Autonomous Materials Handling Robot for Reconfigurable Manufacturing Systems

Louwrens J Butler

School of Mechanical Engineering

University of KwaZulu–Natal

Submitted in fulfillment of the requirement for the degree of

Master of Science in Engineering

June 2010

As the candidate's Supervisor I agree/do not agree to the submission of this dissertation.

Signed: .... Date: ....

Professor Glen Bright

Declaration

I, Louwrens Johannes Butler, declare that

i. The research reported in this dissertation, except otherwise indicated, is my original work.

ii. This dissertation has not been submitted for any other degree or examination at any other university.

iii. This dissertation does not contain other persons' data, pictures, graphs, or other information, unless specifically acknowledged as being sourced from other persons.

iv. This dissertation does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:

   (a) their words have been re-written but the general information attributed to them has been referenced;

   (b) where their exact words have been used, their writing has been placed inside quotation marks, and referenced.

v. Where I have reproduced a publication of which I am an author, co-author, or editor, I have indicated in detail which part of the publication was written by myself alone and have fully referenced such publications.

vi. This dissertation does not contain text, graphics, or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the dissertation and in the References sections.

Signed: .... Date: ....

Acknowledgements

First and foremost I would like to acknowledge the unconditional love, support, and understanding that my parents have given me, not only during this degree, but throughout my life and especially my educational career. Without them I could never have gotten to this point.

I would like to acknowledge the eternal motivation, and direction of my supervisor, Professor Glen Bright. His enthusiasm for engineering and for life is so contagious, every time I step out of his office I feel a renewed sense of direction and enthusiasm for what I am doing.

I would also like to acknowledge my friends and colleagues, Anthony Walker and Jared Padayachee. Ant for introducing me to the open source community and teaching me the ways of mechatronic engineering, I couldn't have asked for a better mentor. Jared for always having some profound and insightful advice at hand for any situation. Without these people this experience would not have been the same.

Abstract

The concept of mass producing custom products, though extremely beneficial to the commercial, and retail industries, does come with some limitations. One of these is the occurrence of bottlenecks in the materials handling systems associated with reconfigurable manufacturing systems tasked with achieving the goal of mass customisation manufacturing. This specific problem requires the development of an intervention system for rerouting parts and materials waiting in line, around bottlenecks and/or work flow disruptions, to alternative destinations. Mobile robots can be used for the resolution of bottlenecks, and similar disruptions in work flow, in these situations. Embedding autonomy into mobile robots in a manufacturing environment, releases the higher level production management systems from routing of parts and materials.

The principle of the inverted pendulum has recently become popular in mobile robotics applications, and is being implemented in research projects around the world. The use of this principle produces a two-wheeled mobile robot that is able to actively stabilise itself while in operation. The dissertation is focused on the research, design, assembly, testing and validation of a two-wheeled autonomous materials handling robot for application in reconfigurable manufacturing systems. This robot should be dynamically or statically stable during different phases of operation. The mechatronic engineering approach of system integration has been used in this project in order to produce a more reliable robotic system.

The application of the inverted pendulum principle requires that a suitable control strategy be formulated. It also necessetates the ues of sensors to track the state of the robot. Control engineering theory was used to develop an optimal control strategy that is robust enough to cope with varying payload characteristics. The Kalman filter is employed as state estimation measure to improve sensor data. For a mobile robot to be deemed autonomous, one of the requirements is that the robot should be able to navigate through its environment without colliding with obstacles in its path, and without human intervention. A navigation system has been designed, through field specific research, to enable this. The robot is also required to communicate with remote computers housing production management systems as well as with mobile robots that form part of the same materials handling system. Performance analysis and testing proves the feasibility of a mobile robot system.

Contents

List of Figures **x**

List of Tables **xi**

1 Introduction **1**

1.1 Mass Customisation Manufacturing 1

1.2 The Inverted Pendulum 4

1.3 The Mechatronic Engineering Approach 5

1.4 Problem Statement 6

1.5 Project Objectives 6

1.6 Research Publications 7

1.7 Chapter Summary 7

2 Relevant Research in Mobile Robotics **8**

2.1 Mobile Robotics in Materials Handling 8

2.2 The Inverted Pendulum in Mobile Robotics 10

2.3 Chapter Summary 12

3 Project Design Specifications **13**

3.1 Operating Environment Parameters 13

3.1.1 Payload Parameters 13

3.1.2 Manufacturing Environment 16

3.2 Design Parameters 17

3.3 Technical Design Specifications 18

3.3.1 Mechanical Hardware Specifications 19

3.3.2 Electronic Hardware and Software Specifications 20

3.4 Chapter Summary 21

4 Mechanical Hardware Architecture **22**
4.1 Mobility . . . 22
4.1.1 Drive System . . . 22
4.1.2 Static Stability System . . . 25
4.2 Payload Interface . . . 29
4.3 Space Frame Support Structure . . . 32
4.4 Mechanical Architecture Safety . . . 37
4.4.1 Failure Mode, Effects and Criticality Analysis . . . 37
4.4.2 FMECA Results . . . 38
4.5 Chapter Summary . . . 43

5 Electronic Hardware Components **44**
5.1 Onboard Computing Facilities . . . 44
5.1.1 Single Board Computer . . . 44
5.1.2 Microcontrollers . . . 46
5.2 Sensory Infrastructure . . . 47
5.2.1 Ultrasonic Rangefinders . . . 47
5.2.2 Wheel Encoders . . . 49
5.2.3 Dual-Axis Accelerometer . . . 50
5.2.4 Rate Gyroscope . . . 54
5.2.5 Strain Gauge Load Cells . . . 56
5.3 Communication System . . . 57
5.4 Actuation Systems . . . 58
5.5 Power Management . . . 60
5.6 Integration of Electronic Components . . . 61
5.7 Electronic System Safety . . . 65
5.8 Chapter Summary . . . 69

6 Software Architecture **70**
6.1 Data Fusion and State Estimation . . . 70
6.2 Control Architecture . . . 74
6.2.1 Operational Platform Control Sequence . . . 74
6.2.2 Static Stability System Control . . . 74
6.2.3 Payload Interface System Control . . . 76
6.2.4 Active Stability Control . . . 77
6.2.5 Motion Control . . . 92
6.3 Device Management . . . 94

6.4 Safety Considerations 97
6.5 Chapter Summary 99

7 System Evaluation and Discussion 100
7.1 Technologies and Concepts of the Future 100
7.2 Technical Findings 101
7.2.1 Design Considerations 101
7.2.2 System Level Considerations 102
7.2.3 Operational Safety 109
7.3 Chapter Summary 110

8 Conclusion and Future Research 111
8.1 Research Conclusion 111
8.2 Recommendations for Future Work 112

A Technical Fabrication Drawings 113

B Dynamics Calculations and Drive Motor Selection 139
B.1 Moment of Inertia for a Lumped Model 139
B.2 Tilt Moment Calculations 141
B.3 Acceleration as a Function of Tilt Angle 142
B.4 Motor Power Output 144

C Mechanical Design Calculations 146
C.1 Drive Shaft Design 146
C.2 Drive Shaft Support Bearings 148
C.3 Payload Interface Shaft Design 149
C.4 Payload Interface Supports Bearings 151

D Mathematical System Model and Simulation 152
D.1 Mathematical System Model Derivation 152
D.2 Mathematical Simulation Code 158
D.3 Simulation Block Diagram 161

E Control System Programming Code 162
E.1 Active Stability Control System based on LQR Design Method 162

References 176

List of Figures

1.1 Graphical representation of two classic inverted pendulum configurations. . 4
1.2 Visual representation of the complete mechatronic system [44]. 5

2.1 Computer simulated model of an inverted pendulum robot. 10
2.2 Self-balancing hobby robots (a) nBot [5], and (b) Bender [28]. 11
2.3 Segway Robotic Mobility Platform [31]. 12

4.1 Mounting configuration of platform drive system. 23
4.2 Platform mobility system in assembly phase. 24
4.3 Simplified model of a wheeled inverted pendulum with free body diagrams. 25
4.4 Static stability system with friction reducing ball transfer units. 28
4.5 Roller load bed payload interface system in assembly phase. 31
4.6 Mechanism of operation of payload manipulation. 32
4.7 Exploded view of a CAD model of the base support structure. 33
4.8 CAD model of the space frame support structure. 34
4.9 Exploded view of a CAD model of the main mechanical structure. 35
4.10 Final assembly of physical mechanical structure including electrical and electronic systems. 36

5.1 EPIC NANO-GX466 SBC with SATA Hard Disk Drive. 45
5.2 ATmega32-based microcontroller. 47
5.3 Devantech SRF02 ultrasonic rangefinders as mounted on the platform. . . . 48
5.4 Visual representation of ultrasonic rangefinder mounting orientation. 49
5.5 HEDS-5500 rotary encoder fastened to the rear of the drive shaft. 50
5.6 Accelerometer (right) and gyroscope (left) as mounted on the platform, with axes indicated in red. 51
5.7 10-bit digital accelerometer output at rest. 52
5.8 Illustration of the accelerometer calibration apparatus setup. 53

5.9 10-bit digital accelerometer output at rest. 55
5.10 Strain gauge load cells mounted at the corners of the roller bed payload interface. 56
5.11 Wireless communication between the various components within the manufacturing environment. 58
5.12 Connection of DC motor speed controllers. 59
5.13 Connection instructions for the Victor 885 speed controller [40]. 59
5.14 SD-50A-12 DC-to-DC voltage converter with connections illustrated as described. 60
5.15 SD-25A-5 DC-to-DC voltage converter with connections illustrated as described. 61
5.16 Graphical representation of the drive controller circuit. 62
5.17 Graphical representation of the ultrasonic rangefinder and static stability system controller circuit. 63
5.18 Graphical representation of the load cell and payload interface controller circuit. 64
5.19 Graphical representation of the integration of the three circuits discussed. . 64

6.1 Graphical representation of the Kalman filtering algorithm. 72
6.2 Platform tilt angle versus time illustrating the settling time of the Kalman filtering technique. 73
6.3 Graphical representation of the operation of the static stability system. . . 75
6.4 Flow chart illustrating the flow of information and commands in the payload interface system control loop. 77
6.5 Simulated displacement state response to a unit step input. 82
6.6 Simulated linear velocity state response. 83
6.7 Simulated tilt angle state response. 84
6.8 Simulated angular velocity state response. 85
6.9 Symbolic block diagram of physical control system implementation. 86
6.10 Test results for angular system states from the LQR based active stability controller. 88
6.11 Test results for linear system states from the LQR based active stability controller. 89
6.12 Test results for the angular system states from the PID active stability controller. 90
6.13 Test results for the linear system states from the PID active stability controller. 91

6.14 Illustration of the differential drive configuration. 93
6.15 Graphical representation of the deployment of the *Player* device management software. 95
6.16 Sequence of execution of the abstract *Player* drivers implemented on the platform. 96

7.1 Sequence of screenshots of *Gazebo* graphical simulation. 105
7.2 Platform carrying a payload in the dynamically stable mode. 106
7.3 Platform carrying a payload in the statically stable mode. 107
7.4 Platform unloading a payload at a manufacturing cell I/O port. 107
7.5 Part distribution and collection application for statically stable mode platform. 108
7.6 Heavy part transportation application for statically stable mode platform. . 109

B.1 Initial lumped model simplification. 139
B.2 Final equivalent lumped model simplification. 140
B.3 Tilt moment force diagram for torque calculations. 141
B.4 Free body diagrams of the two components of the platform model. 143
B.5 Dynamometer results for the NPC 41250 worm-geared DC motor. 145

C.1 Drive shaft force diagram. 146
C.2 Drive shaft bearing positioning. 148
C.3 Payload interface system shaft force diagram. 149
C.4 Payload interface system shaft force diagram. 151

D.1 Free body diagram of a DC motor including electrical characterstics. . . . 152
D.2 Free body diagram of a platform wheel with external forces. 154
D.3 Free body diagram of body of the platform with all external forces. 155
D.4 Functional block diagram used to simulate the LQR control system. . . . 161

List of Tables

3.1 Mechanical hardware specifications. 19
3.2 Electronic hardware and software specifications. 20

4.1 Critical evaluation of static stability system concepts. 27
4.2 Critical evaluation of static stability system concepts. 30
4.3 FMECA for the drive subsystem. 39
4.4 FMECA for the static stability subsystem. 40
4.5 FMECA for the payload interface subsystem. 41
4.6 FMECA for the support structure subsystem. 42

5.1 Data produced during the accelerometer calibration experiment. 53
5.2 FMECA for the drive controller circuit. 66
5.3 FMECA for the ultrasonic rangefinder and static stability system circuit. . 67
5.4 FMECA for the load cell and payload interface circuit. 68

B.1 Lumped mass model variables and values. 140
B.2 Tilt moment parameters and values. 141
B.3 Tilt moments and motor torque versus tilt angle. 142
B.4 Relevant platform parameter definitions and values. 142
B.5 Linear acceleration required for corresponding tilt angle. 143
B.6 Model parameters and power requirements. 144

C.1 Drive shaft parameters and exerted forces. 147
C.2 Summary of stresses occurring in the drive shafts. 148
C.3 Summary of bearing design calculations for drive shaft support bearings. . . 149
C.4 Payload interface shaft parameters. 150
C.5 Payload interface shaft stresses and deflection. 150
C.6 Payload interface shaft parameters and bearing design calculations. 151

Chapter 1

Introduction

Due to current economic conditions, the global manufacturing industry has been under great pressure to produce better products at lower prices, and at higher rates. The demand for custom products on a mass scale has become a reality in today's diverse global community. This rise in demand for custom products has given rise to the concept of *Mass Customisation Manufacturing* (MCM). The research documented in this dissertation is aimed at developing a mobile materials handling platform for assisting manufacturing systems of the future to produce custom products on mass production scale and at mass production efficiency. When referring to efficiency in this context, it refers to the production rate as compared to the production rate of equivalent manufacturing systems producing standard, or generic products on a mass production scale.

1.1 Mass Customisation Manufacturing

Within a manufacturing system various subsystems exist that make up the complete manufacturing system. For the purposes of this discussion the physical subsystems that make up the manufacturing system can be categorised according to the following functional groups:

- Storage and Retrieval,
- Materials Handling and Transporting,
- Materials Processing,
- Quality Assurance.

The Storage and Retrieval subsystem includes taking delivery of materials and parts from suppliers, storage in inventory, and dispatching of final products. The Materials Processing subsystem includes physical processing such as material removal processes as well as assembly of products. The Quality Assurance subsystem inspects parts and assemblies at various stages throughout the production process in order to ensure that the manufacturing system adheres to quality standards. The Materials Handling and Transporting subsystem is tasked with routing and transporting materials and parts between processing stations that make up the Materials Processing subsystem. It is also responsible for material transportation to and from the Storage and Retrieval subsystem.

Mass Customisation was first discussed in the realm of sociology as a means to satisfy niche markets [46]. The concept evolved to include micro-segmentation of mass markets, such as consumer markets, and unique product development [16]. A scientific definition of Mass Customisation was proposed by Frank Piller [38]. This definition reads as follows:

Mass Customisation: "A customer co-design process of products and services which meet the needs of each individual customer with regard to certain product features. All operations are performed within a fixed solution space, characterised by stable but still flexible and responsive processes. As a result the costs associated with customisation allow for a price level that does not imply a switch in an upper market segment" [38].

This definition requires that the design process involves a combined effort between the production team and the individual customer. This brings about the concept of *Design for Manufacture*, and more specifically, *Design for Mass Customisation*. In *Design for Manufacture*, the production team, which typically consists of designers, design engineers, production engineers, accountants, and management representatives, is tasked with designing products that possess features that simplify the manufacturing process. In *Design for Mass Customisation*, this task is extended to incorporating features that may be modified to the taste of individual customers, within reasonable limits that is.

These modifications are referred to as customer-induced variations in product configuration, and they affect the processing sequence and also processing requirements, and thus the work-flow within the specific manufacturing plant. The subsystems within the manufacturing plant that are required to accommodate these variations are the materials processing, and materials handling subsystems. A characteristic that directly affects the efficiency of the manufacturing plant is the physical plant layout. The matter is complicated when the customer-induced variations require the plant layout to be altered in order to maintain a competitive efficiency.

A proposed solution to the plant layout problem is the implementation of a *Reconfigurable Manufacturing System*. The concept of *Reconfigurable Manufacturing Systems*, or RMS, was well-defined by Mehrabi et al. [32]. This definition reads as follows:

Reconfigurable Manufacturing System: "A machining system which can be created by incorporating basic process modules - both hardware and software - that can be rearranged or replaced quickly and reliably. Reconfiguration allows adding, removing, or modifying specific process capabilities, controls, software, or machine structure to adjust production capacity in response to changing market demands or technologies. This type of system provides customized flexibility for a particular part-family, and will be open-ended, so that it can be improved, upgraded, and reconfigured, rather than replaced" [32].

The inherent agility of RMS makes it capable of reacting to the customer-induced variations in product configuration introduced by MCM. This definition also highlights the fact that RMS provides sufficient flexibility to accommodate customisation of products based on a particular part family [24]. RMS based on a modular reconfiguration strategy provides the necessary capability for the manufacturing system to react to variations in product configuration at a rate that can maintain a competitive efficiency, without unwanted redundancy in the system.

Despite the flexibility and agility of RMS, disruptions in the flow of parts and materials, such as bottlenecks, are still a reality that the incorporated Materials Handling and Transportation subsystem is required to absorb and resolve. Bottlenecks occur when multiple parts are scheduled to be processed at a single processing station at such a rate that the build up of queuing parts is greater than the processing rate. Situations such as this can be avoided through proper management of resources within the manufacturing plant. In manufacturing operations where frequent variation in work-flow are required, autonomous mobile materials handling platforms can be deployed, that are dedicated to the resolution of disruptions in work-flow such as bottlenecks. This way, the routing and scheduling of the primary mobile platforms do not have to be affected by these disruptions. More detail on mobile robotic platform based materials handling systems can be found in Section 2.1.

1.2 The Inverted Pendulum

The inverted pendulum may be defined as a standard pendulum that has been inverted to its unstable equilibrium position, with the centre of gravity positioned above a rotational support joint. The configurations of classic inverted pendulum test beds in control engineering are shown in Figure 1.1. Figure 1.1(a) shows an inverted pendulum with an actuated rotational joint at the base, and Figure 1.1(b) shows an inverted pendulum in the form of a mass-pole-cart configuration.

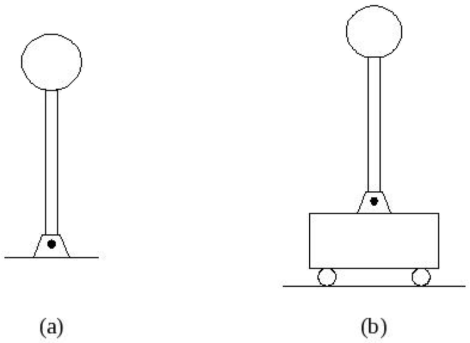


Figure 1.1: Graphical representation of two classic inverted pendulum configurations.

In the case of Figure 1.1(a) a torsional force is applied to the rotational joint in order to maintain the upright position of the pendulum. In the case of Figure 1.1(b) a horizontal force is applied to the cart in order to maintain the upright position of the pendulum, and the rotational joint between the cart and the pole is allowed to rotate freely. Ideally, in both these configurations the weight of the pole is negligible, thus the weight of the pendulum is concentrated at the end of the pole.

Due to the inherent instability of the pendulum in its inverted orientation it is well known in the field of control engineering as a benchmark for testing control algorithms. This problem has been approached from various points of view, and with different goals, from testing newly developed control algorithms to hobby robotics and electronics projects, to research projects in mobile robotics. More detail on the inverted pendulum in mobile robotics can be found in Section 2.2.

1.3 The Mechatronic Engineering Approach

In 1969, Tetsuro Mori, a senior engineer at a Japanese engineering company, coined the term Mechatronics. The field of mechatronic engineering has developed and evolved into an engineering discipline in its own right since its conception in the late 1960's. One definition used for mechatronic engineering can be found in Bolton [8]. This definition may be worded as follows:

Mechatronics: The complete integration across the traditional boundaries of microprocessor control systems, electrical systems, and mechanical systems. This integration has to occur at the earliest stages of the design process in order to develop cheaper, more reliable, and more flexible systems [8]. Figure 1.2 shows a Venn diagram of the complete integrated mechatronic system.

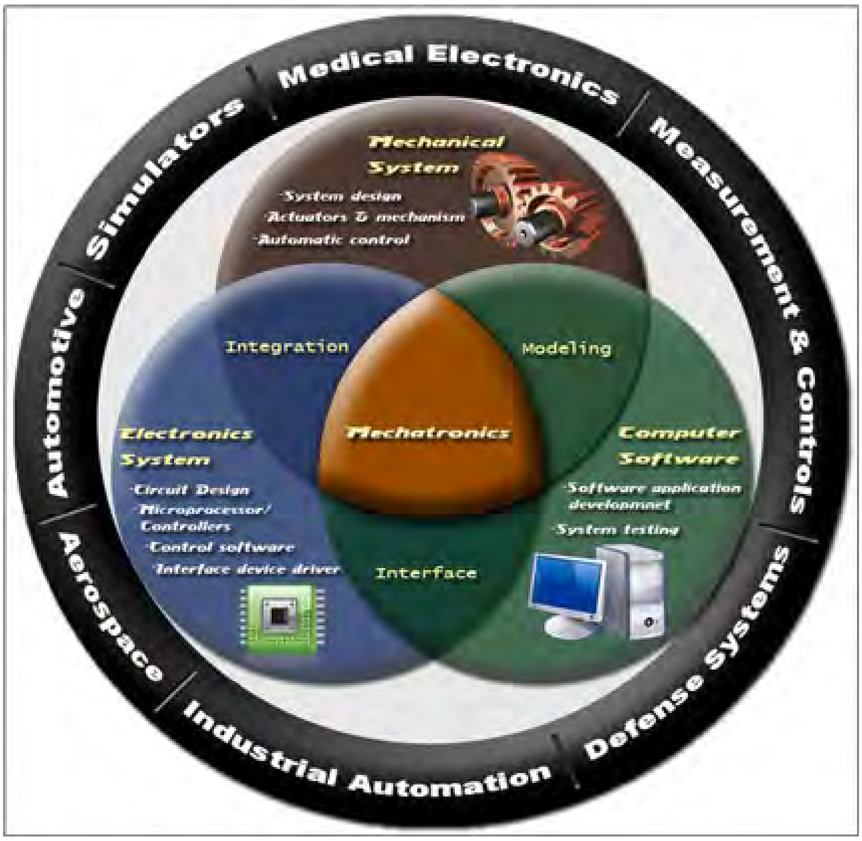


Figure 1.2: Visual representation of the complete mechatronic system [44].

The mechatronic engineering approach is being implemented in a wide range of industries from medical fields, to defense systems, to industrial automation, and also consumer and retail products. The rapid deployment of the mechatronic approach has been accelerated by vast development in the electronic hardware industry. These advances in the electronic hardware industry has enabled manufacturers to embed more powerful electronics into their products. In the context of this research, these advances allowed for more intelligence to be embedded onboard the platform, which in turn allows for more distributed control strategies within the manufacturing system.

1.4 Problem Statement

Reconfigurable Manufacturing Systems provide environments for mass producing custom products efficiently. However, potential bottlenecks, associated with customer-induced product variations, are a major concern for maintaining competitive production rates in reconfigurable manufacturing environments. A mobile materials handling platform is to be developed for the specific task of resolving, and/or preventing disruptions in workflow, such as bottlenecks, in a reconfigurable manufacturing system. The platform is to be capable of autonomously navigating through a reconfigurable manufacturing environment, while actively stabilising itself on two wheels, in a differential drive configuration, during certain stages of operation.

1.5 Project Objectives

The objectives of this research include:

1. To research mobile robotics in materials handling systems for MCM by RMS and the application of the inverted pendulum principle in mobile robotics.
2. To research and design a two-wheeled autonomous materials handling robot for RMS, through the use of the mechatronic engineering approach of system integration. This involves the design and integration of the mechanical hardware, electronic hardware, and the control and software systems.
3. To fabricate and assemble a prototype robot platform for experimentation and testing purposes. This entails the physical integration of the hardware systems.
4. To test and validate the performance and feasibility of a two-wheeled autonomous materials handling robot for RMS.

1.6 Research Publications

1. L. Butler and G. Bright. "Autonomous Materials Handling Robot for Reconfigurable Manufacturing Systems". In *Proceedings of the 24th International Conference on CAD/CAM, Automation, Robotics, and Factories of the Future*. Koriyama, Japan. July 2008.

2. L. Butler and G. Bright. "Autonomous Mobile Robot for Advanced Materials Handling Systems". In *Proceedings of the 2nd Mechatronics and Robotics Symposium*. Bloemfontein, South Africa. November 2008.

3. A. Walker, L. Butler, G. Bright, N. Tlale and C. Kumile. "Mobile Materials Handling Platform Interface Architecture for Mass Production Environments". In *Proceedings of the 15th International Conference on Mechatronics and Machine Vision in Practice*. Auckland, New Zealand. December 2008.

4. L.J. Butler and G. Bright. "Feedback Control of a Self-balancing Materials Handling Robot". In *Proceedings of the 10th International Conference on Control, Automation, Robotics, and Vision*. Hanoi, Vietnam. December 2008.

5. L.J. Butler and G. Bright. Autonomous Materials Handling Robot for Advanced Manufacturing Applications. *Journal of KONBiN*, 8(1):17-26, 2008.

6. L.J. Butler and G. Bright. "Sensory Infrastructure for an Autonomous Self-balancing Mobile Materials Handling Platform". In *Proceedings of the 3rd Mechatronics and Robotics Symposium*. Pretoria, South Africa. November 2009.

7. L.J. Butler and G. Bright. Control Strategy for a Mobile Self-balancing Materials Handling Platform. *Journal of Engineering, Design and Technology*, 8(1). 2010.

1.7 Chapter Summary

This chapter supplies some introductory information on the background that the project has developed from, in order to familiarise the reader with the project. The problem statement is given to encapsulate the bounds of the project problem space. The project objectives are listed and may be referenced throughout this document. Also listed in this chapter are all publications produced by the author through the course of this project. This includes international and domestic conference proceedings and international journal publications.

Chapter 2

Relevant Research in Mobile Robotics

Mobile robotics is an area of research that has intrigued scientists since the late 1940s, with the conception of *Machina Speculatrix*, two tortoise-like robots capable of autonomous exploration, developed by Dr. W. Grey Walter [37]. Modern mobile robots may be deployed in vastly different situations and environments, for performing a variety of different tasks. For the purpose of this project, this document specifically discusses mobile robotics in materials handling. The application of the inverted pendulum principle in mobile robotics is also discussed here.

2.1 Mobile Robotics in Materials Handling

Automated Guided Vehicle (AGV) systems are most frequently implemented in manufacturing environments where they are required to operate alongside humans. In this kind of situation autonomy is vital, not only for the effective operation of the system, but also for the safety of the human "co-workers" [41]. Work has been done to develop AGV systems that impart more flexibility to the materials handling system, through embedding semi-autonomous capabilities into the platforms [34]. Although this AGV system still relied on line following for navigation, the platforms were able to cope with varying payload characteristics and plant layout. The main improvement in the design of the AGV system of [34] is a move to a decentralised control scheme, as opposed to a centralised control strategy employed by traditional AGV systems.

A departure from the traditional and modern AGV system is the implementation of a team of autonomous mobile robots tasked with performing all materials handling within a manufacturing plant. As the supporting technologies have developed, more and more intelligence may be embedded into mobile robots, and thus improving their autonomous capabilities. Essentially more intelligence on a mobile robot opens the possibility to distribute more decision-making capability, to the platforms.

Autonomy can be defined, within a rational behaviour, by the effectiveness and robustness of a robot in carrying out tasks in different and ill-known environments [2]. An autonomous robotic platform is thus capable of gathering information concerning its position and orientation from its environment, and making decisions based on this information. Modern autonomous mobile robots are capable of *Simultaneous Localisation And Mapping* (SLAM) [29]. This is the capability to build a map of its environment and keeping track of its coordinates relative to an external reference frame [45], while moving in the environment. This feature is, however, not a prerequisite to autonomous operation within a known environment.

Manufacturing plants in pursuit of producing custom products on a mass production scale require a materials handling system that is flexible enough to accommodate rapid and frequent variations in flow of parts and materials. Conventional materials handling systems based on rigid conveyor systems linking processing station in a fixed sequence are incapable of accommodating this. A materials handling system based on a team of autonomous mobile robotic platforms is specifically capable of facilitating MCM.

A materials handling system that is based on a team of autonomous mobile robotic platforms may be categorised as a *Flexible Materials Handling System*, due to its intrinsic ability to accommodate rapid and frequent changes in work-flow [27]. In such a materials handling system the key aspects that determine its successful implementation are the routing of platforms, and scheduling of tasks. If the materials handling management system, which is a software system that is integrated into the plant management system, is capable of managing these aspects effectively, the manufacturing system has an improved chance of achieving the targeted production rate and efficiency.

Materials handling specific platform architectures have been developed in order to ensure that transportation of parts and materials occurs with as little constraints as possible and that cooperation between platforms is optimised. Bright and Walker propose a materials handling platform architecture based on a platform base possessing nonholonomic motion constraints, with added degrees of freedom in the payload interface components, producing platforms with essentially omni-directional motion constraints [9]. An approach using the mechatronic engineering principle of system integration was employed in the development of this platform architecture.

For the purpose of this research a fully automated reconfigurable manufacturing system in pursuit of MCM, with a mobile robotic platform based materials handling system, is proposed as the environment for implementation of the product of the project. Thus duplicates of this robotic platform may be deployed to assist the primary materials handling platforms in defending competitive production rates, through the resolution and/or prevention of disruptions in work-flow such as bottlenecks.

2.2 The Inverted Pendulum in Mobile Robotics

The product of applying the principle of the inverted pendulum to the field of mobile robotics is a mobile robot that is able to balance itself in an upright orientation through active stabilisation. The most common configuration of such a robot is a two-wheeled mobile robot with its centre of gravity above the wheel axles. The term active stabilisation refers to the fact that the wheels are driven by motors in order to maintain the upright orientation of the robot. Figure 2.1 shows a computer simulated model representation of this configuration.

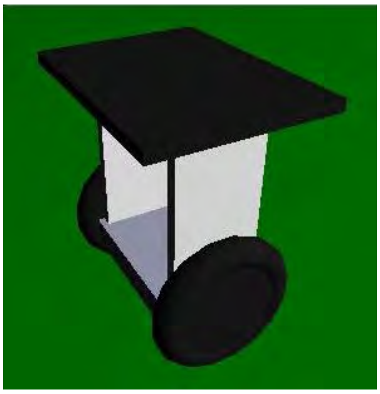


Figure 2.1: Computer simulated model of an inverted pendulum robot.

Advantages of two-wheeled mobile robots in contrast to statically stable robots, in other words robots with more than two support points, include:

- Less contact with the floor surface and thus less mechanical losses,
- zero turning circle - A two-wheeled robot is able to turn in position.

Disadvantages of two-wheeled robots compared to statically stable robots include:

- Less energy efficient, due to the fact that the wheels need to be driven at all times in order to maintain an upright position,
- vulnerable to external disturbances such as obstacles in its path.

Projects in mobile robotics involving self-balancing robots have been undertaken from many different perspectives; from university projects, to hobby projects, to US government funded programmes. University projects have been found to aim primarily at testing different control strategies and algorithms. Amongst others, the basic Proportional Integral Derivative (PID) control approach [12], a Proportional Integral Sliding Mode control approach [1], and a Linear Quadratic Regulator control approach [36], have all been implemented with varying levels of success. The robots used, in the university projects mentioned, were all unmanned vehicles.

Some well-documented hobby projects of prototype two-wheeled balancing human transporters have been done, such as the “Balancing Scooter” developed by Trevor Blackwell [5]. These hobby roboticists attempt to copy the invention of Dean Kamen, known as the Segway Human Transporter [30]. Other hobby roboticists have also developed self-balancing robots. Some well-known robots include David Anderson’s “nBot” [5] and Ted Larson’s “Bender” [28]. Figure 2.2 shows these two robots in operation.

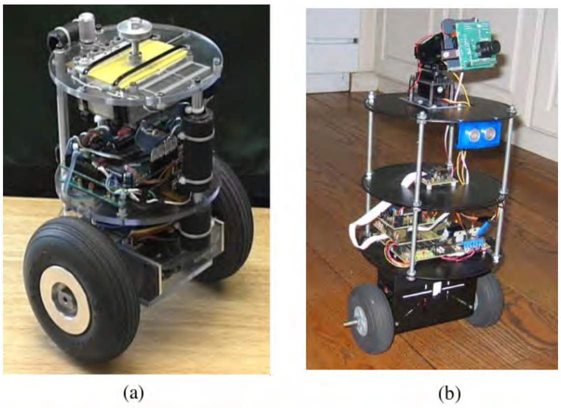


Figure 2.2: Self-balancing hobby robots (a) nBot [5], and (b) Bender [28].

Subsequent to the invention of the Segway Human Transporter, Segway has developed and produced an unmanned self-balancing robot, the Segway Robotic Mobility Platform, or RMP [31]. The RMP100 and –200 are self-balancing robotic platforms produced for research in mobile robotics as well as commercial applications. The first generation Segway RMP has been employed in research projects funded by the US government. These projects ranged from humanoid robot mobility to military search-and-rescue applications [35]. The latest version of the Segway RMP 100 can be seen in Figure 2.3.

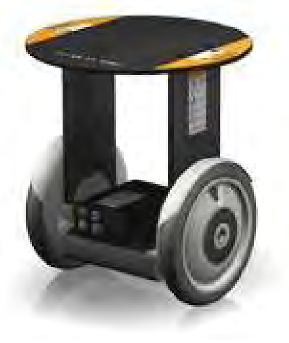


Figure 2.3: Segway Robotic Mobility Platform [31].

2.3 Chapter Summary

This chapter serves to provide information on mobile robotics from the perspective of materials handling, where this research is applied, and from the perspective of the physics that is being applied in this project. It is important to note that, to the knowledge of the author, a self-balancing robot had never been used for materials handling purposes before this project was conceptualised. The autonomous mobile robot in itself is quite a new concept in materials handling applications. This is one of the motivating factors for this research. This chapter also highlights literature that is relevant to the topic and the background of the research.

Chapter 3

Project Design Specifications

Well developed engineering design specifications ensure that the final design adheres to the requirements identified at the outset of the project. The engineering design specifications identified for this project have been categorised according to the general subsystems that make up a mechatronic system.

3.1 Operating Environment Parameters

Final design specifications for the platform are determined by a number of factors in terms of the interaction of the platform with its environment.The interaction with the manufacturing plant infrastructure as well as the interaction with the parts and materials to be transported. Before specifications for the design of the platform could be identified, it is required to accurately define the environment that the platform is to interact with.

3.1.1 Payload Parameters

The practice of *Group Technology* is applied, in the manufacturing industry, to reduce work-in-progress and lead times. *Group Technology* is the practice of grouping parts together in terms of similarities such as geometry or manufacturing processes. These groups are often called part families or product families, when referring to the product level [25]. Part families such as these can be processed by machine cells or processing cells when looking at the entire production process, this is referred to as cellular manufacturing.

Advantages of *Group Technology* include reduction of engineering cost, acceleration of product development, simplification of process planning, reduction of tooling cost, and improvement of costing accuracy [21]. Developing a product family architecture involves some consideration in terms of functional, behavioural and structural perspectives [24]. Here only the structural perspective is considered, as this project has only to do with the physical manufacturing and plant level logistical aspects. These have been developed further into categories in terms of part/material characteristics.

Materials

The assumption has been made that this mobile robot will only be used in the metal machining industry. Thus only metal parts will be transported by the mobile robot. The different metals with their respective densities are listed below.

- Copper @ 8920 kg/m^3,
- Stainless steel @ 8000 kg/m^3,
- Mild steel @ 7850 kg/m^3,
- Cast Iron @ 7000 kg/m^3,
- Aluminium @ 2700 kg/m^3.

Alloys of these metals that fall within the minimum and maximum densities listed above could also be transported, as long as the weight limit as proposed in this section, is not exceeded.

Geometries

A set of basic families of geometries have been defined as allowable for loading onto the mobile robot, for the design of the prototype robot. If this robot is applied in the industry this set of geometries will be replaced by the database of parts that will be transported by the robot, in order for it to compensate correctly for each payload being loaded. The robot will thus know which part is going to be loaded before it is loaded. The proposed geometries are as follows:

- prisms (excluding cylinders),
- pyramids (excluding cones),
- I-profiles, such as I-beam extrusions,
- U-profiles, such as channel extrusions,
- L-profiles, such as angle extrusions,

- flat plates.

The limitation on the transportation of cylinders and cones is that they will have to be mounted on a jig before loaded on the platform load bed in order to stop these parts from rolling off the load bed. These are basic categories of the different shapes that the prototype robotic platform is designed to transport.

Maximum weight

It is proposed that the maximum weight that the mobile robot is designed to carry is 20 kg. The combination of this maximum weight and the material as stated above gives a good range of volumes for testing capabilities of the mobile platform. This weight limit has also been set in order to add structure to the design process of the mobile platform.

Maximum dimensions

From the weight limit and the proposed working materials, the extreme cases in terms of volumes can be calculated, by the formula: $Volume = Mass/Density$. Using the most dense metal with density of 8920 kg/m^3 and substituting the mass of 20 kg into this equation a volume of 0.00224 m^3 is found. This gives a cube with sides of 131 mm. Using the least dense metal with density of 2700 kg/m^3 and substituting the mass of 20 kg into this equation a volume of 0.00741 m^3 is found. This gives a cube with sides of 195 mm.

In order to set the physical size of the mobile robot it is necessary to know the maximum size of the parts that are to be transported. With this in mind it is proposed that the maximum dimension of a payload must be 300 mm. In other words the payload that is being transported may not exceed 300 mm in any dimension.

Overall size

Having set the parameters as discussed up to now, other dimensions may be calculated. As before from the densities and weight, volume can be calculated. From the volume and maximum dimensions other dimensions can be calculated. The following calculations were done using the least dense material, which is Aluminium. For a square prism:

$$Vol = 0.00741\ m^3$$

$$l^2h = 0.00741\ m^3$$

$$l^2 0.3 = 0.00741\ m^3$$

$$l = 157\ mm$$

This produces a square prism shape with sides of 157 mm and height of 300 mm. For a cylinder:

$$Vol = 0.00741\ m^3$$

$$\pi r^2 h = 0.00741\ m^3$$

$$\pi r^2 0.3 = 0.00741\ m^3$$

$$r = 88.7\ mm$$

This produces a cylinder with diameter of 177.4 mm and height of 300 mm. A part that is 300 mm square would be longest in the diagonal direction. According to Pythagoras, the diagonal dimension will be: $Diagonal = \sqrt{2 * 300^2}$. This produces a maximum allowable overall dimension of 424.3 mm, for any part to be loaded onto the platform.

3.1.2 Manufacturing Environment

For the purpose of applying this research in practice a hypothetical cellular reconfigurable manufacturing plant is defined for the application of this project. This manufacturing plant will be used for mass producing customised products.

In order to include as wide a range of industries in the application, without losing focus, it has been defined that the manufacturing plant will only process metal. This industry has been chosen as it is one where automation is relatively easily realised, by the application of *Computer Numerical Control* (CNC) machines [32]. The defined manufacturing plant is capable of performing the following processes:

- Turning/knurling
- Sawing
- Engraving
- Drilling
- Milling
- Grinding
- Tapping
- Polishing/buffing

This cellular manufacturing plant is serviced by a reconfigurable materials handling system, based on autonomous mobile materials handling platforms. This system consists of a team of statically stable autonomous mobile materials handling platforms for transporting materials between the input and output ports of the different machine cells in the manufacturing plant.

This project is involved in designing a mobile robot that will be implemented to resolve bottlenecks due to mass customisation of the work-in-progress in this manufacturing plant. The robot complements the primary system of autonomous mobile materials handling platforms. Due to the reconfigurable nature of the operating environment and the inherent uncertainties in the payload characteristics a degree of intelligence is required to be embedded on the platform. Design parameters arising from this discussion are listed in the next section.

3.2 Design Parameters

Parameters that have been discussed up to this point are imposed on parts that are to be loaded onto the load bed of the platform. This is done in order to develop specifications on the size of the platform load bed as well as the payload capacity of the platform. Design parameters that will be used for developing technical specifications for the electronic hardware and software subsystems are listed below along with design parameters for the mechanical subsystem specifications.

The platform should:

1. Transport payloads from origin to destination within the proposed operating environment, quickly and efficiently.
2. Be dynamically stable throughout the transportation phase.
3. Be able to navigate in a reasonably dynamic, yet mapped environment.
4. Be able to communicate with a server computer and/or other mobile materials handling platforms.
5. Be able to accommodate different geometries of payloads to be loaded.
6. Be able to react to bottlenecks as it is instructed by manufacturing system management software.

7. Interface effectively with loading and unloading stations.
8. Have knowledge of possible parts to be loaded.
9. Be able to transport parts with a set maximum weight.
10. Be able to transport parts with a set maximum diagonal dimension.
11. Be statically stable during certain phases of operation.
12. Be as energy efficient as possible.
13. Not pose any danger, directly or indirectly, to any person working in close vicinity to it.

These parameters have application to the different mechatronic subsystems as discussed in the next section. The design parameter that each technical specification originated from will be stated.

3.3 Technical Design Specifications

Technical engineering design specifications have been developed from the requirements identified in the design parameters listed above. These are discussed here in terms of the mechatronic subsystem that they apply to.

3.3.1 Mechanical Hardware Specifications

Specifications listed in Table 3.1 pertain to the design of the physical structure and actuation of the platform. These specifications need to be met by the final design of the platform in order for it to be successful.

Originating Design Parameter	**Technical Specification**
1, 2	Two driven wheels. Large wheels, ± 500 mm in diameter, with sufficient traction. High centre of gravity. Maximum allowable tilt angle of 5°. Maximum linear velocity of 1,5 m/s.
5	Determine horizontal position of payload centre of gravity. Move payload to align centre of gravity with that of the platform.
7	Platform load bed must be at the same height as the machine cell input/output ports, i.e. 800 mm. Use of mechanism (onboard or not) for transferring payloads to and from payload.
10	Must be able to support a payload of up to 20 kg throughout operation.
11	Load bed set to dimensions of 475 x 430 mm in order to accommodate a part with maximum dimension of 424 mm (from calculations in Section 3.1.1).
13	Design all mechanisms for safe operation.

Table 3.1: Mechanical hardware specifications.

3.3.2 Electronic Hardware and Software Specifications

Specifications listed in Table 3.2 pertain to the design of the electronic and software subsystems of the platform. These specifications have been categorised together due to the close interaction between these two subsystems.These specifications need to be met by the final design of the platform in order for it to be successful.

Originating Design Parameter	**Technical Specification**
2	Control system must have a robust enough response to actively balance the platform while in operation. As little as possible overshoot for smooth operation.
3	Must know the structure and layout of its environment. Must be able to avoid static and dynamic obstacles.
4	Must be able to communicate wirelessly with a server and with other platforms in the vicinity.
5	Must be able to determine where the centre of gravity of the payload is located (in terms of sensor data processing). Sense when the payload centre of gravity is in the correct position for the platform to balance itself effectively.
6	Must know which actions to take to resolve/prevent any bottlenecks at the various machine cells.
8	Receive information from server about the part that is about to be loaded.
12	Motion planning and lower level control systems must consider energy expenditure.
13	Must incorporate failure sequences for different modes of failure.

Table 3.2: Electronic hardware and software specifications.

3.4 Chapter Summary

In this chapter the technical engineering design specifications for the design aspect of this project are developed. From the start point of setting boundaries on payload characteristics and defining the proposed operating environment, design parameters are identified and technical specifications are developed from these parameters. For practicality the technical design specifications are categorised in terms of the mechatronic subsystems.

Chapter 4

Mechanical Hardware Architecture

The mechanical hardware architecture of the robotic platform is characterised by three central goals that the platform is required to achieve, as identified through the development of the design specifications. These include mobility, payload interface, and structural support and integrity.

4.1 Mobility

The first goal is to achieve mobility of the platform. The drive system that has been implemented is discussed in detail. A system, complementary to the primary drive system, has been implemented to achieve static stability, required for certain phases of operation.

4.1.1 Drive System

The two-wheel drive configuration of the platform constrains the trajectory of the platform to be determined by differential drive kinematics. A differential drive system consists of two powered wheels, the relative angular velocities of which determine the translational and rotational motion of the platform within a global Cartesian coordinate system. A differentially driven mobile robot is usually statically supported by one or more additional low friction supports, however in this case for the majority of the time in operation the only contact with the floor surface is through the two driven wheels. Retractable stabilisers are used for static stability during operation, when this is absolutely necessary. The static stability system is discussed in Section 4.1.2. More detail on the kinematics of the platform can be found in Section 6.2.5

Achieving the differential drive configuration involves two wheels, each mounted on a shaft orientated parallel and in line with each other. Each of these shafts is coupled to a *Direct Current* (DC) motor. The coupling between wheel and shaft is achieved by a rectangular key, and the coupling between shaft and motor is by way of a collar joined to the motor output shaft using a rectangular key, while the coupling drive shaft fits tightly inside the other end of the collar. A machine screw through the centres of both prevents these components from rotating independently from each other.

Each drive shaft is mounted to the frame of the platform by two ball bearing pillow block units. The bearings housed in the pillow blocks are self-aligning, and have inner diameters of 20 mm. The nominal diameter of the drive shafts is 20 mm, they are stepped up to 23 mm to house the motor output shaft, and stepped down to 16 mm at the other end for mounting the wheels. The drive shafts have been machined from 25 mm mild steel round bar. Figure 4.1 shows mounting configuration used for the platform drive system. Refer to Appendix A for the technical drawings produced for fabrication and assembly of the physical platform architecture.

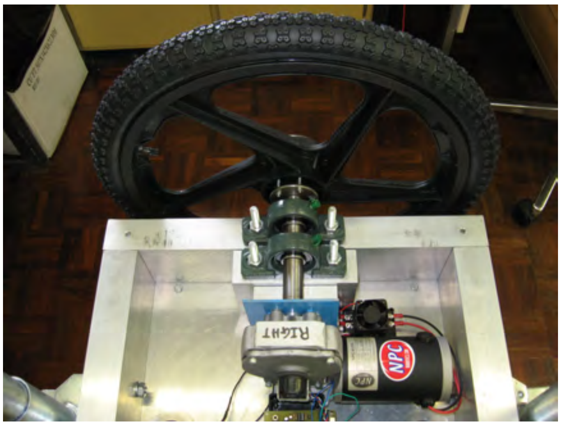


Figure 4.1: Mounting configuration of platform drive system.

Each drive wheel consists of a rim moulded out of a thermoplastic polymer with a diameter suitable for fitting a rubber tire with outside diameter of 20 inches, or approximately 508 mm. The hub of each wheel has been modified, for mounting on to the drive shafts, by press fitting two identical bushes from either side with key ways aligned in the process. The inner geometry of the hub prevented the use of a single, solid bush. Figure 4.2 show the mobility system in the assembly phase. Here the platform base is upside down while in the process of mounting the motors and sensors to the base and preliminary testing of the drive mechanism operation and control system.

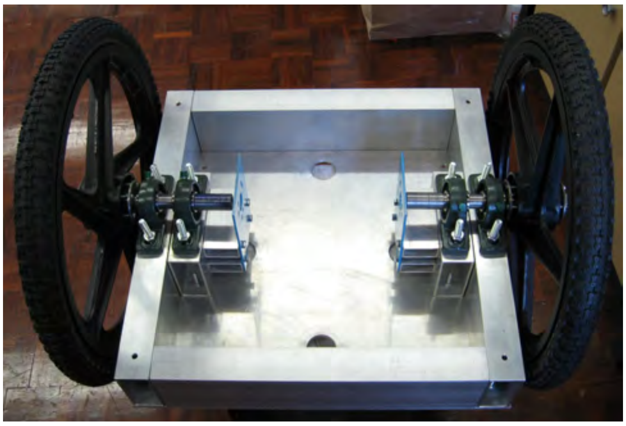


Figure 4.2: Platform mobility system in assembly phase.

Each of the DC motors is mounted to a 3 mm mild steel plate with three cap screws. The steel plate is bolted to the platform frame through a combination Aluminium extrusion profiles. Each wheel is held in place on the drive shaft by an M16 hexagonal nut tightened on to the end of the shaft after the wheel is fitted. The drive train that delivers power to the wheels has been kept as simple as possible due to the fact that a high accuracy is required for the position and velocity of the wheels for the control of the stability of the platform as well as the motion control of the platform. For this reason it was imperative that a motor with a suitable power and speed output was to be specified and sourced. The DC motors that actuate the drive wheels are rated to deliver a approximately 150 W of power. The outputs of these motors are geared down through worm gears and rotate at approximately 174 rpm at 24 V input voltage at no-load.

Basic mechanics calculations, based on theory found in Meriam and Kraige [33], applied to a simplified wheeled inverted pendulum model produces a power requirement of 105 W per drive motor. With the chosen motor this gives a reserve factor of approximately 1,43. This is acceptable due to the fact that losses were not taken into account in the power calculations. For these calculations a maximum platform mass of 60 kg was estimated, and the specifications on maximum acceleration and maximum velocity from Table 3.1 were used to calculate a peak value for the required power. These calculations may be found in Appendix B. Figure 4.3 shows the inverted pendulum model that the calculations are based on, including the free body diagrams.

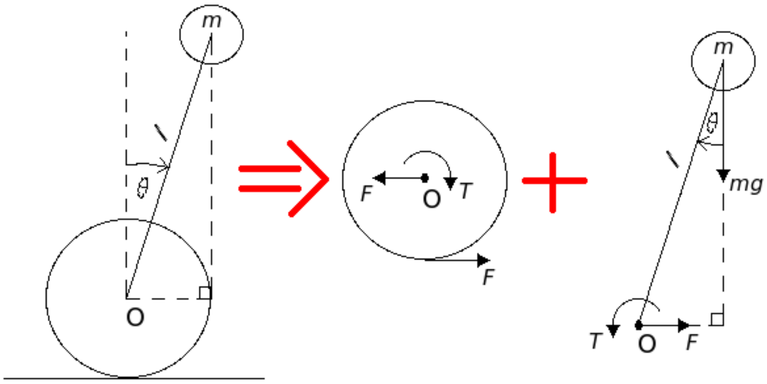


Figure 4.3: Simplified model of a wheeled inverted pendulum with free body diagrams.

From the calculations mentioned above and the specifications on the dynamics of the platform it was found that the maximum allowable linear acceleration that corresponds to the maximum allowable angle of tilt is approximately 2 m/s^2. This produces a time of 0.75 s from stand still to maximum velocity, however this does not take into account the dynamics that are associated with the inverted pendulum.

4.1.2 Static Stability System

Although the static stability system forms part of the mechanical architecture of the platform it only performs a complementary function in the realisation of mobility in the platform, where the primary mobility hardware is described in Section 4.1.1. The functionality of the static stability system only comes into play when the platform is interfacing with an Input/Output (I/O) port of one of the machine cells in the manufacturing plant. The motivation for implementing a static stability system is that the effective interaction between the platform and its infrastructure at this level is crucial for its overall efficiency. It is also for this reason that the design process undertaken for this system was thoroughly considered and is documented in this section.

The first phase in the process of designing a system that realises static stability of the platform effectively is the conceptual design phase. Four concepts that may be able to solve the problem are identified and described here. These concepts are then compared and weighed up against each other. The optimal solution for the problem was chosen according to the findings of this comparison.

Concept SS.1: Galileo Wheels

These wheels were developed by a company called Galileo Mobility Instruments. This company has patented this technology and it is thus illegal to apply it in this project, without agreement of the owners of the technology. It will still be discussed here as a hypothetical solution to the problem at hand. According to Galileo Mobility Instruments [23]: "The Galileo Wheel is a unique robust construction that allows automatic back and forth conversion among a wheel configuration, a track configuration and a special stair climbing configuration." [23].

Concept SS.2: Retractable Stabilisers

This concept involves retractable supports that can be deployed while loading is in progress. The supports will be deployed when the robot is in range for loading to commence. These supports may be fitted with wheels for small adjustments in the loading process.

Concept SS.3: Electromagnets

This concept does not involve any additional contact with the floor surface excepting the two driven wheels of the platform. In this concept the platform is fitted with an electromagnet on each side, front and rear. Whichever magnet comes in contact with an I/O port will be activated, so that it connects to the port. The I/O ports will have to be fitted with metal blocks in order for the connection to be made effectively.

Concept SS.4: Tracked Wheels

This concept was included as a contingency for the solution of this problem, in case none of the above generated concepts were feasible. This concept specifies tracked wheels instead of normal round wheels. This, however, defeats the object of the project to implement a self-balancing mobile platform, due to the fact that tracked wheels make the platform permanently statically stable.

Critical Concept Evaluation

The concepts that have been generated were critically evaluated and their advantages and disadvantages were identified in terms of the specifications set in Chapter 3. Table 4.1

lists the advantages and disadvantages of each concept. The verdict of the critical concept evaluation is that the most applicable concept is the use of retractable supports, with the added feature of fitting some sort of friction reduction mechanism to the bottom of each support in order for the platform to be able to move while the supports are extended.

Concept	Advantages	Disadvantages
SS.1	Excellent stability at low speeds – when interfacing with I/O ports. Stability while moving.	A lot of space required when tracks are deployed – bulky design. More power is required when tracks are deployed while moving due to increased friction forces. Very intricate design.
SS.2	A practical design. Produces very good stability. No additional infrastructure required.	Requires a very flat surface for platform to be level. No movement is possible once supports are deployed (unless they are fitted with wheels).
SS.3	It is a very simple design.	Electromagnets may interfere with onboard electronics. Could produce insensitivity to misalignment of the payload. Requires additional infrastructure to be installed.
SS.4	Provides constant static stability, unless there is only a single connection between each track and the chassis of the platform.	Requires substantially more power to actuate due to increased friction forces. Produces a much more complicated drive system.

Table 4.1: Critical evaluation of static stability system concepts.

Final Design of Static Stability System

The final design of the static stability system can be seen in Figure 4.4. For the mechanism to reduce the friction between the supports and the floor surface a ball transfer units are fitted to the bottom of each support, or stabiliser. Ball transfer units have very good dynamic characteristics at low speeds, however they require a very smooth surface to operate effectively. In this application, the operating environment is a manufacturing plant. It is thus safe to assume that the floor is smooth and level. The ball transfer units are clearly visible in Figure 4.4.

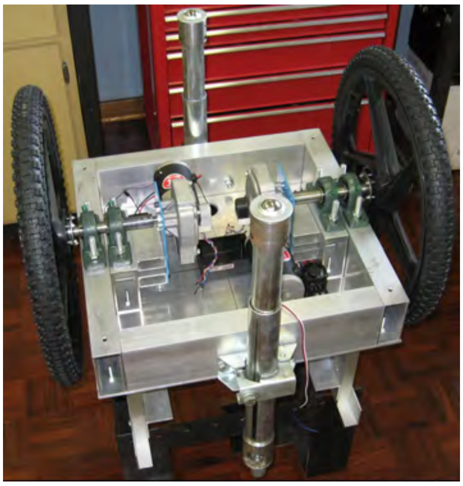


Figure 4.4: Static stability system with friction reducing ball transfer units.

The actuation of the stabilisers is realised through the use of lead screws. Each lead screw is driven by a small DC motor which extends or retracts the stabiliser. The control of the motor will be discussed in Section 6.2. The collar of the lead screw is housed in a metal cylinder to which the ball transfer unit is attached. The lead screw rotates inside a slightly larger metal cylinder. The collar cylinder moves up and down inside this larger cylinder. A pin is fixed to the inner cylinder that protruded through a slot in the outer cylinder. This pin is used to keep the inner cylinder from rotating along with the lead screw. One lead screw assembly is clamped in position onto the front and rear of the platform base, respectively.

4.2 Payload Interface

The interface between the payload and the platform is another crucial aspect of the design of the platform. Careful consideration has been made in the generation and selection of the optimal concept for this system. The design of this aspect of the mechanical hardware architecture has been somewhat simplified by establishing the limits for payload characteristics in Section 3.1.1.

The same process that is outlined in Section 4.1.2 is used here to identify the optimal solution to the payload interface problem. The concepts discussed in this section were generated to address the problem of positioning the payload such that its centre of gravity is in line with the centre of gravity of the platform. This must be done in order for the robot to balance itself effectively. Considerations made here were for weight, ease of interfacing, accuracy of movement, and degree of difficulty in realising the design.Concepts generated for this system are described in the following four sections.

Concept PI.1: Rotating Load Bed

This concept specifies the load bed to be able to rotate about the centre vertical axis of the platform. This is in order to position the centre of gravity of the payload in line with that of the platform. This concept assumes that the position of the payload, in the direction parallel to the wheel axles does not affect the ability of the platform to balance itself.

Concept PI.2: Roller Load Bed

In this concept there are two rows of rollers adjacent to one another. These rows are actuated independently from each other. Thus they will be able to run in opposite directions, and different speeds, at the same time. This configuration enables the load bed to rotate and translate the payload to the orientation where its centre of gravity is in position.

Concept PI.3: Mecanum Wheel Load Bed

This concept makes use of Mecanum wheels to manipulate the payload into position. The wheels are oriented in an upside down configuration. This configuration is similar to the platforms used in the air transport industry for loading luggage and freight containers into the cargo holds of airplanes. The mechanism of operation of these wheels enables the rotation and translation of the payload. The movement of the payload is determined by the relative direction and speed of rotation of the Mecanum wheels that it is in contact with.

Concept PI.4: Ball Transfer Unit Load Bed

This concept is very similar to Concept 3, however in the place of Mecanum wheels, a bed of motorised ball transfer units are used. The mechanism of operation of this concept is similar to that of the legacy mouse where the movement of the ball is used to control the position of the mouse pointer. Here the position of the payload is determined by the movement of the ball, with multiple balls being actuated in unison. Each transfer unit will require two motors, one for each axis in the two-dimensional plane.

Critical Concept Evaluation

Once again the generated concepts were critically evaluated and their advantages and disadvantages were identified in terms of the specifications set in Chapter 3. Table 4.2 lists these advantages and disadvantages.

Concept	Advantages	Disadvantages
PI.1	An uncomplicated design. Easily actuated.	Difficult to determine the centre of gravity of the part in this configuration. Difficult to interface with the a standard conveyor system.
PI.2	Easy to interface with a standard conveyor system. Uncomplicated design. Reasonable accurate positioning of payload is possible.	High friction forces are required for rotation of the payload.
PI.3	Accurate positioning of the payload is possible.	Very complicated programming is required. Heavy mechanism with multiple actuators required.
PI.4	Very accurate positioning of the payload is possible.	Heavy design with many actuators required. Very complicated programming is required. Very small contact area – high friction forces are required.

Table 4.2: Critical evaluation of static stability system concepts.

The verdict of the critical concept evaluation is that the most applicable concept is the Roller Load Bed Concept. An added feature here is that the rollers are tapered toward the centre line of the load bed. This is implemented in order to ease the rotation of the payload.

Final Design of Payload Interface System

The final design of the payload system consists of seven mild steel shafts of 20 mm nominal diameter with two separate rollers, machined out of solid Nylon round bar, mounted on deep groove ball bearings. See Appendix C for the calculations for the design of these components. The rollers on each shaft are separated by a 3 mm washer. The shafts are mounted between two Aluminium extruded channel profiles. Channels were used due to the fact that they provide good stability without the added weight that an equivalent solid bar would contribute. Each row of rollers is driven by a DC motor, power transmission from the driving motors to all the rollers is achieved by circular rubber bands. The payload interface system in the assembly phase is shown in Figure 4.5.

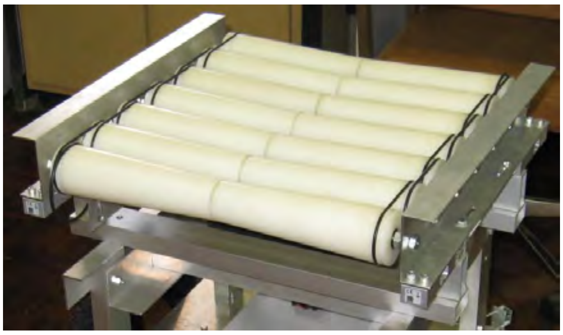


Figure 4.5: Roller load bed payload interface system in assembly phase.

The rollers are mounted in the Aluminium channels extrusions with an eccentricity towards the bottom of the channels in order to create a rail on each side of the load bed for safety considerations. The maximum outer diameter of each roller is 60 mm and they are tapered towards the inside at an angle of approximately seven degrees. The tapered rollers enable the translation and rotation of the payload in order to manipulate it into position. Figure 4.6 shows the movement of the payload due to the relative rotation of the rollers.

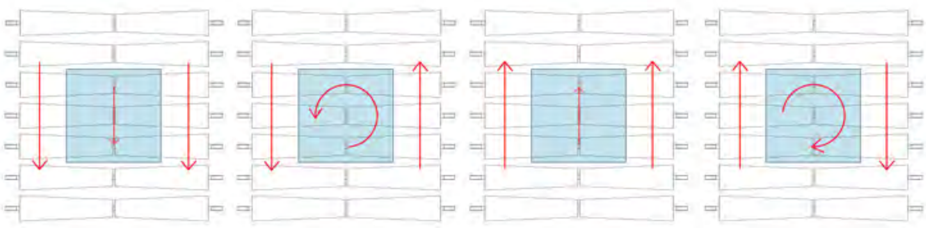


Figure 4.6: Mechanism of operation of payload manipulation.

To fulfill the design specifications stipulating that the platform must be able to sense the position of the payload centre of gravity, a single point strain gauge load cell is mounted underneath each of the four corners of the load bed. The measurements taken from these load cells are used to determine the position of the payload when it is loaded, and to track the position as it is manipulated into the correct position. In other words a feedback control loop is being implemented to achieve this goal. More detail on this control loop can be found in Section 6.2. These load cells form part of the mechanical structure of the platform and can be seen in Figure 4.5.

4.3 Space Frame Support Structure

The main objective of the support structure of the platform is to form a stable base for the integration of the drive system, static stability system, and payload interface system. The integration of the electronic hardware components is also achieved in the successful design of the support structure. Characteristics to be taken into account when designing the support structure of an autonomous mobile robotic platform include weight, stability, conductivity, durability, and safety.

The material that has been employed in the support structure is Aluminium. The reason for this is its low density, fatigue characteristics, resistance to corrosion, and machinability. An eight millimetre thick Aluminium plate forms the base of the platform, which provides the stability of the support structure. This also provides surface for mounting the space frame components. Standard Aluminium extrusion profiles were used for the space frame components, as far as possible. This was done to keep fabrication time and costs to a minimum.

Four rectangular Aluminium extrusions were fixed along the perimeter of bottom of the Aluminium base plate. These components provide extra stability as well as mounting points for the drive system and static stability system. To provide the necessary additional mounting points for the drive system two short rectangular extrusions, of the same size as the base extrusions, are fixed on the insides of the side base extrusions. To each of these are mounted a bearing pillow block and the mount plate for one drive motor, via two smaller Aluminium extrusions. An exploded view of a CAD model of this configuration can be seen in Figure 4.7.

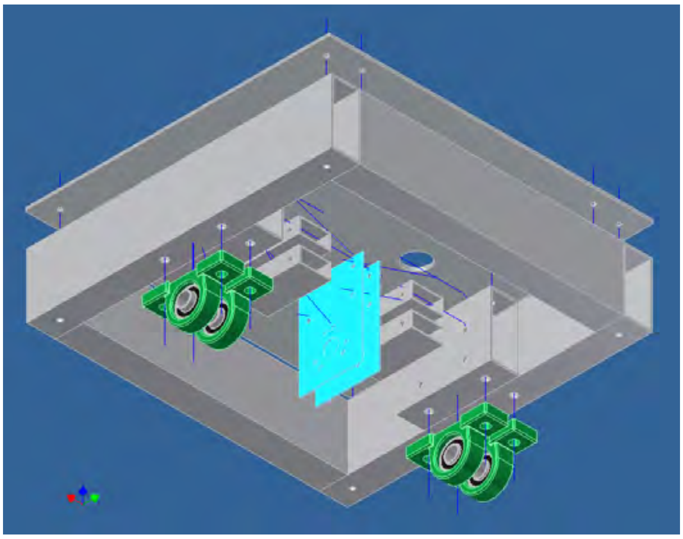


Figure 4.7: Exploded view of a CAD model of the base support structure.

Four small rectangular Aluminium extrusions are mounted in an upright orientation to the top surface of the base plate, to form pillars for mounting the payload interface system and the battery pack. Fixed to the tops of these pillars are two Aluminium angle extrusions to provide horizontal surfaces for mounting the payload interface system. The platform battery pack is suspended between the four pillars on a system of Aluminium angle extrusions. A CAD model of this configuration can be seen in Figure 4.8.

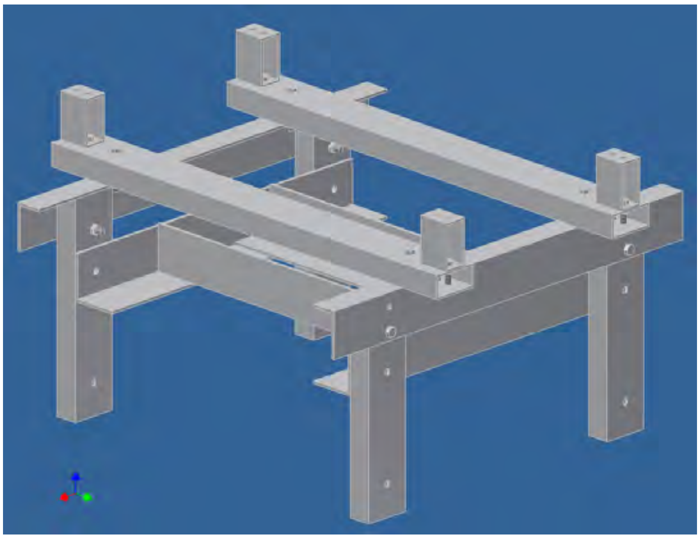


Figure 4.8: CAD model of the space frame support structure.

The heavy battery pack has been mounted as high as possible on the platform in order to increase the height of the centre of gravity of the platform to, in turn, lower the bandwidth of the dynamic response of the platform. This effect is illustrated by the equation of motion, Equation 4.1, of an inverted pendulum in the unstable equilibrium position, supported by an equal coil spring on either side [22].

$$ml^2\ddot{\theta} + \frac{kl^2}{2}\theta - mgl\theta = 0 \tag{4.1}$$

Removing the springs from the system by equating k to zero and simplifying the equation, Equation 4.2 is produced.

$$\ddot{\theta} = \frac{g}{l}\theta \tag{4.2}$$

According to this equation the angular acceleration of the body of the undamped pendulum is inversely proportional to the length of the pendulum shaft. Thus placing the centre of gravity of the platform higher slows down the dynamics of the platform, as this effectively increases the length of the shaft for the simplified, lumped model.

Four large holes are bored through the base plate in strategic places for providing access for wiring to and from actuator and sensors, and controllers. The Single Board Computer (SBC) is mounted to the top of the centre of the base plate in a plastic housing. Mounted along the front and rear perimeter of the base plate are ultrasonic sensors, for navigational purposes. Inertial sensors are mounted to the centre of the bottom of the base plate for balancing purposes.

More detail on the sensors that are employed on the platform along with motivation for specific mounting positions can be found in Chapter 5. Figure 4.9 shows an exploded view of the three systems that constitute the mechanical structure of the platform as designed in CAD, as well as the drive wheels and mounting blocks for the support structure, while Figure 4.10 shows the front view the physical mechanical infrastructure assembly including the electrical and electronic systems, with annotations.

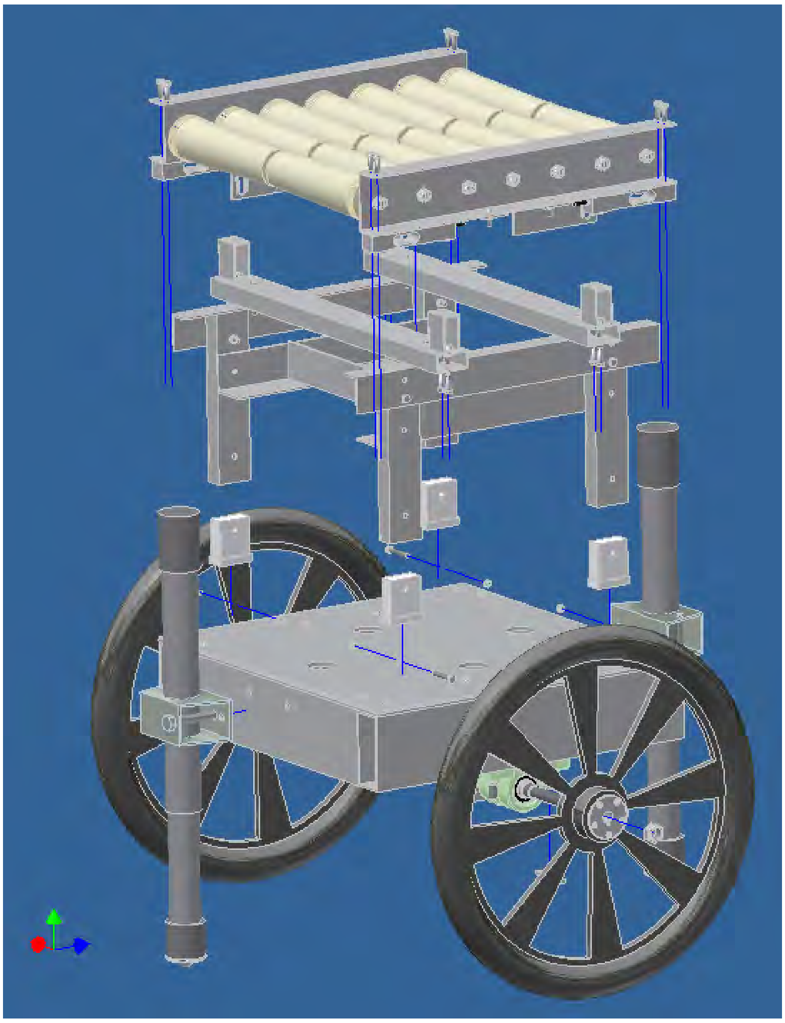


Figure 4.9: Exploded view of a CAD model of the main mechanical structure.

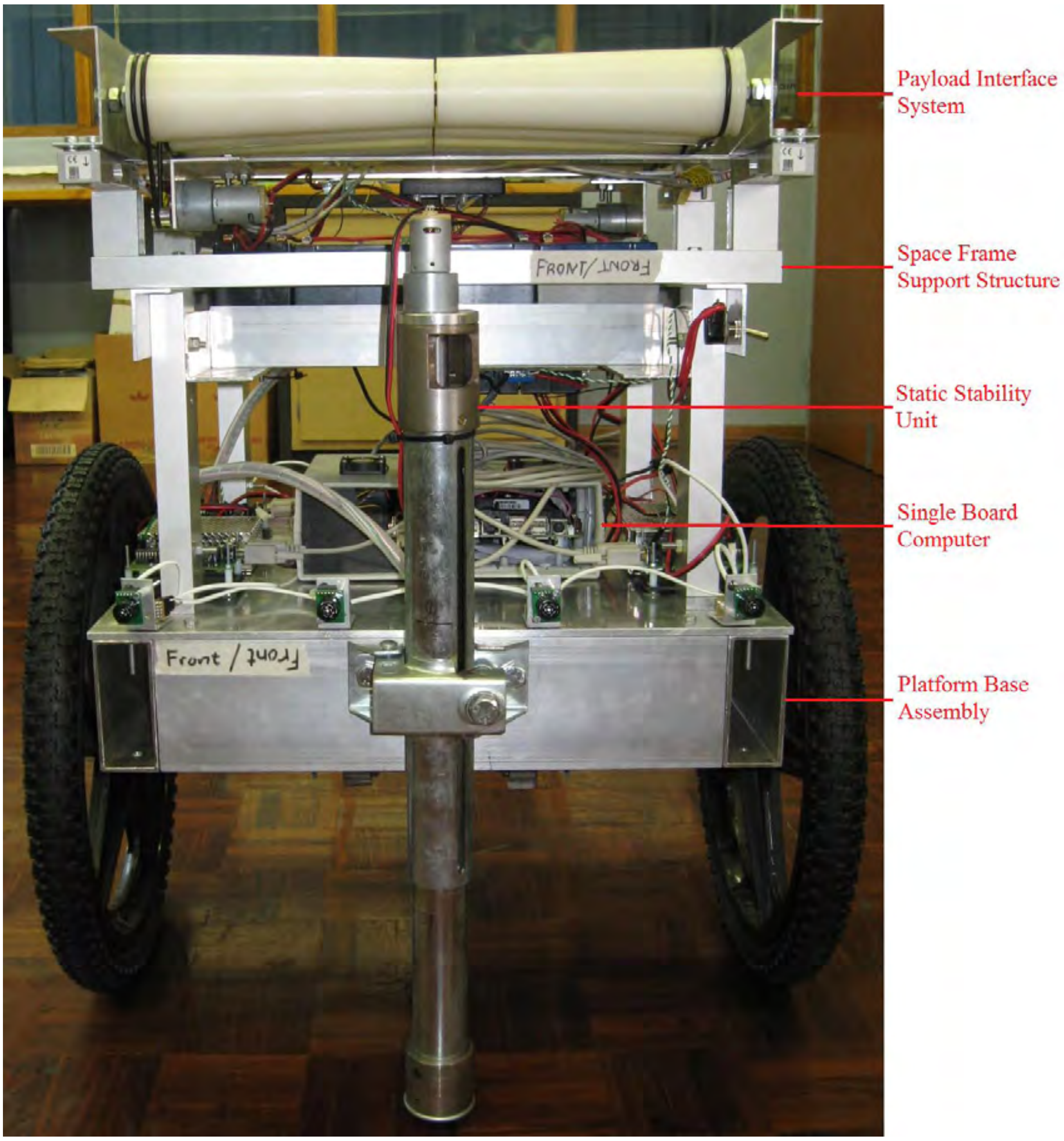


Figure 4.10: Final assembly of physical mechanical structure including electrical and electronic systems.

4.4 Mechanical Architecture Safety

An important aspect to consider in the development of an autonomous mobile robotic platform is the danger involved with its operation alongside humans, in the case of failure of any of its critical components. In order to identify any dangers that may be present in the mechanical system architecture, a Failure Mode, Effects and Criticality Analysis (FMECA) was performed [6].

4.4.1 Failure Mode, Effects and Criticality Analysis

The FMECA was applied to all the subsystems, individually, that make up the mechanical architecture. An FMECA identifies the criticality of all the different modes of failure of each functional component. The probability and severity of each failure mode is evaluated and the findings of the analysis are based on all these considerations. The main objective of an FMECA is to determine what actions are required in order to avoid injury or damage to the health of any person, working in close proximity to the platform. In this instance the FMECA has been applied on a physical entity level, where the various subsystems of the mechanical system represent the physical entities. However, the effects of the different failure modes on the overall system has also been taken into account. The analysis of each subsystem is discussed here.

In the analyses that follow the Severity rating is a numerical value relative to the proposed seriousness of the effects of the corresponding mode of failure. It is scaled to fall in the range of one to ten, where minor effects correspond to low values, and more serious effects correspond to higher values. The Frequency rating is also a numerical value between one and ten, with a lower values corresponding to lower frequency of occurrence and vice versa for higher values for this rating. The Probability rating refers to the probability of detection of the failure before a catastrophic failure occurs. Here a low value indicates a high probability of timely detection while a high value indicates a low probability of timely detection.

In order to combine the effects of each rating a Risk Priority Number (RPN) has been calculated by using Equation 4.3 for each failure mode [6]. The value of RPN gives an indication of the criticality of the corresponding effects of each failure mode.

$$RPN = (severity\ rating)\,(frequency\ rating)\,(probability\ of\ detection\ rating) \tag{4.3}$$

Major design alterations for failure modes with high criticality ratings may be considered, while for failure modes with low criticality ratings minor adjustments may suffice. Tables 4.3, 4.4, 4.5, and 4.6 summarise the analyses of the individual subsystems that make up the overall mechanical architecture.

4.4.2 FMECA Results

According to the values of the RPN's for the failure modes identified, it can be seen that the loss of power and/or loss of power transmission to the drive wheels are the most significant modes of failure. Consequently, these modes of failure also present the most danger to persons working in close vicinity to the platform. Detachment of a drive wheel would be a catastrophic failure however the probability of detection is relatively high, as long as maintenance checks are carried out routinely. This is also why the probability of detection of the loss of power transmission to the wheels is slightly higher than for the loss of power to the motors. Routine inspection and maintenance is the least expensive, and most reliable means of detecting the probability of failure of these mechanisms timeously.

Although the severity is relatively low the loss of power to one or both the static stability units in the up or intermediate position is the second most critical mode of failure in the mechanical system. The reason for this is that without properly functioning static stability units the platform cannot perform its primary functions, unless the static stability units are permanently in the fully extended position. This is why these two situations are distinguished in the FMECA. In order to determine whether the static stability units are extended completely or not, a limit switch is installed at the bottom of the outer cylinder of one of the units. This limit switch is triggered by the pin that is attached to the inner cylinder. Detachment of a static stability unit may be avoided through routine inspection and maintenance of the mechanical system.

The failure modes of the payload interface system functions are of relatively low criticality. They do not pose major danger to humans or functionality of the complete system. Safety functions in the event of these failures are built into the software and control system of the platform. Although the severity of the failure modes of the support structure are relatively high, the probability of detection is also high by simple visual inspection. It is for this reason that the criticality of these failure modes are not high.

Safety functions for some failure modes that have been addressed here are dealt with by the software system of the platform. These will be discussed in Chapter 6. These functions assume that the software system continues to function normally in case of failure

Failure modes	**Potential cause**	**Potential effect(s)**	**Frequency**	**Severity**	**Probability**	**RPN**
Deflated tire(s)	Puncture Faulty valve	Loss of traction and accurate motion control	4	2	2	16
Loss of power to motor(s)	Loose connection Wire breakage Faulty electronics	Loss of ability to maintain active stability Loss of payload	1	9	7	63
Loss of power transmission to wheel(s)	Broken shaft coupling Stripped gears	Loss of ability to maintain active stability Loss of payload	1	9	6	54
Detachment of wheel(s)	Broken shaft Lost wheel nut	Loss of ability to maintain active stability Loss of payload	1	8	4	32

Table 4.3: FMECA for the drive subsystem.

Failure modes	Potential cause	Potential effect(s)	Frequency	Severity	Probability	RPN
Loss of power to lead screw in up/intermediate position	Loose connection Wire breakage Faulty electronics	Safe loading of payloads will be impossible	2	5	6	60
Loss of power to lead screw in down position	As above	Platform must continue operation in permanently statically stable configuration	2	2	6	24
Detachment of static stability unit(s)	Broken bolt connections	Safe loading of payloads will be impossible	1	5	4	20

Table 4.4: FMECA for the static stability subsystem.

Failure modes	**Potential cause**	**Potential effect(s)**	**Frequency**	**Severity**	**Probability**	**RPN**
Loss of power to motor(s)	Loose connection Wire breakage Faulty electronics	Positioning of payload will be impossible – Ability to maintain active stability will be seriously impaired	1	3	6	18
Loss of power transmission to rollers	Broken power transmission band Stripped gears	As above	2	3	4	24

Table 4.5: FMECA for the payload interface subsystem.

Failure modes	**Potential cause**	**Potential effect(s)**	**Frequency**	**Severity**	**Probability**	**RPN**
Detachment of support structure from base	Broken bolt connections	Structural integrity of platform will be compromised	1	8	3	24
Detachment of support structure from payload interface	As above	As above	1	8	3	24

Table 4.6: FMECA for the support structure subsystem.

of a physical mechanism.

4.5 Chapter Summary

This chapter documents the design of the various components that constitute the mechanical hardware architecture of the platform. These components have been discussed according to the specific goal that each component is required to achieve. In each case the final design is clearly described and illustrated. The complete mechanical structure of the platform is also described, as assembled from the various systems described here. A functional safety analysis has been performed and the outcome has been documented here.

Chapter 5

Electronic Hardware Components

Fundamentally the electronic hardware, onboard an autonomous mobile robotic platform, gather data from the environment, processes this data and commands actuators in response to the changes in the environment. This chapter steps through the various components that make up the electronic system onboard this platform, in terms of data gathering, data processing and computational capability, communication, actuation, and power management. The chapter is concluded with a section describing the integration of the various electronic hardware components discussed here.

5.1 Onboard Computing Facilities

The onboard computing facilities are made up of the Single Board Computer (SBC) and the microcontrollers that form the interface between the SBC and the sensory and actuation components onboard the platform.

5.1.1 Single Board Computer

The SBC constitutes the central component of the onboard computing facilities. The microcontroller circuits discussed in Section 5.1.2 are interface with the SBC via serial connections. The SBC is also in communication with a remote desktop computer which deals with the materials handling system management, via a wireless communication connection. More detail on the communication protocol and hardware can be found in Section 5.3 Requirements that were identified for the selection of the onboard SBC include:

- The computer must be able to support the choice of software discussed in Chapter 6.
- The computer must support standard peripherals to avoid compatibility problems.

- The computer must be robust enough to cope with operation in a manufacturing environment.
- The computer must have low power consumption.

Based on these considerations the IEI EPIC NANO-GX466 CPU motherboard was selected to be employed onboard the platform. The EPIC motherboard form factor is an industrial form factor that is designed for industrial applications.

A SATA Hard Disk Drive (HDD), with a capacity of 80 GB, is used for housing the operating system and accompanying software. A SO-DIMM RAM module is installed on the onboard SBC with a capacity of 256 MB. The specified power consumption of this motherboard is 12 V at 0.83 A, when 512 MB of memory is installed. Figure 5.1 shows the SBC with HDD installed as mounted in the enclosure.

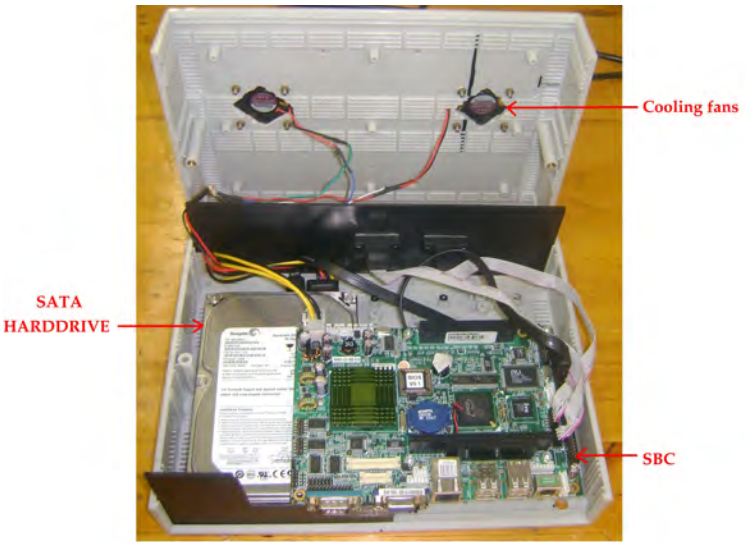


Figure 5.1: EPIC NANO-GX466 SBC with SATA Hard Disk Drive.

Features and specifications that are relevant to the application include:

- AMD GX466 CPU with 333 MHz front-side bus.
- 1 x 200-pin Small Outline Dual In-line Memory Module (SO-DIMM) Random Access Memory (RAM) support.
- 2 x Serial AT Attachment (SATA) function support.

- 6 x RS-232 connectors.
- 4 x USB 2.0 connectors.

It is able to withstand temperatures from 0 °C to 60 °C and non-condensing humidities between 5 % and 95 %.

5.1.2 Microcontrollers

In total there are three microcontrollers implemented onboard the platform. For simplicity of programming they are all identical controllers based on, and designed around the Atmel ATmega32 microprocessor. One microcontroller is implemented to controller each of the three sets of DC motors, these are the drive motors, the static stability system motors, and the payload interface roller motors. These microcontrollers also control the various sensors implemented onboard the platform.

The ATmega32 microprocessor is based on the AVR enhanced Reduced Instruction Set Computing (RISC) architecture. The data bus is eight bits wide, it possesses 32 KB of programmable flash memory, 1 KB of Electrically Erasable Programmable Read Only Memory (EEPROM), and 2 KB of internal Static RAM, or SRAM. Features include an eight channel, 8- or 10-bit analogue-to-digital converter (ADC), with incorporated differential gains, as well as interrupt-based Pulse Width Modulation. The ATmega32 microprocessor possesses Inter-Integrated Circuit (I^2C) communication capability, as well as Transistor-Transistor Logic (TTL) pins which may be used for serial communication. C programming language as well as Assembly language can be used to program these microprocessors.

The single layer Printed Circuit Board (PCB) that the ATmega32 is installed on, is designed to establish serial communication, via the Universal Asynchronous Receiver/ Transmitter (UART) protocol, with the onboard SBC. It takes a supply voltage between 9 V and 12 V and supplies the microprocessor with a regulated 5 V. It also possesses output pins to output the voltage that is supplied to it, as well as the regulated 5 V as supplied to the microprocessor. The ATmega32 microprocessor has four digital input/output (I/O) ports with eight pins each, which are all made available for connection through the PCB. Figure 5.2 shows a photograph of an unconnected microcontroller with the microprocessor installed.

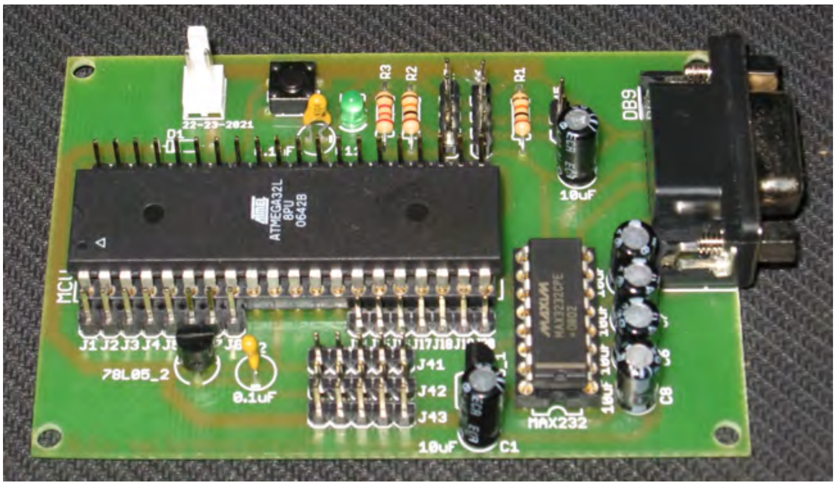


Figure 5.2: ATmega32-based microcontroller.

Additional features include a reset button for in system resetting, an LED to indicate whether the board is powered, and I^{2}C communication capability. The I^{2}C pins are not permanently connected, this way the pins that are used for I^{2}C communication may be used for digital I/O if this is necessary. Serial communication is achieved through a TTL to RS-232 conversion Integrated Circuit (IC) built into the PCB.

5.2 Sensory Infrastructure

The sensory infrastructure in an autonomous mobile platform is a critical component that determines the level of autonomy that the platform is capable of. On this platform there are two main objectives that sensor data are used for. The first, as mentioned is autonomy, and the second is to maintain the upright orientation of the platform, in other words the active stability control of the platform. The roles that the sensors play in the achievement of these goals are documented here.

5.2.1 Ultrasonic Rangefinders

Eight ultrasonic rangefinders are connected in a chain formation, which is connected to one of the microcontrollers as discussed in Section 5.1.2. The data from these sensors are used for local obstacle avoidance and localisation within the map that is provided by the remote materials handling system management server. The rangefinder that is being implemented is the SRF02 which is produced by Devantech [39]. Some of these sensors are shown in Figure 5.3 as mounted on the base of the platform.

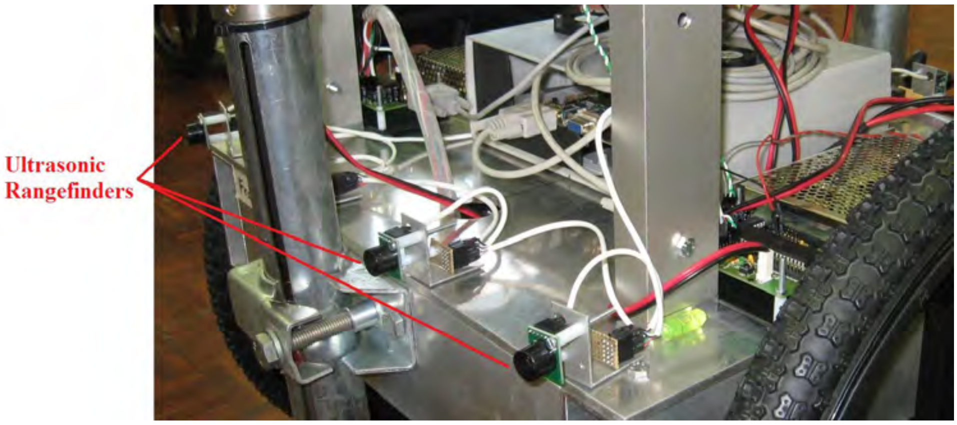


Figure 5.3: Devantech SRF02 ultrasonic rangefinders as mounted on the platform.

This device comes with signal conditioning circuitry and transmits at a frequency of 40 kHz. It makes use of a single transducer for transmission and reception of pulses. The minimum range is 15 cm and the maximum range is 6 m. The principle of operation is based on measuring the time between transmission and reception of a pulse, taking the frequency of the pulse into account, to determine the distance that the reflecting object is away from the sensor. The cone angle of this sensor at half of the maximum range is approximately 45°.

This sensor is capable of communicating via I^2C or UART protocols. In this implementation the sensors are connected as an array via an I^2C bus. Each sensor is connected to the two sensors adjacent to it, and the first is connected to the microcontroller controlling the transmission sequence, and relaying the sensor data to the onboard SBC.

The rangefinders are mounted to the perimeter of the base plate of the platform and are directed outward, four in a forward direction, and four in a rearward direction. Each sensor is mounted at such an angle to maximise the covered area, and minimising crosstalk, or inter-sensor interference. The programmed transmission sequence is also designed to minimise inter-sensor interference. Figure 5.4 shows a visual representation of ultrasonic rangefinder mounting orientation, approximating the beam pattern as a 45° cone. The red dotted circle indicates half of the maximum range. The beams shown in Figure 5.4 are numbered, one through eight, indicating the sequence of transmission as discussed above.

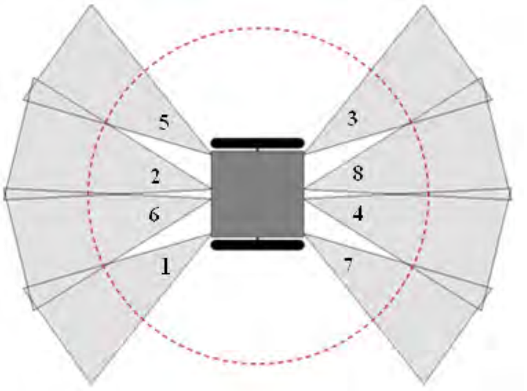


Figure 5.4: Visual representation of ultrasonic rangefinder mounting orientation.

5.2.2 Wheel Encoders

Fixed to the output shaft of each drive motor is a rotary encoder, used to track the rotational displacement of the drive wheels. The specific encoder that is implemented is the HEDS-5500 encoder produced by Avago. It is a dual channel encoder, which implies that the functionality of this encoder is extended, past simply counting pulses, to determining direction of rotation as well. This extended functionality is exploited by the technique implemented for localisation of the platform, discussed in Section 6.2.5.

Channel A of the encoder is connected to an external interrupt pin of the controlling microcontroller. Channel B is connected to a digital input port pin of the same microcontroller. The two channels are $90°e$ (electrical degrees) out of phase. This connection configuration enables the programming of the controller such that channel B is checked each time channel A is triggered. This technique enables the determination of direction of rotation; while channel B is high when channel A is triggered, the wheel is rotating in one direction, and while channel B is low when channel A is triggered the wheel is rotating in the opposite direction.

The encoders are each fastened to a 1/4 in. diameter extension at the rear of the output shaft of the drive motor. In Figure 5.5 the encoder can be seen fastened in position at the rear of the output drive shaft. This fastening configuration produces a 1-1 ratio of encoder rotation relative to wheel rotation.

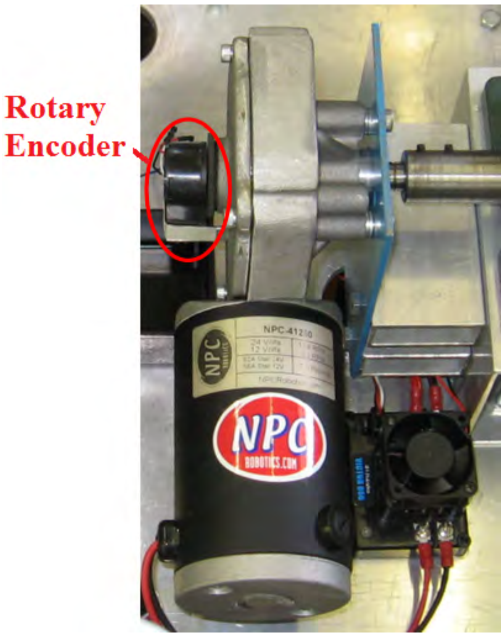


Figure 5.5: HEDS-5500 rotary encoder fastened to the rear of the drive shaft.

5.2.3 Dual-Axis Accelerometer

To determine the tilt angle of the platform relative to the vertical plane the ADXL204 dual-axis accelerometer, produced by Analog Devices, Incorporated (ADI), is being implemented. This accelerometer is based on ADI's inertial Micro Electro Mechanical Systems, or *iMEMS*, technology. The maximum range that this sensor is able to measure is ±1.7g in the direction of each axis. The sensitivity specified by the manufacturer is 1000 mV/g at a supply voltage of 5 V, with a nonlinearity in the output of 0.2 % of full scale [17].

The sensor IC itself is mounted on a PCB, as shown in Figure 5.6 with signal conditioning capacitors included, which set the output bandwidth to 50 Hz. The output of this device is two analogue signals proportional to the acceleration of the package in the directions of the two axes. The data from the relevant axis are gathered by a microcontroller and converted to digital values at a certain frequency, for application by the control program. This sampling frequency is determined by the accuracy and speed of measurement required, as well as the bandwidth of the analogue output.

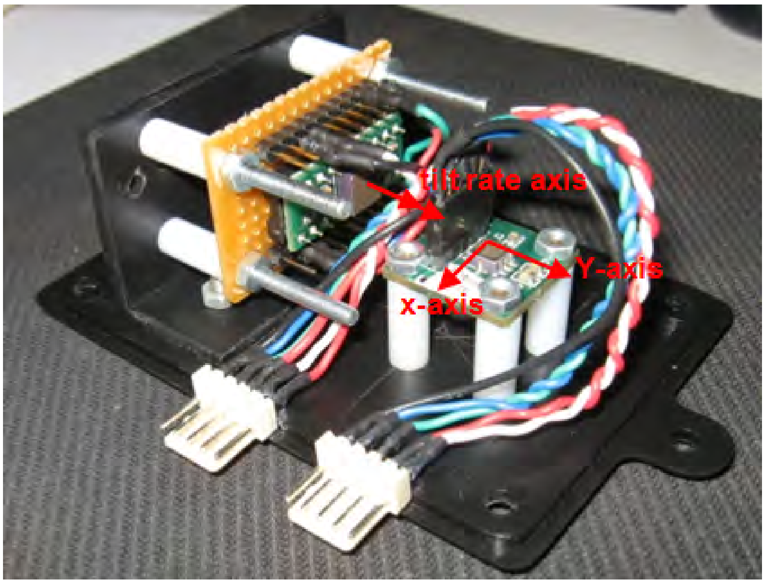


Figure 5.6: Accelerometer (right) and gyroscope (left) as mounted on the platform, with axes indicated in red.

When this sensor is employed as a tilt sensor, the manufacturer recommends that the axis used for tilt sensing be mounted horizontally in the direction of tilt. This orientation produces the greatest sensitivity when the package is tilted away from the horizontal orientation [17]. The x-axis, indicated in Figure 5.6, is being used for tilt sensing. In order to achieve this configuration the sensor is mounted to the bottom of the base plate such that the x-axis is horizontal and perpendicular the the axis that the drive wheel axles form, when the platform is perfectly vertical. This ensures that when the platform is perfectly vertical the voltage output of the x-axis output should be half of the supply voltage, referred to as the nominal output.

The most important shortcoming of this type of sensor is a sensitivity to high frequency noise, due to the high measurement sensitivity, which is in the range of 1000 mV/g. This was noticed in the process of sensor selection as documented in project documentation online [5], [28], as well as university projects [36]. Tests were carried out to determine the extent of this effect. This involved logging the output of the accelerometer while at rest for a period of 20 minutes at a frequency of 60 Hz. Figure 5.7 shows the 10-bit ADC converted values of the accelerometer output.

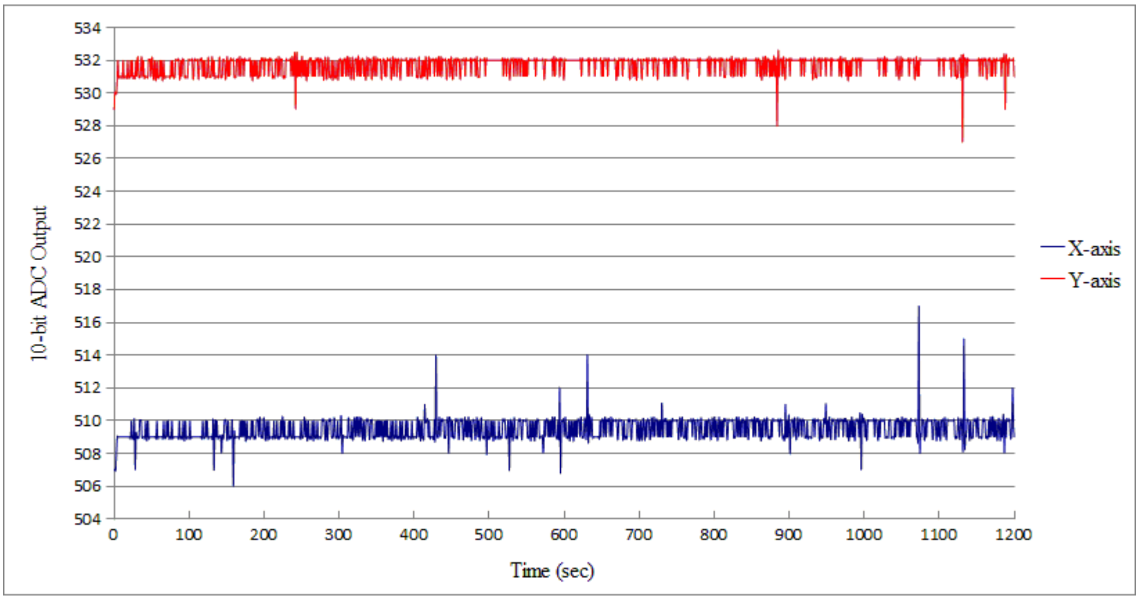


Figure 5.7: 10-bit digital accelerometer output at rest.

The noise mentioned above is manifested in the apparently random outliers that are present in both the X- and Y-axis outputs. These outliers seem to be more frequent in the X-axis output, however these two should be identical. The technique used to take care of this effect is discussed in Chapter 6.

In order to be able to use the measurements from the accelerometer effectively the ADC outputs need to be calibrated. An experiment was carried out in order to determine the response of the sensor to nominal tilt angles. The measurement from the X-axis only were taken during this experiment for the fact that this axis is going to be utilised by the control system.

The setup of the apparatus used for the experiment is shown in Figure 5.8. An angular measurement device was setup and clamped in position at angles of 0°, 5°, 10°, 15°, and 20° , and also in the opposite direction, which corresponds to negative values of these angles. The accelerometer was subsequently clamped in position to the angular measurement device with the sensitive axis in question in line with the ordinate of the angular measurement device.

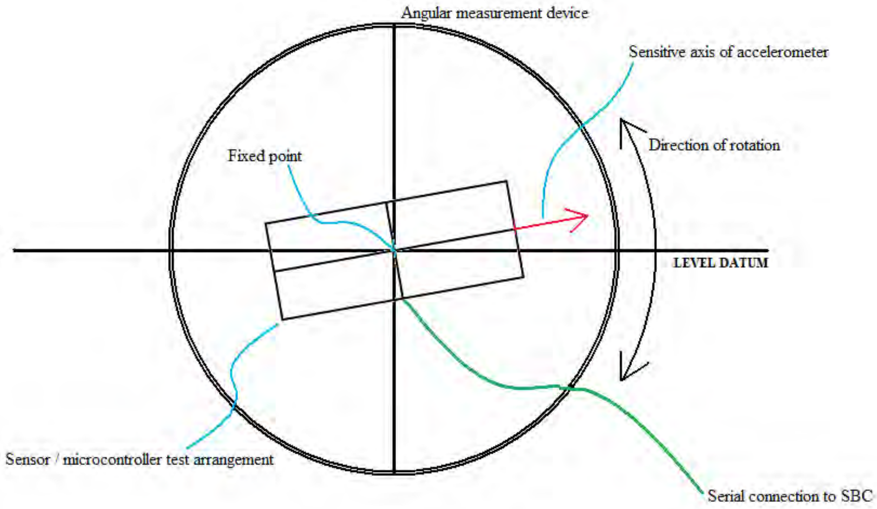

Angular measurement device
Sensitive axis of accelerometer
Direction of rotation
Fixed point
LEVEL DATUM
Sensor / microcontroller test arrangement
Serial connection to SBC

Figure 5.8: Illustration of the accelerometer calibration apparatus setup.

The output of the 10-bit ADC was displayed on screen and logged in a text file format. This was carried out three times in order to find a acceptable statistical average for the overall measurements. Table 5.1 shows the averages taken at each angular position as mentioned.

Angle (degrees)	1^{st} Run	2^{nd} Run	3^{rd} Run	Average	Conversion factor
0	0	0	0	0.0	N/A
-5	-18	-18	-18	-18.0	3.6
-10	-35	-36	-36	-35.7	3.6
-15	-54	-54	-54	-54.0	3.6
-20	-72	-72	-72	-72.0	3.6
5	20	18	19	19.0	3.8
10	37	37	36	36.7	3.7
15	55	55	54	54.7	3.6
20	72	72	72	72.0	3.6

Table 5.1: Data produced during the accelerometer calibration experiment.

The values shown in the 1^{st}, 2^{nd}, and 3^{rd} Run columns are outputs from the 10-bit ADC that is being used in the control program for the control of the active stability. The

Average column is simply the averages of the three runs for statistical purposes. The Conversion factor column is calculated by dividing the Average column entry by the Angle column entry in each corresponding row.

The average conversion factor was found to be 3.6. This value is being used in the control program to convert the accelerometer outputs to measurements in degrees. These values may also be converted to radians by multiplying the average value of the ADC output by a factor of 208 *radians/ADC unit*. Although the range of angles is relatively small, the the platform shouldn't tilt more that approximately 5° in each direction, according to the design specifications. The output was found to be linear for the range of angular settings listed.

5.2.4 Rate Gyroscope

A solid state gyroscope is implemented for determining the tilt rate of the platform. The sensor that was selected to perform this task is the ADXRS614 *iMEMS* solid state rate gyroscope produced by ADI. This sensor produces 25 mV/°/s, with the maximum measurable range being ±50 °/s [18]. The rated supply voltage is 5 V, which is supplied by the controlling microcontroller in this case. The output of this sensor is ratiometric with respect to the reference voltage supplied to it. When the sensor is stationary the nominal output voltage should be half of the supply voltage. The nonlinearity in the output is 0.1 % of full scale.

The gyroscope is mounted to a PCB which also includes signal conditioning circuitry. This circuitry sets the bandwidth of the output to 10 Hz. The rate of rotation of the sensor about its sensitive axis, indicated in Figure 5.6, determines the proportional analogue output. This analogue output is also converted to digital values at a certain sampling frequency, determined by the accuracy and time response required and also the bandwidth of the sensor output.

An important characteristic of solid state rate gyroscopes is the phenomenon of output drift. This is also widely documented in the sources mentioned in the previous section. A similar test was performed as in the case of the accelerometer. The output of the gyroscope at rest was logged for a period of 20 minutes at a frequency of 60 Hz. The 10-bit ADC values were plotted on a graph and are shown in Figure 5.9.

In the figure, the blue graph represents the actual data gathered from the gyroscope,

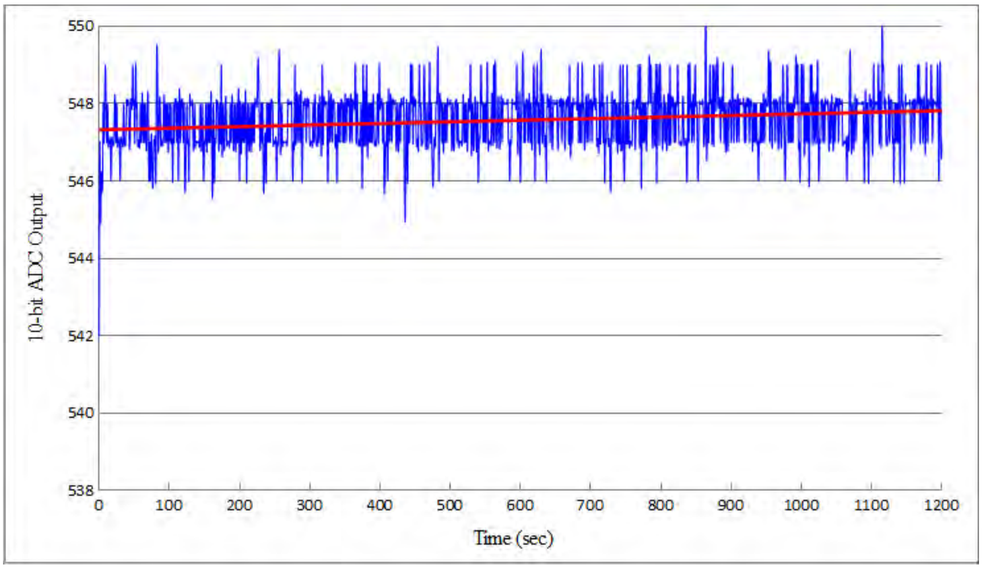


Figure 5.9: 10-bit digital accelerometer output at rest.

while the red line is a linearly calculated trend line. This trend line clearly shows the drift of the output over time. According to the ratiometricity characteristic of the sensor, if the maximum value for the digital output is $2^{10} = 1024$, for a 10-bit ADC, then the nominal output should be half of the maximum output, 512. It can be seen from the data that the sensor also possesses a bias error, which can be taken care of easily in programming by measuring the bias error at every start up when the sensor is stationary.

The ratiometric characteristic of this sensor does however simplify its calibration. After correcting for the bias error in the code the measurement can simply be converted to an angular rate measurement by scaling it by the maximum output range of the sensor, $100\ ^{\circ}/s$ in total, divided by the maximum ADC output (1024).

The sensor is mounted adjacent to the accelerometer, with the sensitive axis parallel to the axis that the drive shafts form. The container housing both inertial sensors is mounted at a height of approximately 100 mm vertically above the drive shaft axis, when the platform is perfectly vertical. The positioning of the sensors on the platform is not critical, due to the fact that all points in a rigid body rotate at the same angular velocity and accelerate at the same angular acceleration, in plane rotational motion. The angle of rotation of all points in a rigid body in plane rotation is also equal [33]. Thus the only critical aspect is the orientation of the sensitive axes.

5.2.5 Strain Gauge Load Cells

A strain gauge load cell is mounted at each corner of the roller bed payload interface described in Section 4.2. The measurements from these load cells are used to determine the position of the payload centre of gravity relative to the centre of gravity of the payload interface in the forward-backward direction. The rollers are used to manipulate the payload into position such that the centre of gravity is aligned with the centre plane of the platform. This is a complimentary function for assisting in the active stabilisation of the platform.

PW6D single point dynamic strain gauge load cells are being implemented in this situation. These are produced by a German company by the name of HBM. These sensors are specifically designed for point weighing applications. The sensitivity is rated at 2 mV/V with a maximum allowable load of 40 kg. Figure 5.10 shows these load cells as mounted on the platform.

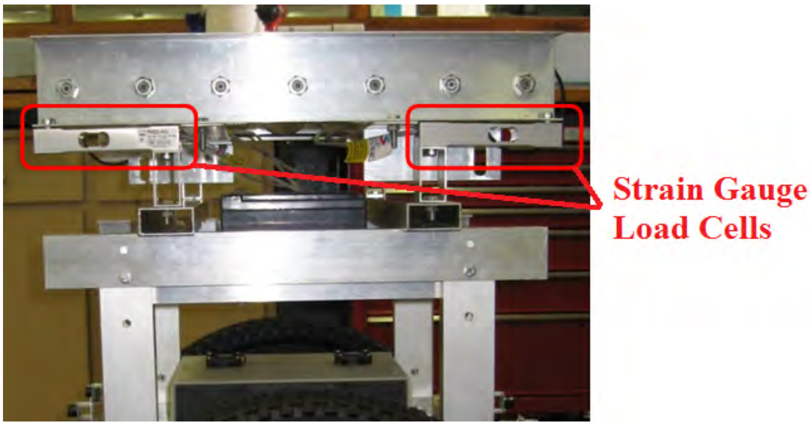


Figure 5.10: Strain gauge load cells mounted at the corners of the roller bed payload interface.

The standard 4-wire load cell connection is implemented for connecting these sensors to the microcontroller that is tasked with gathering their data, with a shielding wire connected to the body of the load cell for protection against electrostatic discharge. These load cells are based on a full Wheatstone bridge [7], for this reason the wires are labeled as positive excitation, positive signal, negative signal, and negative excitation.

The difference between the positive and negative signals is taken as the output. This is an analogue output and is in the order of 10 mV; for this reason it is necessary to amplify the output before it is fed to the microcontroller. This is done through the use of instrumentation amplifier circuits that amplify the difference between two voltages with a wide range of gains.

The instrumentation amplifier used here is the AD620N IC, produced by ADI, with accompanying circuitry that sets the gain at 500. The reason for not using the differential ADC channels on the ATmega32 is that the number of available ports is too small. By using the instrumentation amplifier circuits ports are freed for controlling the payload interface motors.

5.3 Communication System

When referring to the communication system, in this context, the communication interface and hardware are referred to. These components are implemented for establishing connections between the platform and other platforms operating in the same environment, as well as with remote management entities such as servers. The selection of communication system to be implemented on the platform relied on certain critical considerations. These include:

- The selected system must be wireless.
- The selected system must be accepted in industry.
- The technology must be mature to the degree that the hardware does not become obsolete during the course of this research.
- The platform must be able to communicate effectively with other platforms and components being developed that make up the RMS.

After consideration of these criteria the communication system that was selected is the wireless network interface based on the IEEE 802.11 standard specifications, known as *WiFi* [14]. The main reason for this decision is to produce optimal interaction between the other RMS components being developed concurrently with this platform. According to the standard, transmission occurs at a bandwidth of 2.4 GHz, which is a widely used frequency range in manufacturing environments.

In terms of the physical hardware to establish the wireless connections, a USB wireless network adapter is being used on the platform, as well as at the remote materials handling management system server. This adapter converts data from the USB standard to the *WiFi* standard and vice versa for wireless transmission and reception of data. Visually these wireless connections may be represented as in Figure 5.11.

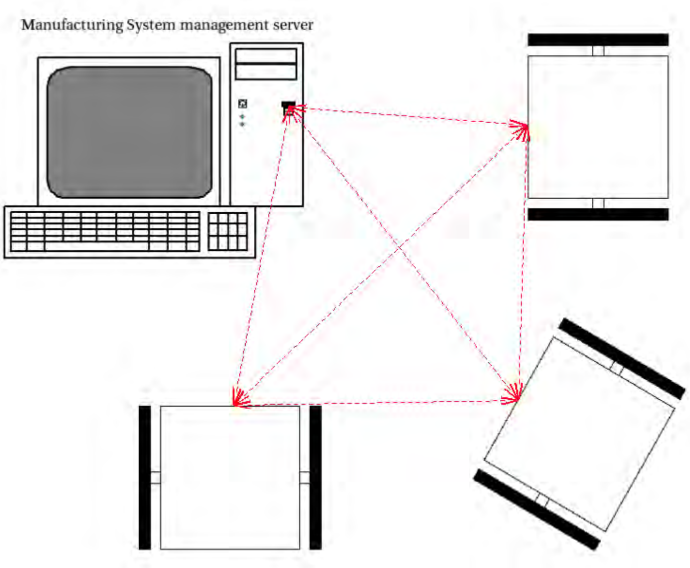


Figure 5.11: Wireless communication between the various components within the manufacturing environment.

The platforms depicted in Figure 5.11 may be instances of the platform discussed in this document or they may be permanently statically stable materials handling platforms that utilise the same communication interface. The red dotted lines represent the wireless connections formed between the platforms and between each platform and the manufacturing system management server.

5.4 Actuation Systems

All motors implemented onboard this platform are DC motors. The DC voltage that each motor receives is controlled by modulating the width of pulses, or duty cycle, of the full voltage relative to zero voltage at a frequency of 50 Hz. This technique of voltage control is known as Pulse Width Modulation (PWM). In a DC motor the output speed is directly proportional to the input voltage [8].

Due to the fact that the microcontrollers cannot deliver the current levels that DC motors draw, motor drivers are implemented to amplify the current to the levels that the motors require. Such a motor driver takes the form of a full H-bridge current amplification circuit, made up of four solid state high voltage switches connected in an “H-configuration” around the motor. The full H-bridge configuration of these motor drivers enable them to drive the motors in both direction [8]. Figure 5.12 shows the connection of the DC motor speed controllers as implemented for the drive motors on the platform.

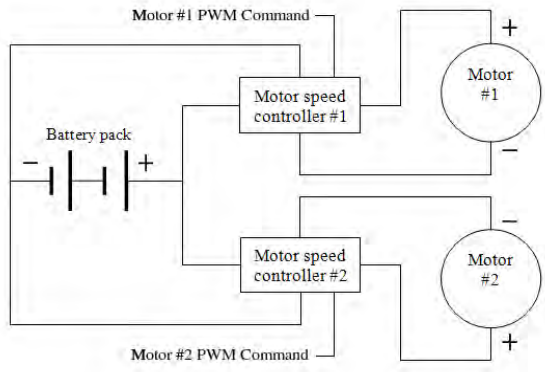


Figure 5.12: Connection of DC motor speed controllers.

The drive motors as discussed in Section 4.1.1 are driven by two IFI Robotics Victor 885 speed controllers. These devices are capable of delivering a continuous current of 120 A and a surge current of 200 A for 2 sec at a nominal voltage of 24 V. Figure 5.13 shows the connection configuration for this device [40].

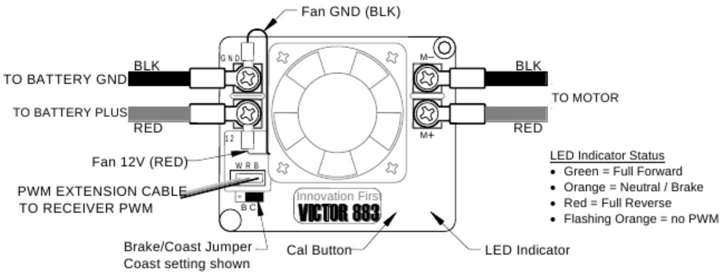


Figure 5.13: Connection instructions for the Victor 885 speed controller [40].

One motor is used for actuating each of the static stability units, the same model of motor is used for actuating each of the sets of rollers that make up the payload interface. These motors are rated to draw 3.5 A of current under stall conditions, and rotate at 180 rpm, through a spur geared gearbox, under no-load conditions drawing a current of 200 mA. Each set of these motors is driven by a single Sabertooth 2X5 dual output motor driver, produced by a company known as Dimension Engineering. This device is capable of producing 5 A of current continuously on both output channels, and peak outputs of 10 A for a short period of time.

5.5 Power Management

The management of electrical power onboard a mobile robotic platform is an important consideration due to the sensitivity to electrostatic discharge and electrical surges of electronic components, as well as limitations on weight and space. Rechargeable sealed Lead-acid batteries are being used onboard the platform to supply the required power to the various electronic components. These batteries are relatively heavy, however they are not expensive and widely available. Lithium-ion batteries were also considered for use here, however the high cost outweighs the savings in weight of this technology.

One 50 W DC to DC voltage converter and one 25 W converter is being implemented onboard the platform. The 50 W converter draws power from a 12 V 10 Ah Lead-acid battery; it supplies power to the SBC, the HDD, and the microcontrollers installed on the platform, at a nominal voltage of 12 V. Figure 5.14 shows the SD-50A-12 DC-to-DC voltage converter with the connections illustrated as described here.

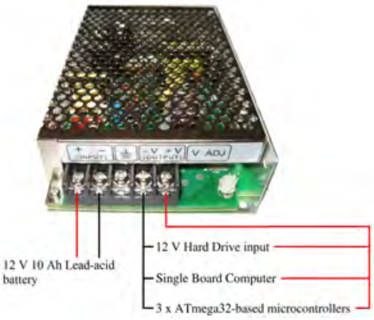


Figure 5.14: SD-50A-12 DC-to-DC voltage converter with connections illustrated as described.

The SD-50A-12 implemented in this instance is able to supply 4.2 A of current continuously. The combined current requirements of the connected components is estimated at 3 A. The 25 W SD-25A-5 converter is used solely to supply 5 V to the HDD, as it requires a 12 V and a 5 V supply. The current requirements of the HDD fall well within the capabilities of the SD-25A-5 converter. Figure 5.15 shows the SD-25A-5 DC-to-DC voltage converter with its described connections illustrated.

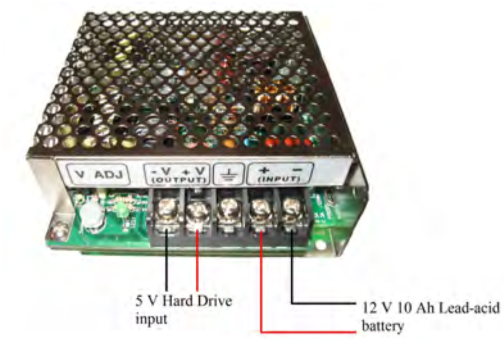


Figure 5.15: SD-25A-5 DC-to-DC voltage converter with connections illustrated as described.

Due to the high current requirements of the drive motors, their drivers are powered by two dedicated 12 V 12 Ah Lead-acid connected in series to deliver 24 V. The drivers driving the motors powering the static stability units and the payload interface rollers are powered by a 12 V 7.2 Ah Lead-acid battery. This is sufficient due to the small power requirements of these motors. All sensors that are implemented onboard the platform make use of 5 V power supply. The microcontrollers are able to provide power to all sensors connected to each controller via an additional built-in 5 V voltage regulator, which draws power from the main 12 V voltage supply to each microcontroller. DC voltage converters are being used to regulate the voltages provided to the electronic components onboard to prevent any damage to these components due to over voltage or short circuits.

5.6 Integration of Electronic Components

Effective integration of sensors and actuation components into the electronics system is crucial to the development of an optimal mechatronic system. The electronic system may be categorised into three components, or circuits, characterised by the functions that the microcontroller concerned performs. This section is purely concerned with the flow of data within the onboard electronic system, and between the platform and the manufacturing system management server.

Drive controller circuit

The control of the primary drive system instantiates one of the electronic circuits mentioned above. This includes the drive motor speed controllers, the two wheel encoders, and the inertial sensors, referring to the dual-axis accelerometer and the rate gyroscope. These components are connected to a single ATmega32-based microcontroller, as described in Section 5.1.2. The drive controller circuit is graphically represented in Figure 5.16.

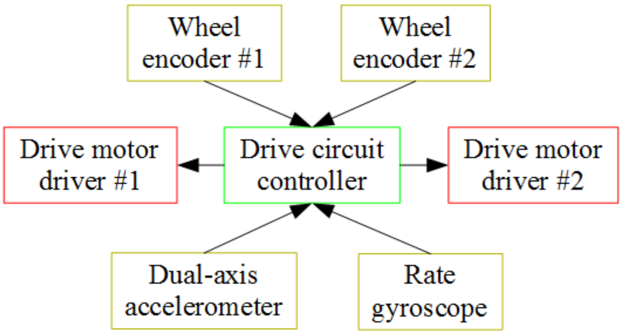


Figure 5.16: Graphical representation of the drive controller circuit.

Each of the wheel encoders utilises an external interrupt pin and a digital I/O pin on the microprocessor, for channels A and B, respectively. The two inertial sensors each produces analogue data gathered by a digital I/O pin for processing by the ADC. Each of the drive motor speed controllers take commands from a digital I/O pin producing a PWM signal. All of these components also take 5 V supply voltage from the microcontroller PCB.

Ultrasonic rangefinder and static stability system circuit

A single ATmega32-based microcontroller is implemented to control the array of eight ultrasonic rangefinders, as well as the motors that actuate the static stability system units. As depicted in Figure 5.17, the ultrasonic rangefinders are connected in a chain configuration on an I^{2}C bus which is connected to, and controlled by, the microcontroller. For this configuration the I^{2}C pins on the microporcessor are connected to the dedicated

I^{2}C connectors on the microcontroller PCB.

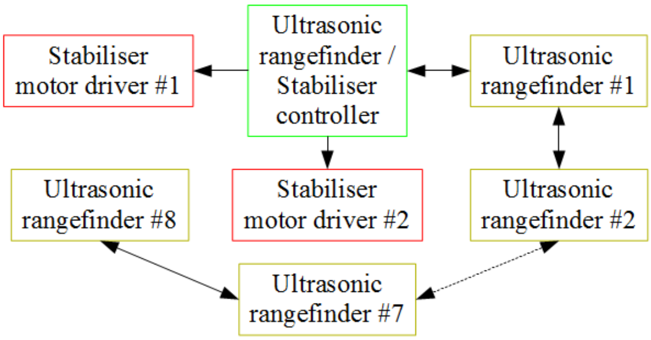

Stabiliser motor driver #1
Ultrasonic rangefinder / Stabiliser controller
Ultrasonic rangefinder #1
Ultrasonic rangefinder #8
Stabiliser motor driver #2
Ultrasonic rangefinder #2
Ultrasonic rangefinder #7

Figure 5.17: Graphical representation of the ultrasonic rangefinder and static stability system controller circuit.

As with the drive motor speed controllers, the static stability unit motor speed controller takes commands from digital I/O pins producing PWM signals. Along with the motor speed controller, the I^{2}C components also take 5 V supply voltage from the microcontroller PCB. In Figure 5.17 the I^{2}C connections are indicated as bidirectional arrows. This is to illustrate that commands are sent via this bus to each rangefinder to transmit a pulse in a certain sequence, and data is relayed back to the microcontroller via the same bus. Each rangefinder has a unique address on the bus, this is in order to identify commands and data from each specific sensor.

Load cell and payload interface circuit

The task of manipulating payloads, loaded on the payload interface, into position is accomplished by a single ATmega32-based microcontroller. This controller performs a feedback control loop actuating the payload interface motors via a speed controller according to data gathered from the load cells mounted underneath the payload interface. The physical circuit is represented graphically in Figure 5.18.

The standard strain gauge load cell connection produces two analogue outputs that need to be combined in order to interpret the effect of the deformation of the load cell through the strain gauges. These two outputs are in the range of 10 mV thus these need to be amplified in order to produce an output that the microcontroller is able to use. These signals are amplified as explained in Section 5.2.5. Each load cell requires a 5 V

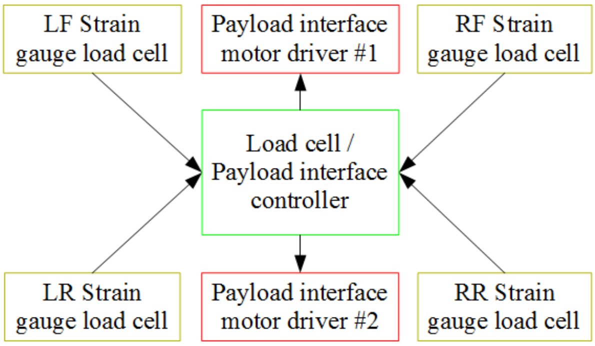

LF Strain gauge load cell
Payload interface motor driver #1
RF Strain gauge load cell
Load cell / Payload interface controller
LR Strain gauge load cell
Payload interface motor driver #2
RR Strain gauge load cell

Figure 5.18: Graphical representation of the load cell and payload interface controller circuit.

voltage supply that is supplied by the microcontroller. The speed controller controlling the payload interface motors take commands from digital I/O pins and take 5 V supply from the microcontroller.

Circuit integration

All three the circuits discussed in this section communicate with the onboard SBC via serial connections between the corresponding microcontroller and the SBC. The SBC is also in constant communication with the manufacturing system management server, via a *WiFi* connection, as discussed in Section 5.3. This configuration is illustrated in Figure 5.19.

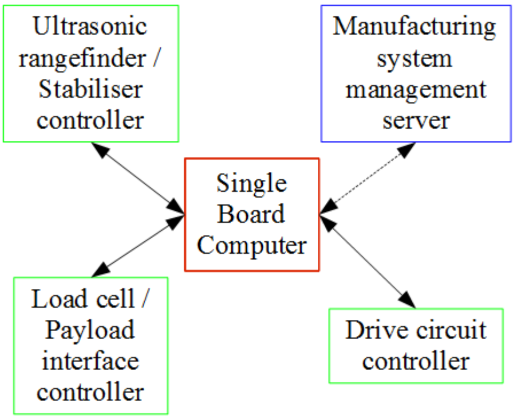

Ultrasonic rangefinder / Stabiliser controller
Manufacturing system management server
Single Board Computer
Load cell / Payload interface controller
Drive circuit controller

Figure 5.19: Graphical representation of the integration of the three circuits discussed.

In this figure, the dotted bidirectional arrow indicates the wireless connection. The sensors and actuation components have been omitted from this figure for simplicity. The serial connections between the SBC and the various microcontrollers make use of the UART communication protocol, while the wireless connection between the SBC and the remote management server utilises the WiFi communication interface with the Internet Protocol Suite (TCP/IP). More on TCP/IP can be found in Chapter 6.

5.7 Electronic System Safety

A FMECA was also carried out for the three main circuits that make up the electronic system onboard the platform. The procedure followed here is exactly the same as the one followed for the mechanical architecture in Section 4.4 and documented by Blanchard and Fabrycky [6]. Tables 5.2, 5.3, and 5.4 summarise the outcomes of the analysis for each circuit.

FMECA Results

When comparing the average RPN values for the three circuits it is evident that failure of the drive controller circuit or the ultrasonic rangefinder and static stability system circuit is more critical than failure of the load cell and payload interface circuit. The reason for this is that the drive and static stability systems directly affect the ability of the platform to actively maintain its stability, while the payload interface system only performs a complementary function.

In the case of failure of any component of the load cell and payload interface circuit the platform can convert to permanently statically stable and continue operation as normal, as long as the failure is detected timeously. However, in the case of the failure of one of the other circuits, the result could be catastrophic, where human safety could be compromised.

The average estimated frequency of these failure modes is relatively low due to the fact that the electronics are fixed to the physical platform structure and the circuits will be stationary relative to the physical platform structure. As long as the initial connections are made securely the occurrence of this kind of failure is highly unlikely. Routine inspection and maintenance will also assist in lowering the occurrence of these failures.

As mentioned in the mechanical architecture FMECA, safety functions built into the software system address some of these failure modes. An example of this is the performance

Failure modes	**Potential cause**	**Potential effect(s)**	**Frequency**	**Severity**	**Probability**	**RPN**
Loss of connection with SBC	Wire breakage Loose connection	Loss of communication with SBC	1	8	5	40
Loss of power to microcontroller	As above	Loss of ability to maintain active stability Loss of payload	1	9	6	54
Loss of data from sensor(s)	As above	Loss of ability to maintain active stability Loss of payload	1	9	6	54
Detachment of encoder(s)	Broken shaft Loose setscrew	Loss of ability to maintain active stability Loss of payload	1	9	4	36

Table 5.2: FMECA for the drive controller circuit.

Failure modes	Potential cause	Potential effect(s)	Frequency	Severity	Probability	RPN
Loss of power to microcontroller	Loose connection Wire breakage	Loss of control over sensors and static stability units Obstacle avoidance will be impossible	1	8	7	56
Loss of connection with SBC	As above	Obstacle avoidance will be impossible Loss of coordination of static stability unit movement	1	9	6	54
Loss of control over static stability unit(s)	As above	Safe loading of payloads will be impossible	2	7	3	42
Loss of data from sensors	As above	Obstacle avoidance will be impossible	1	7	5	35

Table 5.3: FMECA for the ultrasonic rangefinder and static stability system circuit.

Failure modes	**Potential cause**	**Potential effect(s)**	**Frequency**	**Severity**	**Probability**	**RPN**
Loss of connection to SBC	Loose connection Wire breakage	Loss of ability to coordinate payload positioning	1	3	6	18
Loss of power to microcontroller	As above	Positioning of payload will be impossible – Ability to maintain active stability will be seriously impaired	1	4	7	28
Loss of data from sensors	As above	As above	2	3	6	36
Loss of control over motor(s)	As above	As above	2	3	5	30

Table 5.4: FMECA for the load cell and payload interface circuit.

of a diagnostics sequence by the SBC at every start up in order to establish whether connections with all microcontrollers are secure and functional. More on these safety functions can be found in Chapter 6.

5.8 Chapter Summary

This chapter documents the selected electronic hardware components including justification for their selection. The components have been categorised and discussed in terms of the function that they perform onboard the platform. The logical layout and integration of the various electronic components, into the overall electronic system, has also been documented. The electronic system has been designed to be frugal in terms of power consumption in order to maximise operation time and minimise down time for battery charging.

Chapter 6

Software Architecture

The final major component, of the complete mechatronic system, to be discussed is the software system. This chapter documents the design and integration of the various software components that make up the software system. The sequence of discussion begins at the lowest level of control and follows a bottom-up approach. The mathematics underlying the system dynamics and control is discussed concurrently with this.

6.1 Data Fusion and State Estimation

At the lowest level of the software hierarchy is data processing for use at higher levels. In this case, data produced by the inertial sensors possess inherent uncertainties that may cause errors further along in the control system. These uncertainties may be the result of external disturbances due to normal random variations in environmental conditions that affect the measurement system [19]. The technique implemented to curb these uncertainties is known as Kalman filtering [26]. This technique can be described as a recursive-predictive state estimation technique, and is often also used for data fusion, where raw data from multiple sensors are used to produce usable data.

As documented in Chapter 5, uncertainties do exist in the outputs of both the dual-axis accelerometer and the solid state gyroscope. These take the form of sensitivity to high frequency noise in both cases. Less so in the case of the gyroscope, however this sensor is also prone to output drift. High frequency noise was apparent in both the accelerometer test results and gyroscope test results. This phenomenon was more noticeable with the accelerometer however, where the variations were more extreme and more frequent.

A key feature of Kalman filtering is that it only makes use of the estimation of each state from the previous time step and the most recent measurement from the relevant sensors to calculate and predict the state for the current time step. This implies a substantial saving in memory requirements as compared to state estimation techniques that are based on statistical data. Kalman filtering can thus be implemented on a low memory capacity controller such as the microcontroller described in Chapter 5.

Two states need to be estimated for this application, these are the tilt angle of the platform from the vertical, state a, and the tilt rate, state b. The state vector, x is represented in Equation 6.1.

$$x = \begin{bmatrix} a \\ b \end{bmatrix} \tag{6.1}$$

In this case, state b is the time derivative of state a. This characteristic may be used in the Kalman filter algorithm to improve the estimations of these states. The standard notation for a state vector estimation is shown in Equation 6.2.

$$\hat{x} = \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} \tag{6.2}$$

At the current time step, k, the estimated states are predicted, using the system model and the updated estimates from the previous time step, $k-1$. The estimate error covariance is also predicted at the current time step using the estimate error covariance from the previous time step. This covariance along with other elements, is used to calculate the Kalman gain at the current time step.

The Kalman gain is used, along with the predicted state estimate and the actual measurement at the current time step, to update the state estimate at the current time step. The calculation to update the estimate error covariance, at the current time step, also uses the Kalman gain at the current time step, along with the predicted estimate error covariance. Figure 6.1 shows a graphical representation of the Kalman filter algorithm.

In this figure:
$\hat{x}_k^-$ is the predicted state estimate at the current time step,
$\hat{x}_{k-1}$ is the state estimate at the previous time step,
u_k is the control input at the current time step,
A and B are the system model matrices,
P_k^- is the predicted estimate error covariance at the current time step,

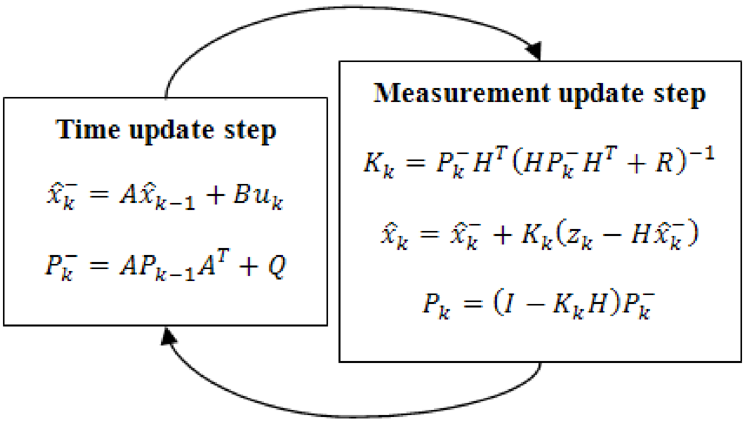

Time update step
$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$
$P_k^- = AP_{k-1}A^T + Q$
Measurement update step
$K_k = P_k^- H^T(HP_k^- H^T + R)^{-1}$
$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$
$P_k = (I - K_k H)P_k^-$

Figure 6.1: Graphical representation of the Kalman filtering algorithm.

P_{k-1} is the estimate error covariance at the previous time step,
Q is the process, or system, noise covariance.

K_k is the Kalman gain vector at the current time step,
H is the measurement system model matrix,
R is the measurement noise covariance,
$\hat{x}_k$ is the state estimate at the current time step,
z_k is the actual measurement at the current time step,
P_k is the estimate error covariance at the current time step.

Initialisation of this algorithm requires initial values for the state estimate and the estimate error covariance. The output of the filtering algorithm after each time step is the vector of state variables at the end of the measurement update step. This output is fed into the feedback control system controlling the active stability of the platform.

Due to the lean operation of the Kalman filtering technique it requires a certain period to settle to a stable condition at every initialisation. This is due to the fact that the technique does not make use of the entire history of the system output, only the most recent state variables. The platform tilt angle was logged during a test to determine the settling time of the Kalman filter that has been implemented here, the data was plotted versus time, the results are shown in Figure 6.2.

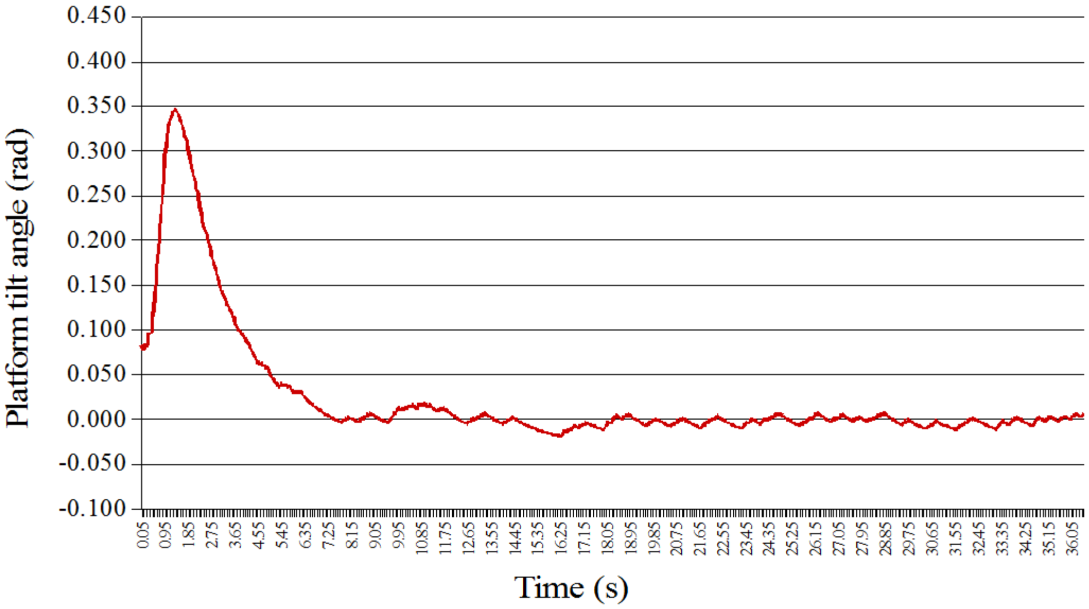


Figure 6.2: Platform tilt angle versus time illustrating the settling time of the Kalman filtering technique.

For this test the platform was held upright in a vertical orientation with the tilt angle measured to be zero, where the Kalman filter settled after a period of approximately 18 sec. This settling time is taken into account in the feedback control system code in order to minimise disturbances in the control input to the motors at the initialisation of the control system. Although this period seems relatively long, it can be easily incorporated into the operation of the platform by starting the Kalman filter algorithm before the balance function is invoked and thus before the static stability system is retracted at startup. The Kalman filter algorithm can also be kept running while the platform is docked at a manufacturing cell I/O port in order to avoid the settling period at initialisation.

The Kalman filter has been implemented in this project in the form of a discrete Kalman filter in order to be implemented onboard the drive system microcontroller. In this form of the filter, the matrix A takes the form of the Jacobian matrix of partial derivatives of the plant matrix with respect to the state vector [49]. Matrix B takes the form of the Jacobian matrix of the system input matrix with respect to the state vector. Hand optimisation of the C code by precalculating constant variable and hard coding these into the control program and only calculating non-zero matrix entries. This was done to minimise CPU time in the microcontroller. Refer to Appendix E for the code.

6.2 Control Architecture

The control architecture for the platform consists of two hierarchical levels. The control system that is tasked with actively maintaining the stability of the platform during operation resides at the lower level. At the level above this is the motion planning scheme for the platform that takes into account the specific kinematic constraints of the platform.

6.2.1 Operational Platform Control Sequence

The control strategy, designed to control the dynamics of the platform during different stages of operation, resides one level above the data fusion and state estimation algorithm. The main component in this strategy is the feedback control system that controls the speed and direction of the drive wheels in order to maintain stability. Complementary to the main component are the control systems that control the extension of the static stability units and the payload interface system rollers.

These systems are activated in a specific sequence which depends on the phase of operation. In the three sections subsequent to this, these systems are discussed in the sequence that they are implemented, from the moment that the platform approaches an I/O port of a machine cell for receiving a payload. The state that the platform is in at the point of departure in this sequence is in active stability mode.

6.2.2 Static Stability System Control

This component of the control strategy is activated as the platform approaches or departs from a machine cell material I/O port, that is, the points where materials enter or leave the machine cell, respectively. When the platform enters a certain radius from the I/O port the static stability units are extended to the floor as the platform approaches the I/O port. The timing is such that the units reach the floor as the platform reaches a smaller radius from the port. This is to ensure that the platform is completely statically stable as it interfaces with the port. In order to avoid interference of the active stability control system by the static stability system, the active stability control system is suspended for the period that the static stability units are in contact with the floor.

As soon as the platform is stationary, the remote management server is signalled that the platform is ready to receive a payload which informs the machine cell control system. Once the payload is loaded onto the platform the payload interface system control is activated. The function of this control system is discussed in the subsequent section below. For the purpose of the static stability system control this simply represents a delay in operation while the payload interface controller performs its function. As soon as this function is completed, the platform is ready to depart for the destination of the payload as specified by the remote management server.

Once the ready signal is relayed the platform departs. As soon as the platform exits the smaller radius from the I/O port the static stability units are retracted and the active stability control system is reinstated as the static stability units break contact with the floor surface. Figure 6.3 shows a graphical representation of the static stability system control operation.

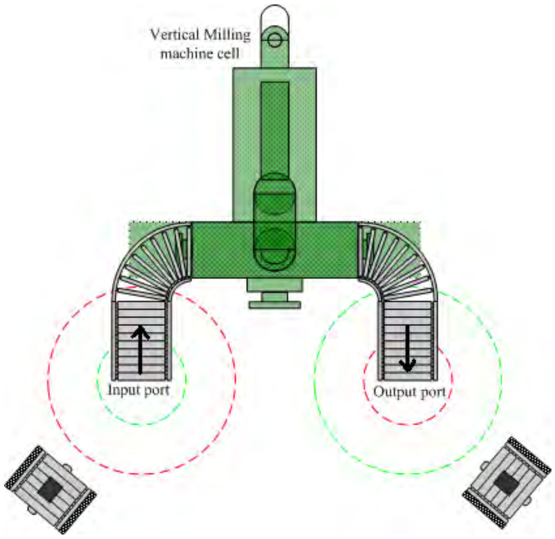


Figure 6.3: Graphical representation of the operation of the static stability system.

The red dotted circle at the input port represents the radius where the static stability units are extended and the green circle represents the radius where the units make contact with the floor surface. At the output port, the red dotted circle is where the static stability units are retracted and the green circle is where the units should be fully retracted. The extension and retraction of the static stability units while the platform is in motion enables a smooth transition from dynamic stability to static stability and vice versa. The reason for this is that the platform does not sway while it is travelling at a constant velocity unless it is disturbed by an external source.

This control system takes the shape of an open loop controller in the upward direction. In the downward direction, the extension of the units is limited by the activation of a limit switch mounted at the correct position for the units to halt at the correct length. In the upward direction, the position of the static stability units is based purely on the time that the actuating motors are driven. The dynamics of these units allow for this. The ratio of extension to rotation of the power screws is accurate and there is very little tolerance between the shaft and the collar of the power screw. The motors also have very little inertial effects that may cause deviation in the position of the power screws. Only one limit switch is required due to the fact that both units are identical, and they are driven by the same source and command. The code produced to achieve the control of these systems is included in Appendix E.

6.2.3 Payload Interface System Control

The payload interface system controller is designed to actuate the motors that drive the two sets of rollers of the payload interface such that the payload centre of gravity is aligned with the vertical centre plane of the platform in the forward-backward direction. This control system is activated as soon as the platform makes contact with the machine cell I/O port, and the payload is loaded.

The position of the payload centre of gravity is calculated using data fed back from the strain gauge load cells mounted at the corners of the payload interface. This implies that the system takes the shape of a feedback control system. As soon as the payload is loaded, the amplified and calibrated measurements from all the load cells are added together and the weight of the payload is determined.

The analogue outputs from the load cells are amplified using instrumentation amplifiers. The amplified outputs are then converted to digital values by the ADC of the controlling microcontroller. The calibration of these values is done in a two step process. The first step is a taring step, where the output of each load cell is set back to zero before the payload is loaded. This is done to ensure that the weight of the payload interface structure does not affect the measurements. The second step is the actual measurement step where the weight distribution of the payload, across the payload interface system, is determined.

The weight of the payload and the distribution between the forward and rear mounted load cells are used to determine the position of the payload centre of gravity. The roller motors are actuated according to this information, which is continuously updated as the payload is manipulated. Refer to Section 4.2 for the mechanism of manipulation of a loaded payload. Figure 6.4 illustrates the flow of information in the payload interface control system.

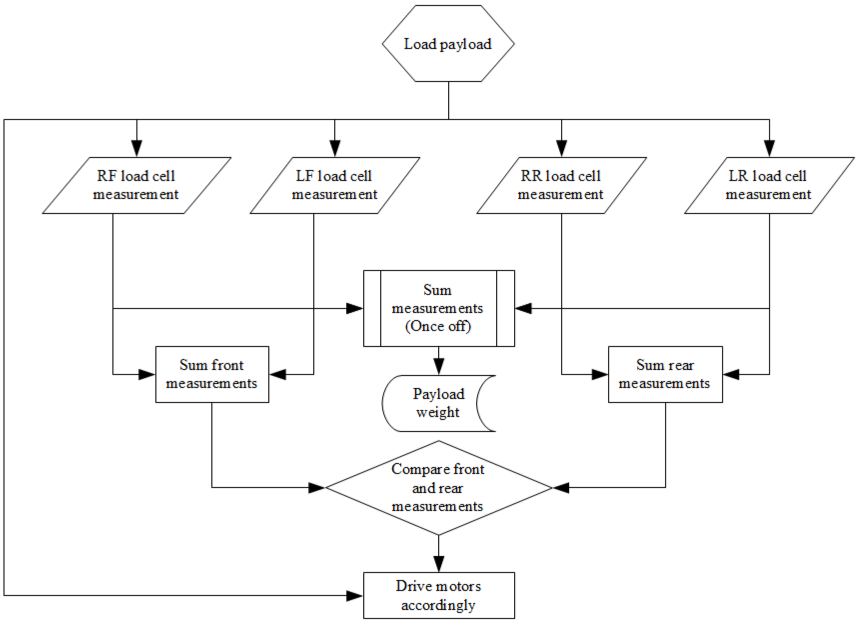


Figure 6.4: Flow chart illustrating the flow of information and commands in the payload interface system control loop.

As soon as the payload is in position, the remote management server is signalled and the platform may depart to the predetermined destination of the payload. The sequence of departure as described under the static stability system control subheading is also followed here. The code produced to achieve the control of this system can be found on the Compact Disk accompanying this document.

6.2.4 Active Stability Control

The active stability control of the platform is achieved by actuating the drive motors according to an algorithm that utilises data from the wheel encoders, accelerometer, and rate gyroscope. This control system is on the same level in terms of the control hierarchy as the control systems discussed thus far in this section.

In order to keep the control algorithm as simple as possible a linear control design approach was taken. Important criteria that this control system needed to fulfill have been derived from the technical design specifications, and are listed below:

- It must be robust enough to handle variations in system characteristics.
- It must be as smooth as possible.
- It must make full use of available sensor data.
- It must be as energy efficient as possible.

Two effective control system design techniques used for this type of system are the Linear Quadratic Regulator (LQR) design technique, and the Pole-placement design technique. More information on both these can be found in Franklin et al. [20]. Based on the above criteria, the LQR design technique was used to produce a robust, energy efficient negative feedback control system that makes use of the encoder, accelerometer, and gyroscope data. Four state variables are derived from the data mentioned, and fed back into the control system. This design technique is based on optimal feedback control theory. The LQR design technique is an effective and widely used technique of linear control system design [20].

Generally, optimal control is concerned with minimising some performance index, or cost function, of the system trajectory. This performance index may incorporate cost of energy exerted, settling time, or some other maximum allowable magnitude of a system variable or constraint. The variables incorporated in the performance index must be related to the the equation of motion of the system [42].

The states variables that were selected for use in the control system are the linear displacement, x, the linear velocity, $\dot{x}$, the tilt angle, θ, and the tilt rate, $\dot{\theta}$. The linear displacement and velocity are calculated from the wheel encoder data, while the tilt angle is determined from the accelerometer data, and the tilt rate is determined from the rate gyroscope data. When stating here that a state is determined from data from a specific sensor this means that the state is based on this sensor however the final state is estimated using the technique described in Section 6.1.

The application of this technique involves minimising a performance index, J, for the time that the control system is in operation. The performance index used in the LQR technique is shown in Equation 6.3.

$$J = \int_0^\infty \left[\rho y^2 \left(t\right) + u^2 \left(t\right)\right] dt \tag{6.3}$$

In this expression, x represents the state space output matrix while u represents the control input to the control system as defined by the classical negative feedback control law. ρ is some user-specified weighting matrix. These are shown in Equation 6.4 below.

$$\begin{aligned} y &= Cx + Du \\ u &= -Kx \end{aligned} \tag{6.4}$$

Here the matrices C and D are system matrices and K is the control input gain matrix. The main objective of this technique in minimising the performance index is to produce the most efficient control input required to drive the system states to zero as quickly as possible. In this case the control input is the voltage level supplied to the drive wheels.

A mathematical model was developed from a simplified approximation of the physical system. The development of this model can be found in Appendix D, this includes an explanation of the system approximations. The governing equations of motion that were produced from the mathematical model were combined in state space form and is written in matrix format in Equation 6.5.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = A \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + BV_a \tag{6.5}$$

System matrices A and B are expanded in Equation 6.6 in terms of the system characteristics used in the mathematical model of the system.

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{2k_m k_e\left(M_p l r - I_p - M_p l^2\right)}{R r^2 \alpha} & \frac{M_p g l^2}{\alpha} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2k_m k_e\left(r\beta - M_p l\right)}{R r^2 \alpha} & \frac{M_p g l \beta}{\alpha} & 0 \end{bmatrix} \\ B &= \begin{bmatrix} 0 \\ \frac{2k_m\left(I_p + M_p l^2 - M_p l r\right)}{R r \alpha} \\ 0 \\ \frac{2k_m\left(M_p l - r\beta\right)}{R r \alpha} \end{bmatrix} \end{aligned} \tag{6.6}$$

Equation 6.7 shows the α and β variables in terms of the system characteristics.

$$\begin{aligned} \beta &= 2M_w + \tfrac{2I_w}{r^2} + M_p \\ \alpha &= I_p\beta + 2M_p l^2 \left(M_w + \tfrac{I_w}{r^2}\right) \end{aligned} \tag{6.7}$$

Refer to Appendix D for the definition of the terms found in these equations. In this model the input voltage to both drive motors are the same, as the aim of this control system is maintaining the active stability of the platform by driving all the states to zero and not controlling the trajectory of the platform. The trajectory control is covered in Section 6.2.5.

Simulations were run in Matlab to test the effectivity and robustness of the feedback control system using the LQR design technique. The system variables were entered and calculated in a Matlab script file where the feedback control system was implemented. The controllability of the system is also determined in this script file through the use of a controllability matrix test.

A Matlab function specifically designed to generate LQR gains was used to calculate the feedback gains for this system. Information required by this function include the system matrices A and B, and user-specified weighting matrices Q and R. The matrix Q corresponds to the system matrix A and is of the same size, while the size of the R corresponds to the number of inputs to the system. The expression for the performance index is generalised in matrix form to resemble Equation 6.8.

$$J = \int_0^\infty \left(\mathbf{x}^\mathbf{T}\mathbf{Q}\mathbf{x} + \mathbf{u}^\mathbf{T}\mathbf{R}\mathbf{u}\right) dt \tag{6.8}$$

For these simulations the size of R is one-by-one corresponding to the single input; this is due to the fact that the same input is used for both motors. Typically, matrix Q is a diagonal matrix with each of the entries on the diagonal corresponding to the system state in the same row of the state vector. Thus Q is of size four-by-four. Bryson's Rule, shown in Equation 6.9, may be used for finding acceptable initial values for the entries of Q and R [10].

$$\begin{aligned} Q_{ii} &= 1/maximum\ acceptable\ value\ of\ \left[x_i^2\right] \\ R_{ii} &= 1/maximum\ acceptable\ value\ of\ \left[u_i^2\right] \end{aligned} \tag{6.9}$$

The weighting matrices may be modified subsequently through iteration in order to find an acceptable balance between performance and control effort [20].

The system response was simulated for five different overall system weights. These different system weights are aimed at simulating variations in payload weight. The unladen platform weight is estimated to be approximately 50 kg. The simulations were ran starting with a 50 kg overall system weight, corresponding to the unladen case, in intervals of 5 kg up to the maximum overall system weight of 70 kg, corresponding to the maximum payload weight of 20 kg. The objective of this exercise was to determine the effect of the variation in overall system weight, i.e. payload weight, on the control system response.

A variation that has been implemented in the application of the LQR designed control system is to supply a reference input for the linear displacement state for the system to follow. A unit step input was selected, in order to investigate and compare the system response with the various overall system weights. For each simulation the system variables and feedback gains were generated using the Matlab script file where the mathematical model is described. These variable were then imported into a functional block diagram in Simulink, where the simulations were ran. The Matlab script file and the Simulink block diagram can be found in Appendix D. The entries to the Q and R weighting matrices were iterated to produce the most acceptable response for all system weights.

The four state variables of most importance, the linear displacement, linear velocity, tilt angle and angular velocity, were plotted for the five system weights in four separate graphs in order to compare the responses. These are shown in Figures 6.5, 6.6, 6.7, and 6.8. These data are used to determine the sensitivity of the control system to variations in system characteristics, specifically platform weight, in order to establish whether an adaptive control strategy is required. If the variation in system response corresponding to the foreseen variation in platform weight is too large then an adaptive control system is required. If the variation in response is within acceptable limits, then the robust control system is feasible.

Figure 6.5 shows the linear displacement response compared to the reference input of a unit step at the start of the simulation. It can be seen here that the variation in system response as a result of the variation in system weight is relatively small. Analysis of the graph shows that the 10 % settling times only vary by approximately 0.75 sec throughout the range of weights. The shortest settling time is produced by the lowest overall system weight, while the highest overall system weight produces the longest settling time.

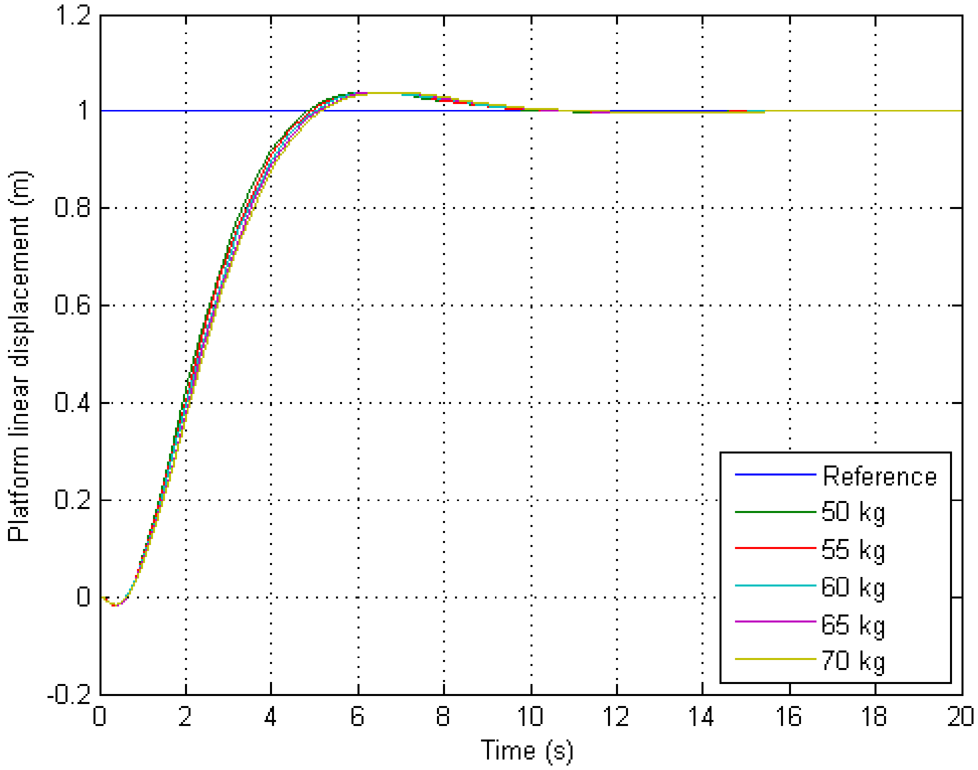


Figure 6.5: Simulated displacement state response to a unit step input.

Figure 6.5 also shows that the peak times only vary by approximately 0.5 sec. The same correlation between peak times and system weights is found as with settling times. The peak values only vary by approximately 0.07 % of the reference value. The lowest overall platform weight yields a peak value of ± 3.96 %, with the 70 kg overall platform weight yielding a peak value of ± 4.02 %.

Figure 6.6 shows the linear velocity state response for the five system weights. At inspection of this graph, it can be seen that the velocity response peaks coincide very closely at approximately 1.8 sec. At 0.36 m/s, the highest peak corresponds to the lowest overall system weight, and the highest system weight produces the lowest peak velocity of approximately 0.33 m/s.

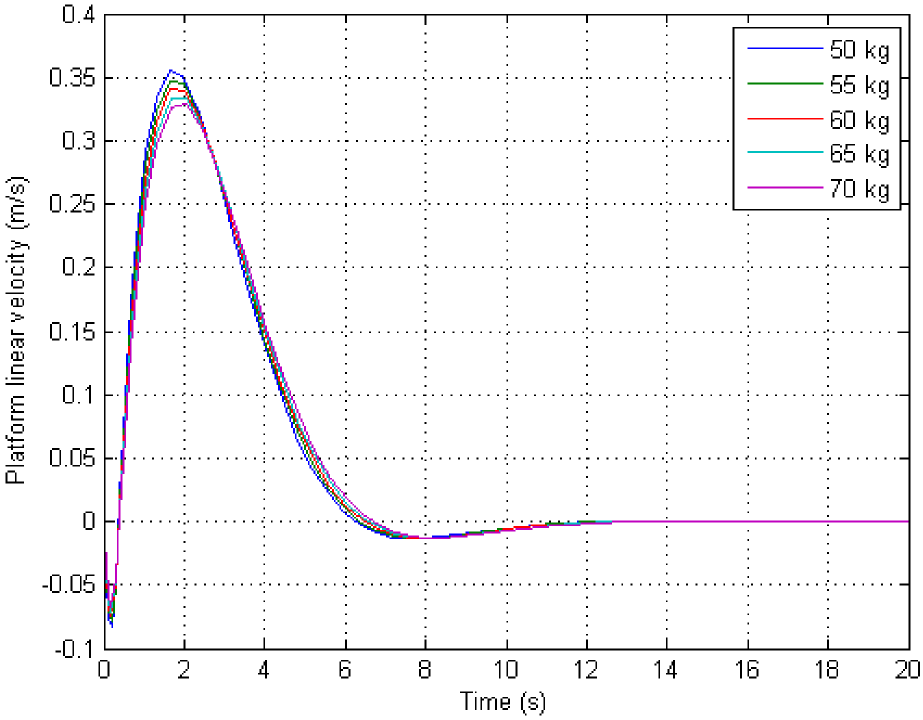


Figure 6.6: Simulated linear velocity state response.

The tilt angle state responses for the various system weights are shown in Figure 6.7. An initial observation can be made that the platform tilts backwards sharper than it tilts forward before travelling forward. This is also manifested in the displacement state in Figure 6.5, where it can be seen that the platform actually travels a maximum distance of approximately 0.02 m backwards before travelling forward.

The explanation for this phenomenon is that the body of the platform needs to gain forward momentum relative to the wheels, or support point, of the platform. This mechanism is taken into account when the platform is approaching an I/O port of an manufacturing cell by way of extending the static stability units while the platform is still in motion.

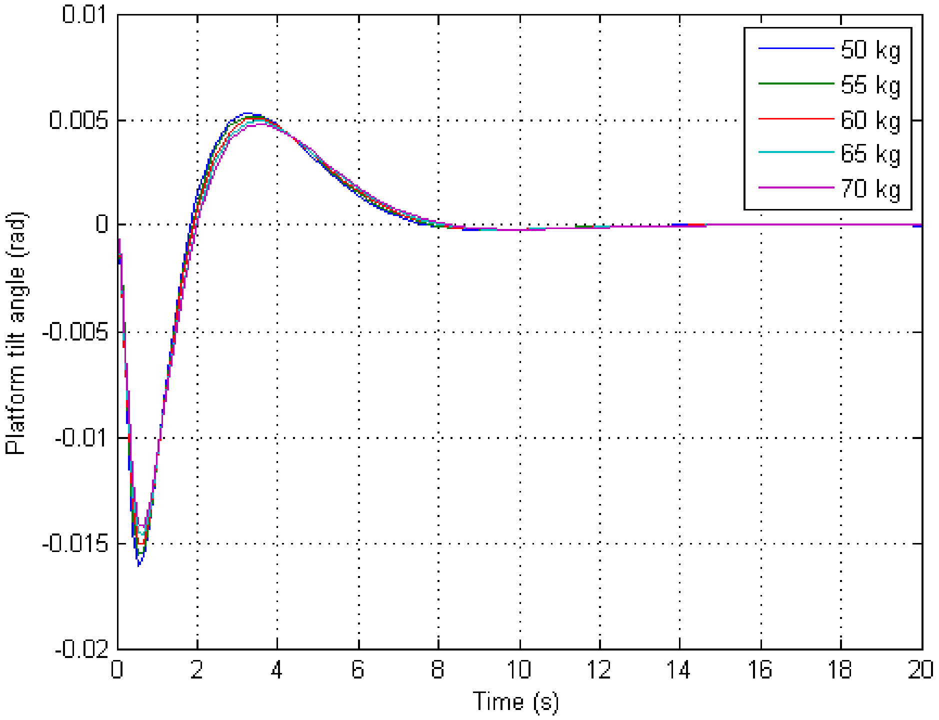


Figure 6.7: Simulated tilt angle state response.

The peak tilt angle varies from -0.016 rad, for the 50 kg system weight, and -0.01425 rad, for the 70 kg system weight. Thus the lowest system weight produces the highest peak tilt angle, while the highest system weight produces the lowest peak tit angle. This can be explained using theory of mechanics and dynamics; as more weight possesses more potential energy, a lower mass is required to tilt over more than a higher mass in order to gain the same amount of forward momentum [33].

The angular velocity state response is shown in Figure 6.8. A sharp initial peak in the negative direction is also seen when examining this graph. Peak angular velocity values range between approximately -0.049 rad/s under no-load conditions and -0.040 rad/s at maximum load. The angular velocity also settles more slowly than the tilt angle state.

From the linear displacement and tilt angle states it is evident that the response of the system slows down as the system weight increases. The linear velocity and angular velocity states also confirm this finding. This can be attributed to the increased inertial effect as the weight increases.

It is assumed that all the states are readily available for measurement in these simulations. In reality however this is not the case. States that are available for measurement, by the sensors present, are the linear displacement by the rotary wheel encoders, the tilt angle by the accelerometer, and the angular velocity by the rate gyroscope. In order to determine the linear velocity the time derivative of the displacement state is taken. The simulations were carried out using a full state feedback control system, while the physical implementation can be described as an observer state feedback control system [20].

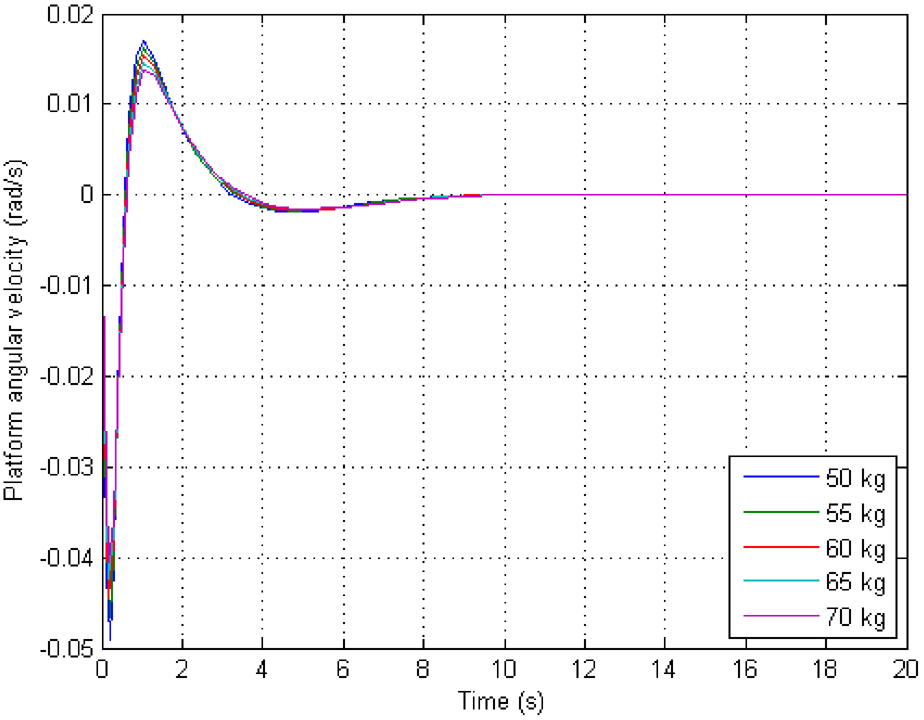


Figure 6.8: Simulated angular velocity state response.

The physical control system is programmed on an ATmega32 based microcontroller, as discussed in Chapter 5. Due to the relatively small memory capacity of microcontrollers it was necessary for the programming to be done as lean and efficient as possible. All matrix algebra calculations that could be done before programming were done and the numbers were hard coded into the control system program.

Figure 6.9 shows the symbolic block digram of the physical implementation of the control system. As shown here, it takes the shape of a traditional negative feedback control system. The reason for selecting four states for the control system as opposed to only tilt angle and angular velocity is in order for the trajectory of the platform to be controlled by the remote management system via the onboard SBC. More detail on this process can be found in Section 6.3.

Here the main concern is the interpretation of the input commands by the control system. As shown in Figure 6.9 the input to the control system is a reference linear displacement as commanded by the *Player* device management software onboard the platform. There are two displacement terms inserted, one for each drive wheel. This way the trajectory can be controlled as described in the next section.

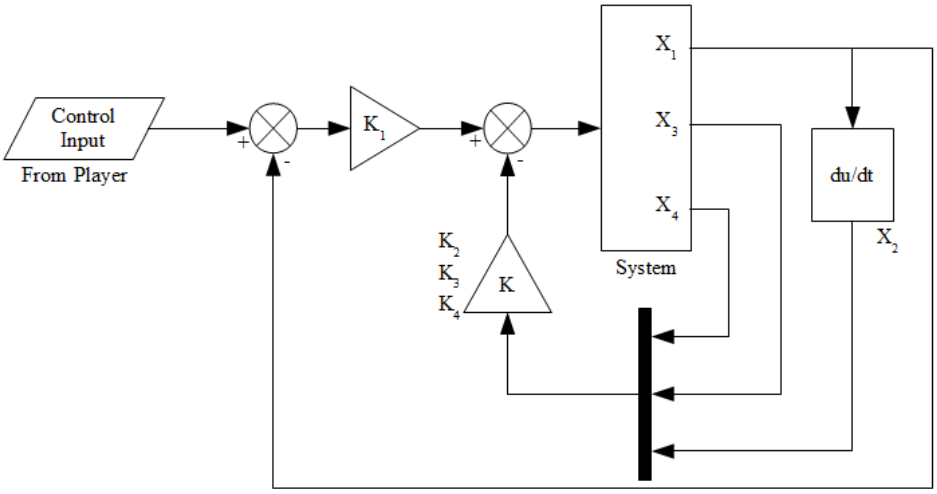


Figure 6.9: Symbolic block diagram of physical control system implementation.

Due to the fact that the LQR design technique makes use of system characteristics to calculate the feedback gains these calculations may be done before implementation in the microcontroller program. The fact that microcontrollers operate in the discrete domain, necessitates that the LQR design technique is transformed to the discrete domain before implementation.

In the discrete domain the performance index for the LQR control system takes the shape as shown in equation 6.10. The elements in the summation are all the same as in the continuous domain integration, however they are taken at discrete time steps where k represents the current time step.

$$J = \sum_{k=0}^{N-1} \left(\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}\mathbf{u}(k) \right) T \tag{6.10}$$

In order to solve for the quadratic performance index, with a time-invariant system the Hamilton-Jacobi partial differential equation is produced which takes the form of the matrix Riccati equation [11]. In the discrete domain the solution of the matrix Riccati equation, equation 6.11, solves for the gain vector, $\mathbf{K}$ in equation 6.12, using an intermediate square symmetrical matrix, $\mathbf{P}$ and the system matrices.

$$\mathbf{PA} + \mathbf{A^T} + \mathbf{Q} - \mathbf{PBR^{-1}B^TP} = \mathbf{0} \tag{6.11}$$

$$\mathbf{K}(N-(k+1)) = \left[T\mathbf{R} + \mathbf{B}^T(T)\mathbf{P}(N-k)\mathbf{B}(T) \right]^{-1} \mathbf{B}^T(T)\mathbf{P}(N-k)\mathbf{A}(T) \tag{6.12}$$

Matrix $\mathbf{P}$ is solved using the user specified weighting matrices $\mathbf{Q}$ and $\mathbf{R}$, along with the gain vector and the system matrices. From these calculations the control input can be expressed as shown in equation 6.13.

$$\mathbf{u}_{opt} = -\mathbf{K}(k)\mathbf{x}(k) \tag{6.13}$$

This expression is in the form of the traditional negative feedback control input. As with the control system simulations the LQR feedback gains may be pre-calculated and used in the control system. Fundamentally the active stability control system receives velocity commands from the SBC and actuates the drive wheels accordingly, while actively maintaining the dynamic stability of the platform. The code produced to achieve this can be found in Appendix E.

Tests were carried out in order to validate the effectiveness of the LQR designed control system. For this purpose a traditional Proportional-Integral-Derivative (PID) control system was also produced in order to compare the results. The objective set for the tests was for the platform to balance in position for a period of 15 sec. In practice the platform will not be required to balance in position, however this is a good method for comparing the performance of the control systems.

The system states, which include tilt angle, angular velocity, linear displacement, and linear velocity, were logged for a period of 15 seconds for both control systems at a rate of two Hertz as measured by the onboard inertial sensors and shaft encoders. The results have been plotted against time and are discussed in this section.

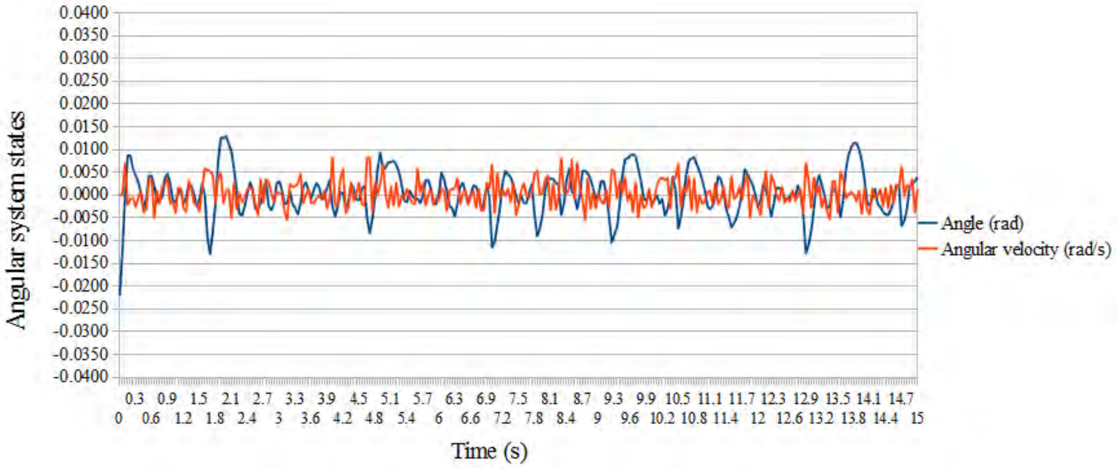


Figure 6.10: Test results for angular system states from the LQR based active stability controller.

Figure 6.10 shows the tilt angle and the angular velocity of the platform. The tilt angle state is measured to be −0.02 rad at the start of the data set due to the fact that the Kalman filter had not settled completely at this point. It does however settle shortly after the start of the graph. After this initial peak value the tilt angle does not deviate more than ±0.0125 rad from the vertical at zero radians. This value is within the limits of tilt of ±0.088 rad, specified at the design specification phase of the project.

The angular velocity of the platform as measured by the gyroscope and processed by the Kalman filter peaks at approximately 0.01 rad/s at various points during the test. It can be seen from the graph that the angular velocity of the platform oscillates more than the tilt angle of the platform. The graph also shows that the angular velocity of the platform returns to zero and subsequently changes sign when the tilt angle peaks.

Figure 6.10 indicates that the platform oscillates in a stable manner. There are however instances where the platform tilts in one direction and then back in the other direction to the extremes before stabilising itself about the vertical. An explanation for this could be a lag in the Kalman filter. Although the Kalman filter is a very effective state estimation tool, it is very complex and needs to be tailored to each application and tuned for the specific conditions in which it is implemented. This can be a long and cumbersome exercise due to the fact that there are not many tuning rules widely available, or applicable. This lag phenomenon can also be seen in Figure 6.11, which shows the linear system states from the same test.

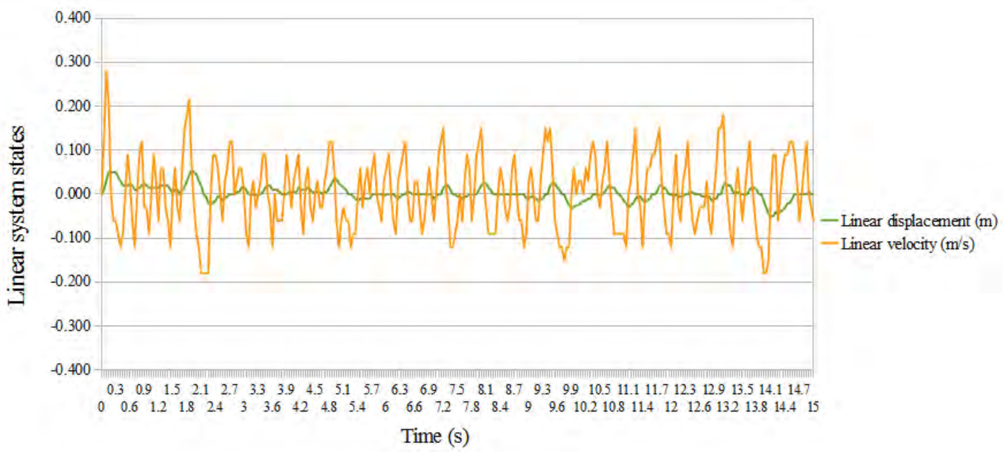


Figure 6.11: Test results for linear system states from the LQR based active stability controller.

In Figure 6.11 points can be seen where the linear displacement state deviates in one direction and back, slightly overshooting, and then stabilising at zero again. This is due to the platform having to regain its balance by sacrificing the position state. The reason for the platform sacrificing the displacement for the tilt angle is that the angular states are more heavily weighted in the LQR control system.

The deviation of the linear displacement of the platform, from the start position, is kept relatively low. This state peaks at approximately ±0.05 m. The deviation of both the linear states shown in Figure 6.11 at the start of the graph is due to the settling of the Kalman filter and the platform attempting to recover from the incorrect tilt angle at this point. The linear displacement state does however recover back to zero relatively quickly.

At the higher peaks of the linear displacement state it can be observed that the linear velocity state peaks and almost instantly changes direction as soon as the platform start moving back towards zero displacement. This behaviour is due to the fact that the velocity state is the time derivative of the displacement state. The linear velocity state peaks at approximately 0.21 m/s, not considering the peak at the start of the data set, and stays within 0.15 m/s for most of the test period.

There are noticeable differences between the response of the LQR based control system and the PID control system. Figure 6.12 shows the tilt angle and angular velocity states of the system from the PID control system test.

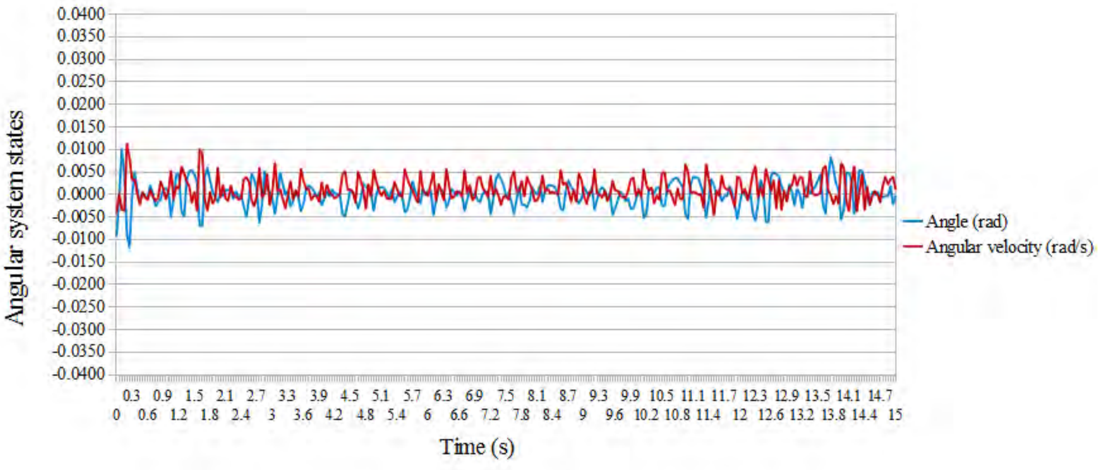


Figure 6.12: Test results for the angular system states from the PID active stability controller.

The first observation that can be made is that the platform oscillates much more under the PID controller than under the LQR based controller. When considering the variation of the angular states it is seen that the tilt angle varies between ±0.05 rad when not considering the initial peak at the start of the data set, while the angular velocity state varies between ±0.05 rad/s for most of the test period. The peak of the tilt angle state is approximately -0.075 rad, while the angular velocity peaks at approximately 0.01 rad/s.

Figure 6.13 shows the linear system states from the PID control test. This graph shows greater variation in linear states for the PID controller than for the LQR based controller. The linear displacement of the platform drifts away from the starting position. The reason for this is that the controller is not concerned with the position of the platform, only the tilt angle.

Figure 6.13 shows that the linear velocity of the platform peaks as high as 0.28 m/s,

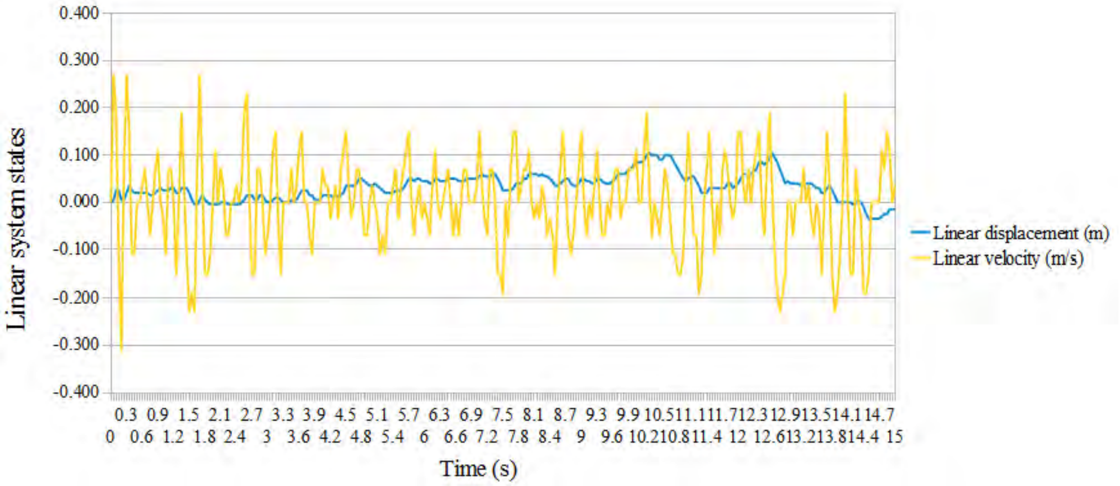


Figure 6.13: Test results for the linear system states from the PID active stability controller.

and the linear displacement is in a positive condition for most of the test period. It is evident from these observations that these states are not being explicitly controlled. When comparing the tilt angle and linear displacement states for the two tests it can be seen that the LQR control system response is much smoother than that of the PID control system response. An explanation for this could be that the LQR based control system was designed to optimise the control input and thus uses less energy by taking smaller corrective action.

Although the PID controller performs better in controlling the tilt angle of the platform it is only concerned with controlling the tilt angle. In the case of the LQR based controller it is constantly working towards driving all the states towards the set reference point. If a position controller were to be incorporated with the tilt angle PID control loop, a nested feedback control system would be the result. This solution is much more complicated than the solution of using the LQR based controller and simply providing it with a reference for the linear displacement state as a function of time.

Improvements can be made to the response of the LQR based control system, by improving the mechanical and sensory system as well as better tuning of the Kalman filter. There is some rotational tolerance in the coupling between the drive shafts of the motors and the shafts connected to the drive wheels. This contributes to the oscillatory response of the platform in both tests. More effective sensor solutions are available for the application that the inertial sensors are employed in this project, however these are much more expensive. As mentioned earlier the correct implementation and tuning of the Kalman filter is a crucial aspect in sensor fusion and state estimation applications.

When comparing the source code produced for implementing the LQR based controller and the PID controller, on the accompanying Compact Disk, it is obvious that the PID controller is already a more complicated control algorithm than the LQR based control loop. This is due to the fact that most calculations required by the LQR design method can be done before the control system is implemented, as long as the system characteristics are known, while the calculations for the PID control system need to be done online.

6.2.5 Motion Control

One level above the operational platform control level in the hierarchy of control is the control of the platform motion. In this section it is assumed that the operational control sequence functions effectively. As with the operational platform control sequence the platform motion control strategy consists of two phases, which are also categorised according to the area of operation. In mobile robotics, the feasibility of motion planning strategies and algorithms depends on the constraints that govern the motion of the specific robotic platform. The kinematics associated with the configuration of the platform drive system are discussed here ahead of the applied motion control strategy.

Differential Drive Kinematics

Differential drive kinematics imply that differential velocity constraints are imposed on the platform. In a system such as the one under investigation there are less action variables than there are degrees of freedom. Such a system is referred to as an under-actuated system. Figure 6.14 illustrates the differential drive configuration. Local and global Cartesian coordinate systems are also indicated, along with the direction of rotation. The origin of the local coordinate system is situated at the centre of the axis that the wheel axles form.

With the actuation vector defined as $u = (\omega_r, \omega_l)$ employing the angular velocities of the right and left wheel respectively, a model of the differential constraints may be developed. Transforming between linear wheel velocities and angular wheel velocities is a simple task. For this model the constants L, the total axle length, and r, the wheel radius, are used. Equation set 6.14 shows the configuration transformation equations applicable

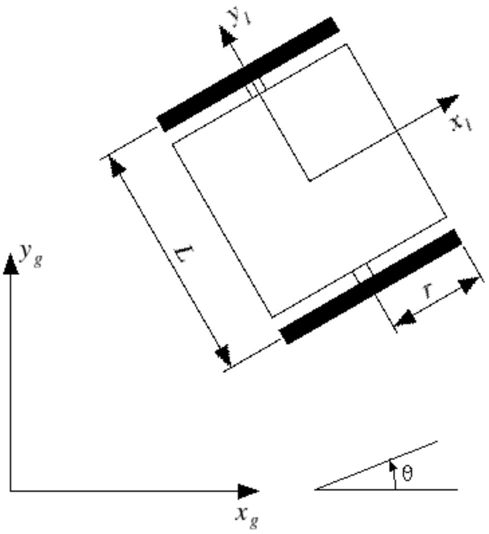


Figure 6.14: Illustration of the differential drive configuration.

for this system, [29].

$$\begin{aligned}\dot{x} &= vcos\,(\theta) \\ \dot{y} &= vsin\,(\theta) \\ \dot{\theta} &= \tfrac{r\Delta\omega}{L}\end{aligned} \tag{6.14}$$

where $v = \frac{r}{2}\,(\omega_r + \omega_l)$, and $\Delta\omega = (\omega_r - \omega_l)$, [29].

Using this configuration transformation it is possible to describe the motion of the platform in all degrees of freedom in terms of the actuation variables available.

Motion Planning and Control

The selection of the optimal motion algorithm to be implemented in a specific situation depends on the system in question in terms of the type of drive system as well as the type of environment that the robot is to operate in. In this case, as discussed in the previous section, a differential drive robot is under investigation. The environment is a manufacturing system environment with as little human interface as possible and with other robotics mobile platforms performing similar tasks.

To this end recent research conducted, by Walker [48], in this area has determined that a motion control algorithm that produces favourable robot responses is an algorithm that replaces the Cartesian coordinate system of the differential drive robot with a polar coordinate system [3]. This algorithm was successfully incorporated into the *Player* device management software, discussed in the following section, by Walker.

The polar coordinate transformation algorithm transforms the Cartesian coordinates to polar coordinates relative to the goal position and orientation of the platform. This is done in order to overcome certain control obstacles imposed on some drive systems, in this case the differential drive system. Due to the nonlinear nature of the drive system the posture, i.e. the two-dimensional position and orientation, is not stabilisable, in terms of the position and orientation, using linearised models and linear controllers.

Although this algorithm does impose some restrictions on the implementation there of in a controller, it does produce favourable results with careful tuning at the implementation stage [48]. With the implementation of this algorithm for posture stabilisation in the motion controller the motion of the platform can be predicted reliably.

6.3 Device Management

Central to the data management onboard the platform is the SBC, and the software installed on it. The operating system running on the SBC is the Linux distribution known as Fedora 7. Reasons for the selection of the Linux operating system include that it is open source software, it is easily customisable to suit specific applications, and that it supports the *Player* device management software.

The *Player* device management software has been developed under the open source Player Project [43] [13]. Essentially the Player Project is a generic programming framework for use in a variety of robotic applications. Due to the open source nature of the project it is highly flexible and may be extended or modified to suit many applications. *Player* serves as a hardware abstraction layer between the robotic, or electronic, hardware and the higher level remote management system.

Figure 6.15 shows a graphical representation of the deployment of *Player*. In this figure "IPR1" represents the physical platform. *Player* supplies a standard network interface for client programs, e.g. remote management software, to subscribe to the onboard *Player* server in order to command the robot, in this case the platform, and gather information from it.

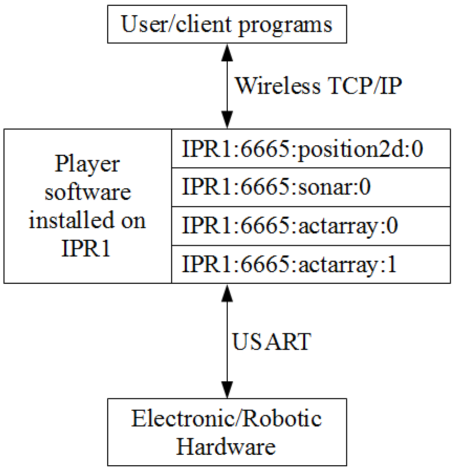


Figure 6.15: Graphical representation of the deployment of the *Player* device management software.

Commands are relayed to electronic hardware through the use of device drivers, analogous to desktop computer drivers for peripheral devices. As it stands *Player* supports a large range of physical and abstract devices, from microcontrollers and sensors to commercial and industrial robots. Drivers may be written for hardware that is not natively supported by *Player*, and implemented as plugin drivers [43]. In the centre block of Figure 6.15, on the right, are the physical device drivers listed that are implemented onboard the platform. The number "6665" is the address of platform IPR1 on the local network.

The "position2d:0" interface accepts commands in the form of trajectories, and sends commands in the form of wheel velocities. This interface also makes odometry data available for use by navigation and localisation algorithms. This is for control of the drive wheels. The "sonar:0" interface makes ultrasonic rangefinder data available for use by navigation and localisation algorithms. The "actarray" interface is a generic interface for an array of actuator that may be customised for a variety of different situations. The "actarray:0" interface represents the actuator array made up of the two static stability units, while "actarray:1" interface represents the payload interface system.

Abstract drivers can be defined as software drivers that do not interface directly with hardware. These drivers make use of data from sensory drivers or other abstract drivers to command other actuating drivers or produce data fro use by other abstract drivers. These abstract drivers are usually based on mathematical algorithms, examples include navigational algorithms, localisation algorithms, and image processing algorithms. Abstract drivers implemented here include drivers for map localisation, global navigation, and local obstacle avoidance. Figure 6.16 shows the sequence of execution of the abstract *Player* drivers.

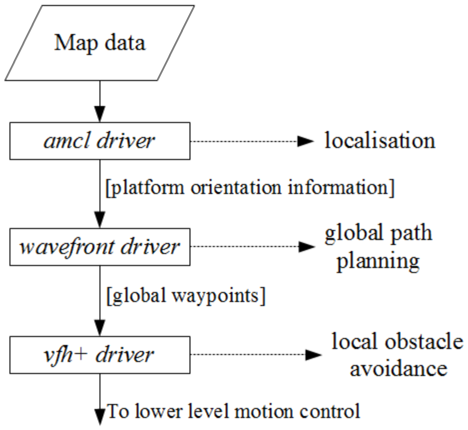


Figure 6.16: Sequence of execution of the abstract *Player* drivers implemented on the platform.

The sequence of execution begins with the provision of map data from the remote management system. The driver involved with localisation within the map provided is known as the *amcl* driver, which is an acronym for “adaptive Monte Carlo localisation”. The driver is based on an adaptive Monte Carlo algorithm specifically used for localisation within a mapped environment [15]. This algorithm makes use of a particle filter and is based on probability distribution theory. The adaptivity of this algorithm comes from the adjustment of the number of particles in the filter according to the certainty of the location of the platform.

The next step in the execution sequence is global path planning. this is achieved by the $wavefront$ driver. This driver makes use of map data and a wavefront propagation algorithm to produce global waypoints [29]. These waypoints connects the longest straight distances that do not intersect with known, stationary obstacles.

The vfh driver is implemented for local obstacle avoidance purposes. This driver makes use of the Vector Field Histogram Plus algorithm and ultrasonic rangefinder data to calculate the optimal platform trajectory around unexpected obstacles that occur in the path as planned by the $wavefront$ driver. [47]. The abstract drivers form the complete navigation system as implemented onboard the platform.

A file, known as the configuration file, encapsulates the capabilities of each instance of the *Player* program, or server. This file informs *Player* which devices are being used, and thus which drivers to use and which interfaces the drivers require. Data are transferred in forms that are acceptable and understandable to the specific device drivers, i.e. device interfaces, that are present. The configuration files writen for instatiating the platform and its devices can be found on the Compact Disk accompanying this document.

Communication between the *Player* running on the SBC and the remote management system occurs according to the Transmission Control Protocol and Internet Protocol (TCP/IP) suite. The physical connection is via a wireless serial connection, as described in Chapter 5. Use of the TCP/IP suite allows for multiple clients to subscribe to the *Player* server onboard the platform. This also allows for efficient transfer of information through the use of packetised data transfer.

Communication between *Player* and the programs running onboard the microcontrollers occurs according to the Universal Simultaneous Asynchronous Receiver/Transmitter (USART) communication protocol. Here, the physical connection is is via a serial connection based on the RS-232 communication standard between the onboard SBC and the microcontrollers.

6.4 Safety Considerations

As discussed in previous chapters, safety functions from the FMECA's of the other systems onboard the platform can be handled by software. In the case of the failure of the actuation system, or measurement system , of the payload interface system, *Player* can diagnose that there is a problem and prevent the static stability system to retract. This way the platform may continue operating in a statically stable mode until the problem can be properly diagnosed and solved by a maintenance team at the next maintenance event.

In the case of the failure of the static stability system, *Player* can also diagnose the situation and depending on the position of the static stability units at the time of failure, *Player* can provide the manufacturing system management software with information to assist in decision making. In the case where the static stability units are in the fully extended position at the time of failure, the platform may continue in a statically stable mode until the problem can be properly diagnosed and solved by a maintenance team at the next maintenance event.

The other situation is where the static stability units are in the up or intermediate position at the time of failure. In this case the only treatment is for the platform to remain in the dynamically stable mode and the manufacturing system management software should prevent scheduling loading of the platform until the platform can be shut down and the problem investigated. In this case maintenance should enjoy high priority due to the fact that the platform cannot continue operating normally.

Safety functions may be designed to take into account failure of electronic components. One such example is the case where connection to multiple ultrasonic rangefinders is lost. In this case it will be virtually impossible for the platform to avoid obstacles that occur on its planned path. In this case, as a safety precaution the static stability units may be extended and the cruising velocity of the platform reduced to the point where collision with an obstacle will not cause damage to the platform or the obstacle. The platform can be commanded to back track along a path that is certainly clear of obstacles to a position where the situation can be safely investigated.

In the event of total systems failure there are two situations to consider. The first is the case where the static stability units are in the extended position at the time of failure. In this case there is no danger of loss of product or catastrophic failure due to the fact that the platform is statically stable. In this case it will simply become inoperable. If this is the case the plant management system should surmise that the platform is non-responsive and dispatch a maintenance team to investigate the situation.

The other situation to consider in the event of total system failure is where the static stability units are in the retracted position at the time of failure. In this case there is a strong possibility that there may be a loss of product if there is a payload loaded on the platform at the time of failure. If there is no payload loaded there is no danger of loss of product, however in both cases there is a possibility of catastrophic failure. The worst case scenario in this situation is loss of power to the drive motors, in which case the platform will lose the ability to balance itself and fall over onto the front or back.

In the event of drive motor command overrun the motor drivers are designed in such a way as to cut off power to the motor when the command reaches the hard coded limit of the driver, which is a built-in safety feature of these devices. It is not recommendable to adjust this limit to a lower value in the case of the self-balancing platform. This is due to the fact that the drivers need the capacity to drive the motors above the average cruising velocity to be able to maintain stability when it is commanded to slow down when it reaches its destination.

6.5 Chapter Summary

This chapter familiarises the reader with the complete software system that has been developed for implementation onboard the platform. The discussion starts with the processing of raw data with inherent uncertainties to produce much more reliable information. The control architecture in terms of each physical mechanism is discussed as well as the integration of the separate control system in the form of the operational control sequence. The higher level software tasked with the control of the platform motion as well as the hardware abstraction layer, in the form of the *Player* device management software, are discussed. Final safety considerations are evaluated and discussed.

Chapter 7

System Evaluation and Discussion

In this chapter the system as a whole in terms of the Mechatronic engineering approach is critically evaluated considering all information that has been collected during the research project. The discussion includes technical findings and safety considerations. The relevant applied technologies are also evaluated in terms of maturity and applicability.

7.1 Technologies and Concepts of the Future

The technologies required for research in mobile robotics for application in automated manufacturing environment are at the point where the routing and scheduling of platforms are the most significant variables in the implementation of a team of autonomous materials handling platforms. This is at the management level of manufacturing systems.

There are proprietary software packages as well as open source solutions available for simulating mobile robotics in the development phases. The computer and electronics industries have advanced to the point where hardware requirements are relatively easily procurable. Mechatronic engineering has played a pivotal role in the advancement of the field of robotics research by formulating a design approach that suits this field of research.

Mass Customisation Manufacturing is a concept of the future. Automated manufacturing systems have the best chance of achieving the goals of this concept. Although the technologies fundamental to automated system are well matured the combination of these technologies with current management strategies is not at the point where these systems are able to cope with the requirements of Mass Customisation Manufacturing.

The principle of the inverted pendulum is a novel application in mobile robotics research. The small margin for error and variation in industrial applications means that the inherently unstable system produced by the development of a two-wheeled self-balancing mobile robotic platform is not suitable for this kind of environment. This statement is supported by the results of the active stability control system tests, and discussed further in the following sections.

7.2 Technical Findings

This section discusses findings of the research in terms of design considerations, systems level considerations, and functional safety considerations. Design considerations include the three subsystems that make up the overall system, i.e. mechanical, electronic, and software systems, while system level considerations involve evaluation of the system integration.

7.2.1 Design Considerations

The mechanical system has been designed using fundamental engineering principles, in terms of material strength, dynamics and kinematics, and power transmission. The primary drive system consists of powerful DC motors, with the power being transmitted to the drive wheels via mild steel shafts. As mentioned in the previous chapter, the couplings between these two components possess some tolerance which is not conducive to achieving the required accuracy in the control of the drive wheels for a self-balancing platform. This is manifested in the oscillatory response visible in the results of the active stability control system tests as discussed in Section 6.2.4.

Due to the dynamics of the platform when it is in the dynamically stable mode the traction between the payload interface rollers and a loaded payload is too small. This will cause payloads to slide when the platform is in the dynamically stable mode, it also causes difficulty in manipulating the payload into the optimal position before the platform transitions from the static stability mode after the payload is loaded. A simple solution to this is to coat the roller with a high traction substance, such as rubber, to improve the traction characteristics.

Considering the drive electronics, implemented to control the drive motors, these components including the motor drivers and the motors themselves, are well suited to the application. The payload interface, as well as the load cells supporting the payload interface, are also well specified for the application. If the platform were to be used in the statically stable mode permanently, the platform will be able to transport payload weighing up to 100 kg.

The SBC implemented onboard the platform is specified for industrial applications, which makes it perfectly suited for this application. The microcontrollers that have been developed in-house are not designed for robust applications instead they are better suited for prototyping purposes. The range of AVR microprocessors, produced by Atmel, are powerful and are well suited for mobile robotics applications, there are also many commercially available microcontrollers based on the Atmel range of AVR microprocessors.

In terms of the operating system and software running onboard, the Linux operating system is well suited, due to the fact that it can be customised to suite applications for lean operation. The *Player* device management software installed on the SBC is specifically designed for implementation in mobile robotic systems, it is also more customisable than commercially available equivalents.

The integration of the integral components required to achieve the objective of developing a self-balancing platform required a knowledge of programming microcontrollers at the level of producing lean code in order to conserve processing time. This is a specialised skill that can only be developed by experience and trial-and-error. This has lead to a system that may not be as optimal as it should.

The instance of the Kalman filter that is currently implemented for estimating the tilt angle and angular velocity states of the platform possesses a slow response to the variation of the data from the corresponding sensors. This negatively affects the performance of the control system for the active stability of the platform. The response can be improved by implementing a more comprehensive Extended Kalman filter algorithm that is specifically designed for the system under investigation and tuned for lean operation in terms of processor time.

7.2.2 System Level Considerations

The integration of the three subsystems that make up a mechatronic system is fundamental to the mechatronic engineering design philosophy. This aspect is evaluated here.

System Evaluation

Evaluating whether the overall system fulfills the design parameters identified at the outset of the research is a test of the success of the development process. As enumerated in Section 3.2 the following statements can be made:

1. The LQR based active stability control system and the selected motion control algorithm has been selected and designed such that the platform operates as time efficiently as possible.

2. The platform is able to actively maintain dynamic stability by controlling the drive motors according to the platform states as measured by inertial sensors and rotary encoders and estimated where necessary.

3. The *Player* device management software incorporates navigational algorithms based on odometry and map data as well as obstacle avoidance techniques based on rangefinder sensor data.

4. A WiFi based communication system has been selected in order for the platform to be able to communicate wirelessly with remote management software as well as other platform within the manufacturing plant.

5. The payload interface system has been designed to be able to accommodate payloads as set out in Section 3.1.1.

6. The control structure relays commands from the remote management software via the *Player* device management software installed on the onboard SBC to command the actuator in order to react appropriately to bottlenecks in the manufacturing system.

7. The control sequence of the static stability system enables the platform to interface with the I/O ports of manufacturing cells while in the statically stable mode in order to simplify the loading and unloading of payloads.

8. The remote management software can inform the platform via the wireless communication link of possible payload characteristics which can be stored on the SBC hard disk drive.

9. The complete platform architecture has been designed and fabricated to support a maximum payload weight of 20 kg.

10. The length and width of the payload interface system has been designed to enable it to accommodate payloads of 424 mm in any direction.

11. The static stability system has been incorporated into the design of the platform to enable it to become statically stable at predetermined phases of operation.

12. The most energy intensive aspect of the platform design is the active stability control system. This has been designed in such a way to minimise the control input and in turn minimise the energy exertion.

13. Various safety functions incorporated into the subsystems of the platform are aimed at minimising the danger to persons working in close vicinity to the platform.

The most important consideration was highlighted by the results of the active stability control system tests from Chapter 6; this is required for developing an autonomous two-wheeled self-balancing platform. Although the underlying technologies, such as sensor data acquisition and drive electronics, are well understood the performance of a mechatronic engineering system does not only depend on the effective integration of these technologies, but also the selection of the most appropriate technologies and the optimisation of the complete system.

Applicable Environments

Other research currently being conducted within the research group is the development of hybrid manufacturing cells combining RMS concepts with dedicated machine tool technologies. The ultimate goal of these manufacturing systems is to achieve Mass Customisation Manufacturing. These manufacturing systems are dependent on flexible materials handling systems for transferring materials and parts between manufacturing cells, in order to achieve this goal. The assistance of these flexible materials handling systems is the primary intended application for the platform.

The platform has been designed specifically to integrate into a framework that has been in development within the research group, based on flexible materials handling system principles for these RMS based manufacturing systems. As mentioned in previous chapters, one of the reasons for the implementation of the *Player* device management software is in order to enable the platform to communicate with other similar platforms and with the manufacturing system management software on a common protocol.

Figure 7.1 shows screenshots of a graphical simulation ran to show the operation of the platform in a hypothetical manufacturing system. The sequence starts at the shot with the position of the platform indicated by the number 1 and runs consecutively to number 5. Videos of the simulation can be found on the accompanying Compact Disk.

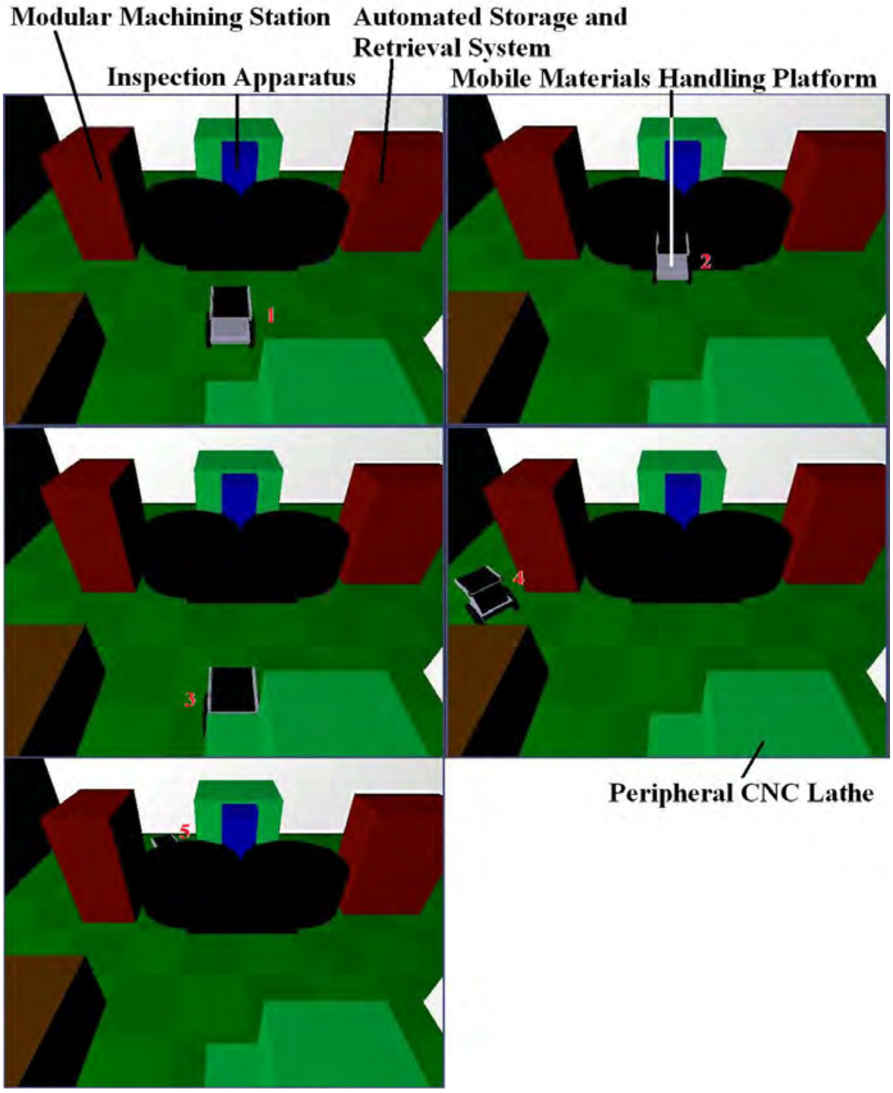

Modular Machining Station
Automated Storage and Retrieval System
Inspection Apparatus
Mobile Materials Handling Platform
1
2
3
4
5
Peripheral CNC Lathe

Figure 7.1: Sequence of screenshots of *Gazebo* graphical simulation.

The manufacturing system consists of a manufacturing cell made up of an automated storage and retrieval system, a modular machining station, and an automated inspection apparatus. peripheral to the manufacturing cell is a CNC lathe for complementary functionality. The simulation uses simplified dynamics and kinematics models, with user defined system characteristics, to model the dynamics of the self-balancing platform. The simulation software is known as *Gazebo* and is a three-dimensional simulation utility developed as part of the *Player* project.

The operation that is proposed in the *Gazebo* simulation starts at position number 1, with a hypothetical payload already loaded. The platform unloads the payload at position number 2, returns to position 3 at the CNC lathe, to load a new payload and transports it to position 5, via intermediate position 4, in order for the second payload to circumvent the modular machining station and proceed to inspection. This represents a typical operation that the platform would perform in the primary intended application of bottleneck resolution.

Figure 7.2 shows the platform carrying a payload in the dynamically stable mode. In this figure the platform is in the initial stage of approaching the I/O port of a Manufacturing cell seen in the background. As the platform approaches the I/O port the static stability system will extend and the platform will transform to the statically stable mode in anticipation of interfacing with the port.

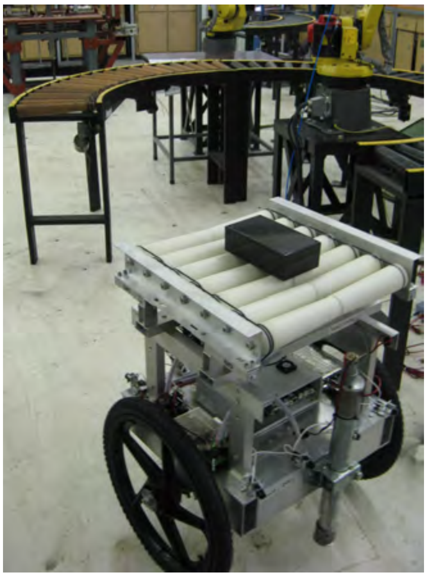


Figure 7.2: Platform carrying a payload in the dynamically stable mode.

Figure 7.3 shows the platform in the statically stable mode and it is ready for interfacing with the conveyor section I/O port. In this figure the platform has entered the critical area around the I/O port where it needs to be statically stable in order to converge towards the port optimally.

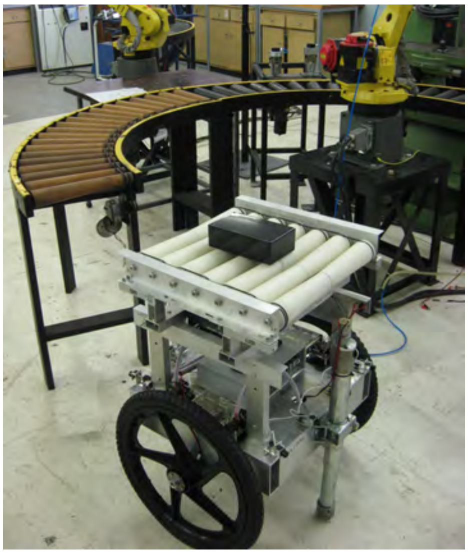


Figure 7.3: Platform carrying a payload in the statically stable mode.

Figure 7.4 shows the platform interfacing with the I/O port and in the process of unloading the payload onto the conveyor section. The platform has been designed to interface effectively with the conveyor sections present in the laboratory for integration purposes.

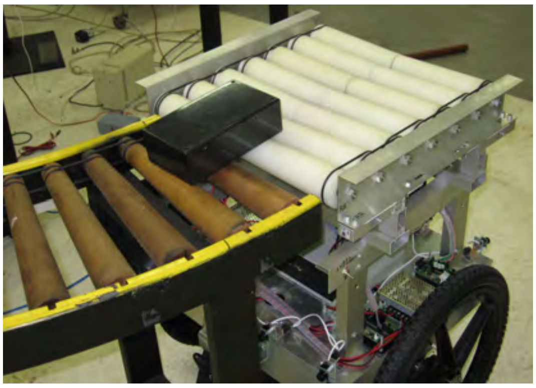


Figure 7.4: Platform unloading a payload at a manufacturing cell I/O port.

Some alternative applications may also be considered other than for the resolution of bottlenecks in RMS. The rollers built into the payload interface system eases transfer of payloads to and from conveyor sections within the operating environment. For this reason the platform may also be employed as a means of distributing payloads from an ASRS to manufacturing cells and also transporting payloads from assembly stations back to the ASRS. A graphical representation of this type of application is shown is Figure 7.5.

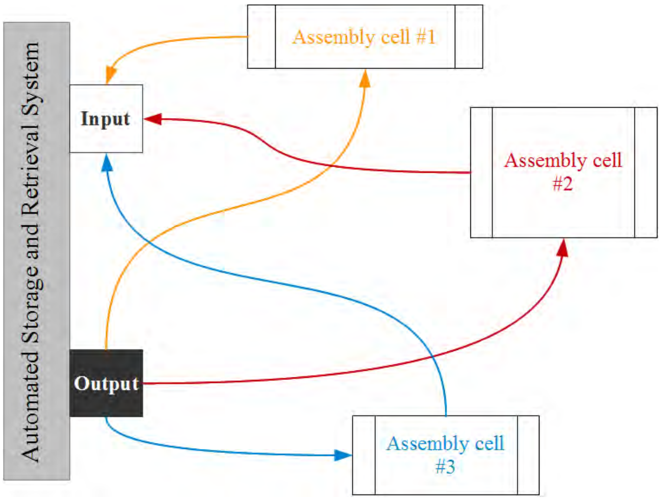


Figure 7.5: Part distribution and collection application for statically stable mode platform.

For operations such as this, that require frequent loading and unloading of payloads, it is more efficient for the platform to operate in the statically stable mode permanently. Examples of this type of manufacturing system include any manufacturing system with products assembled from pre-produced parts, such as subassemblies for the automotive industry, and large household appliance manufacturing operations. This platform could already be employed as bottleneck resolution measure in this situation, in which case this could be a complementary function that the platform is able to perform, if necessary.

Another alternative operation that the platform could perform in the permanently statically stable mode is the transportation of parts and assemblies, across short distances, that are excessively heavy. Transferring parts short distances would traditionally be achieved by serial robotic arms or conventional conveyor systems. The advantages that this platform possesses over these options are that the platform is a cheaper solution than high payload serial robotic arms, and mobile platforms are more flexible than conventional conveyor system. Figure 7.6 represents a graphical explanation of this application. An example of this type of application is the transfer of small internal combustion engine blocks from fabrication and machining to assemblies of variants based on the same engine block.

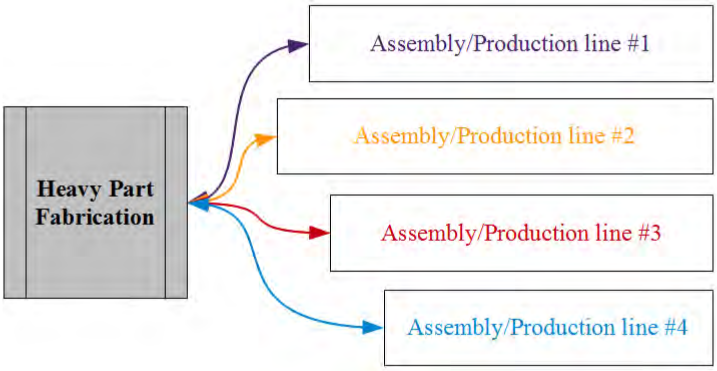

Heavy Part Fabrication
Assembly/Production line #1
Assembly/Production line #2
Assembly/Production line #3
Assembly/Production line #4

Figure 7.6: Heavy part transportation application for statically stable mode platform.

7.2.3 Operational Safety

In order to exploit the advantages of implementing a two-wheeled self-balancing platform, the platform is required to operate in the dynamically stable mode for as much time in operation as possible. More operation time in the dynamically stable mode coincides with more opportunity for catastrophic failure of the platform while in operation.

Safety functions may be built into the various subsystems which can account for certain shortcomings of the other subsystems as discussed in the previous three chapters. This is possible due to the high level of system integration in a mechatronic engineering project such as this. The main concern however, is that a self-balancing platform is inherently unstable and even the most consistent control strategy will not be able to account for unforeseen disturbances of the drive system such as oil spillage that will cause wheel slippage and low level, unsighted obstacles in the path of the platform.

The failure modes of any of the subsystems affect the complete system, a compilation of the FMECA of the individual subsystems would thus constitute an FMECA of the complete system. For this reason the reader is referred back to the various sections on safety considerations in the previous three chapters. The failure modes that have catastrophic effects, at the overall system level, are of concern here. From an operational safety point-of-view a two-wheeled self-balancing materials handling platform poses more disadvantages than advantages. This is shown in the FMECA performed on the various subsystems.

7.3 Chapter Summary

This chapter is aimed at evaluating the product of the research in terms of a complete system in order to validate the feasibility of the system for the intended application. Initially, the system is evaluated technically at the technology level and at the systems level, as well as in terms of operational safety. The system is evaluated according to the design parameters identified at the initial phases of the research.

Chapter 8

Conclusion and Future Research

In this chapter the dissertation is concluded and possible future work is discussed. It is aimed to conclude the dissertation in the most concise manner possible.

8.1 Research Conclusion

A two-wheeled self-balancing robotic materials handling platform, with autonomous operational capabilities, has been researched and designed. The platform was designed for deployment with a flexible materials handling system for application in a Reconfigurable Manufacturing System. The mechatronic engineering approach of system integration was the central theme in the entire process, from the conceptual design phase to the fabrication and assembly phase.

Mobile robotics for application in flexible materials handling system was determined to be a mature technology. The goal of achieving Mass Customisation Manufacturing through the implementation of Reconfigurable Manufacturing Systems is still a concept of the future, however technologies that support manufacturing seem to be developing towards a compromise between full reconfigurability and efficiency through cellular plant layouts in manufacturing plants.

To date, the principle of the inverted pendulum has proved itself as a useful tool in control engineering as a benchmark test for various control theories. It is a novel application in mobile robotics and the implementation of it produces interesting results. Testing and performance analysis has shown that it does not produce an optimal solution for implementation for mobile robotics in materials handling. Safety analysis supports this statement. This does not detract from the effectivity of the design philosophy followed by developing a mechatronic system.

8.2 Recommendations for Future Work

Some recommendations can be made for future work on this project. These recommendations are the result of the critical analysis of the product of the research to date. Improvements that should be made to the mechanical platform architecture are redesign of the motor output to drive shaft couplings and coating the payload interface system rollers with a traction enhancing substance, such as a rubber based compound. These are necessary improvements if the platform is to continue operating as a semi-dynamically stable platform.

As far as safety is concerned certain redundancies could be incorporated into the system to prevent catastrophic failure or at the least to prevent irreparable damage to platform or payload in the case of such a failure. More research would be required in the area of functional safety and occupational safety, specifically in manufacturing environments. This is absolutely crucial if the platform is to continue operating as a semi-dynamically stable platform.

In terms of the programming and software aspect of the mechatronic system there is substantial room for improvement of the Kalman filter technique that has been implemented. It would be recommendable to design an optimal Kalman filter algorithm for state estimation specifically for the measurement of tilt angle and angular velocity, in the bandwidth that a self-balancing mobile robotic platform of this scale operates in.

The control system based on the Linear Quadratic Regulator design technique performs adequately for the linearised system. Linearising the system is not the most effective solution to this control problem since this is a nonlinear system. One possible avenue of research is the implementation of a control system based on nonlinear control theory. The results from this could be compared to the performance of the LQR based control system.

Further research in the area of control engineering could involve the application of fuzzy control laws for the active stability control of the platform. This could also prove a useful testbed for Artificial Intelligence (AI). The implementation of AI could include, not only the active stability control, but also motion control and path planning. AI is also used in swarming applications, which the platform could form part of under AI control.

Appendix A

Technical Fabrication Drawings

This chapter consists of the technical drawings produced for fabrication and assembly of the mechanical platform architecture. Only drawings that resulted in the production of parts in the School of Mechanical Workshop are of interest here. Following, is a list of the drawings that make up this chapter:

1. Main Assembly

    A Payload Interface Assembly

    - A.1 Load bed channel
    - A.2 Roller Assembly
        - A.2.1 Load bed shaft
        - A.2.2 Roller
    - A.3 Load bed motor cross brace
    - A.4 Load bed motor mounting

    B Support Assembly

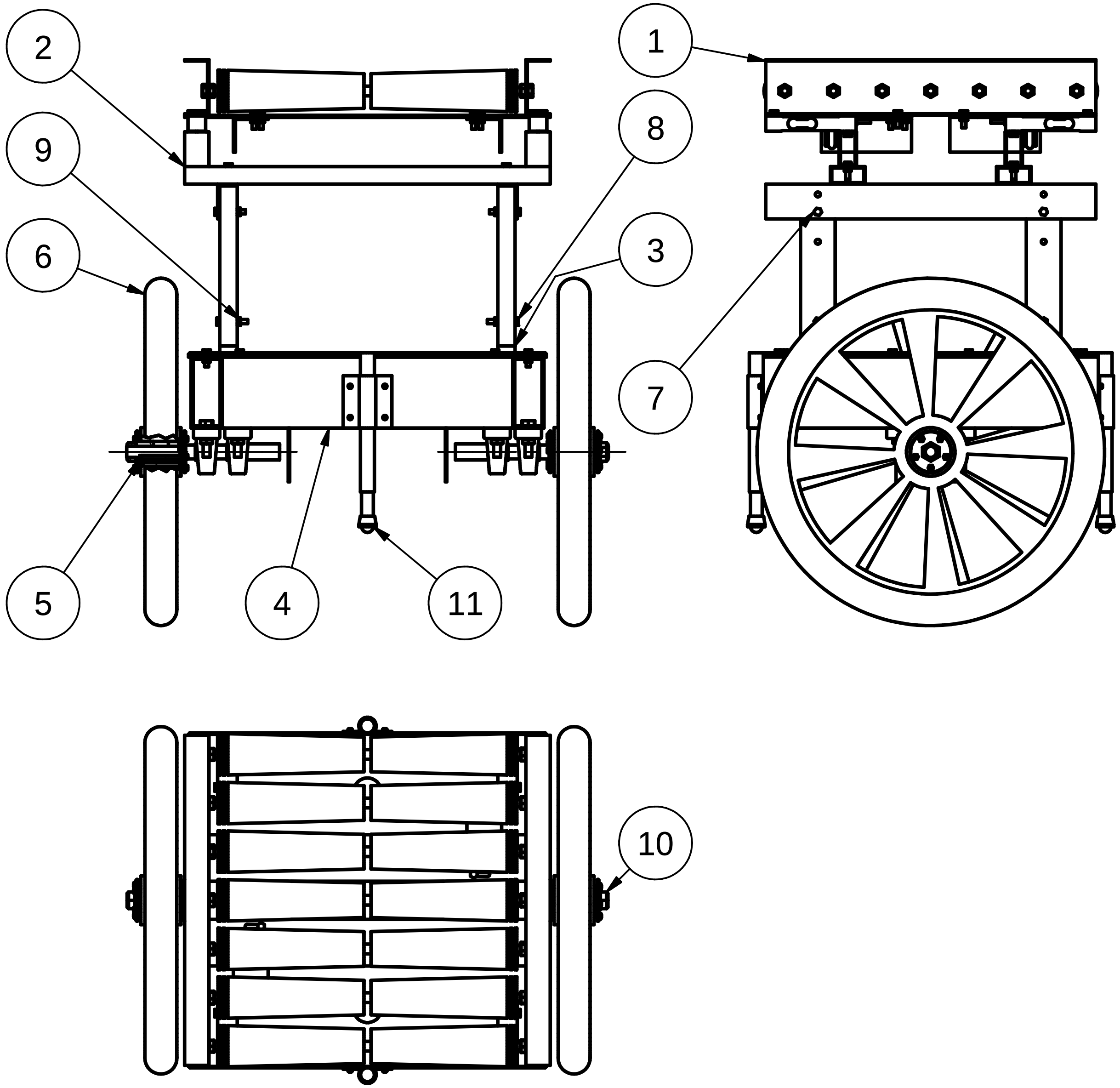
    - B.1 Load bed support pillar
    - B.2 Load bed support level
    - B.3 Load bed cross brace
    - B.4 Load bed mounting block

    C Support mounting block

    D Base Assembly

    - D.1 Base plate
    - D.2 Main base tube
    - D.3 Cross base tube

D.4 Drive support tube
D.5 Drive motor mount tube
D.6 Drive shaft
D.7 Drive shaft collar
D.8 Drive motor mount plate
E Wheel Assembly
E.1 Wheel bush



Parts List			
ITEM	QTY	PART TITLE	DESCRIPTION
1	1	PAYLOAD INTERFACE ASSEMBLY	
2	1	SUPPORT ASSEMBLY	
3	4	SUPPORT MOUNTING BLOCK	50x25x60 AL
4	1	BASE ASSEMBLY	
5	2	PARALLEL KEY	5x5x60
6	2	WHEEL ASSEMBLY	
7	20	ISO METRIC BOLT	M6x1
8	4	ISO METRIC BOLT LONG	M6x1
9	4	ISO METRIC NUT	M6x1
10	2	ISO METRIC NUT	M16x2
11	2	STATIC STABILITY UNIT	

University of
Kwa-Zulu Natal

School of
Mechanical
Engineering

Orthographic Projection

OR

SCALE N/A

UNITS : mm

	Date	Checked
Project Supervisor		
Workshop Technician		
Technical Manager		

TITLE

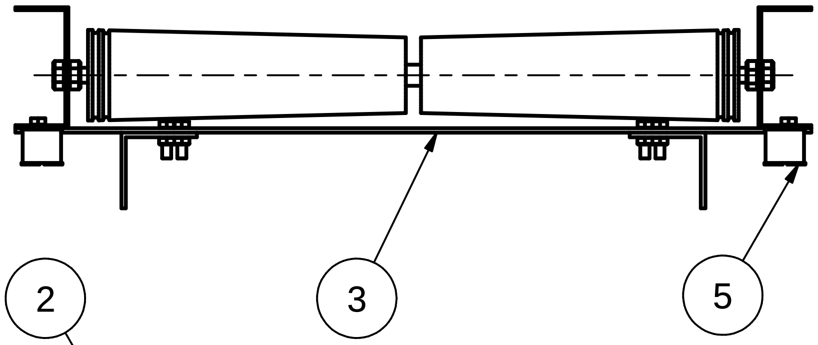
MAIN ASSEMBLY

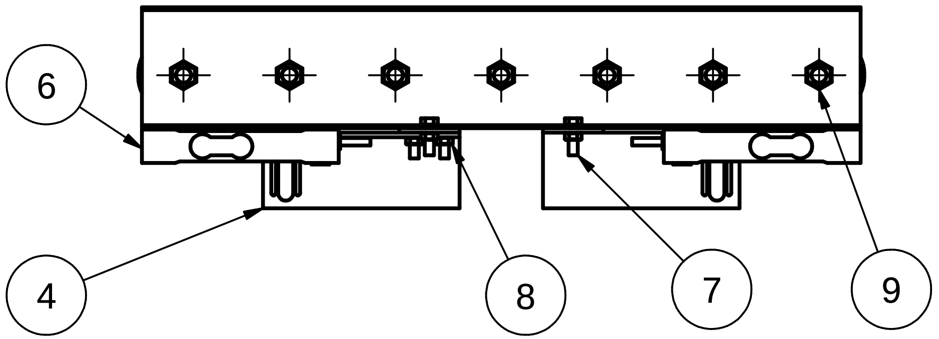
PROJECT

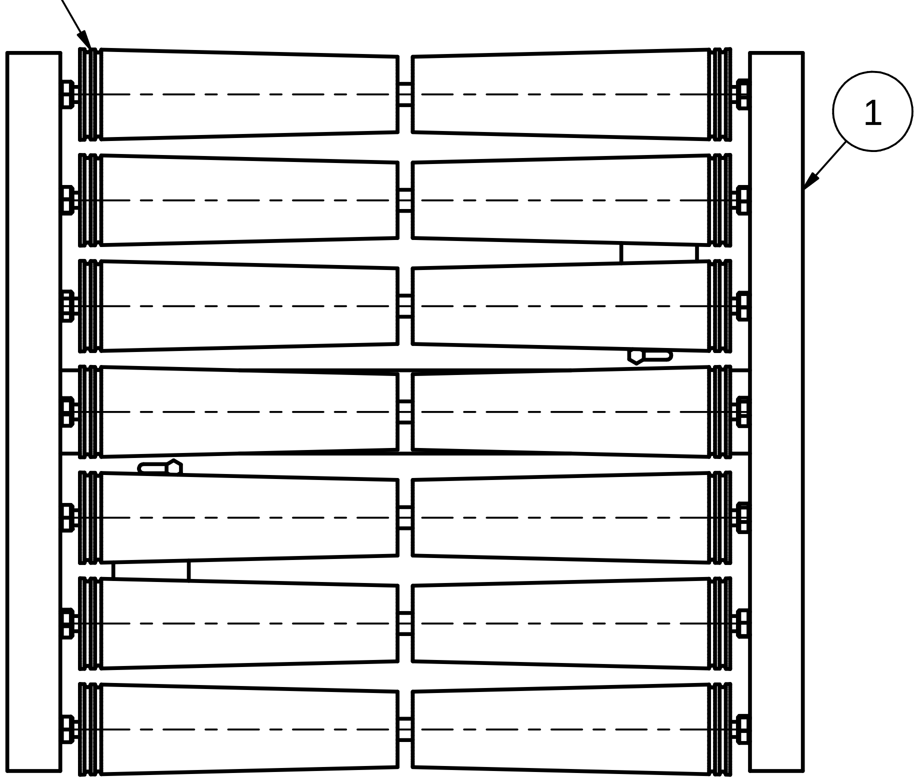
MATERIALS HANDLING ROBOT

STUDENT NAME LOUW BUTLER

TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za







Parts List

ITEM	QTY	PART TITLE	DESCRIPTION
1	2	LOAD BED CHANNEL	AL EXTRUSION
2	7	ROLLER ASSEMBLY	
3	2	LOAD BED MOTOR CROSS BRACE	MILD STEEL PLATE
4	2	LOAD BED MOTOR MOUNTING	MILD STEEL ANGLE
5	16	LOAD CELL WASHER	ID 7 x 1
6	4	PW6D LOAD CELL	HBM
7	8	M6 ISO BOLT	
8	8	M6 ISO NUT	
9	28	M10 ISO NUT	

University of
Kwa-Zulu Natal

School of
Mechanical
Engineering

Orthographic Projection

OR

SCALE N/A

UNITS : mm

	Date	Checked
Project Supervisor		
Workshop Technician		
Technical Manager		

TITLE

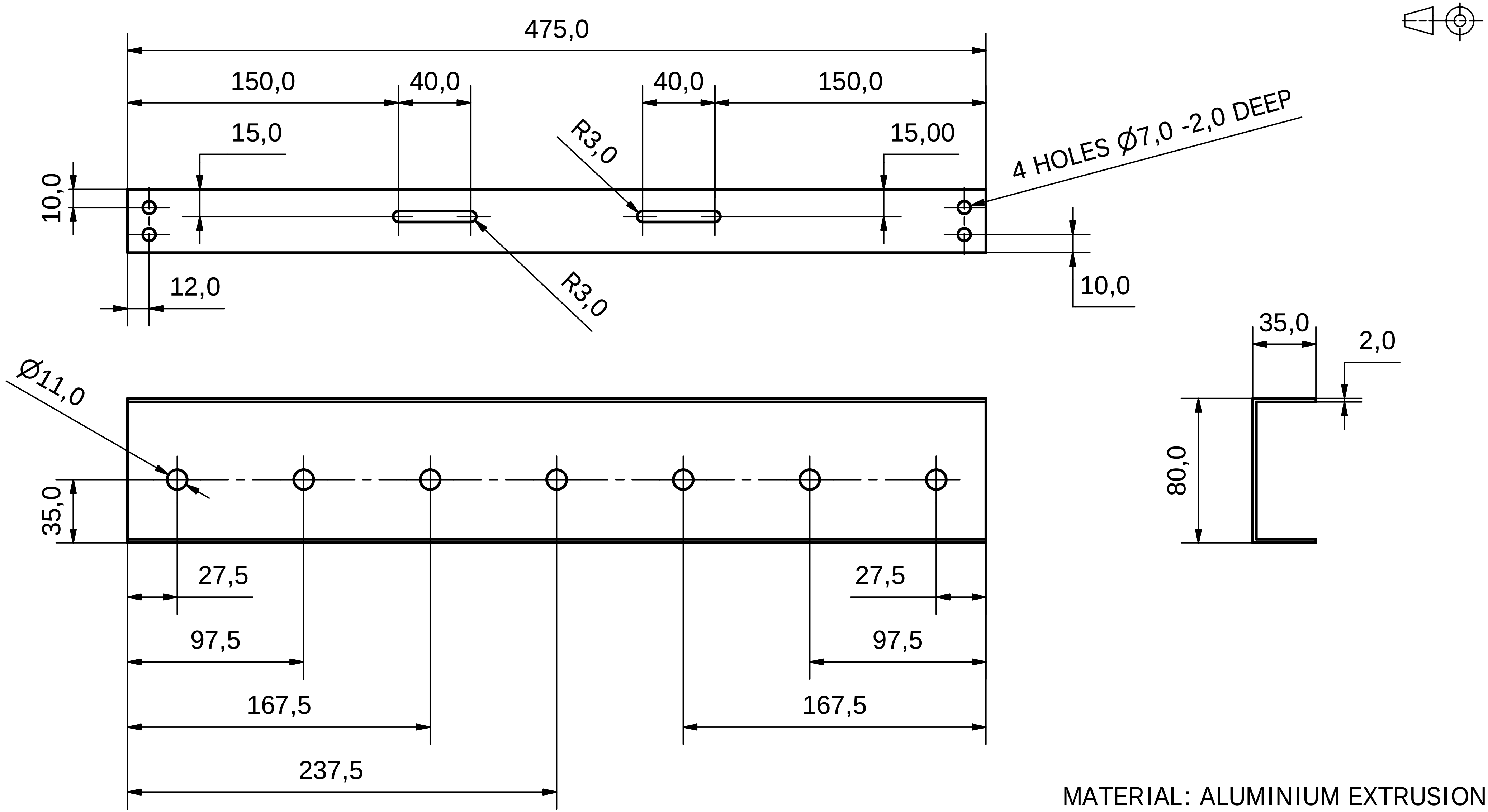
PAYLOAD INTERFACE ASSEMBLY

PROJECT

MAIN ASSEMBLY

STUDENT NAME LOUW BUTLER

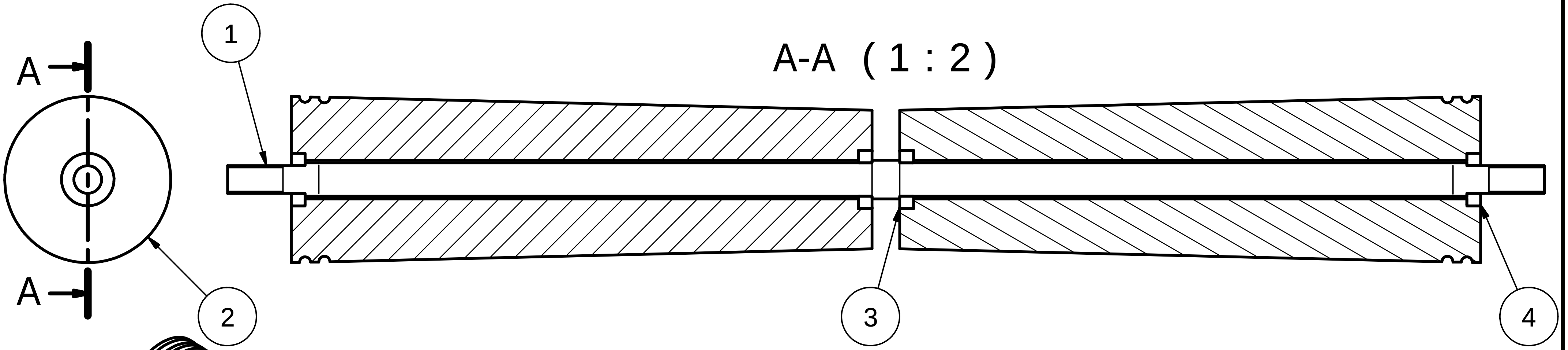
TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za

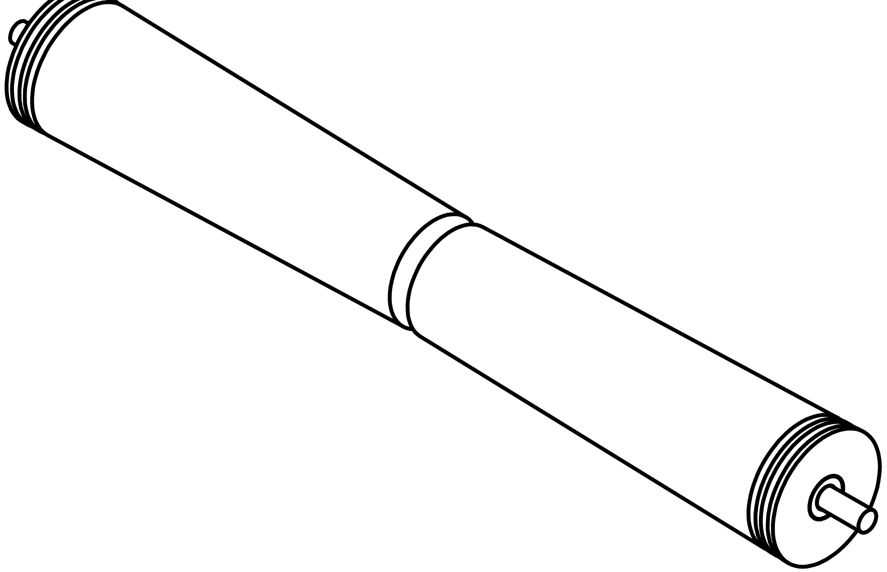


MATERIAL: ALUMINIUM EXTRUSION
2 INSTANCES REQUIRED

University of Kwa-Zulu Natal School of Mechanical Engineering	Orthographic Projection	SCALE 1:3		TITLE LOAD BED CHANNEL
	OR	UNITS : mm		
		Date	Checked	PROJECT PAYLOAD INTERFACE ASSEMBLY
	Project Supervisor			
	Workshop Technician			STUDENT NAME LOUW BUTLER
	Technical Manager			TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za

A-A (1 : 2)





Parts List			
ITEM	QTY	PART TITLE	DESCRIPTION
1	1	LOAD BED SHAFT	MILD STEEL
2	2	ROLLER	NYLON
3	2	INNER BALL BEARING	12x21x5
4	2	OUTER BALL BEARING	10x19x5

University of
Kwa-Zulu Natal

School of
Mechanical
Engineering

Orthographic Projection

OR

SCALE N/A

UNITS : mm

	Date	Checked
Project Supervisor		
Workshop Technician		
Technical Manager		

TITLE

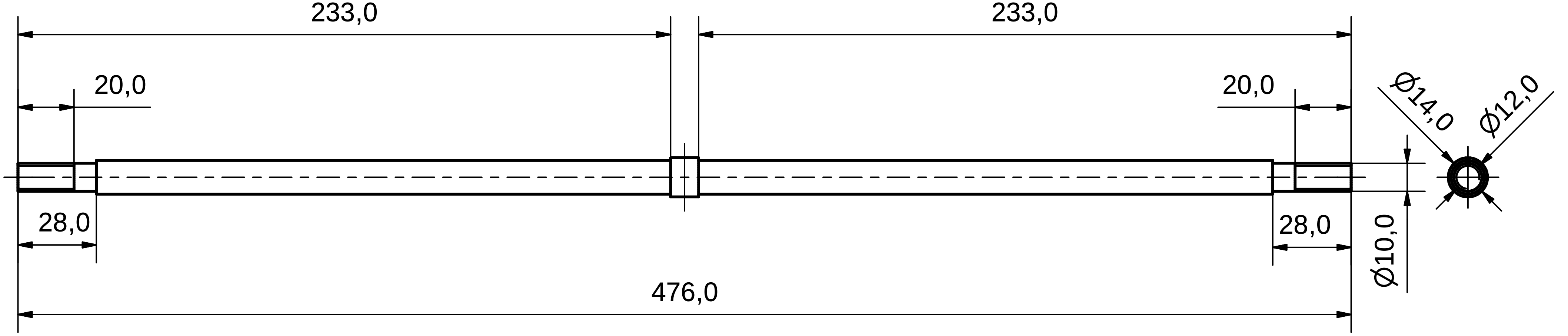
ROLLER ASSEMBLY

PROJECT

PAYLOAD INTERFACE ASSEMBLY

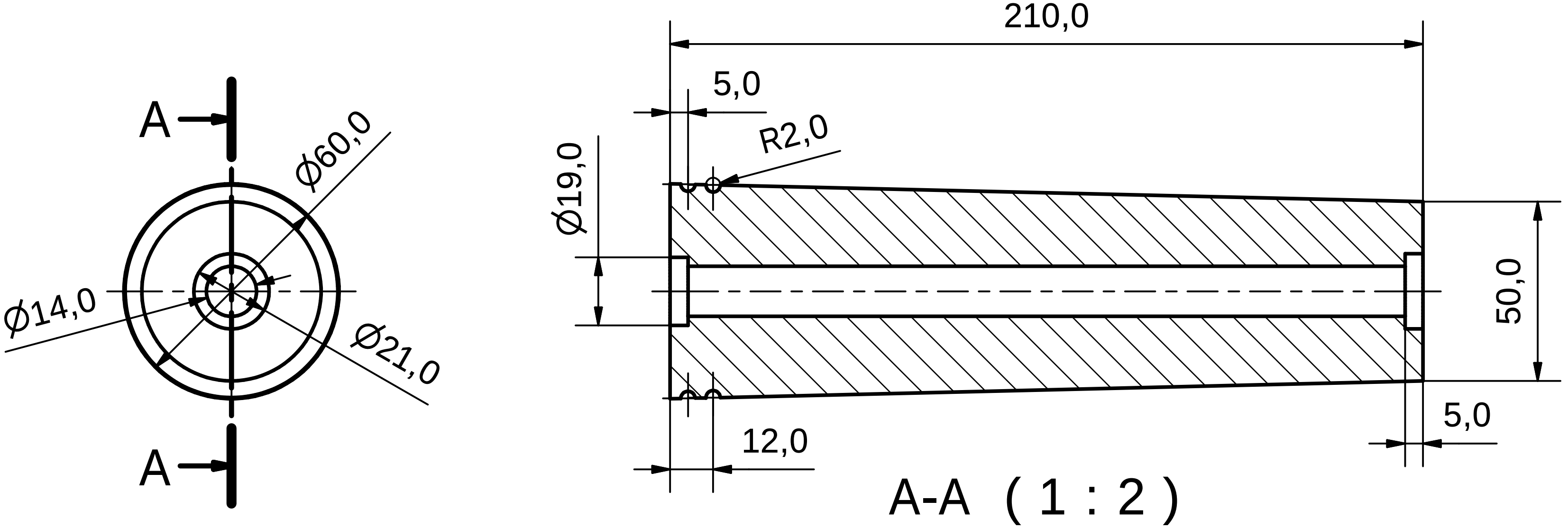
STUDENT NAME LOUW BUTLER

TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za



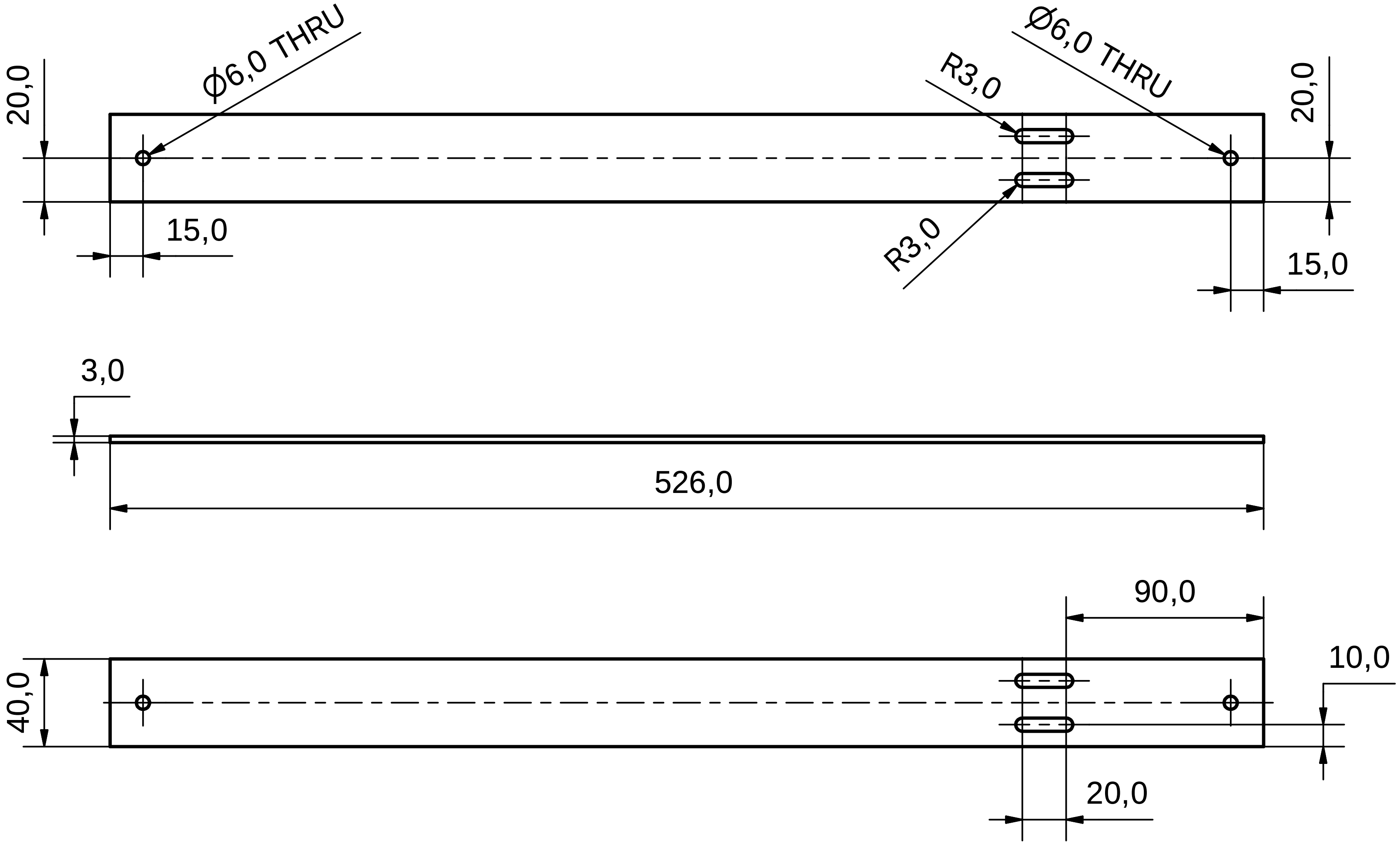
MATERIAL: MILD STEEL
7 INSTANCES REQUIRED

University of Kwa-Zulu Natal School of Mechanical Engineering	Orthographic Projection	SCALE 1:2		TITLE LOAD BED SHAFT
	OR	UNITS : mm		
		Date	Checked	PROJECT ROLLER ASSEMBLY
	Project Supervisor			
	Workshop Technician			STUDENT NAME LOUW BUTLER
	Technical Manager			TEL NO. EXT 1227 EMAIL 207506509@ukz.ac.za



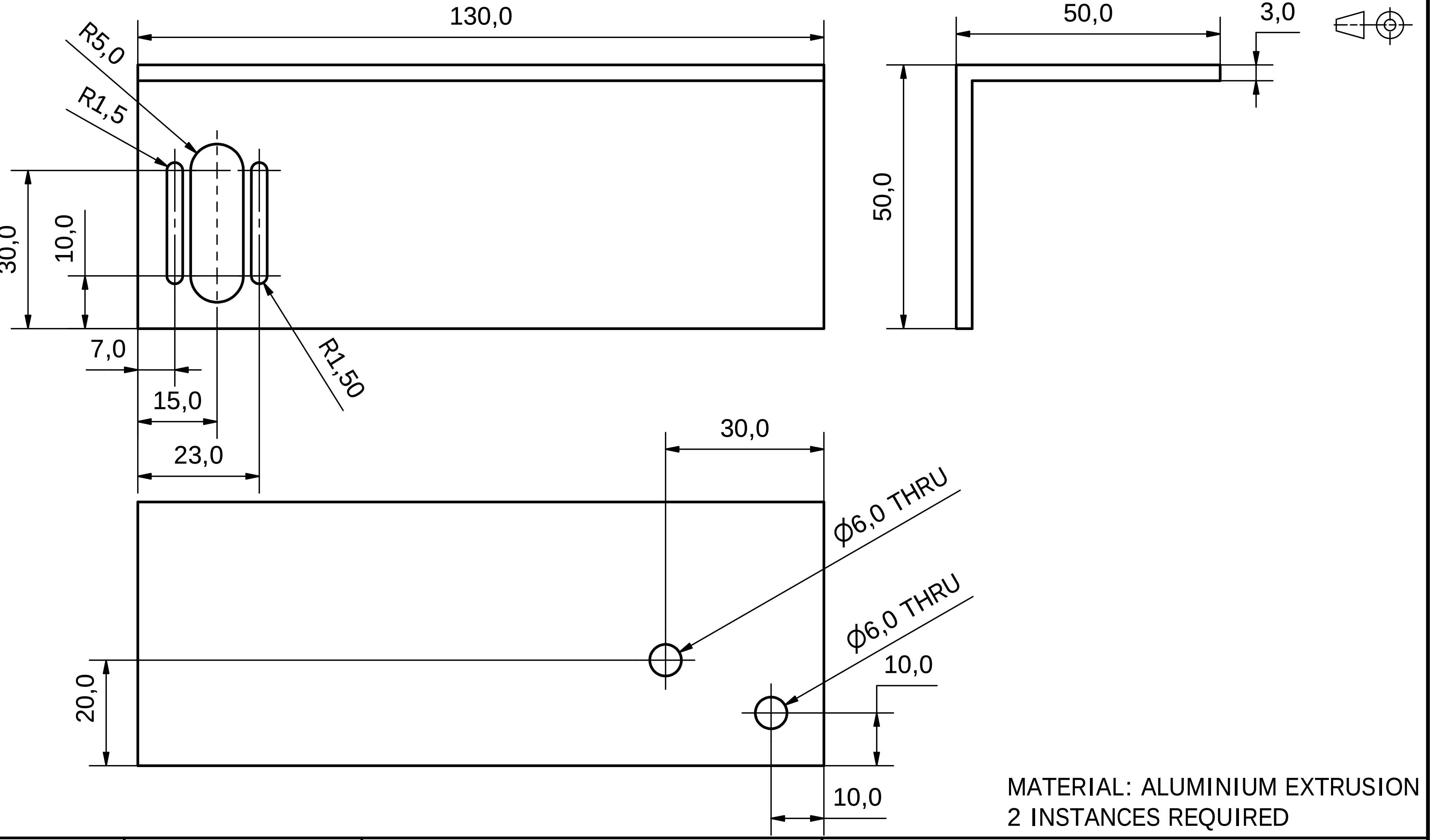
MATERIAL: NYLON
14 INSTANCES REQUIRED

University of Kwa-Zulu Natal School of Mechanical Engineering	Orthographic Projection	SCALE 1:2		TITLE ROLLER
	OR	UNITS : mm		
		Date	Checked	PROJECT ROLLER ASSEMBLY
	Project Supervisor			
	Workshop Technician			STUDENT NAME LOUW BUTLER
	Technical Manager			TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za



MATERIAL: ALUMINIUM
2 INSTANCES REQUIRED

University of Kwa-Zulu Natal School of Mechanical Engineering	Orthographic Projection	SCALE 1:3	TITLE LOAD BED MOTOR CROSS BRACE
	OR	UNITS : mm	
		Date	Checked	
	Project Supervisor			PROJECT PAYLOAD INTERFACE ASSEMBLY
	Workshop Technician			STUDENT NAME LOUW BUTLER
	Technical Manager			TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za



MATERIAL: ALUMINIUM EXTRUSION
2 INSTANCES REQUIRED

University of Kwa-Zulu Natal School of Mechanical Engineering	Orthographic Projection OR	SCALE 1:1 UNITS : mm	
		Date	Checked
	Project Supervisor		
	Workshop Technician		
	Technical Manager		

TITLE

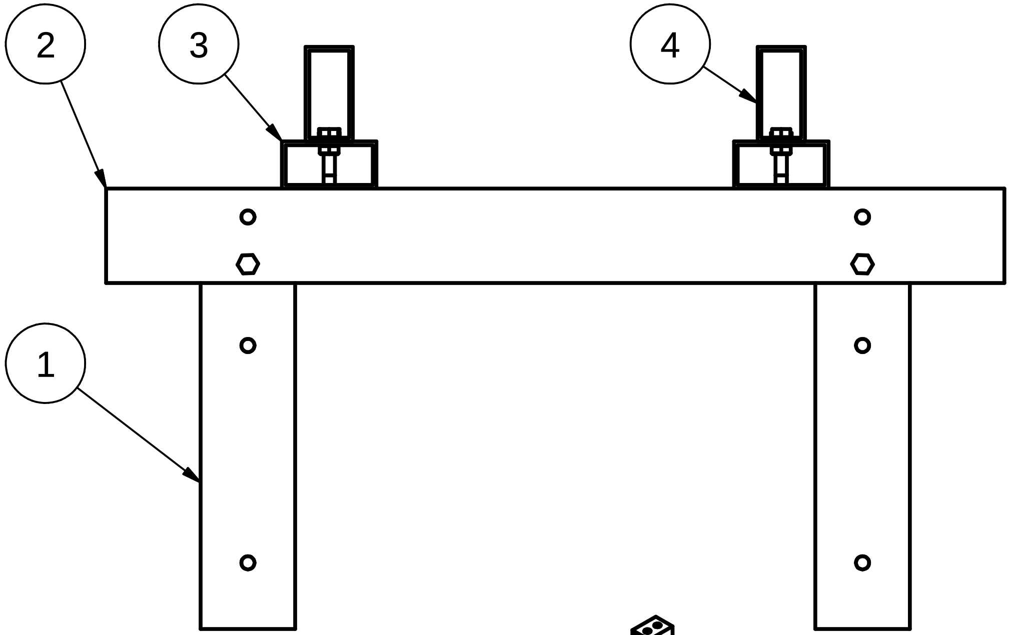
LOAD BED MOTOR MOUNTING

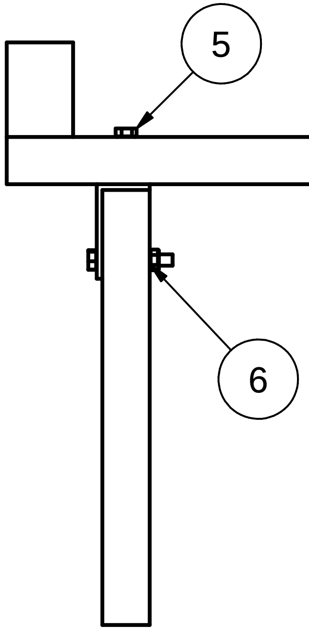
PROJECT

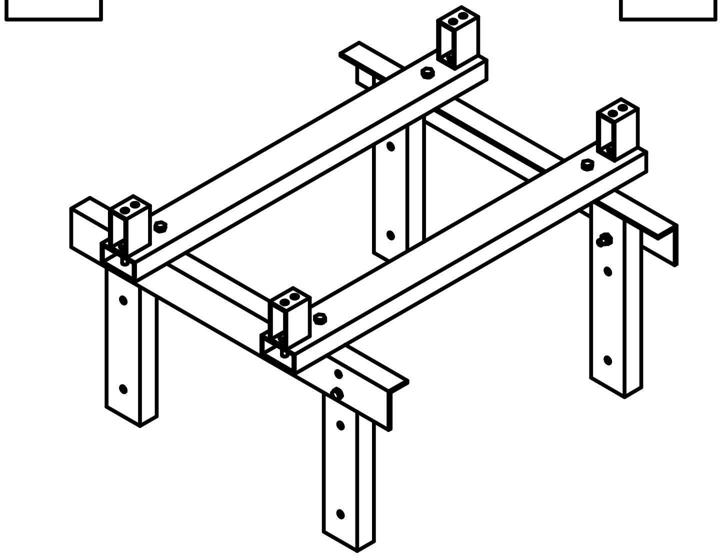
PAYLOAD INTERFACE ASSEMBLY

STUDENT NAME LOUW BUTLER

TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za







Parts List			
ITEM	QTY	PART TITLE	DESCRIPTION
1	4	LOAD BED SUPPORT PILLAR	AL EXTRUSION
2	2	LOAD BED SUPPORT LEVEL	AL EXTRUSION
3	2	LOAD BED CROSS BRACE	AL EXTRUSION
4	4	LOAD BED MOUNTING BLOCK	AL EXTRUSION
5	8	M6 ISO BOLT	
6	16	M6 ISO NUT	

University of Kwa-Zulu Natal

School of Mechanical Engineering

Orthographic Projection

OR

SCALE N/A

UNITS : mm

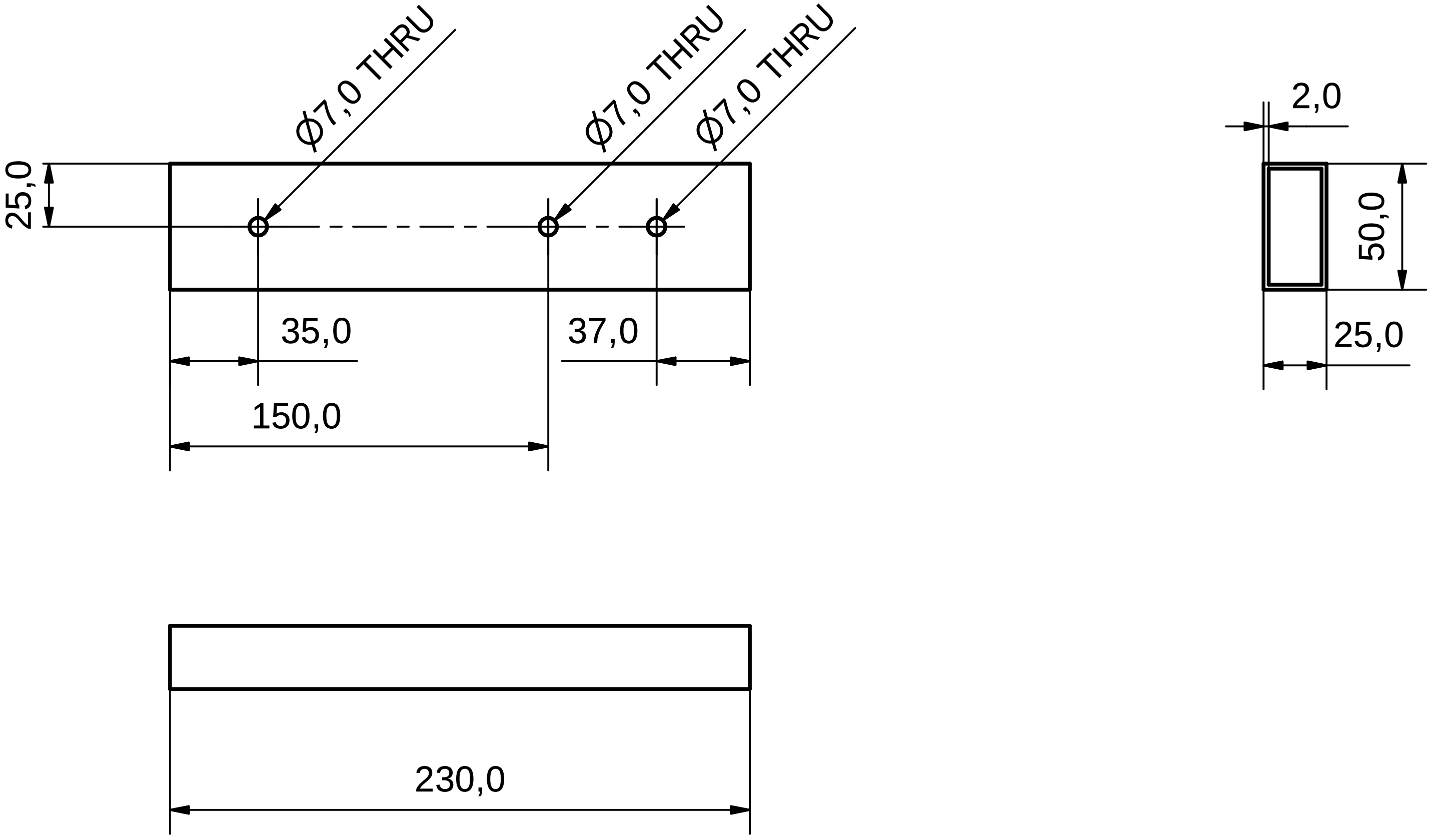
	Date	Checked
Project Supervisor		
Workshop Technician		
Technical Manager		

TITLE
SUPPORT ASSEMBLY

PROJECT
MAIN ASSEMBLY

STUDENT NAME LOUW BUTLER

TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za



MATERIAL: ALUMINIUM EXTRUSION
4 INSTANCES REQUIRED

University of
Kwa-Zulu Natal

School of
Mechanical
Engineering

Orthographic Projection

OR

SCALE 1:4

UNITS : mm

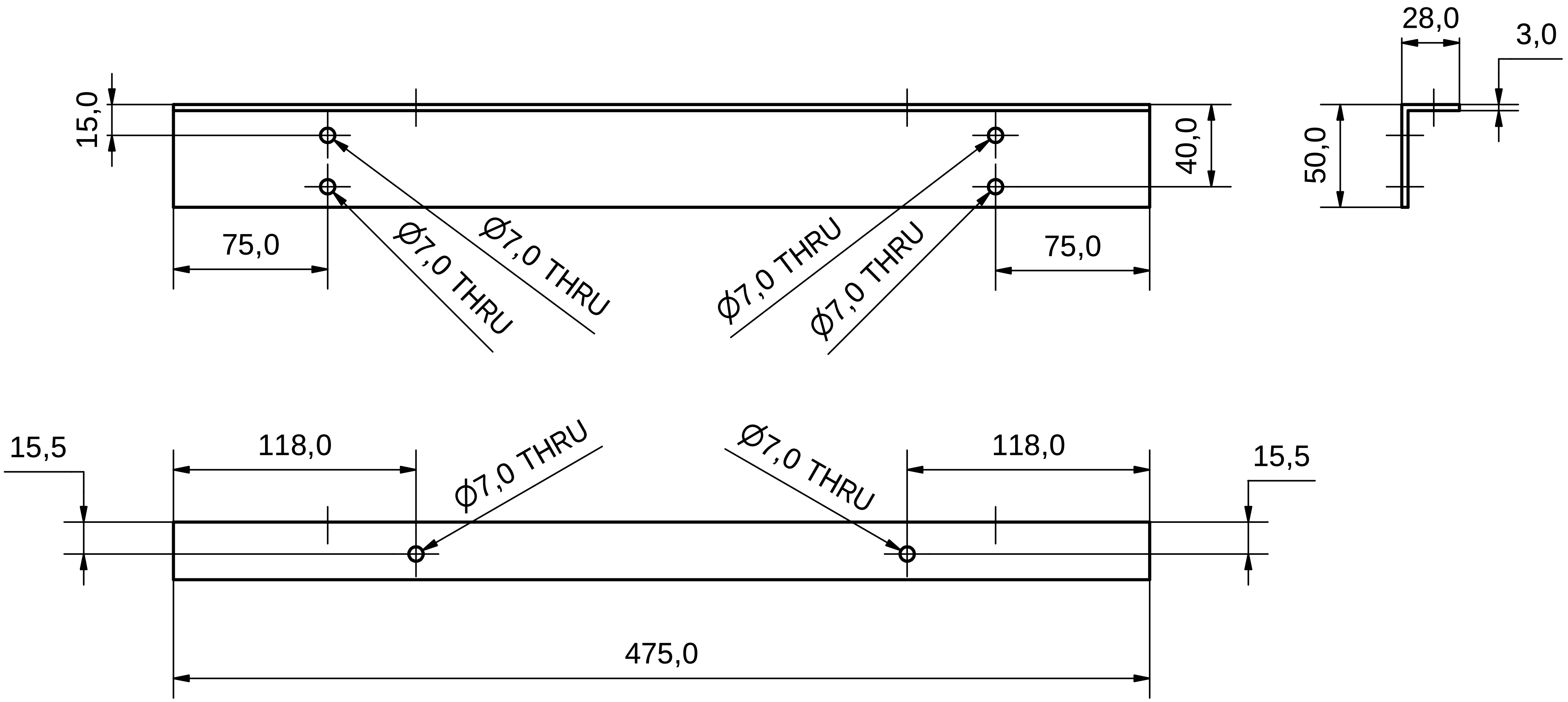
	Date	Checked
Project Supervisor		
Workshop Technician		
Technical Manager		

TITLE
LOAD BED SUPPORT PILLAR

PROJECT
SUPPORT ASSEMBLY

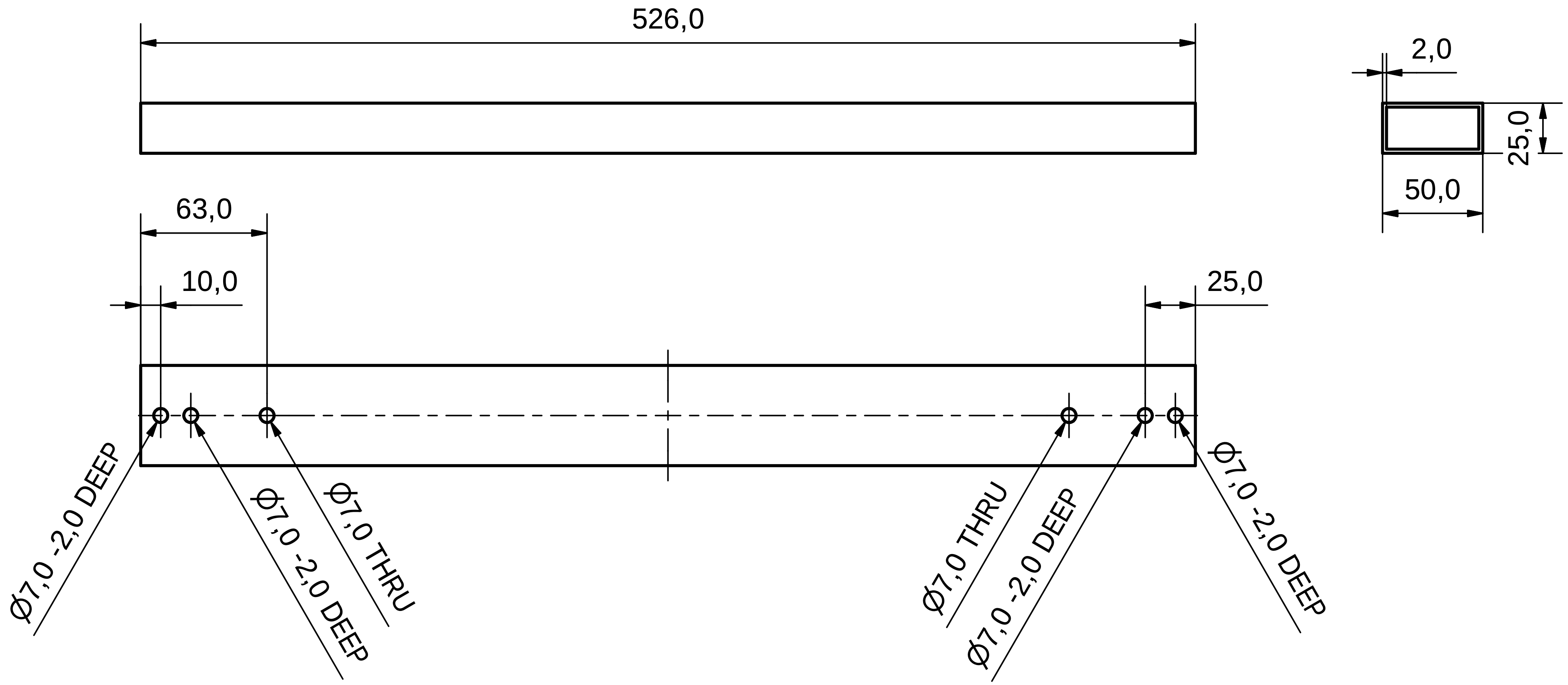
STUDENT NAME LOUW BUTLER

TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za



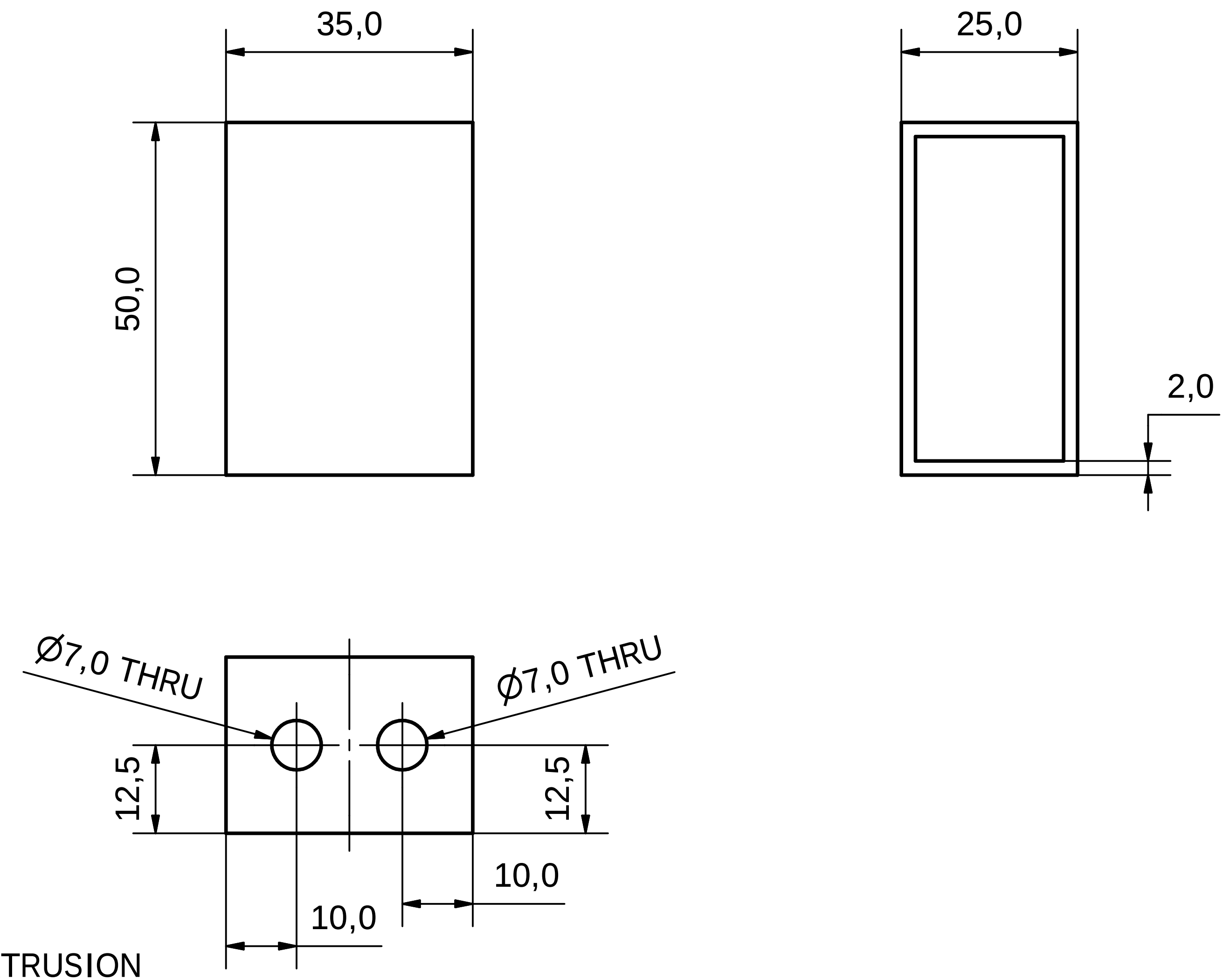
MATERIAL: ALUMINIUM EXTRUSION
2 INSTANCES REQUIRED

University of Kwa-Zulu Natal School of Mechanical Engineering	Orthographic Projection	SCALE 1:3		TITLE LOAD BED SUPPORT LEVEL
	OR	UNITS : mm		
		Date	Checked	PROJECT SUPPORT ASSEMBLY
	Project Supervisor			
	Workshop Technician			STUDENT NAME LOUW BUTLER
	Technical Manager			TEL NO. EXT. 1227 EMAIL 207506509@ukzn.ac.za



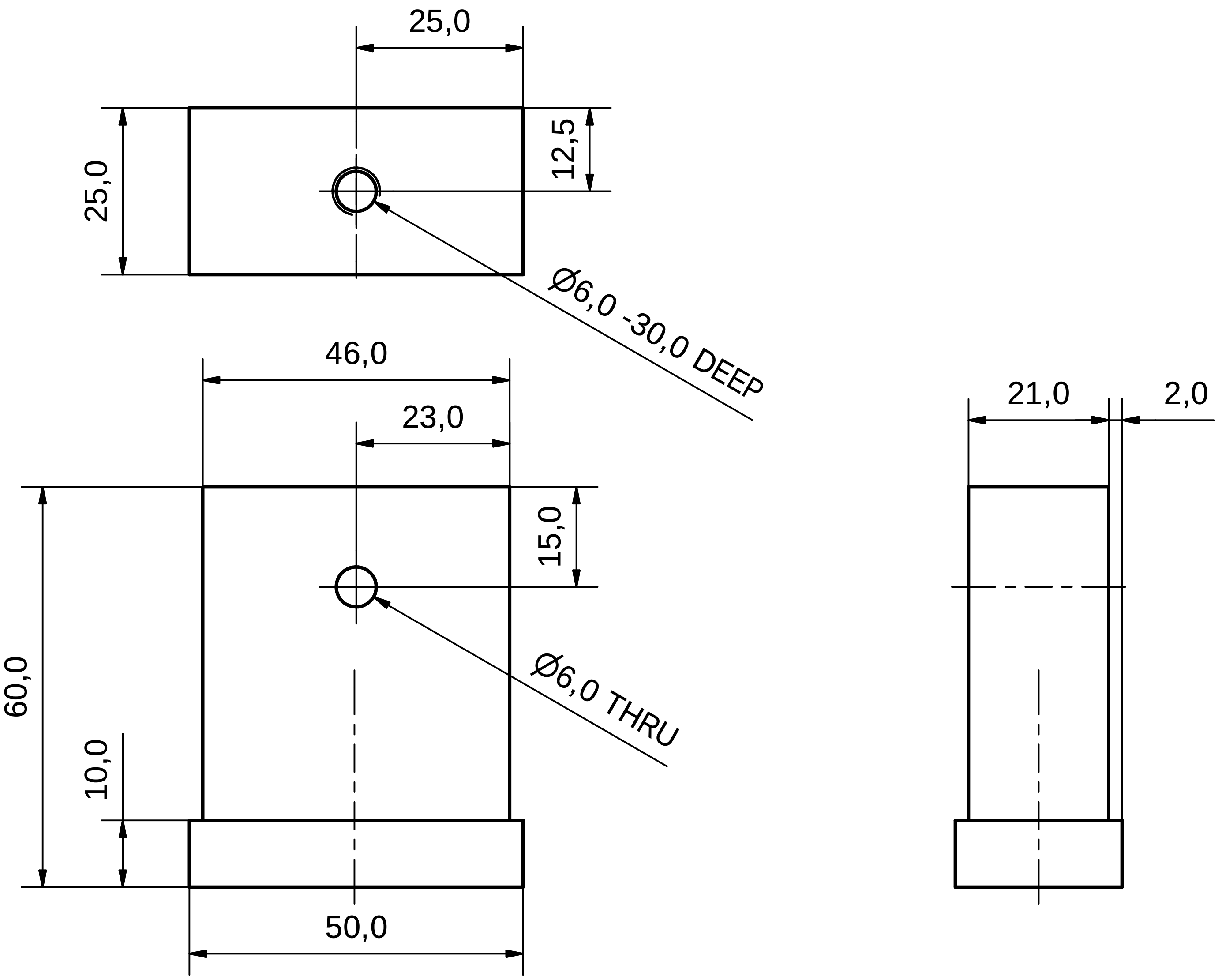
MATERIAL: ALUMINIUM EXTRUSION
2 INSTANCES REQUIRED

University of Kwa-Zulu Natal School of Mechanical Engineering	Orthographic Projection OR	SCALE 1:3		TITLE LOAD BED CROSS BRACE
		UNITS : mm		
		Date	Checked	PROJECT SUPPORT ASSEMBLY
	Project Supervisor			
	Workshop Technician			STUDENT NAME LOUW BUTLER
	Technical Manager			TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za



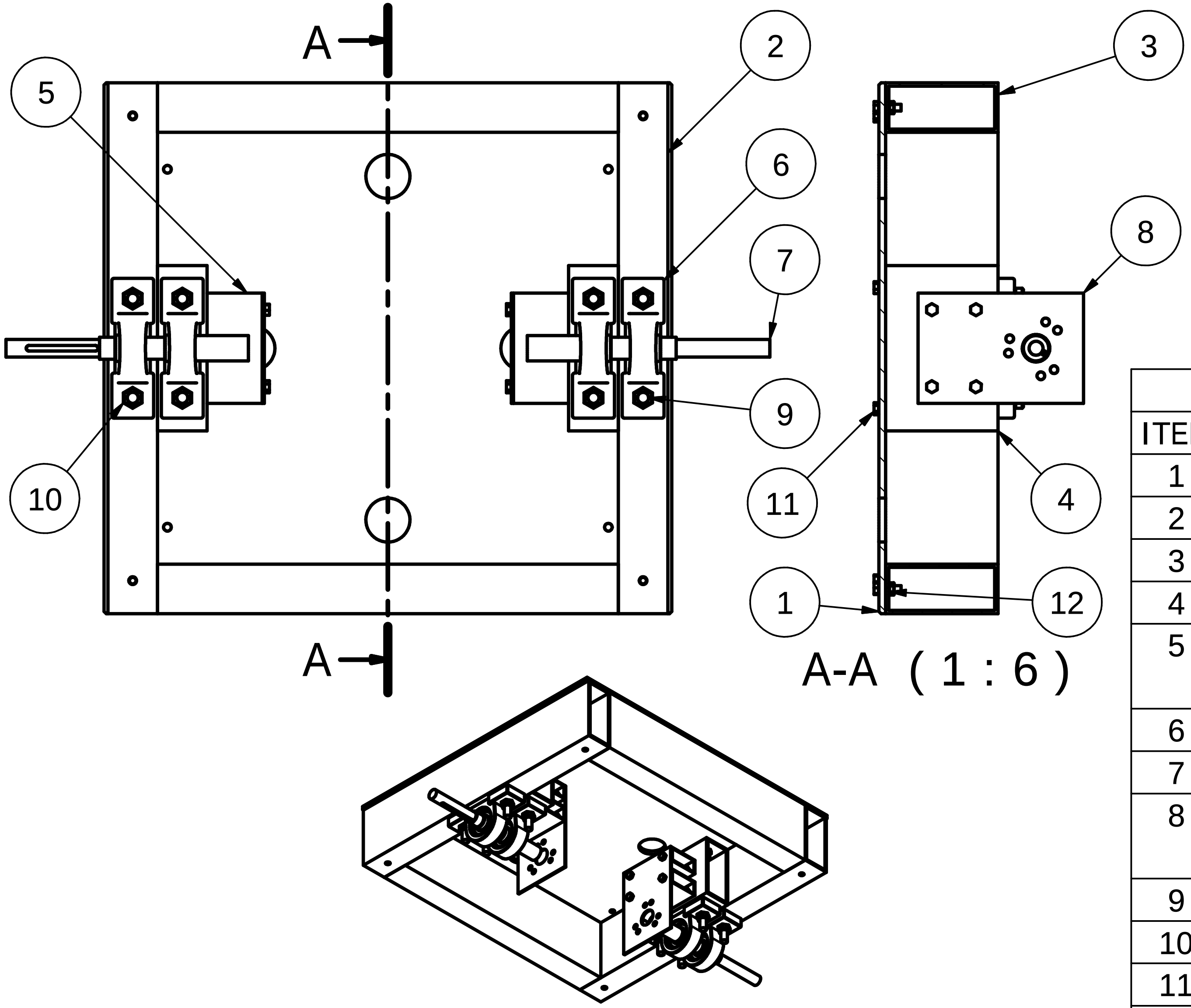
MATERIAL: AUMINIUM EXTRUSION
4 INSTANCES REQUIRED

University of Kwa-Zulu Natal School of Mechanical Engineering	Orthographic Projection	SCALE 1:1		TITLE LOAD BED MOUNTING BLOCK
	OR	UNITS : mm		
		Date	Checked	PROJECT SUPPORT ASSEMBLY
	Project Supervisor			
	Workshop Technician			STUDENT NAME LOUW BUTLER
	Technical Manager			TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za



MATERIAL: ALUMINIUM
4 INSTANCES REQUIRED

University of Kwa-Zulu Natal School of Mechanical Engineering	Orthographic Projection ⊲⊕ OR ⊕⊳	SCALE 1:1 UNITS : mm		TITLE SUPPORT MOUNTING BLOCK
		Date	Checked	PROJECT MAIN ASSEMBLY
	Project Supervisor			
	Workshop Technician			STUDENT NAME LOUW BUTLER
	Technical Manager			TEL NO. EXT 1227 EMAIL 2075065092@ukzn.ac.za



A-A (1 : 6)

Parts List			
ITEM	QTY	PART TITLE	DESCRIPTION
1	1	BASE PLATE	AL PLATE
2	2	MAIN BASE TUBE	AL EXTRUSION
3	2	CROSS BASE TUBE	AL EXTRUSION
4	2	DRIVE SUPPORT TUBE	AL EXTRUSION
5	4	DRIVE MOTOR MOUNT TUBE	AL EXTRUSION
6	4	PILLOW BLOCK	UCP204
7	2	DRIVE SHAFT	MILD STEEL
8	2	DRIVE MOTOR MOUNT PLATE	MILD STEEL
9	8	M10 ISO BOLT	
10	8	M10 ISO NUT	
11	28	M6 ISO BOLT	
12	28	M6 ISO NUT	

University of
Kwa-Zulu Natal

School of
Mechanical
Engineering

Orthographic Projection

OR

SCALE N/A

UNITS : mm

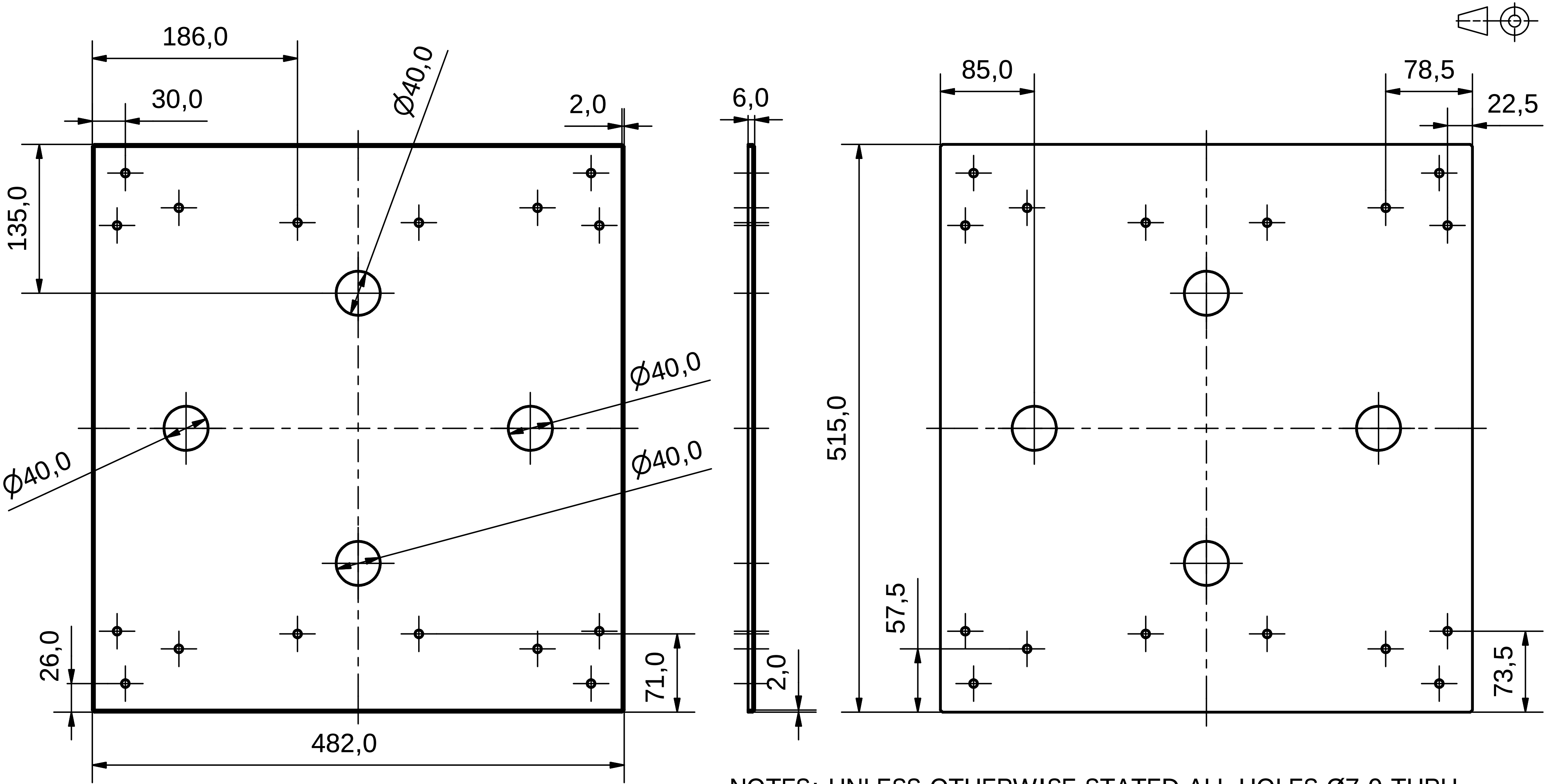
	Date	Checked
Project Supervisor		
Workshop Technician		
Technical Manager		

TITLE
BASE ASSEMBLY

PROJECT
MAIN ASSEMBLY

STUDENT NAME LOUW BUTLER

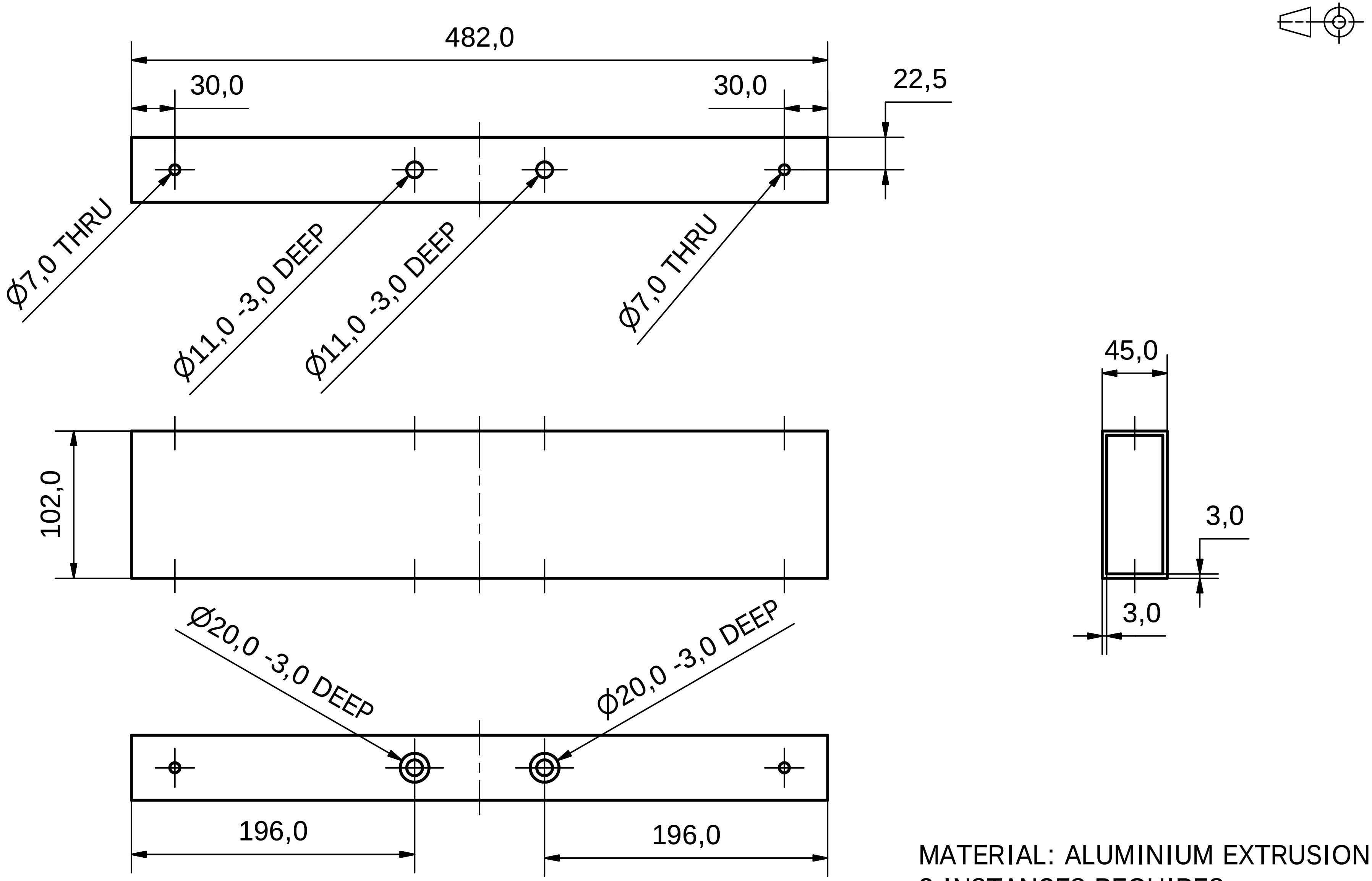
TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za



NOTES: UNLESS OTHERWISE STATED ALL HOLES Ø7,0 THRU
PART IS SYMMETRICAL ABOUT CENTRELINES

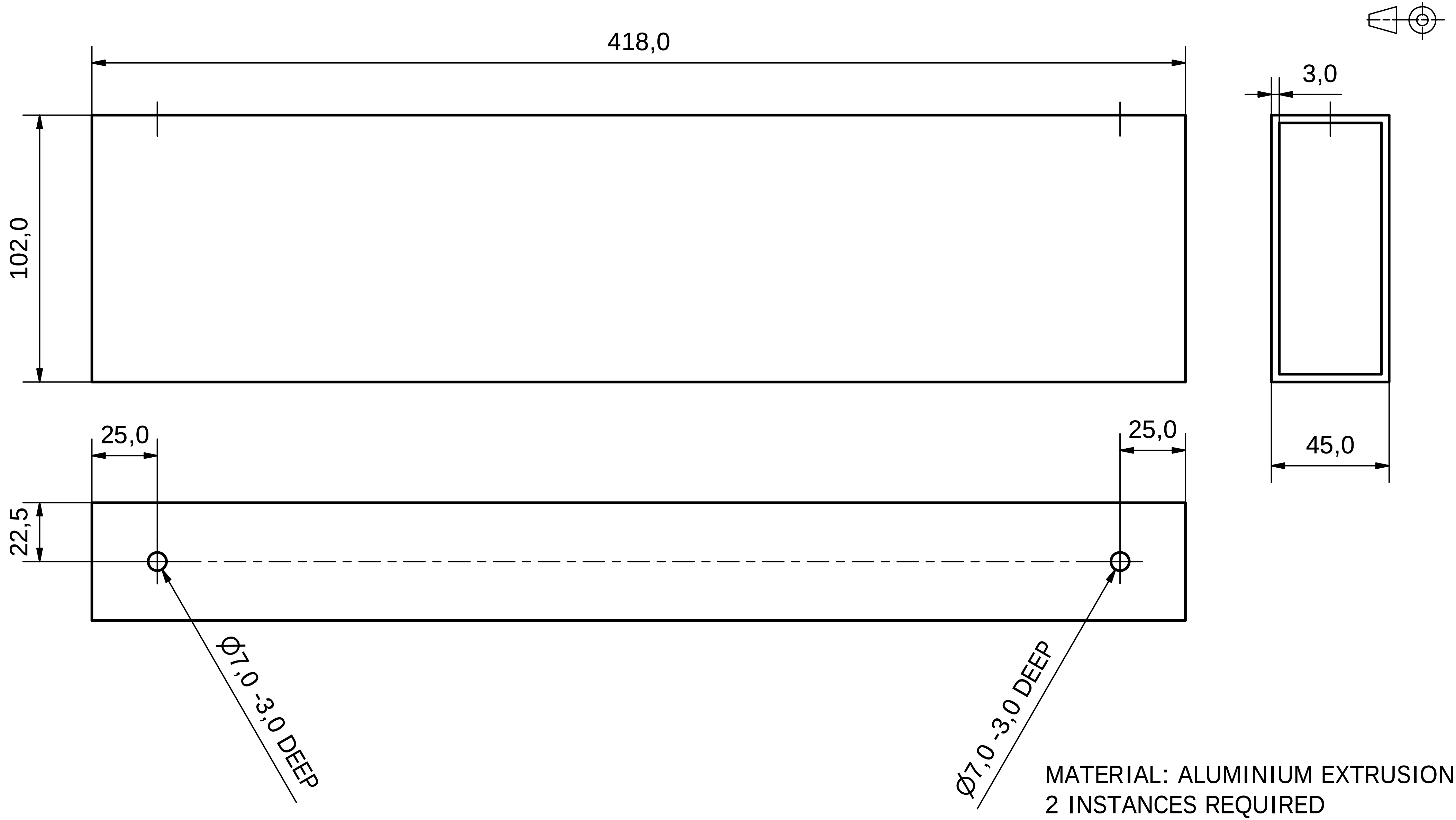
MATERIAL: ALUMINIUM
1 INSTANCE REQUIRED

University of Kwa-Zulu Natal School of Mechanical Engineering	Orthographic Projection	SCALE 1:5		TITLE BASE PLATE
	OR	UNITS : mm		
		Date	Checked	PROJECT BASE ASSEMBLY
	Project Supervisor			
	Workshop Technician			STUDENT NAME LOUW BUTLER
	Technical Manager			TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za



MATERIAL: ALUMINIUM EXTRUSION
2 INSTANCES REQUIRES

University of Kwa-Zulu Natal School of Mechanical Engineering	Orthographic Projection	SCALE 1:4		TITLE MAIN BASE TUBE
	OR	UNITS : mm		
		Date	Checked	PROJECT BASE ASSEMBLY
	Project Supervisor			
	Workshop Technician			STUDENT NAME LOUW BUTLER
	Technical Manager			TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za



MATERIAL: ALUMINIUM EXTRUSION
2 INSTANCES REQUIRED

University of Kwa-Zulu Natal

School of Mechanical Engineering

Orthographic Projection

OR

SCALE 1:2

UNITS : mm

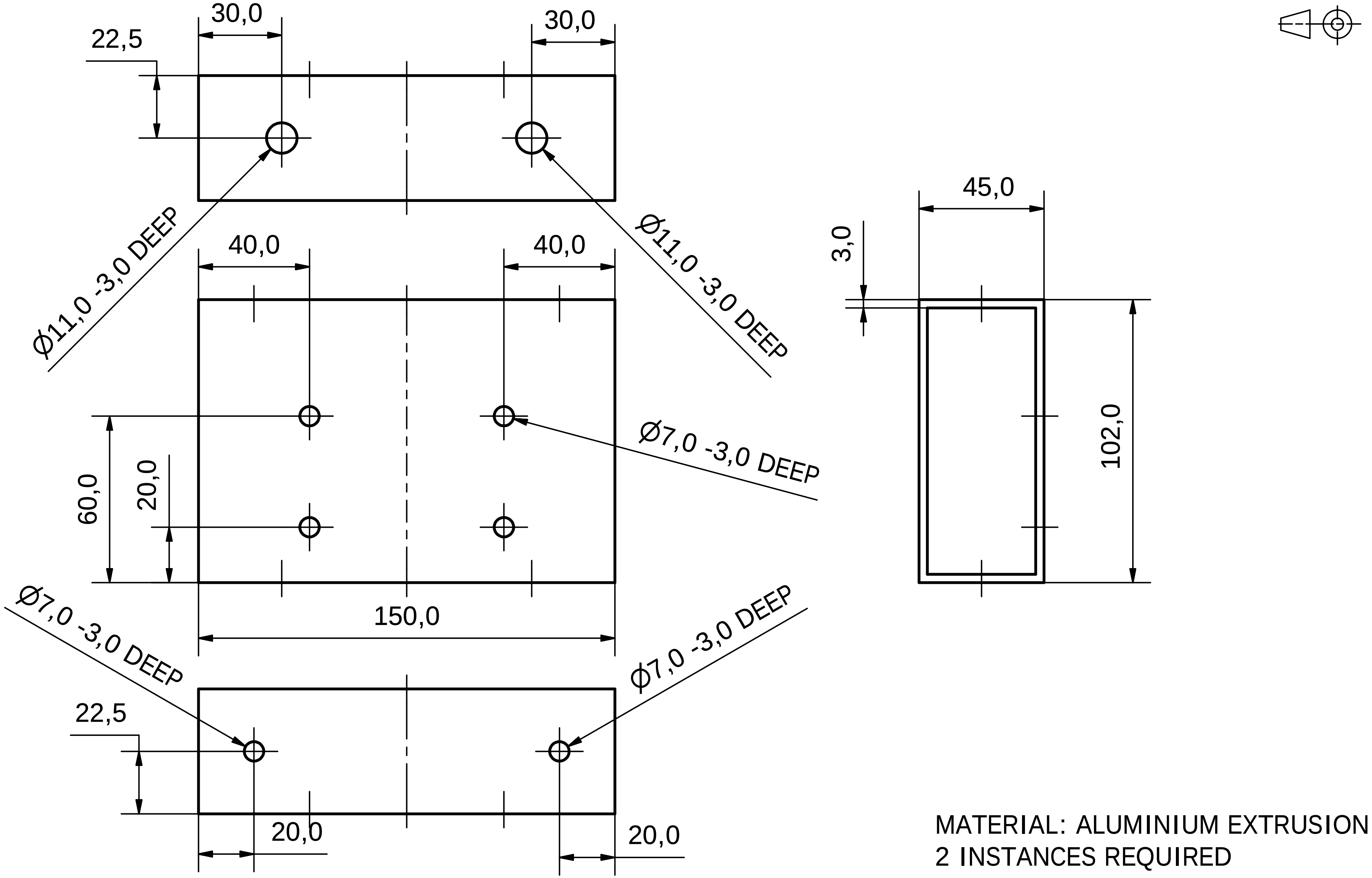
	Date	Checked
Project Supervisor		
Workshop Technician		
Technical Manager		

TITLE
CROSS BASE TUBE

PROJECT
BASE ASSEMBLY

STUDENT NAME 207506509

TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za



MATERIAL: ALUMINIUM EXTRUSION
2 INSTANCES REQUIRED

University of
Kwa-Zulu Natal

School of
Mechanical
Engineering

Orthographic Projection

OR

SCALE 1:2

UNITS : mm

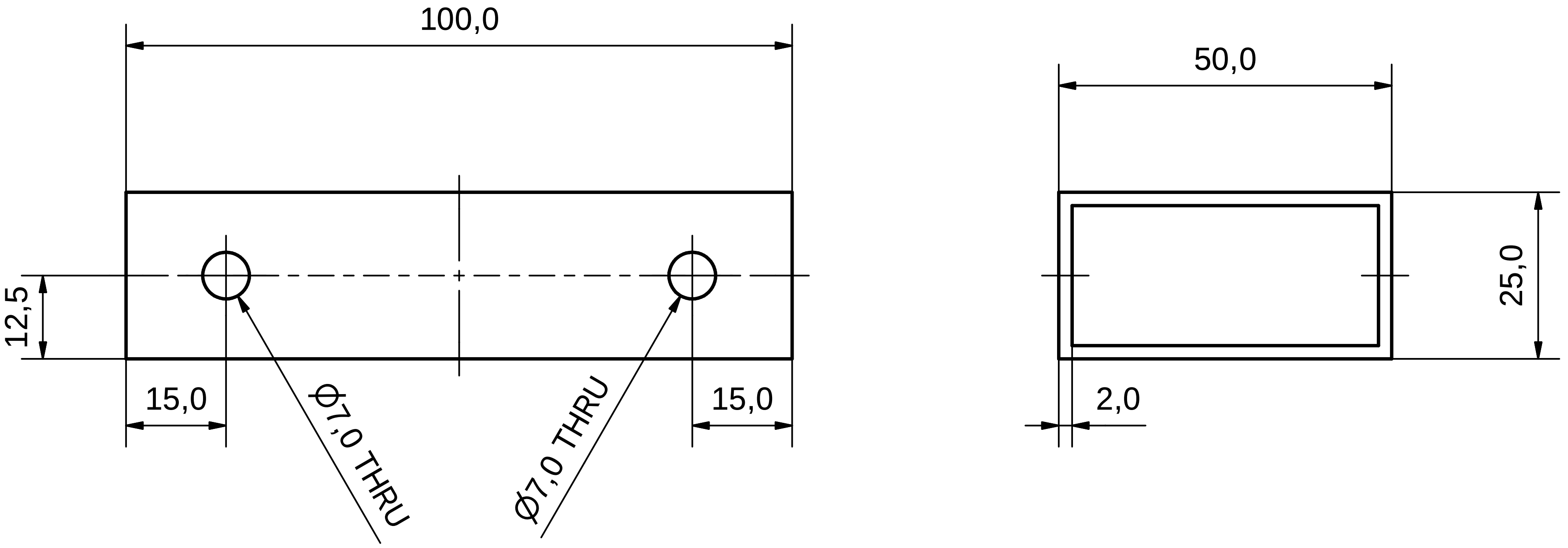
	Date	Checked
Project Supervisor		
Workshop Technician		
Technical Manager		

TITLE
DRIVE SUPPORT TUBE

PROJECT
BASE ASSEMBLY

STUDENT NAME LOUW BUTLER

TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za



MATERIAL: ALUMINIUM EXTRUSION
4 INSTANCES REQUIRED

University of Kwa-Zulu Natal

School of Mechanical Engineering

Orthographic Projection

OR

SCALE 1:1

UNITS : mm

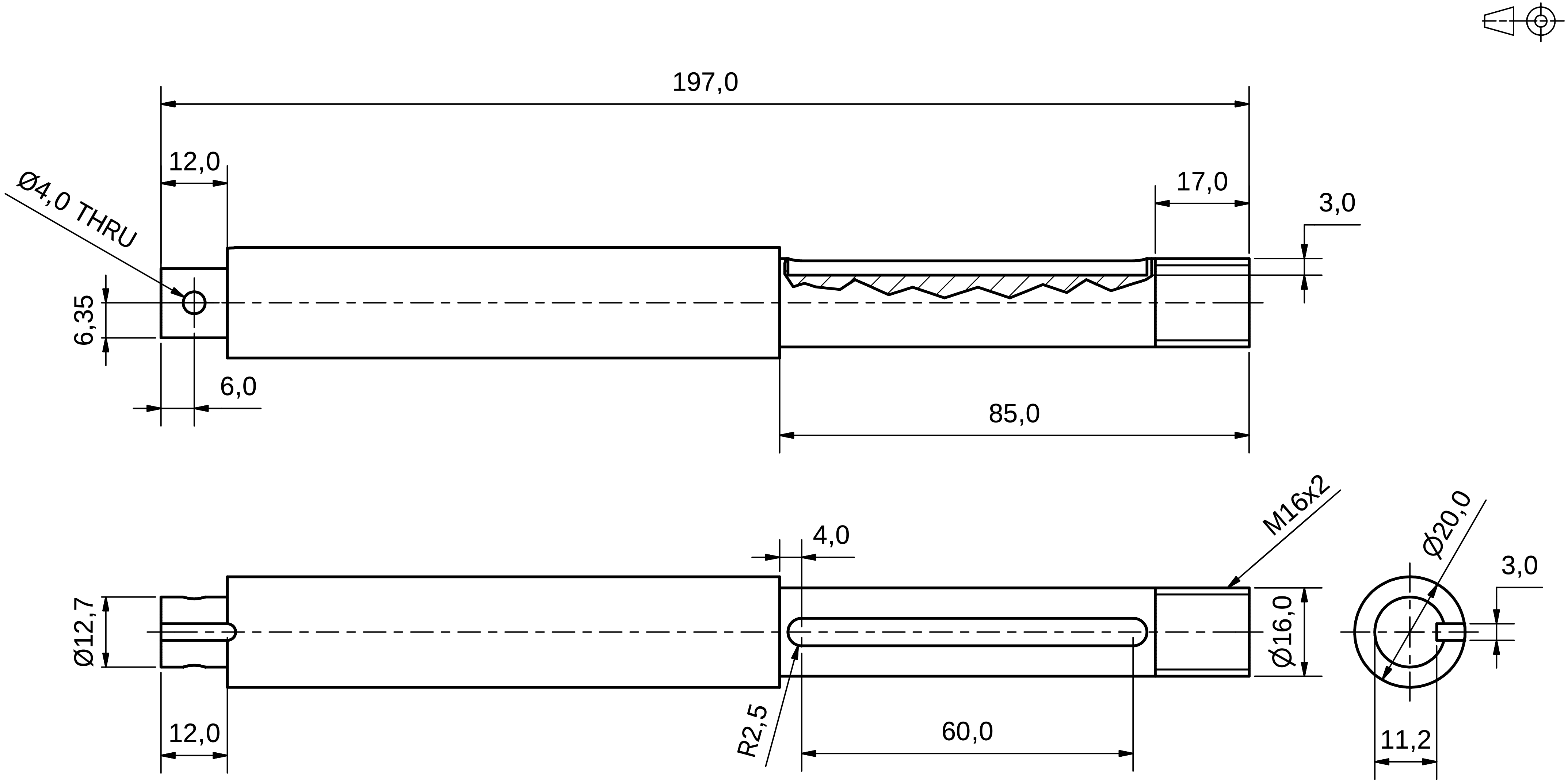
	Date	Checked
Project Supervisor		
Workshop Technician		
Technical Manager		

TITLE
DRIVE MOTOR MOUNT TUBE

PROJECT
BASE ASSEMBLY

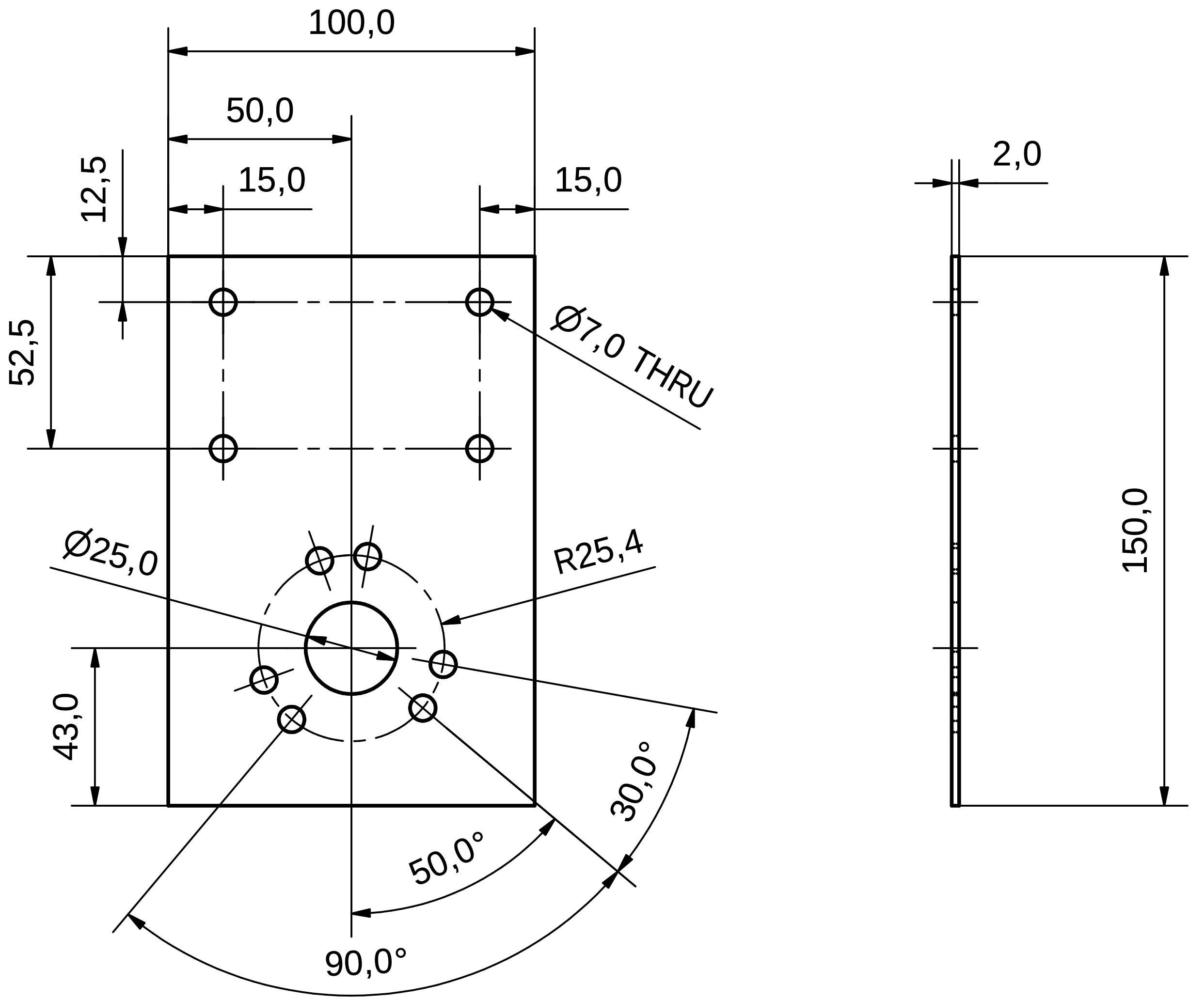
STUDENT NAME LOUW BUTLER

TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za



MATERIAL: MILD STEEL
2 INSTANCES REQUIRED

University of Kwa-Zulu Natal School of Mechanical Engineering	Orthographic Projection	SCALE 1:1		TITLE DRIVE SHAFT
	OR	UNITS : mm		
		Date	Checked	PROJECT BASE ASSEMBLY
	Project Supervisor			
	Workshop Technician			STUDENT NAME LOUW BUTLER
	Technical Manager			TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za



University of Kwa-Zulu Natal

School of Mechanical Engineering

Orthographic Projection

OR

SCALE 1:2

UNITS : mm

	Date	Checked
Project Supervisor		
Workshop Technician		
Technical Manager		

TITLE

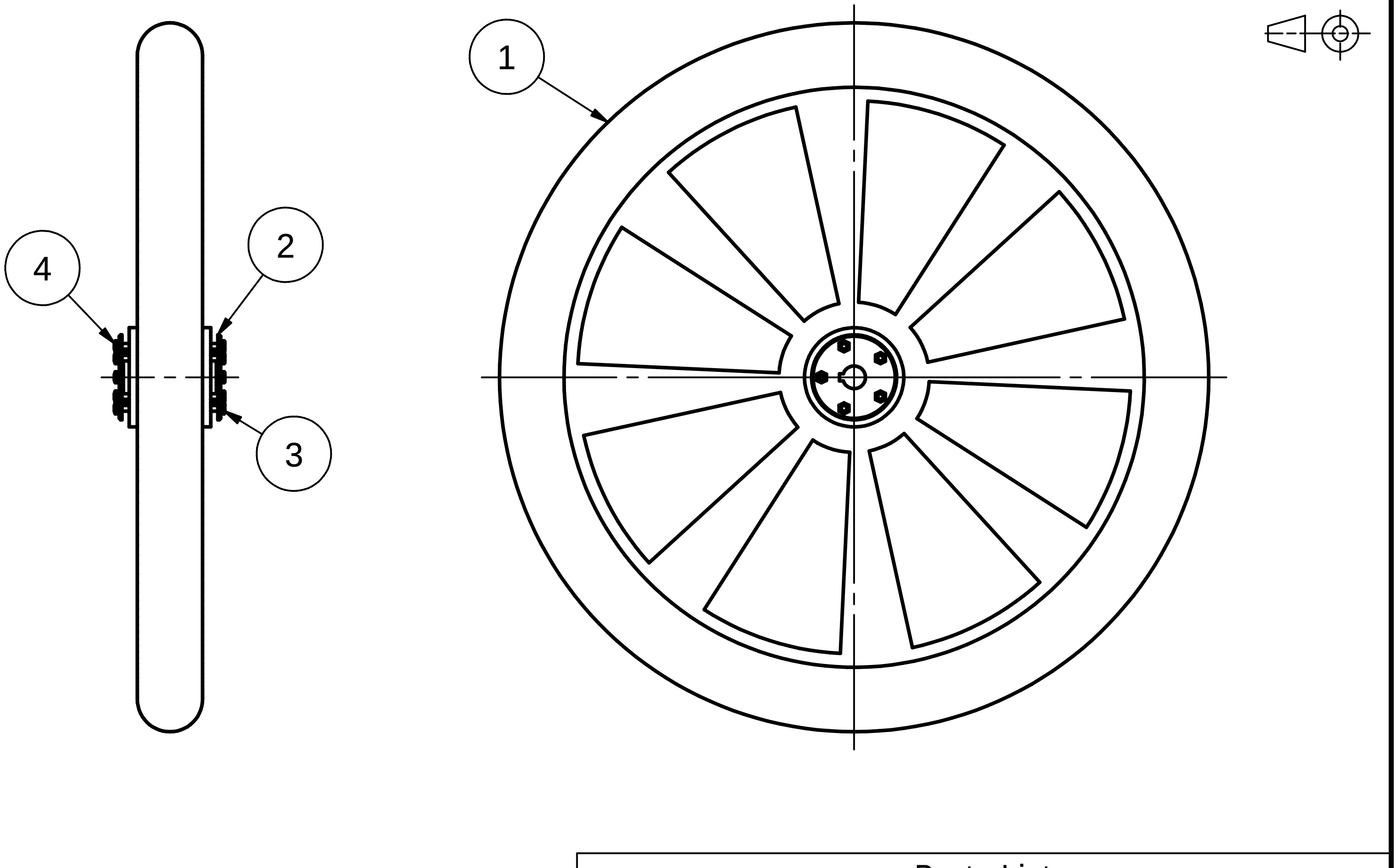
DRIVE MOTOR MOUNT PLATE

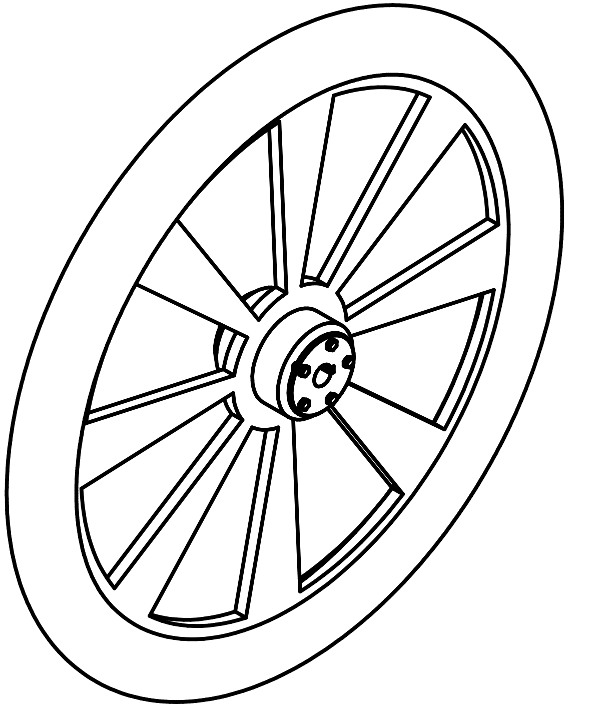
PROJECT

BASE ASSEMBLY

STUDENT NAME LOUW BUTLER

TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za





Parts List

ITEM	QTY	PART TITLE	DESCRIPTION
1	1	WHEEL RIM	MODIFIED BMX RIM
2	2	WHEEL BUSH	MILD STEEL ROUND BAR
3	5	M4 ISO BOLT	
4	5	M4 ISO NUT	

University of
Kwa-Zulu Natal

School of
Mechanical
Engineering

Orthographic Projection

OR

SCALE 1:4

UNITS : mm

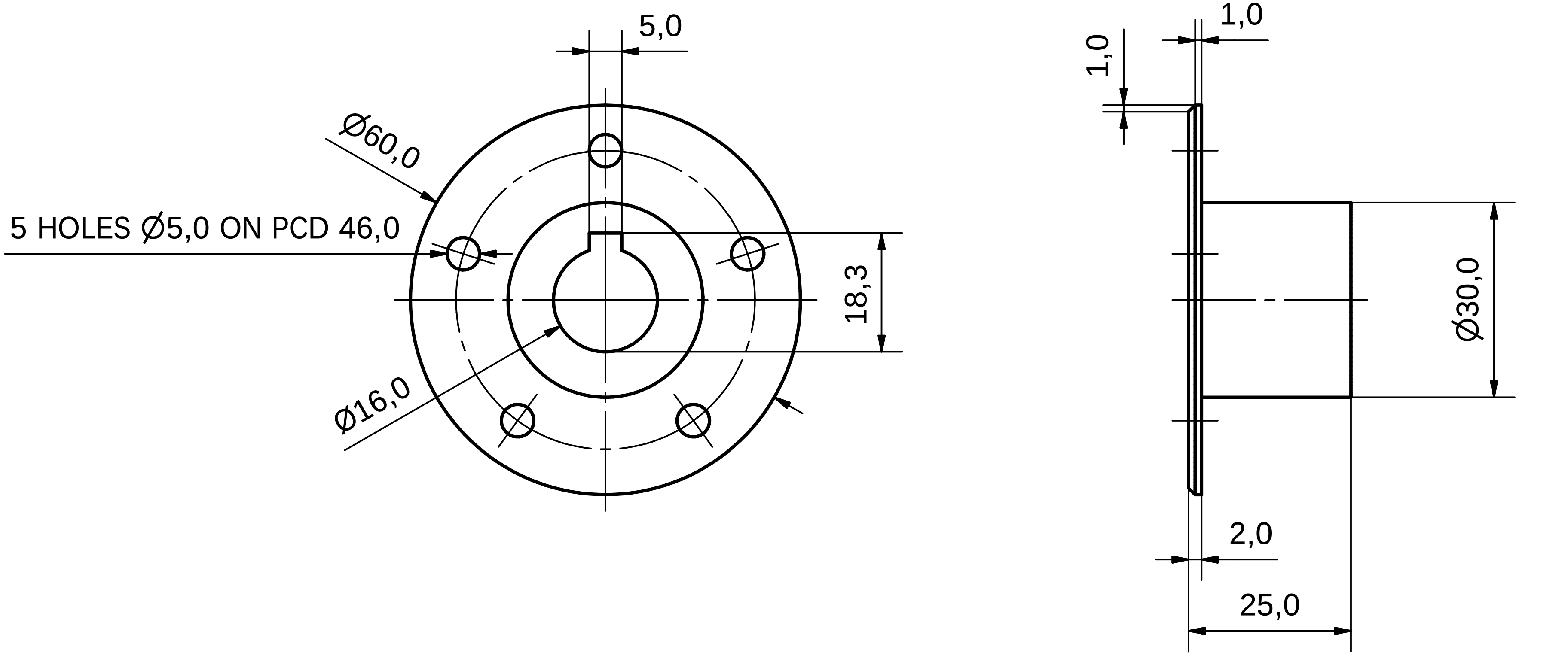
	Date	Checked
Project Supervisor		
Workshop Technician		
Technical Manager		

TITLE
WHEEL ASSEMBLY

PROJECT
MAIN ASSEMBLY

STUDENT NAME LOUW BUTLER

TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za



MATERIAL: MILD STEEL
4 INSTANCES REQUIRED

University of Kwa-Zulu Natal

School of Mechanical Engineering

Orthographic Projection

OR

SCALE 1:1

UNITS : mm

	Date	Checked
Project Supervisor		
Workshop Technician		
Technical Manager		

TITLE

WHEEL BUSH

PROJECT

WHEEL ASSEMBLY

STUDENT NAME LOUW BUTLER

TEL NO. EXT 1227 EMAIL 207506509@ukzn.ac.za

Appendix B

Dynamics Calculations and Drive Motor Selection

This chapter shows calculations carried out to determine the peak power requirements for selection of the drive motors, according to technical engineering specifications. A simplified, lumped mass model was developed for use in these calculations. The parameters of this model are also given here. In order to find the power requirements of the drive motors the dynamics of the platform needed to be investigated.

B.1 Moment of Inertia for a Lumped Model

The mass of the body of the platform has been lumped at the end of a weightless pole. The total mass of the platform was estimated to be 60 kg and was first estimated to consist of three concentrated masses. These are the mass of the payload, M$_L$, the combined mass of the battery pack and payload interface system, M$_B$, and the combined mass of the base and drive motors, M$_M$, each at a distance d$_L$, d$_B$, and d$_M$ away from the wheel axle axis, respectively. This is illustrated in Figure B.1.

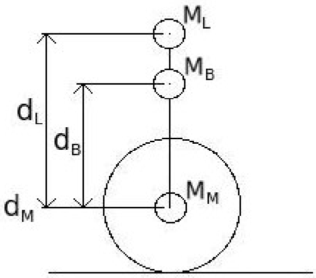


Figure B.1: Initial lumped model simplification.

Subsequently the concentrated masses were combined to form a single concentrated mass at the end of a weightless pole. Mass M$_E$ at a distance of d$_E$ from the wheel axle axis is illustrated in Figure B.2.

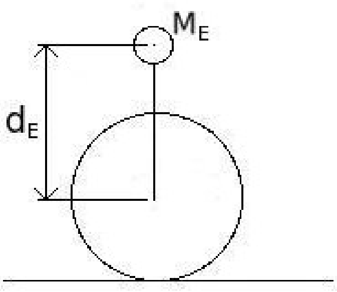


Figure B.2: Final equivalent lumped model simplification.

The first step here is to calculate the equivalent mass moment of inertia for the initial model simplification with the three lumped masses. Equation B.1 shows the expression used.

$$I = \left(M_L.d_L^2\right) + \left(M_B.d_B^2\right) + \left(M_M.d_M^2\right) \tag{B.1}$$

The next step is to convert the initial model into an equivalent model with a single lumped mass at the end of the weightless pole. Equation B.2 was used to calculate the pole length, d_E, for the final equivalent lumped model.

$$I = \left(M_E.d_E^2\right) \tag{B.2}$$

Table B.1 shows values for the variables mentioned in this section.

Variable	Value	Comments
M_L	20 kg	Mass of payload
M_B	15 kg	Mass of batteries and payload interface
M_M	25 kg	Mass of base and drive motors
d_L	0.8 m	Moment arm of payload
d_B	0.6 m	Moment arm of batteries and payload interface
d_M	0.2 m	Moment of base and drive motors
I	19.2 kgm^2	Mass moment of inertia
M_E	60 kg	Total equivalent lumped mass
d_E	**0.566 m**	Calculated moment arm of total lumped mass

Table B.1: Lumped mass model variables and values.

B.2 Tilt Moment Calculations

Using the parameters for the lumped mass model found in the previous section, the torque required to be delivered by the drive motors was calculated. Figure B.3 shows the variables and their location on the force diagram for these calculations.

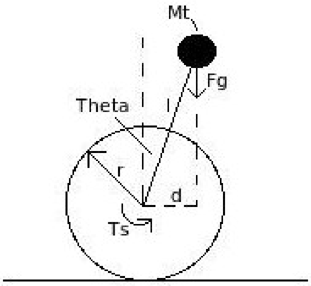


Figure B.3: Tilt moment force diagram for torque calculations.

The moment caused by the body of the platform about the centre of the wheels was calculated for a range of tilt angles, from 0 ° to 15 ° in intervals of 2.5 °. Table B.2 shows the values for the parameters used in these calculations. A gravitational constant of 9.81 m/s^2 was used here.

Variable	Value	Comments
l	0.566 m	Distance between wheel and lumped mass
M_t	60 kg	Total equivalent lumped mass
r	0.254 m	Radius of drive wheels
F_g	588.6 N	Gravitational force
θ	variable	Platform tilt angle
d	variable	Moment arm of lumped gravitational force $d = l.sin\,(\theta)$

Table B.2: Tilt moment parameters and values.

Table B.3 shows the moments and resulting torque for the tilt angle settings as mentioned. The final variable of interest is the torque required to be delivered by each motor at steady state conditions.

Tilt angle $[deg]$	Moment arm $[m]$	Total Moment $[Nm]$	Torque/motor $[Nm]$
0.0	0.0	0.0	0.0
2.5	0.0240	14.15	7.07
5.0	0.0480	28.28	14.13
7.5	0.0719	42.33	21.17
10.0	0.0957	56.32	28.16
12.5	0.1193	70.20	35.10
15.0	0.1426	83.94	41.97

Table B.3: Tilt moments and motor torque versus tilt angle.

As shown in Table B.2 the maximum overall platform weight, including payload weight has been used for these calculations. With this information, it is found that the maximum allowable tilt angle of 5 °, as specified in the mechanical hardware specifications, is an acceptable value. This sets the requirement for continuous operational motor torque at 14.13 Nm.

B.3 Acceleration as a Function of Tilt Angle

In this section the linear acceleration required to recover from the maximum tilt angle is determined. Information from the previous section is used here along with the relevant model parameters, shown in Table B.4.

Variable	Value	Comments
l	0.566 m	Distance between wheels and lumped mass
M_t	60 kg	Total equivalent lumped mass
r	0.254 m	Radius of drive wheels
M_w	1.0 m	Mass of one wheel
I_w	0.129 kgm^2	Moment of inertia of both wheels

Table B.4: Relevant platform parameter definitions and values.

For the purpose of these calculations the simplified model has been separated into two components. The forces and moments exerted one each of these component by each other as well as gravitational and external forces are examined in order to determine the required acceleration variable. Figure B.4 shows the two components.

The relation between the maximum tilt angle and the maximum acceleration of the vehicle can be found by using these free body diagrams and summing forces in the horizontal direction and summing moments about the centre of the wheels for the pendulum body. The sum of the moments about the centre of the wheels, on the wheels is also taken.

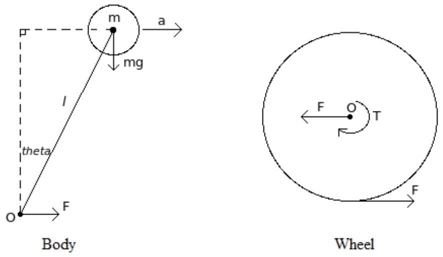


Figure B.4: Free body diagrams of the two components of the platform model.

The pendulum is only accelerating in the horizontal direction for these calculations, thus the angular velocity and -acceleration is zero.

Using the procedure described here Equation B.3 has been derived.

$$a = \frac{mgl}{mr - \frac{I_w}{r}} \sin(\theta) \tag{B.3}$$

This formula was used to calculate the resulting acceleration for different values of tilt angle. The results are shown in Table B.5. Using this relation, one can set the maximum acceleration of the vehicle and find the maximum allowable tilt angle from this, and vice versa.

Tilt angle $[deg]$	Acceleration $[m/s^2]$
0	0.0
1	0.39
2	0.79
3	1.18
4	1.58
5	1.97
6	2.36
7	2.76
8	3.15
9	3.54
10	3.93
11	4.31
12	4.70
13	5.09
14	5.47
15	5.85

Table B.5: Linear acceleration required for corresponding tilt angle.

Setting the maximum allowable tilt angle to 5 °, or 0.087 rad, gives an acceleration of about 2 m/s^2. This is an acceptable value for steady acceleration. The maximum velocity possible must be higher that the cruising velocity, in order for the vehicle to be able to slow down from cruising velocity. This is due to the fact that the base of the platform needs to accelerate ahead of the vehicle, in other words the vehicle will have to tilt backwards in order to slow down, and stop.

B.4 Motor Power Output

From the velocity and acceleration variables as well as the platform model parameters the power requirements of the drive motors may be calculated. The expression for power in rotational movement is shown in Equation B.4.

$$P = T.\omega \tag{B.4}$$

This expression was used to calculate power requirements for normal operating conditions as well as peak conditions where the platform is tilted to its maximum limit and it is required to reach a steady state at the specified allowable velocity. Table B.6 shows the relevant parameters and the calculated values for the power requirement for normal operating conditions as well as for peak conditions.

	Normal conditions	Peak conditions
Maximum linear velocity $[m/s]$	1.5	3
Wheel velocity $[rad/s]$	5.91	5.91+
Linear accelerations $[m/s^2]$	0	2
Motor torque $[Nm]$	14.13	28.26
Motor power $[W]$	**83.44**	**166.89**

Table B.6: Model parameters and power requirements.

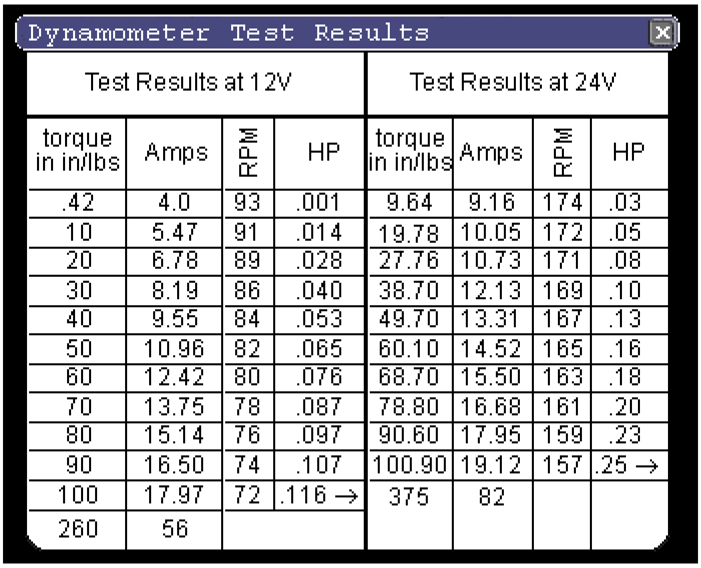
Here, peak conditions refer to the platform traveling at cruising velocity and having to stop suddenly, the motors are required to accelerate to approximately double their speed in order to tilt the platform backwards to achieve negative nett acceleration in order for the platform can stop as quickly as possible. These conditions will only be for a very short time period, however the motor that is selected must be able to deliver this kind of peak power.

The main considerations for the selection of the appropriate motor are as follows:

- Supply voltage of 24 V available

- Power delivery of at least 170 W
- Availability of product
- Price of product

The motor that has been selected for implementation on the platform is the NPC 41250 worm-geared DC motor, produced by a company known as National Power Chair, based in the United States of America. Figure B.5 shows dynamometer test results for this motor. This motor is able to deliver 186 W of power.



Test Results at 12V				Test Results at 24V			
torque in in/lbs	Amps	RPM	HP	torque in in/lbs	Amps	RPM	HP
.42	4.0	93	.001	9.64	9.16	174	.03
10	5.47	91	.014	19.78	10.05	172	.05
20	6.78	89	.028	27.76	10.73	171	.08
30	8.19	86	.040	38.70	12.13	169	.10
40	9.55	84	.053	49.70	13.31	167	.13
50	10.96	82	.065	60.10	14.52	165	.16
60	12.42	80	.076	68.70	15.50	163	.18
70	13.75	78	.087	78.80	16.68	161	.20
80	15.14	76	.097	90.60	17.95	159	.23
90	16.50	74	.107	100.90	19.12	157	.25 →
100	17.97	72	.116 →	375	82		
260	56						

Figure B.5: Dynamometer results for the NPC 41250 worm-geared DC motor.

Appendix C

Mechanical Design Calculations

This chapter shows mechanical design calculations carried out for the design of the drive shafts, their supporting bearings, the payload interface system shafts, as well as the bearings mounted on the payload interface system shafts.

C.1 Drive Shaft Design

Basic force and moment balance procedures were carried out to determine the shear forces and bending moments in the shafts coupling the drive motors to the wheels. A stress analysis was carried out to determine whether the shear and bending stresses in the shafts are acceptable. Shaft parameters, and variable refer to Figure C.1.

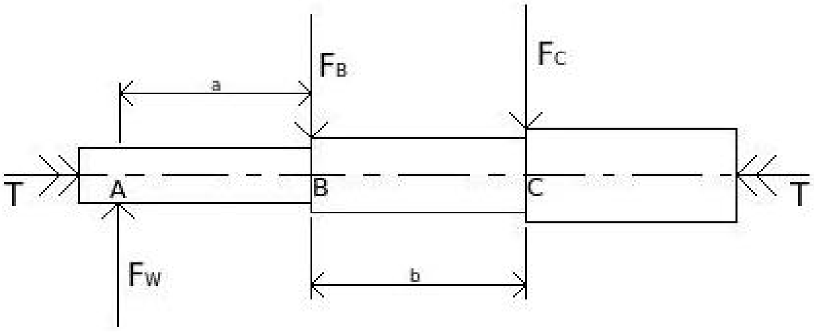


Figure C.1: Drive shaft force diagram.

Forces F_B and F_C are exerted on the shaft by two supporting bearings. The drive shaft is stepped in order for these bearings to be located effectively. Table C.1 lists the drive shaft parameters and forces exerted.

Variable	Value	Comments
M	60 kg	Total mass of platform
a	0.06 m	Distance from A to B
b	0.05 m	Distance from B to C
L_s	0.197	Mass of base and drive motors
F_w	294.3 N	Force on each shaft due to platform weight
F_B	647.46 N	Calculated exerted force at node B
F_C	-353.16 N	Calculated exerted force at node C
T	28.26 Nm	Maximum applied torque
D_a	0.016 m	Diameter at A
D_b	0.020 m	Diameter at B
D_c	0.023 m	Diameter at C
A_a	$2.01 * 10^{-4}\ m^2$	Area at A
A_b	$3.14 * 10^{-4}\ m^2$	Area at B
A_c	$4.15 * 10^{-4}\ m^2$	Area at C
I_a	$3.22 * 10^{-9}\ m^4$	Second moment of inertia at A
I_b	$7.85 * 10^{-9}\ m^4$	Second moment of inertia at B
I_c	$1.37 * 10^{-8}\ m^4$	Second moment of inertia at C
J_a	$6.43 * 10^{-9}\ m^4$	Polar second moment of inertia at A
J_b	$1.57 * 10^{-8}\ m^4$	Polar second moment of inertia at B
J_c	$2.75 * 10^{-8}\ m^4$	Polar second moment of inertia at C

Table C.1: Drive shaft parameters and exerted forces.

The shaft parameters in Table C.1 were used in the analysis of the shear, bending, and torsional stresses in the shaft. Values for these stresses are shown in Table C.2.

The Mohr's circle technique was used to combine the stresses at the critical sections in order to determine the maximum stresses [4]. The maximum calculated stresses have been found to fall well within the physical limits of the material used for the drive shafts, which is EN36 mild steel.

Variable	Value	Comments
F_a	294.3 N	Shear force in section a
τ_{sa}	1.46 MPa	Shear stress in section a
F_b	353.16 N	Shear force in section b
τ_{sb}	1.12 MPa	Shear stress in section b
M_a	0	Bending moment at A
M_b	17.66 Nm	Bending moment at B
M_c	0	Bending moment at C
σ_{ba}	43.91 MPa	Bending stress at B on side of A
σ_{bc}	22.48 MPa	Bending stress at B on side of C
T	28.26 Nm	Torque throughout length of shaft
T_{Ta}	35.14 MPa	Maximum torsional stress at A
T_{Tb}	17.99 MPa	Maximum torsional stress at B
T_{Tc}	11.83 MPa	Maximum torsional stress at C
T_a	35.17 MPa	Maximum shear stress at A
T_b	18.03 MPa	Maximum shear stress at B
T_c	11.83 MPa	Maximum shear stress at C

Table C.2: Summary of stresses occurring in the drive shafts.

C.2 Drive Shaft Support Bearings

Deep groove ball bearings were selected to support the drive shafts on the platform. Calculations have been carried out in order to determine whether the bearings corresponding to the size of the shafts as designed in the previous section are strong enough for the application. Figure C.2 shows the positions of the bearings along the length of the shafts.

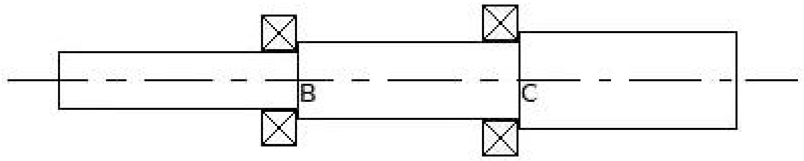


Figure C.2: Drive shaft bearing positioning.

Table C.3 summarises the calculations carried out in this design step. All variables used are explained in the table. The dynamic load, C is calculated using Equation C.1.

$$C = F_{radial}.L^{\frac{1}{3}} \tag{C.1}$$

The required dynamic load capacities for bearings at nodes B and C as calculated fall well within the capacities of bearings with sizes as specified by shaft design calculations.

Variable	Value	Comments
D_B	0.016 m	Minimum inner diameter of bearing B
D_C	0.02 m	Minimum inner diameter of bearing C
F_B	647.46 N	Calculated radial force at bearing B
F_C	353.16 N	Calculated radial force at bearing C
n_d	5.91 rad/s	Design speed applicable to both bearings
	56.39 rpm	Design speed converted to revolutions per minute
L_{10}	36000 hrs	Design life
		From 24 hrs/day, 300 dys/yr, 5 yrs
L	$121.8 * 10^6$ revs	Design life converted to number of revolutions
C_B	3209.44 N	Dynamic load capacity required at bearing B
C_C	1750.60 N	Dynamic load capacity required at bearing C

Table C.3: Summary of bearing design calculations for drive shaft support bearings.

Bearings with designations of ISO 6002, and ISO 16004 have been selected for use at B and C, respectively.

C.3 Payload Interface Shaft Design

As described in Section 4.2, the payload interface consists of seven mild steel shafts mounted between two Aluminium channel sections. Two nylon rollers are fitted onto each shaft in a two sets of ball bearings. The loading situation for these shafts has been simplified such that the total weight of the maximum payload, 20 kg, is concentrated in the centre of a single shaft. Figure C.3 shows this approximated force distribution.

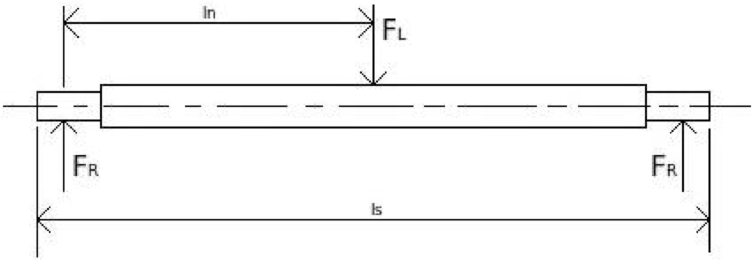


Figure C.3: Payload interface system shaft force diagram.

Here, the end supports may be approximated as clamped joint supports. Table C.4 lists the physical parameter of the shafts as they are to be implemented. A stress analysis was carried out in order to determine whether the shaft dimensions are acceptable.

Variable	Value	Comments
F_L	196.2 N	Exerted force due to payload weight
F_R	98.1 N	Resultant force at supports
l_s	0.46 m	Total length of shaft
l_r	0.43 m	Distance between supports
l_n	0.215 m	Distance between nodes
l_j	0.41 m	Length of major diameter section
D_j	0.012 m	Major diameter
D_n	0.010 m	Minor diameter
I_j	$1.02 * 10^{-9}\ m^4$	Second moment of inertia at major diameter
I_n	$4.91 * 10^{-10}\ m^4$	Second moment of inertia at minor diameter

Table C.4: Payload interface shaft parameters.

The stress analysis was carried out to determine, specifically, whether the deflection of the shaft under the proposed loading conditions is within a limit of 0.5 mm. Table C.5 shows the stresses and ultimate total deflection as calculated using basic beam deflection calculations.

Variable	Value	Comments
T_j	0.867 MPa	Maximum shear stress in major diameter section
T_n	1.25 MPa	Maximum shear stress in minor diameter section
M_s	0.98 Nm	Bending moment at shoulder of shaft
M_c	21.09 Nm	Bending moment at centre of shaft
σ_{sj}	5.78 MPa	Maximum bending stress at shoulder on major diameter side
σ_{sn}	9.99 MPa	Maximum bending stress at shoulder on minor diameter side
σ_c	124.33 MPa	Maximum bending stress, occurring at centre of shaft
δ_{max}	-0.399 mm	Maximum shaft deflection, occurring at centre of shaft

Table C.5: Payload interface shaft stresses and deflection.

From the results shown in Table C.5 it can be seen that the deflection of the proposed shafts fall well within the limits set for the design. material used for fabricating the shafts is EN36 mild steel with a modulus of elasticity of 200 GPa. As shown in Appendix A the shaft has been designed with a 1 mm high step at the centre of the shaft, this has been added to be able to locate the ball bearings effectively.

C.4 Payload Interface Supports Bearings

The nylon rollers, mounted on the payload interface shafts, are supported by ball bearings. Figure C.4 shows the positioning of the bearings designed for here.

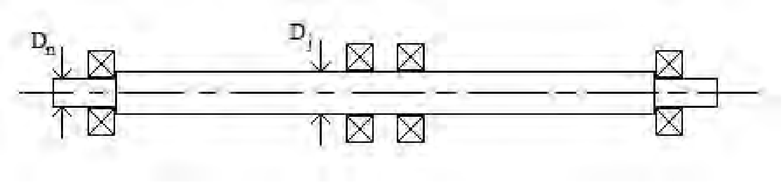


Figure C.4: Payload interface system shaft force diagram.

Table C.6 shows the shaft parameters relevant to the design of ball bearings for supporting the nylon rollers along with calculated design variables.

Variable	Value	Comments
D_j	0.012 m	Major diameter
D_n	0.010 m	Minor diameter
P_{dj}	98.1 N	Radial force at bearing at major diameter
P_{dn}	98.1 N	Radial force at bearing at minor diameter
n_d	3.33 rad/s	Design speed
	31.83 rpm	Design speed converted to revolutions per minute
L_{10}	15000 hrs	Design life
		From 10 hrs/day, 300 dys/yr, 5 yrs
L	$28.65 * 10^6$ revs	Design life converted to number of revolutions
C_d	300.16 N	Dynamic load capacity required for all bearings

Table C.6: Payload interface shaft parameters and bearing design calculations.

The dynamic load capacity required for these bearings, as calculated using Equations C.1, is relatively low, which specifies that small bearings may be implemented here. It is for this reason that the limiting factor is the strength of the shafts, thus the bearings selected for this application are of designation ISO 61802, and ISO 61800, for the major and minor diameter sections, respectively.

Appendix D

Mathematical System Model and Simulation

This chapter shows the derivation of the mathematical model that simulations, for investigation of control system dynamics, are based on. This chapter also shows the MatLab script file used to enter system variables and generate feedback control gains and the SimuLink block diagram used to simulate the control system responses due to system characteristic variations.

D.1 Mathematical System Model Derivation

A simplified mathematical model of the system was derived, based on the model derived by Ooi [36]. This is a linearised model that includes the dynamics of the DC motors that have been implemented. Superposition was used to derive the model, where the dynamics of the wheels and that of the body of the platform were modelled separately, and subsequently concatenated along with the motor model to produce the complete system model. The free body diagram of the DC motor can be seen in Figure D.1.

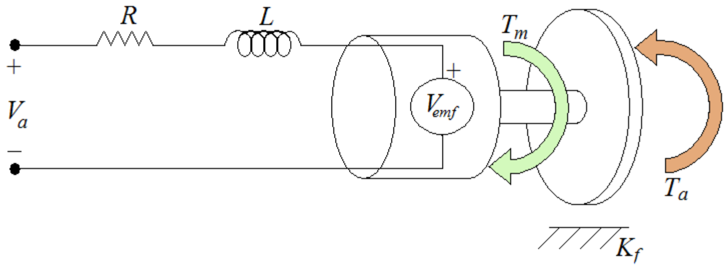


Figure D.1: Free body diagram of a DC motor including electrical characterstics.

The point of departure is the balancing of forces in the individual free body diagrams, assuming that each component is stationary or at constant velocity. In a DC motor the torque produced is proportional to the current drawn by the motor. This is illustrated in Equation D.1.

$$\tau_m = k_m i \tag{D.1}$$

The classic electrical model of a DC motor is used, which is a resistor, inductor and back electromotive force voltage V_{emf} connected in series. This voltage is approximated as being proportional to the motor angular velocity. This is shown in Equation D.2.

$$V_{emf} = k_e \omega \tag{D.2}$$

From here, Kirchoff's Voltage Law is used to develop a linear differential equation for the electrical circuit of the motor. Equation D.3 shows the resulting expression.

$$V_a - Ri - L\frac{di}{dt} - V_e = 0 \tag{D.3}$$

Taking the sum of the moments about the core of the motor including the friction in the motor shaft, approximated as a linear ratio to the angular velocity of the motor, Equation D.4 is produced.

$$T_m - k_f \omega - T_a = I_R \frac{d\omega}{dt} \tag{D.4}$$

With some substitution and manipulation of these expressions, the equation of motion for the system is produced as shown in Equation D.5.

$$\frac{d\omega}{dt} = \frac{k_m k_e}{I_R R}\omega + \frac{k_m}{I_R R}V_a - \frac{T_a}{I_R} \tag{D.5}$$

The state space form of the complete equations of motion governing th DC motor are shown in Equations D.6 and D.7

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-k_m k_e}{I_R R} \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k_m}{I_R R} & \frac{-1}{I_R R} \end{bmatrix} \begin{bmatrix} V_a \\ \tau_a \end{bmatrix} \tag{D.6}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} V_a \\ \tau_a \end{bmatrix} \tag{D.7}$$

The above equations describe the motion of the DC motor as a component of the system. The next components that are investigated are the individual wheels of the platform. The forces and moments involved with the derivation of the equations of motion of the system are shown in Figure D.2.

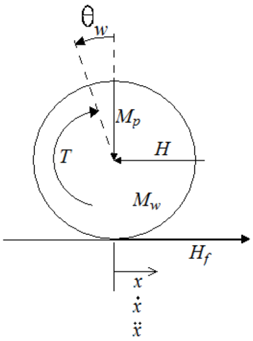


Figure D.2: Free body diagram of a platform wheel with external forces.

This figure depicts the general form of the free body diagram of one wheel from the platform. Where necessary subscripts are added in the equations to denote left or right wheel. The derivation that follows is applicable to the right wheel, however these expressions apply to the left as well.

Using the free body diagram of the wheel the sum of the forces in the horizontal direction is taken and equated to the inertia of the wheel. This is expressed mathematically in Equation D.8.

$$M_w\ddot{x} = H_{fR} - H_R \tag{D.8}$$

The sum of the moments about the centre of the wheel is also taken and equated to the polar moment of inertia of the wheel about its centre. This is expressed in Equation D.9.

$$I_w\ddot{\theta}_w = T_R - H_{fR}r \tag{D.9}$$

Using the expression for the applied torque to the DC motor the torque C_R can be calculated, followed by an expression for the horizontal force H_{fR} in terms of motor characteristics. From these an equation of motion for each wheel can be derived in terms of linear motion, by converting rotational variables to translational variables. These are shown in Equations D.10, for the left wheel, and D.11, for the right.

$$M_w\ddot{x} = \frac{-k_m k_e}{Rr^2}\dot{x} + \frac{k_m}{Rr}V_a - \frac{I_w}{r^2}\ddot{x} - H_L \tag{D.10}$$

$$M_w\ddot{x} = \frac{-k_m k_e}{Rr^2}\dot{x} + \frac{k_m}{Rr}V_a - \frac{I_w}{r^2}\ddot{x} - H_R \tag{D.11}$$

Adding these two expressions together yields an equation that will be used in the derivation of the complete equations of motion. This is shown in Equation D.12.

$$2\left(M_w + \frac{I_w}{r^2}\right)\ddot{x} = \frac{-2k_m k_e}{Rr^2}\dot{x} + \frac{2k_m}{Rr}V_a - (H_L + H_R) \tag{D.12}$$

The next component under investigation is the body of the platform. For this derivation the body has been approximated as a slender pole with centre of mass halfway along the length of the pole. The free body diagram is shown in Figure D.3.

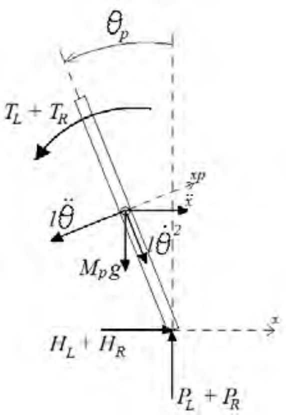


Figure D.3: Free body diagram of body of the platform with all external forces.

Once again for the free body diagram of the platform body the forces in the horizontal direction is taken and equated to the inertial effect of the mass of the body, as expressed in Equation D.13.

$$(H_L + H_R) - M_p l\ddot{\theta}_p \cos\theta_p + M_p l\dot{\theta}^2 \sin\theta_p = M_p\ddot{x} \tag{D.13}$$

In this case the forces in the direction perpendicular to the length of the body is also take in order to find an expression that incorporates the forces that act through both the centre of the body and the attachment point. This is shown in Equation D.14.

$$(H_L + H_R)\cos\theta_p + (P_L + P_R)\, sin\theta_p - M_p g \sin\theta_p - M_p l\ddot{\theta}_p = M_p\ddot{x}\cos\theta_p \tag{D.14}$$

The moments about the centre of the platform are summed and equated to the polar inertial effect of the mass of the pole. Equation D.15 expresses this.

$$-(H_L + H_R)\, l\cos\theta_p - (P_L + P_R)\, l\sin\theta_p - (T_L + T_R) = I_p\ddot{\theta}_p \tag{D.15}$$

With some substitution and manipulation of the above equations, including the equations for the DC motor, in order to eliminate the unknown forces Equations D.16 and D.17 are obtained.

$$\left(I_p + M_p l^2\right) \ddot{\theta}_p - \frac{2k_m k_e}{Rr}\dot{x} + \frac{2K_m}{R}V_a + M_p g l \sin\theta_p = -M_p l \ddot{x}\cos\theta_p \tag{D.16}$$

$$\frac{2k_m}{Rr}V_a = \left(2M_w + \frac{2I_w}{r^2} + M_p\right)\ddot{x} + \frac{2k_m k_e}{Rr^2}\dot{x} + M_p l \ddot{\theta}_p \cos\theta_p - M_p l \dot{\theta}_p^2 \sin\theta_p \tag{D.17}$$

However, these equations are non-linear. In order to be used in a linear control system design Equations D.16 and D.17 need to be linearised about the $\theta_p = 0$ point. It is also assumed that higher order terms are negligible, so $\cos\theta_p = -1$, $\sin\theta_p = -\theta$, and $\left(\frac{d\theta_p}{dt}\right)^2 = 0$

The linearised forms of the equations of motion for the complete system are shown in Equations D.18 and D.19.

$$\left(I_p + M_p l^2\right) \ddot{\theta} - \frac{2k_m k_e}{Rr}\dot{x} + \frac{2K_m}{R}V_a + M_p g l \sin\theta = M_p l \ddot{x} \tag{D.18}$$

$$\frac{2k_m}{Rr}V_a = \left(2M_w + \frac{2I_w}{r^2} + M_p\right)\ddot{x} + \frac{2k_m k_e}{Rr^2}\dot{x} + M_p l \ddot{\theta} \tag{D.19}$$

In order to represent the equations in a standardised format they have been transformed into state space format as seen in Equations D.20 with simplifying expressions: Equations D.21 and D.22.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{2k_m k_e\left(M_p l r - I_p - M_p l^2\right)}{Rr^2\alpha} & \frac{M_p^2 g l^2}{\alpha} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2k_m k_e (r\beta - M_p l)}{Rr^2\alpha} & \frac{M_p g l \beta}{\alpha} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2k_m\left(I_p + M_p l^2 - M_p l r\right)}{Rr\alpha} \\ 0 \\ \frac{2k_m (M_p l - r\beta)}{Rr\alpha} \end{bmatrix} V_a \tag{D.20}$$

$$\alpha = I_p\beta + 2M_p l^2\left(M_w + \frac{I_w}{r^2}\right) \tag{D.21}$$

$$\beta = 2M_w + \frac{2I_w}{r^2} + M_p \tag{D.22}$$

Some assumptions are relevant to the model as derived in Equations D.20, D.21, and D.22. These are that the wheels never break contact with the surface of the floor, zero slip occurs, and that cornering forces are negligible.

Nomenclature

The terms used in this derivation can be explained as follows listed in the order of appearance:

T_m	Torque delivered by the DC motor
k_m	DC motor torque constant
i	Current through drawn by the DC motor
V_{emf}	DC motor back electromotive force voltage
k_e	DC motor voltage-angular velocity constant
ω	DC motor angular velocity
V_a	DC motor applied voltage
R	Resistance over terminals of the DC motor
L	Induction of the DC motor
V_e	Voltage drop over armature of the DC motor
k_f	Approximate frictional force constant
T_a	Torque applied to the output shaft of the DC motor
I_R	Polar moment of inertia of the rotating component of the system
M_w	Mass of one wheel
x	Linear horizontal displacement of the wheel component
H_f	Frictional force between the wheel and the floor, applicable to left and right wheels
H	Force exerted on a wheel by the body of the platform, applicable to both wheels
I_w	Polar moment of inertia of a wheel
θ	Angular displacement of a wheel
T	Torque applied to one wheel
r	Radius of a wheel
M_w	Mass of a wheel
M_p	Mass of platform body
l	Length of pendulum approximation of the platform body
θ_p	tilt angle of platform body
P	Vertical reactive force due to platform body mass, divided into left and right
I_p	Polar moment of inertia of the platform body

D.2 Mathematical Simulation Code

The following pages show the code written for simulation of the system model for the active stability control simulation. The code was written for MatLab, and the file that was produced is a “.m” script file.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LQR_controlMD.m
% Simulation of Balancing Robot with LQR control
% This model includes the motor dynamics
% Author: Rich Chi Ooi
% Modified: Louw Butler
% 13.12.2008
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
clc

% Variable initialisation
g = 9.81;                     % Gravity (m/s^2)
r = 0.27;                     % Radius of wheel (m)
rp = 0.15;                    % Radius of pendulum mass (m)
Mw = 2;                       % Mass of wheel (kg)
Mp = 50;                      % Mass of body (kg)
Iw = (Mw*r^2)/2;              % Inertia of the wheel (kg*m^2)
l = 0.55;                     % Length to the body's centre of mass (m)
Ip = (2/5)*Mp*rp^2 + Mp*l^2;  % Inertia of the body (kg*m^2)

% Motor variables
Km = 0.106123;  % Motor torque constant (Nm/A)
Ke = 0.106087;  % Back EMF constant (Vs/rad)
R = 6;          % Nominal Terminal Resistance (Ohm)

% Va = Voltage applied to motor for controlling the pendulum

% System Matrices
% pre-calculated to simplify the matrix
% Denominator for the A and B matrices
beta = (2*Mw + (2*Iw/r^2) + Mp);
alpha = (Ip*beta + 2*Mp*l^2*(Mw + Iw/r^2));

A = [0         1                      0         0;
     0 (2*Km*Ke*(Mp*l*r-Ip-Mp*l^2))/(R*r^2*alpha) (Mp^2*g*l^2)/alpha 0;
     0        0                       0         1;
     0 (2*Km*Ke*(r*beta - Mp*l))/(R*r^2*alpha)    (Mp*g*l*beta)/alpha 0]

B=[       0;
    (2*Km*(Ip + Mp*l^2 - Mp*l*r))/(R*r*alpha);
          0;
    (2*Km*(Mp*l-r*beta)/(R*r*alpha))]

C = [1 0 0 0;
     0 0 1 0]

D = [0;
     0]

% Obtaining the eigenvalues of the system matrix
disp('The eigenvalues of the system matrix A')
disp('A positive value will indicate an unstable system')
p = eig(A)

%%%%%%%%%%%%%%%%%%%%
% LQR control design
%%%%%%%%%%%%%%%%%%%%

disp('Designing the optimal controller')
disp('Q = C''*C is a 4 x 4 weighting matrix for the outputs')
disp('Q could well be an Identity matrix with size same as system matrix A, as long as
it is positive definite')
disp('R is a 1 x 1 weighting matrix for the input')

% x is the weighting for the cart position
x = 100000;
% y is the weighting for the pendulum position
y = 100000000;

Q = [x 0 0 0;
     0 1 0 0;
     0 0 y 0;
     0 0 0 1];

R = 0.5;

%BRinverse = B*inv(R)*B';
%P = are(A,BRinverse,Q)

% Feedback Gain
disp('Feedback Gains for the system')
%K = inv(R)*B'*P
K = lqr(A,B,Q,R)
```

D.3 Simulation Block Diagram

Figure D.4 depicts the functional block diagram used to simulate the Linear Quadratic Regulator feedback control system in SimuLink.

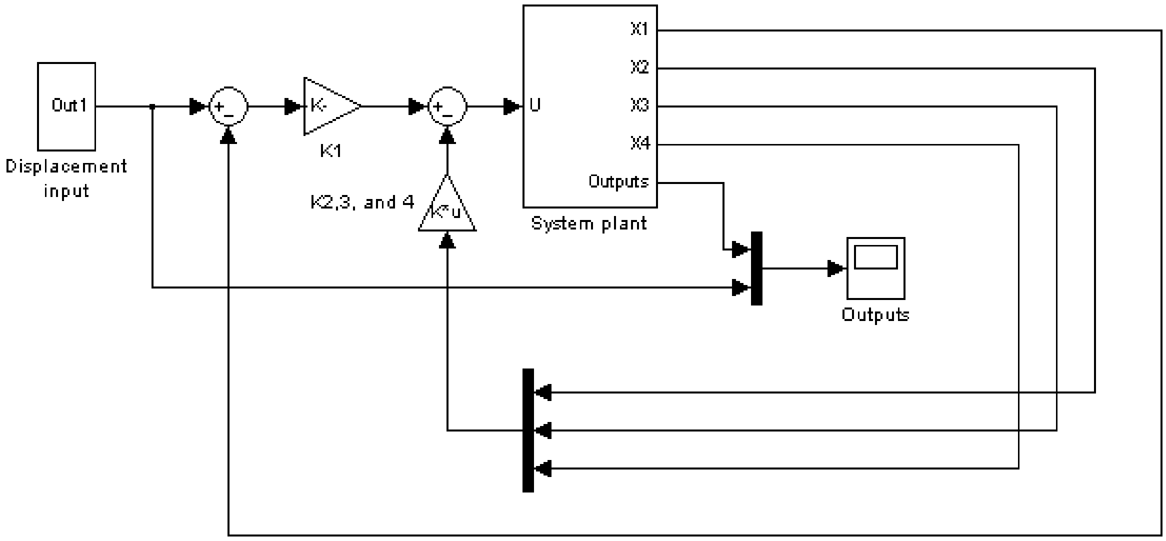


Figure D.4: Functional block diagram used to simulate the LQR control system.

In this figure the states X1, X2, X3, and X4 correspond to the linear displacement, x, linear velocity, $\dot{x}$, tilt angle, θ, and angular velocity, $\dot{\theta}$, respectively. The outputs are these states along with the reference input to the linear displacement. These outputs were ported to MatLab and plotted against time. These plots can be found on the Compact Disk accompanying this document.

Appendix E

Control System Programming Code

This chapter shows excerpts from the code developed for the feedback control system that is tasked with controlling the drive motors, in order to maintain active stability of the platform. The code that was developed for the complementary systems can be found on the Compact Disk accompanying this document. These include the control system that handles the activation sequence of the ultrasonic rangefinders as well as the extension and retraction of the static stability system. The programming of the third microcontroller includes the control of the payload interface system, this includes the collection of measurements from the load cells and the control of the motors that drive the rollers.

Two controllers were designed and tested for the active stability control and the results were compared. The first controller was designed using the Linear Quadratic Regulator design method and the second is a traditional Proportional-Integral-Derivative feedback control system. The code for the LQR controller is shown here. The complete collection of files used for all microcontroller programming can be found on the Compact Disk accompanying this document.

E.1 Active Stability Control System based on LQR Design Method

This section shows only the LQR designed control system code as written in the C programming language. Comments has been added as much as possible for clarity of reading.

```
/*****************************************************
Project      : Inverted Pendulum Robot
Version      : 1.0
Date created : 19/04/2010
Last revised : 28/05/2010
Author       : Louwrens J Butler
Company      : University of KwaZulu-Natal
Comments     :


Chip type           : ATmega32
Program type        : Application
Clock frequency     : 7.372800 MHz
*****************************************************/


/****** Include Files ******/
#include "global.h"        // include our global settings

#include <avr/io.h>        // include I/O definitions (port names, pin names, etc)
#include <util/delay.h>    // include delay functionality
#include <avr/interrupt.h> // include interrupt support
#include <math.h>          // include higher math functionality

#include "uart.h"          // include uart function library
#include "rprintf.h"       // include printf function library
#include "timer.h"         // include timer function library (timing, PWM, etc)
#include "a2d.h"           // include A/D converter function library
#include "encoder.h"       // include encoder function library
#include "vt100.h"         // include VT100 terminal support
#include "stdint.h"        // include support for shortened variable initialisation

// Function prototypes for user defined functions
//void timer0_Ovf_Func(void);
void initAll(void);
void pwmInit(void);
void stateUpdate(const float a_m, const float q_m);
void kalmanUpdate(const float Xtilt_m, const float Xrate_m);
void getLinear(void);
void balanceLQR(void);

/****** Global variable definitions ******/

// Define the real time step according to the timer flag variable
const float dt = 0.025;

// P[2,2] is the covariance matrix and is updated at every time step
// to determine how well the sensors are tracking the actual state
static float P[2][2] = {{1, 0},
                        {0, 1}};

// Set initial values for the normailsed angle and rate variables, globally
float normalXtilt = 0;
float normalXrate = 0;
// Set initial values for the final angle and rate variables, globally
float Xtilt = 0;
float Xrate = 0;

/*
 * R represents the measurement covariance noise.  In this case,
 * it is a 1x2 matrix that says that we expect 0.3 degree jitter
 * from the accelerometer and 0.3 degrees/s from the gyro.
 */
```

```
static const float XtiltR = 0.45;
static const float XrateR = 0.45;

/*
 * Q is a 2x2 diagonal matrix that represents the process covariance noise.
 * In this case, it indicates how much we trust the accelerometer
 * relative to the gyroscope.
 */
static const float XtiltQ = 0.001;
static const float XrateQ = 0.005;

// Global variables for use in multiple functions //
float angle_err = 0;
float rate_err = 0;
float prev_accel = 0;
float countsL = 0;
float countsR = 0;
float posCurrL = 0;
float posCurrR = 0;
float mpsCurrL = 0;
float mpsCurrR = 0;
u16 pwmL;
u16 pwmR;
static s32 count2L = 0;
static s32 count2R = 0;
u32 counter = 0;

/****** Main code ******/
int main(void)
{
  // Counter for Kalman filter settling loop
  u16 i = 0;

  // Initialise variables
  // Raw ADC readings
  u16 rawXtilt = a2dConvert10bit(0);
  u16 rawXrate = a2dConvert10bit(1);

  // Scaled readings
  const float XtiltRef = 511.0;        // 518
  const float XrateRef = 511.0;
  float scaledXtilt = XtiltRef - (float)rawXtilt;
  float XrateBias = (float)rawXrate - XrateRef;
  float scaledXrate = (float)rawXrate - (float)XrateBias;

  // Normalised data
  const float XtiltConv = 3.6;               // a2d / degree
  const float XrateConv = 100.0/1024.0;      // degrees/s / a2d
  normalXtilt = scaledXtilt/XtiltConv;
  normalXrate = (scaledXrate - (float)XrateRef)*XrateConv;

  // initialise libraries and setup IO ports
  initAll();

  // Make sure Kalman filter is stable
  while(i <= 400)
    {
      // Convert raw ADC values to usable state variables
      // Read ADC channel for x tilt variable
      rawXtilt = a2dConvert10bit(0);
      // Scale and normalise x tilt ADC value
      scaledXtilt = (XtiltRef - (float)rawXtilt);
      normalXtilt = scaledXtilt/XtiltConv;
      // Read ADC channel for x tilt rate variable
      rawXrate = a2dConvert10bit(1);
      // Scale and normalise x tilt rate ADC value
      XrateBias = (float)rawXrate - XrateRef;
      scaledXrate = (float)rawXrate - XrateBias;
      normalXrate = (XrateRef - scaledXrate)*XrateConv;

      // Retrieve the filtered inertial sensor measurements
      stateUpdate(normalXtilt, normalXrate);
      kalmanUpdate(normalXtilt, normalXrate);

      // Retrieve linear displacement and velocity states
      getLinear();

      balanceLQR();

      timerPause(25);
      i++;

      // Diagnostics/Data logging block
      if(i%2 == 0)
       {
         rprintfCRLF();
         rprintfFloat(4, Xtilt);
         rprintf(" ");
         rprintfFloat(4, Xrate);
         rprintf("      %d", counter);
       }
    }

  i = 0;

  while(1)
    {
      // Convert raw ADC values to usable state variables
      // Read ADC channel for x tilt variable
      rawXtilt = a2dConvert10bit(0);
      // Scale and normalise x tilt ADC value
      scaledXtilt = (XtiltRef - (float)rawXtilt);
      normalXtilt = scaledXtilt/XtiltConv;

      // Read ADC channel for x tilt rate variable
      rawXrate = a2dConvert10bit(1);
      // Scale and normalise x tilt rate ADC value
      scaledXrate = (float)rawXrate - XrateBias;
      normalXrate = (XrateRef - scaledXrate)*XrateConv;

      // Retrieve the filtered inertial sensor measurements
      stateUpdate(normalXtilt, normalXrate);
      kalmanUpdate(normalXtilt, normalXrate);

      // Retrieve linear displacement and velocity states
      getLinear();

      balanceLQR();

      timerPause(25);
      i++;

      // Diagnostics/Data logging block
      if(i%2 == 0)
       {
         rprintfCRLF();
         rprintfFloat(4, Xtilt);
         rprintf(" ");
         rprintfFloat(4, Xrate);
         rprintf(" ");
         rprintfFloat(3, posCurrL);
         rprintf(" ");
         rprintfFloat(3, posCurrR);
         rprintf(" ");
         rprintfFloat(3, mpsCurrL);
         rprintf(" ");
         rprintfFloat(3, mpsCurrR);
       }
    }

  return 0;
}
/****** End of main code ******/


/****** User defined functions *******/

// Initialisation sequence
void initAll(void)
{
  // configure PORTA as input for A-to-D
  DDRA = 0x00;
  // make sure pull-up resistors are turned off
  PORTA = 0x00;

  // Port D initialisation for PWM: PORTD4 & 5 are outputs
  DDRD = 0x30;
  // Make sure all outputs are set to zero at initialisation
  //  PORTD = 0x00;

  // initialise the UART (serial port)
  uartInit();
  // make all rprintf statements use uart for output
  rprintfInit(uartSendByte);

  // initialise the timer system
  timerInit();
  //  timerAttach(TIMER0OVERFLOW_INT, timer0_Ovf_Func);

  // initialise the encoder library
  encoderInit();
  // Set position of encoders to zero at initialisation
  encoderSetPosition(0,0);
  encoderSetPosition(1,0);

  // initialise user defined PWM functionality
  pwmInit();

  // turn on and initialise ADC
  a2dInit();

  // clear the console screen
  vt100ClearScreen();

  // set the a2d prescaler (clock division ratio)
  a2dSetPrescaler(ADC_PRESCALE_DIV128);

  // set the a2d reference
  a2dSetReference(ADC_REFERENCE_AVCC);

  return;
}

// Initialise the PWM functionality
void pwmInit(void)
{
  timer1PWMInitICR(9216);
  timer1PWMAOn();
  pwmL = 1385;
  timer1PWMASet(pwmL);
  timer1PWMBOn();
  pwmR = 1385;
  timer1PWMBSet(pwmR);
}

/*
 * stateUpdate is called every dt with a biased gyro measurement
 * by the user of the module.  It updates the current angle and
 * rate estimate.
 *
 * Due to the small CPU available on the microcontroller, we've
 * hand optimized the C code to only compute the terms that are
 * explicitly non-zero, as well as expanded out the matrix math
 * to be done in as few steps as possible.  This does make it harder
 * to read, debug and extend, but also allows us to do this with
 * very little CPU time.
 */
void stateUpdate(
              const float accel,      /* Accelerometer measurement */
              const float gyro    /* Gyro measurement */
)
{
  /*
   * Compute the derivative of the covariance matrix
   *
   *   Pdot = A*P + P*A' + Q
   *
   * We've hand computed the expansion of A = [ 0 1, 0 0 ] multiplied
   * by P and P multiplied by A' = [ 0 0, 1 0 ].  This is then added
   * to the diagonal elements of Q, which are XtiltQ and XrateQ.
   */
  const float       Pdot[2 * 2] = {
    XtiltQ + P[0][1] + P[1][0],  /* 0,0 */
                       P[1][1],         /* 0,1 */
                    P[1][1],            /* 1,0 */
                       XrateQ                        /* 1,1 */
  };

  /* Store our gyro estimate */
  Xrate = (prev_accel - accel)*dt;
  /* Store the previous accelerometer reading */
  prev_accel = accel;

  /*
   * Update our angle estimate
   * angle += angle_dot * dt
   *       += q * dt
   */
  Xtilt += gyro * dt;

  /* Update the covariance matrix */
  P[0][0] += Pdot[0] * dt;
  P[0][1] += Pdot[1] * dt;
  P[1][0] += Pdot[2] * dt;
  P[1][1] += Pdot[3] * dt;
}



/*
 * kalmanUpdate is called as soon as new state estimates are available,
 * i.e. once stateUpdate is called.
 *
 * As commented in stateUpdate, the math here is simplified to
 * make it possible to execute on a small microcontroller with no
 * floating point unit.  It will be hard to read the actual code and
 * see what is happening, which is why there is this extensive
 * comment block.
 *
 * The C matrix is a 1x2 (measurements x states) matrix that
 * is the Jacobian matrix of the measurement value with respect
 * to the states.  In this case, C is:
 *
 *     C = [ d(Xtilt_m)/d(Xtilt)  d(Xrate_m)/d(Xrate) ]
 *       = [ 1 1 ]
 */
void kalmanUpdate(
               const float Xtilt_m,       /* X tilt state */
               const float Xrate_m        /* X rate state */
)
{
  /* Compute our measured angle and the error in our estimate */
  angle_err = Xtilt_m - Xtilt;
  rate_err = Xrate_m - Xrate;

  /*
   * C shows how the state measurement directly relates to
   * the state estimate.
   */
  const float C_0 = 1;
  const float C_1 = 1;

  /*
   * PCt<2,1> = P<2,2> * C'<2,1>, which we use twice.  This makes
   * it worthwhile to precompute and store the two values.
   */
  const float PCt_0 = C_0 * P[0][0] + C_1 * P[0][1];
  const float PCt_1 = C_0 * P[1][0] + C_1 * P[1][1];

  /*
   * Compute the error estimate.  From the Kalman filter paper:
   */
  const float       E_0 =
    XtiltR
           + C_0 * PCt_0
           + C_1 * PCt_1
    ;

  const float             E_1 =
    XrateR
           + C_0 * PCt_0
           + C_1 * PCt_1
    ;
  /*
   * Compute the Kalman filter gains.  From the Kalman paper:
   *
   *   K = P C' inv(E)
   */
  const float K_0 = PCt_0 / E_0;
  const float K_1 = PCt_1 / E_1;

  /*
   * Update covariance matrix.  Again, from the Kalman filter paper:
   *
   *   P = P - K C P
   *
   * We first compute t<1,2> = C P.  Note that:
   *
   *   t[0,0] = C[0,0] * P[0,0] + C[0,1] * P[1,0]
   *      t[0,1] = C[0,0] * P[0,1] + C[0,1] * P[1,1]
   */
  const float t_0 = C_0 * P[0][0] + C_1 * P[1][0];
  const float t_1 = C_0 * P[0][1] + C_1 * P[1][1];

  P[0][0] -= K_0 * t_0;
  P[0][1] -= K_0 * t_1;
  P[1][0] -= K_1 * t_0;
  P[1][1] -= K_1 * t_1;

  /*
   * Update our state estimate.  Again, from the Kalman paper:
   *
   *   X += K * err
   *
   * err is a measurement of the difference in the measured state
   * and the estimate state.  In our case, it is just the difference
   * between the two accelerometer measured angle and our estimated
   * angle.
   */
  Xtilt += K_0 * angle_err;
  Xrate += K_1 * rate_err;
}

// Function for calculating the wheel velocity from encoder measurements
void getLinear(void)
{

  // Initialisation general of variables
  float cpm = 636.61977;   // Counts per metre conversion factor

  // Left wheel
  s32 count1L;
  float cpsCurrL;
  s32 deltaCountsL = 0;
  // Right wheel
  s32 count1R;
  float cpsCurrR;
  s32 deltaCountsR = 0;

  // Take encoder readings
  // Left wheel
  count1L = count2L;
  count2L = encoderGetPosition(0);
  // Right wheel
  count1R = count2R;
  count2R = encoderGetPosition(1);
  // Make sure that the encoder counts stay within the limits
  // Left wheel
  if((count2L >= INT16_MAX) | (count2L <= -INT16_MAX))
    {
      encoderSetPosition(0,0);
      count2L = encoderGetPosition(0);
    }
  // Right wheel
  if((count2R >= INT16_MAX) | (count2R <= -INT16_MAX))
    {
      encoderSetPosition(1,0);
      count2R = encoderGetPosition(1);
    }

  // Calculate current counts/sec
  // Left wheel
  deltaCountsL = (count2L - count1L);
  cpsCurrL = (float)deltaCountsL / dt;
  // Right wheel
  deltaCountsR = -(count2R - count1R);
  cpsCurrR = (float)deltaCountsR / dt;

  // Calculate the current m/s values
  // Left wheel
  mpsCurrL = cpsCurrL / cpm;
  // Right wheel
  mpsCurrR = cpsCurrR / cpm;

  // Calculate the linear displacement at each wheel
  // Left wheel
  countsL += deltaCountsL;
  posCurrL = countsL / cpm;
  // Right wheel
  countsR += deltaCountsR;
  posCurrR = countsR / cpm;

  return;
}

// Function for achieving LQR designed feedback control
void balanceLQR(void)
{
  float K1 = 35324;
  float K2 = 8174;
  float K3 = -158;
  float K4 = -598;
  float lqrTheta;
  float lqrdTheta;
  static float theta = 0;
  static float dtheta = 0;
  static float lqrL;
  static float lqrR;

  // Convert degrees to radians and degrees/sec to radians/sec
  theta = Xtilt / 57.2958;
  dtheta = Xrate / 57.2958;
  // Calculate common feedback terms for tilt and angular velocity
  lqrTheta = K1*theta;
  lqrdTheta = K2*dtheta;

  // Calculate the Linear Quadratic Regulator feedback term for each motor
  lqrL = (lqrTheta + lqrdTheta + K3*posCurrL + K4*mpsCurrL) / 24;
  lqrR = (lqrTheta + lqrdTheta + K3*posCurrR + K4*mpsCurrR) / 24;

  // Make sure the Kalman filter is stabilised
  if(counter > 400)
    {
      // These conditionals take care of direction of rotation
      // For the left wheel
      if(lqrL > 0)
       pwmL = 1335 - lqrL;
      else if(lqrL < 0)
       pwmL = 1435 - lqrL;
      else pwmL = 1385;
      // For the right wheel
      if(lqrR > 0)
       pwmR = 1435 + lqrR;
      else if(lqrR < 0)
       pwmR = 1335 + lqrR;
      else pwmR = 1385;

      // Implement pwm veriable in output function
      timer1PWMASet(pwmL);
      timer1PWMBSet(pwmR);
    }

  counter++;

  return;
}
```

References

[1] M.F.B. Abdollah. Proportional integral sliding mode control of a two-wheeled balancing robot. Master's thesis, Malaysia University of Technology, Faculty of Electrical Engineering, 2006. 11

[2] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand. An architecture for autonomy. *International Journal of Robotics Research*, 17(4):315–337, 1998. 9

[3] A. Balestrino, A.M. Casalino, and A. Bicchi. Closed loop steering of unicycle-like vehicle via lyapunov techniques. *IEEE Robotics & Automation Magazine*, 2:27–35, 1995. 94

[4] P.P. Benham, R.J. Crawford, and C.G. Armstrong. *Mechanics of Engineering Materials*. Pearson Prentice Hall, Harlow, England, second edition, 1996. 147

[5] Trevor Blackwell. Building a balancing scooter [online]. `http://tlb.org/scooter.html`. [Accessed: 18 September 2009]. viii, 11, 51

[6] B.S. Blanchard and W.J. Fabrycky. *Systems Engineering and Analysis*. Pearson Prentice Hall, Upper Saddle River, New Jersey, fourth edition, 2006. 37, 65

[7] S.A. Boctor, P.F. Ryff, P.D. Hiscocks, M.T. Ghorab, and M.R. Holmes. *Electrical Concepts and Applications*. West Publishing Company, St. Paul, Minneapolis, 1997. 56

[8] W. Bolton. *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering*. Pearson Education Limited, London, United Kingdom, third edition, 2003. 5, 58, 59

[9] G. Bright and A. Walker. A mobile mechatronic platform architecture for flexible materials handling. In *Proceedings of the Australasian Conference on Robotics and Automation*, Brisbane, Australia, December 2007. 10

[10] A.E. Bryson and Y.C. Ho. *Applied Optimal Control: Optimization, Estimation, and Control*. Blaisdell Publishing Co., Waltham, MA, 1969. 80

[11] R.S. Burns. *Advanced Control Engineering*. Butterworth-Heinemann, Oxford, 2001. 87

[12] A. Cardi and M. Wagner. Control of an rc segway [online]. www.prism.gatech.edu/~gth856d/buffalo/paper.pdf. [Accessed: 18 September 2009]. 11

[13] T. Collett, G. MacDonald, and B. Gerkey. Player 2.0. toward a practical robot programming framework. In *Proceedings of the Australasian Conference on Robotics and Automation*, Sydney, Australia, 2005. 94

[14] B.P. Crow, I. Widjaja, L.G. Kim, and P.T. Mitre Corp. ieee 802.11 wireless local area network. *Communications Magazine, IEEE*, 36(9):116–126, 1997. 57

[15] F. Dalleart, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, USA, 1999. 96

[16] S. Davis. *Future Perfect*. Basic Books, 1997. 2

[17] Analog Devices. Adxl204: Precision 1.7g single/dual-axis imems accelerometer [online]. http://www.analog.com/en/sensors/inertial-sensors/adxl204/products/product.html, . [Accessed: 18 October 2009]. 50, 51

[18] Analog Devices. Adxrs614: 50 degree per sec yaw rate gyro [online]. http://www.analog.com/en/mems/gyroscopes/adxrs614/products/product.html, . [Accessed: 18 October 2009]. 54

[19] R.S. Figliola and D.E. Beasley. *Theory and Design for Mechanical Measurement*. John Wiley & Sons, Inc., New Jersey, third edition, 2000. 70

[20] G.F. Franklin, J.D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Prentice Hall, New Jersey, fourth edition, 2002. 78, 81, 85

[21] Strategos Inc. Group technology: Group technology simplifies lean manufacturing [online]. http://www.strategosinc.com/group_technology.htm. [Accessed: 25 September 2009]. 13

[22] D.J. Inman. *Engineering Vibration*. Prentice Hall International, Inc., New Jersey, second edition, 2001. 34

[23] Galileo Mobility Instruments. Galileo mobility: Wheels when possible tracks when needed [online]. http://www.galileomobility.com/. [Accessed: 06 October 2009]. 26

[24] J. Jiao and M. Tseng. Fundamentals of product family architecture. *Integrated Manufacturing Systems*, 11(7):469–483, 2000. 3, 14

[25] J. Jiao and M.M. Tseng. A methodology of developing product family architecture for mass customization. *Journal of Intelligent Manufacturing*, 10:3–20, 1999. 13

[26] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 85:35–45, 1960. 70

[27] S. Kara and B. Kayis. Manufacturing flexibility and variability: an overview. *Journal of Manufacturing Technology Management*, 15(6):466–487, 2004. 9

[28] Ted Larson. Balancing robot project – "bender" [online]. http://www.tedlarson.com/robots/balancingbot.htm. [Accessed: 18 September 2009]. viii, 11, 51

[29] S. LaValle. *Planning Algorithms*. Cambridge University Press, Illinois, 2006. 9, 93, 96

[30] Segway LLC. Segway: Simply moving [online]. http://www.segway.com, . [Accessed: 18 September 2009]. 11

[31] Segway LLC. Segway robotic mobility platform (rmp) website [online]. http://www.segway.com/business/products-solutions/robotic-mobility-platform.php, . [Accessed: 18 September 2009]. viii, 12

[32] M.G. Mehrabi, A.G. Ulsoy, Y. Koren, and P. Heytler. Trends and perspectives on flexible and reconfigurable manufacturing systems. *Journal of Intelligent Manufacturing*, 13(2):135–146, 2002. 3, 16

[33] J.L. Meriam and L.G. Kraige. *Engineering Mechanics: Dynamics*. John Wiley & Sons, Inc., New Jersey, fifth edition, 2003. 24, 55, 84

[34] P. Moore, S. Ujvari, J. Pu, J. Lundgren, and C. Xie. Intelligent semi-autonomous vehicles in materials handling. *Mechatronics*, 9:881–892, 1999. 8

[35] H.G. Nguyen, J. Morrell, K. Mullens, A. Burmeister, S. Miles, N. Farrington, K. Thomas, and D.W. Gage. Segway robotic mobility platform. In *SPIE Proc. 5609: Mobile Robots XVII*, Philadelphia, Pensylvania, 2004. 12

[36] R.C. Ooi. Balancing a two-wheeled autonomous robot [online]. robotics.ee.uwa.edu.au/theses/2003-Balance-Ooi.pdf, 2003. [Accessed: 18 September 2009]. 11, 51, 152

[37] R.M.E. Sabbatini PhD. Imitation of life: A history of the first robots [online]. http://www.cerebromente.org.br/n09/historia/turtles_i.htm. [Accessed: 17 September 2009]. 8

[38] Frank Piller. Frank piller's web site on mass customization and open innovation: Glossary [online]. http://www.mass-customization.de/. [Accessed: 11 September 2009]. 2

[39] Acroname Robotics. Devantech srf02 sensor [online]. http://www.acroname.com/robotics/parts/R287-SRF02.html, . [Accessed: 18 October 2009]. 47

[40] IFI Robotics. Ifi robotics 24v victor 883/885 [online]. http://www.ifirobotics.com/victor-885-speed-controller-robots.shtml, . [Accessed: 22 October 2009]. ix, 59

[41] R. Siegwart and I.R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. MIT Press, Cambridge, Massachusetts, 2004. 8

[42] E.D. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Springer-Verlag New York, Inc., New York, second edition, 1998. 78

[43] Open Source. The player project: Free software tools for robot and sensor applications [online]. http://playerstage.sourceforge.net. [Accessed: 23 April 2010]. 94, 95

[44] Virtual Logic Systems. Virtual logic systems – product engineering [online]. http://www.virtuallogicsys.com/services_product.htm. [Accessed: 30 October 2009]. viii, 5

[45] S. Thrun, W. Burgard, and D Fox. *Probabilistic Robotics*. MIT Press, Cambridge, Massachusetts, 2005. 9

[46] A. Toffler. *Future Shock*. Bantam Books, New York, 1971. 2

[47] I. Ulrich and J. Borenstein. Vfh+: Reliable obstacle avoidance for fast robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 1998. 97

[48] A.J. Walker. Autonomous mobile materials handling platform architecture for mass customisation. Master's thesis, University of KwaZulu-Natal, School of Mechanical Engineering, 2008. 94

[49] G. Welch and G. Bishop. An introduction to the kalman filter [online]. `http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html`, 2004. [Accessed: 13 April 2010]. 73