University of Southern Queensland

Faculty of Engineering and Surveying

Building a Two Wheeled Balancing Robot

A dissertation submitted by

Mr Peter Miller

In fulfilment of the requirements of

Courses ENG4111 and 4112 Research Project

towards the degree of

Bachelor of Engineering (Electrical and Electronic)

Submitted: October, 2008

Abstract

Two wheeled balancing robots are an area of research that may well provide the future locomotion for everyday robots. The unique stability control that is required to keep the robot upright differentiates it from traditional forms of robotics. The inverted pendulum principle provides the mathematical modelling of the naturally unstable system. This is then utilised to develop and implement a suitable stability control system that is responsive, timely and successful in achieving this objective.

Completing the design and development phase of the robot requires careful consideration of all aspects including operating conditions, materials, hardware, sensors and software. This process provides the ongoing opportunity of implementing continued improvements to its perceived operation whilst also ensuring that obvious problems and potential faults are removed before construction.

The construction phase entails the manufacture and assembly of the robots circuits, hardware and chassis with the software and programming aspects then implemented. The later concludes the robots production where the final maintenance considerations can be determined. These are essential for ensuring the robots continued serviceability.

The analysis and evaluation of the completed robot provides the ability to assess the robots effectiveness and efficiency in maintaining stability. This allows a comparison to be undertaken between the actual system performances and the anticipated project objectives. The opportunity to calibrate and perform additional fine tuning of the design is also explored. The project is concluded with comments on each aspect of the project with recommendations for improvement, additional capabilities and future areas of investigation.

University of Southern Queensland

Faculty of Engineering and Surveying

ENG4111 Research Project Part 1 & ENG4112 Research Project Part 2

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Engineering and Surveying, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Engineering and Surveying or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course "Project and Dissertation" is to contribute to the overall education within the student's chosen degree programme. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Professor Frank Bullen
Dean
Faculty of Engineering and Surveying

Certification

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Peter Miller

Student Number: 0050031502

____________________ Signature

29OCT08 Date

Acknowledgements

I would like to extend my appreciation and love to my family for all their continuing support, care and consideration. For my loving wife who has given me every opportunity to succeed, for my fantastic son who has always shared the interest, and my beautiful daughter who always knew I could do it, thank you for sharing the journey with me. And, thank you to my mum and dad, Audrey and Phil, for all the support and guidance over the years.

Finally, I would like to thank my supervisor Mark Phythian.

Table of Contents

List of Figures

List of Tables

List of Appendices

List of References

Anderson, DP, 2007, nBot Balancing Robot, viewed 20th March 2008, <http://www.geology.smu.edu/~dpa-www/robo/nbot/>

Bush, L 2001, Fuzzy Logic Controller for the Inverted Pendulum Problem.

Deepa, S, Sivanandam S & Sumathi, S 2007, Introduction to Fuzzy Logic using MATLAB, Springer, New York.

Florian, RV, 2007, Correct equations for the dynamics of the cart-pole system, Center for Cognitive and Neural Studies, viewed 15th May 2008, <http://www.coneural.org/florian/papers/05_cart_pole.pdf>

Larson, T, 2008, Balancing Robot Project - Bender, viewed 20th March 2008, <http://www.tedlarson.com/robots/balancingbot.htm>

McComb, G & Predko, M 2006, Robot Builders Bonanza, McGraw-Hill, Sydney.

Segway Inc, 2008, Simply moving, viewed 20th March 2008, <http://www.segway.com/>

Simon, D 2001, Kalman Filtering, Embedded Systems Programming.

Chapter 1

Introduction

Robotics has always been played an integral part of the human psyche. The dream of creating a machine that replicates human thought and physical characteristics extends throughout the existence of mankind. Developments in technology over the past fifty years have established the foundations of making these dreams come true. Robotics is now achievable through the miniaturisation of the microprocessors which performs the processing and computations. New forms of sensor devices are being developed all the time further providing machines with the ability to identify the world around them in so many different ways.

Effective and efficient control system designs provide the robot with the ability to control itself and operate autonomously. Artificial intelligence (AI) is becoming a definite possibility with advancements in non-linear control systems such as neural networks and fuzzy controllers. Improved synthetics and materials allow for robust and cosmetically aesthetic designs to be implemented for the construction and visual aspects of the robot.

Two wheeled robots are one variation of robot that has become a standard topic of research and exploration for young engineers and robotic enthusiasts. They offer the opportunity to develop control systems that are capable of maintaining stability of an otherwise unstable system. This type of system is also known as an inverted pendulum. This research project aims to bring this, and many of the previously mention aspects of a robot together into the building of a two wheeled balancing robot with a non-linear, fuzzy controller.

This field of research is essential as robots offer an opportunity of improving the quality of life for every member of the human race. This will be achieved through the reduction of human exposure to hazardous conditions, dangerous environments and harmful chemicals and the provision of continual 24 Hr assistance and monitoring for people with medical conditions, etc. Robots will be employed in many applications within society including carers, assistants and security.

1.1 Two Wheeled Balancing Robots

A robot that is capable of balancing upright on its two wheels is known as a two wheeled balancing robot. The following figure contains the physical view for the robot designed as part of this project. The process of balancing is typically referred to as stability control. The two wheels are situated below the base and allow the robot chassis to maintain an upright position by moving in the direction of tilt, either forward or backward, in an attempt to keep the centre of the mass above the wheel axles. The wheels also provide the locomotion thus allowing the robot to transverse across various terrains and environments.

This type of robot provides a challenging problem and has resulted in many useful and interesting designs being developed. One such two wheeled robot that has become a commercial success is the Segway by Segway Inc. The immediate impact has been within the personal transportation area where an alternative to cumbersome wheelchairs is now available. Segway proves a comfortable mobility opportunity for the elderly or people with disability thus improving there individual sense of independence at the same time. The theory used to maintain stability of these robots is based on the inverted pendulum theory which is covered in the following section 1.2.

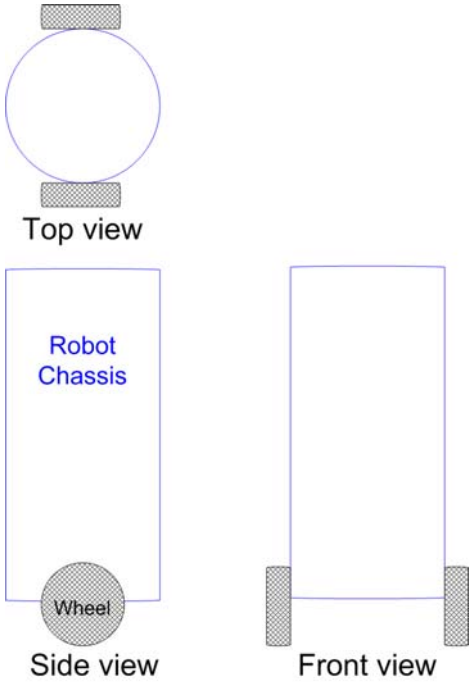

Figure 1.1 Views of the two wheeled balancing robot

1.2 Inverted Pendulum Theory

To develop a reliable and capable control system for a two wheeled balancing robot, an understanding of the parameters within the system is essential. Representation of these can be achieved through a mathematical model. Inverted pendulum theory is more traditionally known as Pole and Cart theory and although the two wheeled balancing robot does not directly compare to the Pole and Cart, the same principles are in effect. Within the system model, the cart equates to the wheels whilst the pole equates to the robot's chassis.

(Florian 2007) presents an updated version of the pole and cart system which have been utilised as the theory basis for this project. The equations were derived by utilising Newton's second law with added corrections from previous versions published by researchers. These corrections concluded that the gravitation acceleration is in fact a positive value, and that the frictional force between the wheel and horizontal surface should be incorporated in the equation.

Friction coefficients have been neglected in this project as the robot will be expected to transverse across numerous types of terrains and surfaces. If the coefficients were to be considered during the control systems design and implementation, then additional sensors, circuitry and power consumption would be required to derive these new values whilst in operation. The time, effort and resources required to create this capability far exceed any benefits that could be expected with there inclusion.

It is necessary to generalise the effects of the left and right wheels and incorporate them together under the combined term “wheels”. This simplifies the calculations as both wheels will work in unison to maintain stability. For determining specific torque (forces) requirements for each individual wheel, the wheels value can be halved for an approximate single wheel value. This approach is considered acceptable as the terrain and surface will vary between the wheels on certain terrains.

The aim of the inverted pendulum principle is to keep the wheels beneath the centre of the robot chassis’s mass. If the robot begins to tilt forward, then to maintain stability, the wheel will need to move forward to return beneath the chassis mass. If this is not maintained, the robot will simply fall over. The following system dynamics are associated with the mathematical problem.

System Dynamics

The following system dynamics are utilised within the mathematical problem of the two wheeled balancing robots stability control (inverted pendulum approach).

x	Displacement (Horizontal) (m)
$\dot{x}$	Velocity (Horizontal) (ms^{-1})
$\ddot{x}$	Acceleration (Horizontal) (ms^{-2})
θ	Angular displacement (Vertical) (rad s)
$\dot{\theta}$	Angular velocity (Vertical) (rad s^{-1})
$\ddot{\theta}$	Angular acceleration (Vertical) (rad s^{-2})
M_{wh}	Mass of the wheels and drive shafts (Kg)
M_{rc}	Mass of the robot chassis (Kg)
2l	Height of the robot chassis (m)

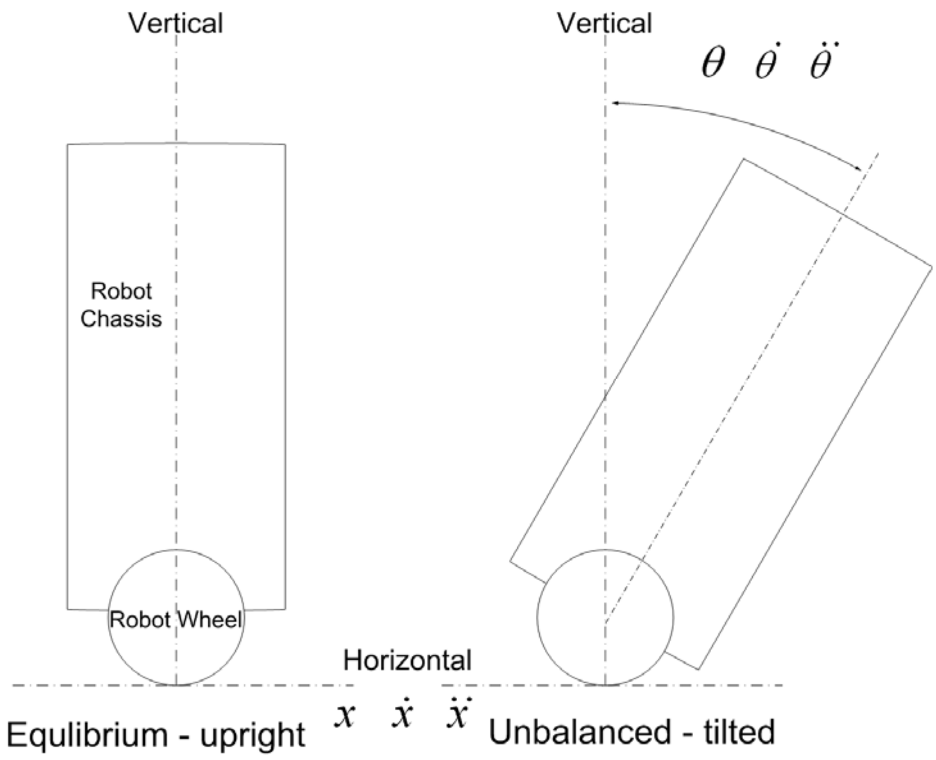

Figure 1.2 Displacement, velocity and acceleration parameters

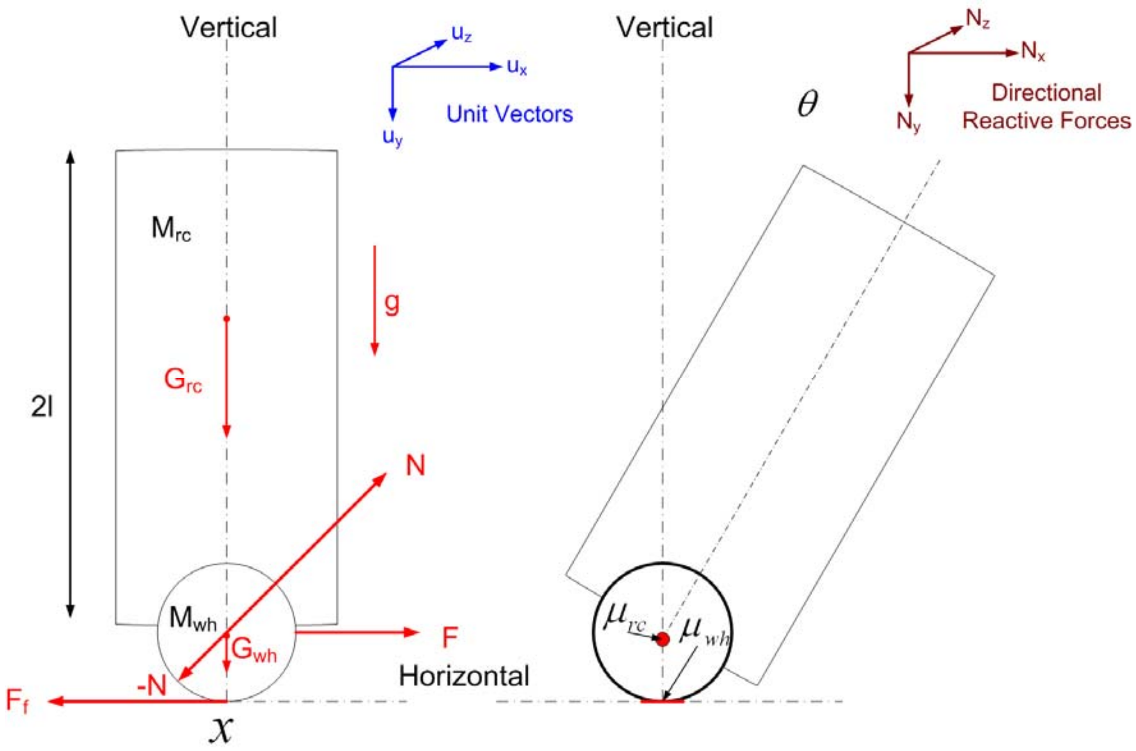

Figure 1.3 Inverted pendulum parameters

F Horizontal force applied by wheels (N)

$$F = F * u_x$$

F_f Friction force between wheels and surface (N)

$$F_f = -F_f * u_x$$

$$F_f = \mu_{wh}[(M_{wh} + M_{rc})g - M_{rc} l(\ddot{\theta} Sin\theta + \dot{\theta}^2 Cos\theta)]Sgn(N_{wh} * \dot{x})$$

N Reactive force on robot chassis from the wheel (N)

$$N = N_x * u_x - N_y * u_y$$

N_{wh} Reactive force on wheels (N)

$$N_{wh} = -N_{wh} * u_y$$

$$N_{wh} = (M_{wh} + M_{rc})g - M_{rc}(\ddot{\theta} Sin\theta + \dot{\theta}^2 Cos\theta)$$

g Gravitation acceleration ($9.8ms^{-2}$)

G_{wh} Gravity effect on wheels ($Kg*m*s^{-2}$)

$$G_{wh} = M_{wh} * g * u_y$$

G_{rc} Gravity effect on robot chassis ($Kg*m*s^{-2}$)

$$G_{rc} = M_{rc} * g * u_y$$

μ_{wh} Friction coefficient of wheels on surface

μ_{rc} Friction coefficient of drive shafts

N_x Reactive force in the direction of x (N)

$$N_x = M_{rc}(\ddot{x} + l * \theta Cos\theta - l * \dot{\theta}^2 Sin\theta)$$

N_y Reactive force in the direction of y (N)

$$N_y = M_{rc} * g(l * \ddot{\theta} Sin\theta - l * \dot{\theta}^2 Cos\theta)$$

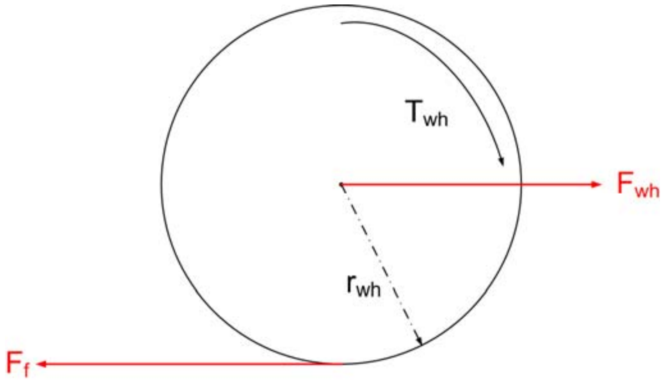

Figure 1.4 Wheel parameters

τ_{wh} Wheel torque (rotational force) (Nm)

$$\tau_{wh} = F_{wh} * r_{wh} * Cos\theta$$

r_{wh} Wheel radius (m)

F_{wh} Horizontal force of a wheel (left or right) (N)

$$F_{wh} = \frac{F}{2}$$

The desired motor torque is estimated during the motor considerations within Chapter 3. Motor torque is the force applied to the wheel before frictional and mechanical losses are subtracted. The resultant torque, also known as applied torque, is referred to as wheel torque in these equations. The importance of selecting the correct radius of wheel is evident from the effects displayed in the wheel torque equation. This indicates that a larger radius for the wheel would require a larger torque to be applied by the motor.

The cosine function within the wheel torque equation details the advantage of tilting the robot chassis in the direction desired locomotion. If the robot was to begin moving forward and the tilt angle in that direction is increased, then the torque required to move the robot in that direction will be reduced. For example, when the angle of tilt is 0^o, the cosine multiplier will be 1 equating to 100% of the torque. When the angle of tilt increases to 20^o or even 30^o, then the multiplier will equal 0.94 and 0.87 respectively. This could be view as 94% or 87% of the torque available is now required compared to the previous 100%.

The following equations will provide a reliable and accurate model for developing and implementing a suitable non linear control system for the two wheeled balancing robot.

Angular acceleration
$$\ddot{\theta} = \frac{g * Sin\theta + Cos\theta\left(\frac{-F - M_{rc} * l * \dot{\theta}^2 Sin\theta}{M_{wh} + M_{rc}}\right)}{l\left(\frac{4}{3} - \frac{M_{rc} * Cos\theta^2\theta}{M_{wh} + M_{rc}}\right)}$$

Horizontal acceleration
$$\ddot{x} = \frac{F + M_{rc} * l\left(\dot{\theta}^2 Sin\theta - \ddot{\theta} Cos\theta\right)}{M_{wh} + M_{rc}}$$

1.3 Autonomous Stability

Stability for the two wheeled balancing robot lies in its ability to maintain the robot chassis in an upright, equilibrium, position. Balancing a robot automatically without human interaction is known as autonomous stability because it does this by self governance. The inverted pendulum theory provides the equations required to ascertain motion, force and reactions that occur in the process. It is then necessary to apply an effective and efficient control system that is capable of responding to the sensory inputs within a minimal timeframe so that stability can be attained, and then maintained.

This project aims to solve this problem of maintaining stability of the two wheeled balancing robot by designing a reliable control system capable of functioning within a PIC microcontroller. A non-linear control system known as a fuzzy controller will be developed allowing a more robust and stable system compared to the linear equivalents currently available. Additional problems inherent in such a design will be explored within chapter 2.

1.4 Project Objectives and Timeline

The project objectives were broken into several key goals. The first was to review and evaluate literature encompassing inverted pendulum theory and two wheeled balancing robots. This provided the basis for an informed design approach based on previous experiences and procedures by others. The second goal was associated with the development of a microcontroller based control system capable of maintaining stability of the robot. The resultant physical circuitry requirements were then finalised in a microcontroller unit design as part of the third goal.

Goal four required a finalised robot design including integration all of its necessary components, sensors and PCB's before the manufacturing and assembly of the robot was completed. The following goal was to attain an analysis and evaluation of the measured performance data to establish its limitations, capabilities and potential shortfalls. Information obtained during each of these steps including the resultant outcomes were then compiled, concluding this research project. These key goals/objectives are defined as follows:

1. Research will be conducted into the theory of an inverted pendulum and the various considerations that may be necessary during the construction of a two wheeled balancing robot. This will present an indication of the potential problems, resources required and timeframes expected. This will provide overall direction of the project.

2. A non-linear control system will be designed and then simulated to ascertain if it is capable of stabilising the initial design concepts. Once a design is established, progression to the next objective may occur.

3. Design of a microcontroller unit capable of achieving the control system derived above will be completed including the necessary circuitry, sensors and circuit boards. Additional, the unit will be configured with an interface for future incorporation of locomotion (trajectory) control.

4. The robot design will be finalised with the resultant resource requirements sourced. Manufacture and assembly will begin with the final outcome of this step being the construction of the two wheeled balancing robot.

5. An analysis and evaluation will be performed on the robot to assess the overall system performance. This will provide continual improvements to the robustness of the stability and locomotion before completion of the project dissertation.

6. A completed final dissertation consisting of the design, simulation, manufacture and testing processes used to derive the robot will be submitted to the University of Southern Queensland for assessment. This will examine the successfulness of the project in overcoming the problems and objectives previously defined.

The timeline in the following figure contains the chapters that were completed during the course of this project. Initially the research and literature review chapters were completed together in late May. The design and development component was then completed in August where the manufacturing and assembly components of the following chapter then begun. Data analysis was completed concurrently with the assembly stage in an attempt to identify any potential shortfalls inherent in the design as well as improving overall effectiveness and efficiency of the robot. Conclusions and recommendations were completed in late October with the final submission.

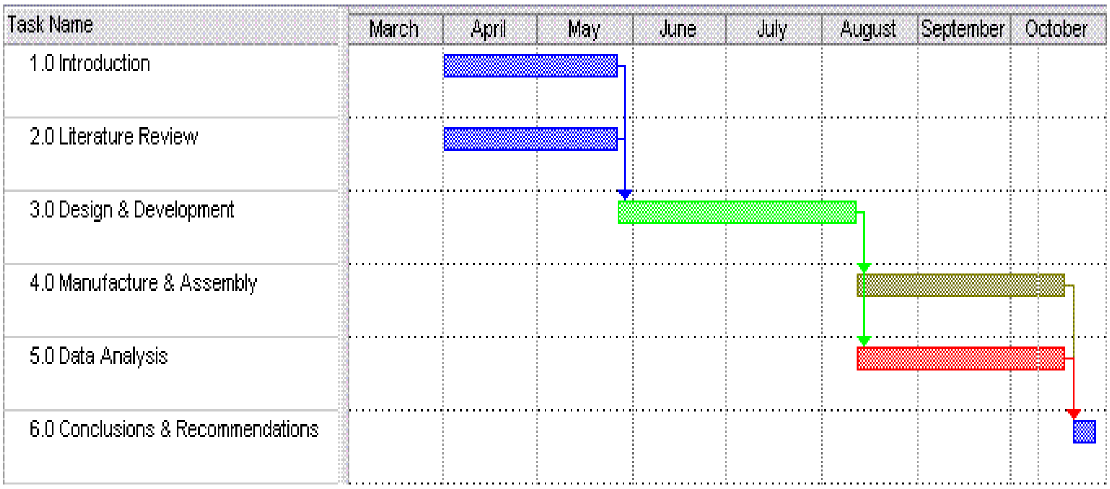

Figure 1.5 Dissertation completion timeline

1.5 Methodology

The research project was divided into chapters, each a sequential step in the process of developing and building the two wheeled balancing robot. This approach was utilised in an attempt to progress the project from one task to the next as it was undertaken. Each is defined so that it builds on the previous task thus evolving the robot within the goals and requirements generated. This ultimately led to the completion of the two wheeled balancing robot that met the objectives within the timeframe available.

The first chapter formed the first step where key points and objectives were established including the idea of what a two wheeled balancing robot actually is. Understanding your project is critical in determining plans for conducting research and performing the design work. Chapter 2 provides the second step in which a thorough understanding of previous projects and approaches is desired. This established the foundations for making informed decision based on the previous experiences and problems encountered. This can help introduce an avoidance of problems, adequate planning of resources and the effective application of effort.

The third step in the methodology was to apply the acquired knowledge in key areas so that the control system was achieved with the resources available. Once this was determined, simulation of the stability control system and additional parameters was attempted before settling on the next step of constructing the robot. The simulation task tested the derived sub-routines of the software based controller which was undertaken within the 'MPLAB' program. This form of testing provides valuable data on the likely stability capability of the control system. This provides the opportunity to design out faults or shortfalls before resources; time and money are inadvertently wasted. This ensured a complete, well planned, and capable machine was developed.

The next step entailed the manufacture of PCB's, chassis, drive shafts, etc followed by the process necessary to make the robot a realisation. This was the rewarding section where the hard work finally began evolving into an actual machine. The following step was to analyse the actual performance of the robot and ascertain its ability in achieving the objectives of stability and balance. This also provided the opportunity to calibrate and perform additional fine tuning of the design allowing the machine to become more effective and efficient in its performance.

The final component comprises of a complete assessment of each process undertaken, the choices made and achievements obtained during the project as well as evaluation of the final robots effectiveness. This expanded to include recommendations for future work that could be undertaken in an effort to improve areas of the process or design, addition of capabilities, or how to overcome problems that may have been encountered.

1.6 Risk Assessment

The risk assessment contained within the following table is an assessment of the perceived risks associated with the work to be undertaken in the research project. The table is broken into several components consisting of the task to be performed, the hazard, the associated risks, possible risk control measures, the rating of the risk and potential exposure expected. No substantial risks or hazards are expected during the course of this project. Risk Ratings are defined as follows:

- Extremely slight risks are practically impossible to occur.
- Very slight risks are unlikely to occur.
- Slight risks are possible but unlikely to occur.
- Significant risks are possible and likely to occur.
- Substantial risks are very likely to occur.

How often the persons involved will be exposed to defined risks is known as risk exposure. These have been defined as follows:

- Rarely occurs only few times during the course of the project.
- Occasionally occurs once or twice in a month.
- Regularly occurs weekly.
- Frequently occurs each day.
- Continuously is a constant exposure to the risk at all times.

Task	Hazard	Associated Risks	Risk Control Measures	Risk Rating	Risk Exposure
Literature review and research tasks.	Eye strain. Muscular cramps.	Short term pain. Discomfort.	Regular breaks. Use of adequate lighting and ergonomic furniture.	Very Slight	Regularly
Manufacture and assembly of robot assemblies and components.	Physical injury from tools or materials. Electrocution from power tools. Exposure to Battery. Damage to sensitive components and costly parts.	Pain and/or physical injury. Chemical burn from battery. Lost project time. Increased costs due to replacement components and parts.	Use of Personal Protection Equipment (PPE). Adequate training in tools and material usage is undertaken.	Slight	Regularly
Programming, software design, simulation and analysis.	Eye strain. Muscular cramps. Fatigue.	Short term pain. Discomfort. Loss of concentration.	Regular breaks. Correct calibration of monitors and positioning of computer peripherals. Use of adequate lighting and ergonomic furniture.	Significant	Frequently
Report writing and compilation.	Eye strain. Muscular cramps. Fatigue.	Short term pain. Discomfort. Loss of concentration.	Regular breaks. Correct calibration of monitors and positioning of computer peripherals. Use of adequate lighting and ergonomic furniture	Slight	Regularly
Robots operation.	Robot falling, tipping or colliding.	Injury to people or damage to equipment and surroundings.	Reduce robots weight. Incorporate sensors to avoid potential collisions.	Slight	Continuously

Table 1.1 Risk Assessment of Research Project

1.7 Conclusion

The dissertation aims to evaluate the process from gathering knowledge in the field of control system and robotics, through to apply them in the design, simulation, manufacture and subsequent analysis of the completed two wheeled balancing robot. A review of the literature available on this type of robot suggests that a non-linear control system has not been readily achieved. This research project achieved a non-linear stability control system based on the fuzzy controller thus proving that this type of system can actually be achieved.

It is hoped that the results from this research project will encourage future engineers and robotic enthusiasts to attempt a non-linear design for use in other robot control systems. For this two wheel balancing robot, it is anticipated that further research beyond this project, will progress into visual and audio recognition systems allowing communication and autonomous navigation to be incorporated.

Chapter 2

Literature Review

A two wheeled balancing robot consists of a robot chassis and two wheels. As its name suggests, it has the ability of maintaining an upright “balanced” position which is referred to as its stability. It is unique compared to multi-wheeled or track robots because of this ability. It also has the capacity to turn on the spot making it far easier to manoeuvrer. This makes the two wheeled balancing robot an ideal candidate for working in confined areas or in transportation applications.

This area of research is typically undertaken by engineers and enthusiasts as an approach towards developing their research, design and analysis techniques. This robot is chosen as the inverted pendulum system is naturally unstable and provides a classical control problem. Developing and implementing a suitable stability control system that is responsive, timely and successful can be achieved via linear or non-linear approaches. This chapter reviews the literature that is available in an attempt to gain an understanding and appreciation of two wheeled balancing robots.

2.1 Introduction

A two wheeled robot is basically a robot chassis comprising of a symmetrical shape. This shape can be sectored into several layers where the various components are installed and integrated. Components of the robot may include a core processing unit, sensors, wheels, power source and associated printed circuit board circuitry. If the robot is complex, then additional remote control, autonomous navigations systems, visual or audio recognition systems, etc may be incorporated. The two wheels at the base of the robot chassis provide the locomotion which is typically DC motor driven with gearboxes and shaft encoders affixed.

This type of robot is typically undertaken as a self development process allowing young engineers and robot enthusiasts to learn and develop control systems to satisfy their needs. It also provides a tool for comparing the success of various types of control systems for the typical stability control problem. Segway has led the world with its commercial success of its two wheeled transportation device. As more and more people realise the applications of this type of robot within transportation and beyond, the demand will continue to grow steadily in the future. This includes the field of human mobility.

Human mobility is currently, just a dream for many people with disabilities who as otherwise home bound or limited in opportunity. These robots offer new hope that these people can become part of the world around them again. With an aging population in the western world, the market will also continue to grow for home environment robots that could act as care-takers, providing services to the elderly. In the office environment, two wheeled robots will act as servants and perform basic tasks such as mail collection and distribution.

Two wheeled balancing robots comprise of sensors that provide the ability to see and feel the environment around them. Accelerometers, inclinometers, motor encoders and gyroscopes form the stability and motion sensor families available to provide inputs or feedback to control systems. These include information corresponding to the robots current vertical or horizontal positioning and direction. These inputs can be combined in a process known as sensor fusion. Kalman filter is one type that provides a best approximation from the multiple sensor inputs, allowing the devices inherent inaccuracies to be overcome.

A microcontroller provides the computational power to allow the robot to balance itself, based on the sensor input information. However, the effectiveness is related to the control system that has been implemented. Two different approaches can be undertaken for implementing a control system, they can be either software or hardware based. For this project, a software based control system will be derived. Control systems are further classed as either Linear or Non-Linear control systems. Due to the difficulty and complexity in applying non-linear systems, linear is generally preferred.

Linear systems include State Space control, Proportional Integral and Derivative (PID) controller, Linear Quadratic Regulator (LQR) and pole placement controllers. More recently, developments in non-linear control systems have evolved approaches such as neural networks and fuzzy control. The later will be attempted within this project as non-linear control systems provide a far more effective and efficient controller. The motor driver signals will be enhanced by incorporation of in-line PID controllers.

2.2 Existing Two Wheeled Balancing Robots

Previous two wheeled balancing robot projects include the Segway, nBot, Bender, Emiew and Emiew 2. The Emiew 2 robot is the enhanced (evolved) version of the original Emiew. They were both designed and created by Hitachi whilst the Segway was designed and developed by Dean Kamen who later formed the company Segway Inc. The remaining robots that were reviewed were created by robot enthusiasts who have continued to improve the robustness of their designs over time.

The design concepts between these robots are very similar. Each typically utilise a gyroscope to measure tilt, shaft encoders to measure distance and a microcontroller for performing the computations. These components combine to provide the basis of maintaining stability. Inclinometers or accelerometers are sometimes added to reduce the effects of gyroscope drift thus enabling a more accurate input signal for the control system.

Linear control systems are the most common due to a larger availability of literature in this field and an easier implementation process. Modern developments into neural networks and fuzzy controllers are encouraging a new approach in non-linear control systems. Many of the enthusiast suggested that a non-linear approach would have improved the robustness of the stability control in there robots hence a fuzzy controller will be attempted in this project.

Segway (Segway 2008) is the commercially available two wheeled robot that is currently in its 2nd generation of released models. It is marketed to the world as a transport alternative with the image contained within the following figure. Its advertising suggests the robot is ideal for adventure, commuting, law enforcement and transportation in general. Its trajectory control is based on the tilting direction of the handlebars which is provided by the rider. This robot is capable of achieving a speed of 20 Km/h and is available in Australia for a cost between $9385 and $10795 depending on the model.

Figure 2.1 Segway HTi series two wheeled transport (McComb & Predko 2006)

EMIEW (Kageyama 2007) stands for “Excellent Mobility and Interactive Existence as Workmate”. It was the first two wheeled robot produced by Hitachi and was released in March of 2005. It stood at a height of 1.3 m and weighed over 70 Kg. Emiew 2 followed in November 2007 and is approximately half the size of Emiew at 0.8 m and 13 Kg. Its design concept hoped to reduce the safety risks that were associated with Emiew larger size, incorporating reductions in height and weight.

The robot responds to speech commands and is intended to work in office type environments. To avoid obstacles and people within its path, the robot utilises laser radar to derive a map of its surround area. The knees contain an additional set of wheels that can be utilised when stability becomes too difficult to maintain. In additional to these features, the robot has the ability to raise each wheel by approximately 30 mm allowing it to avoid small obstacles. Emiew is contained in the following figure.

Figure 2.2 Emiew 2 by Hitachi (Kageyama 2007)

David Anderson, an enthusiast, has developed the robot named nBot (Anderson 2007). This robot utilises a gyroscope and accelerometer whose outputs are fused together by a Kalman filter, thus providing an accurate input to control the stability. At present, the robot is in its fourth revision and has the ability of navigating a 7.3 metre distance before returning and repeating the lap once again. One of the strong capabilities of this robot is the ability to transverse rough terrain and even travel down sets of stairs.

Figure 2.3 nBot by David Anderson (Anderson 2007)

Bender (Larson 2008) is a robot made from aluminium and PVC plastics. Its weight is mounted higher in the chassis as it was suggested by Ted Larson that this makes the robot easier to balance. Whilst experimenting with the gyroscope and accelerometer positioning within the robot, he also discovered that the system was much more stable when they were positioned lower in the chassis. These ideas for improving the stability of the robot could be very beneficial during the design phase and initial development.

Other projects on the internet that have not been specified here have presented interesting ideas in the application of the robots construction, control approaches and the resources used. Some robots have been made from Lego blocks whilst others from old materials found around the home. Different approaches to measuring the tilt were also employed with one such robot achieving its tilt measurement through use of paired Infra-Red (IR) sensors. One is place at the front of the undercarriage whilst the other at the back. The distance to the ground is measured by both and then compared for the actual difference. This difference provides the magnitude value whilst the shorter measured distance indicates a tilt in that particular direction.

The majority of these projects have employed linear based control systems in the designs and projects. Many stated that their system did provide a stable system but oscillations about the vertical position were very common. It was also noted that frequent over or under corrections lead to incidences where total loss of stability was experienced. This evidence suggests that robust stability can be achieved if a non-linear approach was undertaken; this project aims to fill this void by providing an insight into the non-linear fuzzy controller, its capabilities and limitations through this application.

2.3 Purpose and Benefits of Two Wheeled Robots

The purpose of two wheeled robots is difficult to limit to a specific role as they can complete numerous tasks with the necessary attachments installed. One purpose could be to access a hazardous or confined environment which would be difficult to manoeuvrer around for a track or multi-wheeled vehicle. This is easily achieved by a two wheel robot as it can turn on the spot by rotating the right wheel forward whilst the left rotates backward and vice versa.

Robots don’t require annual leave, superannuation or a salary. There down-time would be minimal being an electronic based machine when compared to there mechanical counterparts. Maintenance costs could be equated to a human’s annual health insurance and would form the only expenditure after procurement. These ideas contribute to the benefits of any two wheeled robot independent of their purpose that they may be called upon to fulfil.

They could be controlled remotely or autonomously, depending on the terrain and frequency of obstacles around them. In cities and communal areas of society, it is very common for recreational, social and vocational areas to be reasonably flat, therefore traction would be the main concern. Waterways and very rough terrain would be the only problem with this current level of design. Society could benefit substantially from sharing their environment with two wheeled robot assistants. The following roles are only a small sample of the greater potential of possibility.

A robot could fulfil the role of a home health assistant as it doe not require rest. It could provide 24 Hr comfort for families with sick or elderly members. They could easily manoeuvre about the home and provide wireless connectivity to the internet or private networks. This could be used to raise alarms if an incident occurs or if other additional sensors such as a heart monitor enter an alarm state. A built in camera, speaker and microphone could provide a means of communication for medical personnel online, direct to the patient.

It could be utilised as a warehouse tracking robot where item statuses can be easily tracked as part of a schedule, or manually when requested by clients. It could automatically update the companies system and databases through a wireless connection. Other available features would be the ability to perform audits, stock checks as well as conduct security tasks whilst on duty within the warehouse environment.

In a fire warden role, the robot could access a smoke filled area and evaluate the risks before any human lives would be placed in danger. A camera would provide visual images to the fire commander whilst temperature sensors would measure hot spots on bulkheads, deck heads and flooring. IR sensors could search for trapped people or living creatures and provide mapping of obstacles within the room. All indications, measurements and information could be relayed directly to fire commanders allowing a swift and effective effort to prevent life loss and reduce potential damage.

Another purpose could be a general communication robot. They would perform the tasks of a travelling telephone, videophone, fax, Voice Over IP (VOIP), MSN messenger or interface for any other messaging means that may become available in the future. When it receives a call request, it could utilise it sensors to locate the person by itself. Room sensors could be wirelessly integrated with the robot so it can easily identify people within a larger area. This could also be extended to include the detection of intruders where the robot would investigate detections before raising the alarm with authorities.

An aim of this project is for the completed two wheeled robot to be capable of recovering deliberate tilt caused by physical force. This could be caused in the real world by an ill adult or child falling over or even passing out thus bumping into the robot. This would allow the robot to maintain stability and seek assistance from other people by raising the alarm. Beyond this project timeline, it is anticipated that the robot will also become capable of remote control trajectory and multimedia interaction.

2.4 Modes of Operation and Control

Autonomous operation is the goal of most robot developers. Being autonomous suggests the robot is capable of making the decisions itself and performing the necessary actions. It is self-directed and self reliant. This type of control system is self contained with no outside control interaction from humans. Autonomous systems are typically difficult to implement due to the complexity of inputs and outputs, and the variables available. Common problems are associated with the sensitivity, responsiveness and other reactive factors of the components used within the final system.

Non-autonomous operation is much easier to implement as a human still makes the decisions on what actions to undertake. This is typically achieved via remote control facilities for the human. Within the two wheeled balancing robot projects encountered during the literature review, it can be seen that stability control is normally autonomous controlled whilst the trajectory is predominantly non-autonomous based. Sensory such as cameras and radar devices could provide terrain identification and object detection to assist an autonomous trajectory design but this is outside the scope of this project due to time constraints.

2.5 Implications and Ethics of Technology

Implications of technologies, especially associated with robotics, bring a fear that people will no longer be competitive for employment opportunities, particularly, in the future. This idea is encouraged with the offer of cheap labour robots that are slowly becoming available in the marketplace. Although this appears to be true on face value, the majority of jobs that robots are undertaking include those that people consider to be hard labour, repetitive and hazardous.

Reducing the exposure of people to harmful chemicals, environments and conditions as well as the risk of physical injury should be embraced by society. A robot can easily have parts replaced if an incident occurs but replacing an arm for a human is not quite as simple. This idea also applies with repetitive tasks were human workers may become fatigued or bored which ultimately leads to mistakes or potentially fatal errors. These mistakes can cause significant resource and time loss for a company or organisation. This is easily avoided by using robots as they don't become bored and with adequate maintenance schedules in place, they will remain accurate and precise in performing their roles with minimal downtime.

It is unlikely that society will witness a large reduction in employment. In fact, there would be a redistribution of employment due to changing vocational needs. Instead of mining deep underground with the threat of suffocation, collapse or expose, the worker can now be employed on the surface, providing maintenance and engineering functions on the robots and associated equipment. Ultimately, robots will increase the collective work output of nations further strengthening economic security for its citizens and providing an opportunity for people to achieve a balanced work/life/home lifestyle.

Other implications suggest that robots could fulfil roles of personal assistants, carers and helpers thus reducing the stress and workloads of families living with disability or medical conditions. This would reduce medical and household costs further decreasing burdens on families and people in general. This could also extend to the public health system where each person receiving treatment can be monitored and attended to for 100% of the time. Quality of life could also be improved for those who lack the ability of motion. People around the world will have the ability to retain their independence, even into the later years of their lives.

Negative implications may occur if corporations or governments begin eliminating a large volume of human job positions without creating positions in other fields/areas. Unfortunately, this relies on all members of society accepting their moral and ethical responsibilities and obligations as people of the community. This can be achieved by implementing law to protect this from occurring. There is some potential for negative impacts due to negligent use of the technology but the positive gains within all levels of society far exceed any perceived risks.

2.6 Conclusion

Two wheeled balancing robots consist of a robot chassis, two wheels and a stability control system. Additional components and attachments may be fitted depending on the task required to be performed. There strengths lie in the ability to turn on the spot and easily manoeuvre in confined areas. Stability is achieved by keeping the wheels beneath the mass of the robot chassis. Utilising a non-linear control system for stability will allow a more effective and robust control to be implemented.

Two wheeled balancing robots offer a revolutionary transportation capability. They also offer the ability to fulfil any role and complete any task such as a carer or worker within the community. Now that an understanding and knowledge of two wheeled balancing robots have been achieved, the following chapter will begin the design and development section of this project.

Chapter 3

Design and Development

The design phase of a project is fundamental in evolving the ideas, requirements and objectives of the components that together, will form the completed robot. Development and careful design considerations provide the engineer with the ability of ensuring that the concept remains viable as it progresses. It also provides the opportunity to make continued improvements in its operation, ensuring that obvious problems and potential faults are removed early in the project. This ultimately saves valuable time and resources over the duration of the project.

3.1 Introduction

The literature review conducted in the previous chapter provided a wealth of information on past robotic projects that had been completed by enthusiast and engineers alike. This indicated that linear control systems with remote controlled locomotion were very common which inspired this project to undertake a completely different approach. This revolved around a design based on a non-linear fuzzy controller for stability. The review provided ideas and consideration which will be explored in the following sections of this chapter. These include how it will move, how it will sense the environment around it, how it will be powered and what it will be made from.

The two wheeled balancing robot will be broken into two distinctive components of operation. The first is the balancing component “Microcontroller system” which will fulfil the fundamental goals of the project. This encompasses the sensors, chassis, locomotion and microcontroller with its associated control system. The second involves the interaction of the robot with the controller as well as with people it may encounter. This component is called the “Interaction system” and encompasses the wireless network connectivity, interaction computer, audio components and web cameras.

3.2 Control System

The purpose of a control system is to keep a system or plant, within a specified range of elements and set variables. This could refer to numerous applications such as production, assembly and industrial plants through to computer, electrical and electronic systems. For this project, it refers to the control system charged with maintaining stability of the robot chassis. Controlling the stability of the balancing robot requires sensors to detect the direction and rate of motion as well as a decision based application that will provide the response signals to the motors. Many forms of linear systems such as LQR, lead-lag, PID, state feedback and pole placement are commonly used in robotic stability designs.

Advancements in non-linear control systems such as neural networks and fuzzy controllers present the opportunity for developing far superior, responsive control systems. Non-linear is the preferred control system for this robot as a reliable, robust and stable platform can be achieved. There are several types of fuzzy controllers derived to date including the direct fuzzy (non-adaptive), adaptive, fixed and supervisory fuzzy controllers.

The fuzzy controller operates differently to these classical linear controllers as it refers to a set of rules to provide a range of responses, rather then a single predetermined output variation to the input. The input of the fuzzy controller undergoes fuzzification upon entry. Fuzzification is the process of modifying the input into the required rule base format. The inference mechanism then determines which set of rules are referred to for deciding on the necessary output response. Once this resultant action is determined, it then undergoes defuzzification before being outputted to the system under control.

Ideally, adaptive fuzzy control would have been chosen as it is capable of learning how to improve its performance but it had to be overlooked due to time constraints. This is achieved by comparing the output response and the corresponding feedback values returned to the controller. By monitoring and reducing system errors, the controller will learn to control more effective and efficiently then it would by utilising its default rule set. The idea of combining several fuzzy controllers was also contemplated but the risk of increased complexity and difficulty controlling changes was too large. These cascaded systems could have comprised of speed, stability and locomotion controllers.

The goal of the fuzzy controller being implemented is to be capable of recovering from deliberately induced tilt above the rate of 15^o per second. These results are not easily realized under linear control systems thus a fuzzy controller approach is preferred. Choosing a sampling rate that will be capable of performing successfully was taken on the principle of Nyquist sampling theorem. This suggests that the sampling rate is twice that of the highest operating frequency of the system.

A Direct Fuzzy Controller was designed for this project encompassing four inputs and a split output. These inputs are displacement (horizontal) provided by the shaft encoders, velocity (horizontal) derived from the derivative of the displacement, angle (vertical) provided by the inertial sensors and angular velocity (vertical) derived from the derivative of the angle. The output signal will be split and provided to the left and right motor controller routines.

Size	Value	Input 1 (rad)	Input 2 (rad/s)	Input 3 (m)	Input 4 (m/s)	Output (%)
Negative Large (NL)	1	-pi/5	-5	-0.1	-0.45	-100
Negative Medium (NM)	2	-pi/9	-2	-0.07	-0.3	-70
Negative Small (NS)	3	-pi/28	-1	-0.04	-0.15	-30
Centre (CE)	4	0	0	0	0	0
Positive Small (PS)	5	pi/28	1	0.04	0.15	30
Positive Medium (PM)	6	pi/9	2	0.07	0.3	70
Positive Large (PL)	7	pi/5	5	0.1	0.45	100

Table 3.1 Fuzzy controller membership functions

For the system to be balanced at equilibrium, all four inputs are ideally at zero. The fuzzy process encompasses seven memberships for rating the strength of the inputs within this system and is contained in the previous table. These memberships are then incorporated with each of the inputs and output to form functions known as membership functions. The name of these types of membership functions comes from its shape, such as triangular which was chosen for this project.

Triangular, trapezoid, gaussian and bell membership functions are a few that may be produced. They represent linguistic values such as positive large through to negative large with a minium, maximum and centre value. The shape they undertake reflects how the values a portrayed. This is particularly important as centroid of area was used in the defuzzification of the triangular output membership function. This allowed an actual value to be selected based on the fuzzy rule set.

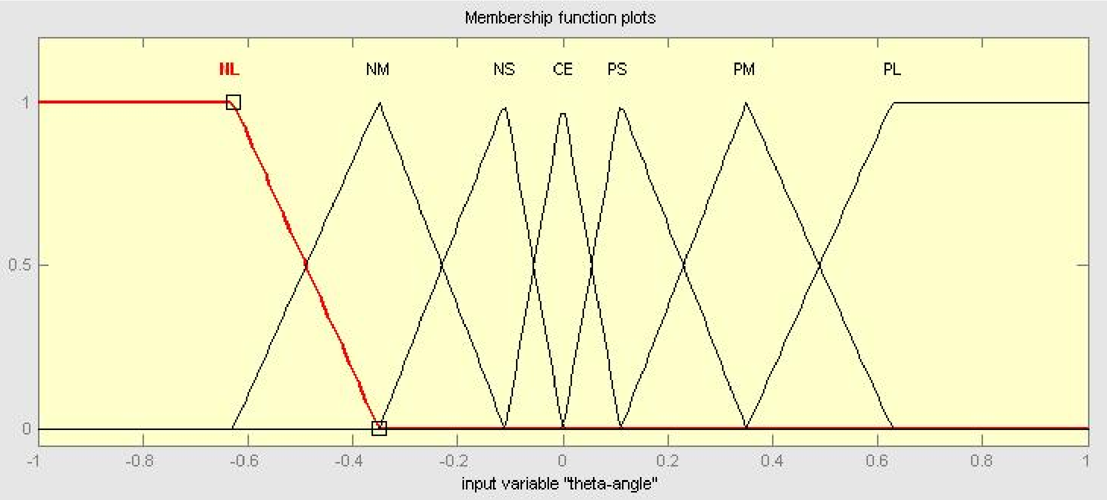

Figure 3.1 Input 1 - Angle membership function

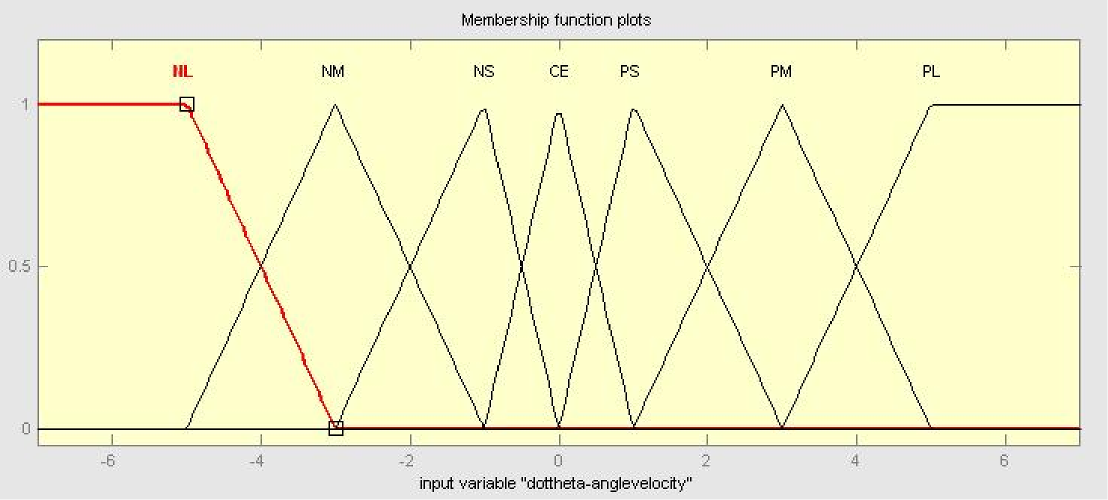

Figure 3.2 Input 2 - Angular velocity membership function

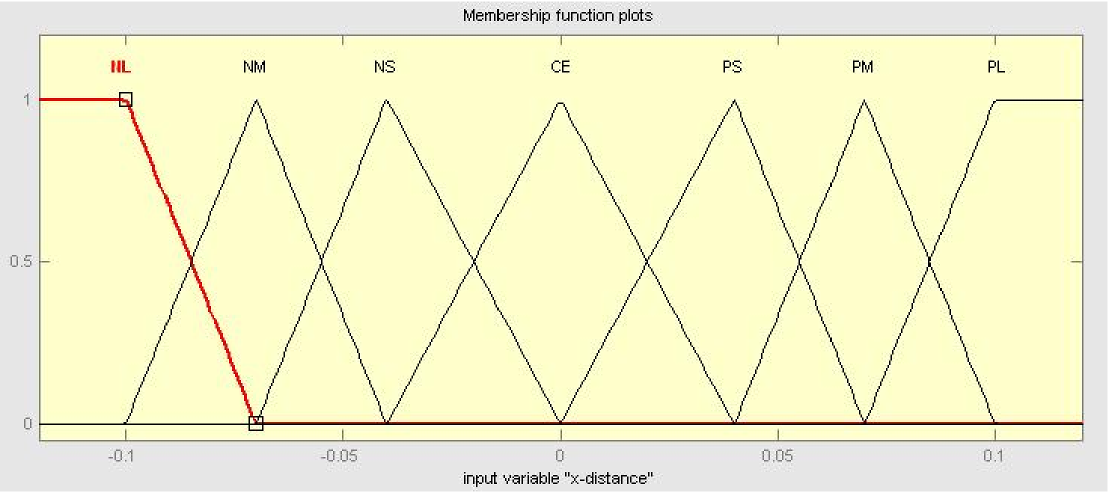

Figure 3.3 Input 3 - Displacement membership function

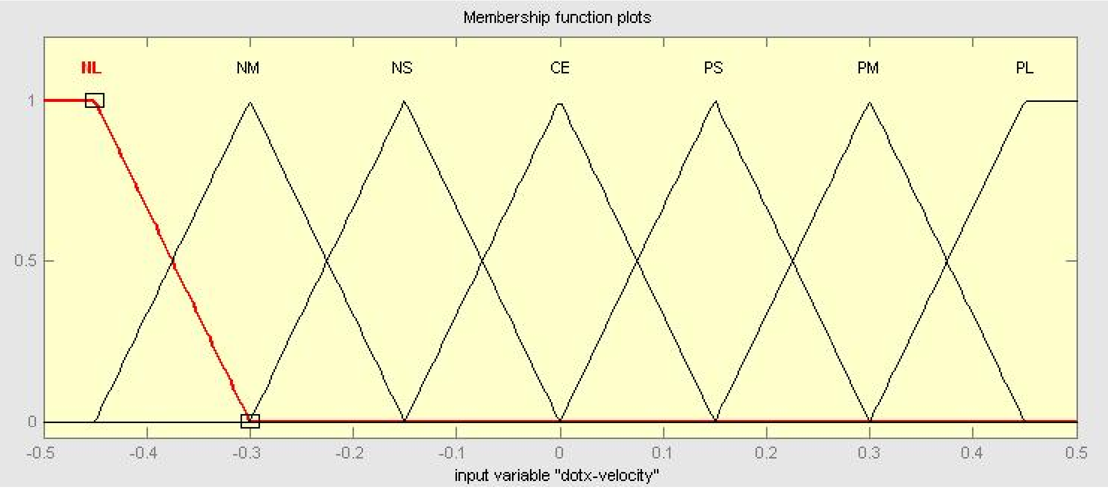

Figure 3.4 Input 4 - Velocity membership function

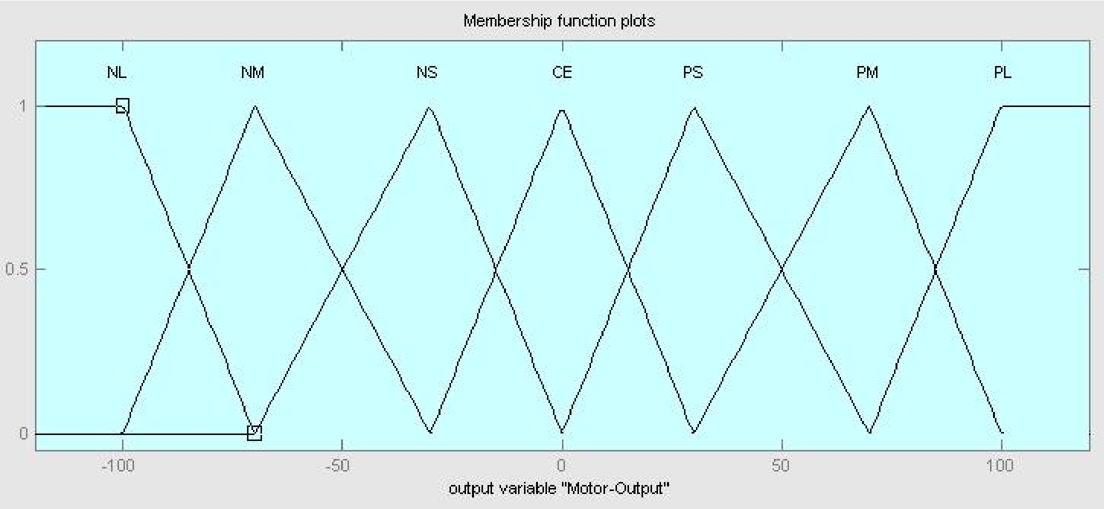

Figure 3.5 Output – Motor output membership function

The previous figures contain the visual representations of the respective membership functions. As seven membership functions were decided on for each of the four inputs, 2401 conjunctive rules (7*7*7*7=2401) would be required. This was reduced to approximately 427 rules by giving the vertical inputs a higher priority, as maintaining a small vertical angle is crucial for preserving stability. The horizontal inputs are only enacted upon when the vertical is in a centred state. It is essential to reduce the rule set as much as possible as its size directly impacts on the processing time necessary to search it.

	Angle						
Angular Velocity	**NL**	**NM**	**NS**	**CE**	**PS**	**PM**	**PL**
NL	PL	PL	PM	PM	PS	PS	CE
NM	PL	PM	PM	PS	PS	CE	NS
NS	PM	PM	PS	PS	CE	NS	NS
CE	PM	PS	PS	CE	NS	NS	NM
PS	PS	PS	CE	NS	NS	NM	NM
PM	PS	CE	NS	NS	NM	NM	NL
PL	CE	NS	NS	NM	NM	NL	NL

Table 3.2 Angle and angular velocity rule relationship

	Displacement						
Velocity	**NL**	**NM**	**NS**	**CE**	**PS**	**PM**	**PL**
NL	PM	PM	PM	PS	PS	PS	CE
NM	PM	PM	PS	PS	PS	CE	NS
NS	PM	PS	PS	PS	CE	NS	NS
CE	PS	PS	PS	CE	NS	NS	NS
PS	PS	PS	CE	NS	NS	NS	NM
PM	PS	CE	NS	NS	NS	NM	NM
PL	CE	NS	NS	NS	NM	NM	NM

Table 3.3 Displacement and velocity rule relationship

The previous tables contain the rule relationships of the vertical and horizontal input sets respectively. This provides the fundamental component of the rule base. For example, if the angle and angular velocity were both negative medium, then a response of positive medium is required at the motors. If the output required was nil (centre), then the horizontal relationship would provide the output response. The defuzzification method known a Centroid of Area (COA) was the preferred method for the output as it allows an accurate translation to real output values. With the output already scaled between 0 and 100, this could be translated directly to the percentage of PWM required.

Procedures (Deepa, Sivanandam & Sumathi 2007) for creating the fuzzy control system within Matlab “Simulink” are simplified by a built in fuzzy toolbox. The following figures contain surface plots for each of the rule sets that were created. These reflect the rule relationships contained within the previous tables. The resultant plots are very similar at first glance but the vertical rule set contained a more rigid surface. The horizontal rule set provided a more gradual variance which is more reflective of a fuzzy control system. Simulation of the controller was then possible with the use of the ‘slcp’ command although this avenue was not pursued.

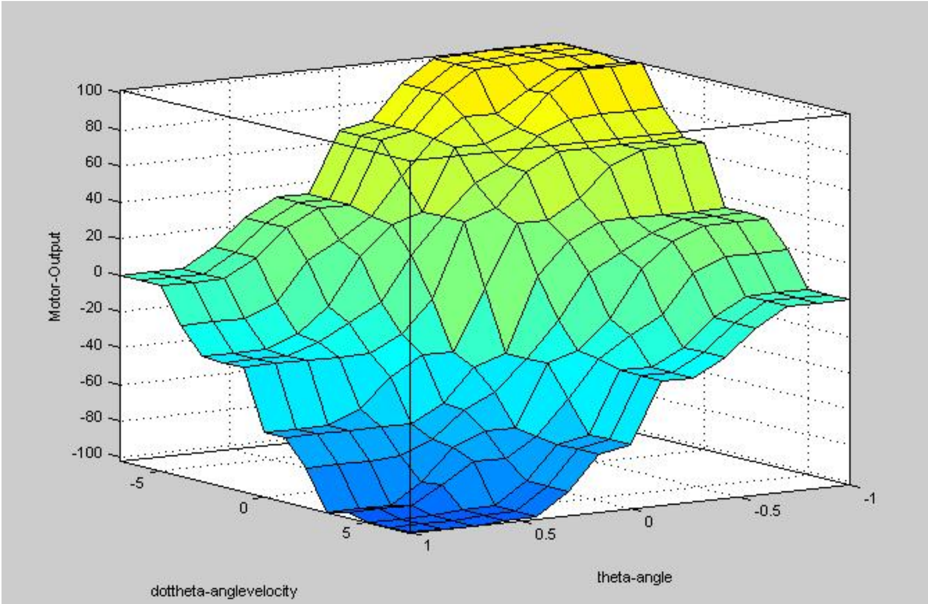

Figure 3.6 Angle and angular velocity rule set surface plot

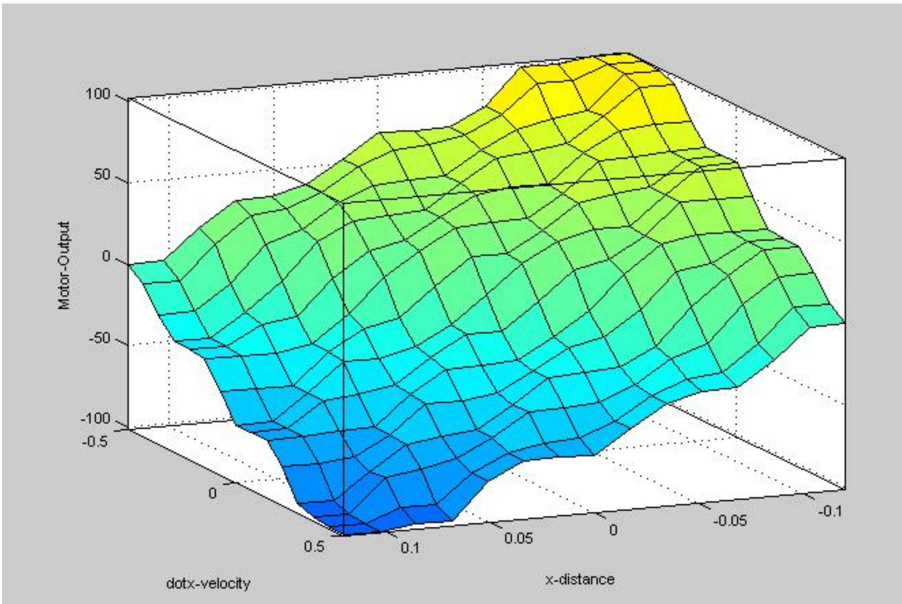

Figure 3.7 Displacement and velocity rule set surface plot

3.3 Software and Programming Concept

Programming involves the structure of logical sequences and processes that are incurred in reaction to various input variables of a system. This programming will enact an output action or outcomes in response depending upon its construction. The program containing this programming is known as software. Software may comprise of several programs incorporated together to form a 'software package'. Software is typically used to control hardware or external devices although it is more prominently used to control other software programs and processes today.

When a software program appears to be performing numerous tasks seamlessly, it usually contains several programs working in conjunction. To achieve this, programs are again broken into routines and sub-routines which reduce the overall size of the main program. This improves efficiency and responsiveness by only calling blocks of programming that are necessary for the specific task at hand. This has the effect of reducing the microcontrollers processing time.

The programming necessary for the two wheeled robot encompasses the microcontroller and interaction systems. The microcontroller is required to read in an assortment of sensors, provide computation on these inputs, make a judgement on the necessary actions to undertake and then respond by providing drive signals to the motors. This is purely to enact stability and locomotion control programming. The programming was completed in C programming language and then compiled for the microcontroller by the software tool, MPLAB.

The programming goal of the interaction system was to incorporate a wireless (WiFi) connection thus allowing remote locomotion control, access to onboard sound and visual devices, and to provide a means of reprogramming the microcontroller. The Microsoft Robotics development Studio package would have provided the programming tool using visual basic programming techniques. This component was not completed due to the timeframe but it will be endeavoured in the future.

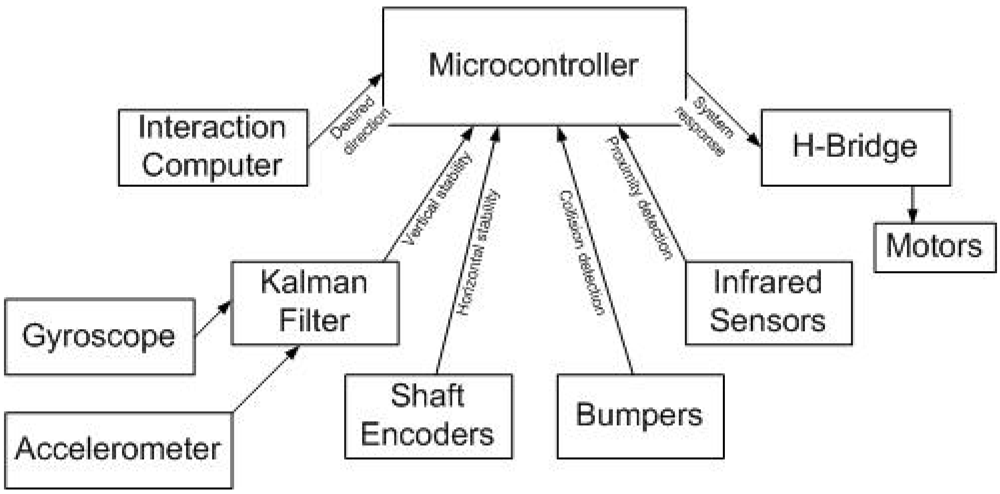

Figure 3.8 Microcontroller structure chart

The structure chart in the previous figure represents the basic direction and purpose of system signals. Design of the main program was achieved by using this chart to create flowcharts to represent each of the tasks and actions necessary for completing the microcontroller's function. The main program will be quite small, calling on the sub-routines to perform specific actions. The sub-routines represent blocks of steps needed for successful completion of the task. Understanding the variables and identifying the input/output ports necessary is crucial for simplifying the process of deriving these sub-routines.

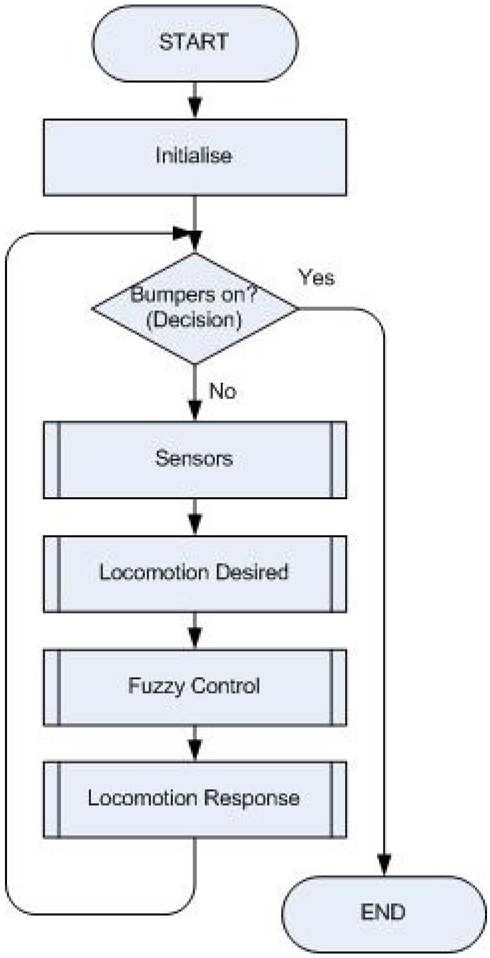

Figure 3.9 Microcontroller main program flowchart

The main program begins with an 'Initialise' block which sets up the stored values and lookup tables. This block is only executed upon start-up or reset. The remainder of the looped program contains four main sub-routines and a decision block. The sub-routines are discussed in the subsequent paragraphs below. The decision block has been incorporated to determine if the bumper collision switches have been activated.

Upon activation the program will end causing the robot to stop driving the motors, thus releasing any driving force that may have been applied. The program may be reinitialised at this point by resetting the microcontroller. If the switches are not activated, then the program will continue through the looped 'while' sequence.

The first sub-routine is called 'Sensors'. Its initial block reads the digital sensor inputs before storing there values. The next reads analogue sensor inputs before applying an A/D conversion and storing the value in memory. The next block performs a Kalman Filtering estimation of the inertial sensors with the final block deriving the angle, angular velocity, displacement and velocity values (stability values) which are then stored for use within the 'Fuzzy Control' sub-routine.

Sub-routine 'Locomotion Desired' reads the user provided forward, backward, left and right direction control signals. The following two blocks then determine a resultant left and right motor response respectively, which is stored for use within the next locomotion sub-routine. These values are also used within the subsequent block to derive an offset value for the vertical angle measurement. This will allow the robot chassis to be tilted into the direction of desired travel thus allowing more torque to be achieved.

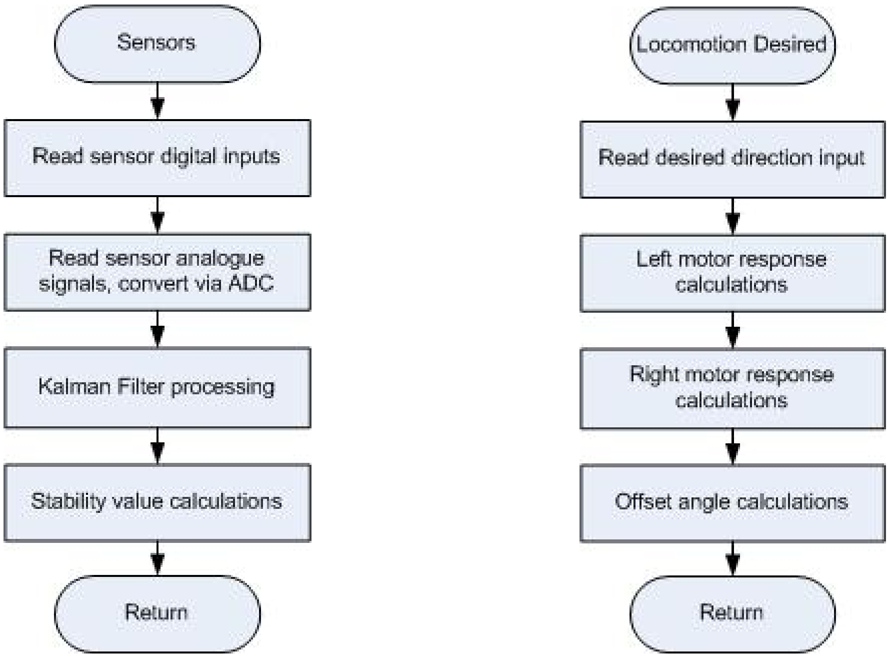

Figure 3.10 Sensors and locomotion desired sub-routine flowcharts

The 'Fuzzy Control' sub-routine forms the core of the control system. The initial block converts the stability values to there respective equivalent linguistic membership values. The next block then compares these values with the rule set to determine the desired output response for the motors. The remaining block then performs a defuzzification to arrive at a motor output signal response.

The final sub-routine 'Locomotion Response' begins by utilising the desired locomotion output and fuzzy control output signals within the calculation of the final motor output responses for each of the motors. Two more blocks then perform PID calculations on each of these motor responses with the final block actually providing the output to drive the motor circuitry.

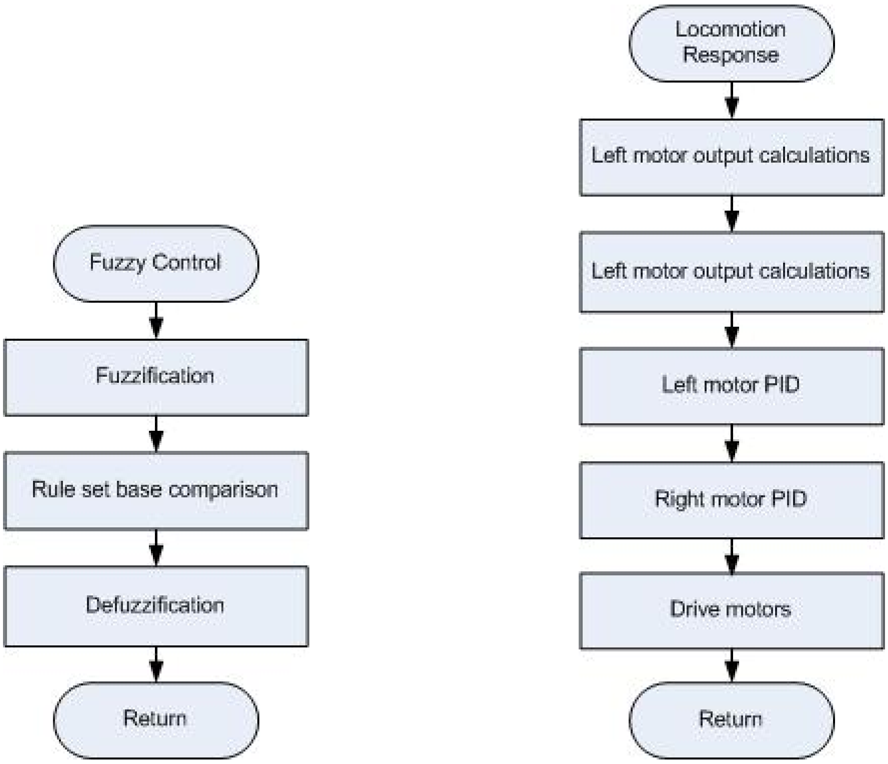

Figure 3.11 Fuzzy control and locomotion response sub-routine flowcharts

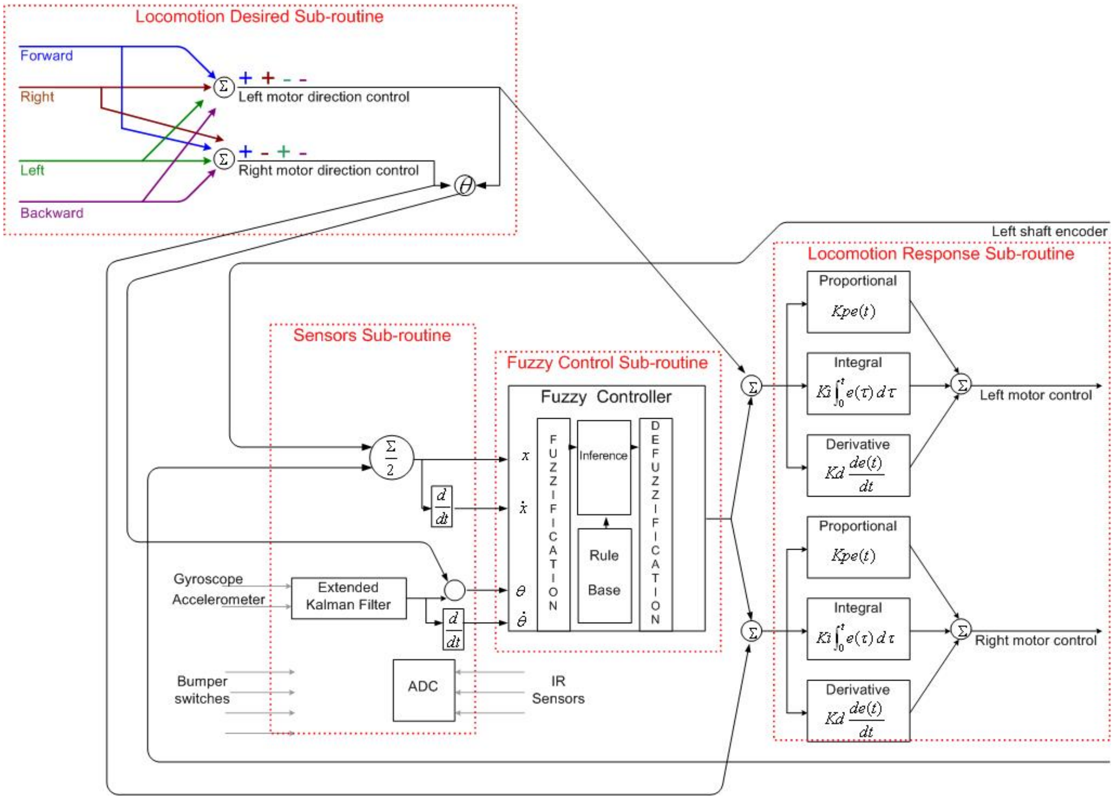

Figure 3.12 Control system functionality

3.4 Design Considerations

Design considerations are an important factor when compiling a design because they facilitate an informed decision on the limitations, operating conditions and capabilities of the final product. This process may also highlight areas for further investigation that could improve, simplify or make a product more cost effective. This includes commercial aspects that may be undertaken if the product was to become mass produced or if an alternate material or component usage was to be implemented to reduce the potential manufacturing costs.

Design considerations for the two wheeled robot encompassed operating environments, motors, wheels, sensors, microcontroller, communications, power source, safety features, electromagnetic factors and various forms of control. All these considerations were then sub-investigated to determine dimensions, configuration, temperature parameters, maintenance requirements, availability, efficiency, etc. Consideration of these factors combine together to guarantee the necessary capability is achieved successfully within the timeframe and resources available.

3.4.1 Operating Environments

The operating environment of the machine refers the conditions under which it will be performing its tasks. Primarily, this can be broken down into two main environments, indoors and outdoors. The outdoor environment would be much more hash with exposure to numerous varieties of weather including rain and hail. Temperature variation could be as much as 50^{o}C and when combined with continuous direct sunlight, could pose serious performance degrading if a robust design was not incorporated.

An indoor design proves to be a simpler process as the environment is normally regulated by air-conditioners. This also reduces the expose to dust and dirt particles, direct wetting and even direct sunlight (UV). The terrain is also smoother when compared to the rough outdoors through paving, carpeting and tiling. Neglecting to adequately prepare a machine for its operating environment may lead to premature breakdown, potential safety implication and loss of productivity in a commercial situation.

The operating conditions expected for this robot includes both indoor and outdoor environments. Ideally, the robot will be capable of travelling across outdoor areas with up to 100mm crevices within the confines of local infrastructure. These include sandy beaches, sports ovals and bicycle tracks. Incorporating some form of waterproofing would provide protection from rain or accidental wetting as moisture entering the robot could cause corrosion (mechanical) or short circuit (electronic) damage. It will not be operated during wet weather conditions so external casing and coverings are not deemed essential.

The maximum temperature expected is approximately 60°C. This remains below the rated maximum temperatures of IC's, components, cable and part ratings. Direct sun could raise the temperature beyond this posing a potential problem unless cooling is implemented within the robot chassis but this will not be pursued. Vibration would remain manageable for the final size and weight of the robot. Extra care was taken in reducing the vibration risk by securely fastening all components and circuitry in place as well as the inclusion of rubber matting between the motors and chassis.

3.4.2 Materials, Size and Weight

The materials used in a design limit the durability, strength, maintainability, energy efficiency and operating capability of the product. They also impact on the size and contribute to the total weight of the final design. A heavier material such a metal is typically stronger then lighter materials such as plastics but it requires more energy to move. The size of the design impacts on its ability to navigate around, through or over obstacles as well as its transportability.

Materials that could have been used in the robots construction include metals, alloys, wood, plastics and other synthetics. Heavier metals were excluded early due to the increased weight that would impair the stability control of the robot. Although wood was an option, it is a potential ignition source in higher temperatures and is also very difficult to mould. Rubber strips could also be utilised in the shock resistant precautions that may be experienced from falls, bumps or rough terrain but this was only incorporated to support the battery and motors within their mounts.

As the robot will be utilised in a people environment, a maximum height of 0.8 m was chosen as it approximates to a child's height. The robot chassis has a multi-platform layout where each platform (layer) contains a unique component of the robot systems. An average height of 120 mm will separate each layer with the exception of the power level due to the limited battery sizes available. This is 150 mm. Additional layers could be added to the top of the chassis, permitting easy expansion of the robots capabilities thus allowing greater flexibility for adaptability and enhancements.

A diameter of 280 mm was chosen for the layers as it provided ample area to mount the PCB's, components, etc. This being a rounded shape as opposed to square allows the robot to easily revolve on the spot and around narrow areas without having to square itself up with the perimeter. A combination of alloy metal for the frame and plastic for the layers was used as they provided a compromise between weight, durability and strength. Utilising trigonometry mathematics, a metal length of 275 mm was derived for mounting the layers and adjoining the side lengths of 510 mm. The metal width was a standard 97 mm wide.

Several options are available for joining the chassis pieces together such as bolts, screws and welding but rivets were the preferred method in this case. Epoxy adhesives were utilised in joining the plastic pieces together. Additional plastic brackets were manufactured to attach sensors and PCB's which were also affixed into place with epoxy. The completed chassis weight goal was initially 5 Kg but this was exceeded by 1 Kg by the final manufactured product. The weight values were calculated for each respective layer as follows:

- Upper layer, 0.5 Kg,
- Interactive system layer, 0.5 Kg,
- Power layer, 7 Kg,
- Spare layer 0 Kg,
- Microcontroller layer 1 Kg,
- Motor & wheels level, 2.5 Kg,
- Chassis structure, 6 Kg,
- **TOTAL** combined weight, 17.5 Kg.

An outer skin layer such as a thin plastic sheeting (1-2 mm) could be applied to the robot allowing a sleek appearance. The sensors would require that the outer skin is cut to allow proximity detection, etc. Removable panels would be required to allow the maintainer or operator to access certain areas of the robot for maintenance and connection of peripherals. This option has not been pursued.

To improve the ability to balance, the aim was to place the larger weights higher in the chassis layers. This principle can be equated to balancing a broom in the palm of your hand. When the broom head is within your palm and the handle upwards, it is difficult to maintain the length in a vertical position. Reattempting this task with the handle in your palm and the broom head high proves far easier to balance. Another goal was to centre the weight over the wheel axles, for each layer, to contribute to the stability of the robot.

The battery acts as a destabilising influence on the system due to its large weight of 7 Kg in the upper half of the robot chassis. This allows the robot to maintain stability easier as more torque can now be generated. Total weight of the robot was initially set at a maximum value of 15 Kg but this was overshoot by a total of 2.5 Kg.

3.4.3 Wheels

The wheels provide traction and locomotion for the robot as well as support for the robot chassis. The diameter of the wheel should be chosen to best reflect the torque requirements of the machine. The tyre used will impact on traction and the smoothness of the motion experienced by the inertial sensors and sensitive components onboard. The number of wheels to be used in a design should depend on requirements of the capabilities.

To overcome poor traction, higher levels of torque are required which is ultimately supplied by the motor. Ideally the tyre traction would have maximum contact with the ground and minimal resistance through the drive system. A width between 30-45 mm of a rubber tread was deemed acceptable as it would provide reasonable grip for most terrains and weather conditions. It would also prove ideal for turning on the spot which is a major feature of a two wheeled balancing robot.

Each of the two wheels will be independently controlled in either direction by a dedicated motor. The wheel could potentially have a large diameter such as a bicycle wheel but this would prove impractical for manoeuvring in confined areas or indoors. The diameter required for the wheels was approximated between 150-190 mm so that proportionality could be maintained with the anticipated robot height of 800 mm and weight of 17.5 Kg. The diameter played an important factor in the torque calculations that are further discussed within the motor section that follows.

Two plastic wheels with solid rubber tyres were chosen due to there reduced cost and lower associated weight compared to its metal counterparts. The wheel diameter was 187 mm with a tyre width of 40 mm. As the tyre is solid, it is not susceptible to being punctured although this tyre will likely increase the vibration experienced by the onboard circuitry if rough terrain was encountered.

3.4.4 Motors

Motors are an essential component for providing mobility. Without motors to move the machine, it would prove to be an expensive paperweight. Several types of motor are available including Alternating Current (AC) or Direct Current (DC) types. Variances of these motors include the brushless, brush, servo and permanent magnet motors. Another consideration for the motor is its Revolutions Per Minute (RPM). As most motors are rated between 2000-7000 RPM, some form of gearing is required to reduce the rate experienced at the wheel.

The idea of a motor for each side of the robot, also known as differential drive, was decided upon early to reduce the potential mechanical complication involved with implementing a single motor system. Using two motors will also allow the torque to be more effectively applied to the wheel shafts thus reducing the need for complex calculations for frictional and rotational losses. Differential drive is displayed in the following figure.

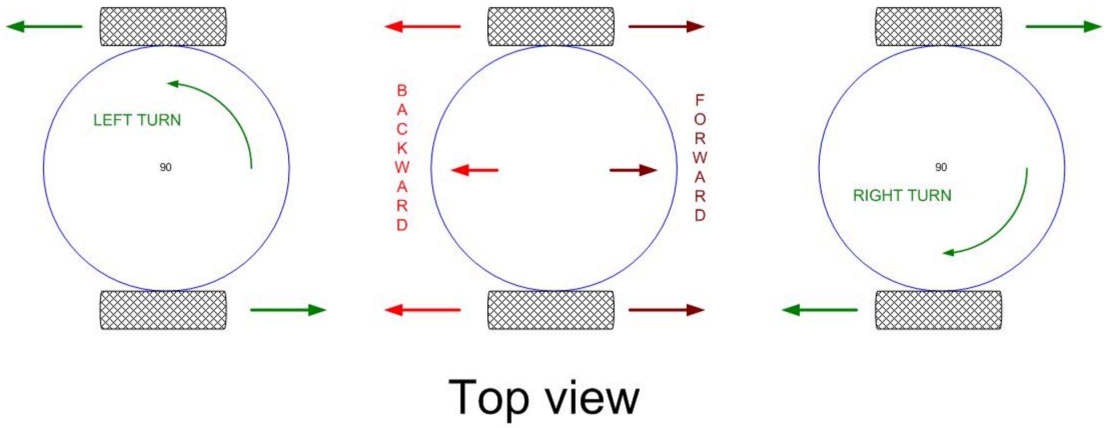

Figure 3.13 Differential drive motion

As the right motor moves forward and the left motor moves backwards, the robot chassis will be turned to the left. If equal torque is applied to each of the motors, the robot will turn on the spot. The same principle applies if both of the motor directions are reversed. If the left motor turns forward whilst the right motor turns backwards, then the robot will turn to the right. If both motors are moving forward, then the robot will move forward and vice versa. Other considerations for the turning of the robot should include the angle of the robot chassis. With this in mind, the main goal would be to tilt the chassis in the direction the robot is about to travel.

A supply voltage of 12V was chosen for the motors although 6V and 24V are also fairly common voltages. The motors would be better mounted under the robot chassis to allow a higher clearance from the ground. Otherwise the lower chassis layer may drag into the ground at a very lower angle of tilt. Stepper motors are too slow to react due to the actually stepping action and will not be capable of balancing the robot efficiently. A motor with a high volume of torque was a necessity with the final choice being two HG37D670WE12-052 DC Brush motors with a 1:52 gear head and shaft encoders fitted.

The required torque for the robot can be reduced by leaning forward into the direction of anticipated travel. This will be attempted within the control system design component. The RPM estimation of the motor is 160 RPM. With the chosen wheel diameter of 187mm, the expected approximated maximum speed is 5.64 km/h. For a rated motor torque of 7.1Kg/cm, the anticipated acceleration at the rated input voltage is 0.086 m/s^2.

Velocity, $v = d_{WH} * \pi * \frac{RPM}{60} = 0.187 * \pi * \frac{160}{60} = 1.57 ms^{-1}$

$v = 1.57 * \frac{3600}{1000} = 5.64 kmh^{-1}$

Motor force, $F = \frac{T}{r} = \frac{7.1}{\left(\frac{18.7}{2}\right)} = 0.759 Ncm^{-1}$

Acceleration, $a = \frac{F}{\left(\frac{mass}{2}\right)} = \frac{0.759}{8.75} = 0.086cms^{-2}$

Power, $P = V * I = T * rps = 7.1 * 2.67 = 19W$

The motors have shaft encoders incorporated into their case, rated as 12 pulses per motor revolution or 624 pulses per shaft revolution. Shaft encoders are necessary to obtain linear position measurements before the velocity calculations may be performed. They are covered further under the sensor section within this chapter. Although the retailer recommended a 7.2 V input, the manufacturer "Hennkwell" rated the motors input between 6 and 24V. It was concluded that 12V may be safely applied in its operation. The additional voltage is expected to provide much needed torque to move the total weight of 17.5 Kg.

Motor Control

Motor controllers control the motors rate of turn and its direction by varying the output voltage signal and setting its polarity respectively. There a several forms of motor control with the application dependant on the role and the motor in use. Direct, Relay, transistor or H-bridge (motor bridge) controllers are all common types available. The later provides the facility for a pulse width modulation (PWM) signal output for varying the motor speed.

PWM provides an output voltage that is varied by a duty cycle. This duty cycle is a percentage of voltage time high compared to voltage output and specifies the 'pulse width'. The speed of the motor is altered by altering this duty cycle percentage. A benefit of this approach is that there is no digital to analogue conversion necessary so there is no potential for induced noise or interference on the signal. It is recommended (McComb & Predko 2006) that PWM is run at frequencies over 18 KHz, higher than the maximum human hearing frequency, to avoid hum noise generation within the motors.

For controlling the robots motors, PWM will provide the best option as using direct voltage level inputs may cause the robot to 'hunt' about its position. This occurs as the equilibrium point is exceeded consistently in either direction thus causing an oscillating effect. The microcontroller cannot drive the motors directly at the necessary current so a H-bridge, called this due to its transistor configuration, will boost the PWM signal to the motor. Smaller motor control changes such as those in PWM, reduce the robots power requirements hence it is an added improvement to the robots design.

A custom designed H-bridge circuit incorporating protection diodes and dual H-bridges was created but not pursued after a commercially available kit, "L298 Motor driver" was discovered. This proved more economical and saved substantial time when compared to the cost of manufacturing and constructing the PCB. Configuring the purchased H-bridge circuit for 12V operation allows each bridge circuit peak output current to be 2.92 A.

Figure 3.14 L298 motor driver PCB (Solarbotics)

The H-bridge circuit also offers LED indication of the direction each motor is driven as well as an additional regulated 5V power supply output. This proved beneficial for isolating the gyroscope circuitry from the main 5V distribution and reducing the gyroscopes EMI impact on the system. The control inputs to the driver allow forward or reverse rotation as well as braking of the motors. Another benefit of the motor driver is the thermal shutdown protection incorporated into the PCB.

3.4.5 Sensors

Sensors provide the robot with the ability to interpret the environment around them. Without sensors, the robot would blindly execute a series of instructions without the capacity to re-evaluate its progress and make adjustments. Sensors are available in two forms, analogue and digital. Analogue sensors provide a range of values in response to its sensitivity which have the benefits of measuring rates or volumes. Digital sensors only provide an on or off value and are ideal for an absolute indication. Sensors may be connected by parallel or serial connections to the microcontroller or Analogue to Digital Converters (ADC) depending on the actual sensor used.

Accelerometers, gyroscopes and inclinometers are the most common types of sensors utilised in robots and machines that require stability control. They provide a means of measuring acceleration, velocity and direction. If multiple axes were to be measured with certain type of sensor then a sensor would be required for each particular axis. By combining different sensors through a process known as sensor fusion, certain sensor problems such as gyroscope drift and noise can be overcome and kept within the required accuracy. Commonly, a gyroscope and inclinometer are combined.

Laser range finders, sonar range finders, ultrasound, microwave or Infra-Red (IR) sensors can be utilised for detection of physical obstacles in the robots path or proximity. The range finders provide a greater range of detection but are also much more expensive. These would have been a preferred option if mapping was part of the robots task. These sensors are mostly active devices as they require a transducer to transmit an energy signal whilst another transducer receives the reflected signal. Some sensors such as the IR type also come in passive form which relies on the signals already present in the surround environment to reflect from other objects.

Other forms of sensors include contact sensors to detect a bump with an object, light sensors to detect lighting changes or conditions, thermal sensors to detect temperature variation and sound sensors that can detect the presence of sound waves. Contact sensors will be incorporated into the robots design to stop the robot moving in any direction. This is aimed at preventing damage to obstacles or injury to people. The five most common sensors used in robotic design are IR, accelerometers, gyroscopes, inclinometers and shaft encoders.

The sensor requirements for the project include inertial sensors for stability sensing, proximity detection for object detection and bumper sensors for contact detection. An accelerometer, gyroscope and shaft encoders form the inertial sensors. IR sensors form the proximity detection whilst standard bumper switches provide the contact detection. They are discussed further in the following paragraphs.

Accelerometers

Accelerometers provide a means of measuring acceleration along an axis, also known as force. They are susceptible to position noise which can be avoided by incorporating filtering on their digital or PWM output lines. Accelerometers are commonly found in airbag and braking control systems of vehicles. Their main purpose is to measure changes in speed. It is classed as an analogue device as the voltage varies with respect to the rate of change.

This device will be used to determine the change in speed with relation to the robots tilt, its change in acceleration. The model chosen was the ADXL311JE accelerometer provided by Excelpoint, NSW. It is specified as a high shock resistant, low power, dual axis accelerometer. This model provides a bandwidth capability within the 1-3 KHz range. The accelerometer output will be combined with the gyroscope output to compensate for gyroscope drift thus creating a better stability control signal. Acceleration of the tilt can be defined by the expression:

$$\ddot{\theta} = Gravity * Sin\theta$$

A common problem of being unable to differentiate between the tilt acceleration and the actual locomotion acceleration will be avoided by using both axes. One axis will detect upward/downward motion whilst the other will detect forward/backward motion. Rough terrain could pose a problem if large bumps or vibration are encountered through the robot chassis. Fitting the device higher in the robot chassis implies a more reliable reading as greater distance will be covered for the same angle of tilt but previous research (Larson 2008) suggests that better accuracy is achieved when the device is placed lower in the chassis. This will be attempted.

Gyroscopes

A gyroscope presents a means of measuring rate of rotation (angular rate) but a common problem is the output drifts over time. This occurs when the vibrating resonator is rotated, causing additional vibration at an angle to the originating vibration. This is also known as Coriolis Effect. A more reliable and accurate reading can be achieved by fusing both a gyroscope and inclinometer together. This uses the inclinometer to detect the rotation and correct the output signal from the gyroscope.

This device will be used to determine the angle of tilt of the robot chassis. The original model chosen was the ADXR150 gyroscope provided by Excelpoint, NSW. The device was not utilised due to the difficulty experienced in attaching the 32 lead ball grid array to a circuit. The replacement device was the Angular-Rate sensor XV-3500CB. The device is capable of 5V or 3.3V operation with a deflection range of +/-100^{o}. The gyroscope output will be combined with the accelerometer output at set intervals to compensate for the Coriolis effect thus creating a better stability control signal. As per the accelerometer, the device will be mounted low within the robot chassis to provide better accuracy.

Sensor Integration – Kalman Filter

Sensor integration is the process of combining multiple sensor responses together into a single, more reliable output signal. In the sensor integration of an accelerometer and a gyroscope, the resultant data provides a much more reliable tilt angle value. This is achieved as the system has now compensated for gyroscope drift that would otherwise provide imprecise indications. The majority of sensors have a degree of unreliability that becomes inherent over time as well as introduced noise which can also be overcome by this process.

Originally developed in 1960 by DR R Kalman for Apollo spacecraft, Kalman filters are one approach that is used to perform sensor integration. They are described as a recursive filter and provide output estimation for the state of the system based on past and present sensor measurements whilst also overcoming sensor noise and other system disturbances. Simple and extended Kalman filters are two of the types that are available which are used on linear or non-linear systems respectively.

The extended Kalman filter was the chosen option for the project. An inclinometer could have also been incorporated with the gyroscope to reduce “gyro drift” but this approach as overlooked with the addition of an accelerometer into the inertial system. Simon (2001) presents a Kalman filtering approach that will be utilised as the basis for performing the sensor fusion within this project. This is covered further within the programming and software component of chapter 4.

Shaft Encoders

Shaft encoders are typically mounted on the motor shaft before the gearing down at the wheels as a higher count per revolution increases its effective accuracy. They provide a means of measuring distance travelled and its direction which is then used as feedback within the motor control computations. This also provides a function for deriving velocity within the microcontroller. Optical encoders are one type that counts every time it detects a light reflection. Magnetic encoders are another type which counts every time it detects a magnetic influence passing its sensor.

The best approach to efficiently utilise the encoders is to use the microcontroller's onboard counter inputs. This does not utilise the microcontroller processing directly which allows other computations to be performed simultaneously. The microcontroller programming should reset the counter input upon actuation of the motors and then perform a read of the counter once the motors are stopped. These counts can then be multiplied by the distance of each pulse to produce a distance reading for horizontal displacement.

This approach will be undertaken in the project. The motors chosen were HG37D form which were pre-fitted with shaft encoders rated at 12 pulses per motor revolution or 624 pulses per shaft revolution. With a wheel diameter of 187 mm (circumference 587.4778mm), this equates to 48.9565 mm per pulse of motor revolution or 0.9415 mm per pulses of shaft revolution. Potential errors in counting could be induced by travelling over uneven terrain or through the loss of traction.

Infra-Red (IR) Sensors

IR sensors can be utilised for detection of physical obstacles in the robots proximity. These sensors are mostly active devices as they require a transducer to transmit an energy signal whilst another transducer receives the reflected signal. The output voltage varies with the distance to the detected object. The IR transmitted signal is often modulated to allow differentiation between the required IR signal and spurious IR emissions. This allows the sensor to overcome noise and false triggering.

Ideally, IR sensors would be placed at 90^{o} angles around the lower and upper layers of the robot chassis to provide more stringent detection network but the added cost of approximately $15 per sensor made this approach uneconomical. IR sensors could have been utilised for measuring the distance from the chassis to the ground also but since this robot is expected to traverse uneven grounds and environments, it would be impractical to use in this manner.

Figure 3.15 Sharp GP2D12 sensor (Sharp)

Sharp IR GP2D12 sensors have been incorporated in the robots design for proximity detection. This will ensure the robot maintains a safe distance from obstacles, people and hazards. Three have been fitted in total with two attached to the lower layer with one facing forward whilst the other rearward. The third has been fitted much higher up facing forward as a fail safe sensor for elevated object detection. This was aimed at detecting objects that may not have a lower base such as tabletops or lamp shades. The pulse cycle duration of the device is 39ms with a 44ms start up delay.

The orientation of the IR sense had the emitter LED on top to improve the reception of the signal as suggested by the manufacturer. This improves the reflectivity of the signal. It was also suggested that addition of 10uF bypass capacitors near the sensor would improve its performance, so they have been incorporated also. Distance detection will be set for 20 cm on the lower layer sensors and 30 cm on the upper layer. The rated voltages for distance detection are as follows:

- 10 cm, 2.45 V,
- 20 cm, 1.35 V,
- 30 cm, 0.95 V,
- 40 cm, 0.75 V,
- 50 cm, 0.55 V,
- 80 cm, 0.4 V.

3.4.6 Microcontroller

The purpose of a microcontroller is to complete computations and process data by executing instructions or programming. They contain all the necessary components expected of a computer system in a single Integrated Chip (IC). These include the processor itself, RAM, ROM and input/outputs. Since they are small in size, they are typically implemented in embedded systems and can be found in numerous household appliances.

Ideally, a microcontroller will maintain a fast response with minimal power consumption and low susceptibility to interference. The IC should contain all the devices necessary for operational requirements such as timers, ADC, DAC, and PWM outputs. This reduces the need for additional peripherals and hardware to be incorporated which would only increase the complexity required and the power requirements needed. The microcontroller should also be reprogrammable to allow improved operation and upgrades over it intended lifespan and beyond.

The choice of microcontroller was difficult due to the large range and variations available. A 24-bit wide instruction microcontroller was favoured over the common 8-bit counterparts due to the extra inherent capabilities. The PIC chips are very common and easily available from the manufacturer Microchip, hence they are the preferred supplier. It was also noted that the manufacturer has incorporated lead-free product packaging for its products identifying an environmentally friendly product.

The idea of a redundant microcontroller was contemplated but overlooked due to the nature of this robots design. For safety critical or essential applications, additional microcontrollers could be incorporated to ensure operability in all circumstances. There are 20, 30 and 40 Million Instructions per Second (MIPS) versions with the high performance controller, PIC24H, chosen for its superior capability. This could be substituted for a cheaper, low performance controller in the future if commercial production of this robot was viable.

The added benefits of using 16-bit addressing IC's include the 0 to 65535 range as opposed to the 0 to 255 range from an 8-bit IC. This also provides a 2^{16} memory addressing range. Other considerations include de-rating of the bus to an approximate 50-80% of its rated raw maximum. This reduces power consumption and improves efficiency as well as lowers the operating temperature. Features that are favoured in a microcontroller include the following:

- Lower power function, energy saving features,
- Reliable,
- Reprogrammable,
- Onboard ADC,
- Useable PWM input/outputs,
- Low cost,
- Necessary number of Input / Output pins,
- Timers,
- Voltage comparator.

The PIC24HJ256GP206 microcontroller was the chosen option at 40 MIPS with low power consumption. It operates on a voltage between 3-3.6V and provides all the functionality required for the project including a total of 53 input/output pins. This includes 8 PWM outputs and an 18 channel analogue to digital converter. PWM will provide the output driver control for the motors whilst the ADC will provide sensor input digitalising. The IC package supplied was a 64 pin TQFP which initially proved difficult gaining a circuit board in the correct size scale but this was eventually overcome.

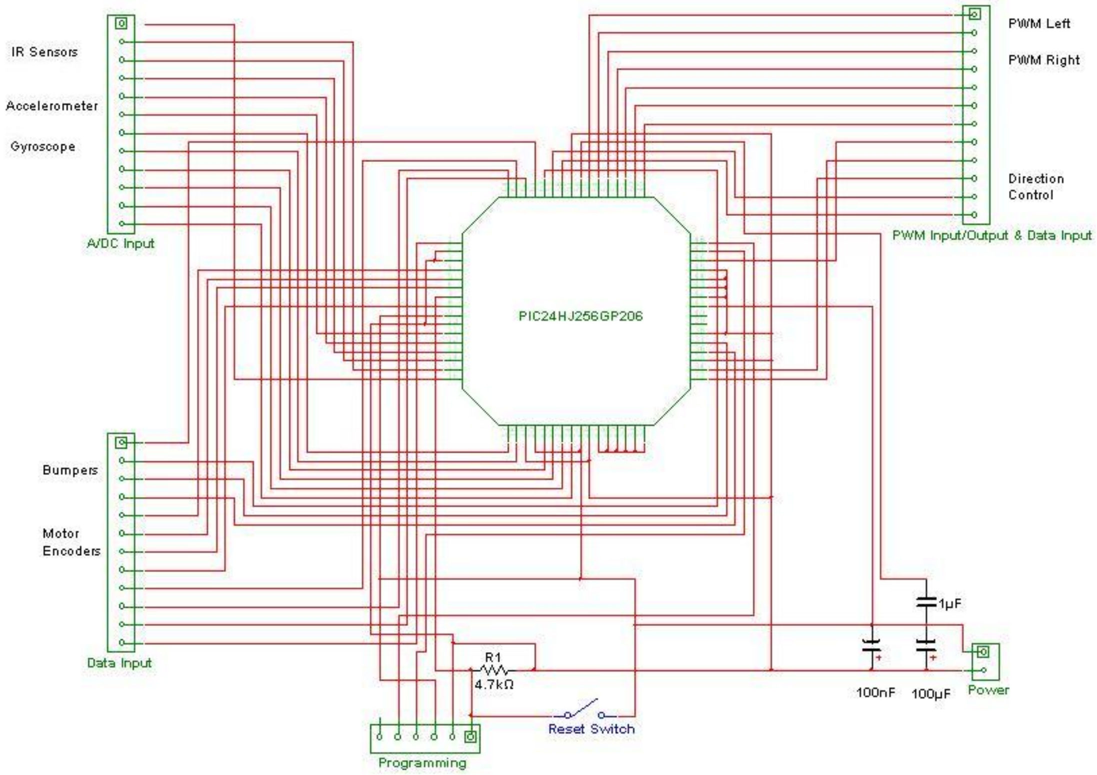

Figure 3.16 Microcontroller circuit

The controller contains 16 KB of RAM and 256 KB of programmable flash allowing the stability control algorithm and associated control programming to be easily stored and run. The chip can handle 24-bit wide instructions with the C programming completed within the software package 'MPLAB'. A debugger/programmer ICD 2.5 was sourced form an electronic supplier online for performing the microchip flash programming. Additional connectivity was added to the design to allow reprogramming whilst fitted inside the robot. The previous figure contains the developed microcontroller circuit.

3.4.7 Components / Circuits

Components form the building blocks of circuits which in turn provide the capability and reality of a machines operation. Components include resistors, capacitors, transistors, switches, sensors and IC's. They are typically assembled together on a PCB to form a robust, compact and localised circuit. Components are given tolerances in which they may vary from the prescribed rated value. They are also rated on there power carrying capability and temperature variation characteristics.

Ideally, circuits are designed to be maintenance free with materials and components that will maintain tolerance over significant time and operational circumstances. If a single component was to fail, significant damage could be imposed on the robot or those around it through the malfunctioned actions. This may also occur if the PCB track widths or wire gauges were not checked to ensure they are capable of maintaining the currents that will be developed whilst in operation.

Electrolytic capacitors pose the only significant maintenance consideration to this robot's circuitry as they are prone to leakage after they exceed there short lifespan. The components utilised within the robot designs will be those commercially available through local and online electronic stores. Higher reliability and tolerance items with wider temperature ranges could also be incorporated into the design if the robot's operational specifications were to be improved on. Additional considerations that were encompassed in the circuit designs included the use of pull up resistors, bypass capacitors and shorter leads to provide extra protection of signals and resistance to noise.

Circuits for voltage monitoring of the battery, motor control and the interaction computer could have been designed and created uniquely, but commercial available options proved more economical both financially and time wise. The major peripherals such as the web cameras and speaker system were purchase for similar reasons. Power distribution circuitry was specifically designed and created for application within the robots system. The main circuits utilised within the design can be summarised into the following PCB's:

- 12V power distribution,
- 3.3V & 5V power distribution,
- Voltage monitor,
- Microcontroller,
- H-bridge,
- Interaction computer,
- Accelerometer sensor board.

3.4.8 Stability Control Requirements

Stability refers the ability to maintain a firm upright position without inherent movement. Being balanced suggests that stability has been achieved as no further motion should occur. To achieve this, there must be some type of control or force in place that inhibits the unwanted motion. This control would need to counteract forces or influences in a reasonable timeframe or stability would be easily lost.

Stability control is the fundamental aim of this project. In terms of the robot, without stability control, it will merely fall over. The way in which the robot chassis will maintain stability revolves around the control instructions programmed within the microcontroller system. These will read inputs from the inertial devices (accelerometer, gyroscope, shaft encoders), bump detectors and IR sensors, then create a reactive output signal for controlling the motors. This will be implemented by a fuzzy control system as discussed previously.

A possible stability method that was overlooked during the project would be to incorporate a system that relies on a form of brakes attached to the wheels. This would allow the motor torque to be directly applied to the robot chassis when the brakes were engaged. The brakes could be disengaged and torque applied again through motion of the wheels. This option would have provided stability control with, theoretically, no horizontal displacement or acceleration thus reducing the complexity required in the fuzzy control system. Unfortunately, this idea was not conceived early enough in the project to allow investigation.

Stability is even more of a challenge when travelling up or down steep inclines. This will cause the gravitational force to move from the vertical plane and if the system cannot account for this, it will become unstable. Ideally, a value will be added to the measured angle of rotation to compensate for these forces. Determining the maximum allowable angle of tilt for the robot chassis required a trigonometric approach. As the motor will be mounted to the undercarriage, the extra height will add to the radius of the wheel and thus provide an adjacent length value of 125 mm. A hypotenuse length of 140 mm is then found. The following calculations were performed:

Hypotenuse Length L_H = 140mm
Adjacent Length L_A = 125mm

$$Cos\,\vartheta_{Cal} = \frac{L_A}{L_H} = \frac{125}{140}$$

$$\vartheta_{Cal} = Cos^{-1}\frac{125}{140} = 26.77°$$

$$\vartheta_{Max} + \vartheta_{Cal} = 90°$$

$$\vartheta_{Max} = 90° - 26.77° = 63.23°$$

Therefore, the maximum tilt before recovery is no longer possible occurs at an angle greater then 60^o or pi/3 rad.

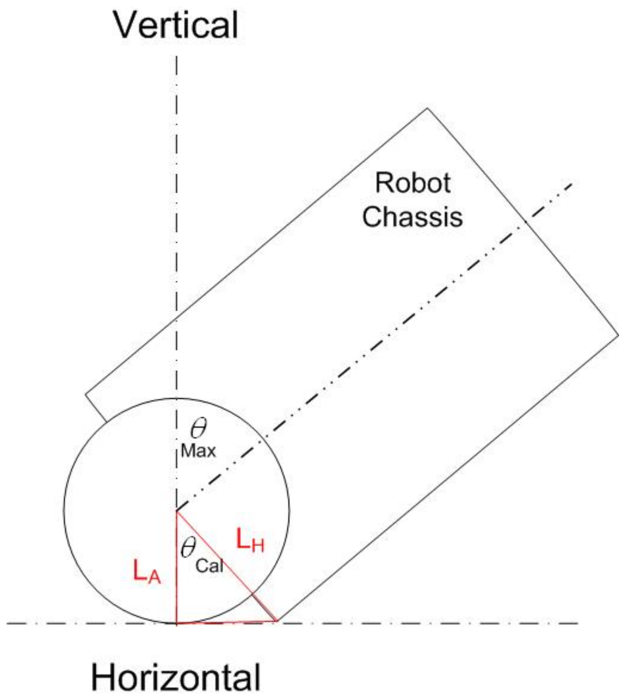

Figure 3.17 Maximum tilt angle variables

3.4.9 Power Source

A power source is essential for providing energy to a machine. Without energy, a machine would simply not operate. If an insufficient power source was implemented, a machine would not function correctly nor would it perform adequately. Power sources may be AC through electrical main supplies or generators. They may also be DC from devices such as batteries. Batteries are rated by voltage output and ampere hours (AH). AH refers to the current that they can supply per hour.

Batteries will be an essential component in allowing the robot to maintain autonomous operation. Otherwise a direct connection between the robot and a power source would need to be maintained at all times. Ideally, the best practise is to minimise the total power consumption during this design phase so a smaller power system may be implemented. This is an important factor as power systems are typically the heaviest element of a robot or device.

The design should aim for low power devices such as the microcontrollers and other components that have already been chosen. 6V, 12V and 24V supplies are the most common battery systems available. The robots system voltage was chosen to be 12V because of the better options available when choosing the motors and various circuits to implement. It also permits the creation of simple DC-DC conversion circuits encompassing 3.3V and 5V within the robots system. The battery proved to be a significant expenditure of this projects budget.

PWM was chosen for the motor control in an attempt to reduce the total current (power) supply required in maintaining stability. The idea of a redundant power supply system was considered unnecessary in this application. It could be an additional option if it was to be utilised in critical applications. Ideally the chosen battery will also have a lower internal resistance which will allow the maximum power delivery to the motors through higher current capability. The types of batteries available include the following:

- **Nickel-Metal Hydride (NiMH),**
  NiMH is the best rechargeable battery option available as it may be recharged hundred of times without memory effect. It is also better for high current applications such as robotics and has a low internal resistance.
- **Nickel-Cadmium (NiCad),**
  NiCad batteries are prone to memory effect and are highly toxic.
- **Alkaline,**
  Zinc batteries provide a better performance and are low current charging.
- **Lead-Acid,**
  Lead-acid are a long lasting battery but they are typically heavier and larger compared to its counterparts. There is also a potential problem that gas may build up in poorly circulated areas during charge up.
- **Zinc based,**
  Zinc batteries are prone to draining at fast rates.

Various configurations of battery connections (McComb & Predko 2006) were considered including a series connection between two batteries that would allow the voltage to be doubled. This could be beneficial for higher motor voltage usage. A parallel connection of the batteries would allow the current to be doubled. The disadvantage with these approaches is that both batteries would require replacing if one begins to exhibit poor performance. If a series connection between two batteries was used as well as a single connection from one of them, they would discharge at different rates proving a potential disadvantage.

A solar panel for recharging whilst the robot was active could have provided a boost to the battery. Although feasible, it was not pursued due to the extra costs involved. Isolation of power supply cables within the robot could have been achieved with opto-isolators. They are available in IC form and may have been beneficial for use with the sensor power connections.

Filtering out noise on the power source with bypass capacitors was an essential addition for preserving stable operation. These were installed on the input voltage connector for each of the PCB's. A filter capacitor was installed, close to the battery terminals as another interference preventative measure. Unfortunately, noise is typically introduced by motors and other analogue devices which form the majority of the robots components.

A voltage monitoring circuit was initially designed but became uneconomical both financially and time wise when a similar 12V LM3914 LED display based voltage meter was discovered from an electronics supplier. Although not implemented, one of the LED's could provide a signal for use as an interrupt. This would effectively warn the control system of imminent power failure which inturn would call a recharge sub-routine, redirecting the robot back to the user for recharging. The implemented voltage monitoring can be initiated at any time via a switch. The switch ensures that the voltage meter can be isolated from the battery, preventing it from being inadvertently drained.

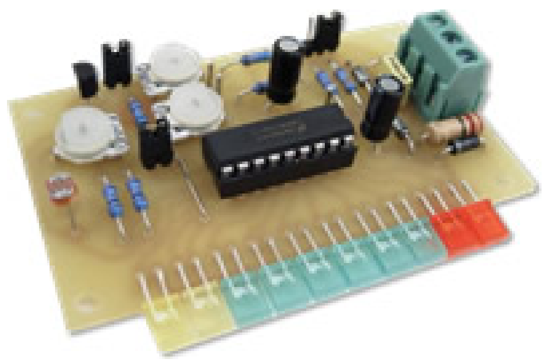

Figure 3.18 Voltage monitoring circuit (Dick Smith Electronics)

The power requirements of the robot were initially estimated at approximately 20A. Protecting the robots components from damage by over voltage or voltage spikes during the recharge process requires that they are disconnected from the battery. This has been incorporated by selecting a DC female connector that breaks the connection upon charger connection. A physical isolation switch before the 12V distribution board was also implemented. Overcharge protection circuitry could have been developed but manually checking the voltage meter was deemed acceptable for the project.

The simpler approach of utilising a single battery and implementing DC-DC conversions for lower voltages was chosen. The battery selected to meet these requirements was the lead acid type, rated at 12V 26Ah. Choosing the type of battery proved difficult due to the inherent large weight but economically it was the best choice for the current required. The following current allowances we determined with a peak current of 21A expected:

- Interaction computer system 10 Amps,
- 3.3V & 5V distribution circuitry, 5 Amps,
- Motor circuit, 6 Amps.

This battery is easily capable of supplying the motor stall currents as well as maintaining maximum current draw of the circuits and interaction computer. Voltage regulators were designed into the distribution boards. A LM 7812 CT regulator in conjunction with high power transistors regulates the 12V supply. A LM 7805 CT and LM317 regulator supports the 5V and 3.3V supplies respectively. A diode was also used on the input of 12V board to protect from accidental reversal of the polarities. Blade fuses were utilised for circuit protect of the 12V circuits. The following figures contain the voltage distribution circuit boards.

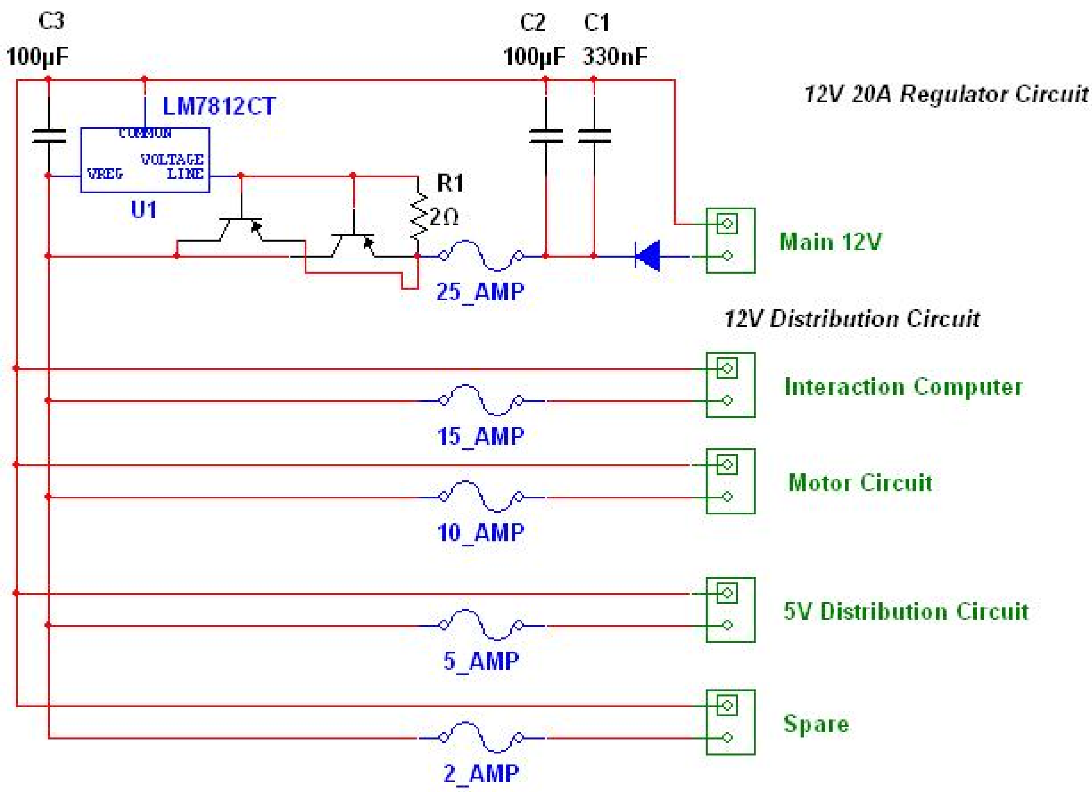

Figure 3.19 12V distribution circuit

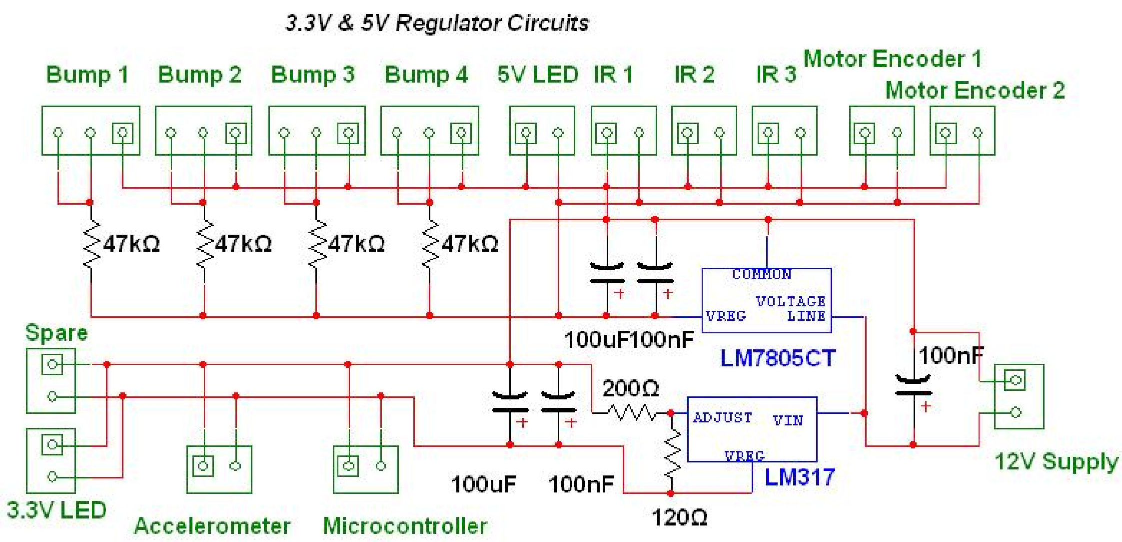

Figure 3.20 Low voltage distribution circuit

3.4.10 Locomotion Control

Locomotion refers to the ability to move. This can be achieved numerous ways through the use of wheels, legs or any other means. Control of locomotion implies that the direction, speed and distance the object travels can be determined and directed by either the object itself or an operator. This can be implemented through an algorithm or control system that could incorporate various components such as visual based detection systems, proximity detection sensors or even GPS devices.

Time restraints have restricted the achievement of this section but the goal was to establish a wireless network connection via a USB WiFi device attached to the interaction computer. Microsoft Robotics Developer Studio would have provided the tools to control the robot through a simple visual basic program utilising internet explorer. The web cameras would be available for providing a visual display to the operator at the remote location.

3.4.11 Communications

Communications allows the operator or maintainer to interact with the robot through either a direct connection or wireless link. This would allow the robots actions such as locomotion to be controlled. It would also allow maintenance activities, monitoring of the systems and reprogramming of the microcontroller to be completed remotely. Allowing remote control through a wireless connection has the benefits of operating in areas or regions considered uninhabitable for human beings. This also allows the robot to be operated or maintained by skilled users in a completely different geographic location.

A WiFi wireless network connection has been incorporated within the interaction system which allows the robot to maintain a communication link with its operator or fellow robots. Interaction with people can also be achieved by streaming the video imagery captured by the robots cameras and the audio from the microphones back to the operator. The speaker system is capable of projecting the user’s voice at the robots location as well as the facility to add a 7in LCD screen to the upper layer of the robot for displaying the operator’s video image.

3.4.12 System Integration

System integration refers to ensuring the components and sections of the robot, each work together to ensure a smooth operation and performance of its tasks. Integrating each of the systems ensures signals and responses between each is interpreted and responded to as expected. Size of cabling between systems, layers and components including its current capacity and EMI susceptibility will impact on the successful transmission and reception of signals.

The idea of either a single control or distributed control for the robots functions was considered. ‘Distributed’ means that each task/component has its own microcontroller for managing its function. The advantages are that this approach is more reliable and less complex although it may cost more in hardware to implement. Centralised control refers to a single control system.

The microcontroller chosen to provide the stability control function provides the overall system integration although the interaction computer maintains the interaction tasks. The functionality that the microcontroller brings, include standard digital data, analogue and PWM input/output signal integration. The following overall system block diagram shows four main components of that will make up the complete system.

The interaction system is integrated by the PC based interactive computer which will provide the locomotion, WiFi connection, reprogramming and interaction functionality in the projects mature form. But for this project, the integration computer does not integrate with the microcontroller therefore each remains unique at this stage. The upper layers of the robot form the interaction computer whist the sensors and microcontroller of the lower half form the integrated microcontroller unit.

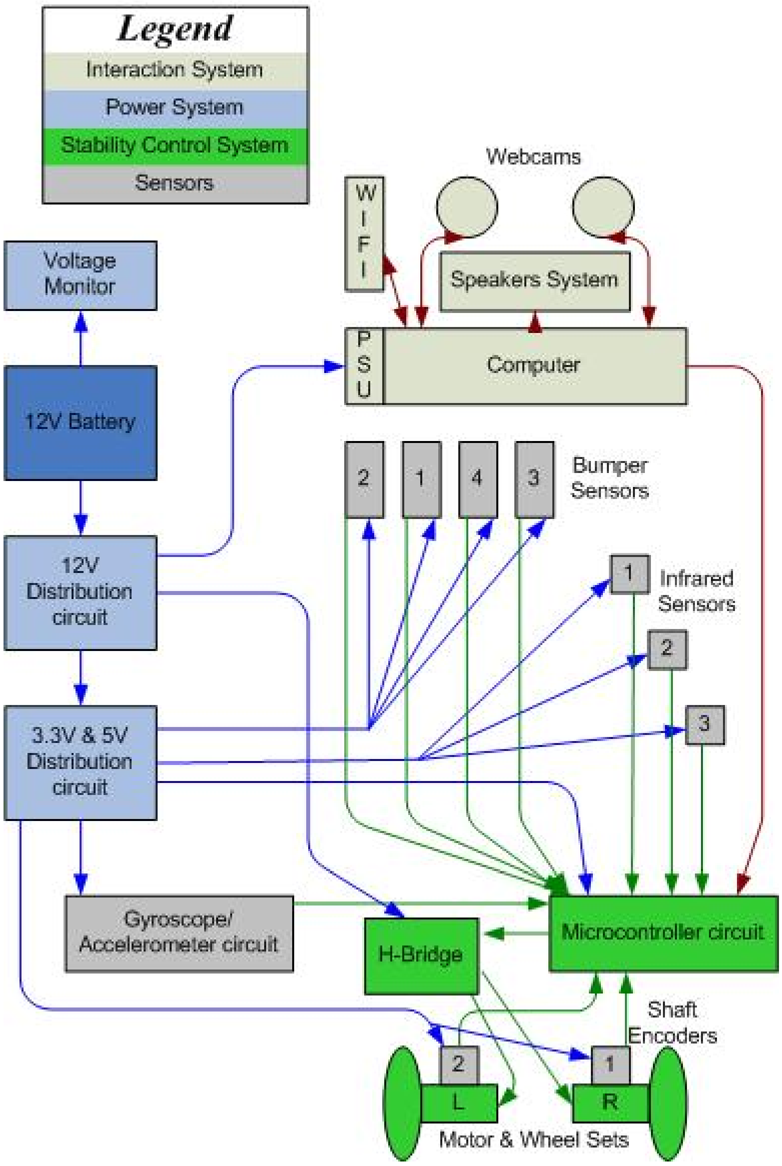

Figure 3.21 Overall system interaction block diagram

Other considerations that are necessary when incorporating these components together include cable lengths as power losses and reduced signals strength may occur. Physical separation of cabling was used to restrict interference between system components. This was achieved by spacing four separate cable-through holes on each layer of the robot chassis. The front two were used to pass data cabling whilst the rear two contained analogue and power cables. The wiring utilised within the robot design for the following circuits were as follows:

- Battery connectivity, 14 AWG,
- 12 V distribution, 18 AWG,
- Motor driver circuits, 18 AWG,
- 3.3V & 5V distribution, 30 AWG,
- Data carrying cables, 30 AWG.

Control and functionality

The core control and functionality facility of the robot is contained on the rear panel of the upper layer. There are two reset buttons installed for the microcontroller unit and interaction computer. Two LED displays are provided for voltage indication of the low voltage supplies while the WiFi antenna contains an LED to indicate when the device is active.

A robot on/off switch was incorporated to provide an isolation point from the battery as well as a separate switch for isolation of the voltage meter. A recharge connector for charging onboard battery has been fitted for convenience thus forming a central location for all the frequent operation tasks that the operator may be required to perform.

3.4.13 Safety Features

Safety features are an essential element of any design. They assure the safety of people involved with the machine, the machine itself and the world around it. Reducing the risk to people should always be the first objective in all designs. If potential hazards can be identified during the design and testing phases, then safety features targeting those particular hazards can be developed to reduce the risks.

The main potential hazards created by a two wheeled balancing robot are those associated with it falling. Damage to the robot itself may expose lead-acid from the battery or create sharp surfaces which may cause cuts or abrasions to people. To reduce the lead-acid risk, the battery has been securely fastened within the centre of the power layer to create a physical separation from the outside world. As an additional feature, the battery is covered with sheets of rubber before being encased in a protective metal shield. The rubber improves the already snug fit and allows absorption of any potential impact forces.

The physical weight of the robot itself has been kept to a minimum to reduce the impact force and applied pressure upon contact. It may become possible to sustain an electrical shock from the power circuits if they become exposed during a fall. An outer skin can easily be applied as a future addition to present a physical barrier of protection. Electrical shock hazards remain a low risk due to the voltages used in the design so residual current detectors are not required.

Other hazards may be induced from crashing, bumping or even accidental ramming of objects and people. Due to the robots intended use, redundant stability control and power supplies have not been incorporated. Ideally, deployable skids could have been developed to maintain an upright position until the microcontroller is capable of maintaining stability. A form of magnetic actuation could manoeuvre the skids in place. This would be beneficial in the event of a power failure or system lock up. This would require consideration of actuated angles, battery voltage levels, etc so has not been employed yet.

Incorporation of a cushion belt around the horizontal centre of the robot chassis would offer buffer protection to itself and its surroundings. This would utilise bumper contact switches underneath to detect contact. Four contact sensors have been included around the robots waist but not the cushion itself. This provides a safety feature whereby it could either back away once contact has been made or completely shutdown removing any applied force through its locomotion.

It is intended that the IR sensors will maintain adequate distance hence the contact switches are only fitted for safety. Smooth design and rounded layers should allow the robot to easily clear obstacles near it and reduce the risk of injury to others.

3.4.14 EMR / EMI

Electromagnetic radiation (EMR) and Electromagnetic interference (EMI) are representational of noise introduced by radio frequency or magnetic sources. They inhibit signal detection and introduce noise which may corrupt control signals and distort communication signals. EMR relates to the emission of electromagnetic radiation by the device whilst EMI relates to its susceptibility to spurious emissions. Ideally a device will exhibit low EMR and high EMI.

The most well-known sources of interference to a signal are the power supplies and common conductors which cause induced current fluctuations or cross coupling. The mechanisms available to counter these occurrences revolve around physical separation and/or barrier separation which are used to reduce the magnitude of the coupling. Both forms together provide physical isolation with the added shielding protection. Coupling occurs through magnetic, electric, conductive and electromagnetic means.

The separation technique in practical terms is the isolation between the analogue and digital circuits through a physical distance &/or by providing a barrier to restrict coupling thus minimising its effects. Grouping of similar signal types also adds the benefit of localising fault finding during the devices life span. Power supply distribution can be optimised for minimal interference by providing independent voltage supplies for each of the analogue and digital circuits.

Several separation techniques were utilised in the design of the two wheeled robot. This included a separate power supply for the inherently noisy gyroscope. This was provided by the H-bridge circuit which itself is high in current usage and connected to noisy motors. Physical separation of power, data and analogue signals was achieved by provision of four separate cable-through holes on each layer of the robot chassis. The front two holes were specified for data cables whilst the back two were analogue and power cables respectively. Common signal cabling was also grouped together.

Where possible, power cabling was run low on the layer whilst the data cabling was run high. When this could not be avoided, cables were run at 90^{o} angles to reduce there susceptibility. Filtering on both the power supplies and input of each PCB was made by the use of bypass capacitors. Separate PCB's were utilised for digital and analogue circuits where possible. Reducing the clock frequency and increasing the rise time of circuits was considered in an attempt to reduce the EMR factor of the robot. Due to the low DC voltages used within the robot, shielding or screening was not incorporated.

3.5 Design Concept "Oshy"

The final design concept offers a goal for completing the robot and making it a reality. It provides an overview of what the system will comprise of and provides an insight into how it will function as a whole. Interpreting the interoperability and integration of these components will allow problems or incompatibilities to be determined earlier in the manufacturing and assembly stage. This will reduce the potential losses that may be encountered in rework or redesign that may become necessary due to problems found late in the creation of the robot. The two wheeled robots design concept is summarised as follows:

1. Expected operating environments.
    a. Indoor & outdoor environments,
    b. Operated in dry conditions only,
    c. Be capable of terrain with 100mm crevices,
    d. Operate in temperatures below 50^{o}C.

2. Materials, Size and Weight.
    a. A riveted alloy metal chassis frame,
    b. Layers and brackets manufactured from plastic Perspex,
    c. Weight fitted higher in robot's chassis,
    d. Maximum weight of 17.5Kg.
    e. Maximum height of 0.8 m.

3. Control System
    a. Fuzzy control system,
    b. Inputs: Angle, angular velocity, displacement, velocity,
    c. Output: Motor control,
    d. Implement 7 value membership functions,
    e. Utilise centroid of area defuzzification,
    f. Implement a 427 Rule set,
    g. Recover before 60^{o} or pi/3 of tilt occurs.

4. Programming
    a. Main program with several sub-routines,
    b. Simulation of programming within MPLAB.

5. Locomotion components.
    a. 187mm diameter plastic wheels with solid rubber tyres, 40mm width,
    b. 12V DC Brush motors HG37D670WE12-052 with a 1:52 gear head,
    c. L298 H-Bridge Motor Driver,
    d. PWM driven control.

6. Sensors.
    a. ADXL311JE Accelerometer,
    b. XV-3500CB Gyroscope and board,
    c. Kalman Filter for sensor fusion of inertial sensors,
    d. Proximity detection by Sharp IR GP2D12 sensors,
    e. Bumper contact switches,
    f. Motor shaft encoders (pre-fitted on specified motor).

7. Microcontroller,
    a. Microchip 24-bit PIC24HJ256GP206.

8. Manufacture printed circuit boards.
    a. 12 V power distribution,
    b. 3.3V & 5V power distribution,
    c. Voltage monitor,
    d. Microcontroller,
    e. Accelerometer.

9. Power Source.
    a. 12V Lead-Acid Battery 26Ah,
    b. LM3914 Battery voltage monitor,
    c. Transformer 12V 1.5A for recharging,
    d. Current estimations,
        - Interaction computer system, 10A.
        - 5V distribution circuitry, 5A.
        - Motor circuit, 6A.

10. Interaction system.
    a. EPIA-V PIII 800MHz computer,
    b. WiFi network adapter,
    c. USB Web cameras,
    d. USB Speaker system,
    e. PCI USB port card.

11. System integration.
    a. Create a central control panel on the rear panel of the upper layer,
    b. Install a reset button for the microcontroller unit,
    c. Install a reset button for the interaction computer,
    d. LED displays for voltage indication of the voltage supplies,
    e. A main on/off switch for robot circuits,
    f. Voltage meter isolation switch,
    g. Add a recharge connector for charging the onboard battery.

12. Safety Features.
    a. Add collision detection bumper switches (see sensors),
    b. Incorporate a smooth and circular chassis design,
    c. Fit brackets and attach sensors within the robot chassis,
    d. Encase the battery inside robot chassis with rubber shock absorption.

13. EMR / EMI
    a. Provide physical separation of cable types via four spaced through layer holes,
    b. Cross cabling at 90^{o} angles to reduce there susceptance to interference,
    c. Add filtering and bypass capacitors to circuits,
    d. Separate boards where possible for analogue and digital components,
    e. Group common cabling together,
    f. Provide the gyroscope with a 5V supply from the H-bridge driver,
    g. Utilise the following wiring sizes for circuits,
        - Battery connectivity, 14 AWG,
        - 12 V distribution, 18 AWG,
        - Motor driver circuits, 18 AWG,
        - 5V & below distribution, 30 AWG,
        - Data carrying cables, 30 AWG.

3.6 Resources

The resources required during the project included computer hardware and specialist software for the construction, programming and report compilation. Materials for the construction of the robots chassis included metal cable tray lengths, plastic perspex and rivets. Microcontrollers and some inertial sensors were supplied by larger companies as samples whilst additional sensors for proximity and collision detection, cabling for data and power, DC motors and a battery were purchased specifically for the project.

The majority of the resources required during the project were predominantly available although voltage regulator components proved the most difficult to acquire locally. These include the following items:

1. Chassis construction materials,
    a. 2x 1800mm x 100mm Cable tray lengths, $50.
    b. 6x 300mm x 300mm Plastic Perspex, sheet $30.
    c. 100x 5mm Rivets, $5.

2. Locomotion,
   d. 2x DC motor 160RPM 12V, $100.
   e. 2x Plastic wheel 187mm, $25.
   f. 2x Shaft 13.001mm to 6mm, $15.
   g. 1x L298 H-Bridge Motor Driver, $30.

3. Power system,
   h. 1x 12V DC Battery 26Ah, $125.
   i. 1x Battery voltage meter, $15.
   j. 1x Transformer 12V 1.5A, $30.

4. Stability control and sensors,
   k. 4x Contact sensors, $20.
   l. 3x IR proximity sensors, $36.
   m. 1x Gyroscope, $20.
   n. 1x Accelerometer, sample.
   o. 1x Microcontroller PIC24HJ256GP206, sample.
   p. 1x ICD 2.5 Debugger/Programmer USB MPLAB, $65.
   q. Assorted circuit board components, materials and cabling, $200.

5. Interactive components,
   r. 1x EPIA-V PIII 800MHz embedded computer system, $100.
   s. 1x Pico DC-DC power supply, $40.
   t. 1x WiFi adapter, $30.
   u. 2x USB Web cameras, $30.
   v. 1x USB Speaker system, $25.
   w. 1x PCI USB port card, $10.

It is estimated that the total budget cost was approximately $1000. Items that were more economical to purchase included the pre-designed H-bridge motor driver and battery voltage meter. Although circuit versions for these had been derived, the additional time making the PCB's and equivalent costs in components did not provide any benefits within the project. This allowed more time to be focused on the control system and software components later in the project.

Matlab, MPLAB and Microsoft programs were required for performing the analysis, interpretation and representation of resultant data. Access to the necessary tools, equipment and hardware was achieved locally in Newcastle, NSW where the project was undertaken. This did not incur additional expenditure to that of the resources listed above.

3.7 Conclusion

A fully capable design concept has been developed for the two wheeled balancing robot which will provide the fundamental hardware necessary in achieving the goals outlined by the project. Manufacturing and assembly of the components, circuits and devices will provide the conversion from concept to actual physical robot. The software and programming direction has been provided by this chapter but subsequent development and simulation will be continued within the following.

Chapter 4

Manufacture and Assembly

The construction phase entails the manufacture of PCB's, chassis, drive shafts, etc which are then assembled to make the robot a realisation. This is the most rewarding chapter where the hard work of design and development will finally evolve into the actual physical robot. The software and programming concludes the robots production which will also encompass a simulation process aimed at improving the effectiveness and efficiency of the programming prior to its implementation. The maintenance considerations necessary in ensuring the robots ongoing serviceability are also explored.

4.1 Introduction

The design and development considerations of the previous chapter were thorough and investigated numerous aspects of the robots potential functionality, capability and adaptability. The outcome defined the required resources and devices necessary in manufacturing the systems, PCBs, brackets and chassis. Once the availability of the resources and components was confirmed, the microcontroller programming was able to commence. The construction was broken into the following stages:

1. Order and purchase the resources and commercial items required,
2. Manufacture and assemble the robot chassis,
3. Assembly and installation of the interaction system and upper layer peripherals,
4. Manufacture the necessary PCBs, brackets, motor mounts and battery mount,
5. Installation of IR sensors, bumper sensors and PCB's,
6. Installation of battery, motors and remaining components,
7. Manufacture and installation of all cabling.

During each stage of the construction, testing of the devices, PCB's and components was completed in an effort of reducing the need for fault finding further within the project. Unfortunately, a defective motor gear head and a delay in several components lead to this phase exceeding the anticipated completion time. It was then necessary to extend the timeline to October for this chapter thus reducing the schedule for data analysis. This also impacted on the ability to effectively test and implement the microcontroller programming.

4.2 Printed Circuit Boards

Printed Circuit Boards can be manufactured via several different processes. The most common method is to utilise a prototyping PCB which contains predrilled holes requiring the links to be soldered in place to form the branches. Another option is to purchase a blank board and etch the design into it. This would have allowed a more efficient PCB to be developed during the initial stages of the designing process.

Ideally, it would have been beneficial to design and develop the PCB’s with software programs such as “Electronic Workbench” or “Altium Designer” but the additional costs and time involved were the deciding factor to move to prototype boards. The ease of modifying and making changes to the layout provided a better option for the robot’s circuitry both economically and time wise.

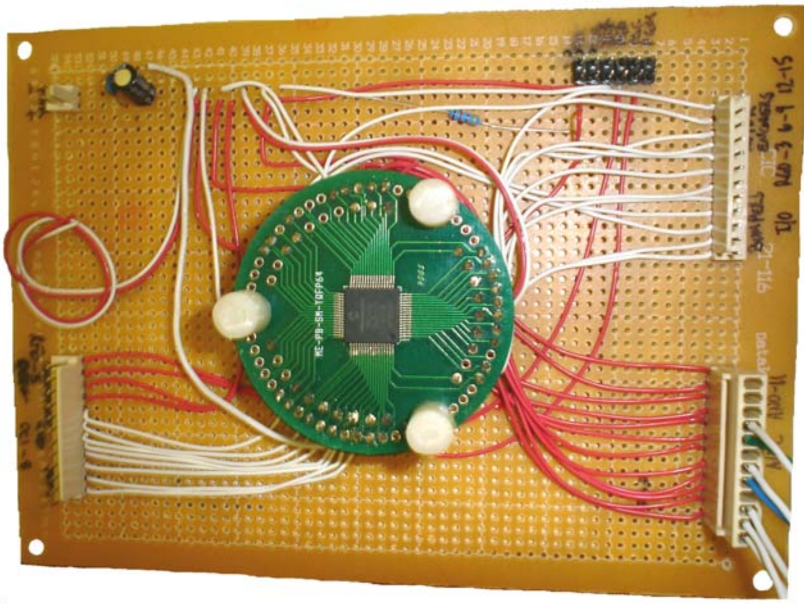

Figure 4.1 Microcontroller circuit PCB

The previous figure contains the manufactured microcontroller PCB. The aim was to allow ease of traceability and removability for fault finding or future modification procedures. The use of connectors and plugs throughout the PCB’s allowed ease of removal in all cases. The L298 bridge motor driver PCB and the battery voltage monitor PCB were pictured within chapter 3. The following figures contains the manufactured power distribution PCB’s.

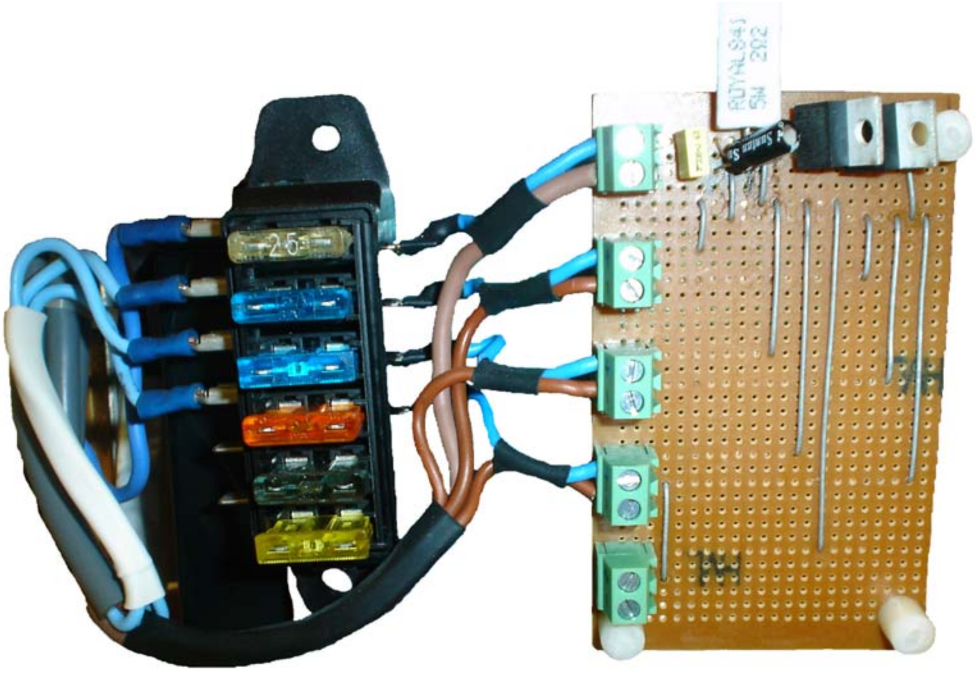

Figure 4.2 12V distribution PCB

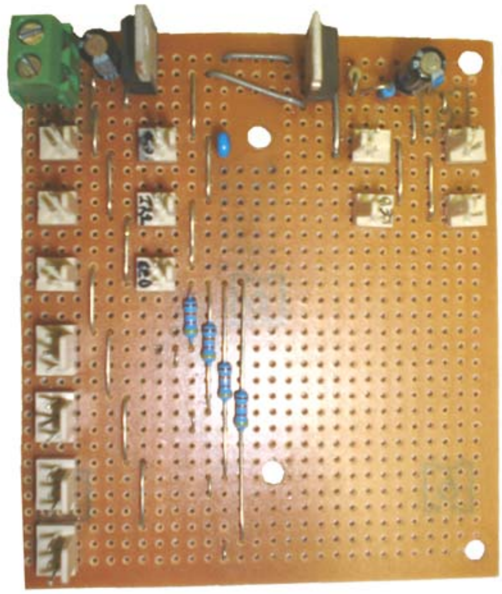

Figure 4.3 5V and 3.3V distribution PCB

The following figure contains the inertial sensors. The gyroscope PCB was received preassembled but the accelerometer board had to be designed to allow a simpler implementation within the robots system. These PCB's were then mounted on the underside of the spare layer.

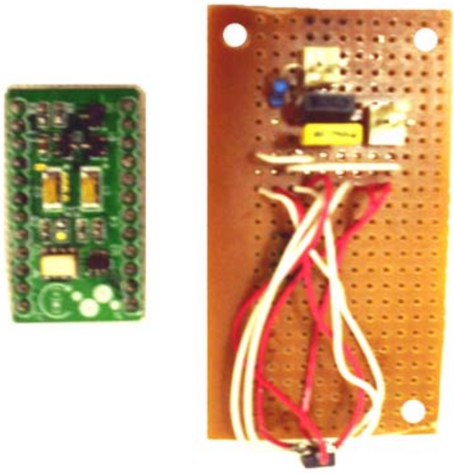

Figure 4.4 Gyroscope and accelerometer PCBs respectively

4.3 Wheel Base

The wheel base comprises of the motors and wheels layer, and the microcontroller layer. These contain the microcontroller PCB, H-bridge, motors and the low voltage distribution PCB. The layer above these remains vacant as the battery and power components of the robot were raised within the chassis when it was discovered that it would achieve better stability control. The following figure contains the wheel base layers.

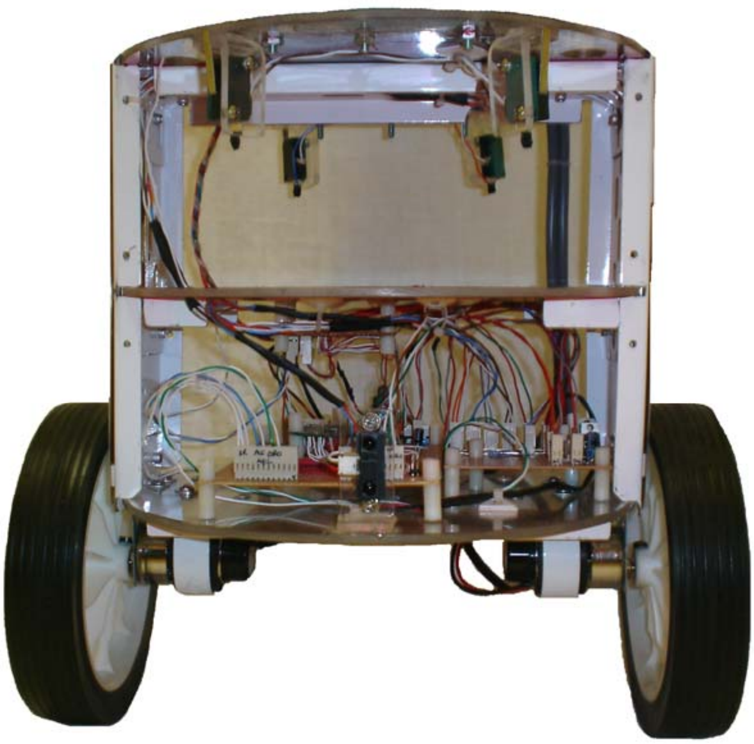

Figure 4.5 Wheel base layers

4.4 Power Source

The power source layer contains the 12V distribution PCB, fuse rack and battery. The 3.3V & 5V (low voltage) PCB was fitted within the wheel base layers to reduce the susceptibility to noise and losses. The power distribution PCB's were previously pictured within the PCB section. The bumper switches have been glued on the underside of the power layer to allow ease of battery removal in the future. These bumper switches could be moved directly onto the power layer at a later stage if they were fastened with bolts or similar thus fully freeing up the spare layer.

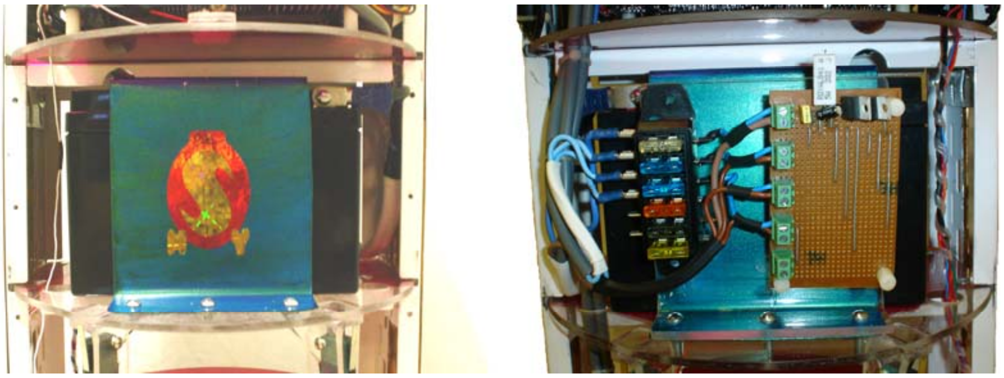

Figure 4.6 Power source layer front and rear views respectively

4.5 Interaction System and Upper layer

The interaction system comprises of the interaction PIII EPIA-V computer, 80GB HDD, webcams, speakers and WiFi network device. The hardware was implemented but due to time constraints, the planned visual basic programming for performing the microcontroller reprogramming, video and audio streaming, and locomotion control was not completed. The primary form of connectivity with these devices was USB via a PCI host adaptor. The following figure contains the interaction system.

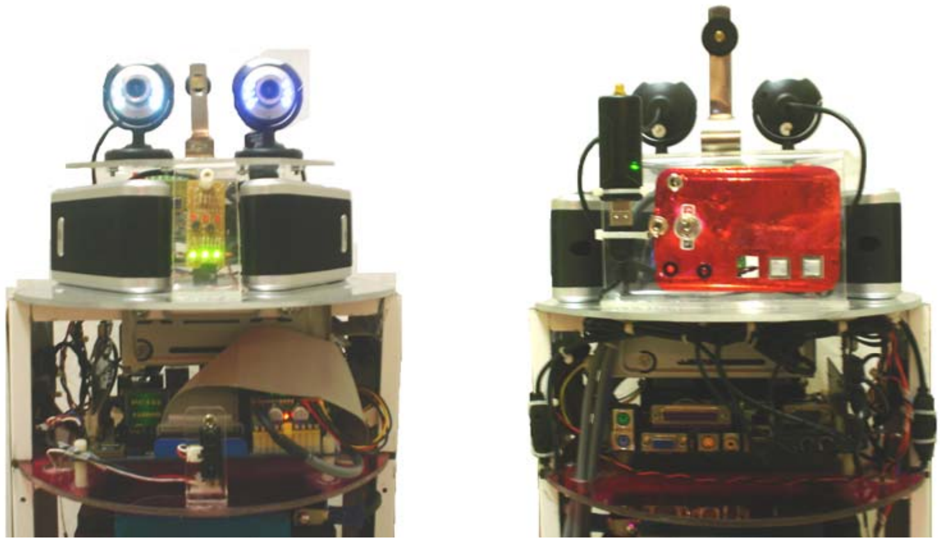

Figure 4.7 Interaction system and upper layer front and rear views respectively

4.6 Software and Programming

The software and programming phase was expected to complete both a visual basic software application that would be used to control the interaction system and a program for controlling the microcontroller system. As discussed previously, the interaction software was not created due to time limitations. The microcontroller program primarily focused on the controlling the wheel base thus the stability of the robot. This was achieved by reading various digital and analogue inputs and providing a motor control output signal.

The task of creating the program code was first attempted within Matlab where work begun on the fuzzy controller. The rule base and membership functions that were created are displayed within chapter 3. The original intent was to develop the full complement of the programming within Matlab to allow a complete simulation to be performed prior to implementation. An added benefit of this approach was that the Matlab code can be easily converted to C programming language through the Matlab application “Real Time Workshop”.

This attempt failed to achieve workable results within a reasonable timeframe at which point the decision to directly code the program into C language was made. This approach utilised the “MPLAB” Microchip software application designed for programming the PIC24HJ microcontroller that was previously chosen. The immediate problem with this approach was that the freely available student version would not be an efficient compiler. This had a potential to impact on the speed and efficiency of the fuzzy controller.

4..6.1 Simulation

Simulations are used to provide a cheap and simple method for evaluating the effectiveness and efficiency of a control system, circuit or device. They reduce the potentially high expenses and long time delays that are inherent with using and possibly damaging live systems. It also allows similar systems to be evaluated under the same test parameters and conditions thus permitting a fair and accurate comparison of the derived results. To achieve this, a model is derived either mathematically or theoretically from the real system that best represents the accuracy needed and the variables to be evaluated.

This model is then used to simulate the real system in response to the control system, circuit or device being assessed. Simulation was first attempted within Matlab but difficulties in simulating the robots movements, etc caused time to become a concern. The Matlab programming is contained within Appendix B. As MPLAB provided a simulation tool that reflects the microcontroller’s response and operation, it was decided that pursuing this option would stop the project from running overtime. A discussed in the following section, the microcontroller programming was not completed hence the simulation phase was not conducted.

4.6.2 Microchip Program

The main program derived for operation within the microcontroller is contained at Appendix C. The aim was to first validate the completed program by the simulation component followed by implementation testing within the completed robot. Ideally, the final program would have had the PID motor controllers, Kalman filter, locomotion desired sub-routine and proximity sensors incorporated but this was not achieved within the timeframe available. Difficulty was also experienced with initialising the timers, converters and data lines.

These were further compounded by the ICD 2.5 programming device problems encountered. Upon initial checks, the device was able to read and access the microcontroller chip but later during the testing phase of the motor sequences, it was discovered that the device no longer recognising the microcontroller. Additional testing and fault finding failed to restore the connection between the devices which cost significant time. Although implementation was not achieved, the derived program contains the fundamental structure and routines that, with supplementary research and testing, can be completed successfully.

4.7 System Integration and Overview

Integration of the sensors, PCB's and motors with the microcontroller was successful although it was disappointing that the interaction system was not completed. This would have allowed true system integration with all components contained within the robot. The components and devices chosen allowed simple connectivity with the microcontroller through its digital, analogue and PWM input/outputs.

Upon completion of the construction, operation was undertaken as expected with no rubbing, noise and vibration experienced. The completed robot is contained in the following figure.

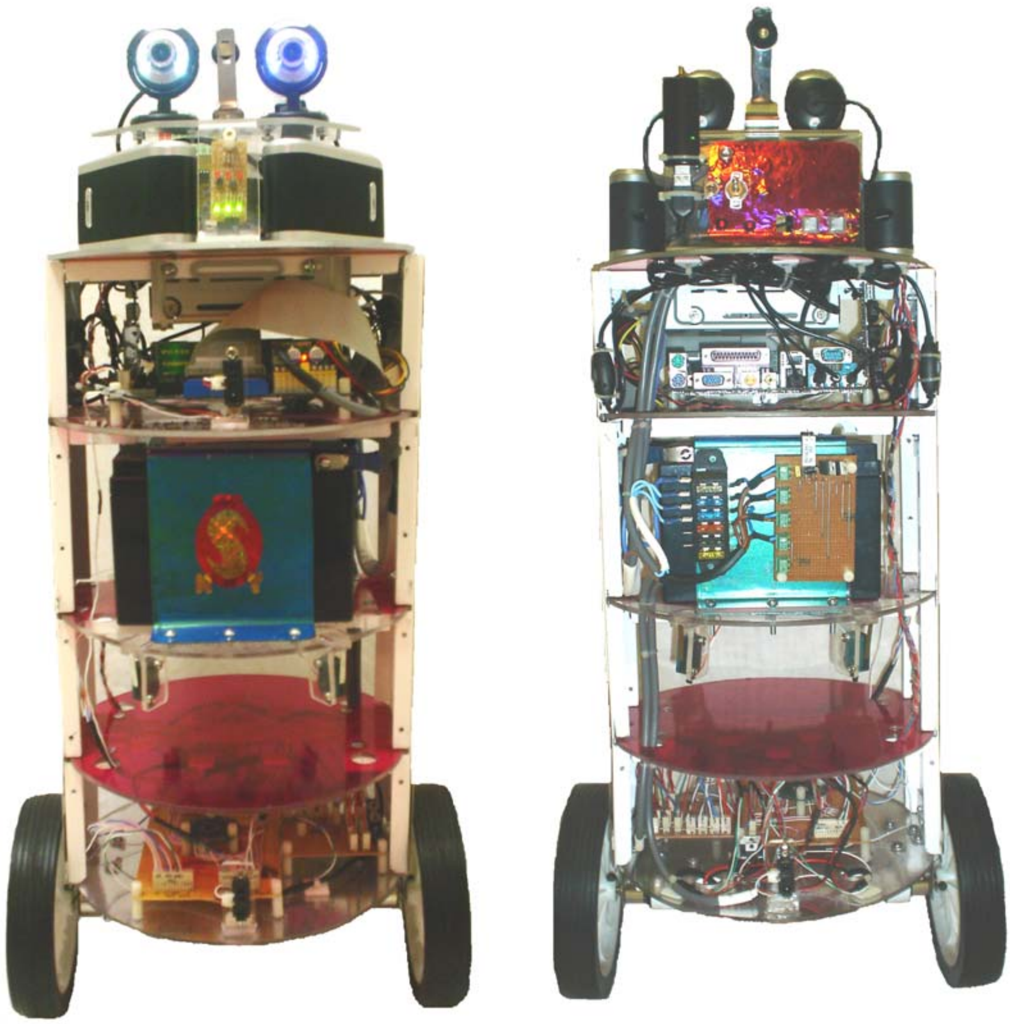

Figure 4.8 Manufactured and assembled robot front and rear views respectively

4.8 Future Maintenance Considerations

Maintenance considerations are essential for ensuring the availability of electronic or electrical equipment and devices. It provides a means of determining potential failures before they occur, allowing the maintenance actions to be completed at the operators convenience and not during essential operational time. These are often performed periodically or even before use. For the constructed robot, the following maintenance inspection activities are required at the necessary time intervals:

- **Before use,**
  - Check and ensure wheel shafts are clear and free of obstructions.
- **Daily,**
  - Clean and wipe IR sensors, webcams and all surfaces ensuring no dust build up is evident, especially on PCB's,
  - Ensure battery terminals are securely fastened,
  - Check and test bumper switch operation, repair if necessary.
  - Check and ensure the motors, battery and PCB's are securely fastened.

- **Weekly,**
    - Check the battery for leakage and corrosion, repair/replace as required,
    - Check electrolytic capacitors for leakage, repair/replace as required.
- **Monthly,**
    - Test and check motor and gear head operation, replace if necessary.
- **Six monthly,**
    - Test and check battery overall operation, replace if necessary.

Additional maintenance activities may be implemented depending on the robots operating environments. There may be unforeseen problems that will become evident with further exposure to high levels of vibration throughout the robots chassis. If the robot was to become commercially available, then further action such as development of replacement procedures encompassing motor, PCB and other component repairs may be developed but this is not required for this project.

4.9 Conclusion

A successful design and development stage cemented the foundations for a successful manufacture and assembly of the actual two wheeled balancing robot. The implementation of the robots hardware was easily achieved providing the system integration and functionality. The majority of the difficulties encountered throughout the project were associated with the software and programming phase of this stage.

Software and hardware failure impacted upon completion of the interaction system whilst programming alone inhibited the projects completion. An analysis and evaluation of the systems performance will now be undertaken to ascertain the resultant limitations and capabilities of the programming and to provide a means of overcoming the discovered operational shortfalls.

Chapter 5

Data Analysis

The analysis phase provides an opportunity to assess and evaluate the robots effectiveness and efficiency in maintaining stability and providing locomotion. This allows a comparison to be undertaken between the actual system performances and the anticipated project objectives. The opportunity to calibrate and perform additional fine tuning of the design is also explored with the aim of achieving the best result possible in mind.

5.1 Introduction

Data assessment and evaluation was to be performed by measuring a combination of input sensors. These include the gyroscopes rate of rotation, the accelerometers rate of acceleration, the shaft encoders counter values and the motor output drive values. These and the response times would then be evaluated to determine recoverability, system response and ultimately efficiency. The original aim was to have Matlab form the basis of the complete system simulation but this was discarded due to coding problems and time restraints.

The resultant data would provide results for various levels of tilt and sudden bursts of acceleration experienced by the robot chassis. These could then be compared with each set of derived response times and values for the varying applied motor power. The expectations of the initial results were that an effective stability platform would be easily achieved. It was anticipated that effective and efficient stability would be accomplished before completion of the simulation stage as the programming can be altered to reduce response time and improve output reaction.

5.2 Calibration and Tuning

Calibration and tuning are essential methods for establishing and setting the baseline of devices, sensors and systems. Misalignment from poor or incorrect calibration may cause a device or system to provide inadequate responses and detection possibly leading to further failure or significant damage. The aim of this topic was to identify potential benefits that may be acquired through calibration of the two wheeled balancing robots components. The items to be calibrated and the level to perform these too must first be determined. The following items are calibration candidates within the project:

- Drive motors,
- Sensors and encoders,
- PID controller (gains),
- Fuzzy controller membership functions.

Calibrating and testing the motors, encoders and sensors before operation will ensure operability and that the best configuration of the control system is achieved. This will account for possible ‘out of tolerances’ and other unforeseen losses that may be inherent. If the motors are not sequential in operation, then inadvertent turning of the robot chassis will occur. Incorrect detection by the encoders and other sensors would also impact on the robots sense of positioning. This would cause delays in the correct positioning and further increase the volume of calculations the program must undertake.

Tuning the controllers is an essential step in ensuring the stability control system is responsive and timely. The fuzzy controller is considered quite difficult to tune due to the increased complexity of utilising numerous membership functions and inputs. Incorporating PID controllers for the enhancement of the motor driver signals also requires that the proportional, integral and derivative values are optimised for peak motor performance. The programming difficulties encountered have restricted the implementation of these intended steps and procedures.

5.3 Stability Analysis

Performing an analysis requires the particular performance to be measurable or reasonably estimated. Accurately measuring the actual stability of the robot chassis is difficult but it can be achieved by logging the values recorded by the robots sensors. Another approach would be to take video footage of the robot with markings and measurement indications set in the background to allow determination of distances, angles and compare these to the response times.

By assessing the potential impact on the analysis data from mechanical, electrical and programming properties, potential improvements and redesign aspects can be derived. These would contribute to increasing the overall stability of the system as well as its efficiency. The mechanical properties of the robots design included physical layout, shape, weight, weight distribution, and height. The electrical properties encompassed capability and installed hardware whilst the programming properties include performance, efficiency and ability.

The particular attributes that were to be investigated from the stability analysis activities include the response to various rates of acceleration both forward and backward, the affect of induced force and tipping, and the addition or removal of mass onto the robot chassis. The preference was to assess the recorded inertial sensors over time with respect to motor output signals but an actual analysis was not conducted due to the microcontroller’s program difficulties.

5.4 Locomotion Analysis

Locomotion analysis is secondary to the primary requirement of achieving stability. Unfortunately, the interaction system was designed to provide the locomotion control via the WiFi connection and as the system was not completed, an analysis of locomotion cannot be achieved. The physical, hardware, software and program would have been fully explored potentially requiring much more time then would have been necessary for the stability analysis previously.

Completing this stage would have been achieved by measuring the response times, output drive signals, and assessing the robots ability to maintain stability for varying sensor inputs and terrains. This would assist with determining the minimum and maximum interval rates that the microcontroller system could process data, feedback and send out the compensating motor control signals. This would have allowed further improvements to be made to the locomotion control and its system components.

5.5 System Performance Overall

The overall system performance is difficult to determine due to the program difficulties encountered but it is highly likely that a successful implementation of a two wheeled balancing robot would have been achieved. Additional proximity sensors, an inclinometer and incorporation of a much more capable interaction system would allow the robot to conduct fully autonomous control and interaction with the world around it.

The microcontroller is a 40 MIPS high performance controller thus providing the maximum opportunity for the programming to perform. The programming of the microcontroller provides the only known area of potential improvement from this design approach and implementation; therefore, system performance would have been a reflection on the efficiency, effectiveness and robustness of the final program.

5.6 Conclusion

Performing calibration of devices and components prior to tuning of systems allows a reasonable analysis of the robots performance. Analysing each of the performance objectives allows further improvements to be made in the robots operation and responsiveness. Evaluating the overall performance of the completed robot also provides a benchmark to work from, and provides potential for redesign and enhancement work.

Unfortunately, the project was unable to implement the ideas and approaches originally intended for providing a thorough analysis and evaluation due to the programming. Ideally, this phase would have also seen the developed robot utilised as a test platform for several types of linear control system. This would have provided a direct comparison between the successfulness of numerous forms of controllers.

Chapter 6

Conclusion and Recommendations

Conclusions and recommendations are necessary to provide comment on each aspect undertaken as part of the project. The analysis and evaluation stage provided the specific information that can be used to ascertain the successfulness of several key areas of performance. This is then expanded into several recommendations for improvements, addition of extra capabilities and future areas of investigation. This also provides the opportunity to share an awareness of the problems encountered that would complement the future work recommendations.

6.1 Introduction

The data analysis performed within the previous chapter was initiated to provide a reasonable assessment and evaluation of the robots ability to meet its design objectives. The review of the analysis performed suggested that the simulation process could have been competed more successfully if more time was available to develop a complete system model within Matlab. From the observations and discussions previously presented, this chapter will provide recommendations for future work, improvements and additional areas of investigation.

6.2 Discussion

Several key aspects are discussed in the following sub-sections. These include the difficulties experienced, advantages and disadvantages of the design phase, the results versus the expectations of the robots performance.

6.2.1 Difficulties Experienced

The difficulties experienced throughout the project include theory, hardware, software and programming related problems. The main theoretical difficulty encountered was associated with understanding the operation and mathematics behind Kalman filtering techniques. This was necessary for implementing sensor fusion of the inertial sensors within the programming. This aspect would have required substantial time to rectify hence the simpler approach of following the programming provided by Simon (2001) was pursued.

Gaining a thorough understanding of fuzzy control systems and their implementation appeared to take a substantial amount of time early within the project. The tuning and simulating aspects could have been further investigated possibly leading to the development of an adaptive (self-tuning) fuzzy controller. This would have been a substantial improvement over the fuzzy controller implemented as part of the project.

The 1:52 gear head supplied with the right drive motor stripped the teeth off a cog within a very short period of operation. Unfortunately, the item was supplied from the USA which delayed the assembly stage for sometime. It is unclear at this time if the gear head and motor combination is suitable for the loads and weight of the derived robot. Difficulty was also experienced in acquiring critical components such as the 7805 and 7812 voltage regulators. These remained on back order for six weeks before they were eventually received.

The interaction system encountered several difficulties early in the project. On installation, the interaction computer was functioning and it appeared that the visual basic software development would be relatively straight forward. It was then discovered that an incompatibility between the computer and PCI host adaptor was causing the computer system to crash. This was soon followed by a hard drive failure causing the stage to exceed the allocated timeframe. It was at this time the interaction system was deemed non-essential and no longer pursued.

Programming of the microcontroller proved to be the most difficult challenge of the entire project. Although learning C programming was not overly difficult, completing the hardware initialisation, fuzzy controller, Kalman filter and other sub-routines within the programming caused excessive time consumption for the project. This was further compounded as the simulation phase could not be adequately completed. It is unlikely that the project would have been as completed on time if the simulation modelling was pursued fully even though a robust program could have been developed.

6.2.2 Design Advantages and Disadvantages

The shape, size and layout of the robot proved the most advantageous aspects of the robots design. The cylinder shape proved ideal for manoeuvring on the spot. This was evident during assembly and performance maintenance activities on the robot as turning it on the spot allowed easy access from the one location. The child like size made the robot easy to move, handle and to access each of the layers. The platform layout allowed grouping of systems and PCB's without overcrowding thus sampling the maintenance and fault finding procedures.

There were no disadvantages discovered with the design of the robot. The battery posed a potential disadvantage from its inherent weight of 7 Kg. This accounts for a significant portion of the total weight as well as the space required to store it within the robots chassis. The impact from the battery may have been reduced by selecting a smaller physical size and rated battery but then the capability and operability would be adversely affected. There is also a potential that the motor and gear head combination may have been underrated for the specific task.

6.2.3 System Performance and Results

The actual performance of the hardware aspects of the robot cannot be fully guaranteed without completion of the microcontrollers programming. It is anticipated that the manufactured robot would have been fully capable in achieving stability, locomotion and interaction streaming. No actual performance results are available at this time.

6.2.4 Results versus Expectations

At the beginning of the project, the expectations were quite substantial. The goal was to develop a robot that was so robust that it could recover from significant and deliberately induced tilt. As problems and difficulties started to become evident in the programming, the expectations reduced to simply providing a robot that could maintain stability. Incompletion of the program has proved devastating as hence the results are not reflective of the expectations.

6.3 Conclusion

The project as a whole was very successful in deriving a two wheeled balancing robot with all the considerations necessary tin ensuring it could meet the required capabilities and objectives. The software and programming side was disappointing with the majority of the difficulties encountered in these areas. This could be overcome with time but unfortunately, time was not plentiful. The following recommendations provide direction for future work and suggested areas of further investigation.

6.4 Recommendations

It is recommended that this approach be undertaken again but with an experience programming background. The difficulties encountered could have been easily overcome if significant time was available to further develop the programming necessary to perform the significant functions of Kalman filter, controllers and other sub-routines. The following sub-sections contain the suggested improvements, additions and possible areas for future investigation.

6.4.1 Improvements and Alterations

Improvements to the current version of the robot can easily be achieved through the microcontrollers programming. This would provide the fundamental benefits faster control responses, effective and efficient computations and improved overall system performance. Completing the software implementation of the interaction system would allow the operator to have video and audio streaming capabilities, locomotion control and microcontroller reprogramming capability. Although the hardware would not require any significant change, the completion of visual basic programs would complete the process.

6.4.2 Additional Features and Capabilities

Additional feature or capabilities that may improve the effectiveness, efficiency or operability of the robot include the addition of arms. This would allow the robot to interact with its surrounds thus providing a larger impact on the operability it may now achieve. This could be further extended to include grippers at the end of each arm that would be capable of effectively lifting large masses or obscure shapes. The number of arms fitted would increase the complexity of the control systems in place but the enhanced capability may be beneficial.

Although not fully explorer, the webcam vision could be expanded to incorporate stereo vision. This would allow the robot to autonomously navigate, identify and avoid obstacles through the added vision capability. The voltage monitoring circuit could provide a signal to the microcontroller that would indicate that a low voltage measurement has been established. This would call upon a ‘return to charge station’ procedure which would ensure the robot returns safely before power is lost to its systems.

To better transverse uneven terrain such as stairs, the wheels could be enhanced to allow an upward and/or downward extension to its height. This could either apply a force in the direction of travel onto the obstacle to push itself up or it could raise the wheel over the obstacle itself. This would allow the robot to travel up stairs or over obstacles whilst maintaining balance.

An outer skin may be incorporated for cosmetic and well as shielding effect for the robot. A skin that allows the IR sensors to easily see through would be ideal, allowing a smooth covering of the outer side. This would also be protective against dust, water and other forms of intrusion into the sensitive circuits within. A cushion belt could be added to the bumper switch arrangement to allow a softer collision in the event one may occur.

6.4.3 Future Areas of Investigation

The following recommended areas of investigation involve different approaches to the stability control problem. The potential benefits from these approaches include the reduction of energy consumption, reduced horizontal movement and protection from falling over. The first area focuses on a rotating robot chassis that is capable of a potential 180^o horizontal rotation on its vertical axis. The weight could be distributed in such a way that any chassis movement would redistribute the weight thus perform a form of stability control. The benefit of allowing the chassis to rotate would remove the need for the wheels to move forward or backward along the horizontal plane.

A form of braking system could be developed that in effect, locks the wheels to the ground thus maintaining the robots horizontal positioning. The motor force is then used to move the robot chassis and not the wheels to maintain stability. It would appear as a forward or backward swaying motion from the side of the robot. This would reduce the complexity of the stability control system that is currently in place as both the horizontal and vertical axes would both be easily controlled.

Incorporating a form of gravity powered counter weight system within the chassis would alleviate the need for excessive power consumption that a motor requires. This would have a butter effect by further reducing the power source presence required within the completed robot. Incorporation of a deployable skid scheme that would actuate when the microcontroller is offline, losses power or once a certain tilt angle is reached. This could be incorporated with a form of energised magnetic lock that raises the skids when system is operable and once de-energised, the skids are deployed thus preventing the robot from falling over.

Bibliography

[1] Anderson, DP, 2007, *nBot Balancing Robot*, viewed 20th March 2008, <http://www.geology.smu.edu/~dpa-www/robo/nbot/>

[2] Andrade-Cetto, J & Sanfeliu, A 2006, *Environment Learning for Indoor Mobile Robots*, Springer, New York.

[3] Angeles, J 2007, *Fundamentals of Robotic Mechanical Systems*, Springer, New York.

[4] Banks, D 2006, *Microengineering MEMs and Interfacing*, Taylor & Francis, London.

[5] Bates, Hellebuyck, Ibrahim, Jasio, D, Morton, Smith, D, Smith, J & Wilmshurst 2008, *PIC Microcontroller*, Elsevier Inc, New York.

[6] Bergren, C 2003, *Anatomy of a Robot*, McGraw-Hill, Sydney.

[7] Bishop, R 2002, *The Mechatronics Handbook*, CRC Press, London.

[8] Bishop, R 2006, *Mechatronics - An Introduction*, Taylor & Francis, London.

[9] Bokor, J, Hangos, K & Szederkenyi, G 2004, *Analysis & Control of Nonlinear Process Systems*, Springer, New York.

[10] Braunl, T 2006, *Embedded Robotics*, Springer, Perth.

[11] Bush, L 2001, Fuzzy Logic Controller for the Inverted Pendulum Problem.

[12] Carroll, T & Miles, P 2002, *Build Your Own Combat Robot*, McGraw-Hill, Sydney.

[13] Chuan-Kai 2001, *A Reinforcement Learning Adaptive Fuzzy Controller for robots*.

[14] Dawson, D & Lewis, F 2004, *Robot Manipulator Control - Theory and Practice*, Marcel Dekker, New York.

[15] De Almeida, A & Khatib, O 1998, *Autonomous Robotic Systems*, Springer, Palo Alto.

[16] Deepa, S, Sivanandam S & Sumathi, S 2007, *Introduction to Fuzzy Logic using MATLAB*, Springer, New York.

[17] Dickmanns, E 2007, *Dynamic Vision for Perception and Control of Motion*, Springer, London.

[18] Florczyk, S 2005, *Robot Vision – Video-based Indoor Exploration with Autonomous & Mobile Robots*, Wiley-VCH, Weinheim.

[19] Florian, RV, 2007, *Correct equations for the dynamics of the cart-pole system*, Center for Cognitive and Neural Studies, viewed 15th May 2008, <http://www.coneural.org/florian/papers/05_cart_pole.pdf>

[20] Holland, J 2004, *Designing Mobile Autonomous Robots*, Elsevier, Sydney.

[21] Iovine, J 2002, *Robots Androids and Animatrons*, McGraw-Hill, Sydney.

[22] Iovine, J 2004, *PIC Robotics*, McGraw-Hill, Sydney.

[23] Jain, L, Konar, A, Patnaik, S, Resconi, G & Tzafestas, S 2005, *Innovations in Robot Mobility and Control*, Springer, New York.

[24] Kageyama, Y, 2007, *New Hitachi robot has a rough debut*, Associated Press, USA Today, viewed 22nd May 2008, <http://www.usatoday.com/tech/news/robotics/2007-11-21-robot-crash_N.htm>

[25] Kurfess, T 2005, *Robotics and Automation Handbook*, CRC Press, London.

[26] Larson, T, 2008, *Balancing Robot Project - Bender*, viewed 20th March 2008, <http://www.tedlarson.com/robots/balancingbot.htm>

[27] Lewis, F & Sam Ge, S 2006, *Autonomous Mobile Robots*, Taylor & Francis, Boca Raton.

[28] McComb, G & Predko, M 2006, *Robot Builders Bonanza*, McGraw-Hill, Sydney.

[29] Pelz, G 2003, *Mechatronic Systems – Modelling & Simulation with HDLs*, Wiley, Heidelberg.

[30] Passino, K & Yurkovich, S 1998, *Fuzzy Control*, Addison-Wesley, Sydney.

[31] Sandin, P 2003, *Robot Mechanisms and Mechanical Devices Illustrated*, McGraw-Hill, Sydney.

[32] Segway Inc, 2008, *Simply moving*, viewed 20th March 2008, <http://www.segway.com/>

[33] Shircliff, D 2002, *Build a Remote-Controlled Robot*, McGraw-Hill, Sydney.

[34] Simon, D 2001, *Kalman Filtering*, Embedded Systems Programming.

[35] Stone, P 2007, *Intelligent Autonomous Robotics - A Robot Soccer Case Study*, Morgan & Claypool, Austin.

[36] USQ, 2008, *ENG4111/4112 Research Project 1 & 2*, Study Material, University of Southern Queensland.

[37] Vernon, D 1991, *Machine Vision – Automated Visual Inspection & Robot Vision*, Prentice Hall, Sydney.

[38] Williams, D 2003, *PDA Robotics – Using Your Personal Digital Assistant to Control Your Robot*, McGraw-Hill, Sydney.

[39] Williams, K 2003, *Amphibionics - Build Your Own Reptilian Robot*, McGraw-Hill, Sydney.

[40] Wise, E 2005, *Robotics Demystified*, McGraw-Hill, Sydney.

[41] Xie, M 2003, *Fundamentals of Robotics - Linking Perception to Action*, World Scientific Publishing, London.

Appendix A

University of Southern Queensland

FACULTY OF ENGINEERING AND SURVEYING

ENG4111/4112 Research Project
PROJECT SPECIFICATION

FOR: **PETER MILLER**

TOPIC: BUILDING A TWO WHEELED BALANCING ROBOT BASE

SUPERVISOR: Mark Phythian

SPONSORHSIP: USQ

PROJECT AIM: This project aims to design and create a stable robot base utilising a PIC microprocessor and suitable control system algorithm.

PROGRAMME: **Issue A, 25th March 2008**

1. Research the theory of an inverted pendulum and the various considerations necessary during the construction of a robotic machine.
2. Design and simulate a control system capable of stabilising the final platform design concept.
3. Design and implement a microprocessor unit to perform the control and stability features of the robot base. Additional, the unit will interface with a form of motion/direction control.
4. Manufacture and assemble the robot base including completion of the microprocessor programming and software configurations.
5. Analyse the overall system performance providing continual improvements to the robustness of the stability and locomotion.
6. Submit a final dissertation consisting of the design, simulation, manufacture and testing of the derived robot base.

As time permits:

7. Design and implement a form of navigation/mapping system to allow autonomous motion control of the completed robot.

AGREED: ______________ (Student) ______________ (Supervisor)

Date: 25/03/2008 Date: 25/3/2008

Examiner/Co-examiner: ______________

Appendix B

Matlab Programming

```
[System]
Name='Stability'
Type='mamdani'
Version=2.0
NumInputs=4
NumOutputs=1
NumRules=427
AndMethod='prod'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='theta-angle'
Range=[-1 1]
NumMFs=7
MF1='NL':'trimf',[-625 -0.628 -0.35]
MF2='NM':'trimf',[-0.628 -0.35 -0.11]
MF3='NS':'trimf',[-0.35 -0.11 0]
MF4='CE':'trimf',[-0.11 0 0.11]
MF5='PS':'trimf',[0 0.11 0.35]
MF6='PM':'trimf',[0.11 0.35 0.628]
MF7='PL':'trimf',[0.35 0.628 625]

[Input2]
Name='dottheta-anglevelocity'
Range=[-7 7]
NumMFs=7
MF1='NL':'trimf',[-4000 -5 -3]
MF2='NM':'trimf',[-5 -3 -1]
MF3='NS':'trimf',[-3 -1 0]
MF4='CE':'trimf',[-1 0 1]
MF5='PS':'trimf',[0 1 3]
MF6='PM':'trimf',[1 3 5]
MF7='PL':'trimf',[3 5 4000]

[Input3]
Name='x-distance'
Range=[-0.12 0.12]
NumMFs=7
MF1='NL':'trimf',[-24 -0.1 -0.07]
MF2='NM':'trimf',[-0.1 -0.07 -0.04]
MF3='NS':'trimf',[-0.07 -0.04 0]
MF4='CE':'trimf',[-0.04 0 0.04]
```

```
MF5='PS':'trimf',[0 0.04 0.07]
MF6='PM':'trimf',[0.04 0.07 0.1]
MF7='PL':'trimf',[0.07 0.1 24]

[Input4]
Name='dotx-velocity'
Range=[-0.5 0.5]
NumMFs=7
MF1='NL':'trimf',[-100 -0.45 -0.3]
MF2='NM':'trimf',[-0.45 -0.3 -0.15]
MF3='NS':'trimf',[-0.3 -0.15 0]
MF4='CE':'trimf',[-0.15 0 0.15]
MF5='PS':'trimf',[0 0.15 0.3]
MF6='PM':'trimf',[0.15 0.3 0.45]
MF7='PL':'trimf',[0.3 0.45 100]

[Output1]
Name='Motor-Output'
Range=[-120 120]
NumMFs=7
MF1='NL':'trimf',[-10000 -100 -70]
MF2='NM':'trimf',[-100 -70 -30]
MF3='NS':'trimf',[-70 -30 0]
MF4='CE':'trimf',[-30 0 30]
MF5='PS':'trimf',[0 30 70]
MF6='PM':'trimf',[30 70 100]
MF7='PL':'trimf',[70 100 10000]

[Rules]
1 1 -1 -1, 7 (1) : 1
1 1 1 1, 7 (1) : 1
1 2 -1 -1, 7 (1) : 1
1 2 1 1, 7 (1) : 1
2 1 -1 -1, 7 (1) : 1
2 1 1 1, 7 (1) : 1
1 3 -1 -1, 6 (1) : 1
1 3 1 1, 6 (1) : 1
1 4 -1 -1, 6 (1) : 1
1 4 1 1, 6 (1) : 1
1 5 -1 -1, 5 (1) : 1
1 5 1 1, 5 (1) : 1
1 6 -1 -1, 5 (1) : 1
1 6 1 1, 5 (1) : 1
2 2 -1 -1, 6 (1) : 1
2 2 1 1, 6 (1) : 1
2 3 -1 -1, 6 (1) : 1
2 3 1 1, 6 (1) : 1
2 4 -1 -1, 5 (1) : 1
2 4 1 1, 5 (1) : 1
2 5 -1 -1, 5 (1) : 1
```

2 5 1 1, 5 (1) : 1
2 7 -1 -1, 3 (1) : 1
2 7 1 1, 3 (1) : 1
3 1 -1 -1, 6 (1) : 1
3 1 1 1, 6 (1) : 1
3 2 -1 -1, 6 (1) : 1
3 2 1 1, 6 (1) : 1
3 3 -1 -1, 5 (1) : 1
3 3 1 1, 5 (1) : 1
3 4 -1 -1, 5 (1) : 1
3 4 1 1, 5 (1) : 1
3 6 -1 -1, 3 (1) : 1
3 6 1 1, 3 (1) : 1
3 7 -1 -1, 3 (1) : 1
3 7 1 1, 3 (1) : 1
4 1 -1 -1, 6 (1) : 1
4 1 1 1, 6 (1) : 1
4 2 -1 -1, 5 (1) : 1
4 2 1 1, 5 (1) : 1
4 3 -1 -1, 5 (1) : 1
4 3 1 1, 5 (1) : 1
1 7 1 1, 7 (1) : 1
1 7 1 2, 7 (1) : 1
1 7 1 3, 6 (1) : 1
1 7 1 4, 6 (1) : 1
1 7 1 5, 5 (1) : 1
1 7 1 6, 5 (1) : 1
1 7 1 7, 4 (1) : 1
1 7 2 1, 7 (1) : 1
1 7 2 2, 6 (1) : 1
1 7 2 3, 6 (1) : 1
1 7 2 4, 5 (1) : 1
1 7 2 5, 5 (1) : 1
1 7 2 6, 4 (1) : 1
1 7 2 7, 3 (1) : 1
1 7 3 1, 6 (1) : 1
1 7 3 2, 6 (1) : 1
1 7 3 3, 5 (1) : 1
1 7 3 4, 5 (1) : 1
1 7 3 5, 4 (1) : 1
1 7 3 6, 3 (1) : 1
1 7 3 7, 3 (1) : 1
1 7 4 1, 6 (1) : 1
1 7 4 2, 5 (1) : 1
1 7 4 3, 5 (1) : 1
1 7 4 4, 4 (1) : 1
1 7 4 5, 3 (1) : 1
1 7 4 6, 3 (1) : 1
1 7 4 7, 2 (1) : 1
1 7 5 1, 5 (1) : 1

1 7 5 2, 5 (1) : 1
1 7 5 3, 4 (1) : 1
1 7 5 4, 3 (1) : 1
1 7 5 5, 3 (1) : 1
1 7 5 6, 2 (1) : 1
1 7 5 7, 2 (1) : 1
1 7 6 1, 5 (1) : 1
1 7 6 2, 4 (1) : 1
1 7 6 3, 3 (1) : 1
1 7 6 4, 3 (1) : 1
1 7 6 5, 2 (1) : 1
1 7 6 6, 2 (1) : 1
1 7 6 7, 1 (1) : 1
1 7 7 1, 4 (1) : 1
1 7 7 2, 3 (1) : 1
1 7 7 3, 3 (1) : 1
1 7 7 4, 2 (1) : 1
1 7 7 5, 2 (1) : 1
1 7 7 6, 1 (1) : 1
1 7 7 7, 1 (1) : 1
2 6 1 1, 7 (1) : 1
2 6 1 2, 7 (1) : 1
2 6 1 3, 6 (1) : 1
2 6 1 4, 6 (1) : 1
2 6 1 5, 5 (1) : 1
2 6 1 6, 5 (1) : 1
2 6 1 7, 4 (1) : 1
2 6 2 1, 7 (1) : 1
2 6 2 2, 6 (1) : 1
2 6 2 3, 6 (1) : 1
2 6 2 4, 5 (1) : 1
2 6 2 5, 5 (1) : 1
2 6 2 6, 4 (1) : 1
2 6 2 7, 3 (1) : 1
2 6 3 1, 6 (1) : 1
2 6 3 2, 6 (1) : 1
2 6 3 3, 5 (1) : 1
2 6 3 4, 5 (1) : 1
2 6 3 5, 4 (1) : 1
2 6 3 6, 3 (1) : 1
2 6 3 7, 3 (1) : 1
2 6 4 1, 6 (1) : 1
2 6 4 2, 5 (1) : 1
2 6 4 3, 5 (1) : 1
2 6 4 4, 4 (1) : 1
2 6 4 5, 3 (1) : 1
2 6 4 6, 3 (1) : 1
2 6 4 7, 2 (1) : 1
2 6 5 1, 5 (1) : 1
2 6 5 2, 5 (1) : 1

2 6 5 3, 4 (1) : 1
2 6 5 4, 3 (1) : 1
2 6 5 5, 3 (1) : 1
2 6 5 6, 2 (1) : 1
2 6 5 7, 2 (1) : 1
2 6 6 1, 5 (1) : 1
2 6 6 2, 4 (1) : 1
2 6 6 3, 3 (1) : 1
2 6 6 4, 3 (1) : 1
2 6 6 5, 2 (1) : 1
2 6 6 6, 2 (1) : 1
2 6 6 7, 1 (1) : 1
2 6 7 1, 4 (1) : 1
2 6 7 2, 3 (1) : 1
2 6 7 3, 3 (1) : 1
2 6 7 4, 2 (1) : 1
2 6 7 5, 2 (1) : 1
2 6 7 6, 1 (1) : 1
2 6 7 7, 1 (1) : 1
3 5 1 1, 7 (1) : 1
3 5 1 2, 7 (1) : 1
3 5 1 3, 6 (1) : 1
3 5 1 4, 6 (1) : 1
3 5 1 5, 5 (1) : 1
3 5 1 6, 5 (1) : 1
3 5 1 7, 4 (1) : 1
3 5 2 1, 7 (1) : 1
3 5 2 2, 6 (1) : 1
3 5 2 3, 6 (1) : 1
3 5 2 4, 5 (1) : 1
3 5 2 5, 5 (1) : 1
3 5 2 6, 4 (1) : 1
3 5 2 7, 3 (1) : 1
3 5 3 1, 6 (1) : 1
3 5 3 2, 6 (1) : 1
3 5 3 3, 5 (1) : 1
3 5 3 4, 5 (1) : 1
3 5 3 5, 4 (1) : 1
3 5 3 6, 3 (1) : 1
3 5 3 7, 3 (1) : 1
3 5 4 1, 6 (1) : 1
3 5 4 2, 5 (1) : 1
3 5 4 3, 5 (1) : 1
3 5 4 4, 4 (1) : 1
3 5 4 5, 3 (1) : 1
3 5 4 6, 3 (1) : 1
3 5 4 7, 2 (1) : 1
3 5 5 1, 5 (1) : 1
3 5 5 2, 5 (1) : 1
3 5 5 3, 4 (1) : 1

3 5 5 4, 3 (1) : 1
3 5 5 5, 3 (1) : 1
3 5 5 6, 2 (1) : 1
3 5 5 7, 2 (1) : 1
3 5 6 1, 5 (1) : 1
3 5 6 2, 4 (1) : 1
3 5 6 3, 3 (1) : 1
3 5 6 4, 3 (1) : 1
3 5 6 5, 2 (1) : 1
3 5 6 6, 2 (1) : 1
3 5 6 7, 1 (1) : 1
3 5 7 1, 4 (1) : 1
3 5 7 2, 3 (1) : 1
3 5 7 3, 3 (1) : 1
3 5 7 4, 2 (1) : 1
3 5 7 5, 2 (1) : 1
3 5 7 6, 1 (1) : 1
3 5 7 7, 1 (1) : 1
4 4 1 1, 7 (1) : 1
4 4 1 2, 7 (1) : 1
4 4 1 3, 6 (1) : 1
4 4 1 4, 6 (1) : 1
4 4 1 5, 5 (1) : 1
4 4 1 6, 5 (1) : 1
4 4 1 7, 4 (1) : 1
4 4 2 1, 7 (1) : 1
4 4 2 2, 6 (1) : 1
4 4 2 3, 6 (1) : 1
4 4 2 4, 5 (1) : 1
4 4 2 5, 5 (1) : 1
4 4 2 6, 4 (1) : 1
4 4 2 7, 3 (1) : 1
4 4 3 1, 6 (1) : 1
4 4 3 2, 6 (1) : 1
4 4 3 3, 5 (1) : 1
4 4 3 4, 5 (1) : 1
4 4 3 5, 4 (1) : 1
4 4 3 6, 3 (1) : 1
4 4 3 7, 3 (1) : 1
4 4 4 1, 6 (1) : 1
4 4 4 2, 5 (1) : 1
4 4 4 3, 5 (1) : 1
4 4 4 4, 4 (1) : 1
4 4 4 5, 3 (1) : 1
4 4 4 6, 3 (1) : 1
4 4 4 7, 2 (1) : 1
4 4 5 1, 5 (1) : 1
4 4 5 2, 5 (1) : 1
4 4 5 3, 4 (1) : 1
4 4 5 4, 3 (1) : 1

4 4 5 5, 3 (1) : 1
4 4 5 6, 2 (1) : 1
4 4 5 7, 2 (1) : 1
4 4 6 1, 5 (1) : 1
4 4 6 2, 4 (1) : 1
4 4 6 3, 3 (1) : 1
4 4 6 4, 3 (1) : 1
4 4 6 5, 2 (1) : 1
4 4 6 6, 2 (1) : 1
4 4 6 7, 1 (1) : 1
4 4 7 1, 4 (1) : 1
4 4 7 2, 3 (1) : 1
4 4 7 3, 3 (1) : 1
4 4 7 4, 2 (1) : 1
4 4 7 5, 2 (1) : 1
4 4 7 6, 1 (1) : 1
4 4 7 7, 1 (1) : 1
5 3 1 1, 7 (1) : 1
5 3 1 2, 7 (1) : 1
5 3 1 3, 6 (1) : 1
5 3 1 4, 6 (1) : 1
5 3 1 5, 5 (1) : 1
5 3 1 6, 5 (1) : 1
5 3 1 7, 4 (1) : 1
5 3 2 1, 7 (1) : 1
5 3 2 2, 6 (1) : 1
5 3 2 3, 6 (1) : 1
5 3 2 4, 5 (1) : 1
5 3 2 5, 5 (1) : 1
5 3 2 6, 4 (1) : 1
5 3 2 7, 3 (1) : 1
5 3 3 1, 6 (1) : 1
5 3 3 2, 6 (1) : 1
5 3 3 3, 5 (1) : 1
5 3 3 4, 5 (1) : 1
5 3 3 5, 4 (1) : 1
5 3 3 6, 3 (1) : 1
5 3 3 7, 3 (1) : 1
5 3 4 1, 6 (1) : 1
5 3 4 2, 5 (1) : 1
5 3 4 3, 5 (1) : 1
5 3 4 4, 4 (1) : 1
5 3 4 5, 3 (1) : 1
5 3 4 6, 3 (1) : 1
5 3 4 7, 2 (1) : 1
5 3 5 1, 5 (1) : 1
5 3 5 2, 5 (1) : 1
5 3 5 3, 4 (1) : 1
5 3 5 4, 3 (1) : 1
5 3 5 5, 3 (1) : 1

5 3 5 6, 2 (1) : 1
5 3 5 7, 2 (1) : 1
5 3 6 1, 5 (1) : 1
5 3 6 2, 4 (1) : 1
5 3 6 3, 3 (1) : 1
5 3 6 4, 3 (1) : 1
5 3 6 5, 2 (1) : 1
5 3 6 6, 2 (1) : 1
5 3 6 7, 1 (1) : 1
5 3 7 1, 4 (1) : 1
5 3 7 2, 3 (1) : 1
5 3 7 3, 3 (1) : 1
5 3 7 4, 2 (1) : 1
5 3 7 5, 2 (1) : 1
5 3 7 6, 1 (1) : 1
5 3 7 7, 1 (1) : 1
6 2 1 1, 7 (1) : 1
6 2 1 2, 7 (1) : 1
6 2 1 3, 6 (1) : 1
6 2 1 4, 6 (1) : 1
6 2 1 5, 5 (1) : 1
6 2 1 6, 5 (1) : 1
6 2 1 7, 4 (1) : 1
6 2 2 1, 7 (1) : 1
6 2 2 2, 6 (1) : 1
6 2 2 3, 6 (1) : 1
6 2 2 4, 5 (1) : 1
6 2 2 5, 5 (1) : 1
6 2 2 6, 4 (1) : 1
6 2 2 7, 3 (1) : 1
6 2 3 1, 6 (1) : 1
6 2 3 2, 6 (1) : 1
6 2 3 3, 5 (1) : 1
6 2 3 4, 5 (1) : 1
6 2 3 5, 4 (1) : 1
6 2 3 6, 3 (1) : 1
6 2 3 7, 3 (1) : 1
6 2 4 1, 6 (1) : 1
6 2 4 2, 5 (1) : 1
6 2 4 3, 5 (1) : 1
6 2 4 4, 4 (1) : 1
6 2 4 5, 3 (1) : 1
6 2 4 6, 3 (1) : 1
6 2 4 7, 2 (1) : 1
6 2 5 1, 5 (1) : 1
6 2 5 2, 5 (1) : 1
6 2 5 3, 4 (1) : 1
6 2 5 4, 3 (1) : 1
6 2 5 5, 3 (1) : 1
6 2 5 6, 2 (1) : 1

6 2 5 7, 2 (1) : 1
6 2 6 1, 5 (1) : 1
6 2 6 2, 4 (1) : 1
6 2 6 3, 3 (1) : 1
6 2 6 4, 3 (1) : 1
6 2 6 5, 2 (1) : 1
6 2 6 6, 2 (1) : 1
6 2 6 7, 1 (1) : 1
6 2 7 1, 4 (1) : 1
6 2 7 2, 3 (1) : 1
6 2 7 3, 3 (1) : 1
6 2 7 4, 2 (1) : 1
6 2 7 5, 2 (1) : 1
6 2 7 6, 1 (1) : 1
6 2 7 7, 1 (1) : 1
7 1 1 1, 7 (1) : 1
7 1 1 2, 7 (1) : 1
7 1 1 3, 6 (1) : 1
7 1 1 4, 6 (1) : 1
7 1 1 5, 5 (1) : 1
7 1 1 6, 5 (1) : 1
7 1 1 7, 4 (1) : 1
7 1 2 1, 7 (1) : 1
7 1 2 2, 6 (1) : 1
7 1 2 3, 6 (1) : 1
7 1 2 4, 5 (1) : 1
7 1 2 5, 5 (1) : 1
7 1 2 6, 4 (1) : 1
7 1 2 7, 3 (1) : 1
7 1 3 1, 6 (1) : 1
7 1 3 2, 6 (1) : 1
7 1 3 3, 5 (1) : 1
7 1 3 4, 5 (1) : 1
7 1 3 5, 4 (1) : 1
7 1 3 6, 3 (1) : 1
7 1 3 7, 3 (1) : 1
7 1 4 1, 6 (1) : 1
7 1 4 2, 5 (1) : 1
7 1 4 3, 5 (1) : 1
7 1 4 4, 4 (1) : 1
7 1 4 5, 3 (1) : 1
7 1 4 6, 3 (1) : 1
7 1 4 7, 2 (1) : 1
7 1 5 1, 5 (1) : 1
7 1 5 2, 5 (1) : 1
7 1 5 3, 4 (1) : 1
7 1 5 4, 3 (1) : 1
7 1 5 5, 3 (1) : 1
7 1 5 6, 2 (1) : 1
7 1 5 7, 2 (1) : 1

7 1 6 1, 5 (1) : 1
7 1 6 2, 4 (1) : 1
7 1 6 3, 3 (1) : 1
7 1 6 4, 3 (1) : 1
7 1 6 5, 2 (1) : 1
7 1 6 6, 2 (1) : 1
7 1 6 7, 1 (1) : 1
7 1 7 1, 4 (1) : 1
7 1 7 2, 3 (1) : 1
7 1 7 3, 3 (1) : 1
7 1 7 4, 2 (1) : 1
7 1 7 5, 2 (1) : 1
7 1 7 6, 1 (1) : 1
7 1 7 7, 1 (1) : 1
4 5 -1 -1, 3 (1) : 1
4 5 1 1, 3 (1) : 1
4 6 -1 -1, 3 (1) : 1
4 6 1 1, 3 (1) : 1
4 7 -1 -1, 2 (1) : 1
4 7 1 1, 2 (1) : 1
5 1 -1 -1, 5 (1) : 1
5 1 1 1, 5 (1) : 1
5 2 -1 -1, 5 (1) : 1
5 2 1 1, 5 (1) : 1
5 4 -1 -1, 3 (1) : 1
5 4 1 1, 3 (1) : 1
5 5 -1 -1, 3 (1) : 1
5 5 1 1, 3 (1) : 1
5 6 -1 -1, 2 (1) : 1
5 6 1 1, 2 (1) : 1
5 7 -1 -1, 2 (1) : 1
5 7 1 1, 2 (1) : 1
6 1 -1 -1, 5 (1) : 1
6 1 1 1, 5 (1) : 1
6 3 -1 -1, 3 (1) : 1
6 3 1 1, 3 (1) : 1
6 4 -1 -1, 3 (1) : 1
6 4 1 1, 3 (1) : 1
6 5 -1 -1, 2 (1) : 1
6 5 1 1, 2 (1) : 1
6 6 -1 -1, 2 (1) : 1
6 6 1 1, 2 (1) : 1
6 7 -7 -7, 1 (1) : 1
6 7 7 7, 1 (1) : 1
7 2 -7 -7, 3 (1) : 1
7 2 7 7, 3 (1) : 1
7 3 -7 -7, 3 (1) : 1
7 3 7 7, 3 (1) : 1
7 4 -7 -7, 2 (1) : 1
7 4 7 7, 2 (1) : 1

7 5 -7 -7, 2 (1) : 1
7 5 7 7, 2 (1) : 1
7 6 -7 -7, 1 (1) : 1
7 6 7 7, 1 (1) : 1
7 7 -7 -7, 1 (1) : 1
7 7 7 7, 1 (1) : 1

Appendix C

Microcontroller Programming

```
///////////////////////////////////////////////////////////////////
// ENG4112 Research Project Part 2
// Microcontroller Stability Program
// "Bullding a Two Wheeled Balancing Robot"
//
// Author:      Peter Miller
// Version:     27OCT08
///////////////////////////////////////////////////////////////////

#include "p24HJ256GP206.h"                    /*Include header definition
file*/

///////////////////////////////////////////////////////////////////
// Defined values
///////////////////////////////////////////////////////////////////

// Vertical angle membership function for fuzzy controller
#define ANGLE_NL_LOWER -625
#define ANGLE_NL_UPPER -0.5
#define ANGLE_NM_LOWER -0.5
#define ANGLE_NM_UPPER -0.22
#define ANGLE_NS_LOWER -0.22
#define ANGLE_NS_UPPER -0.08
#define ANGLE_CE_LOWER -0.08
#define ANGLE_CE_UPPER 0.08
#define ANGLE_PS_LOWER 0.08
#define ANGLE_PS_UPPER 0.22
#define ANGLE_PM_LOWER 0.22
#define ANGLE_PM_UPPER 0.5
#define ANGLE_PL_LOWER 0.5
#define ANGLE_PL_UPPER 625

// Vertical anglular velocity membership function for fuzzy controller
#define ANGLE_VEL_NL_LOWER -4000
#define ANGLE_VEL_NL_UPPER -4
#define ANGLE_VEL_NM_LOWER -4
#define ANGLE_VEL_NM_UPPER -2
#define ANGLE_VEL_NS_LOWER -2
#define ANGLE_VEL_NS_UPPER -0.06
#define ANGLE_VEL_CE_LOWER -0.06
#define ANGLE_VEL_CE_UPPER 0.06
#define ANGLE_VEL_PS_LOWER 0.06
#define ANGLE_VEL_PS_UPPER 2
#define ANGLE_VEL_PM_LOWER 2
#define ANGLE_VEL_PM_UPPER 4
```

```
#define ANGLE_VEL_PL_LOWER 4
#define ANGLE_VEL_PL_UPPER 4000

// Horizontal displacement membership function for fuzzy controller
#define DISP_NL_LOWER -24
#define DISP_NL_UPPER -0.08
#define DISP_NM_LOWER -0.08
#define DISP_NM_UPPER -0.06
#define DISP_NS_LOWER -0.06
#define DISP_NS_UPPER -0.025
#define DISP_CE_LOWER -0.025
#define DISP_CE_UPPER 0.025
#define DISP_PS_LOWER 0.025
#define DISP_PS_UPPER 0.06
#define DISP_PM_LOWER 0.06
#define DISP_PM_UPPER 0.08
#define DISP_PL_LOWER 0.08
#define DISP_PL_UPPER 24

// Horizontal velocity membership function for fuzzy controller
#define VEL_NL_LOWER -100
#define VEL_NL_UPPER -0.45
#define VEL_NM_LOWER -0.45
#define VEL_NM_UPPER -0.3
#define VEL_NS_LOWER -0.3
#define VEL_NS_UPPER -0.15
#define VEL_CE_LOWER -0.15
#define VEL_CE_UPPER 0.15
#define VEL_PS_LOWER 0.15
#define VEL_PS_UPPER 0.3
#define VEL_PM_LOWER 0.3
#define VEL_PM_UPPER 0.45
#define VEL_PL_LOWER 0.45
#define VEL_PL_UPPER 100

// Output membership function for fuzzy controller
#define NL_LOWER -625
#define NL_UPPER -0.38
#define NM_LOWER -0.38
#define NM_UPPER -0.23
#define NS_LOWER -0.23
#define NS_UPPER -0.085
#define CE_LOWER -0.085
#define CE_UPPER 0.085
#define PS_LOWER 0.085
#define PS_UPPER 0.23
#define PM_LOWER 0.23
#define PM_UPPER 0.38
#define PL_LOWER 0.38
#define PL_UPPER 625
```

```
#define NL 1
#define NM 2
#define NS 3
#define CE 4
#define PS 5
#define PM 6
#define PL 7

///////////////////////////////////////////////////////////////////
// Initiate global variables
///////////////////////////////////////////////////////////////////
static float Accel_input;
static float Accel_digital;
static float Gyro_input;
static float Gyro_digital;
static float Bumpers1_4_input;
static float Angle;
static float Angle_vel;
static float Displace;
static float Disp_vel;
static float Fuzzy_output;
static float Kalman_ouput;
static float K_output;
static float PID_output;
static float Left_motor_PID;
static float Right_motor_PID;
static int Left_shaft_input;
static int Right_shaft_input;
static int Left_encoder_counter;
static int Right_encoder_counter;
static int Angle_fuzzied;
static int Angle_vel_fuzzied;
static int Displace_fuzzied;
static int Disp_vel_fuzzied;
static int Fuzzy_motor_output;
static int Motor_output;
static int Left_encoder_input;
static int Left_motor_output;
static int Right_motor_ouput;
static int Right_encoder_input;
static int Left_drive_outputl1;
static int Left_drive_outputl2;
static int Right_drive_ouputl3;
static int Right_drive_ouputl4;
static int Enable_drive_output;
```

```
////////////////////////////////////////////////////////////////////
// Function sets
////////////////////////////////////////////////////////////////////
int Sensors();
//void Locomotion_Desired();          // Function not implemented at this time
static int Fuzzy_Control();
int Horizontal_Fuzzy ();
int Locomotion_Response();
//float PID_Function ();              // Function not implemeneted at this time
static float Kalman_Function ();

////////////////////////////////////////////////////////////////////
// Main program
//
// Input          : Nil.
// Output         : Nil.
// Summary        : Initialises the system, then calls functions within a while loop.
//
////////////////////////////////////////////////////////////////////
int main ()

{
// Initialisation of microcontroller hardware
//PORTA = 0x00;                 // Turn off port A
PORTB = 0x00;                   // Turn off port B
PORTC = 0x00;                   // Turn off port C
PORTD = 0xFF;                   // Turn on port D
//PORTE = 0x00;                 // Turn off port E
PORTF = 0x00;                   // Turn off port F
PORTG = 0xFF;                   // Turn on port G

TRISB = 0xFFFF;                 // Configure all PortB as input,
TRISC = 0xFFFF;                 // Configure all PortC as input
TRISD = 0x0000;                 // Cconfigure all PortD as output
//TRISE = 0xFFFF;               // Configure all PortE as input
TRISF = 0xFFFF;                 // Configure all PortF as input,
TRISG = 0xFFFF;                 // Cconfigure all PortG as input

AD1PCFGL = 0xFFFF;              // Assign analogue operation to pins

Gyro_input= ;                   // Gyroscope input ANO 7-8
Accel_input=;                   // Accelerometer input ANO 4-5
AD_converters=;                 // A/D conversion sequence

Bumpers1_4_input=_LATG0x000F;         // Bumpers 1-4 input RG 0-3
Left_shaft_input=_LATG0x3000;         // Left shaft encoder input RG 12-13
Right_shaft_input=_LATG0xC000;        // Right shaft encoder input RG 14-15

Left_drive_outputl1=LATDbits.LATD0x10;          // Left shaft output RD 4
```

```
Left_drive_outputl2=LATDbits.LATD0x08;        // Left shaft output RD 3
Right_drive_outputl3=LATDbits.LATD0x04;       // Left shaft output RD 2
Right_drive_outputl4=LATDbits.LATD0x02;       // Right shaft output RD 1
Enable_drive_output=LATDbits.LATD0x01;        // Motor output enable RD 0

Left_encoder_counter=;               // Left encoder counter sequence
Right_encoder_counter=;              // Right encoder counter sequence
Timer_interupt=;                     // Timer for completing cycle

// Initialisation of timing cycles and counters.

while (Bumpers1_4_input)             // while (Bumpers are true) sequence

{
Sensors (Angle, Angle_vel, Displace, Disp_vel);
//Locomotion_Desired ();             // Function not implemented at this time
Fuzzy_Control (Fuzzy_motor_output);
Locomotion_Response ();
}

return (0);
}
```

```
//////////////////////////////////////////////////////////////////////
// Sensor function
//
// Input         : Left_shaft_input, Right_shaft_input, Bumpers1_4_input,
Gyro_input, //           Accel_input.
// Output        : Angle, Angle_vel, Displace, Disp_vel, Gyro_digital, Accel_digital.
// Summary       : Reads the digital and analogue sensor inputs. Converts to digital,
//                 calls the Kalman function and provides the 4 stability output values.
//
//////////////////////////////////////////////////////////////////////
static Sensors (Angle, Angle_vel, Displace, Disp_vel)

{
// Read digital sensor inputs for Bumpers and Shaft encoders
Left_shaft_input                    // Left shaft encoder input RG 12-13
Right_shaft_input                   // Right shaft encoder input RG 14-15
Bumpers1_4_input                    // Bumpers 1-4 input RG 0-3

// Read analog sensor inputs & convert to digital (Gyroscope & accelerometer)
Gyro_input                          // Gyroscope input ANO 7-8
Accel_input                         // Accelerometer input ANO (x)4-5(y)

// Perform A/D conversion calculations
Gyro_digital= ;                     // Gyroscope digital value
Accel_digital= ;                    // Accelerometer digital value

// Proximity detection sensor input (ANO 0-2), Not implemented at this time.

// Kalman filtering sequence
Kalman_ouput=Kalman(K_output);           // Kalman filter sensor fused output

// Stability value calculations
Angle=Kalman_output;
Angle_vel= "calculate derivative of Angle";
Displace=((Right_shaft_input+Left_shaft_input)/2);
Disp_vel= "calculate derivative of Displace";

return (Angle, Angle_vel, Displace, Disp_vel);
}
```

```
//////////////////////////////////////////////////////////////////////
// Locomotion Desired function
//
// Input       : Nil at this time.
// Output      : Nil at this time.
// Summary     : Function not implemented as the interaction computer input is not
//               available at this time.
//
//////////////////////////////////////////////////////////////////////
static void Locomotion_Desired (void)

{
// Read desired direction control – Not implemented under current project.

// Left motor desired response calculations - Not implemented under current project.

// Right motor desired response calculations - Not implemented under current
project.

// Offset angle calculations for tilt in the direction of travel - Not implemented under
current project
}
return (0);
}
```

```
////////////////////////////////////////////////////////////////////
// Fuzzy Control function
//
// Input        : Angle, Angle_vel, Displace, Disp_vel.
// Output       : Fuzzy_motor_output.
// Summary      : Performs the fuzzy controller function. Fuzzification, rule base
//                and defuzzification is completed.
//
////////////////////////////////////////////////////////////////////
static int Fuzzy_Control (Fuzzy_motor_output)
{

{       // Angle fuzzification sequence
if (Angle<=ANGLE_NL_UPPER)
        Angle_fuzzied=NL;
if ((Angle<=ANGLE_NM_UPPER)&&(Angle>ANGLE_NM_LOWER))
        Angle_fuzzied=NM;
if ((Angle<=ANGLE_NS_UPPER)&&(Angle>ANGLE_NS_LOWER))
        Angle_fuzzied=NS;
if ((Angle<=ANGLE_CE_UPPER)&&(Angle>ANGLE_CE_LOWER))
        Angle_fuzzied=CE;
if ((Angle<=ANGLE_PS_UPPER)&&(Angle>ANGLE_PS_LOWER))
        Angle_fuzzied=PS;
if ((Angle<=ANGLE_PM_UPPER)&&(Angle>ANGLE_PM_LOWER))
        Angle_fuzzied=PM;
if (Angle>ANGLE_PL_LOWER)
        Angle_fuzzied=PL;
}

{       // Anglular velocity fuzzification sequence
if (Angle_vel<=ANGLE_VEL_NL_UPPER)
        Angle_vel_fuzzied=NL;
if
((Angle_vel<=ANGLE_VEL_NM_UPPER)&&(Angle_vel>ANGLE_VEL_NM_L
OWER))
        Angle_vel_fuzzied=NM;
if
((Angle_vel<=ANGLE_VEL_NS_UPPER)&&(Angle_vel>ANGLE_VEL_NS_LO
WER))
        Angle_vel_fuzzied=NS;
if
((Angle_vel<=ANGLE_VEL_CE_UPPER)&&(Angle_vel>ANGLE_VEL_CE_LO
WER))
        Angle_vel_fuzzied=CE;
if
((Angle_vel<=ANGLE_VEL_PS_UPPER)&&(Angle_vel>ANGLE_VEL_PS_LO
WER))
        Angle_vel_fuzzied=PS;
```

```
if
((Angle_vel<=ANGLE_VEL_PM_UPPER)&&(Angle_vel>ANGLE_VEL_PM_LO
WER))
          Angle_vel_fuzzied=PM;
if (Angle_vel>ANGLE_VEL_PL_LOWER)
          Angle_vel_fuzzied=PL;
}

{         // Displacement fuzzification sequence
if (Displace<=DISP_NL_UPPER)
          Displace_fuzzied=NL;
if ((Displace<=DISP_NM_UPPER)&&(Displace>DISP_NM_LOWER))
          Displace_fuzzied=NM;
if ((Displace<=DISP_NS_UPPER)&&(Displace>DISP_NS_LOWER))
          Displace_fuzzied=NS;
if ((Displace<=DISP_CE_UPPER)&&(Displace>DISP_CE_LOWER))
          Displace_fuzzied=CE;
if ((Displace<=DISP_PS_UPPER)&&(Displace>DISP_PS_LOWER))
          Displace_fuzzied=PS;
if ((Displace<=DISP_PM_UPPER)&&(Displace>DISP_PM_LOWER))
          Displace_fuzzied=PM;
if (Displace>DISP_PL_LOWER)
          Displace_fuzzied=PL;
}

{         // Displacement velocity fuzzification sequence
if (Disp_vel<=VEL_NL_UPPER)
          Disp_vel_fuzzied=NL;
if ((Disp_vel<=VEL_NM_UPPER)&&(Disp_vel>VEL_NM_LOWER))
          Disp_vel_fuzzied=NM;
if ((Disp_vel<=VEL_NS_UPPER)&&(Disp_vel>VEL_NS_LOWER))
          Disp_vel_fuzzied=NS;
if ((Disp_vel<=VEL_CE_UPPER)&&(Disp_vel>VEL_CE_LOWER))
          Disp_vel_fuzzied=CE;
if ((Disp_vel<=VEL_PS_UPPER)&&(Disp_vel>VEL_PS_LOWER))
          Disp_vel_fuzzied=PS;
if ((Disp_vel<=VEL_PM_UPPER)&&(Disp_vel>VEL_PM_LOWER))
          Disp_vel_fuzzied=PM;
if (Disp_vel>VEL_PL_LOWER)
          Disp_vel_fuzzied=PL;
}

// Rule based sequence
{
if (Angle_fuzzied<CE)
{
          if (Angle_fuzzied=NL)                     /* [NL, -, -, -] */
                    if (Angle_vel_fuzzied=NL||(Angle_vel_fuzzied=NM))
                              Fuzzy_output=PL;
                    if ((Angle_vel_fuzzied=NS)||(Angle_vel_fuzzied=CE))
```

```
                    Fuzzy_output=PM;
                if ((Angle_vel_fuzzied=PS)||(Angle_vel_fuzzied=PM))
                    Fuzzy_output=PS;
                if (Angle_vel_fuzzied=PL)
                    Horizontal_Fuzzy (Fuzzy_output);
            if (Angle_fuzzied=NM)                   /* [NM, -, -, -] */
                if (Angle_vel_fuzzied=NL)
                    Fuzzy_output=PL;
                if ((Angle_vel_fuzzied=NM)||(Angle_vel_fuzzied=NS))
                    Fuzzy_output=PM;
                if ((Angle_vel_fuzzied=CE)||(Angle_vel_fuzzied=PS))
                    Fuzzy_output=PS;
                if (Angle_vel_fuzzied=PM)
                    Horizontal_Fuzzy (Fuzzy_output);
                if (Angle_vel_fuzzied=PL)
                    Fuzzy_output=NS;
            if (Angle_fuzzied=NS)                   /* [NS, -, -, -] */
                if (Angle_vel_fuzzied=NL||(Angle_vel_fuzzied=NM))
                    Fuzzy_output=PM;
                if ((Angle_vel_fuzzied=NS)||(Angle_vel_fuzzied=CE))
                    Fuzzy_output=PS;
                if (Angle_vel_fuzzied=PS)
                    Horizontal_Fuzzy (Fuzzy_output);
                if ((Angle_vel_fuzzied=PM)||(Angle_vel_fuzzied=PL))
                    Fuzzy_output=NS;
}

{
if (Angle_fuzzied=CE)                               /* [CE, -, -, -] */
        if (Angle_vel_fuzzied=NL)
            Fuzzy_output=PM;
        if ((Angle_vel_fuzzied=NM)||(Angle_vel_fuzzied=NS))
            Fuzzy_output=PS;
        if (Angle_vel_fuzzied=CE)
            Horizontal_Fuzzy (Fuzzy_output);
        if ((Angle_vel_fuzzied=PS)||(Angle_vel_fuzzied=PM))
            Fuzzy_output=NS;
        if (Angle_vel_fuzzied=PL)
            Fuzzy_output=NM;
}

{
if (Angle_fuzzied>CE)
        if (Angle_fuzzied=PS)                       /* [PS, -, -, -] */
            if ((Angle_vel_fuzzied=NL)||(Angle_vel_fuzzied=NM))
                Fuzzy_output=PS;
            if (Angle_vel_fuzzied=NS)
                Horizontal_Fuzzy (Fuzzy_output);
            if ((Angle_vel_fuzzied=CE)||(Angle_vel_fuzzied=PS))
                Fuzzy_output=NS;
```

```
			if ((Angle_vel_fuzzied=PM)||(Angle_vel_fuzzied=PL))
				Fuzzy_output=NM;
		if (Angle_fuzzied=PM)				/* [PM, -, -, -] */
			if (Angle_vel_fuzzied=NL)
				Fuzzy_output=PS;
			if (Angle_vel_fuzzied=NM)
				Horizontal_Fuzzy (Fuzzy_output);
			if ((Angle_vel_fuzzied=NS)||(Angle_vel_fuzzied=CE))
				Fuzzy_output=NS;
			if ((Angle_vel_fuzzied=PS)||(Angle_vel_fuzzied=PM))
				Fuzzy_output=NM;
			if (Angle_vel_fuzzied=PL)
				Fuzzy_output=NL;
		if (Angle_fuzzied=PL)				/* [PL, -, -, -] */
			if (Angle_vel_fuzzied=NL)
				Horizontal_Fuzzy (Fuzzy_output);
			if ((Angle_vel_fuzzied=NM)||(Angle_vel_fuzzied=NS))
				Fuzzy_output=NS;
			if ((Angle_vel_fuzzied=CE)||(Angle_vel_fuzzied=PS))
				Fuzzy_output=NM;
			if ((Angle_vel_fuzzied=PM)||(Angle_vel_fuzzied=PL))
				Fuzzy_output=NL;
}
}

// Defuzzification sequence
{
if (Fuzzy_output<CE)
		if (Fuzzy_output=NL)
			Fuzzy_motor_output=-100;
		if (Fuzzy_output=NM)
			Fuzzy_motor_output=-70;
		if (Fuzzy_output=NS)
			Fuzzy_motor_output=-30;
if (Fuzzy_output=CE)
		Fuzzy_motor_output=0;
if (Fuzzy_output>CE)
		if (Fuzzy_output=PS)
			Fuzzy_motor_output=30;
		if (Fuzzy_output=PM)
			Fuzzy_motor_output=70;
		if (Fuzzy_output=PL)
			Fuzzy_motor_output=100;
}

return (Fuzzy_motor_output);
}
```

```
//////////////////////////////////////////////////////////////////////
// Horizontal Fuzzification function
//
// Input          : Displace_fuzzied, Disp_vel_fuzzied.
// Output         : Fuzzy_output.
// Summary        : The function performs the second series of rule base comparison
//                  for the Fuzzy control function (Horizontal adjustment).
//
//////////////////////////////////////////////////////////////////////
int Horizontal_Fuzzy (Fuzzy_output)

{
if (Displace_fuzzied<CE)
        if (Displace_fuzzied=NL)
                if
((Disp_vel_fuzzied=NL)||(Disp_vel_fuzzied=NM)||(Disp_vel_fuzzied=NS))
                        Fuzzy_output=PM;
                if
((Disp_vel_fuzzied=CE)||(Disp_vel_fuzzied=PS)||(Disp_vel_fuzzied=PM))
                        Fuzzy_output=PS;
                if (Disp_vel_fuzzied=PL)
                        Fuzzy_output=CE;
        if (Displace_fuzzied=NM)
                if ((Disp_vel_fuzzied=NL)||(Disp_vel_fuzzied=NM))
                        Fuzzy_output=PM;
                if
((Disp_vel_fuzzied=NS)||(Disp_vel_fuzzied=CE)||(Disp_vel_fuzzied=PS))
                        Fuzzy_output=PS;
                if (Disp_vel_fuzzied=PM)
                        Fuzzy_output=CE;
                if (Disp_vel_fuzzied=PL)
                        Fuzzy_output=NS;
        if (Displace_fuzzied=NS)
                if (Disp_vel_fuzzied=NL)
                        Fuzzy_output=PM;
                if
((Disp_vel_fuzzied=NM)||(Disp_vel_fuzzied=NS)||(Disp_vel_fuzzied=CE))
                        Fuzzy_output=PS;
                if (Disp_vel_fuzzied=PS)
                        Fuzzy_output=CE;
                if ((Disp_vel_fuzzied=PM)||(Disp_vel_fuzzied=PL))
                        Fuzzy_output=NS;
if (Displace_fuzzied=CE)
        if
((Disp_vel_fuzzied=NL)||(Disp_vel_fuzzied=NM)||(Disp_vel_fuzzied=NS))
                Fuzzy_output=PS;
        if (Disp_vel_fuzzied=CE)
                Fuzzy_output=CE;
        if ((Disp_vel_fuzzied=PS)||(Disp_vel_fuzzied=PM)||(Disp_vel_fuzzied=PL))
                Fuzzy_output=NS;
```

```
if (Displace_fuzzied>CE)
        if (Displace_fuzzied=PS)
                if ((Disp_vel_fuzzied=NL)||(Disp_vel_fuzzied=NM))
                        Fuzzy_output=PS;
                if (Disp_vel_fuzzied=NS)
                        Fuzzy_output=CE;
                if
((Disp_vel_fuzzied=CE)||(Disp_vel_fuzzied=PS)||(Disp_vel_fuzzied=PM))
                        Fuzzy_output=NS;
                if (Disp_vel_fuzzied=PL)
                        Fuzzy_output=NM;
        if (Displace_fuzzied=PM)
                if (Disp_vel_fuzzied=NL)
                        Fuzzy_output=PS;
                if (Disp_vel_fuzzied=NM)
                        Fuzzy_output=CE;
                if
((Disp_vel_fuzzied=NS)||(Disp_vel_fuzzied=CE)||(Disp_vel_fuzzied=PS))
                        Fuzzy_output=NS;
                if ((Disp_vel_fuzzied=PM)||(Disp_vel_fuzzied=PL))
                        Fuzzy_output=NM;
        if (Displace_fuzzied=PL)
                if (Disp_vel_fuzzied=NL)
                        Fuzzy_output=CE;
        if
((Disp_vel_fuzzied=NM)||(Disp_vel_fuzzied=NS)||(Disp_vel_fuzzied=CE))
                        Fuzzy_output=NS;
                if
((Disp_vel_fuzzied=PS)||(Disp_vel_fuzzied=PM)||(Disp_vel_fuzzied=PL))
                        Fuzzy_output=NM;
return (Fuzzy_output);
}
```

```
////////////////////////////////////////////////////////////////////////
// Locomotion Response function
//
// Input          : Motor_output.
// Output         : Nil.
// Summary        : The function calculates the motor responses and drives the motors as
//                  required.
//
////////////////////////////////////////////////////////////////////////
int Locomotion_Response ()

{
//float Left_motor_PID=0, Right_motor_PID=0;

// Left motor output calculations
// Desired input has not implemented under current project
Motor_output=Fuzzy_motor_output;

// Left motor PID calculations
Left_motor_PID=PID_Function(PID_output);

// Right motor output calculations
// Desired input has not implemented under current project
Motor_output=Fuzzy_motor_output;

// Right motor PID calculations
Right_motor_PID=PID_Function(PID_output);

// Simplified version of motor output programming due to no locomotion control
implemented at this time.

// Apply motor brakes
if (Motor_output=0)
Left_drive_outputl1=1;
Left_drive_outputl2=1;
Right_drive_outputl3=1;
Right_drive_outputl4=1;
Enable_drive_output=1;

// Backward motor direction
if (Motor_ouptut<0)
        if (Motor_ouptut=-30)
                Left_drive_outputl1=0;
                Left_drive_outputl2=1;
                Right_drive_outputl3=0;
                Right_drive_outputl4=1;
                Enable_drive_output=1;
// Time sequence for 30% duty cycle not implemented at this time.

        if (Motor_ouptut=-70)
```

```
                Left_drive_outputl1=0;
                Left_drive_outputl2=1;
                Right_drive_outputl3=0;
                Right_drive_outputl4=1;
                Enable_drive_output=1;
// Time sequence for 70% duty cycle not implemented at this time.

        if (Motor_ouptut=-100)
                Left_drive_outputl1=0;
                Left_drive_outputl2=1;
                Right_drive_outputl3=0;
                Right_drive_outputl4=1;
                Enable_drive_output=1;
// Time sequence for 100% duty cycle not implemented at this time.

// Forward motor direction
if (Motor_ouptut>0)
        if (Motor_ouptut=30)
                Left_drive_outputl1=1;
                Left_drive_outputl2=0;
                Right_drive_outputl3=1;
                Right_drive_outputl4=0;
                Enable_drive_output=1;
// Time sequence for 30% duty cycle not implemented at this time.

        if (Motor_ouptut=70)
                Left_drive_outputl1=1;
                Left_drive_outputl2=0;
                Right_drive_outputl3=1;
                Right_drive_outputl4=0;
                Enable_drive_output=1;
// Time sequence for 70% duty cycle not implemented at this time.

        if (Motor_ouptut=100)
                Left_drive_outputl1=1;
                Left_drive_outputl2=0;
                Right_drive_outputl3=1;
                Right_drive_outputl4=0;
                Enable_drive_output=1;
// Time sequence for 100% duty cycle not implemented at this time.

return (0);
}
```

```
////////////////////////////////////////////////////////////////////
// PID controller function
//
// Input          : Motor_output.
// Output         : PID_output.
// Summary        : The PID function has not been implemented at this time.
//
////////////////////////////////////////////////////////////////////
static float PID_Function (PID_output)

{
        PID_output=1;

// Motor_ouput used in calculations not implemented at this time.

{
// Proportional calculations not implemented at this time.

// Integral calculations not implemented at this time.

// Derivative calculations not implemented at this time.
}

return (0);
}
```

```
//////////////////////////////////////////////////////////////////
// Kalman Filter function
//
// Input         : Gyro_digital, Accel_digital.
// Output        : K_output.
//
// Summary       : The function performs the Kalman filtering computations.
//                 Code is provided by Simon (2001) "Kalman Filtering".
//                 This function does not contain original content at this time.
//
//////////////////////////////////////////////////////////////////
static float Kalman_Function(K_output)

{
Gyro_digital= ;
Accel_digital= ;

// A is an n by n matrix
// B is an n by m matrix
// C is an r by n matrix
// xhat is an n by 1 vector
// y is an r by 1 vector
// u is an m by 1 vector
// Sz is an r by r matrix
// Sw is an n by n matrix
// P is an n by n matrix
float AP[n][n];              // This is the matrix A*P
float CT[n][r];              // This is the matrix CT
float APCT[n][r];            // This is the matrix A*P*CT
float CP[r][n];              // This is the matrix C*P
float CPCT[r][r];            // This is the matrix C*P*CT
float CPCTSz[r][r];          // This is the matrix C*P*CT+Sz
float CPCTSzInv[r][r];       // This is the matrix (C*P*CT+Sz)-1
float K[n][r];               // This is the Kalman gain.
float Cxhat[r][1];           // This is the vector C*xhat
float yCxhat[r][1];          // This is the vector y-C*xhat
float KyCxhat[n][1];         // This is the vector K*(y-C*xhat)
float Axhat[n][1];           // This is the vector A*xhat
float Bu[n][1];              // This is the vector B*u
float AxhatBu[n][1];         // This is the vector A*xhat+B*u
float AT[n][n];              // This is the matrix AT
float APAT[n][n];            // This is the matrix A*P*AT
float APATSw[n][n];          // This is the matrix A*P*AT+Sw
float CPAT[r][n];            // This is the matrix C*P*AT
float SzInv[r][r];           // This is the matrix Sz-1
float APCTSzInv[n][r];             // This is the matrix A*P*CT*Sz-1
float APCTSzInvCPAT[n][n];         // This is the matrix A*P*CT*Sz-1*C*P*AT
```

```
// The following sequence of function calls compute the K matrix.
MatrixMultiply((float*)A, (float*)P, n, n, n, (float*)AP);
MatrixTranspose((float*)C, r, n, (float*)CT);
MatrixMultiply((float*)AP, (float*)CT, n, n, r, (float*)APCT);
MatrixMultiply((float*)C, (float*)P, r, n, n, (float*)CP);
MatrixMultiply((float*)CP, (float*)CT, r, n, r, (float*)CPCT);
MatrixAddition((float*)CPCT, (float*)Sz, r, r, (float*)CPCTSz);
MatrixInversion((float*)CPCTSz, r, (float*)CPCTSzInv);
MatrixMultiply((float*)APCT, (float*)CPCTSzInv, n, r, r, (float*)K);

// The following sequence of function calls updates the xhat vector.
MatrixMultiply((float*)C, (float*)xhat, r, n, 1, (float*)Cxhat);
MatrixSubtraction((float*)y, (float*)Cxhat, r, 1, (float*)yCxhat);
MatrixMultiply((float*)K, (float*)yCxhat, n, r, 1, (float*)KyCxhat);
MatrixMultiply((float*)A, (float*)xhat, n, n, 1, (float*)Axhat);
MatrixMultiply((float*)B, (float*)u, n, r, 1, (float*)Bu);
MatrixAddition((float*)Axhat, (float*)Bu, n, 1, (float*)AxhatBu);
MatrixAddition((float*)AxhatBu, (float*)KyCxhat, n, 1, (float*)xhat);

// The following sequence of function calls updates the P matrix.
MatrixTranspose((float*)A, n, n, (float*)AT);
MatrixMultiply((float*)AP, (float*)AT, n, n, n, (float*)APAT);
MatrixAddition((float*)APAT, (float*)Sw, n, n, (float*)APATSw);
MatrixTranspose((float*)APCT, n, r, (float*)CPAT);
MatrixInversion((float*)Sz, r, (float*)SzInv);
MatrixMultiply((float*)APCT, (float*)SzInv, n, r, r, (float*)APCTSzInv);
MatrixMultiply((float*)APCTSzInv, (float*)CPAT, n, r, n,
(float*)APCTSzInvCPAT);
MatrixSubtraction((float*)APATSw, (float*)APCTSzInvCPAT, n, n, (float*)P);
return (K_output);
}

// End of file
```