show the robot's legs performing the stance cycle (step 1 in Table



(a) (b)

(c) (d)

Figure B.11. Experimental Robot and Three Legs Gait Generation.

B.17). On the other hand, Fig. B.11(d) shows the robot's legs performing the swing cycle (step 2 in Table B.17).

B.5.5. Two Legs Gait Generation Results

The results for the two legs gait generation, where the experimental robot drags its body with different combinations of two legs, are shown in Table B.18. Once again, the generated gait is periodic and it is composed of 2 steps.

Table B.18. Two Legs Gait Generation.
(Experimental Robot Dragging Its Body with Legs L_A and L_B).

Step		1	2
Leg	L_A	■	
	L_B	■	

Figure B.12 shows the experimental robot dragging its body with two legs, after legs 1, 2, 3 and 4 have been severed. Therefore, $\{L_A, L_B\} = \{5, 6\}$ in Table B.18. Figure B.12(a) shows the corresponding home position for this leg configuration. As a consequence of the missing legs, the robot has a backward locomotion. In general, the direction of movement will vary as it depends on the robot’s leg configuration. Figure B.12(b) shows the robot’s legs performing the stance cycle (step 1 in Table B.18). Here, the robot is lifted and dragged towards the direction of the light source. Finally, Fig. B.12 (c and d) show the robot’s legs performing the swing cycle (step 2 in Table B.17). In this cycle, the legs are first lifted (Fig. B.12(c)) and then stretched (Fig. B.12(d)). Whilst the legs are extended, they are gradually lowered until they touch the floor and the stance cycle begins again.

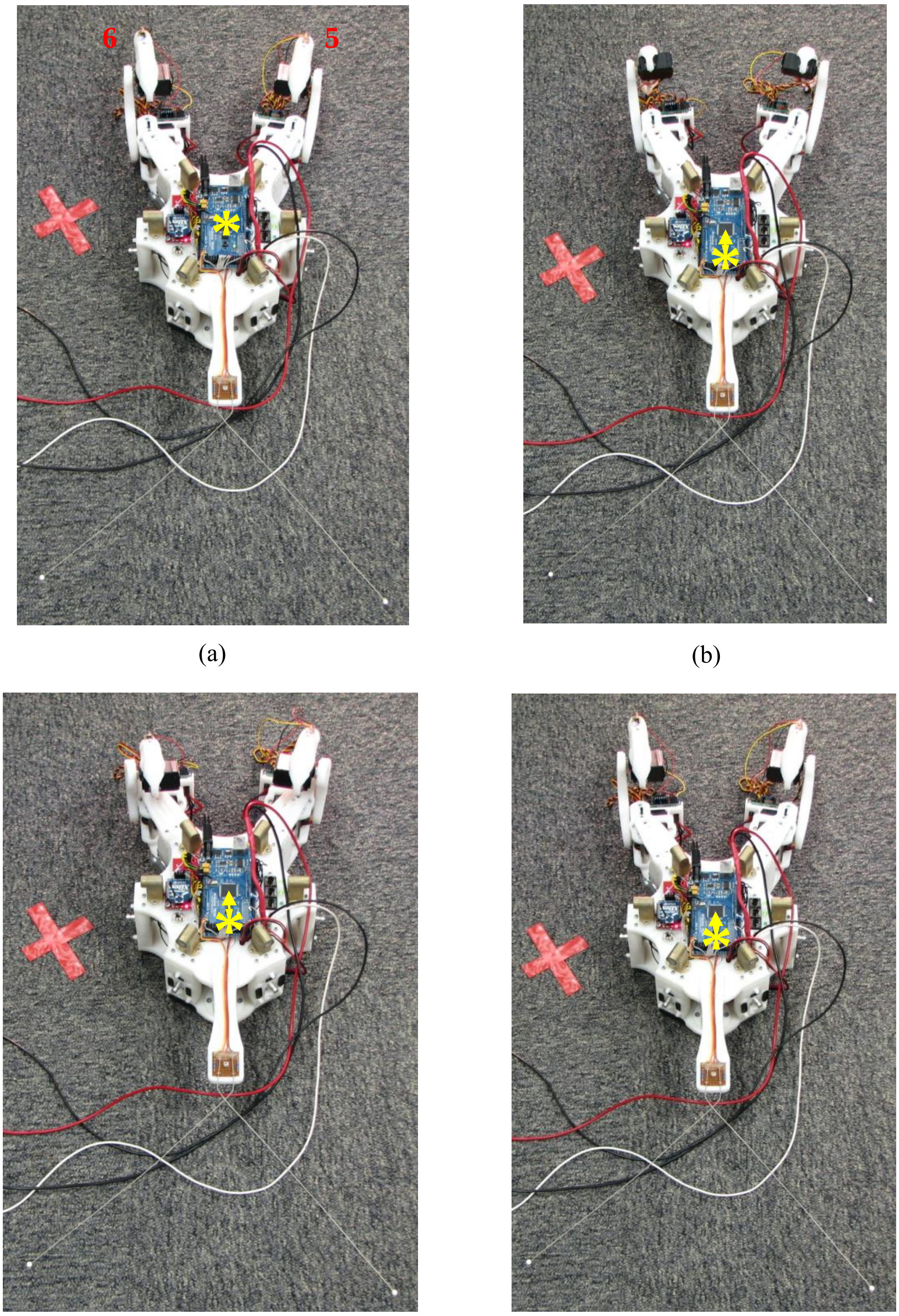
B.5.6. One Leg Gait Generation Results

The results for the one leg gait generation, where the experimental robot drags its body with a single leg, are shown in Table B.19. As expected, the generated gait is periodic and it is composed by 2 steps.

Table B.19. One Leg Gait Generation.
(Experimental Robot Dragging Its Body with Leg L_A).

Step		1	2
Leg	L_A	■	

Figure B.13 shows the experimental robot dragging its body with only one leg, after legs 1, 2, 3, 4 and 5 have been severed. Therefore, $L_A = 6$ in Table B.18. Figure B.13(a) shows the corresponding home position for this leg configuration. Once again, the robot has a backward locomotion as a consequence of the missing legs.

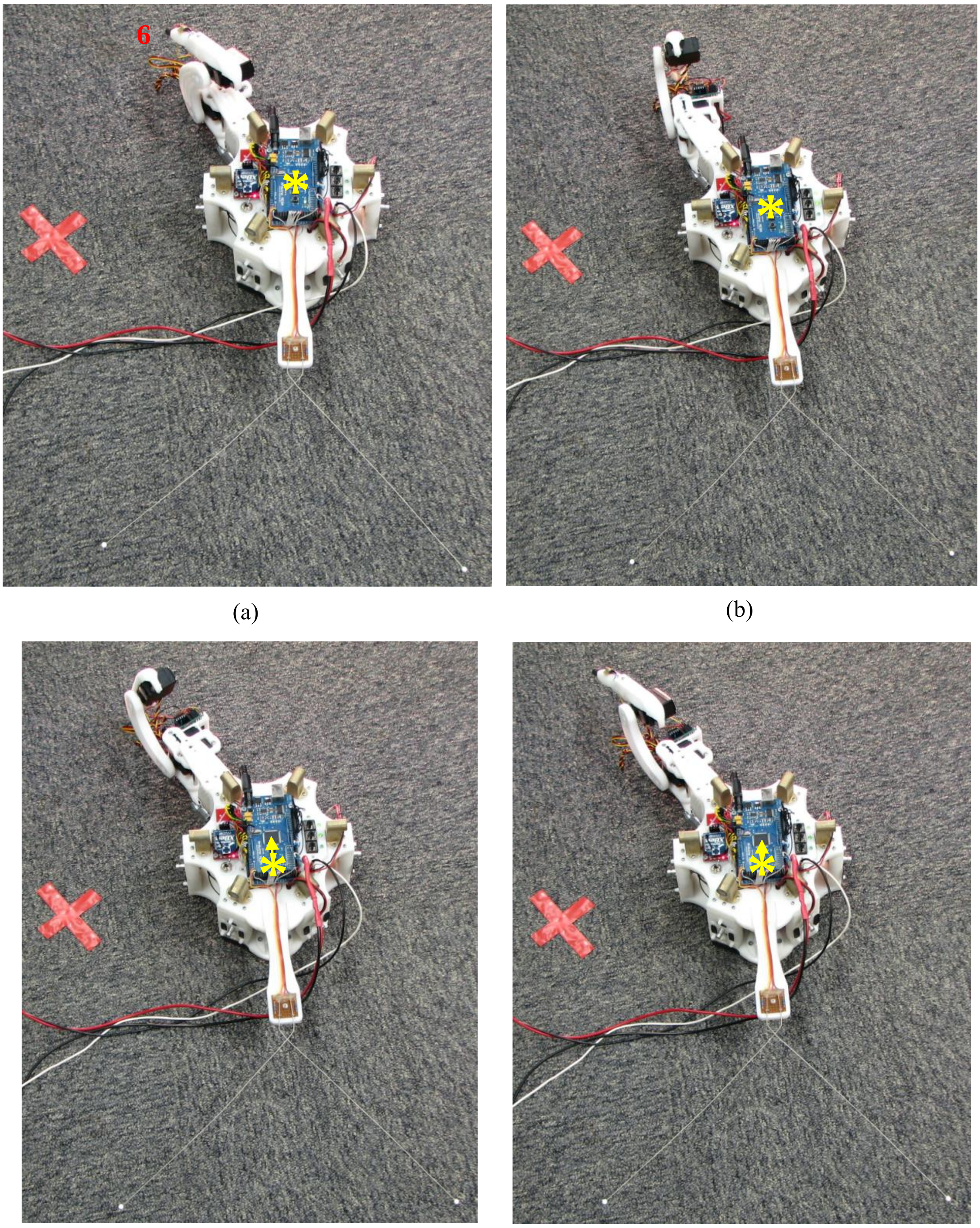


(a) (b)

(c) (d)

Figure B.12. Experimental Robot and Two Legs Gait Generation.

In general, the direction of movement will vary as it depends on the robot's leg configuration. Figure B.13(b) shows the robot's leg performing the stance cycle (step 1 in Table B.19). Here, the robot is lifted and dragged towards the direction of the light source in Table B.19). In this cycle, the leg is first lifted (Fig. B.13(c)) and then



(a)

(b)

(c)

(d)

Figure B.13. Experimental Robot and One Leg Gait Generation.

stretched (Fig. B.13(d)). Whilst the leg is extended, it is gradually lowered until it touches the floor and the stance cycle begins again.

This section has shown the generation of different stable gaits using the theory explained in section B.1. All the generated gaits are periodic, but the number of steps of each period depends on the available number of robot's legs. Experimental results show that the proposed approach generates a stable gait from almost any leg configuration, which includes between 3 and 6 legs. In addition, those cases where the robot is unable to walk in a stable fashion are covered by gaits that involve dragging the robot's body. Table B.20 shows the average distance per gait cycle covered by the experimental robot. This data shows that symmetrical stable gaits (six and four legs gait generation) cover more distance per gait cycle in comparison with the asymmetrical stable gait (five legs gait generation). On the other hand, unstable gaits (one, two and three legs gait generation) show only slight differences in their distance per gait cycle. However, the trend indicates that the more legs are used the more distance is cover per gate cycle. This can be attributed to sliding effects on the robot's leg(s), which are increased as fewer legs drag the robot during the stance phase.

Table B.20. Average Distance per Gait Cycle.

Number of Legs	1	2	3	4	5	6
Average Distance per Gait Cycle [mm]	60	65	67	107	99	225

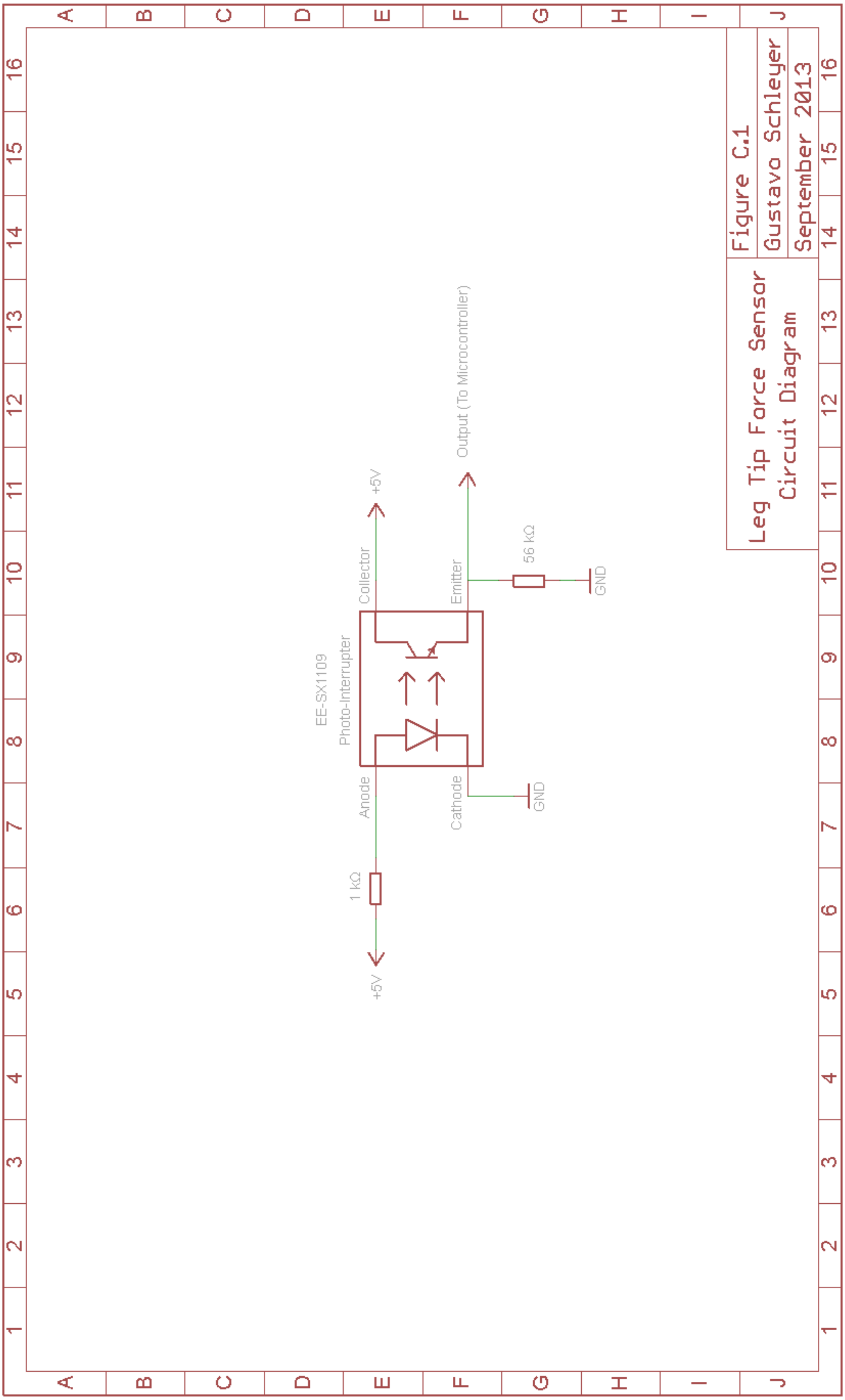
B.6. Summary

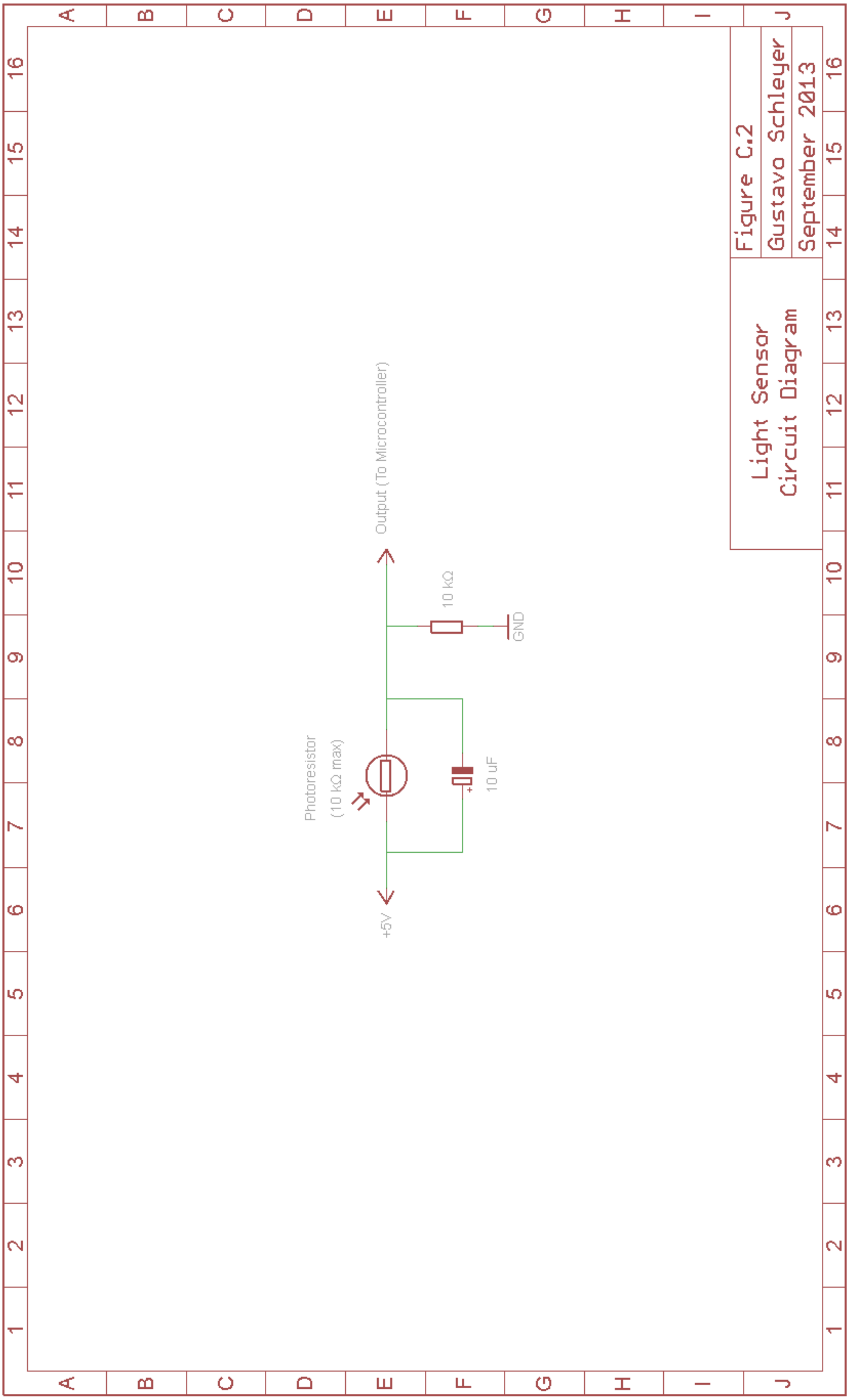
Appendix B has presented an adaptable gait generation method that allows damaged or autotomised robots to walk with different combinations of leg losses. The proposed technique maximises the stable mobility of robots once the position of their centre of mass has been calculated. This means that if the position of the robot's centre of mass changes, either because the robot is carrying a load or its hardware configuration has been altered, by using this method the target robot will still be able to generate a stable gait. Instead of proposing a different gait generation method for each possible number of functional legs, slight modifications of a six legs gait generation technique are used. This allows the extension of the method to other leg configuration and to produce stable gaits when fewer functional legs are available. The proposed technique also covers those cases where the robot is unable to walk in a stable fashion. In this kind of situations, locomotion is achieved by dragging the robot’s body.

Finally, the stable gaits introduced in this appendix (used when the robot has between 4 and 6 legs) generate straight line locomotion in any desired direction, which allows the robot to modify its direction of locomotion without actually rotating the robot. This could be useful when is necessary to manoeuvre the robot in restricted spaces.

Appendix C
Robot Circuit Diagrams

This appendix contains electronic circuit diagrams for a selection of the experimental robot hardware components. The design and construction of the experimental robot electronics was part of the work executed during this investigation. In this appendix, circuit diagrams corresponding to leg tip force sensor, light sensor, whiskers, control panel and leg PCB were drawn by using CadSoft Eagle (Easily Applicable Graphical Layout Editor). These circuit diagrams are shown in Figs. C.1 to C.5.


Leg Tip Force Sensor
Circuit Diagram
Figure C.1
Gustavo Schleyer
September 2013
EE-SX1109
Photo-Interrupter
Anode
Cathode
Collector
Emitter
+5V
+5V
1 kΩ
56 kΩ
GND
GND
Output (To Microcontroller)
A
B
C
D
E
F
G
H
I
J
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16



Light Sensor
Circuit Diagram

Figure C.2
Gustavo Schleyer
September 2013

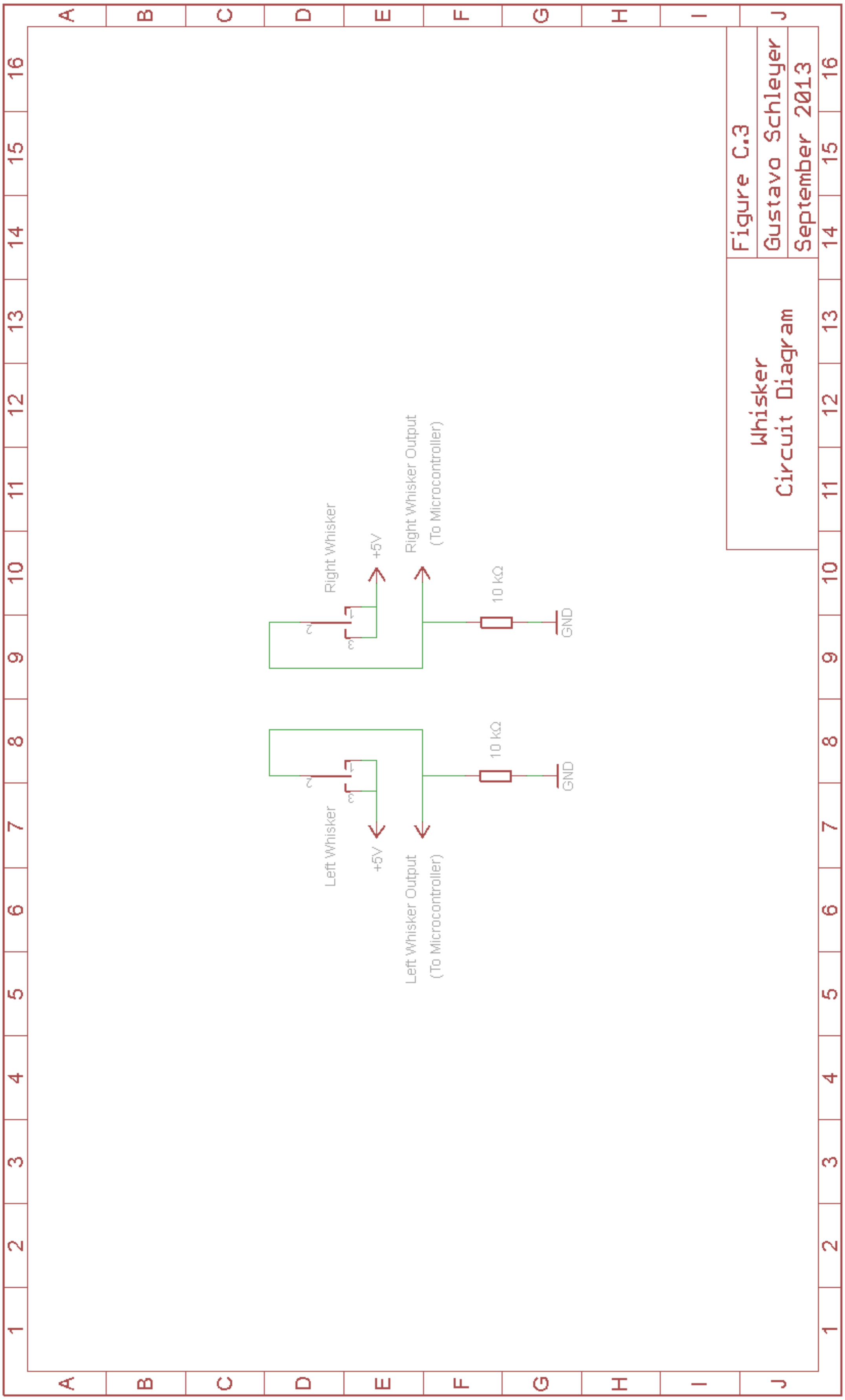


Figure C.3
Gustavo Schleyer
September 2013

Whisker
Circuit Diagram

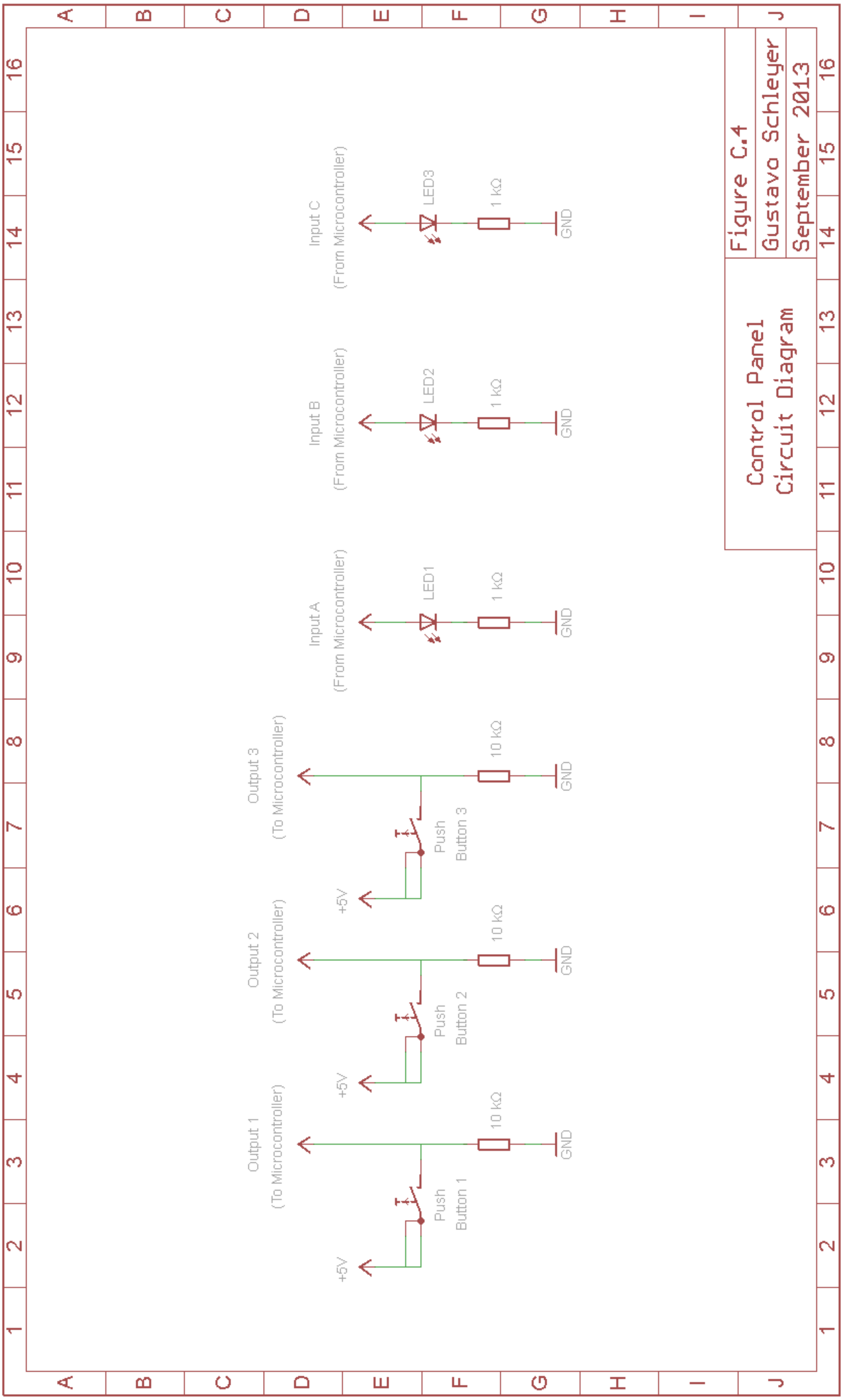


Figure C.4
Gustavo Schleyer
September 2013
Control Panel
Circuit Diagram
+5V
Push
Button 1
Output 1
(To Microcontroller)
10 kΩ
GND
+5V
Push
Button 2
Output 2
(To Microcontroller)
10 kΩ
GND
+5V
Push
Button 3
Output 3
(To Microcontroller)
10 kΩ
GND
Input A
(From Microcontroller)
LED1
1 kΩ
GND
Input B
(From Microcontroller)
LED2
1 kΩ
GND
Input C
(From Microcontroller)
LED3
1 kΩ
GND

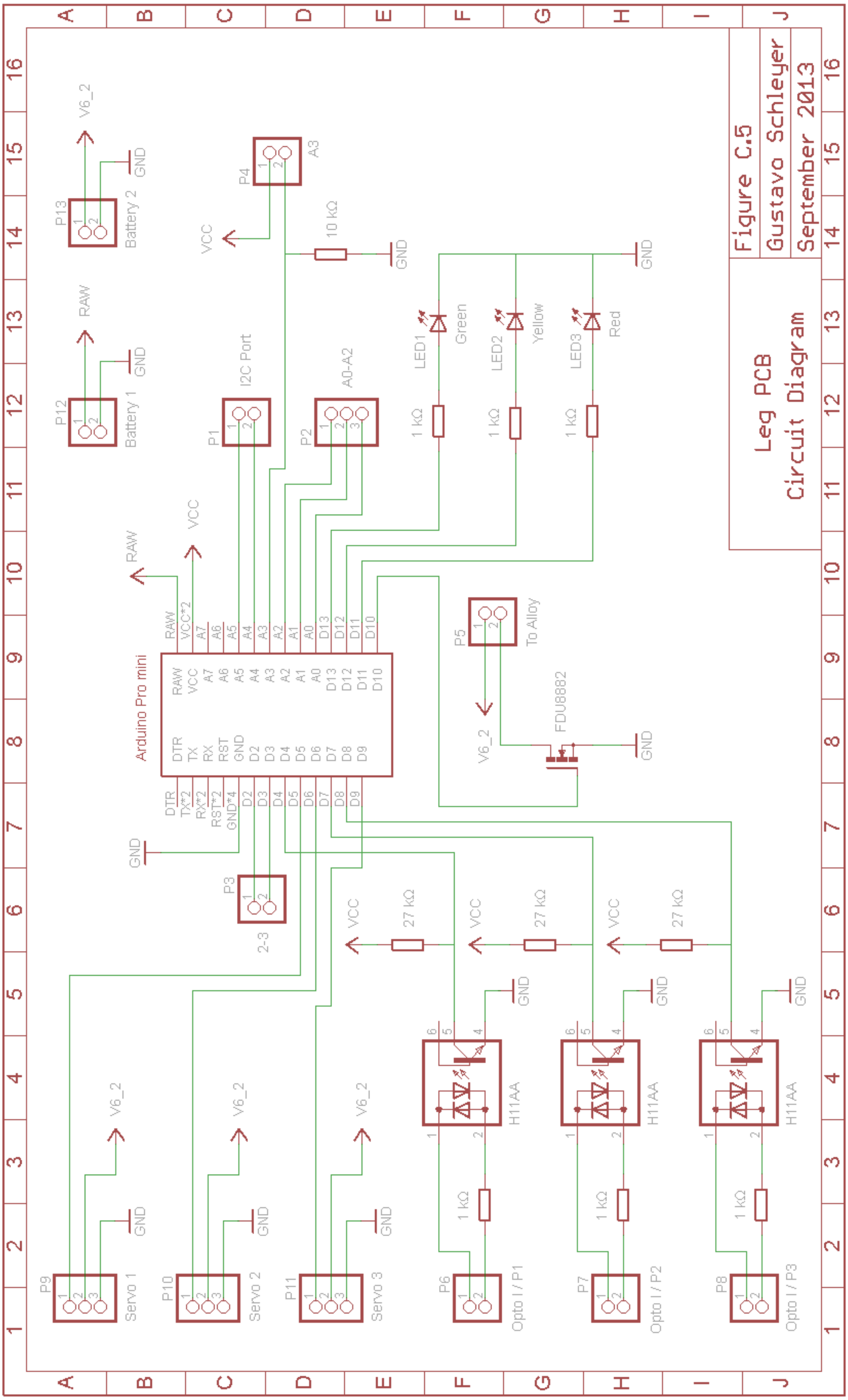


Figure C.5
Gustavo Schleyer
September 2013
Leg PCB
Circuit Diagram
Arduino Pro mini
RAW
VCC
A7
A6
A5
A4
A3
A2
A1
A0
D13
D12
D11
D10
DTR
TX
RX
RST
GND
D2
D3
D4
D5
D6
D7
D8
D9
VCC*2
TX*2
RX*2
RST*2
GND*4
P1
I2C Port
P2
A0-A2
P3
2-3
P4
A3
P5
To Alloy
FDU8882
P6
Opto I / P1
P7
Opto I / P2
P8
Opto I / P3
P9
Servo 1
P10
Servo 2
P11
Servo 3
P12
Battery 1
P13
Battery 2
LED1
Green
LED2
Yellow
LED3
Red
1 kΩ
10 kΩ
27 kΩ
H11AA
V6_2
RAW
VCC
GND

Appendix D
External Links

This appendix contains a list of links to videos of the experimental robot executing basic movements, research actions and compensation of abnormal situations.

Link D01: Turning Movement.
This video shows the experimental robot performing a turning movement, a combined translation and rotation to the right.

https://www.dropbox.com/s/noch3chvltxp7ur/D01%20Turning%20Movement.mpg?dl=0

Link D02: Rotation on the Spot.
This video shows the experimental robot performing a clockwise rotation on the spot.

https://www.dropbox.com/s/ps5ibivo473mn3x/D02%20Rotation%20on%20the%20Spot.mpg?dl=0

Link D03: Leg Release Mechanism.
This video shows the leg release mechanism in action. The mechanism is activated at 00:08 minutes and leg separation occurs at 03:01 minutes.

https://www.dropbox.com/s/vmijhj4hoy3eh1x/D03%20Leg%20Release%20Mechanism.mpg?dl=0

Link D04: Accelerometer Research Action.

This video shows the experimental robot performing the accelerometer research action.

https://www.dropbox.com/s/pac2wj7jtzl2wuf/D04%20Accelerometer%20Research%20Action.mpg?dl=0

Link D05: Faulty Light Sensor Compensation - Method 0.

This video shows an experiment where one of the robot's light sensors (light sensor 2) has been fully covered. As a result, the robot is unable to correctly calculate the direction of the light source and a cyclic lack of progress is detected. Then, the robot performs a sweep action in order to clean the covered light sensor. This time, the robot successfully uncovers the sensor and is able to continue with its mission in a normal fashion.

https://www.dropbox.com/s/lhsz30mhyflmsid/D05%20Faulty%20Light%20Sensor%20Compensation%20-%20Method%200.mpg?dl=0

Link D06: Faulty Light Sensor Compensation - Method 1.

This video shows another experiment where one of the robot's light sensors (light sensor 2) has been fully covered. Once again, the robot is unable to correctly calculate the direction of the light source and a cyclic lack of progress is detected. Then, the robot performs a sweep action in order to clean the covered light sensor. In this case, the robot fails to uncover the sensor and after resuming the mission a new cyclic lack of progress is detected. This time, the robot compensates the abnormal situation by changing its direction of locomotion by 180º. Therefore, the robot continues with its mission by walking backwards towards the light source.

https://www.dropbox.com/s/81mr2din6oe4p3p/D06%20Faulty%20Light%20Sensor%20Compensation%20-%20Method%201.mpg?dl=0

Link D07: Faulty Light Sensor Compensation - Method 2.

This new video shows an experiment where light sensors 2, 4 and 6 have been fully covered. As a result, the robot is unable to correctly calculate the direction of the light source and a cyclic lack of progress is detected. Then, the robot performs a sweep action in order to clean the covered light sensor. In this experiment, the robot fails to uncover the sensor and after resuming the mission a new cyclic lack of progress is detected. This time, the robot fails to find two adjacent uncovered light sensors. Then, the robot compensates the abnormal situation by changing its direction of locomotion by 30º. Therefore, the robot continues with its mission by walking diagonally towards the light source.

https://www.dropbox.com/s/adjawmrw1eehhwe/D07%20Faulty%20Light%20Sensor%20Compensation%20-%20Method%202.mpg?dl=0

Link D08: Soft Terrain Compensation.

This video shows an experiment where leg 3 of the experimental robot walks over soft terrain. As a result, the robot is unable to move forward and a cyclic lack of progress alarm is triggered. Then, the robot performs position sensor research actions and nothing abnormal is detected. Once the robot has discarded the possibility that the abnormal sensor readings are due to intrinsic anomalies, it searches for an abnormal situation, similar to the one currently experienced, in its database of anomalies. Finally, the abnormal situation is identified and compensatory actions corresponding to soft terrain are executed.

https://www.dropbox.com/s/lth6emwrks0bs40/D08%20Soft%20Terrain%20Compensation.mpg?dl=0

Link D09: Low Power Compensation.

This video shows an experiment where the robot walks towards the light source with battery Bl_2 almost completely discharged. The robot manages to take some steps towards the light source but soon it is unable to move further. Then, a cyclic lack of progress alarm is triggered. The robot is able to identify that the abnormal situation was generated by a low power condition and a distress signal (represented by the flashing green LEDs) is sent.

https://www.dropbox.com/s/6npxodsa4704j7i/D09%20Low%20Power%20Compensation.mpg?dl=0

Link D10: Five Legs Gait Generation.

This video shows the experimental robot walking with 5 legs. In this case, leg 5 has been disabled by positioning it in its home position so it does not interfere with the robot's locomotion.

https://www.dropbox.com/s/9mbkegffe19vnov/D10%20Five%20Legs%20Gait%20Generation.mpg?dl=0

Link D11: Frontal Collision Compensation

This video shows the compensatory measure taken by the experimental robot after colliding with a transparent obstacle. The collision is detected by the robot's whiskers. Then the robot changes its direction of locomotion and is able to continue moving towards the light source.

https://www.dropbox.com/s/zwi56banvzlmkeg/D11%20Obstacle%20Avoidance.mpg?dl=0

Link D12: Lights Off Compensation.

This video shows the experimental robot performing the lights off innate compensatory measure. Initially, the robot walks normally towards the light source until the lights are switched off. Then, the robot performs a sweeping movement in an attempt to clean all of its light sensors. When the robot realises that this action does not improve the situation, it decides to wait until a new light source appears. After a short time, lights are switched on again. This is detected by the robot and the mission is resumed.

https://www.dropbox.com/s/0kuk8wx383py0lt/D12%20Lights%20Off%20Compensation.mpg?dl=0

References

[1] K. Lorenz, Evolution and Modification of Behavior, Chicago: Chicago University Press, 1965.

[2] J. W. Kimball, “Kimball's Biology Pages,” 2013. [Online]. Available: http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/.

[3] C. Darwin, On the Origin of the Species, London, 1859.

[4] D. Purves, G. J. Augustine, D. Fitzpatrick, W. C. Hall, A.-S. LaMantia, J. O. McNamara and S. M. Williams, Neuroscience 3rd Edition, Massachusetts, USA: Sinauer Associates, Inc., 2004.

[5] P. Fleming, D. Muller and P. Bateman, “Leave It All Behind: A Taxonomic Perspective of Autotomy in Invertebrates,” *Biological Reviews of the Cambridge Philosophical Society,* vol. 83, no. 3, pp. 481-510, 2007.

[6] R. Murphy and D. Hershberger, “Classifying and Recovering from Sensing Failures in Autonomous Mobile Robots,” *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI'96),* vol. 2, August 1996.

[7] R. Lindsay, B. Buchanan, E. Feigenbaum and J. Lederberg, “Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project,” *New York, McGraw-Hill,,* 1980.

[8] R. Murphy, “Biological and Cognitive Foundations of Intelligent Sensor Fusion,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,* vol. 26, pp. 42-51, 1996.

[9] R. Murphy and D. Hershberger, “Handling Sensing Failures in Autonomous Mobile Robots,” *The International Journal of Robotics Research,* vol. 18, pp.

382-400, 1999.

[10] C. Laurentys, G. Ronacher, R. Palhares and W. Caminhas, “Design of an Artificial Immune System for Fault Detection: A Negative Selection Approach,” *Expert Systems with Applications,* vol. 37, no. 7, pp. 5507-5513, July 2010.

[11] B. Jakimovski and E. Maehle, “Artificial Immune System Based Robot Anomaly Detection Engine for Fault Tolerant Robots,” *5th International Conference on Autonomic and Trusted Computing (ATC-08),* pp. 177-190, 2008.

[12] B. Jakimovski, “Biologically Inspired Approaches for Anomaly Detection within a Robotic System,” *Biologically Inspired Approaches for Locomotion, Anomaly Detection and Reconfiguration for Walking Robots Cognitive Systems Monographs,* vol. 14, pp. 127-150, 2011.

[13] L. De Castro and F. Von Zuben, “Learning and Optimization using the Clonal Selection Principle,” *IEEE Transaction on Evolutionary Computation,* pp. 239-251, 2002.

[14] A. Johnson, G. Haynes and D. Koditschek, “Disturbance Detection, Identification, and Recovery by Gait Transition in Legged Robots,” *Proceedings of the International Conference on Intelligent Robots and Systems (IROS),* pp. 5347-5353, 2010.

[15] A. Vitko, L. Jurišica, F. Duchoň, A. Babinec, M. Dekan, D. Kaštan, M. Klúčik and M. Bachratý, “Simultaneous Navigation and Fault Detection of Legged Robot,” *Proceedings of the 2nd International Conference on Robotics in Education (RiE),* pp. 91-94, 2011.

[16] A. Vitko, L. Jurišica, F. Duchoň, A. Babinec, M. Dekan, D. Kaštan and M. Kľúčik, “Navigation and Diagnosis of Walking Robot,” *Journal of Mechanics Engineering and Automation,* vol. 2, pp. 568-572, 2012.

[17] A. L. Christensen, R. O'Grady, M. Birattari and M. Dorigo, “Exogenous Fault Detection in a Collective Robotic Task,” *Proceedings of the 9th European*

Conference on Artificial Life (ECAL2007). Volume LNAI 4648 of Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, Germany, pp. 555-564, 2007.

[18] A. L. Christensen, R. O'Grady, M. Birattari and M. Dorigo, "Automatic Synthesis of Fault Detection Modules for Mobile Robots," *Proceedings of NASA/ESA Conference on Adaptive Hardware and Systems (AHS-2007), IEEE Computer Society Press, Los Alamitos, CA,* pp. 693-700, 2007.

[19] A. L. Christensen, R. O'Grady, M. Birattari and M. Dorigo, "Fault Detection in Autonomous Robots Based on Fault Injection and Learning," *Autonomous Robots,* vol. 24, no. 1, pp. 49-67, 2008.

[20] D. Tarapore, A. Christensen, P. Lima and J. Carneiro, "Abnormality Detection in Multiagent Systems Inspired by the Adaptive Immune System," *Proceedings of the 12th International Conference on Autonomous Agents and Multi-Agent Systems,* pp. 23-30, 2013.

[21] O. Zweigle, B. Keil, M. Wittlinger, K. Häussermann and P. Levi, "Recognizing Hardware Faults on Mobile Robots using Situation Analysis Techniques," *Advances in Intelligent Systems and Computing,* vol. 193, pp. 397-409, 2013.

[22] S. Roumeliotis, G. Sukhatime and G. Bekey, "Fault Detection and Identification in a Mobile Robot using Multiple-Model Estimation," *Proceedings of the International Conference on Robotics and Automation (ICRA),* pp. 2223-2228, 1998.

[23] A. Roennau, G. Heppner, T. Kerscher and R. Dillmann, "Fault Diagnosis and System Status Monitoring for a Six-Legged Walking Robot," *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics,* pp. 874-879, 2011.

[24] V. Chandola, A. Banerjee and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys,* vol. 41, no. 3, Article 15, July 2009.

[25] D. Ghosh, R. Sharman, H. Raghav and S. Upadhyaya, "Self-Healing Systems –

Survey and Synthesis," *Decision Support Systems,* vol. 42, no. 4, pp. 2164-2185, 2007.

[26] H. Psaier and S. Dustdar, "A Survey on Self-Healing Systems: Approaches and Systems," *Computing,* vol. 91, no. 1, pp. 43-73, January 2011.

[27] P. Lepej, J. Maurer, G. Steinbauer, S. Uran and S. Zaman, "An Integrated Diagnosis and Repair Architecture for ROS-Based Robot Systems," *23rd International Workshop on Principles of Diagnosis,* 2012.

[28] R. Isermann, "Fault-Tolerant Systems -A Short Introduction," *Fault-Diagnosis Applications,* pp. 285-289, 2011.

[29] M. Visinsky, J. Cavallaro and I. Walker, "Robotic Fault Detection and Fault Tolerance: A Survey," *Reliability Engineering and System Safety,* vol. 46, pp. 139-158, 1994.

[30] U. Asif, "Improving the Navigability of a Hexapod Robot using a Fault-Tolerant Adaptive Gait," *International Journal of Advanced Robotic Systems,* vol. 9, pp. 33-44, 2012.

[31] G. Arunalatha and C. Jhoncy, "Self-Healing Robots," *Proceedings of the International Conference on Emerging Trends in Robotics and Communication Technologies (INTERACT),* pp. 283-287, 2010.

[32] S. Koos, A. Cully and J. Mouret, "Fast Damage Recovery in Robotics with the T-Resilience Algorithm," *arXiv:1302.0386 [cs.RO],* submitted on 2 February 2013.

[33] J. Bongard, V. Zykov and H. Lipson, "Resilient Machines Through Continuous Self-Modeling," *Science,* vol. 314, no. 5802, pp. 1118-1121, 2006.

[34] H. Seebach, F. Nafz, J. Holtmann, J. Meyer, M. Tichy, W. Reif and W. Schäfer, "Designing Self-Healing in Automotive Systems," *Proceedings of the 7th International Conference on Autonomic and Trusted Computing,* pp. 47-61, 2010.

[35] M. Namvar and F. Aghili, "Failure Detection and Isolation in Robotic

Manipulators using Joint Torque Sensors," *Robotica,* vol. 28, no. 4, pp. 549-561, July 2010.

[36] M. Shi, "Fault Detection, Isolation and Tolerance Control for Robots: A Brief Survey for Last Decade," *Advanced Materials Research,* Vols. 532-533, pp. 479-482, 2012.

[37] V. Krishnan, P. Pathak, S. Jain and A. Samantaray, "Reconfiguration of Four-Legged Walking Robot for Actuator Faults," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering,* vol. 226, pp. 11-26, 2012.

[38] U. Schultz, M. Bordignon and K. Stoy, "Robust and Reversible Execution of Self-Reconfiguration Sequences," *Robotica,* vol. 29, pp. 35-57, 2011.

[39] A. McVean, "Autotomy: Mini-review," *Comparative Biochemistry and Physiology,* vol. 51, no. 3, pp. 497-505, 1975.

[40] T. Maginnis, "The Costs of Autotomy and Regeneration in Animals: A Review and Framework for Future Research," *Behavioral Ecology,* vol. 17, no. 5, pp. 857-872, June 2006.

[41] M. Grabowska, E. Godlewska, J. Schmidt and S. Daun-Gruhn, "Quadrupedal Gaits in Hexapod Animals – Inter-Leg Coordination in Free-Walking Adult Stick Insects," *The Journal of Experimental Biology,* vol. 215, pp. 4255-4266, 2012.

[42] D. Spenneberg, K. McCullough and F. Kirchner, "Stability of Walking in Multilegged Robot Suffering Leg Loss," *Proceedings of the 2004 IEEE International Conference on Robotics and Automation,* vol. 3, pp. 2159-2164, May 2004.

[43] K. Inagaki, "Gait Study for Hexapod Walking with Disabled Leg," *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems,* vol. 1, pp. 408-413, 1997.

[44] K. Mostafa, C. Tsai and I. Her, "Alternative Gaits for Multiped Robots with Leg Failures to Retain Maneuverability," *International Journal of Advanced*

Robotic Systems, vol. 7, no. 4, 2010.

[45] T. Shih, C. Tsai and I. Her, "Comparison of Alternative Gaits for Multiped Robots with Severed Legs," *International Journal of Advanced Robotic Systems,* vol. 9, pp. 157-164, 2012.

[46] M. Görner and G. Hirzinger, "Analysis and Evaluation of the Stability of a Biologically Inspired, Leg Loss Tolerant Gait for Six- and Eight-Legged Walking Robots," *Proceedings of the IEEE International Conference on Robotics and Automation,* May 2010.

[47] A. El Sayed Auf, F. Mösch and M. Litza, "How the Six-Legged Walking Machine OSCAR Handles Leg Amputations," *Proceedings of the Workshop on Bio-Inspired Cooperative and Adaptive Behaviours in Robots at the SAB IX,* 2006.

[48] A. El Sayed Auf, M. Litza and E. Maehle, "Distributed Fault-Tolerant Robot Control Architecture Based on Organic Computing Principles," *Biologically-Inspired Collaborative Computing,* pp. 115-124, 2008.

[49] B. Jakimovski, B. Meyer and E. Maehle, "Swarm Intelligence for Self-Reconfiguring Walking Robot," *IEEE Swarm Intelligence Symposium, SIS 2008,* pp. 21-23, September 2008.

[50] R. Maas, E. Maehle and K. Großpietsch, "Applying the Organic Robot Control Architecture ORCA to Cyber-Physical Systems," *38th Euromicro Conference on Software Engineering and Advanced Applications,* pp. 250-257, 2012.

[51] B. Jakimovski, B. Meyer and E. Maehle, "Self-Reconfiguring Hexapod Robot OSCAR using Organically Inspired Approaches and Innovative Robot Leg Amputation Mechanism," *Proceedings of the International Conference on Automation, Robotics and Control SystemsSystems (ARCS-09),* July 2009.

[52] B. Jakimovski and E. Maehle, "In Situ Self-Reconfiguration of Hexapod Robot OSCAR using Biologically Inspired Approaches," *Climbing and Walking Robots by Behnam Miripour (Ed.), INTECH,* pp. 311-332, 2010.

[53] P. Fleming and P. Bateman, "Just Drop It and Run: The Effect of Limb

Autotomy on Running Distance and Locomotion Energetics of Field Crickets (Gryllus Bimaculatus)," *The Journal of Experimental Biology,* vol. 210, pp. 1446-1454, 2007.

[54] S. Zill, J. Schmitz and A. Büschges, "Load Sensing and Control of Posture and Locomotion," *Arthropod Locomotion Systems: from Biological Materials and Systems to Robotics,* vol. 33, no. 3, pp. 273-286, July 2004.

[55] A. El Sayed Auf, N. Dudek and E. Maehle, "Hexapod Walking as Emergent Reaction to Externally Acting Forces," *Proceedings of Robotica,* pp. 67-72, 2009.

[56] K. Espenschied, R. Quinn, R. Beer and H. Chiel, "Biologically-Based Distributed Control and Local Reflexes Improve Rough Terrain Locomotion in a Hexapod Robot," *Robotics and Autonomous Systems,* vol. 18, pp. 59-64, 1996.

[57] M. Görner and A. Stelzer, "A Leg Proprioception Based 6 DOF Odometry for Statically Stable Walking Robots," *Autonomous Robots,* vol. 34, no. 4, pp. 311-326, May 2013.

[58] J. Zhao, H. Zhang, Y. Liu, J. Yan, X. Zang and Z. Zhou, "Development of the Hexapod Robot HITCR-II for Walking on Unstructured Terrain," *Proceedings of 2012 IEEE International Conference on Mechatronics and Automation,* pp. 64-69, August 2012.

[59] U. Asif and J. Iqbal, "On the Improvement of Multi-Legged Locomotion over Difficult Terrains using a Balance Stabilization Method," *International Journal of Advanced Robotic Systems,* vol. 9, pp. 1-13, 2012.

[60] K. Kamikawa, T. Takubo, Y. Mae, K. Inoue and T. Arai, "Omni-Directional Gait of Multi-Legged Robot on Rough Terrain by Following the Virtual Plane," *Journal of Robotics and Mechatronics,* vol. 24, no. 1, pp. 71-85, 2012.

[61] K. Walas and D. Belter, "Supporting Locomotive Functions of a Six-Legged Walking Robot," *International Journal of Applied Mathematics and Computer Science,* vol. 21, no. 2, pp. 363-377, June 2011.

[62] E. Garcia, P. Gonzalez de Santos and F. Matia, “Dealing with Internal and External Perturbation on Walking Robots,” *Autonomous Robots,* vol. 24, pp. 213-227, 2008.

[63] J. Hartmann, W. Stechele and E. Maehle, “Self-Adaptation for Mobile Robot Algorithms using Organic Computing Principles,” *Lecture Notes in Computer Science,* vol. 7767, pp. 232-243, 2013.

[64] D. Sanz-Merodio, E. Garcia and P. Gonzalez-de-Santos, “Analyzing Energy-Efficient Configurations in Hexapod Robots for Demining Applications,” *Industrial Robot: An International Journal,* vol. 39, no. 4, pp. 357-364, 2012.

[65] S. Roy and D. Pratihar, “Dynamic Modeling, Stability and Energy Consumption Analysis of a Realistic Six-Legged Walking Robot,” *Robotics and Computer-Integrated Manufacturing,* vol. 29, no. 2, pp. 400-416, April 2013.

[66] S. Roy and D. Pratihar, “Effects of Turning Gait Parameters on Energy Consumption and Stability of a Six-Legged Walking Robot,” *Robotics and Autonomous Systems,* vol. 60, no. 1, p. 72–82, January 2012.

[67] B. Jin, C. Chen and W. Li, “Power Consumption Optimization for a Hexapod Walking Robot,” *Journal of Intelligent & Robotic Systems,* 2012.

[68] J. Yuan, G. Liu and B. Wu, “Power Efficiency Estimation-Based Health Monitoring and Fault Detection of Modular and Reconfigurable Robot,” *IEEE Transactions on Industrial Electronics,* vol. 58, no. 10, pp. 4880-4887, October 2011.

[69] X. Li and L. Parker, “Distributed Sensor Analysis for Fault Detection in Tightly-Coupled Multi-Robot Team Tasks,” *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA),* pp. 3103-3110, 2009.

[70] J. Bezdec, R. Ehrlich and W. Full, “FCM: The Fuzzy C-Means Clustering Algorithm,” *Computers & Geosciences,* vol. 10, no. 2-3, pp. 191-203, 1984.

[71] D. Grollman, O. Jenkins and F. Wood, “Extensible Data-driven Classification

of Robot Sensor Data," *Brown Computer Science Department, Technical Report, cs05-11,* 2005.

[72] R. Hathaway and J. Bezdek, "Optimization of Clustering Criteria by Reformulation," *IEEE Transactions on Fuzzy Systems,* vol. 3, pp. 241-245, 1995.

[73] W. Wang, Y. Zhang, Y. Li and X. Zhang, "The Global Fuzzy C-Means Clustering Algorithm," *6th World Congress on Intelligent Control and Automation (WCICA 2006),* vol. 1, pp. 3604-3607, 2006.

[74] G. Dudek and P. Giguere, "Clustering Sensor Data for Autonomous Terrain Identification using Time-Dependency," *Autonomous Robots,* vol. 26, no. 2-3, pp. 171-186, 2009.

[75] S. Lenser and M. Veloso, "Classification of Robotic Sensor Streams using Non-Parametric Statistics," *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* vol. 3, pp. 2719-2724, 2004.

[76] Fort Eustis, Va.: U.S. Army Transportation Combat Developments Agency, "Logistical Vehicle Off-Road Mobility," *Project TCCO 62-5,* February 1967.

[77] G. Reina and M. Foglia, "On the Mobility of All-Terrain Rovers," *Industrial Robot: An International Journal,* vol. 40, no. 2, pp. 121-131, 2013.

[78] O. Janrathitikarn and L. Long, "Gait Control of a Six-Legged Robot on Unlevel Terrain using a Cognitive Architecture," *Proceedings of the IEEE International Conference on Aerospace,* pp. 1-9, March 2008.

[79] P. Arena, L. Fortuna, M. Frasca, L. Patane and M. Pavone, "Realization of a Cnn-Driven Cockroach-Inspired Robot," *Proceedings of the IEEE International Symposium on Circuits and Systems,* pp. 2649-2652, 2006.

[80] Y. Pan and F. Gao, "A New 6-Parallel-Legged Walking Robot for Drilling Holes on the Fuselage," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* article in press, published online on 10 May 2013.

[81] K. Byl, “Metastable Legged Robot Locomotion,” *Unpublished Thesis (PhD), Massachusetts Institute of Technology,* September 2008.

[82] J. Wong and W. Huang, “Wheels vs. Tracks –A Fundamental Evaluation From the Traction Perspective,” *Journal of Terramechanics,* vol. 43, pp. 27-42, 2006.

[83] M. Hunt, “Robotic Walking in the Real World,” *Science,* vol. 339, no. 6126, pp. 1389-1390, 2013.

[84] F. Hardarson, “Locomotion for Difficult Terrain. A Survey Study,” *Technical Report, Mechatronics Division, Department of Machine Design, Royal Institute of Technology,* 1997.

[85] F. Pfeiffer, S. Josef and T. Roßmann, “Legged Walking Machines,” *Autonomous Robotic Systems,* vol. 236, pp. 235-263, 1998.

[86] X. Zhou and S. Bi, “A Survey of Bio-Inspired Compliant Legged Robot Designs,” *Bioinspiration & Biomimetics,* vol. 7, no. 4, December 2012.

[87] A. Crespi, K. Karakasiliotis, A. Guignard and A. Ijspeert, “Salamandra Robotica II: An Amphibious Robot to Study Salamander-Like Swimming and Walking Gaits,” *IEEE Transactions on Robotics,* vol. 29, no. 2, pp. 308-320, April 2013.

[88] R. King, “State of the Art in Robotics and Robotic Actuation,” *A Biologically Inspired Robot with Walking and Rolling Locomotion, Biosystems & Biorobotics,* vol. 2, pp. 29-47, 2013.

[89] R. Dillmann, J. Albiez, B. Gaßmann, T. Kerscher and M. Zöllner, “Biologically Inspired Walking Machines: Design, Control and Perception,” *Philosophical Transactions of the Royal Society A,* vol. 365, no. 1850, pp. 133-151, 2007.

[90] T. Allen, R. Bachmann, D. Kingsley, G. Nelson, J. Offi and R. Quinn, “Parallel Complementary Strategies for Implementing Biological Principles into Mobile Robots,” *The International Journal of Robotics Research,* vol. 22, no. 3, pp. 169-186, 2003.

[91] R. Beer, R. Quinn, H. Chiel and R. Ritzmann, "Biologically-Inspired Approaches to Robotics: What Can We Learn from Insects?," *Communications of the ACM 40,* pp. 30-38, 1997.

[92] H. Cruse, "What Mechanisms Coordinate Leg Movement in Walking Arthropods," *Trends in Neurosciences,* vol. 13, no. 1, pp. 15-21, January 1990.

[93] K. Espenschied, R. Quinn, H. Chiel and R. Beer, "Leg Coordination Mechanisms in Stick Insect Applied to Hexapod Robot Locomotion," *Adaptive Behavior,* vol. 4, no. 1, pp. 455-468, 1993.

[94] K. Berns, S. Cordes and W. Ilg, "Adaptive, Neural Control Architecture for the Walking Machine Lauron," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 1172-1177, 1994.

[95] S. Cordes, K. Berns and L. Leppanen, "Sensor Components of the Six-Legged Walking Machine LAURON II," *Proceedings of the International Conference on Advanced Robotics (ICAR),* pp. 71-76, 1997.

[96] B. Gaßmann, K. Scholl and K. Berns, "Locomotion of LAURON III in Rough Terrain," *Proceedings of the International Conference on Advanced Mechatronics,* July 2001.

[97] A. Ljspeert, "Central Pattern Generators for Locomotion Control in Animals and Robots: A Review," *Neural Networks,* vol. 21, no. 4, pp. 642-653, 2008.

[98] H. Chiel, R. Beer and J. Gallagher, "Evolution and Analysis of Model CPGs for Walking I. Dynamical Modules," *Journal of Computational Neuroscience,* vol. 7, no. 2, pp. 99-118, 1999.

[99] R. Beer, H. Chiel and J. Gallagher, "Evolution and Analysis of Model CPGs for Walking II. General Principles and Individual Variability," *Journal of Computational Neuroscience,* vol. 7, no. 2, pp. 119-147, 1999.

[100] Q. Wu, C. Liu, J. Zhang and Q. Chen, "Survey of Locomotion Control of Legged Robots Inspired by Biological Concept," *Science in China Series F: Information Sciences,* vol. 52, no. 10, pp. 1715-1729, October 2009.

[101] S. Inagaki, H. Yuasa and A. Tamio, "CPG Model for Autonomous Decentralized Multi-Legged Robot System-Generation and Transition of Oscillation Patterns and Dynamics of Oscillators," *Robotics and Autonomous Systems,* vol. 44, no. 3-4, pp. 171-179, 2003.

[102] Ö. Ekeberg, M. Blümel and A. Büschges, "Dynamic Simulation of Insect Walking," *Arthropod Locomotion Systems: from Biological Materials and Systems to Robotics,* vol. 33, no. 3, pp. 287-300, July 2004.

[103] G. Brambilla, J. Buchli and J. Ijspeert, "Adaptive Four Legged Locomotion Control Based on Nonlinear Dynamical Systems," *From Animals to Animats - Proceedings of the 9th International Conference on the Simulation of Adaptive Behavior (SAB'06),* vol. 409, 2006.

[104] J. Barron-Zambrano and C. Torres-Huitzil, "FPGA Implementation of a Configurable Neuromorphic CPG-Based Locomotion Controller," *Neural Networks, 2013 Special Issue,* article in press, published online on 12 April 2013.

[105] C. Wang, J. Gao, X. Duan, Y. Liu, X. Li, Z. Xu and W. Sun, "The CPG Gait Generate Method of the Quadruped Robot Based on Iterative Learning Control Algorithm," *Advanced Materials Research,* vol. 677, pp. 296-303, 2013.

[106] P. Arena, L. Patane, M. Schilling and J. Schmitz, "Walking Capabilities of Gregor Controlled through Walknet," *Proceedings of SPIE, The International Society for Optical Engineering,* vol. 6592, 2007.

[107] H. Cruse, T. Kindermann, M. Schumm, J. Dean and J. Schmitz, "Walknet -A Biologically Inspired Network to Control Six-Legged Walking," *Neural Networks,* vol. 11, no. 7-8, pp. 1435-1447, October 1998.

[108] V. Dürr, J. Schmitz and H. Cruse, "Behaviour-Based Modelling of Hexapod Locomotion: Linking Biology and Technical Application," *Arthropod Structure & Development,* vol. 33, no. 3, pp. 237-250, July 2004.

[109] A. Schneider, J. Paskarbeit, M. Schaeffersmann and J. Schmitz, "HECTOR, a New Hexapod Robot Platform with Increased Mobility - Control Approach,

Design and Communication," *Advances in Autonomous Mini Robots,* pp. 249-264, 2012.

[110] D. Wettergreen, H. Pangels and J. Bares, "Behavior-Based Gait Execution for the Dante II Walking Robot," *Proceedings of the International Conference on Intelligent Robots and Systems,* vol. 3, pp. 274-279, 1995.

[111] J. Porta and E. Celaya, "Reactive Free-Gait Generation to Follow Arbitrary Trajectories with a Hexapod Robot," *Robotics and Autonomous Systems,* vol. 47, no. 6, pp. 187-201, July 2004.

[112] B. Jakimovski, B. Meyer and E. Maehle, "Firefly Flashing Synchronization as Inspiration for Self-Synchronization of Walking Robot Gait Patterns using a Decentralized Robot Control Architecture," *Architecture of Computing Systems - ARCS 2010, 23rd International Conference,* 2010.

[113] M. Schilling, J. Paskarbeit, J. Schmitz, A. Schneider and H. Cruse, "Grounding an Internal Body Model of a Hexapod Walker Control of Curve Walking in a Biologically Inspired Robot," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* pp. 2762-2768, October 2012.

[114] P. Manoonpong, U. Parlitz and F. Wörgötter, "Neural Control and Adaptive Neural Forward Models for Insect-Like, Energy-Efficient, and Adaptable Locomotion of Walking Machines," *Frontiers in Neural Circuits,* vol. 7, no. 12, pp. 1-28, February 2013.

[115] G. Morse, S. Risi, C. Snyder and K. Stanley, "Single-Unit Pattern Generators for Quadruped Locomotion," *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2013),* July 2013.

[116] D. Pratihar, K. Deb and A. Ghosh, "Optimal Path and Gait Generations Simultaneously of a Six-Legged Robot using GA-Fuzzy Approach," *Robotics and Autonomous Systems,* vol. 41, no. 1, pp. 1-20, 2002.

[117] C. Theeravithayangkura, T. Takubo, K. Ohara, Y. Mae and T. Arai, "Adaptive Gait for Dynamic Rotational Walking Motion on Unknown Non-Planar

Terrain by Limb Mechanism Robot ASTERISK," *Journal of Robotics and Mechatronics,* vol. 25, no. 1, pp. 172-182, 2013.

[118] K. Seo and S. Hyun, "Toward Automatic Gait Generation for Quadruped Robots using Cartesian Genetic Programming," *Applications of Evolutionary Computation, Lecture Notes in Computer Science,* vol. 7835, pp. 599-605, 2013.

[119] M. Erden and K. Leblebicioğlu, "Free Gait Generation with Reinforcement Learning for a Six-Legged Robot," *Robotics and Autonomous Systems,* vol. 56, no. 3, pp. 199-212, March 2008.

[120] J. Porta and E. Celaya, "Efficient Gait Generation using Reinforcement Learning," *Proceedings of the 4th International Conference on Climbing and Walking Robots,* pp. 411-418, 2001.

[121] N. Zeitlin, P. Abbeel and R. Fearing, "Reinforcement Learning Methods to Enable Automatic Tuning of Legged Robots," *University of California at Berkeley, Technical Report No. UCB/EECS-2012-129,* May 2012.

[122] J. Zhang, X. Han and X. Han, "Walking Quality Guaranteed CPG Control Method," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* article in press, published online on 8 May 2013.

[123] S. Chernova and M. Veloso, "An Evolutionary Approach To Gait Learning For Four-Legged Robots," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems,* vol. 3, pp. 2562-2567, 2004.

[124] G. Hornby, S. Takamura, J. Yokono, O. Hanagata, T. Yamamoto and M. Fujita, "Evolving robust gaits with AIBO," *Proceedings of the IEEE International Conference on Robotics and Automation,* vol. 3, pp. 3040-3045, 2000.

[125] M. Silva and J. Tenreiro, "A Literature Review on the Optimization of Legged Robots," *Journal of Vibration and Control,* vol. 18, no. 12, pp. 1753-1767, October 2012.

[126] Prihastono, K. Anam, R. Effendi, A. Santoso and A. Jazidie, "Autonomous Five Legs Rescue Robot Navigation in Cluttered Environment," *Academic Research International,* vol. 2, no. 1, pp. 155-163, January 2012.

[127] E. Karalarli, A. Erkmen and I. Erkmen, "Intelligent Gait Synthesizer for Hexapod Walking Rescue Robots," *Proceedings of the 2004 IEEE International Conference on Robotics and Automation,* pp. 2177-2182, April 2004.

[128] E. Vargas-Soto, E. Gorrostieta, A. Sotomayor-Olmedo, J. Ramos-Arreguin and S. Tovar-Arriaga, "Design of Fuzzy Algorithms Locomotion for Six Legged Walking Robot," *International Journal of Physical Sciences,* vol. 7, no. 11, pp. 1811-1819, March 2012.

[129] P. Jensen, P. Karoly and S. Braver, "The Measurement of Clinical Pain Intensity: A Comparison of Six Methods," *Pain,* vol. 27, no. 1, pp. 117-126, 1986.

[130] W. Agosta, "Chemical Communication: the Art of Pheromones," *Scientific American Library,* 1992.

[131] S. B. Klein, Learning: Principles and Applications, 4ed., Mississippi State University: McGraw-Hill Higher Education, 2002.