Evaluation of a Mobile Platform for Proof-of-Concept Autonomous Site Selection and Preparation

by

Jonathan Gammell

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of the Institute for Aerospace Studies
University of Toronto

Abstract

Evaluation of a Mobile Platform for Proof-of-Concept Autonomous Site Selection and Preparation

Jonathan Gammell
Master of Applied Science
Graduate Department of the Institute for Aerospace Studies
University of Toronto
2010

A mobile robotic platform for Autonomous Site Selection and Preparation (ASSP) was developed for an analogue deployment to Mauna Kea, Hawai'i. A team of rovers performed an autonomous Ground Penetrating Radar (GPR) survey and constructed a level landing pad. They used interchangeable payloads that allowed the GPR and blade to be easily exchanged. Autonomy was accomplished by integrating the individual hardware devices with software based on the ArgoSoft framework previously developed at UTIAS. The rovers were controlled by an on-board netbook. The successes and failures of the devices and software modules are evaluated within. Recommendations are presented to address problems discovered during the deployment and to guide future research on the platform.

Acknowledgements

I would like to thank my parents, brother and sister for their untiring love and support throughout my life. I would not be here without them.

This work would not have been possible without the direction of my supervisor, Prof. Gabriele D'Eleuterio. I will never forget the trust he showed in me throughout the ISRU project and I would like to express my sincerest gratitude for this and all of his guidance. Hopefully he will find this report useful for future work on the project. Also, Gabe, the Rams will win again soon.

The ISRU project depended on the tireless dedication of Dr. Ernest Earon. Without his constant advice, direction and motivationally unrealistic Gantt charts, the demo may well have been a summer student with a shovel. I, and everyone who worked on the ISRU project, owe him a great debt. It is my opinion that, perhaps, the Vatican made a mistake.

I would like to thank everyone from the project; Ken, Francis, Terence, Paul and Alex. It was a great pleasure to work with all of you and I hope you recognize how impressive your contributions truly were. I would also like to thank our ISRU summer students, Heather and Tommy, for all their work.

Everyone else in the lab, Adam, Pete, Dave and our other summer students Vidya and Ali, were immensely supportive and understanding throughout the project. Even when they were not asked to contribute, they constantly offered timely help and advice. The project would not have been the same without their constant companionship and support.

Finally, special thanks go to everyone who helped me edit the various versions of this document. Peter Szabo was willing to read the initial draft before I ever could, demonstrating a level of bravery rarely necessary in academics. Prof. Chris Damaren, Prof. D'Eleuterio and Dr. Earon then provided excellent advice without which this report would have been unintelligible. It should be clear, that any remaining mistakes are solely my fault.

Contents

1 Introduction **1**

1.1 Demand for In-Situ Resource Utilization (ISRU) . . . 1

1.2 Demand for Autonomous Site Selection and Preparation (ASSP) . . . 2

1.3 The In-Situ Resource Utilization (ISRU) Project . . . 2

1.4 The Musketeer Rovers . . . 4

1.5 Purpose of This Report . . . 4

2 Base Rover **6**

2.1 The Juno Mobility Chassis . . . 6

2.2 Locomotion System . . . 10

3 Payloads **17**

3.1 Blade Payload . . . 17

3.2 Ground Penetrating Radar (GPR) Payload . . . 22

3.3 Supervisory Control and Data Acquisition (SCADA) Box . . . 25

4 Autonomous Control System **31**

4.1 Sensors and Actuators . . . 31

4.2 Power . . . 32

4.3 Device Interfaces . . . 32

4.4 Internal Rover Network . . . 36

4.5 Software . . . 36

5 Autonomous Control System Devices **40**

5.1 RoboteQ AX2850 Brushed DC Motor Controller . . . 40

5.2 Omega OMB-DAQ-3001 Data Acquisition Module . . . 46

5.3 Sensors & Software OEM Noggin Network Interface Card (NIC) Ground Penetrating Radar (GPR) . . . 48

5.4 Musketeer Interface Box (MIB) . . . 49

5.5 Cisco SD208 Ethernet Switch . . . 52
5.6 Perle IOLAN SDS4 Device Server . . . 53
5.7 Lantronix UBox 2100 . . . 54
5.8 USB Serial Converter . . . 56
5.9 Musketeer Control Box (MCB) . . . 57
5.10 ASUS S101 Eee PC . . . 60
5.11 Hawking Wireless System . . . 62
5.12 Belkin 7-Port Mobile USB Hub . . . 64
5.13 Logitech Dual Action Gamepad . . . 65
5.14 MicroStrain 3DM-GX2 Attitude Heading Reference System (AHRS) . . . 65
5.15 u-blox EVK-5H Global Positioning System (GPS) Receiver . . . 67
5.16 Trimble Global Positioning System (GPS) Pathfinder ProXRT . . . 70
5.17 QualityKits VK011 Serial Temperature Sensor . . . 75
5.18 DirectedPerception PTU-D47 Pan-Tilt Unit (PTU) . . . 77
5.19 SICK LMS111 Laser Measurement System . . . 78
5.20 Microsoft LifeCam VX-3000 Cameras . . . 80

6 Autonomous Control System Software Modules 82

6.1 *Configuration Manager Server (CMS)* . . . 82
6.2 *Musketeer Control Server (MCS)* . . . 83
6.3 *Kalman Filter Server (KFS)* . . . 88
6.4 *Calibration and Testing (CaT)* . . . 90
6.5 *GPRSurvey* . . . 91
6.6 *C1*: Artificial Neural Tissue (ANT) Controller . . . 94
6.7 *C2*: Alternative Neural Network (ANN) Controller . . . 95
6.8 *C3*: Model-Based Control . . . 98
6.9 *ThermalMon* . . . 102
6.10 *WinGCS* . . . 103
6.11 *Logger* . . . 104
6.12 *HealthMon* . . . 106
6.13 *CameraMon* . . . 107

7 Recommendations 108

7.1 Areas Requiring Immediate Attention . . . 108
7.2 Areas for Rapid Advancement . . . 115

8 Conclusions 118

Bibliography **120**

List of Tables

2.1 Rover work times for February 10th, 2010. 14
2.2 Rover work times presented as a percentage of operational time. 15

4.1 Power requirements of the autonomous control system devices. 32
4.2 Interfaces of the autonomous control system devices. 36

6.1 Logging solutions organized by server. 106

7.1 Autonomous control system devices requiring Microsoft Windows. 117

8.1 Summary of autonomous control system device evaluations. 119

List of Figures

1.1 Projected regolith viewed from the Apollo 12 lunar module. 2
1.2 Operational flow of Autonomous Site Selection and Preparation (ASSP). 3
1.3 Diagram of target landing pad topology. 4

2.1 Base chassis 7
2.2 Rear view of a rover showing the battery compartments partially-obstructed by the walking arm suspension. 8
2.3 Location of the rover motors between fore and aft wheels. 9
2.4 Rover motor temperatures on February 10th, 2010 15

3.1 Blade payload 18
3.2 Actuator control board 19
3.3 Failed actuator control board 20
3.4 Horizontal load cell measurements and lift position during a typical excavation pass. 21
3.5 Example of an excavation pass with unacceptable oscillations. 21
3.6 GPR Payload 23
3.7 Simplified electrical schematic of the SCADA Box installation. 26
3.8 Minimum recommended filters for Vicor Micro DC-DC converters. 27
3.9 Output of the 12 V Vicor Micro DC-DC converter installed in a SCADA Box. . . 28

4.1 Power topology of a Musketeer rover. 33
4.2 Power topology of the autonomous control system. 34
4.3 Interface topology of the autonomous control system. 35
4.4 Network topology of the autonomous control system. 37
4.5 Software topology of the autonomous control system. 38

5.1 Simplified electrical schematic of the recommended RoboteQ AX2850 installation. 44

5.2 Simplified electrical schematic of the RoboteQ AX2850 installation in the Musketeer rovers. 45

6.1 *Athos* testing the digging controller during the Mauna Kea deployment. 84
6.2 Diagram of the driving controller. 85
6.3 GPR survey during the Mauna Kea deployment. 92
6.4 Landing pad berm constructed during the Mauna Kea deployment. 99
6.5 Landing pad constructed during the Mauna Kea deployment. 101

7.1 Simplified electrical schematic of a proposed solution to the problems experienced with the RoboteQ AX2850. 111
7.2 Alternative interface topology removing the Perle IOLAN SDS4 and the Lantronix UBox 2100. 114

List of Acronyms

ACPI Advanced Configuration and Power Interface

AHRS Attitude Heading Reference System

ANN Alternative Neural Network

ANT Artificial Neural Tissue

ASSP Autonomous Site Selection and Preparation

CaT Calibration and Testing

CLI Command-Line Interface

CMS Configuration Manager Server

CSA Canadian Space Agency

DAQ Data Acquisition

DPST Double-Pole, Single-Throw

DoF Degree-of-Freedom

EUI External User Interface

GCS Ground Control Station

GPR Ground Penetrating Radar

GPS Global Positioning System

GUI Graphical User Interface

HDD Hard Disk Drive

HID Human Interface Device

KFS Kalman Filter Server

ICD ISRU GPR Server Interface Control Document

IMU Inertial Measurement System

ISRU In-Situ Resource Utilization

LAN	Local Area Network
MCS	Musketeer Control Server
MCB	Musketeer Control Box
MFC	Microsoft Foundation Class
MIB	Musketeer Interface Box
NIC	Network Interface Card
NASA	National Aeronautics and Space Administration
NORCAT	Northern Centre for Advanced Technology Inc.
ODG	Ontario Drive and Gear Ltd.
HDOP	Horizontal Dilution of Position
PTU	Pan-Tilt Unit
R/C	Radio Control
SCADA	Supervisory Control and Data Acquisition
SD	Secure Digital
SDK	Software Development Kit
SPST	Single-Pole, Single-Throw
SPDT	Single-Pole, Double-Throw
SSD	Solid-State Drive
UKF	Unscented Kalman Filter
UTIAS	University of Toronto Institute for Aerospace Studies
UTM	Universal Transverse Mercator
WAAS	Wide Area Augmentation System
UVC	USB Video Class

Chapter 1

Introduction

1.1 Demand for In-Situ Resource Utilization (ISRU)

A permanent Moon base has been proposed as a milestone for space exploration beyond Earth's orbit [1]. When compared to other options, its proximity to Earth and its weak gravity reduce both the work required to support it and the energy spent leaving it for distant destinations. The National Aeronautics and Space Administration's (NASA) previous experience with manned Moon missions during the Apollo program only further increase its attractiveness.

The high cost and complexity of launching supplies from Earth places a strong emphasis on producing as many life-necessities on the Moon as possible. The lunar regolith, while being devoid of any organic matter, has been shown to be high in ilmenite which through a process known as ROxygen can yield oxygen to be used to support outposts [2]. The use of available resources in extraterrestrial missions has been dubbed In-Situ Resource Utilization (ISRU).

ISRU, much like its terrestrial cousins construction and mining, will be a labour intensive process. Given the limited manpower available during space missions and the difficulty of performing manual labour on hostile extraterrestrial bodies, it is unrealistic to expect the work to be carried out directly by astronauts. Ideally a large amount of the ISRU infrastructure would precede the astronauts arrival, providing them with a safe-haven in an otherwise inhospitable environment. This could be accomplished with a team of autonomous robotic rovers who would perform general site preparation in addition to ISRU activities in advance of the astronauts' arrival.

1.2 Demand for Autonomous Site Selection and Preparation (ASSP)

The unique characteristics of the Moon surface make the construction of a permanent base challenging. Lacking a protective atmosphere, the Moon is subjected to frequent meteor strikes that leave craters and cover the surface in loose regolith. Coupled with low gravity, its lack of atmosphere allows the abrasive regolith to be projected by rocket boosters and damage surrounding equipment (Figure 1.1). The techniques of ISRU can be used address these challenges. Given candidate locations, autonomous rovers can verify their suitability and prepare them for use. The loose regolith can be removed to create flat and level landing pads from harder-packed lower layers, reducing the amount of free regolith available to be projected. Landing pads can be surrounded by protective berms to block the energetic debris from travelling along the Moon surface. Landing pads located away from the base itself can be connected via a network of roads. This has been dubbed Autonomous Site Selection and Preparation (ASSP).

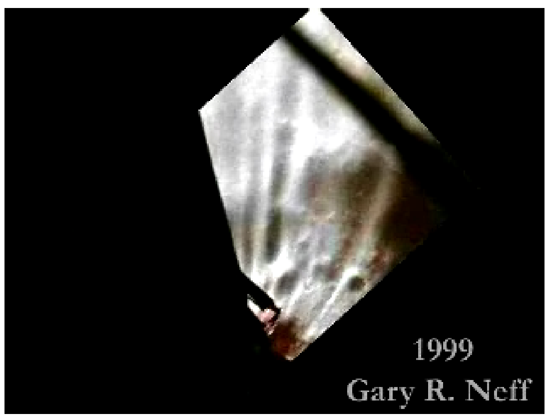


Figure 1.1: Projected regolith viewed from the Apollo 12 lunar module [3].

1.3 The In-Situ Resource Utilization (ISRU) Project

The ISRU project was designed to apply Canada's unique technical expertise and infrastructure to the challenges of ISRU and ASSP. A collaborative project, it leveraged the expertise of the Canadian Space Agency (CSA), the Northern Centre for Advanced Technology Inc. (NORCAT), Ontario Drive and Gear Ltd. (ODG), Neptec Design Group Ltd., Xiphos Technologies Inc., and the Space Robotics Group at the University of Toronto Institute for Aerospace Studies (UTIAS) to demonstrate a proof-of-concept ISRU platform for an ISRU analogue mission to Mauna Kea, Hawai'i from January 24^{th} to February 13^{th}, 2010. The mission was to include both traditional ISRU tasks, such as ROxygen [2] and ASSP tasks such as site verification through subsurface scans and drill samples, road and landing pad construction and landing pad preparation through regolith sintering. The analogue mission was designed to test the applicability of these techniques for a hypothetical lunar mission. This required using extraterrestrial-applicable solution techniques wherever reasonable.

UTIAS was involved in the development of hardware and software tools to perform subsurface scans and road and landing pad construction. Being commanded only via high-level remote commands, a potential area for a landing pad was to be selected, verified by subsurface Ground Penetrating Radar (GPR) scans and then, pending remote confirmation, prepared by a team of three rovers dubbed *The Musketeers* (Figure 1.2). A flat and level landing pad surrounded by a protective berm and accessed by a road would be constructed on the specified site (Figure 1.3). The topography of the working area was provided by Neptec via the TriDAR. The TriDAR provides the rovers with three-dimensional scans of the target work area. The GPR data collected during the subsurface scans was postprocessed and analyzed by Xiphos.

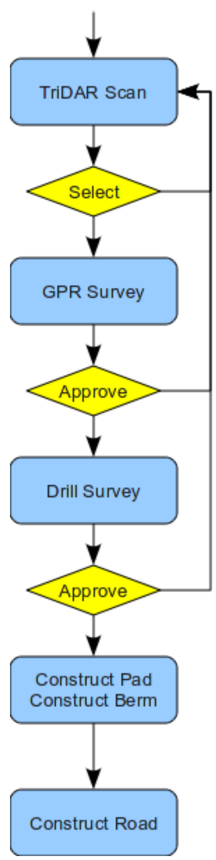


Figure 1.2: Operational flow of Autonomous Site Selection and Preparation (ASSP).

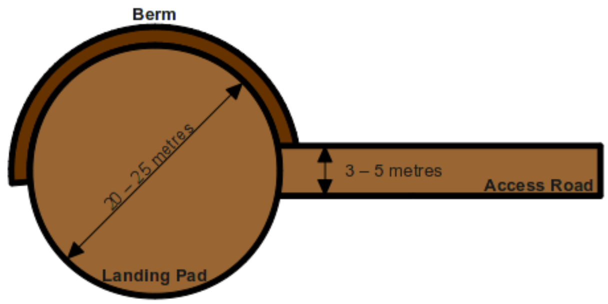


Figure 1.3: Diagram of target landing pad topology.

1.4 The Musketeer Rovers

The ISRU project was developed around the Juno mobility chassis designed by Neptec and ODG and manufactured by ODG (§2.1). Equipped with a locomotion system designed and installed by NORCAT (§2.2) and supporting interchangeable payloads (§3), configurations of the Juno chassis were used for all the ISRU and ASSP tasks. UTIAS was provided with three chassis, as well as accompanying blade (§3.1) and GPR (§3.2) payloads. They were equipped with an autonomous control system that was designed and installed by UTIAS (§4). This configuration of the Juno chassis was dubbed the *Musketeers*, and the three chassis, 002, 005 and 006 were named *Porthos*, *Aramis* and *Athos* respectively.

1.5 Purpose of This Report

The completion of the Mauna Kea deployment provides an invaluable opportunity to reevaluate the hardware and software decisions made during the project and adjust accordingly. As the group that spent the most amount of time working with the rovers, UTIAS is in the unique position of being able to provide end-user-level feedback to project partners. This report is separated into specific sections for each piece of hardware and software. In each section, the successes and problems encountered with the item are presented as well as a final evaluation of its suitability for continued use as viewed by UTIAS. In a project of this nature, these analyses will not be independent as the ultimate functionality of components depend on their interaction with the entire system. The reader will find exhaustive cross-referencing throughout the report in order to aid them in navigating these interconnections. It is hoped that this will facilitate the reader in understanding the Musketeer rovers.

It is the opinion of the author that the majority of the design decisions were successful. The areas that will require attention before continued research and development are the locomotion system (§2.2), the blade payload (§3.1) and the positioning systems (§5.14, §5.15, §5.16). The end result of the analogue deployment is that that, given an appropriate response to these issues, the CSA has an opportunity to solidify a unique advanced foothold for Canadian industry in the area of ISRU and ASSP.

Chapter 2

Base Rover

The base rover consists of the Juno mobility chassis designed by Neptec and ODG and manufactured by ODG (§2.1) and a locomotion system designed and installed by NORCAT (§2.2).

2.1 The Juno Mobility Chassis

The Juno mobility chassis was designed by Neptec and ODG and manufactured by ODG. It is rugged enough to handle difficult terrain as well as stiff and strong enough to carry various ISRU payloads (§3). It is sealed against dirt and moisture allowing it to withstand the extreme conditions present during the Mauna Kea deployment. The chassis is U-shaped with the opening of the U in the front (Figure 2.1). The wheels on each side of the rover pivot together about the centroid of their drivetrain. The pivoting units on each side are coupled together through a connection across the back of the rover dubbed the *walking arm*. This allows the rover to maintain at least three wheels in contact with the surface while driving over obstacles.

Motors (§2.2) installed on the sides of the chassis make the rover a differential drive vehicle. The motors can be disengaged from the rover drivetrain by loosening four screws and sliding them away from the chassis. The wheels use commercial all-terrain tires but can be replaced with continuous tracks in the future.

2.1.1 Successes

2.1.1.1 Motor Disengagement

Disengaging the motors from the drivetrain allowed early driving algorithms to be tested with full autonomous control hardware without moving the rover and risking damage. Disengaging the motors proved necessary for returning damaged rovers to the lab as well as troubleshooting

(a) (b)

Figure 2.1: Base chassis: (**a**) Solidworks model (**b**) Descending a ramp in the MarsDome. Photographer unknown.

locomotion problems (§2.2.2). Without the quick disengaging motors, debugging the reoccurring hardware problems (§2.2.2.2, §2.2.2.6, §5.1.3.6) would have been more onerous and time consuming. The motor disengagement worked as intended and its frequent use quickly validated its inclusion in the chassis design.

2.1.1.2 Suspension and Handling

A chassis for autonomous ISRU and ASSP work must be stiff and strong enough to support heavy payloads and the forces transmitted from them (§3) while remaining flexible and compliant enough to tolerate autonomously navigating difficult terrain. The stiffness of the chassis, its relatively long track and wheelbase and its low centre of gravity keep the rover stable with most payloads, while the differential drive keeps it manoeuvrable. The coupled, pivot suspension provides continuous wheel contact with the ground preventing the chassis from becoming stuck in all but the most extreme conditions. The chassis provided an able platform for the development of an autonomous control system and easily meets the requirements of the project.

2.1.1.3 Payload Flexibility

The ISRU project was designed around interchangeable payloads (§3) on a common robotic platform. By placing the drivetrain and other locomotion components around the perimeter of the chassis, the U-shape allows payloads to be mounted in the centre while still extending outside the track of the chassis through the open end of the U. While the weight of some payloads and their tolerances (§2.1.2.7) could make installations nontrivial, the general concept of interchangeable payloads was successful and invaluable to the autonomous work.

2.1.2 Problems Encountered

2.1.2.1 Lack of Motor Brake

The direct connection of the wheels, motors and drivetrain can allow rovers on uneven ground to roll away. This is particularly dangerous if a rovers loses power or stops on a slope during autonomous operation. A mechanical brake in the drivetrain would protect the rover and its surroundings from accidental damage. A manual brake would be acceptable; however, an electronically triggered brake that could be activated by the autonomous control system or remote operators to augment the emergency stop system would be preferred.

2.1.2.2 Obstructed Battery Pack Compartments

The battery packs (§2.2) slide into compartments that run the length of the two parallel beams of the U-chassis. The battery packs were removed for charging via covers located on the back of the chassis under the walking arm. On some chassis, there is insufficient clearance between the top of the battery compartment opening and the bottom of the walking arm (Figure 2.2). This prevents the removal or insertion of new batteries while on level ground. Battery replacement is only possible by elevating the front wheel on one side and replacing the opposite battery pack.

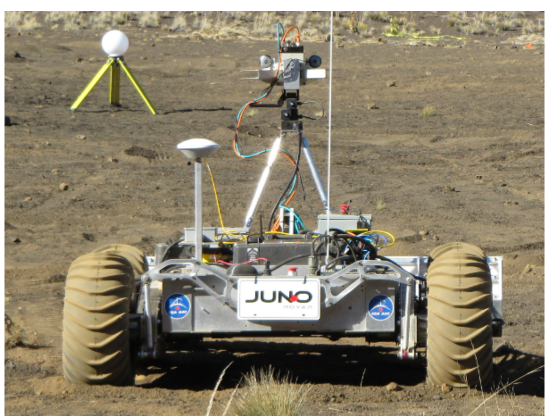


Figure 2.2: Rear view of a rover showing the battery compartments partially-obstructed by the walking arm suspension. Battery compartment covers can be identified by their CSA stickers. Photograph courtesy of Paul Grouchy.

2.1.2.3 Difficult Battery Pack Replacement

Inserting the battery packs (§2.2) into the battery compartments requires precise alignment of the battery pack with respect to the chassis frame, a task made difficult by their weight and the compartment location. Tolerances between the battery packs and the battery compartment allow for batteries to become misaligned and stuck during insertion. The only solution is to remove the partially inserted battery pack and realign. It is unclear if this is the fault of the battery compartments or battery packs.

2.1.2.4 Susceptibility to High-Centring

The motors (§2.2) protrude perpendicular to the chassis between the rover wheels on each side (Figure 2.3). Obstacles that the rover can otherwise easily drive over do not clear the motors and can leave the rover stuck, damaging locomotion system components. ODG has provided an alternative design of the effected locomotion system parts.

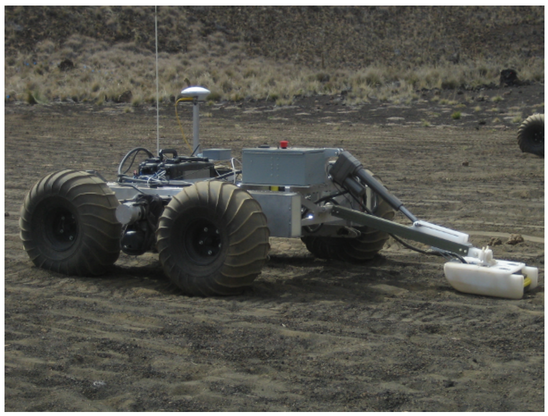


Figure 2.3: Location of the rover motors between fore and aft wheels. Photo courtesy of Terence Fu.

2.1.2.5 Motors Too Hot to Disengage

Disengaging a motor (§2.1.1.1) requires pulling the motor by hand. While under normal operating conditions this is a realistic expectation, the problems experienced with the NPC-T74 motors (§2.2.2.6) frequently left them too hot to be touched. Solutions to the problem used whatever tools were available to pry at the lip of the motor mounting plate and disengage the motor. While this worked, it was difficult and potentially dangerous to the operators and the rover. The incorporation of a handle into the design of the motor mount to facilitate disengagement would be preferred.

2.1.2.6 Insufficient Traction

During excavation, the rovers tires slip and spin, a phenomenon not often seen in traditional excavation machinery. Possible solutions to increase traction include using different tires with more aggressive tread or equipping the rovers with continuous tracks. Tracks had been considered in the initial design but were abandoned in order to limit the forces placed on the drivetrain and chassis.

2.1.2.7 Incompatible Payload Tolerances

The chassis is equipped with a mounting rail that runs along the two parallel sides of the payload area. This mounting rail consists of a regular pattern of holes into which payloads (§3) can be slotted or bolted in. Experience proved that payload installation would sometimes require the use of a spreader to open up the two parallel arms of the chassis. It is unclear if this problem is a result of the payload tolerances, the chassis tolerances or a combination of the two. At the time this report was written, the accepted explanation is that twist in the chassis beams exceeded what was expected by the payload design.

2.1.3 Evaluation

The Juno mobility chassis furnished to UTIAS provided an able platform on which to base an autonomous rover. The development of the autonomous control system would have been significantly delayed and complicated if a less flexible (§2.1.1.3) and forgiving (§2.1.1.1, §2.1.1.2) design had been used. The problems listed within the previous sections (§2.1.2) are provided as end-user feedback to improve an already impressive design and should not overshadow its success. UTIAS unequivocally recommends the continued use of these chassis for ongoing and future work.

2.2 Locomotion System

The base chassis (§2.1) was outfitted with an electric locomotion system that was designed and installed by NORCAT. The locomotion system equips the chassis with the power sources, motors, sensors and controllers to make it a complete Radio Controled (R/C) rover. Battery packs installed into the two parallel arms of the U-chassis provide 24 V to all rover systems.

Each side of the rover is driven by NPC-T74 motors from NPC Robotics Inc. that come equipped with an integral gear box. The motors are controlled by a RoboteQ AX2850 brushed DC motor controller (§5.1) that also reads encoders mounted on the motor shafts. The RoboteQ is built into a circuit that allows it to be controlled by either a Radio Control (R/C) receiver or a serial connection from a computer. The RoboteQ also provides low-level safety protections such as motor current limits, motor stall protection and motor acceleration limits and can control the motor speeds in either open-loop or closed-loop control modes. The motors are rated for 1240 W and draw 50 A at 24 V [4].

2.2.1 Successes

2.2.1.1 RoboteQ AX2850 Brushed DC Motor Controller

For an in-depth evaluation of the RoboteQ AX2850 brushed DC motor controller, see §5.1.

2.2.1.2 Battery Life

The battery packs provide power to all the devices on the rover, including the various payloads (§3) and the autonomous control system (§4). There design was a successful compromise between capacity and footprint, providing enough power for a full day of experimentation while fitting into the parallel arms of the U-chassis.

2.2.1.3 Encoders

The encoders installed into the drivetrain as a part of the locomotion system are used for closed-loop control of motor speed by the RoboteQ AX2850 (§5.1). They are also used to sense when the rover has stalled or become stuck by the *Musketeer Control Server (MCS)* (§6.2). They were used to estimate relative positioning through dead reckoning and can be incorporated into a Kalman filter (§6.3) to provide a prediction of rover location. The installed optical encoders had 2500 pulses per revolution and provided sufficient resolution and update rate.

2.2.2 Problems Encountered

2.2.2.1 RoboteQ AX2850 Brushed DC Motor Controller

For an in-depth evaluation of the RoboteQ AX2850, see §5.1.

2.2.2.2 Unreliable Encoder Wiring

The original installation of the encoder wiring harnesses were unreliable. They were susceptible to randomly disconnecting, shorting or otherwise failing to provide a consistent connection between the encoders and the RoboteQ AX2850 (§5.1). Since the RoboteQ receives raw encoder pulses, it is unable to differentiate between a stopped motor and a failed encoder. This is problematic when operating the RoboteQ in closed-loop control mode. Unable to measure if the motor is turning, the RoboteQ continues to increase the motor current attempting to reach the desired motor speed. This results in a brief period of uncontrolled acceleration followed by the RoboteQ hitting a current limit and either stopping the effected motor or continuing at full speed. These failures can be difficult and time consuming to diagnose as the resulting behaviour is unpredictable and easily confused with failed motors (§2.2.2.6) or malfunctioning autonomous controllers.

Troubleshooting by UTIAS and NORCAT has found that these problems are the result of unreliable encoder wiring. The wiring harness connectors and the junction box inside the chassis required replacement. This task fell to UTIAS and has not been fully implemented as insufficient materials were available. As of this writing, some Musketeer rovers still remain susceptible to encoder failure and the accompanying problems and delays.

2.2.2.3 Inconsistent Battery Lead Length

Battery packs are meant to be identical and interchangeable between chassis to allow for continual work. The battery-pack leads and the matching leads in the chassis battery pack compartments are not all manufactured to the same length. This means that some battery packs can not be used with certain rovers and that certain rovers can only use specific battery packs.

2.2.2.4 Insufficient Battery Lead Clearance

The sealed nature of the chassis requires that the entire battery pack connection be inside the battery compartments. This is accomplished by placing the power connectors at the opening of the compartment and the end of the battery packs. This requires that the chassis connectors be held out of the way while sliding in the battery packs. The tolerances of the chassis and the battery packs make this difficult and further complicate the replacement of battery packs (§2.1.2.2, §2.1.2.3). Once connected, the battery pack leads have to be tucked into the space available at the end of the compartment. Depending on the lead length (§2.2.2.3), this can also be difficult.

2.2.2.5 Insufficient Motor Lead Gauge

The NPC-T74 motors come with integrated electrical leads. These leads are the smallest power conductors in the locomotion system. It was found that the current applied to the motors heated the leads enough to soften or melt their outer insulator and allow the inner insulated conductors to be exposed. This was particularly prevalent when the rover became heavily loaded or stuck. While the leads were insulated at all times, this remains a safety concern.

2.2.2.6 Inadequate Motors

A blade moving at a constant height through the ground collects an increasing amount of material until an equilibrium is reached where as much material spills off the ends of the blade as is collected. This creates forces on the rover that start out initially small, climb as material is collected by the blade and plateau when equilibrium is reached. This is a significant, sustained load on the rover motors, a condition for which the supplier states the motors are not qualified. The motors are described as "wheelchair motors" designed such that two of them are capable of "accelerating 400-500 lbs up to speed on a flat floor and maintaining it." [5]. Such an inertial load would be initially large to accelerate the mass and then small to overcome friction and maintain the speed.

The sustained, high-load occurred during standard rover operations results in increased winding temperatures in the motors and their eventual failure. During the development of the autonomous control system, UTIAS had multiple motors fail, which because of the time required to receive new parts and replace them resulted in a significant loss in available rover-days. Once a motor is damaged, it draws more current to achieve the same speed, heating up faster and increasing the amount of damage. The motors can tolerate high currents, but not high winding temperatures. Observable damage and failures occurred during all required rover uses. Motors failed from the shorter duration, high-load use of digging as well as from the longer duration, low-load use of GPR surveys. No amount or combination of current protection prevented motor failure without completely crippling the capabilities of the rover as reducing the maximum current mainly reduces the rate of temperature increase and not the final temperature. The maximum current was decreased multiple times from the values initially specified by NORCAT. The final current limit of 50 A supplied by NORCAT [4] prevented rovers from driving over a simple 3-in lip into the UTIAS lab or doing any meaningful work and could not be maintained. The rover was operated under the previously specified current limit of 80 A.

In an effort to prevent damage from overheating, the motors were equipped with QualityKits VK0111 Serial Temperature Sensors (§5.17) so that a strict thermal duty cycle could be enforced (§6.9). Mounted on the outside of the motor case, the sensors provided an estimate of the winding temperature inside the motor. The maximum winding temperature had been specified by the supplier as being 130°C [4]. During the Mauna Kea deployment, the motors were shut down if the temperature sensors rose above 80°C. Work would be paused until the motors had cooled back down below 60°C.

On the last day of the deployment, February 10th, 2010, all three rovers were available for excavation work which provided an opportunity to calculate the effective duty cycle of their motors (Tables 2.1, 2.2). *Aramis* and *Porthos* were executing the landing pad controller *C3* (§6.8) while *Athos* was creating an access road. The listed deployment time is the time that any part of the autonomous control system software was running on the rover. The operational time is the time during which the autonomous control system was configured for standard digging;

this correlates to the time that the *MCS* (§6.2) was running. The idle time is the time during which the rover was operational yet not working and is a direct result of overheated motors. The working time is the amount of time the rover spent doing work, either driving or digging. The digging time is the subset of the working time that was spent digging.

The resulting average duty cycle of three rovers is 25% working, 20% digging. However, it may actually be lower as the low operational time for *Athos* indicates that its *MCS* had been stopped by the operators to allow for alignment or demonstrations during a thermal cooling period.

All three rovers started the day in various different conditions. *Aramis* had had both motors replaced with new units during the Mauna Kea deployment GPR survey. By February 10th, 2010, *Aramis* had been operating on its motors for approximately a week and the difference in their wear compared to the similarly used *Porthos* is reflected in their higher duty cycles of 20.3% working, 17.7% digging (Table 2.2). *Aramis* reached thermal shutdown only once during the day (Figure 2.4a).

A lack of documentation makes the age of *Athos*' motors uncertain and its limited operational time makes an analysis of the duty cycle statistically questionable. Data shows that the motor temperatures rose as high as 100 °C after thermal protection was triggered (Figure 2.4b). The temperature rise was observed to continue for as long as 30 minutes after the rover was halted as a result of the considerable thermal mass in the motor housings.

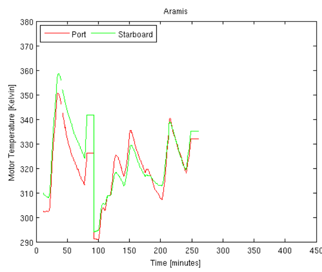
Porthos had motors that were installed at UTIAS in December and only used for one GPR survey before arriving at the Mauna Kea deployment. During the landing pad construction on February 10th, *Porthos* went into thermal shutdown four times (Figure 2.4c) and was also frequently manually paused by the operators to allow for cooling before thermal shutdown was reached. The resulting duty cycles of 12.3% working, 10.5% digging (Table 2.2) were lower than the comparatively used *Aramis* who had newer motors. The difference in their duty cycles can only be attributed to the one extra GPR survey done by *Porthos* and their varying excavation work. The motor temperature plot (Figure 2.4c) reinforces that duty cycle was predominantly caused by motor overheating.

Table 2.1: Rover work times for February 10th, 2010.

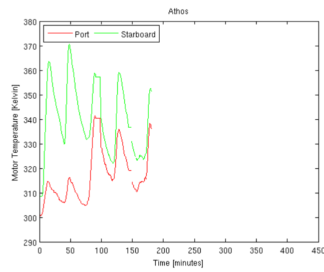
Rover	Deployment Time	Operational Time	Idle Time	Working Time	Digging Time
Aramis	05:47:26	04:45:33	03:47:30	00:58:02	00:50:28
Athos	07:03:55	02:55:34	01:50:46	01:04:47	00:56:57
Porthos	06:41:45	06:22:35	05:35:32	00:47:03	00:40:02

Table 2.2: Rover work times presented as a percentage of operational time.

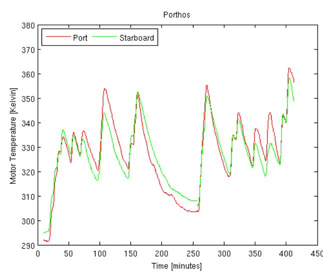
Rover	Idle Time	Working Time	Digging Time
Aramis	79.7%	20.3%	17.7%
Athos	63.1%	36.9%	32.4%
Porthos	87.7%	12.3%	10.5%



(a) *Aramis*



(b) *Athos*



(c) *Porthos*

Figure 2.4: Rover motor temperatures on February 10th, 2010

2.2.3 Evaluation

The locomotion system was one of the largest sources of delays and problems during the development of the autonomous control system. The RoboteQ AX2850 (§5.1), the encoders (§2.2.2.2), and the NPC-T74 motors (§2.2.2.6) presented major problems that have not been satisfactorily resolved. The status of the RoboteQ is discussed elsewhere (§5.1.4); however, some progress has been made identifying possible improvements (§7.1.2). A solution to the encoder problems appears to have been found but insufficient resources were available to complete the upgrade (§2.2.2.2).

The motor and integrated gear box appear to be insufficiently specified for the rover. The NPC-T74 motors are incapable of providing enough power when operating under the supplier specified limitations or withstanding continuous operation. Even the GPR survey, which required the least amount of work from the motors, resulted in the observable degradation of motor performance. Finding a replacement will be difficult as the existing drivetrain was designed around the motor but does appear necessary.

The locomotion system provides the core of any autonomous control system and simply must work better. Without a reliable, robust locomotion system, it is impossible to develop autonomous control systems of any complexity or novelty. UTIAS lost considerable development and operational time to troubleshooting, repairing or bypassing the problems in the locomotion system. They will continue to delay and inhibit work on the autonomous control systems until they are permanently addressed. Continued use of the locomotion system for ongoing work is only recommended while these problems are addressed. Use of the locomotion system as designed is not recommended for future work. A concerted effort is recommended to assure that solutions are found as soon as possible and to develop a locomotion system that compliments the success of the base chassis (§2.1).

Chapter 3

Payloads

The rovers are equipped with the tools necessary to complete the ISRU tasks through interchangeable payloads. The two payloads used by UTIAS for ASSP are a blade payload (§3.1) and a GPR payload (§3.2). Both use the Supervisory Control and Data Acquisition (SCADA) Box (§3.3) and were designed and manufactured by NORCAT. Other payloads designed for the chassis include the RESOLVE drill, a front-loading bucket (the *Load-Haul-Dump*) and a fuel cell.

3.1 Blade Payload

The blade payload was designed and manufactured by NORCAT. It provides the rover with a three Degree-of-Freedom (DoF) blade and the electronics necessary to control it (Figure 3.1). The blade's lift, roll and yaw are controlled by Warner Linear Inc. linear actuators that have integrated positional feedback. The blade is equipped with strain gauges to measure the vertical (blade lift direction) and longitudinal forces. All these devices are interfaced to the autonomous control system through the SCADA Box (§3.3) which is also used for the GPR payload (§3.2).

3.1.1 Successes

3.1.1.1 Supervisory Control and Data Acquisition (SCADA) Box

For an in-depth evaluation of the SCADA Box, see §3.3.

3.1.1.2 Actuator Speed and Range of Motion

The ability of the rover's digging controller (§6.2) to maintain a level cut and/or constant blade load depends on how well it can respond to external disturbances. These take the form of hills, valleys or other ground oscillations. A larger blade actuation range allows the rover to handle

(a) (b)

Figure 3.1: Blade payload: (**a**) Solidworks model (**b**) *Porthos* digging during the Mauna Kea deployment. Photograph courtesy of Paul Grouchy.

larger ground oscillations. A faster blade actuation allows the rover to handle steeper ground oscillations. Given the dimensions of the chassis and the operational speed of rover, the blade actuation range and speed proved appropriate.

3.1.1.3 Blade Strength

Autonomous excavation requires a blade that can handle unusually high levels of stress. Sensors are unable to differentiate partially-buried solid objects from soil and do not specifically target them. Instead, rocks and other solid objects are frequently caught with the bottom edge or corner of the blade during a normal excavation pass. This places an unbalanced load on the blade as the rock is pried out from the soil. The blade proved strong enough to withstand this abuse regardless of the digging material and was quite successful in freeing and removing buried objects from the ground.

3.1.1.4 Ease of Installation

The ISRU project depends on the ability to quickly and simply exchange payloads. While weight and tolerances (§2.1.2.7) make the blade payload more difficult to install than the GPR payload, its installation and removal is simple given the appropriate equipment. Installation during the Mauna Kea deployment could take as little as 30 minutes with the proper personnel.

3.1.2 Problems Encountered

3.1.2.1 Supervisory Control and Data Acquisition (SCADA) Box

For an in-depth evaluation of the SCADA, see §3.3.

3.1.2.2 Incompatible Payload Tolerances

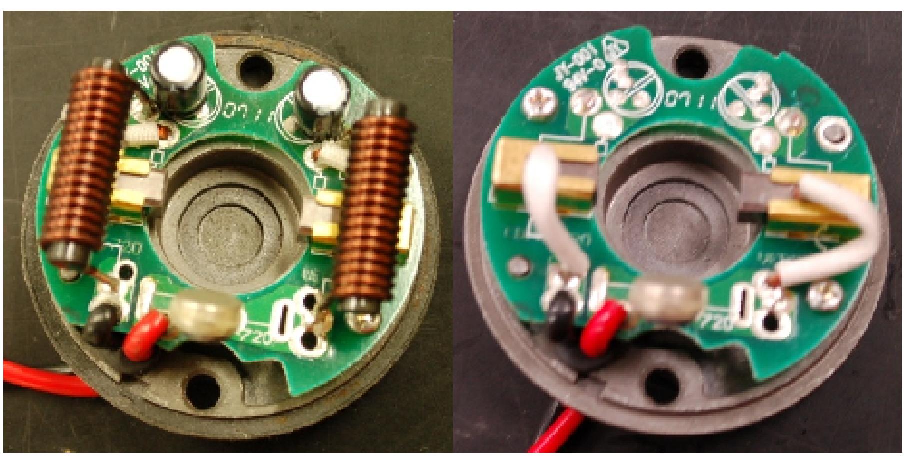
For an in-depth discussion of payload tolerances, see §2.1.2.7.

3.1.2.3 Inconsistent Wiring Harnesses

For an in-depth discussion of wiring harness inconsistency, see §3.3.2.2.

3.1.2.4 Broken Actuators

Inside the Warner Linear Inc. linear actuators are boards that contain capacitors and inductors as part of the control circuit (Figure 3.2a). These capacitors were found to fail catastrophically (Figure 3.3b), filling the actuator control area with dielectric, jamming the commutator (Figure 3.3a) and causing the actuator to intermittently stall. No analysis was done to identify the reason for the failures; NORCAT made the decision to remove the capacitors and inductors from the control circuit (Figure 3.2b). UTIAS hypothesizes that the blown capacitors may be a result of the voltage instability seen in the SCADA Box (§3.3.2.4) as the capacitor failure is typical to that of exposure to higher-than-rated voltages. The removal of this circuit may now be a contributor to noise problems observed with the SCADA Box (§3.3.2.7).

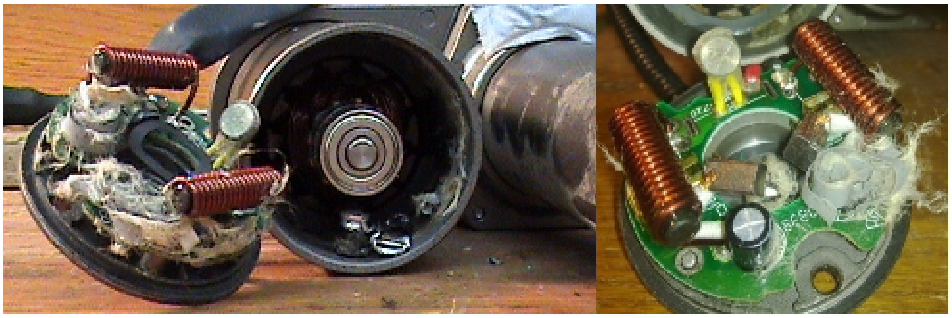


(a) **(b)**

Figure 3.2: Actuator control board: (**a**) Original control board with capacitors installed. (**b**) Control board as modified by NORCAT.

3.1.2.5 Limited Load Cell Sensitivity

The autonomous control system uses the load cells to measure and control excavation. Their resolution and sensitivity will limit the minimum depth of excavation passes. Sensitive load cells will enable shallow passes and allow for smoother and more level final results. The load



(a) (b)

Figure 3.3: Failed actuator control board: (**a**) Damaged control board removed from a linear actuator. Dielectric can be seen inside the actuator. (**b**) Close up of damaged control board.

cells installed in the blade payload output 0-10 V signals through a Weidmüller WAS5 Pro Bridge Controller in the SCADA Box (§3.3).

The load cells had a limited range compared to the tractive effort available from the Juno chassis (§2.1). An unloaded blade averaged a horizontal load cell reading of 4.5 V. The maximum force that could be tracked by the low-level digging controller without slipping (§6.2) depended on the soil conditions but averaged 6 V. Considering back-blading, this results in an approximate horizontal working range of only 3 V within the potential 10 V range of the Weidmüller WAS5. The vertical load cells suffered from a similar lack of sensitivity, averaging 7 V when the blade was held aloft. It now appears that it may be possible to recalibrate the Weidmüller WAS5 and increase the load cell sensitivity [6].

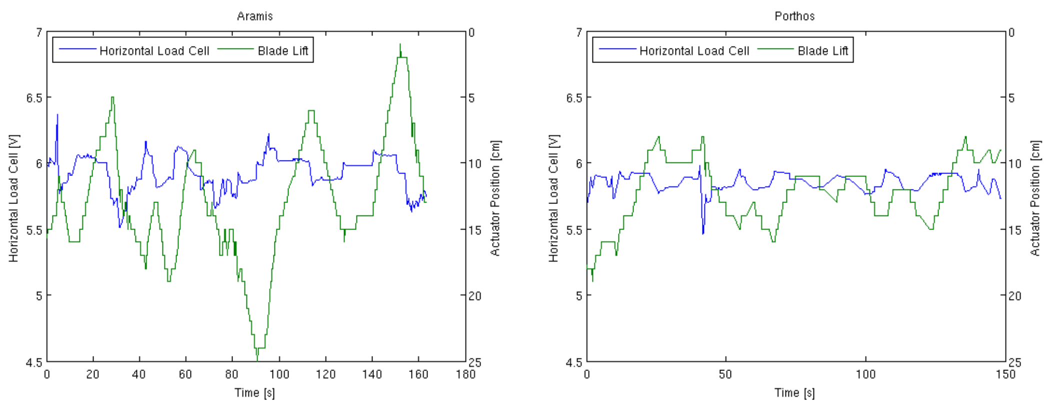
3.1.2.6 Load Cell Noise

The usability of the load cells is further reduced by offsets and noise. After a typical excavation pass consisting of lowering the blade, digging a set distance and then depositing the collected material by raising the blade, the unloaded horizontal load cell could vary by as much as 0.5 V from its initial reading. This variation represents 33% of the forward load cell working range (16.7% of the bidirectional range). Such varying offsets require the low-level digging controller (§6.2) to take deeper excavation passes to guarantee a suitable signal-to-noise ratio. The load cells also had high frequency noise during periods of constant loading. This noise may be a result of their design or the larger electrical problems plaguing the SCADA Box (§3.3.2).

3.1.2.7 Load Cell Hysteresis

The blade payloads utilized by UTIAS during the Mauna Kea deployment exhibited previously unobserved load-cell hysteresis and sticking. Vertical load cells could become stuck, resulting in rovers erroneously measuring ground contact at the beginning of excavation passes. This tricked the digging controller (§6.2) into registering the ground level as higher than it actually was, a measurement that persists for the whole excavation pass. It was found that sharply striking the blade would free the stuck load cells.

This hysteresis was also observed in the horizontal load cells, effecting the performance of the low-level digging controller (§6.2). While forces frequently varied smoothly, periods of suspicious constant force were observed in both *Aramis* and *Porthos* (Figure 3.4). This may have contributed to observed digging oscillations (Figure 3.5).



(a) *Aramis* **(b)** *Porthos*

Figure 3.4: Horizontal load cell measurements and lift position during an excavation pass. Lift position is measured as stroke length and increases as the blade is lowered. Suspicious load cell readings are present for *Aramis* at approximately 95-110 and 130-140 seconds and for *Porthos* at approximately 120-130 seconds.

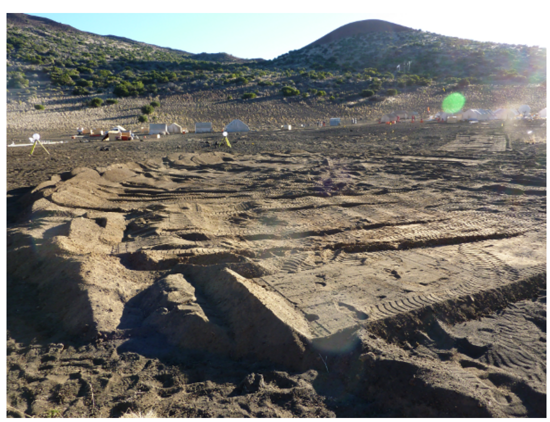


Figure 3.5: Example of an excavation pass with unacceptable oscillations. A flat excavation pass (foreground) is contrasted by an oscillating excavation pass (midground) during the Mauna Kea deployment. Photograph courtesy of Kenneth Law.

3.1.3 Evaluation

The blade, support and actuators are solidly constructed (§3.1.1.3) and reasonably specified (§3.1.1.2) for the chassis. Considering its weight, the payload proved to be simple to install and remove (§3.1.1.4); however, there remains room for improvement (§3.1.2.2). Problems experienced with the failed actuators (§3.1.2.4) are most logically attributed to problems seen in the SCADA Box (§3.3.2). The initial problems with inconsistent implementation (§3.1.2.3) were time consuming but have been overcome.

The load cells showed insufficient sensitivity compared to the force available from the motors and the chassis (§3.1.2.5). The load cells also showed an unacceptable level of noise (§3.1.2.6), and hysteresis or sticking (§3.1.2.7). While working within the limitations of imperfect sensor data is the fundamental problem of experimental robotics, blade force sensing limits the quality of autonomous excavation. To improve future results, improvements must be made to the designs of the load cells and their accompanying circuits in order to provide better data to the autonomous controllers.

The blade payload depends heavily on the SCADA Box (§3.3) whose problems are detailed elsewhere (§3.3.2). If a solution can be found to these problems, continued use of the blade payload is recommended for ongoing and future work.

3.2 Ground Penetrating Radar (GPR) Payload

The Ground Penetrating Radar (GPR) payload was designed and assembled by NORCAT (Figure 3.6a) to deploy a Sensors & Software OEM Noggin Network Interface Card (NIC) GPR (§5.3). It consists of a GPR sleigh attached via a passive three DoF joint (Figure 3.6b) to a fibreglass arm controlled by a single Warner Linear Inc. actuator. The actuator was equipped with positional feedback and was controlled by the same SCADA Box (§3.3) as used in the blade payload (§3.1).

3.2.1 Successes

3.2.1.1 Supervisory Control and Data Acquisition (SCADA) Box

For an in-depth evaluation of the SCADA Box, see §3.3.

(a) (b)

Figure 3.6: GPR Payload: (**a**) *Aramis* equipped with a GPR during the Mauna Kea deployment. Photograph courtesy of the author. (**b**) GPR Sleigh

3.2.1.2 Sensors & Software OEM Noggin Network Interface Card (NIC) Ground Penetrating Radar (GPR)

For an in-depth evaluation of the Sensors & Software OEM Noggin NIC GPR, see §5.3.

3.2.1.3 Actuator Speed and Range of Motion

The lift actuator is the only active DoF on the GPR sleigh. It must extend enough that the GPR contacts the ground while also contracting enough to raise it for safe transport. The actuation speed is less of a concern, as the lack of feedback on the passive DoFs (§3.2.2.4) prevents implementing controllers to track ground height. Similar to their implementation on the blade payload (§3.1.1.2), the Warner Linear Inc. actuators prove to be an appropriate selection.

3.2.1.4 Ground Tracking

The GPR transmits pulses of microwave energy at 1000 MHz and measures the time it takes for reflections to return. These reflections are caused by the different dielectric constants of materials under the surface. If the GPR is not sitting directly on the surface of the ground, a significant amount of this energy will be reflected by the ground surface and lost. The purpose of the three DoF sleigh was to allow the GPR to passively follow the ground, maintaining good contact at all times and assuring an optimal GPR measurement. While there were some minor problems with the initial sleigh design, the final design followed ground profiles reasonably well when moving straight. The sleigh was not designed to turn while deployed.

3.2.1.5 Ease of Installation

The ISRU project depends on the ability to quickly and simply exchanged payloads. The GPR payload's installation and removal is simple given the appropriate equipment. Installation during the Mauna Kea deployment could take as little as 15 minutes with the proper personnel.

3.2.2 Problems Encountered

3.2.2.1 Supervisory Control and Data Acquisition (SCADA) Box

For an in-depth evaluation of the SCADA Box, see §3.3.

3.2.2.2 Sensors & Software OEM Noggin Network Interface Card (NIC) Ground Penetrating Radar (GPR)

For an in-depth evaluation of the Sensors & Software OEM Noggin NIC GPR, see §5.3.

3.2.2.3 Broken Actuators

For an in-depth discussion of the problems with the Warner Linear Inc. linear actuators, see §3.1.2.4.

3.2.2.4 Lack of Feedback on Passive Degrees-of-Freedom (DoFs)

The three unactuated DoFs on the GPR sleigh allow it to track the profile of the ground passively but provide no information about their position. If position feedback were available, controllers could be implemented to track the ground height and keep the passive DoFs within safe operating ranges. This would prevent damage to the GPR payload and decrease the number of inaccurate measurements caused by poor ground contact. Currently the autonomous control system requires human observers to make sure that the GPR sleigh is not damaged by changes in the ground that are larger than its passive range of motion. Provided that surveyed areas were relatively flat and smooth, these limitations are not a problem, however position feedback would be necessary for a truly autonomous rover.

3.2.3 Evaluation

The Ground Penetrating Radar (GPR) payload was largely successful in its final implementation, showing good actuator range (§3.2.1.3) and passive ground tracking (§3.2.1.4) on a easily interchangeable payload (§3.2.1.5). This allowed for the successful development of a robotic, multiagent GPR survey, however the lack of position sensors (§3.2.2.4) limited the system autonomy in all but near-ideal environments. Autonomous GPR surveys were only possible in areas where the ground was smooth and free of obstacles. To develop proper autonomous behaviour for GPR surveys, sensors on these DoFs are necessary. They would allow the autonomous control system to actuate the arm in response to ground changes, tracking a safe displacement in the passive DoFs and preventing damage to the payload and inaccurate measurements. The addition of this feedback is required for development of complete autonomous GPR surveying.

The GPR payload depends heavily on the SCADA Box (§3.3) whose problems are detailed elsewhere (§3.3.2). If a solution can be found to these problems, continued use of the GPR payload is recommended for ongoing and future work.

3.3 Supervisory Control and Data Acquisition (SCADA) Box

The Supervisory Control and Data Acquisition (SCADA) Box was designed and manufactured by NORCAT to provide the devices necessary to interface computers to the blade (§3.1) and GPR (§3.2) payloads. Built around the Omega OMB-DAQ-3001 (§5.2) data acquisition module, the SCADA Box contains Advanced Motion Controls 25A8 Analog Servo Drives to control the linear actuators, Weidmüller WAS5 Pro Bridge ACT20P bridge controllers to control the load cell Wheatstone bridges on the blade payload and power supplies based around Vicor Micro DC-DC converters to power the devices (Figure 3.7).

3.3.1 Successes

3.3.1.1 Omega OMB-DAQ-3001 Data Acquisition Module

For an in-depth evaluation of the Omega OMB-DAQ-3001 Data Acquisition Module, see §5.2.

3.3.1.2 Common Autonomous Control Interface to Multiple Payloads

When implemented properly, the design consistency between payloads offered by the common SCADA Box was beneficial to the development of the autonomous control system. The common interface it provided sped up development as well as facilitated debugging by reducing the number of devices in the autonomous control system.

3.3.2 Problems Encountered

3.3.2.1 Omega OMB-DAQ-3001 Data Acquisition Module

For an in-depth evaluation of the Omega OMB-DAQ-3001 Data Acquisition Module, see §5.2.

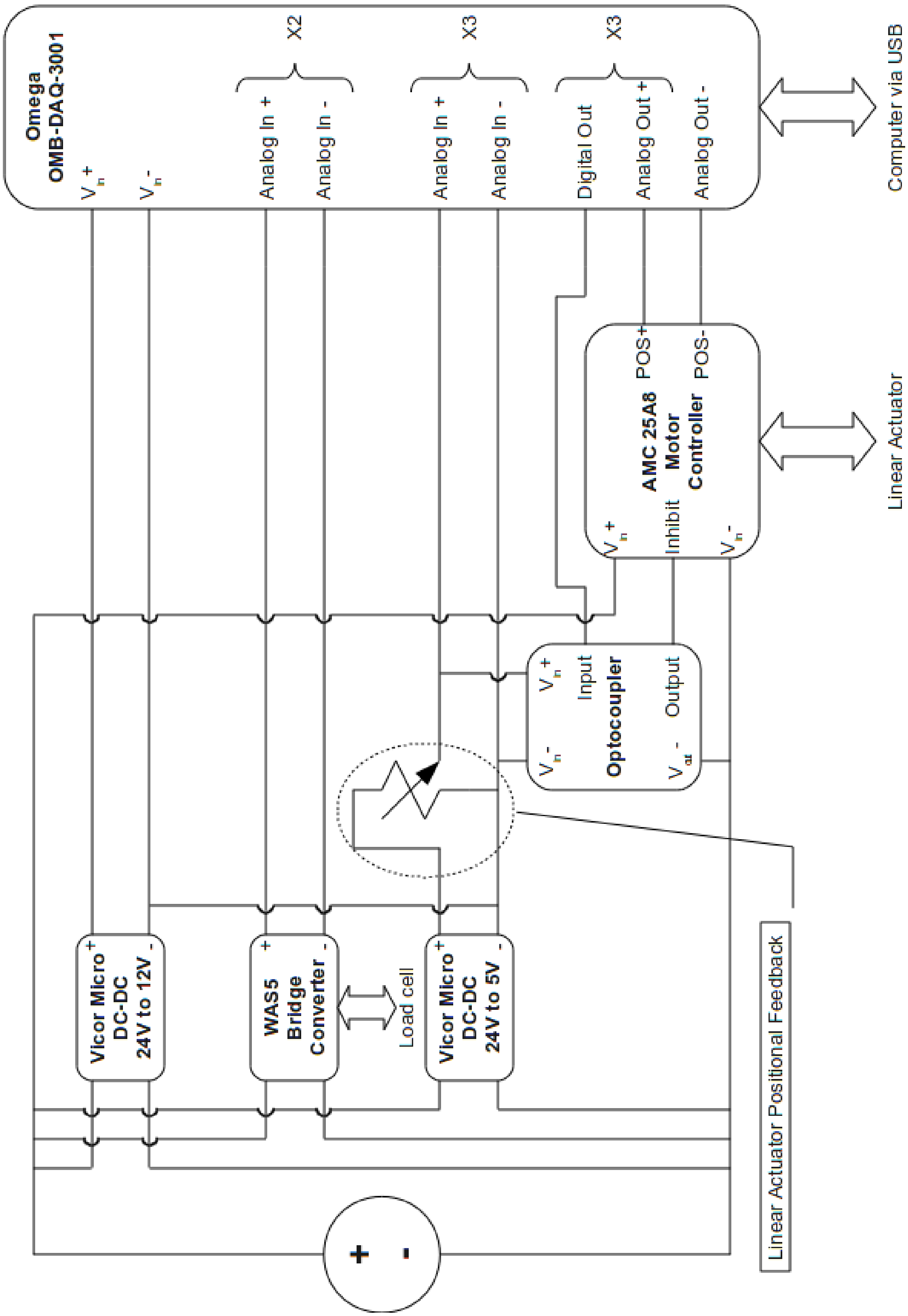


Figure 3.7: Simplified electrical schematic of the SCADA Box installation [7].

3.3.2.2 Inconsistent Wiring Harnesses

The SCADA Boxes are used on both the blade (§3.1) and GPR payloads (§3.2) so that the actuators can share the same configurations and controllers. The first SCADA Box and blade payload provided to UTIAS used different wiring for the actuator cable harnesses than documented or found elsewhere. This meant that a GPR payload had to be modified to work with the inconsistent SCADA Box. The troubleshooting and repair were undertaken by UTIAS.

3.3.2.3 Flipped Actuator Enable Bits

The Advanced Motion Controls 25A8 Analog Servo Drives inside the SCADA Box must be enabled by the Omega OMB-DAQ-3001 (§5.2) to prevent accidental operation of the linear actuators. The original blade payload and SCADA Box sent to UTIAS utilized an incorrect design that resulted in the actuators always being operable. This was fixed by changing the default configuration file for the OMB-DAQ-3001.

3.3.2.4 Voltage Oscillations

Inside the SCADA Box are two Vicor Micro DC-DC converters acting as 5 V and 12 V power supplies for the actuator feedback and the Omega OMB-DAQ-3001 (§5.2) (Figure 3.7). The Vicor Micro DC-DC converters are isolated, switching converters, but are not self-contained power supplies [8]. They require external circuitry to guarantee their stability (Figure 3.8). This circuitry was not installed around the Vicor converters. As a result, the converters oscillated by as much as 30% of their output voltage (Figures 3.9a, 3.9d) at high frequencies. It is reasonable to hypothesize that this may have been a contributing factor to the failure of actuators (§3.1.2.4). These problems were diagnosed and repaired by UTIAS following the design recommendations laid out in the Vicor *Design Guide & Application Manual* [8] which resulted in significantly improved performance (Figures 3.9b, 3.9c, 3.9e, 3.9f).

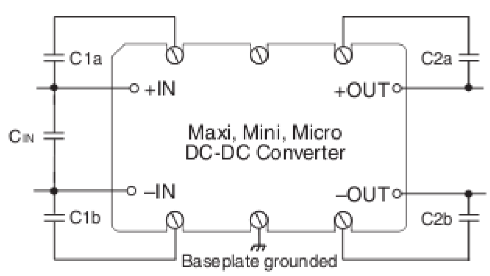
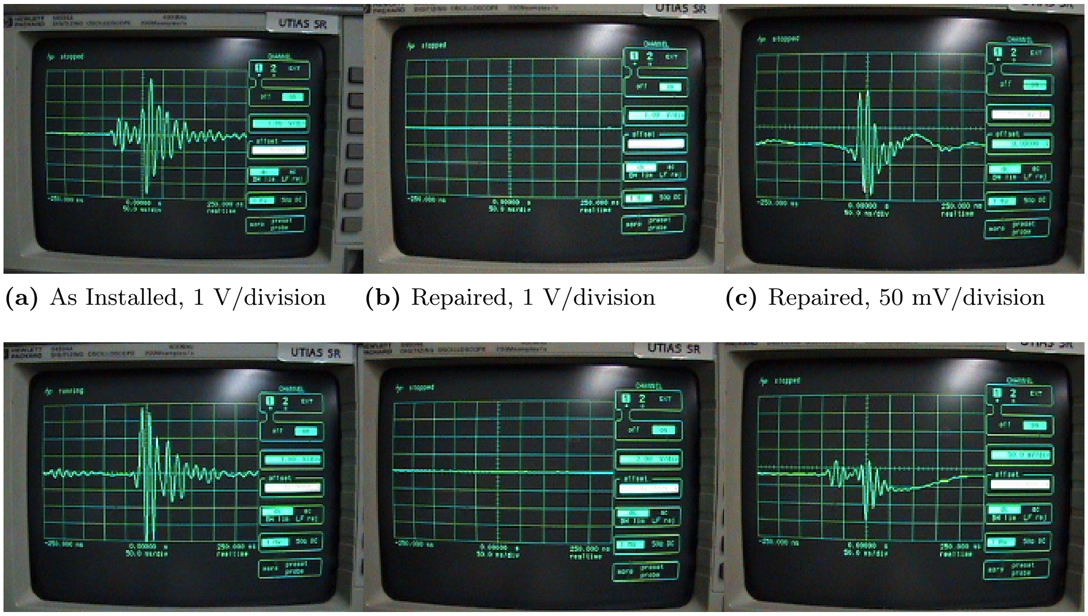


Figure 3.8: Minimum recommended filters for Vicor Micro DC-DC converters [8].



(a) As Installed, 1 V/division **(b)** Repaired, 1 V/division **(c)** Repaired, 50 mV/division

(d) As Installed, 1 V/division **(e)** Repaired, 1 V/division **(f)** Repaired, 50 mV/division

Figure 3.9: Output of the 12 V Vicor Micro DC-DC converter installed in a SCADA Box. Figures (**a**)-(**c**) show the low output terminal relative to input ground, figures (**d**)-(**e**) show the high output terminal relative to 12 V above input ground.

3.3.2.5 Voltage Offsets

The input and output channels of the Vicor Micro DC-DC converters are electrically isolated. This provides no reference between the input and output voltages and allows for output circuits to be electrically independent of the input voltages. Care must be taken to make sure that the isolation is not unexpectedly bridged. If it is, the circuits are susceptible to developing offsets, ground-loops and other instabilities.

The original SCADA Box was designed to contain three independent circuits. There was to be no connection between the 5 V and 12 V Vicor converter low output terminals or the battery ground. It was found that this was not the case. In an attempt to remedy the problem, NORCAT installed optoisolators [9] and a connection between the low output terminals of the two Vicor converters. This did not repair the unexpected connection between the Vicor output terminals and the battery ground.

Experimental analysis showed that this nontrivial resistance between the Vicor outputs and the battery packs was creating voltage offsets. All circuits connect to the Omega OMB-DAQ-3001 (§5.2) (Figure 3.7) and this may be the source of the bridging (§5.2.3.1). The 5 V Vicor converter powers the actuator feedback. The 12 V Vicor converter powers the OMB-DAQ-3001 itself. The 24 V battery packs power the load cells through the Weidmüller WAS5 Pro Bridge ACT20P bridge controller and the Warner Linear Inc. actuators through the Advanced Motion Controls 25A8 Analog Servo Drives.

UTIAS chose to connect the low output terminals to battery ground, bypassing the Vicor Micro DC-DC converter isolation and providing a constant, near-zero resistance path. This removes the unstable voltage offsets and assures that the outputs of the converters will remain fixed relative to the battery mains. While this can allow power devices to discharge current through signal grounds, the use of low-resistance connections reduces this risk and it remains preferential to a design that was creating voltage discharges (§3.3.2.6).

3.3.2.6 Discharge of Voltage Offsets through USB

The SCADA Box communicates to the autonomous control package via USB to a computer or other USB host device. If the power supplies for the two devices are completely independent and isolated, the USB connection will bridge them and may result in current spikes. This can be dangerous to connected equipment and may be the source of observed driving surges (§5.1.3.3).

3.3.2.7 Irrational Analogue Sensor Readings

The actuator feedback and load cell readings are read by the analogue inputs of the Omega OMB-DAQ-3001 (§5.2). They are used in various autonomous control algorithms, including digging controllers and closed-loop blade positioning controllers. The OMB-DAQ-3001 would frequently read values for the load cells and actuator positions that were higher than their power supplies could provide. Attempting to operate the payloads under these conditions would result in behaviour dangerous to the hardware. Values could only be fixed by reconnecting the software to the OMB-DAQ-3001. This problem was observed using both the UTIAS developed autonomous control software (§6.2) and the provided *Omega DAQView*. The problem may be related to voltage oscillations or noise in the SCADA Box circuitry (§3.3.2.4), however no causes or fixes have been found.

3.3.2.8 Actuation Surges

The Musketeer rovers were found to be prone to sudden surges of motion when unrelated electrical devices were connected. A common example was the connection of the SCADA Box to the ASUS S101 Eee PC (§5.10) via USB. If powered, the linear actuators were observed to occasionally move when the connection was made. Actuation of the rover main drive motors was sometimes also observed (§5.1.3.3). These behaviours may be the result of improperly isolated circuits developing voltage offsets that are then discharged through the computer. In addition to creating an unsafe workspace around the rovers, these events are dangerous to electronics as they may create damaging levels of current. At least one failed RoboteQ was attributed to these driving surges and they may also have been responsible for the failed actuators (§3.1.2.4). Adapting standard operating procedures so that devices were never connected when the rover was powered helped prevent these behaviours from occurring. However, this behaviour is unacceptable and may be symptomatic of underlying problems in the design of the SCADA Box and/or accompanying RoboteQ circuitry (§5.1.3.6).

3.3.3 Evaluation

The Supervisory Control and Data Acquisition (SCADA) Box was a challenge throughout the development and deployment of the Musketeer rovers. It was implemented inconsistently (§3.3.2.2, §3.3.2.3) and proved to be unstable (§3.3.2.4, §3.3.2.7) and electrically unsafe (§3.3.2.5, §3.3.2.6) having a history of causing damage to external circuits connected to it (§3.3.2.8). The original circuit was designed under fundamental misconceptions and misunderstanding about the properties and performance of its parts, especially the Omega OMB-DAQ-3001 (§5.2.3.1) and the Vicor Micro DC-DC Converters and as a result has had to be repeatedly patched and modified by UTIAS to address issues discovered during operation. Finding and implementing fixes were a constant drain on the limited resources of UTIAS as they required many hours of electrical and software debugging in addition to the simultaneous development of the autonomous control system. The SCADA Box is now so far removed from its original design, that a complete redesign appears necessary. If electrical isolation is desired, care must be made to make sure that it is consistent, properly thought out and complete, including power and communication between all devices that connect to the SCADA Box. A specific area of concern will be guaranteeing that USB devices do not bridge the electrical isolation. Continued use of the SCADA Box for ongoing work is only recommended while these problems are being repaired. Use of the SCADA Box as currently designed is not recommended for future work.

Chapter 4

Autonomous Control System

The autonomous control system was designed and assembled by UTIAS. It consists of a sensor mast mounted at the front of the rover, interoceptive and exteroceptive sensors mounted throughout the chassis, actuators, an ASUS S101 Eee PC netbook (§5.10) in the Musketeer Control Box (MCB) (§5.9) and various interface converters and power supplies in the Musketeer Interface Box (MIB) (§5.4).

4.1 Sensors and Actuators

The autonomous control system uses sensors installed specifically for it as well as those that are a part of other rover systems (§5). There are general interoceptive sensors measuring the rover's health and state, such as motor current and battery voltage available through the RoboteQ AX2850 (§5.1), temperature readings from various devices as well as the QualityKits VK011 Serial Temperature Board (§5.17). There are sensors measuring the actuator positions of the blade (§3.1) and GPR (§3.2) payloads available through the Omega OMB-DAQ-3001 (§5.2) in the SCADA Box (§3.3). The DirectedPerception PTU-D47 Pan-Tilt Unit (§5.18) also provides feedback of its orientation. There are also position specific interoceptive sensors such as the encoders available through the RoboteQ.

The autonomous control system makes use of exteroceptive sensors as well. Load cells in the blade payload accessible through the OMB-DAQ-3001 measure the vertical and horizontal force on the blade. A SICK LMS111 Laser Measurement System (§5.19) is mounted on the Pan-Tilt Unit (PTU) along with two Microsoft LifeCam VX-3000 cameras (§5.20). The MicroStrain 3DM-GX2 Attitude Heading Reference System (AHRS) (§5.14) uses a magnetometer to measure the rover heading and accelerometers to measure the rover roll and pitch. Two separate Global Positioning System (GPS) receivers were used for position information. A u-blox EVK-5H provides Wide Area Augmentation System (WAAS) corrected positions (§5.15) while the Trimble GPS Pathfinder ProXRT provides OmniSTAR corrected differential positions (§5.16).

4.2 Power

The main battery packs provide an unregulated 24 V to all rover components (Figure 4.1). The various devices and sensors of the autonomous control system all have different power requirements (Table 4.1). The Musketeer Interface Box (MIB) (§5.4) supplies and distributes the power for the devices (Figure 4.2). Inside the MIB are 3 independently-fused Vicor Micro DC-DC converters that convert the power to 5 V, 8 V and 12 V. The converters are operable from around 28 V down to 18 V and can provide over 100 W each. This provides clean, regulated power to the devices even with heavy current draw during rover acceleration and excavation. Each device is independently fused at its power supply's stated maximum power (Table 4.1). This assures that malfunctioning or damaged devices will not damage the power bus or the DC-DC converter.

Table 4.1: Power requirements of the autonomous control system devices.

Voltage [V]	**Device**	**Max Current [A]**	**Fuse [A]**
5	Lantronix UBox 2100	3	3
5	Belkin 7-Port Mobile USB Hub	4	5
5	QualityKits VK011 Serial Temperature Sensor	0.1	0.1
5	**Total**	**7.1**	**-**
8	MicroStrain 3DM-GX2 AHRS	0.56	0.6
8	**Total**	**0.56**	**-**
12	Cisco SD208 8-Port 100Mbps Switch	0.5	0.5
12	MIB Interface	-	0.5
12	ASUS Eee PC S101 (MCB)	3	
12	Hawking HSB2 Hi-Gain Wireless Signal Booster (MCB)	1.25	5
12	MCB Interface (MCB)	-	
12	**Total**	**5.75**	**-**
24	Perle IOLAN SDS4 Device Server	1	1
24	DirectedPerception PTU-D47	3	3
24	SICK LMS111 Laser Measurement System	2.1	3
24	**Total**	**6.1**	**-**

4.3 Device Interfaces

The various devices in the autonomous control system use different communication protocols (Table 4.2). As it is impractical to find a computer with a sufficient number of ports to communicate directly with every device, hubs, converters and switches are necessary to connect them to the computer. In order to provide the Musketeer Control Box (MCB) with a single data connection to the MIB, all devices in the MIB are converted to RJ45 Ethernet (Figure 4.3).

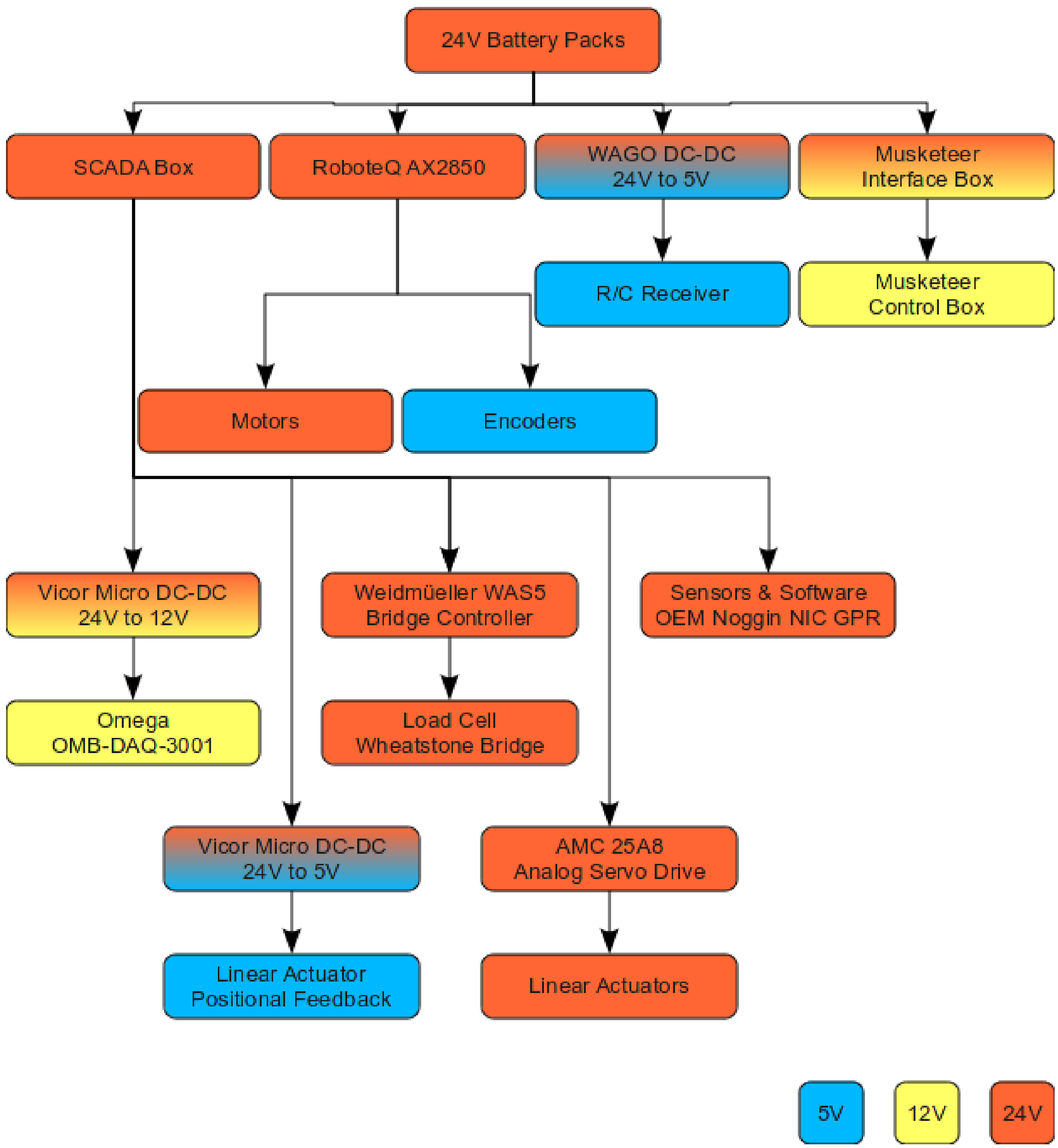


Figure 4.1: Power topology of a Musketeer rover.

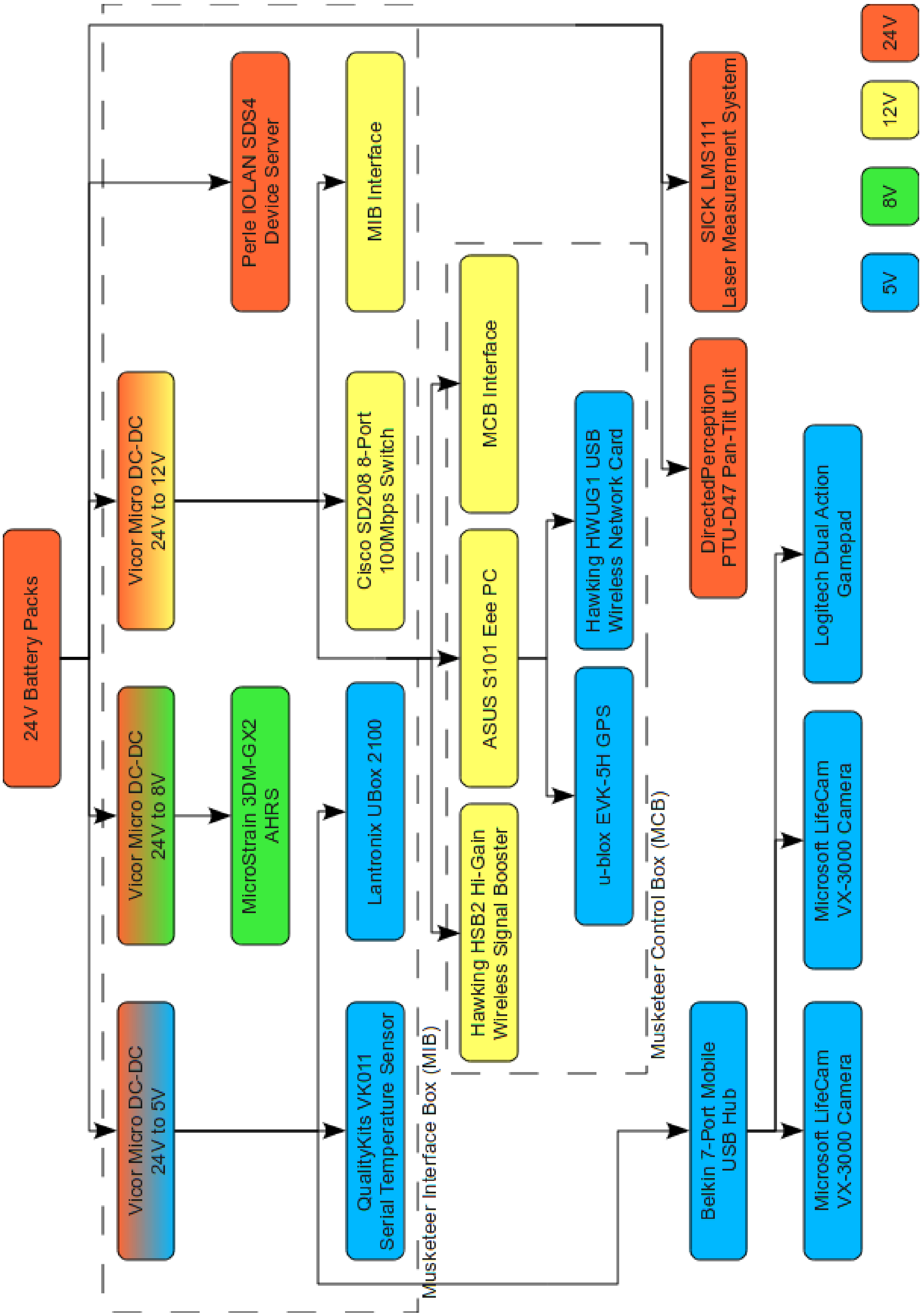


Figure 4.2: Power topology of the autonomous control system.

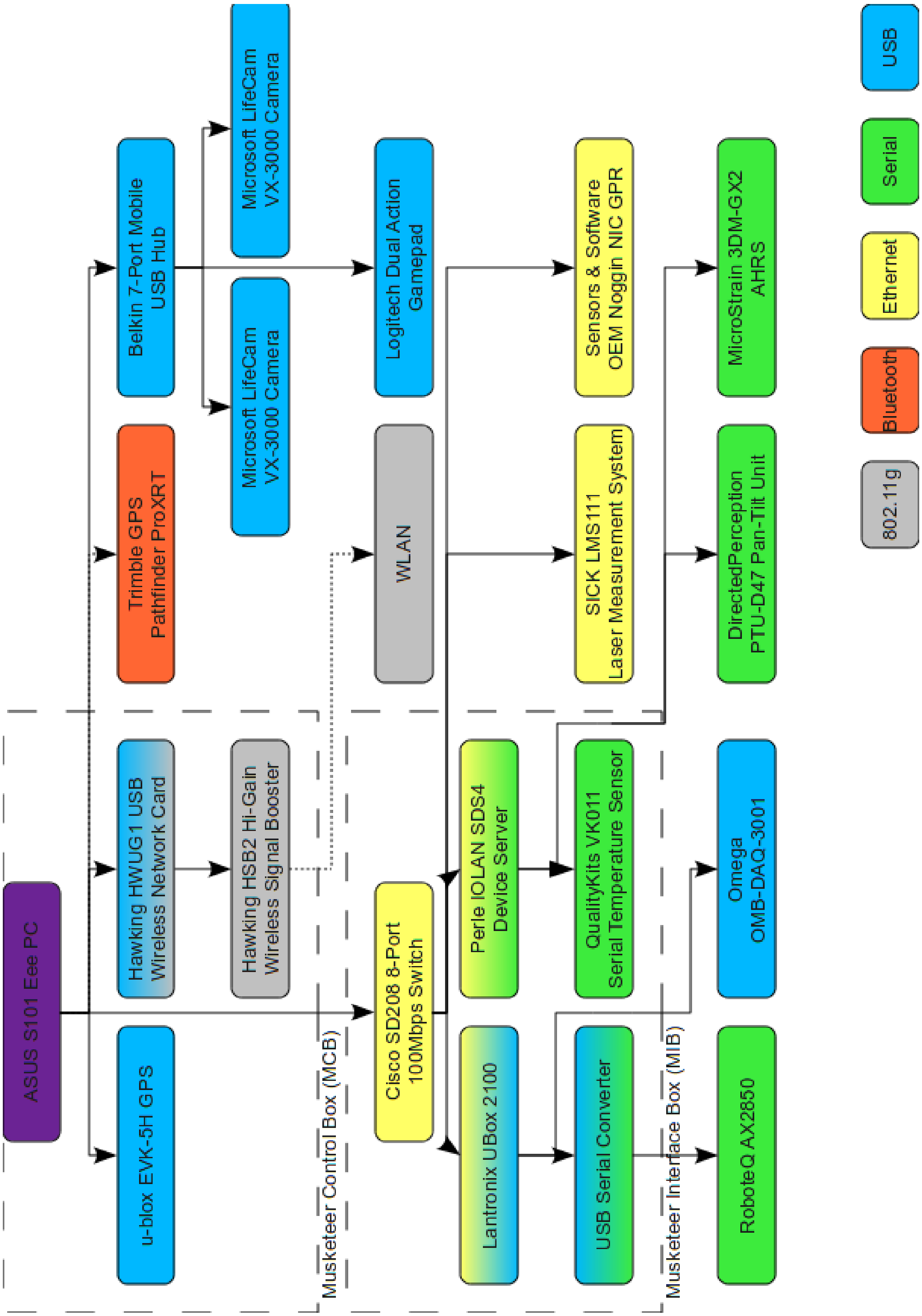


Figure 4.3: Interface topology of the autonomous control system.

Table 4.2: Interfaces of the autonomous control system devices.

Device	Native Interface	Final Interface
Musketeer Control Box		
u-blox EVK-5H GPS	USB	USB
Hawking Wireless System	USB	USB
Microsoft LifeCam VX-3000	USB	USB
Trimble GPS Pathfinder ProXRT	Bluetooth	Bluetooth
Musketeer Interface Box		
Sensors & Software OEM Noggin NIC GPR	Ethernet	Ethernet
SICK LMS111 Laser Measurement System	Ethernet	Ethernet
Omega OMB-DAQ-3001 Data Acquisition Module	USB	Ethernet
RoboteQ AX2850 Brushed DC Motor Controller	Serial	Ethernet
MicroStrain 3DM-GX2 AHRS	Serial	Ethernet
QualityKits VK011 Serial Temperature Sensor	Serial	Ethernet
DirectedPerception PTU-D47	Serial	Ethernet

4.4 Internal Rover Network

Conceptually the rover is a mobile Local Area Network (LAN). The Ethernet devices sit on a local subnet behind the Eee PC that acts as a wireless gateway to the larger network (Figure 4.4). This allows each rover's network topology to be identical and globally accessible without creating network conflicts.

4.5 Software

The autonomous control system software is based on the ArgoSoft framework developed by UTIAS. The ArgoSoft framework uses a server/client structure to make hardware devices, such as motors and sensors, available to multiple software clients. Conflicting commands are dealt with by assigning each client a priority which the server uses to decide which command will be executed. The modularity of the ArgoSoft framework provides a large degree of platform portability which reduces the work required to integrate old hardware and develop algorithms on new platforms.

The autonomous control system uses the ArgoSoft framework to create a three level structure consisting of four general types of programs. First are Device Servers (§4.5.1), implementations of basic ArgoServers that serve sensors and hardware to connected clients. Last are Control Clients (§4.5.3) and Logging Clients (§4.5.4), implementations of ArgoClients that connect to servers either to read inputs and issue commands or simply to record data. In the middle sit Algorithm Servers (§4.5.2), intermediate servers that connect to the low-level Device Servers and provide high-level algorithms to connected clients (Figure 4.5).

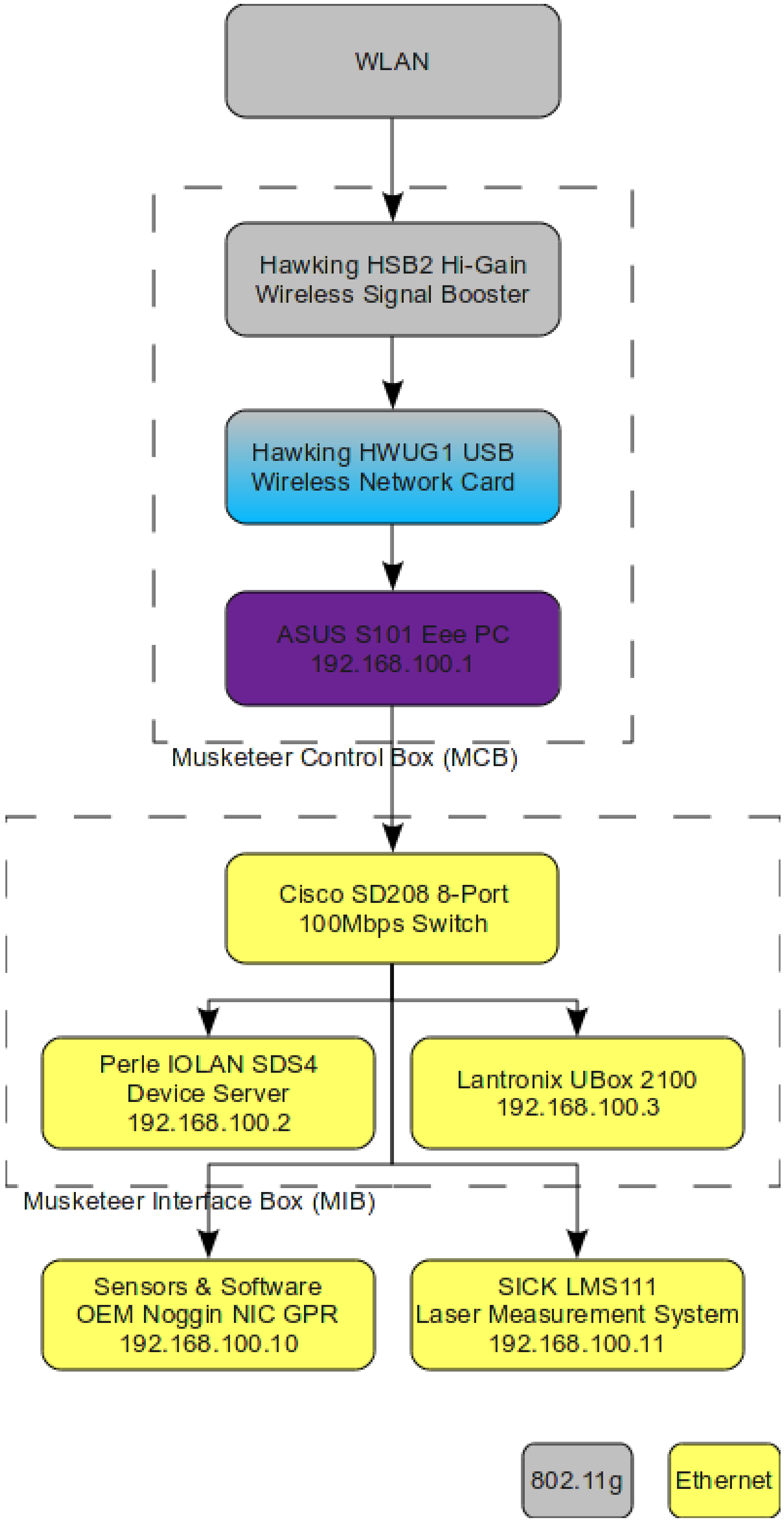


Figure 4.4: Network topology of the autonomous control system.

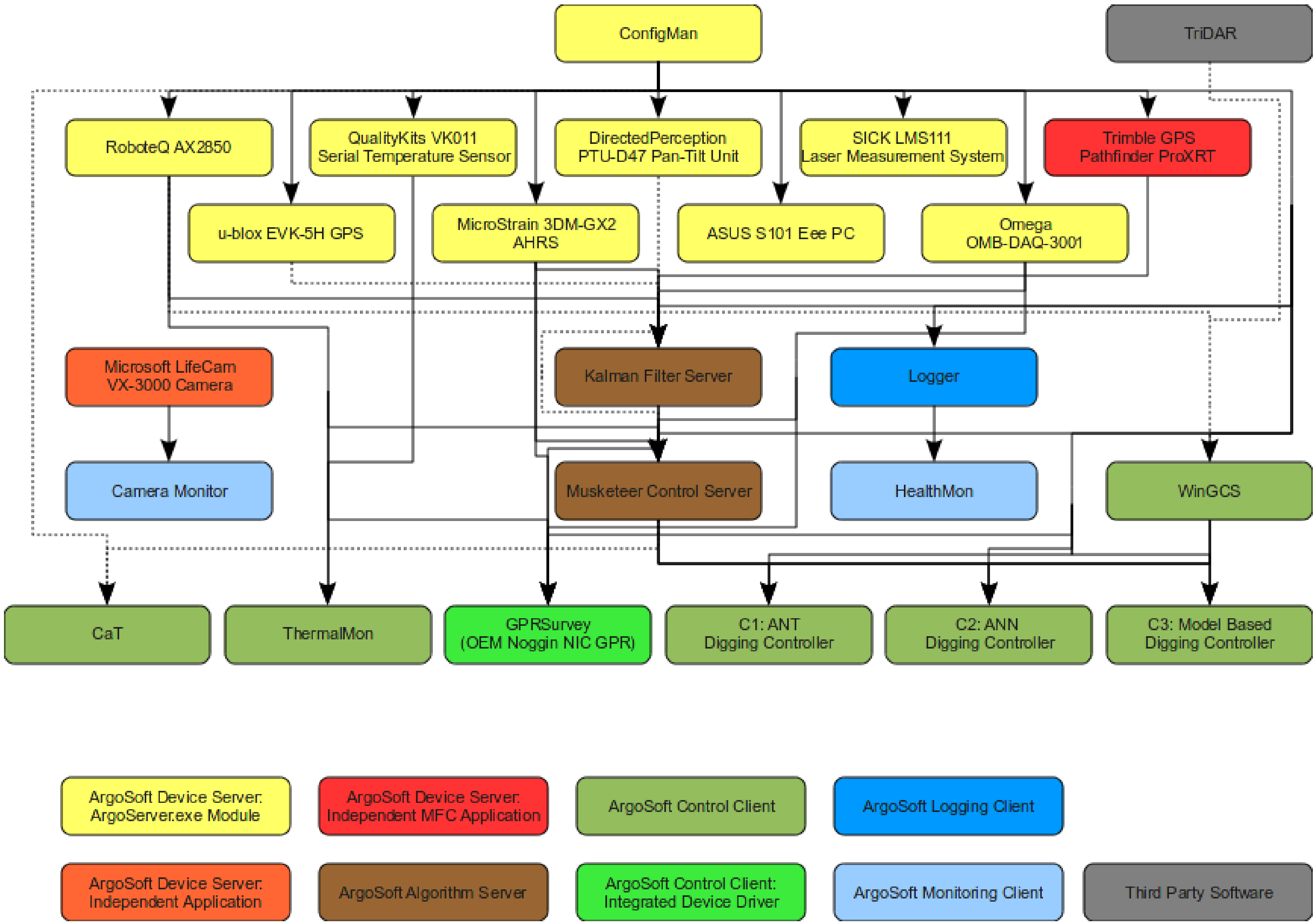


Figure 4.5: Software topology of the autonomous control system. Dashed lines denote optional connections. Optional connections were omitted from *CaT*, *Logger* and *WinGCS* wherever clarity permitted.

4.5.1 Device Servers

Device servers are low-level, driver-like programs that expose a device's functionality to connected clients. Like all servers in the ArgoSoft framework, they are also responsible for resolving priority when multiple clients issue conflicting commands. Every device server interfaces to only one piece of hardware and does not depend on any other software.

4.5.2 Algorithm Servers

Algorithm servers were designed to provide complicated algorithms that require temporal persistence across multiple connected clients. These algorithms cannot be implemented in device servers as they may require connection to more than one device or make control decisions. They cannot be implemented in the connection layer between clients and servers, as they are always running and need to be in the same state for all connected clients. Like all servers in the ArgoSoft framework, they are also responsible for resolving priority when multiple clients issue conflicting commands.

4.5.3 Control Clients

Control clients are software that may issue actuation commands to either low-level device servers or to mid-level algorithm servers. Care must be taken to guarantee that all actuation commands are issued with an appropriate command priority so that emergency-level stops, whether they are issued by the same control client or not, will behave properly.

4.5.4 Logging/Monitoring Clients

Logging or monitoring clients are software that attach to device or algorithm servers solely to process, store or present information. The clients may present warnings or in some other way draw attention to information when it hits various levels or states. Under no condition does a logging client issue an actuation command, operator intervention is required.

Chapter 5

Autonomous Control System Devices

The autonomous control system uses a large number of devices, some as intended by the manufacturer and others only after modification. Some devices were manufactured by UTIAS and project partners.

5.1 RoboteQ AX2850 Brushed DC Motor Controller

The RoboteQ AX2850 Brushed DC Motor Controller was installed as part of the locomotion system by NORCAT (§2.2). It is a dual-channel motor controller that uses FET controlled H-bridges to provide bidirectional control of DC motors. It can provide motors with instantaneous currents of up to 120 A for 30 seconds and sustained current of 60 A for over an hour [10]. It also provides closed-loop motor speed control using encoders that are driven by the motor shafts. It is installed into an accompanying circuit to provide it with electrical protection and to switch the controller input between an R/C receiver and a serial connection to a computer.

5.1.1 ArgoSoft Device Server (*roboteq*)

The *roboteq* ArgoServer provides clients with an ArgoSoft interface to the RoboteQ AX2850. Clients can control motors directly by specifying the starboard and port linear speeds or indirectly by specifying the rover linear and angular speed. Clients can receive encoder information as well as controller status information like supply voltage, controller temperature and applied current.

5.1.2 Successes

5.1.2.1 Closed-Loop Motor Speed Control

The RoboteQ AX2850 provides a closed-loop controller for rotational motor speed with no additional computational overhead to the autonomous control system. Appropriate rover parameters like gear ratio and wheel diameters allows the *roboteq* ArgoServer to provide a closed-loop linear speed controller to connected clients. Experimental observations show that the result is an adequate solution for the autonomous control system.

5.1.3 Problems Encountered

5.1.3.1 Stall Protection Without Event Notification

When operating under closed-loop speed control, the RoboteQ will stop the motors if they do not reach the targeted speed within a set period of time. This is meant to protect stalled motors from being damaged by excessive levels of current. The RoboteQ does not provide any notification to the connected computer of this event which can be confusing to autonomous control algorithms. A firmware update provided by RoboteQ Inc. exposed the necessary settings to disable the stall protection. It was instead implemented in the *MCS* (§6.2), giving the autonomy control system notification of stall events.

5.1.3.2 Runaway Conditions With No Input

As installed, the RoboteQ AX2850 was unstable. Some of the rovers were susceptible to driving away if an otherwise disconnected Ethernet cable was connected to the chassis RoboteQ serial port. Adapting standard operating procedures to assure that the RoboteQ serial port was always connected to a powered MIB (§5.4) regardless of the control mode restored a measure of confidence in the rover stability (§5.4.1.8). This behaviour remains unacceptable and may be symptomatic of underlying problems in the design of the accompanying RoboteQ circuitry (§5.1.3.6).

5.1.3.3 Driving Surges

For an analogous discussion on linear actuator surges, see §3.3.2.8.

5.1.3.4 Loss of Motor Power Under Radio Control (R/C)

One of the Musketeer rovers developed a loss of motor power under R/C. The rover lurches when commanded to perform manoeuvres requiring high levels of current, such as turning tightly or driving up slopes. The lurching is reminiscent of intermittent power being applied to the motors. Its correlation with high-power demands could suggest that brownouts are effecting a device in the control circuit. These problems did not develop until after a RoboteQ that had failed when switching between input modes (§5.1.3.5) was replaced; however, similar problems were reported on other chassis by NORCAT. The appearance of the problem with the installation of a new RoboteQ suggests that there could be an incompatibility between newer versions of RoboteQ hardware and the accompanying circuit. The problems only existed under R/C and were not detrimental to the performance of the rover under autonomous control.

5.1.3.5 RoboteQ Failure When Switching Between Control Input

The RoboteQ AX2850 is switched between R/C and serial control by an external circuit that changes the input and resets the RoboteQ. Under some conditions, this can reboot the RoboteQ while it is still under electrical load, a condition that RoboteQ Inc. states is dangerous to the controller. The operating procedure of the Musketeer rovers do not allow human operators inside the workspace while rovers are under autonomous control as the unpredictable nature of a malfunctioning controller creates unacceptable human risk. This leaves switching the RoboteQ control input as the sole method to stop a malfunctioning rover. Multiple RoboteQ controllers failed when operators were forced to stop rovers from damaging themselves or their surroundings by switching their control input. These failures would likely be avoidable if the switching circuit provided the electrical requirements specified by RoboteQ Inc. in their documentation (§5.1.3.6). Implementing these recommendations and making a minor design change would definitely allow for R/C RoboteQ shutdown under all conditions (§7.1.2).

5.1.3.6 Incomplete RoboteQ Electrical Protection

RoboteQ Inc. provides design guidelines for the circuitry that should be installed around the RoboteQ AX2850 (Figure 5.1) [10], [11]. These circuits are designed to protect the RoboteQ controller and prevent unstable electrical conditions that may damage other devices. They are necessary since DC motors can generate power if they are turned manually or by rover inertia through a process known as regeneration. This regenerated power can unexpectedly power the RoboteQ even if it is disconnected from the battery packs. This results in unpredictable conditions that can include runaway rovers and failed controllers. The circuit installed around the controller by NORCAT (Figure 5.2) does not include many of these recommendations:

- The RoboteQ power input ($V_{in}+$) is not fused from the battery mains. This can allow dangerous levels of current to reach the controller.
- The RoboteQ power input ($V_{in}+$) is allowed to float when powered off instead of being connected to ground. This allows regenerated power from the motors to build up in the

RoboteQ instead of safely flowing to ground.

- The control line (V_{ctrl}) is allowed to float when the rover is powered off instead of being connected to ground. This is specifically and strongly discouraged by both the design guidelines [11] and the user manual [10].

> Do not rely on cutting power to the controller for it to turn off if the power control (V_{ctrl}) is left floating. If motors are spinning because the robot is pushed or because of inertia, they will act as generators and will turn the controller on, possibly in an unsafe state. ALWAYS ground the power control wire (V_{ctrl}) to turn the controller off and keep it off.

- Motor outputs are not tied to ground. This is recommended to prevent low frequency oscillations but may not be necessary as newer hardware versions of the controller implement the protection internally [11].

5.1.4 Evaluation

Evaluation of the RoboteQ AX2850 motor controller is difficult to separate from an evaluation of its installation. The problems that clearly a result of the controller itself (§5.1.3.1) have been overcome, leaving only integration problems that remain incompletely identified. Experience with a RoboteQ AX2550 on previous generations of rovers as well as recommendations from colleagues reinforces a certain level of confidence in the RoboteQ AX2850.

The disregard of mandatory design guidelines (§5.1.3.6) erodes the confidence of UTIAS in the controller's installation. The majority of UTIAS RoboteQ failures were from switching control input under load (§5.1.3.5), a solvable problem (§7.1.2). It is unclear if other problems, such as runaway (§5.1.3.2) and unstable (§5.1.3.3) rovers as well as loss of motor power under R/C (§5.1.3.4), will also prove to be a result of the controller's current installation.

Continued use of the RoboteQ AX2850 is recommended for ongoing and future work only with a new circuit that follows the design guidelines laid out by RoboteQ Inc. (§7.1.2). Further investigation of the other serious problems (§5.1.3.2, §5.1.3.3, §5.1.3.4) should be undertaken only after installing the new circuit, as it is the opinion of UTIAS that it is possible that these problems would be solved by the proposed repair. If these problems are not addressed or continue to effect the rovers after all attempts to fix them have been exhausted, the RoboteQ AX2850 cannot be recommended for future work.

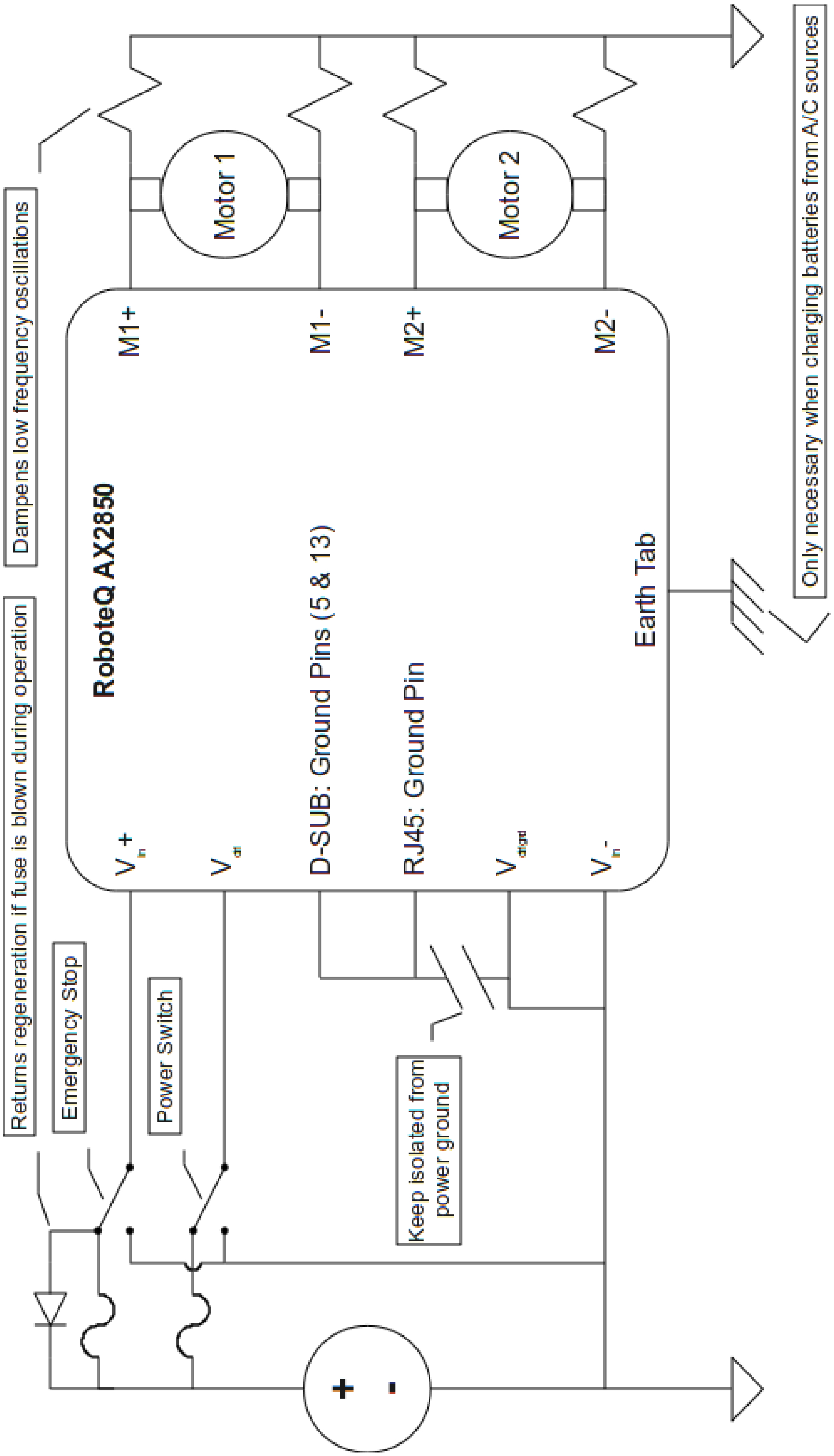


Figure 5.1: Simplified electrical schematic of the recommended RoboteQ AX2850 installation [10],[11].

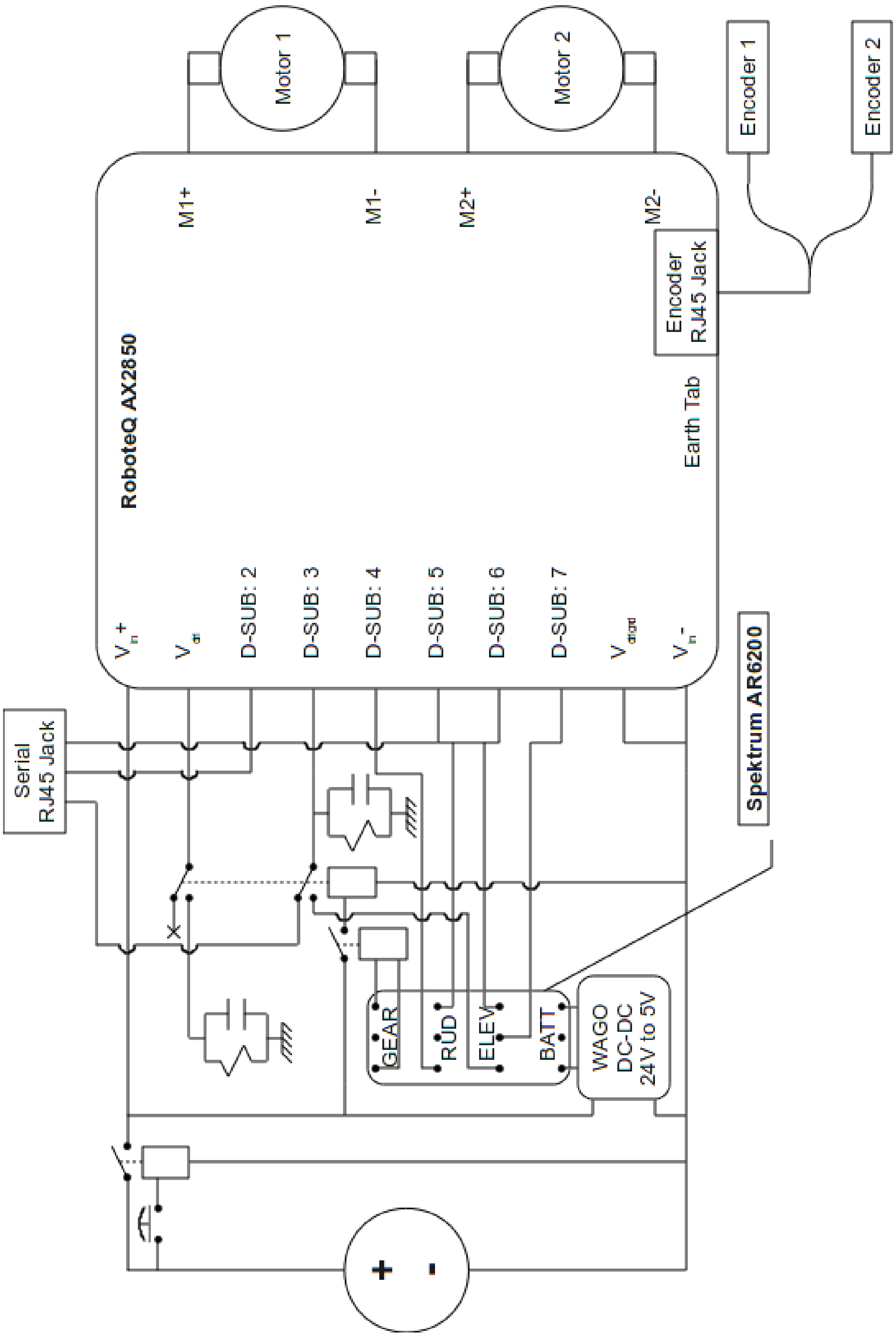


Figure 5.2: Simplified electrical schematic of the RoboteQ AX2850 installation in the Musketeer rovers [12].

5.2 Omega OMB-DAQ-3001 Data Acquisition Module

The Omega OMB-DAQ-3001 Data Acquisition Module was installed in the SCADA Box (§3.3) as part of the blade (§3.1) and GPR (§3.2) payloads. It is a USB analogue/digital Data Acquisition (DAQ) board with 4 analogue output channels, 16 analogue input channels and 24 digital input/output channels [13]. It was used in the SCADA Box to command the motor controllers for the linear actuators as well as read the load cells and actuator positions.

5.2.1 ArgoSoft Device Server (*scada-blade*, *scada-gpr*)

The *scada* ArgoServers provides access to the Omega OMB-DAQ-3001 abstracted as either a blade or GPR specific device. This allows clients to set specific actuators and read sensors without needing to track individual DAQ channels. Actuator position is handled by a closed-loop controller that runs as a thread in the device server allowing client software to continue to execute while the actuators move.

5.2.2 Successes

The Omega OMB-DAQ-3001 successfully functioned as a DAQ.

5.2.3 Problems Encountered

5.2.3.1 Insufficient Electrical Isolation Between Channels

The SCADA Box (§3.3) design (Figure 3.7) connects multiple independent circuits to the OMB-DAQ-3001 under the assumption that the Omega OMB-DAQ-3001 has sufficient electrical isolation to keep them separated. Investigation has shown that the OMB-DAQ-3001 is not an isolated device. Resistances between some channels and/or the device power can be as low as 10 kΩ depending on the channel type (digital or analogue) and function (input or output). These resistances are low enough to bridge independent circuits, such as those powered by Vicor Micro DC-DC converters as used in the SCADA Box. These bridges can cause voltage offsets and other electrical instabilities in the circuits (§3.3.2.4, §3.3.2.5, §3.3.2.6, §3.3.2.7). It is hypothesized that these may have been the source of the actuator surges observed when connecting the SCADA Box (§3.3.2.8).

5.2.3.2 Windows Support Only

Modern desktop operating systems are not designed for the requirements of autonomous mobile robotic controllers. Microsoft Windows is not a real-time operating system and contains unnecessary services that consume computing resources. While efforts can be made to customize and streamline specific Windows installations, a lot of these processes cannot be stopped or removed. The Omega OMB-DAQ-3001 only supports Windows and as a result, precludes the use of any other operating system. More suitable operating systems, like QNX or Linux, would have decreased the bottlenecks experienced by the ASUS Eee PC S101 (§5.10).

5.2.3.3 Computationally Expensive

During development, it was found that the Omega OMB-DAQ-3001 drivers took an abnormally large amount of processing power to communicate between the computer and the device as a result of the driver message layer being built on the .NET framework. This disproportionate use of processing power limits the speed of the blade controller and places an unnecessary load on the ASUS Eee PC S101 (§5.10).

5.2.4 Evaluation

It appears that the Omega OMB-DAQ-3001 would be an effective and reliable DAQ when used as intended. Its suitability for use in mobile robotics, however, is less clear. It appears to be simultaneously over- and under-specified for its intended use and its installation. Its 24 digital channels are excessive; however, its 4 analogue output channels leave no room for expansion and were only available on the top-of-the-line model. It provides limited electrical isolation (§5.2.3.1) despite being installed into a SCADA Box design (§3.3) that requires it. It also imposed design constraints on the autonomous control system by requiring Microsoft Windows (§5.2.3.2) and implementing a computationally expensive message layer inside its operating system driver (§5.2.3.3).

The continued use of the Omega OMB-DAQ-3001 can only be considered if the SCADA Box design is changed to accommodate its real-world electrical characteristics. If this resolves associated problems observed in the payloads (§3.1, §3.2), use of the OMB-DAQ-3001 for ongoing and future work is recommended as long as a proper understanding of its limitations is maintained. If such conditions cannot be met, then it is possible that the OMB-DAQ-3001 itself is incompatible with the avowed goals of the SCADA Box design and its continued use cannot be recommended. Given the availability of DAQ devices designed for mobile installations and the incompatibility of OMB-DAQ-3001 specifications with the project requirements its purchase would not have been recommended.

5.3 Sensors & Software OEM Noggin Network Interface Card (NIC) Ground Penetrating Radar (GPR)

The Sensors & Software OEM Noggin Network Interface Card (NIC) Ground Penetrating Radar (GPR) is a two-part device designed to scan terrain for subsurface features such as changes in soil composition, rocks and water. It consists of a GPR that passes over the ground emitting microwaves at 1000 MHz and measuring their reflections off buried objects. It is integrated into the GPR payload (§3.2) and communicates to a computer over RJ45 Ethernet using a NIC.

5.3.1 Successes

The OEM Noggin NIC GPR successfully functioned as a GPR.

5.3.2 Problems Encountered

5.3.2.1 Client Disconnection Freezes Network Card

It was discovered during development that after a connected computer disconnects from the OEM Noggin NIC GPR that the NIC must be rebooted before it will accept a new connection. Upon being informed of the problem, Senors & Software informed UTIAS that this is a known issue and that no fixes are expected [14]. While this problem did not effect the functionality of the GPR, it did further reduce the autonomy of the GPR surveys (§3.2.2.4).

5.3.2.2 Windows Support Only

For an analogous discussion on the effect of selecting the Microsoft-Windows-dependent OEM Noggin NIC GPR over the operating-system-independent serial OEM Noggin GPR, see §5.2.3.2

5.3.3 Evaluation

Interfacing the Sensors & Software OEM Noggin NIC GPR into the autonomous control system was ultimately successful despite limitations discovered during development (§5.3.2.1). Its dependence on Microsoft Windows (§5.3.2.2) is undesirable and frustrating considering the existence of the operating-system-independent serial OEM Noggin GPR that UTIAS had worked with before. The NIC version of the Noggin GPR was purchased as it Ethernet connection provides higher bandwidth than serial, however this was never required during development or deployment. Considering the monetary investment made, continued use of the OEM Noggin NIC GPR is none-the-less recommended for ongoing and future work.

5.4 Musketeer Interface Box (MIB)

The Musketeer Interface Box (MIB) provides the communication connection between the ASUS S101 Eee PC (§5.10) in the Musketeer Control Box (MCB) (§5.9) and the autonomous control devices distributed around the chassis. The MIB provides regulated power to the MCB and other devices and houses communication hubs and converters that allow the Eee PC to interface to the rest of the autonomous control system. Efficiency is improved by centralising power supplies and communication hubs.

The MIB presents an External User Interface (EUI) on the status of the rover. A latching, illuminated Single-Pole, Single-Throw (SPST) pushbutton controls the power to the MIB. The pushbutton is only illuminated if the MIB is properly connected to the battery packs. When the MIB is turned on, a green LED on the lid is illuminated. The lid also has 2 rows of 9 LEDs functioning as bar graphs. One displays the internal MIB temperature and the other the battery pack voltage.

The power distribution inside the MIB consists of 3 regulated power supplies built around the Vicor Micro DC-DC converter. The available voltages are 5 V, 8 V and 12 V. Each DC-DC converter has fused inputs and outputs to protect itself and the devices it powers from current spikes. Each converter circuit can be manually switched on or off and its status is presented by a board-mounted LED.

All wired devices in the autonomous control package communicate in one of three ways: Ethernet, Serial or USB. The MIB is responsible for converting the disparate communication protocols of its devices to a single RJ45 Ethernet interface to the MCB. The MIB uses three devices to achieve this: a Cisco SD208 Ethernet Switch (§5.5); a Perle IOLAN SDS4 Device Server serial converter (§5.6); and a Lantronix UBox 2100 USB host (§5.7).

5.4.1 Successes

5.4.1.1 Cisco SD208 Ethernet Switch

For an in-depth evaluation of the Cisco SD208 Ethernet Switch, see §5.5.

5.4.1.2 Perle IOLAN SDS4 Device Server

For an in-depth evaluation of the Perle IOLAN SDS4 Device Server, see §5.6.

5.4.1.3 Lantronix UBox 2100

For an in-depth evaluation of the Lantronix UBox 2100, see §5.7.

5.4.1.4 Dust and Moisture Protection

The MIB was constructed by modifying a Pelican 1450 case that was a IP-67 rated against dirt and water [15] to include sealed power and interface connections and an EUI. The final MIB remained sealed to the external environment and was successful in protecting the electronics inside from the harsh operating conditions of development and the deployment. The MIB was used successfully in various temperatures, significant wind and dust, as well as rain and snow.

5.4.1.5 Single Power Solution for the Autonomous Control System

The devices that constitute the autonomous control system are distributed around the entire rover. If each device was powered independently, it would be difficult to assure that the entire autonomous control system was in the same power state during startup or shutdown. Centralising the power to the MIB provides operators a singular power control for the entire system and an easy way to shutdown or reboot it when necessary.

5.4.1.6 Single Communication Point for the Musketeer Control Box

The variety of communication protocols used by the autonomous control devices and their distribution around the rover makes it impossible to connect them all directly to the ASUS S101 Eee PC (§5.10) in the MCB (§5.9). The MIB successfully provided a single interface to the autonomous control system simplifying assembly and operation.

5.4.1.7 Immediate Battery Pack Level Display

Battery discharge is nonlinear and voltage can drop suddenly after a long period of relative stability. Continuing to use the rover when battery levels are low can be damaging to devices and the battery packs themselves. Heavy loads, such as digging or strong accelerations, result in temporary voltage drops that can cause devices to brownout or otherwise behave erratically. The MIB provides an immediately viewable measure of the battery pack voltage via its 9 LED status bar. This was helpful for troubleshooting when battery packs were low.

5.4.1.8 Clamping of Floating RoboteQ Input

The runaway conditions experienced by the RoboteQ AX2850 (§5.1.3.2) required that that the chassis serial port be kept connected to a powered device at all times. Leaving the MIB connected and in the on-position guaranteed that the serial input was clamped whenever the rover was powered, even when driving under R/C. The MIB thus provided a simple and convenient fix to a dangerous problem.

5.4.2 Problems Encountered

5.4.2.1 Cisco SD208 Ethernet Switch

For an in-depth evaluation of the Cisco SD208 Ethernet Switch, see §5.5.

5.4.2.2 Perle IOLAN SDS4 Device Server

For an in-depth evaluation of the Perle IOLAN SDS4 Device Server, see §5.6.

5.4.2.3 Lantronix UBox 2100

For an in-depth evaluation of the Lantronix UBox 2100, see §5.7.

5.4.2.4 Incorrect DC-DC Converter Debounce Circuit

The Vicor Micro DC-DC converters have a control pin that switches them on and off. The MIB design uses this pin to allow operators to control specific power channels through the use of an on-board switch. The debounce circuit attached to the switch used incorrect capacitance values [8]. This resulted in damage to the control circuitry of the Vicor DC-DC converters that necessitated the complete removal of the debounce circuit to allow for continued operation of the MIB.

5.4.2.5 Voltage Offsets and Oscillations

As was the case with the design of the SCADA Box (§3.3), the MIB design initially used incorrect assumptions about the nature of the Vicor Micro DC-DC converters. The initial design did not include the required electrical filtering or account for their electrical isolation. This resulted in voltage offsets and oscillations. The design was amended to include the filtering components and to connect the output grounds together to remove any voltage offsets.

5.4.2.6 Nonlinear Status Bar LEDs

The LED status bars were created from 9 independent high-intensity LEDs triggered by 9 independent circuits. The circuit design uses Zener diodes to trigger the individual LEDs when the signal being measured surpassed the reference value. The tolerances of the Zener diodes could be larger than the spacing between successive lights, meaning that individual LEDs of the status bar could be illuminated out-of-order. While this decreased the apparent quality of the design, the LEDs still provided a readily viewable, user gauge of rover power and MIB temperature.

5.4.2.7 Parasitic Powering of Devices

Individual devices connected to the MIB are of unknown electrical characteristics. Some may contain capacitors or otherwise maintain a charge after being powered down; putting electricity back out on to the supply circuit. While individually fused, all devices sharing a power supply were electrically common. It was specifically observed that the QualityKits VK011 Serial Temperature Sensor (§5.17) could remain powered for a short time after the MIB was switched off and it is hypothesized that this is because of power coming from other devices on the bus.

5.4.3 Evaluation

The Musketeer Interface Box (MIB) was designed to provide power to the autonomous control system (§4.2) and communication between the MCB and the various distributed devices (§4.3). It did so from a case that despite modifications, remained sealed to the environment (§5.4.1.4). It also provided an accessible External User Interface (EUI) that allowed field operators to monitor the rover autonomous control system (§5.4.1.7). While there were mistakes in design and development (§5.4.2.4, §5.4.2.5, §5.4.2.6, §5.4.2.7), the MIB remains a integral part of the autonomous control system. Continued use and development of the Musketeer Interface Box is recommended for ongoing and future work.

5.5 Cisco SD208 Ethernet Switch

The Cisco SD208 functioned as a standard 8-port Ethernet switch, allowing the single Ethernet jack on the ASUS S101 Eee PC (§5.10) to communicate to a variety of Ethernet devices, including the SICK LMS111 Laser Measurement System (§5.19), Sensors & Software OEM Noggin NIC GPR (§5.3), the Perle IOLAN SDS4 Device Server (§5.6) and the Lantronix UBox 2100 (§5.7). The SD208 was installed into the MIB (§5.4).

5.5.1 Successes

The Cisco SD208 worked flawlessly as an Ethernet switch.

5.5.2 Problems Encountered

The Cisco SD208 worked flawlessly as an Ethernet switch.

5.5.3 Evaluation

The Cisco SD208 was a clear success. Unless the number of Ethernet devices surpasses the number of available ports, there is no reason to reconsider the device. Continued use of the Cisco SD208 is highly recommended for ongoing and future work.

5.6 Perle IOLAN SDS4 Device Server

The Perle IOLAN SDS4 Device Server provides 4 RJ45 terminated serial ports to a connected computer over RJ45 Ethernet. It was designed to be used in the autonomous control system for the RoboteQ AX2850 (§5.1), the MicroStrain 3DM-GX2 AHRS (§5.14), the DirectedPerception PTU-D47 Pan-Tilt Unit (PTU) (§5.18) and the QualityKits VK011 Serial Temperature Sensor (§5.17). During development it was discovered that it could not be used for the RoboteQ AX2850 as a result of dropped characters (§5.6.2.2). The Perle IOLAN is installed into the MIB (§5.4) where it connects to the Cisco SD208 Ethernet Switch (§5.5).

5.6.1 Successes

The Perle IOLAN reliably provided 8-bit serial ports over Ethernet.

5.6.2 Problems Encountered

5.6.2.1 Busy Serial Ports

Occasionally the Perle software on the ASUS S101 Eee PC (§5.10) would believe that the serial ports were busy and unavailable for a new connection. This predominantly happened after an abnormal or incomplete autonomous control system shutdown. This problem was easily fixed with associated Perle software that provided a method to reset all serial ports. This solution was found to be a dependable fix to the problem.

5.6.2.2 Dropped Characters Communicating to the RoboteQ

The RoboteQ AX2850 (§5.1) is the only autonomous control system device that requires a 7-data-bit, even-parity serial port. It was discovered during development that communication to it through the Perle IOLAN was failing. Investigation showed that commands sent via the Perle IOLAN were intermittently losing characters somewhere between the ASUS S101 Eee PC's Ethernet port and their reception at the RoboteQ. This meant that the RoboteQ was not reliably receiving complete or valid commands and would not respond. Investigations with Perle Systems' tech-support did not find a solution but were not concluded due to time constraints.

The problem was circumvented by connecting the RoboteQ to the Lantronix UBox 2100 (§5.7) via a USB Serial Converter (§5.8).

5.6.2.3 Possible Critical System Error

During development and deployment, it was discovered that irregular critical system errors that were crashing Microsoft Windows were the result of the driver for either the Perle IOLAN SDS4 or the Lantronix UBox 2100 (§5.7). These critical system errors immediate halt the ASUS S101 Eee PC (§5.10) and require the entire autonomous control system to be rebooted. They were more likely to occur following an emergency stop of the SCADA Box, but could also occur without warning. Their intermittent nature makes investigation tedious, and the exact cause has not yet been identified. It is possible that once it is, firmware or driver updates may resolve the issue.

5.6.3 Evaluation

The final evaluation of the Perle IOLAN SDS4 Device Server will depend on the source of the critical system error (§5.6.2.3). If it is found that it is a result of the Perle IOLAN, the following evaluation will be invalid.

The apparent inability of the Perle IOLAN SDS4 to handle 7-data-bit, even-parity serial devices (§5.6.2.2) was problematic, but has now been identified and circumvented. Perle Systems had been open to investigating the problem with UTIAS until time constraints necessitated moving on with alternative solutions. Within this known limitation, the Perle IOLAN proved to be robust, reliable and easy-to-work-with in the field during development and the Mauna Kea deployment. Minor problems with serial port conflicts were readily and reliably fixed using the included software (§5.6.2.1). Maintaining consideration of its limitations and the uncertain source of the critical system error, the continued use of the Perle IOLAN SDS4 Device Server is recommended for ongoing and future work.

5.7 Lantronix UBox 2100

The Lantronix UBox 2100 provides two USB ports to a connected computer over RJ45 Ethernet. It was selected by NORCAT to be used in the autonomous control system for the Omega OMB-DAQ-3001 (§5.2). As a result of problems uncovered with the Perle (§5.6), it was also used for the RoboteQ AX2850 (§5.1). It is installed into the MIB (§5.4) where it connects to the Cisco SD208 Ethernet Switch (§5.5).

5.7.1 Successes

The Lantronix UBox 2100 successfully provided two USB ports over Ethernet.

5.7.2 Problems Encountered

5.7.2.1 Frozen Hardware

When trying to establish new connections, the Lantronix UBox could occasionally become frozen and unresponsive, no longer serving its devices or responding to network queries. There was no evidence of these problems once a connection was made to the device and the rover was operational. A repeatable but nontrivial fix was found. Using the included software and the device serial number, the UBox can be reset to factory defaults. Once the device was reconfigured to work on the autonomous control system network, it would return to functioning normally. This fix was time consuming and difficult to undertake in the field.

5.7.2.2 Lack of Availability

While care is taken to protect electronics in sealed enclosures like the MIB (§5.4), it is best practise to maintain replacement parts for critical devices. This was the case for all electronic devices used in the autonomous control system except for the Lantronix UBox. The UBox is no longer produced and is unavailable from its manufacturer [16]. This leaves the autonomous control system without a replacement for a critical part of the interface topology. If a UBox were to fail, it would require the redesign of the effected rover's infrastructure.

5.7.2.3 Possible Critical System Error

For a discussion on the unidentified critical system error that may be caused by the Lantronix UBox 2100, see §5.6.2.3.

5.7.2.4 Windows Support Only

For an analogous discussion on the effect of the Lantronix UBox 2100's Microsoft Windows operating system requirement, see §5.2.3.2.

5.7.3 Evaluation

The final evaluation of the Lantronix UBox 2100 will depend on the source of the critical system error (§5.7.2.3). If it is found that the critical system error is a result of the UBox, the following evaluation will be invalid.

The UBox worked well when operational but did suffer from unpredictable failures during initial connection that were nontrivial and time-consuming to fix in the field (§5.7.2.1). When considered in addition to the planned obsolescence by the manufacturer (§5.7.2.2) and its Microsoft Windows requirement (§5.7.2.4), continued longterm use of the Lantronix UBox appears ill-advised. If the MIB (§5.4) were redesigned to use a USB connection to the MCB instead of Ethernet (§7.1.6), the UBox can be replaced with a cheaper, more reliable USB Hub (§5.12). Continued use of the Lantronix UBox 2100 is only be recommended for ongoing work while long-term alternatives are investigated.

5.8 USB Serial Converter

The USB Serial Converter is a USB to RS-232 serial dongle that was used to connect the RoboteQ AX2850 (§5.1) to the ASUS S101 Eee PC (§5.10) through the Lantronix UBox 2100 (§5.7). This solution became necessary when it was found that the Perle IOLAN SDS4 Device Server (§5.6) could not handle the 7-data-bit serial (§5.6.2.2) used by the RoboteQ. The USB Serial Converter was located inside the MIB (§5.4).

5.8.1 Successes

The USB Serial Converter reliably provided a serial port over USB to the connect computer.

5.8.2 Problems Encountered

5.8.2.1 Limited Buffer Size

USB devices cannot provide the timing and throughput requirements of real serial ports. They attempt to emulate this by buffers that discretize the communication between the device and computer. For small, well defined messages this works well. It becomes problematic when devices output a continuous stream of data or when long messages need to be transmitted to the device. The USB Serial Converter could not be used to update the firmware on the RoboteQ AX2850 (§5.1) and would probably not be suitable for devices that stream data. This did not appear to be an issue as used during deployment.

5.8.3 Evaluation

The use of the USB Serial Converter in the autonomous control system was unplanned. It was introduced to address incompatibilities with the Perle IOLAN SDS4 Device Server (§5.6) and the 7-data-bit serial (§5.6.2.2) from the RoboteQ AX2850 (§5.1). In this relief role, the USB Serial Converter was successful even if not ideal (§5.8.2.1). Continued use of the USB

Serial Converter is recommended for connection to the RoboteQ for ongoing and future work however any further implementation into the autonomous control system is not recommended. In the future, the purchase of a higher quality device like the Keyspan USA-19HS USB Serial Adapter would be recommended instead of the USB Serial Converter as they provide more robust functionality and a higher-level of end-user support.

5.9 Musketeer Control Box (MCB)

The Musketeer Control Box (MCB) houses the ASUS S101 Eee PC (§5.10), the Hawking Wireless Booster (§5.11) and the u-blox EVK-5H GPS (§5.15). It connects externally to the MIB (§5.4) for power and connection to the remainder of the autonomous control system devices over Ethernet. It also connects directly to an external USB hub which is used to connect the two Microsoft LifeCam VX-3000 cameras (§5.20) and when required a Logitech Dual Action Gamepad (§5.13).

The MCB presents the operator with an EUI to the Eee PC. This reduces the need to open the MCB to access it and protects the computer from damaging dirt and moisture during field operation. The interface consists of two SPST illuminated momentary pushbuttons, a green LED indicating PC status, a red LED indicating screen status, a blue LED indicating drive access and a 9 LED bar graph presenting MCB temperature.

A blue pushbutton controls the Eee PC lid sensor and a green pushbutton operates the power switch. The pushbuttons follow the design choices of the MIB and only illuminate when they will have an effect on the Eee PC. The two pushbuttons are necessary as a result of the design of the Eee PC which only allows its power switch to function when the laptop lid is open. When the lid pushbutton is activated, the laptop lid is virtually open and the green pushbutton illuminates. The operator can then turn on the Eee PC by pressing the power pushbutton while continuing to hold the lid pushbutton. Both pushbuttons can then be immediately released. This design also helps prevent accidental power cycles. While the illumination does not function without external power, the pushbuttons will and can still be used when Eee PC is running on its battery.

The modification of the lid sensor from the Eee PC means that as long as the netbook is connected to the MCB, the laptop screen will remain extinguished even when the lid is open. To facilitate operating the Eee PC in the field, a Double-Pole, Single-Throw (DPST) switch is mounted internally that can also control the lid sensor. When the switch is activated, the lid is virtually open and the lid pushbutton is extinguished as it no longer has an effect. Whenever the lid is virtually open, the red external screen status LED is illuminated to warn the operator that the screen is on. Leaving the screen on consumes more energy and contributes to temperature buildup inside the MCB. The power and drive LEDs mimic the function of their counterparts on the Eee PC.

5.9.1 Successes

5.9.1.1 ASUS S101 Eee PC

For an in-depth evaluation of the ASUS S101 Eee PC, see §5.10.

5.9.1.2 Hawking Wireless System

For an in-depth evaluation of the Hawking Wireless System, see §5.11.

5.9.1.3 u-blox EVK-5H Global Positioning System (GPS)

For an in-depth evaluation of the u-blox EVK-5H, see §5.15.

5.9.1.4 Dust and Moisture Protection

For an analogous discussion on the protection provided by the MCB, see §5.4.1.4.

5.9.1.5 Temperature Control

Operating the ASUS S101 Eee PC (§5.10) inside the sealed MCB can result in increased temperatures. If the temperature of the Eee PC rises too high, it will shutdown and may be damaged. To facilitate heat transfer without compromising the environmental seal, the MCB was outfitted with an externally-finned heatsink manufactured by NORCAT. The heatsink proved to work well in a variety of field conditions, from high temperatures and strong sunlight to low temperatures and clear night skies. Throughout, the Eee PC remained at safe operating temperature as recorded by its internal temperature sensor and the MCB temperature bar graph.

5.9.1.6 External User Interface to ASUS S101 Eee PC

The environmental protection provided by the MCB would be greatly reduced if it had to be repeatably opened to access the Eee PC. The EUI was designed to reduce the need by allowing the operator to start, restart or shutdown the Eee PC without opening the MCB. The LEDs of the EUI, especially the drive access LED, allow the operator to diagnose and recognize system problems, such as the Critical System Error caused by either the Perle IOLAN SDS4 or the

Lantronix UBox 2100 (§5.6.2.3). These problems could then be dealt with via the external power controls. The double-switch design of the power controls prevented accidental power cycles.

5.9.2 Problems Encountered

5.9.2.1 ASUS S101 Eee PC

For an in-depth evaluation of the ASUS S101 Eee PC, see §5.10.

5.9.2.2 Hawking Wireless System

For an in-depth evaluation of the Hawking Wireless System, see §5.11.

5.9.2.3 u-blox EVK-5H Global Positioning System (GPS)

For an in-depth evaluation of the u-blox EVK-5H, see §5.15.

5.9.2.4 Nonlinear Status Bar LEDs

For an analogous discussion on the LED problems experienced with the MCB, see §5.4.2.6.

5.9.2.5 Dimly Lit Pushbuttons

Wherever possible the EUI used high-intensity LEDs that could be viewed in strong, direct sunlight. The illuminated pushbuttons proved to be inadequate for use in such conditions, making it hard to recognize when they were illuminated. The green pushbutton used for the power control was especially dim.

5.9.3 Evaluation

The Musketeer Control Box (MCB) was successful in providing a sealed environment (§5.9.1.4) to the ASUS S101 Eee PC (§5.10) that was still capable of dissipating the heat generated by the computer (§5.9.1.5). Its External User Interface (EUI) was successful in reducing the need to open the case and a time-saving debugging tool (§5.9.1.6). While there were mistakes in design and development (§5.9.2.4, §5.9.2.5), the MCB remains a integral part of the autonomous control system. Continued use and development of the Musketeer Control Box is recommended for ongoing and future work.

5.10 ASUS S101 Eee PC

The ASUS S101 Eee PC is a netbook equipped with an 1.6 GHz Intel Atom N270 processor, 1 GB of RAM, a 16 GB Solid-State Drive (SSD) and a 16 GB Secure Digital (SD) flash card. It has 3 USB ports, 1 Ethernet port, a flash card slot, an internal 802.11b/g wireless card and an internal Bluetooth card.

The Eee PC is housed in the MCB (§5.9) and is the centre of the autonomous control system. It runs the various algorithms and controllers that control the sensors and actuators that make up the autonomous control system. As a requirement of some of the autonomous control devices (§7.2.7), it runs Microsoft Windows XP.

5.10.1 ArgoSoft Device Server (*PCMon*)

The status of the Eee PC is monitored by *PCMon*, an ArgoServer that monitors the temperature and power status of a computer. This is accomplished by using the Advanced Configuration and Power Interface (ACPI) open standard for operating-system-level device communication. *PCMon* works on both Linux and Windows machines and can return the temperature of a specified sensor, the charge level and rate of a battery as well as information about connected power adaptors and batteries.

5.10.2 Successes

5.10.2.1 Robustness

The operating conditions experienced during development and the Mauna Kea deployment were well outside standard levels of moisture, debris, temperature, shock and restricted ventilation. While this was a concern for all electronic devices, no device was more important to the autonomous control system than the ASUS S101 Eee PC. The Eee PC demonstrated a level of robustness that far exceeded realistic expectations, rivalling performance expected from toughbook-class laptops at a fraction of the cost. The Eee PC was even opened and modified, having wires soldered directly to the motherboard (§5.10.2.3) for integration with the MCB (§5.9). Only one Eee PC out of five has failed to date and it was an original unit that had been repeatedly subjected to environmental testing, including conditions that were increased until Eee PC failure was observed.

5.10.2.2 Accessibility

Using a netbook provided operators access to the autonomous control system regardless of location and facilities. This aided development and deployment work by providing a quick method for operators to troubleshoot hardware and software without the need for other computers or network connections. This was used extensively in the back lots of UTIAS during development to speed up design iterations.

5.10.2.3 Modifiability

An embedded controller on an autonomous mobile robotic platform is presumably not a targeted application of the Eee PC. In order to protect it from the elements it was operated inside the sealed MCB (§5.9), which created the need for a way to control it externally. The design of the Eee PC made it easy to be opened and modified. Connections to external controls for its lid sensor and power switch as well as external LEDS were successfully added to the motherboard.

5.10.3 Problems Encountered

5.10.3.1 Slow Solid-State Drive (SSD) Throughput

The ASUS S101 Eee PC was selected because it uses a SSD instead of a traditional Hard Disk Drive (HDD). The high-speed rotating platters and moving read/write heads of a HDD make it susceptible to damage from vibration. Given its installation on a mobile, autonomous platform, the Eee PC is guaranteed to be subjected to a number of nontrivial shocks, making the lack of moving parts in a SSD particularly attractive. Unfortunately, SSDs can have slower throughput speeds than HDDs, especially when writing contiguous data to disk. This limitation was observed during development and deployment. Modifications to the autonomous control system architecture and software design should be able to mitigate the effects of these limitations (§7.2.6).

5.10.3.2 Weak Bluetooth

Bluetooth was used to communicate to the Trimble GPS Pathfinder ProXRT (§5.16) so that the unsealed proprietary Trimble serial dongle did not have to be used. The Eee PC used on *Porthos* however appeared to have a faulty or weak Bluetooth adaptor that made consistent connection with the Pathfinder receiver impossible. The Pathfinder receiver had to be connected via serial to the Eee PC via a USB Serial Converter (§5.8). In order to assure smooth operation on all rovers, it may be necessary to upgrade the integrated Bluetooth adaptors.

5.10.3.3 Limited Interface Options

Autonomous mobile robotic platforms require a large number of devices to be connected to the central computer. The devices frequently use disparate interface standards such as DE-9 serial, RJ45 Ethernet and USB. Converting them to a common, single interface can be complicated and difficult, as converters are frequently problematic when used on nonstandard implementations. This was accomplished in the autonomous control system by the MIB (§5.4). It would have been preferred to have a computer with more and different interface connections built-in; however, given the options found on modern laptops, this was not realistic.

5.10.4 Evaluation

The ASUS S101 Eee PC was selected as the centre of the autonomous control system because it was a netbook (§5.10.2.2) that used a SSD. Its robustness (§5.10.2.1) and modifiability (§5.10.2.3) quickly made it an indispensable part of the autonomous control system. While it did prove to have some computation limitations (§5.10.3.1), it was adequate for running the autonomous control algorithms, logging data, compiling source code and supporting remote desktop connections. With consideration of its limitations in mind, continued use of the Eee PC is highly recommended for ongoing and future work.

5.11 Hawking Wireless System

The Hawking Wireless System consisted of the HWUG1, an external USB wireless card with a removable antenna, and the HSB2, a powered wireless booster. The output from the system can be configured to either 100 mW, 200 mW or 500 mW enabling the ASUS S101 Eee PC (§5.10) to communicate wirelessly to distant wireless access points. This network connection provides the autonomous control system with a data connection to the ground control station and operators with a connection to the rover. The Hawking Wireless System is installed in the MCB (§5.9).

5.11.1 Successes

5.11.1.1 Range

The range of a wireless system depends on its output power as well as the location and gain of its antenna. Given clear line-of-sight, the Hawking Wireless System was strong enough that even the stock antennas could communicate over long distances. This enabled the rovers to be operated safely away from other equipment or personnel as well as provide remote operation for the Mauna Kea analogue deployment.

5.11.2 Problems Encountered

5.11.2.1 Network-Induced Driver Lock

The HWUG1 wireless card or its driver could not gracefully handle communication failures with the Mauna Kea access point. When external events knocked the rovers off the network, they would occasionally not reconnect until the network device driver had been restarted by disabling and reenabling it. This had to be done manually in the field by opening up the MCB (§5.9) and operating the ASUS S101 Eee PC (§5.10) directly. No other solution was available as the computer was no longer accessible on the network and rebooting it did not fix the problem. This failure proved to be unpredictable and disruptive to ongoing experiments as well as time consuming to fix. It had not been observed during development.

5.11.2.2 Antenna Interference

Wireless signals are significantly attenuated by solid materials. This increased attenuation decreases their range and can prevent network connectivity between wireless devices that would otherwise be able to communicate. The wireless antennas were mounted directly on the MCB which placed them below numerous features of the rover, such as the Trimble GPS Pathfinder ProXRT (§5.16) antenna mast and the neck holding the DirectedPerception PTU-D47 (§5.18) and its associated sensors. The loss of line-of-sight to the Mauna Kea access point could result in the loss of network connectivity to the rover. The rover would have to be rotated in order to regain a connection.

5.11.3 Evaluation

During development, the Hawking Wireless System was used to create an ad-hoc network between available rovers and other computer equipment. During the Mauna Kea deployment, the Wireless System was used to communicate to an access point. While the connection to an access point seemed to cause some problems with the HWUG1 wireless card driver (§5.11.2.1), the HSB2 Hi-Gain Wireless Signal Booster enabled communication over long distances (§5.11.1.1). The operable distance and reliability could be extended further by relocating the antenna to a higher location on the rover (§5.11.2.2), a modification that is highly recommended. Continued use of the HSB2 Hi-Gain Wireless Signal Booster and the HWUG1 wireless card as a Wireless System is highly recommended for ongoing and future work.

5.12 Belkin 7-Port Mobile USB Hub

The Belkin 7-Port Mobile USB functioned as a standard USB hub outside the MCB (§5.9). It allowed a single USB port on the ASUS S101 Eee PC (§5.10) to communicate to multiple USB devices outside. It was used with the Microsoft LifeCam VX-3000 Cameras (§5.20) and when necessary, the Logitech Dual Action Gamepad (§5.13). It was mounted directly on the rover chassis.

5.12.1 Successes

5.12.1.1 Multiple Cameras Simultaneously

USB ports provide up to 500 mA of current to connected devices. When using a unpowered USB hub, care must be taken to assure that the total current needed by the devices does not exceed this limit. If it does, unexpected and unpredictable behaviour can occur. The powered Belkin 7-Port Mobile Hub was the only hub that was found to be capable of simultaneously streaming two Microsoft LifeCam VX-3000 Cameras (§5.20). It is unclear if this is because the combined current draw of the two cameras is above 500 mA or if the powered hub is better at maintaining the throughput and latency expected by the cameras.

5.12.2 Problems Encountered

5.12.2.1 Not Rated for External Use

Devices like cameras that require high bandwidth and/or low latency need to be connected as directly as possible to the host computer. Since the ASUS S101 Eee PC is located inside the sealed MCB (§5.9), this requires individual sealed USB ports for each device. Time and resource conflicts did not allow this, so the Belkin 7-port Mobile USB Hub was mounted directly on the rover chassis so only one port had to be added to the MCB for up to 7 devices. The Belkin hub is not weatherized or otherwise designed to be used outdoors. While this was acceptable in the dry conditions of the Mauna Kea deployment it cannot be used as a long term solution.

5.12.3 Evaluation

The Belkin 7-Port Mobile USB Hub functioned well to provide extra external USB ports for the Microsoft LifeCam VX-3000 Cameras (§5.20) and other occasional USB devices for the duration of the Mauna Kea deployment. As the hub is not sealed or rated for external use, this installation cannot be continued indefinitely (§5.12.2.1). Continued use of the Belkin 7-Port Mobile USB hub is recommended for ongoing and future work within a proper enclosure.

5.13 Logitech Dual Action Gamepad

The Logitech Dual Action Gamepad functioned as a Human Interface Device (HID) to the rover through the *Calibration and Testing (CaT)* client (§6.4). Operators could directly position the rover and, if attached, the blade or GPR payloads (§3.1, §3.2). Given the lack of payload control under R/C and the loss of power problems (§5.1.3.4) experiences on some rovers, this functionality proved essential. It was connected to the Belkin 7-Port Mobile USB Hub (§5.12) when required.

5.13.1 Successes

The Logitech Dual Action Gamepad functioned well as an HID to the rover and payloads.

5.13.2 Problems Encountered

There were no specific problems with the Logitech Dual Action Gamepad.

5.13.3 Evaluation

Having a Human Interface Device (HID) to the rover and payloads proved invaluable during development and deployment for both experimentation and dealing with problems. This functionality became necessary when problems with power under R/C (§5.1.3.4) meant some rovers could only be driven under serial control. The continued use of the Logitech Dual Action Gamepad is recommended for ongoing and future work.

5.14 MicroStrain 3DM-GX2 Attitude Heading Reference System (AHRS)

The MicroStrain 3DM-GX2 is an Attitude Heading Reference System (AHRS) as well as an Inertial Measurement System (IMU). It measures all six linear and angular accelerations using accelerometers and gyroscopes, and uses a magnetometer to measure heading relative to magnetic north. It can output linear and angular accelerations, angular rates, Euler angles and rotation matrices. It receives power and a RS-232 serial connection over a proprietary connector. It is used by the autonomous control system to provide angular position of the rover about all three axes. It is mounted in a protected box on the Trimble GPS Pathfinder ProXRT (§5.16) sensor mast and communicates to the autonomous control system through the Perle IOLAN SDS4 Device Server (§5.6).

5.14.1 ArgoSoft Device Server (*imu*)

The MicroStrain 3DM-GX2 AHRS is served by *imu*, an ArgoServer that provides clients access to its sensor values. This allows users to get the linear and angular accelerations as well as the orientation of the rover and the Euler angles. The ArgoServer also provides access to more esoteric information about the 3DM-GX2 such as the magnetometer reading measured in teslas and the temperature of the device.

5.14.2 Successes

The MicroStrain 3DM-GX2 AHRS successfully provided heading readings as well as linear and angular accelerations.

5.14.3 Problems Encountered

5.14.3.1 Angular Offsets

Whether as a result of mounting tolerances, differences in each rover or measurement accuracy, each individual MicroStrain 3DM-GX2 AHRS measured different heading information on each rover. Values could vary by as much as 10°. This lack of repeatability is a fundamental challenge of multiagent robotics as it reduces the task interchangeability of rovers.

5.14.3.2 Slow Update Rate

The MicroStrain AHRS proved to have notably slow heading measurement. Readings would continue to change in the direction of motion for almost a second after the rover had stopped turning. This reduced the general operational speed and responsiveness of the rover and effected its ability to orientate itself to specified starting directions. It is unclear if this problem was caused by limitations of hardware or software.

5.14.3.3 Expensive Proprietary Connector

The MIB (§5.4) and the Perle IOLAN SDS4 Device Server (§5.6) require serial cables to be terminated with RJ45 connectors. The MicroStrain AHRS used an expensive proprietary mini connector that was only available from the manufacturer directly. Not only did this create a significant development cost but it also makes field repairs, if one were to become necessary, dependent on a part of limited availability.

5.14.3.4 Sensitivity to External Fields

Large pieces of ferromagnetic metal or unshielded current-carrying wire can create variations in the local magnetic field. These variations will differ between rovers and over both time and location as the global magnetic field varies. The MicroStrain AHRS proved so sensitive to local field variations that it had to be placed on an elevated platform away from the rover to avoid interference from devices such as the motors, the battery packs and even the Perle IOLAN SDS4 Device Server (§5.6).

5.14.3.5 Lunar Incompatibility

The MicroStrain AHRS uses a magnetometer to measure heading. Magnetometers are not viable sensors for calculating orientation during extraterrestrial missions as they depend on the existence of a known and stable global magnetic field. Analogue missions like the Mauna Kea deployment are designed to demonstrate and test the suitability of solution methods for extraterrestrial missions. If the solution technique is not applicable, it diminishes the feasibility of the analogue deployment.

5.14.4 Evaluation

As the only sensor capable of providing orientation information about the rover, the MicroStrain 3DM-GX2 AHRS is a fundamental part of the autonomous control system. While its integration proved expensive (§5.14.3.3) and it demonstrated minor limitations (§5.14.3.1, §5.14.3.2) including a sensitivity to interference (§5.14.3.4), it was generally successful. Its dependence on the Earth's gravitational and magnetic fields (§5.14.3.5) is an acceptable conceit given the autonomous control system's use of terrestrial positioning systems (§5.15, §5.16). Continued use of the MicroStrain 3DM-GX2 AHRS is recommended for ongoing and future work so that development of other systems can continue. It is recommended that a nonterrestrial localization system that provides position and orientation be identified and developed in parallel.

5.15 u-blox EVK-5H Global Positioning System (GPS) Receiver

The u-blox EVK-5H Global Positioning System (GPS) Receiver is an evaluation kit for the u-blox 5 GPS receiver chip that uses USB for communication and power. The u-blox EVK-5H supports WAAS and can provide up to 1 metre of accuracy, however in the field 5 to 6 metres of reported accuracy was more common. The u-blox EVK-5H was originally selected to be a temporary positioning system while a more precise solution was sourced (§5.16). The u-blox remained in use as a secondary positioning device. It is installed in the MCB (§5.9) and communicates to the autonomous control system directly through the ASUS S101 Eee PC (§5.10).

5.15.1 ArgoSoft Device Server (*gps-ublox*)

The u-blox EVK-5H GPS is served by *gps-ublox*, an ArgoServer that provides clients access to the position information calculated by the evaluation kit. This includes positional information like latitude, longitude, Universal Transverse Mercator (UTM) coordinates, altitude and height-above-ellipsoid as well as measures of the calculation quality like Horizontal Dilution of Position (HDOP), estimated accuracy and the number of satellites used in the calculation. The u-blox EVK-5H also provides GPS time received from the satellites, two and three dimensional velocities and heading when moving.

5.15.2 Successes

5.15.2.1 Stable

The autonomous control system depends on external position information to localize the rovers in a global frame. Without a functioning Kalman filter (§6.3), this position information is required before the rovers can even move. In order to be able to execute experiments continuously, a GPS receiver that maintains a solution lock in varying conditions without large changes in reported accuracy is important. The u-blox EVK-5H proved to be slow finding an accurate solution, but once it attained one capable of holding it reliably.

5.15.3 Problems Encountered

5.15.3.1 Insufficient Accuracy

The rover depends on the positioning system to provide its location in a global frame. This information is used to correlate GPR data, the landing pad location and topography maps provided by the TriDAR. The accuracy of the positioning system will provide an absolute limit on how precise these correlations can be done. The average accuracy of 5 to 6 metres provided by the u-blox EVK-5H GPS receiver was insufficient. This meant that the rover could not be guaranteed to find buried objects that were big enough to preclude landing pad construction or move repeatably to topographical features selected from the TriDAR map.

5.15.3.2 Insufficient Update Rate

Autonomous controllers can only run as fast as sensors provide information. The u-blox EVK-5H runs at maximum of 1 Hz, limiting controllers to 1 Hz or slower. This problem can be overcome by combining the u-blox with a faster sensor, such as an encoder, through an estimator such as an Unscented Kalman Filter (UKF) (§6.3).

5.15.3.3 Dependence on External Satellites

The MarsDome is a large, enclosed testing facility located at UTIAS providing a work area protected from the environment that is large enough for testing multiagent autonomous rovers. It allows control algorithms to be tested and developed in a controlled environment with minimal unknown external disturbances. It is especially useful for excavation tests as it filled with uniform digging material that does not freeze in the winter. The use of a GPS receiver like the u-blox EVK-5H for position information precluded the use of indoor environments for autonomous control testing. This meant that development had to be exclusively undertaken outdoors where varying ground and weather conditions made algorithm development more difficult and time-consuming.

5.15.3.4 Lunar Incompatibility

The u-blox EVK-5H depends on a constellation of satellites orbiting Earth to calculate its position. This is not a viable solution as there is no plan for similar systems to be installed around target destinations. Analogue missions like the Mauna Kea deployment are designed to demonstrate and test the suitability of solution methods for extraterrestrial missions. If the solution technique is not applicable, it diminishes the feasibility of the analogue deployment.

5.15.4 Evaluation

The u-blox EVK-5H Global Positioning System (GPS) receiver does not have sufficient resolution (§5.15.3.1) or update rate (§5.15.3.2) to be the sole positional sensor for the autonomous control system. Its dependence on external satellites not only makes it a poor analogue sensor (§5.15.3.4), but precludes development from taking advantage of facilities available at UTIAS that would have helped advance development (§5.15.3.3). Given its stability and robustness (§5.15.2.1), its addition to the UKF (§6.3) as a secondary input would be recommended whenever the rovers are used out-of-doors. Continued use of the u-blox EVK-5H GPS receiver is recommended for ongoing and future work. It is recommended that a nonterrestrial localization system that provides position and orientation be identified and developed in parallel. While the current positional requirements make the u-blox EVK-5H inadequate as a primary positioning solution, its selection and purchase would be again recommended as a temporary positioning solution.

5.16 Trimble Global Positioning System (GPS) Pathfinder ProXRT

The Trimble Global Positioning System (GPS) Pathfinder ProXRT is a self-contained GPS receiver equipped with an internal battery that communicates to computers via serial or Bluetooth. It can be configured to use differential corrections provided by OmniSTAR Inc. that are available with a subscription from a geostationary satellite. The differential corrections provide it with the information required to reduce nonlocal GPS errors such as atmospheric effects and clock errors. With the corrections enabled, the Pathfinder ProXRT is capable of decimetre estimated accuracy, without corrections, it gets approximately 5 metre accuracy.

The Trimble Pathfinder was selected by NORCAT to serve as the primary absolute positioning sensor. The Trimble Pathfinder is a rugged device that is installed directly into the autonomous control saddle with the MIB (§5.4) and MCB (§5.9). The antenna is mounted on a mast in the saddle.

5.16.1 ArgoSoft Device Server (*WinTrimbleServer*)

The Trimble Pathfinder is served by *WinTrimbleServer*, an ArgoServer that configures the receiver and provides client access to the position and solution information available from the device. The server configures the receiver to use the appropriate differential corrections (OmniSTAR or WAAS) as well as the proper UTM coordinate system origin and various other solution configurations. The client can get position information such as latitude, longitude, UTM coordinates as well as solution information such as HDOP, estimated accuracy and the number of satellites used in the solution. As a result of the requirements of the Trimble Software Development Kit (SDK), the Trimble ArgoServer must be a Microsoft Foundation Class (MFC) application. This means that it can only run in Microsoft Windows and that it must have a Graphical User Interface (GUI) window associated with it.

5.16.2 Successes

5.16.2.1 Estimated Accuracy

The rover depends on the positioning system to provide absolute location in a global frame. This information is used to correlate GPR data, the landing pad location and topography maps provided by the TriDAR. The accuracy of the positioning system will provide an absolute limit on how precise these correlations can be done. The theoretical estimated accuracy of 10 centimetres provided by the Trimble Pathfinder receiver would be sufficient for the requirements of the deployment. It would assure that the rover found any buried objects that were large enough to preclude landing pad construction as well as move repeatably to topographical features selected from the TriDAR map.

5.16.2.2 Physical Robustness

The Trimble Pathfinder receiver is IP-67 rated [17] allowing it to be mounted directly into the autonomous control saddle without a protective enclosure. This was helpful in integrating it into a preexisting autonomous control system. Experience reinforced the manufacturer's claims that it would handle the rigours of field experiments.

5.16.2.3 Battery Life

The Trimble Pathfinder contains a battery designed to last for a full day of work. Charging the battery requires using a proprietary dongle that is not sealed and would compromise the robustness of the unit. The Mauna Kea deployment showed that the battery could last for a full day of experimentation and be charged fully overnight.

5.16.2.4 Connectivity

The Trimble Pathfinder has Bluetooth and serial connections for communication to computers. Bluetooth was used during development and the Mauna Kea deployment on all but one device, where Bluetooth problems required the use of a serial connection (§5.10.3.2). The Bluetooth connection was preferred as it did not require adaptors or other connectors that could have reduced the Pathfinder's environmental seal.

5.16.3 Problems Encountered

5.16.3.1 OmniSTAR Subscription

The Trimble Pathfinder depends on differential corrections from satellites to achieve decimetre accuracy. The corrections are provided by OmniSTAR Inc. through a subscription model. Subscriptions are purchased for a specific device on monthly or annual terms, however free subscriptions are sometimes provided for research purposes. UTIAS was able to acquire extended free subscriptions up until the end of the Mauna Kea deployment. It is unrealistic to expect OmniSTAR Inc. to continue to furnish free subscriptions for three devices indefinitely. Without the corrections, the Pathfinder is no more accurate than the u-blox EVK-5H GPS receiver (§5.15). This means that continued use of the Pathfinders requires the nontrivial reoccurring cost of a subscription for the duration of their use.

5.16.3.2 OmniSTAR Connection

The OmniSTAR corrections are continuously calculated and sent to the Trimble Pathfinder via a geostationary satellite. Loss of communication with the satellite for any reason will result in a rapid loss of accuracy. Experience showed that satellite communication was frequently lost either as a result of physical obstruction or signal interference.

Each subscription area is serviced by one geostationary satellite. At any one given terrestrial location, it will remain stationary in the sky at a fixed heading and azimuth. This means that some experimental areas near buildings or within valleys may be occluded from the satellite. This limitation had to remain in consideration during development at UTIAS as well as when selecting the Mauna Kea deployment site.

The corrections are transmitted over the L-band (1 to 2 GHz), the same frequencies used by the GPS satellites themselves. The L-band is a subset of the Ultra High Frequency radio spectrum band that is shared with cellphones, wireless networking, Bluetooth, R/C transmitters and two-way personal radios such as Family Radio Service and General Mobile Radio Service devices [18] as well as radiation from the Sensors & Software OEM Noggin NIC GPR (§5.3). Experience during the Mauna Kea deployment demonstrated that the Trimble Pathfinder's ability to receive the OmniSTAR corrections could be disrupted by the use of radios and other broadcasting electronics within the vicinity of the antenna. Care must be taken to reduce sources of interference, a task that is near impossible in uncontrolled areas given the proliferation of devices in this band.

5.16.3.3 Poor Software Development Kit (SDK) Implementation

The integration of the Trimble Pathfinder into the autonomous control system was well within the scope of its design. It used a component object model based SDK which was meant to facilitate designing software that connected to the receiver. The SDK, however, was poorly implemented in C++. It had limited examples that did not function as described as well as sparse documentation, incomplete methods and difficult-to-troubleshoot errors. For example, when queried, the SDK did not return the entire device serial number, just the first 8 characters which proved to be nonunique between the three rovers. This prevented checking to see if the proper Bluetooth pairing had occurred. The SDK also created irrecoverable errors if the OmniSTAR frequency, specified as the data variable type double, contained more than two decimal places of precision. These problems delayed integration and limited the ultimate functionality of *WinTrimbleServer*. Support requests had to go through Cansel, the Canadian distributer for

Trimble products, who had no software support and forwarded them to Trimble. This meant that support was not timely nor easy to obtain.

5.16.3.4 New Constellation Events Starving the Communication Bus

The autonomous control system depends on being provided with high-quality, reliable data from its sensors. The decisions made by the control algorithms can only be as good as the data they are based on. To this end, care is taken to only use data when it is coming from sensors that are connected, successfully communicating and otherwise functioning properly. Most sensors provide some low-level ways to verify this. The Trimble Pathfinder was no exception, however, during development a critical error was found in its communication protocol that allowed it to provide erroneous data.

Communication with the Pathfinder uses a SDK that is built around the MFC library. This allows *WinTrimbleServer* to respond to events generated by the Pathfinder object class. The most important event for an autonomous system being the "New Position" event. When the Pathfinder operated properly, this event occurs every second, providing the server with the new position calculated by the receiver. Another event "New Constellation Information" provided information on the data used to calculate the solution. It is the relative importance of these events that are a concern.

During development it was discovered that the "New Constellation Information" event could starve the connected application of any other Pathfinder information. This was observed in *WinTrimbleServer* as well as with supplied sample programs. The Pathfinder would generate a continuous stream of "New Constellation Information" events so rapidly that no other information was provided to the application. Having the application ignore the events did not help, they still starved the communication bus. It is unclear if this is a fault in the MFC based SDK that handles event notifications or in the Pathfinder itself. With information no longer updating from the Pathfinder, information such as position, solution state, connection state and estimated accuracy remain at their last received values. This both provided connected clients with incorrect position information as well as incorrectly reported that the information was valid. Until it was isolated, this had disastrous results on algorithm development.

Once the problem was identified, the algorithms could use features in the ArgoSoft framework to help recognize the onset of the starved communication bus. The only solution found to restore positional information was to wait. The problems could last over an hour. There is some empirical evidence that it may have been triggered by specific satellites in the sky, as it repeatedly occurred at the same time of day. This leads to the hypothesis that there was something about a satellite signal (e.g. signal-to-noise ratio) that was problematic for the Trimble Pathfinder. This problem was only observed in Toronto, not during the Mauna Kea deployment.

5.16.3.5 Windows Support Only

For an analogous discussion on the effect of the Trimble Pathfinder's Microsoft Windows operating system requirement, see §5.2.3.2.

5.16.3.6 Requires Microsoft Foundation Class (MFC) Library

The use of the MFC libraries in the Trimble Pathfinder SDK increases the complexity of its device server. For an analogous discussion on the importance of light-weight communication between devices, see §5.2.3.3.

5.16.3.7 Insufficient Update Rate

For an analogous discussion on the effect of the Trimble Pathfinder's 1 Hz update rate, see §5.15.3.2.

5.16.3.8 Dependence on External Satellites

For an analogous discussion on the effect of the Trimble Pathfinder's dependence on external satellites, see §5.15.3.3.

5.16.3.9 Lunar Incompatibility

For an analogous discussion on the lunar incompatibility of a GPS positions system, see §5.15.3.4.

5.16.4 Evaluation

The Trimble Global Positioning System (GPS) Pathfinder ProXRT shares a lot of limitations with the u-blox EVK-5H (§5.15). It does not have a sufficient update rate (§5.16.3.7) to be the only positioning sensor for the autonomous control system. Its dependence on external satellites not only makes it a poor analogue sensor (§5.16.3.9), but precludes development in indoor facilities (§5.16.3.8).

Physical integration into the autonomous control system was facilitated by its rugged enclosure (§5.16.2.2), self-contained battery (§5.16.2.3) and convenient connection options (§5.16.2.4).

Software integration however proved difficult. Communication depends on a poorly implemented SDK (§5.16.3.3) that not only requires Windows (§5.16.3.5) but specifically MFC-based GUI applications (§5.16.3.6). This not only limits the functionality of *WinTrimbleServer* but also the design of the entire autonomous control system. The SDK also demonstrated catastrophic communication failures (§5.16.3.4) that prevented rovers from receiving updated positions for over an hour.

If not for the Trimble Pathfinder's increased accuracy (§5.16.2.1), these problems would make the u-blox EVK-5H (§5.15) a better choice for the needs of the autonomous control system. When differential corrections are applied, the accuracy becomes high enough to be used as the primary absolute positioning sensor in a Kalman filter (§6.3). These corrections depend on a subscription from OmniSTAR and are tied to specific geographic regions and devices, creating a reoccurring cost to continued operation of the rovers (§5.16.3.1). They are transmitted from a geostationary satellite which is susceptible to physical obstruction and interference from terrestrial electronics (§5.16.3.2). Once corrections are lost, it can be upwards of an hour to regain decimetre accuracy.

These problems make the Trimble GPS Pathfinder ProXRT a poor long-term solution for providing positioning information to terrestrial mobile, autonomous robotic systems. To allow for continued algorithm development, continued use of the Trimble GPS is conditionally recommended for ongoing and future work as long as OmniSTAR subscriptions are available. Once OmniSTAR subscriptions are unavailable use of the Trimble Pathfinder is not recommended as the u-blox EVK-5H provides a better positioning solution for the needs of the system. It is recommended that a nonterrestrial localization system that provides position and orientation be identified and developed in parallel.

5.17 QualityKits VK011 Serial Temperature Sensor

The QualityKits VK011 is a serial temperature sensor control board that reads up to 4 Maxim Integrated Products DS18S20 temperature sensors and communicates to an attached computer via serial. The Maxim sensors use the 1-wire communication standard which allows sensors to be placed tens-of-meters from the control board. The VK011 can be powered over the serial port or from an external power supply. It was added to the autonomous control system in order to monitor the temperature of the motors (§2.2.2.6). The VK011 was installed into the MIB (§5.4) where it communicates to the autonomous control system through the Perle IOLAN SDS4 Device Server (§5.6). Its sensors were mounted inside the MIB, outside in the saddle and on both motors.

5.17.1 ArgoSoft Device Server (*thermal*)

The QualityKits VK011 temperature sensors are served by *thermal*, an ArgoServer that provides client access to the temperature sensors. A connected client can get the temperature and status of any of the four connected sensors.

5.17.2 Successes

5.17.2.1 Accuracy

While exhaustive quantitative tests were not undertaken, reported accuracy for the DS18S20s are 0.5°C within an operating temperature range of -10°C – 85°C [19].

5.17.2.2 Sensor Responsiveness

The Maxim sensors are lightweight with very little thermal mass which allows them to react quickly to changes in temperature. When coupled with the 1 Hz update rate of the control board, this allowed the QualityKits VK011 Serial Temperature Sensor to provide temperature values rapidly enough to monitor the motors and take appropriate action (§6.9).

5.17.3 Problems Encountered

5.17.3.1 Frozen Hardware

During deployment it was discovered that the QualityKits VK011 Serial Temperature Sensor could become frozen and unresponsive to serial commands. The only known solution was a complete power cycle of the board. As a result of a flaw in the design the MIB (§5.4.2.7), this could only be accomplished by disconnecting and reconnecting the power cables to the VK011. It is possible that this parasitic powering was also the source of the problem.

5.17.4 Evaluation

The QualityKits VK011 Serial Temperature Sensors were simple to integrate, accurate (§5.17.2.1) and responsive (§5.17.2.2). Given their late addition to the autonomous control system, this makes them a success. The observed hardware freezes (§5.17.3.1) may have been caused by parasitic power available to the VK011 inside the MIB after shutdown (§5.4.2.7) or software and will require further investigation. Continued use of the QualityKits VK011 Serial Temperature Sensor is recommended for ongoing and future work.

5.18 DirectedPerception PTU-D47 Pan-Tilt Unit (PTU)

The DirectedPerception PTU-D47 is a weatherized Pan-Tilt Unit (PTU) capable of positioning payloads of up to 12 pounds. It is capable of high speed panning at up to 300°/s, as well as tilt positioning as precise as as 0.003° [20]. It is used to position the SICK LMS111 Laser Measurement System (§5.19) as well as the Microsoft LifeCam VX-3000 Cameras (§5.20). The PTU-D47 is mounted on the sensor mast at the front of the rover and communicates to the autonomous control system through the Perle IOLAN SDS4 Device Server (§5.6) located in the MIB (§5.4).

5.18.1 ArgoSoft Device Server (*ptud46*)

The DirectedPerception PTU-D47 is served by *ptud46*, an ArgoServer that provides clients access to DirectedPerception PTU device family. Clients can read the current pan and tilt positions as well as command the PTU-D47 to move to a new orientation. A client can also control the behaviour of the PTU-D47 by changing the actuation speed and speed profiles. Speed profiles allow the user to specify an acceleration as well as a maximum and base speed that the PTU-D47 will use when moving.

5.18.2 Successes

5.18.2.1 Positional Accuracy

The accuracy of the DirectedPerception PTU-D47 will effect the usefulness of data collected by its sensor payload. Considering the uncertainties of the rover sensor mast and the rover as a whole, the PTU-D47 has more than sufficient positional accuracy.

5.18.2.2 Range of Motion

Using the SICK LMS111 (§5.19) for low-level collision avoidance requires that it can scan ahead of the rover regardless of the direction of motion. The PTU-D47 provides 360° of yaw rotation, guaranteeing that both the SICK and the Microsoft LifeCam VX-3000 Cameras (§5.20) can be pointed in any direction of travel. The tilt range also allows for the sensors to be positioned up or down any reasonable slope as well as at the ground for integration into excavation controllers.

5.18.2.3 Speed of Motion

In order for the SICK LMS111 (§5.19) to be used for low-level collision avoidance it must be able to remain positioned ahead of a moving rover. The PTU-D47 provides fast enough positioning to guarantee that both the SICK and the Microsoft LifeCam VX-3000 Cameras (§5.20) will remain directed ahead of the rover regardless of its speed.

5.18.3 Problems Encountered

5.18.3.1 Limited Load Capacity

The SICK LMS111 (§5.19) is near the maximum payload limit of the DirectedPerception PTU-D47. To decrease the chance of damage to the PTU-D47, mounting modifications were made to the SICK to limit its moment of inertia (§5.19.3.1). This limits future payload flexibility for the autonomous control system and may prove to reduce the lifespan of the PTU-D47.

5.18.3.2 Unexplained Interference with RoboteQ AX2850

During the Mauna Kea deployment, the DirectedPerception PTU-D47 was found to cause unexplained interference with the RoboteQ AX2850 (§5.1) when used in the *MCS* (§6.2). Given the limited time available and the secondary nature of the PTU-D47, the decision was made to deactivate it for the remainder of the mission. This is a new development and investigation is still ongoing, possibile causes include interference, power brownouts, or software.

5.18.4 Evaluation

The DirectedPerception PTU-D47 Pan-Tilt Unit (PTU) is a mature device to UTIAS, being used in autonomous control systems on previous rovers. It is well suited for the required dynamic needs (§5.18.2.1, §5.18.2.2, §5.18.2.3) of orientating sensing payloads on the rover. While loading limitations must be considered for future modifications (§5.18.3.1), the PTU-D47 was sufficient for its expected payload. Observed interference between the unit and the RoboteQ AX2850 (§5.18.3.2) is a concern but considering the history with similar devices is likely a resolvable integration problem. Continued use of the DirectedPerception PTU-D47 is recommended for ongoing and future work.

5.19 SICK LMS111 Laser Measurement System

The SICK LMS111 Laser Measurement System is an IP-67 rated scanning laser range finder capable of measuring distances up to 20 metres in a 270° area around the sensor at up to 50 Hz. The measured distances are typically correct within +/- 30 millimetres at an angular accuracy of +/- 0.5° [21]. The LMS111 was to be used for both low-level collision prevention as well as ground measurement for improved digging performance (§7.2.4), however time and resource constraints meant that it was never fully integrated into the autonomous control system. The

SICK LMS111 is mounted on the DirectedPerception PTU-D47 (§5.18) and communicates to the autonomous control system through the Cisco SD208 Ethernet Switch (§5.5) located in the MIB (§5.4).

5.19.1 ArgoSoft Device Server (*sick*)

The SICK LMS111 Laser Measurement System is served by *sick*, an ArgoServer that provides clients the ability to request a scan from the LMS111 and receive the measured values. The client can also read status information from the device, such as laser state and cleanliness. Due to time constraints, the ArgoServer was never fully developed.

5.19.2 Successes

Unfortunately the SICK LMS111 was not integrated to a level of maturity that allowed an assessment of its success.

5.19.3 Problems Encountered

5.19.3.1 Difficulty Mounting

The SICK LMS111 Laser Measurement System was mounted on the DirectedPerception PTU-D47 (§5.18) so that the LMS111 could be positioned to scan for obstacles during transit as well as in front of the blade during excavation. As a result of the size and shape of the LMS111 it was felt it would put an excessive load on the PTU-D47 and that it should be mounted upside-down. This moved the mass closer to the centre-of-rotation and minimized its moment of inertia as experienced by the PTU-D47. It means, however, that the scanning surface of the LMS111 is more susceptible to the accumulation of dust. It remains to be seen if this will be an issue.

5.19.3.2 Lack of Documentation

The SICK LMS111 lacked complete documentation. This lack of information delayed integration and would have limited the functionality of the final device server. The SICK LMS111 lacked the information required to establish a connection between a client computer and the device using any piece of software other than the provided monitoring program. While SICK Inc., the North-American subsidiary of the German SICK AG was able to furnish a summary of the basic messages necessary to establish connection and request basic scans in a *Quick Manual*, they informed UTIAS that since the device was a new design, they would have to attain further documentation from their German head office, SICK AG. [22]. No further documentation was ever provided.

5.19.4 Evaluation

The SICK LMS111 Laser Measurement System contains the core functionality necessary to be a rover mounted scanning laser range finder. While time and resource constraints along with outside delays meant that it was never fully integrated into the autonomous control system before the Mauna Kea deployment, UTIAS remains confident that it will demonstrate its benefit to the autonomous control system as a whole by providing information on the immediate surroundings of the rover. No further evaluation can be made at this time. Continued development and integration is thoroughly and enthusiastically recommended.

5.20 Microsoft LifeCam VX-3000 Cameras

The Microsoft LifeCam VX-3000 Camera provides 640x480 resolution video over a USB interface at up to 30 frames per second [23]. Two were used on the Musketeers to provide live pictures from the rovers to remote operators at the ground control station. They are mounted on the DirectedPerception PTU-D47 (§5.18) and communicate to the autonomous control system through the Belkin 7-Port Mobile USB Hub (§5.12).

5.20.1 ArgoSoft Device Server (***ArgoVis***)

The LifeCam VX-3000 Cameras are served by *ArgoVis*, a specialized ArgoServer that provides clients a connection to USB video cameras. Clients can be running on the same computer or remotely over the network and are capable of changing image settings, such as compression, resolution, colour balance and others. The client can receive individual frames or continuous video from the selected cameras.

5.20.2 Successes

The Microsoft LifeCam VX-3000 Cameras functioned successfully as USB video cameras.

5.20.3 Problems Encountered

5.20.3.1 Frozen Hardware

During development, it was discovered that the Microsoft LifeCam VX-3000 Cameras require a physical reboot if anything goes wrong with their connection. This problem was not seen during their deployment.

5.20.4 Evaluation

Most of the experience with the Microsoft LifeCam VX-3000 Cameras occurred during development at UTIAS. The decision was made during the Mauna Kea deployment to not operate them to limit the load on the ASUS S101 Eee PC (§5.10). As such, it is difficult to say how prevalent and disruptive problems seen in development (§5.20.3.1) would prove to be in deployment. The selection and of a USB Video Class (UVC) compliant camera would have been preferred as devices that meet the UVC standard do not require a driver in most modern operating systems. Regardless, continued use of the Microsoft LifeCam VX-3000 are recommended for ongoing and future work.

Chapter 6

Autonomous Control System Software Modules

The autonomous control system uses a large number of software modules. All were written by UTIAS and use the ArgoSoft framework. The following section details the modules not directly associated with hardware devices.

6.1 *Configuration Manager Server (CMS)*

The *Configuration Manager Server (CMS)* provides ArgoSoft programs with rover-specific configurations at run-time. This allows parameters to be changed without requiring the recompilation of code. The values are stored in a comma separated file as either rover-specific or general values. Examples include device serial port addresses, actuator configuration values and digging controller parameters. The *CMS* runs anytime a server module in *ArgoServer.exe* is run.

6.1.1 Successes

6.1.1.1 Simple Method for Rover Specific Configurations

Even though they are constructed from the same components, each rover requires unique calibrations to account for differences in individual hardware. The goal of the *CMS* server is to provide software with a flexible and transparent source of these rover specific configurations. This is accomplished by storing data in keys that can be sub-identified by rover name. Client programs request values by key name (e.g. *key*) and the *CMS* will first, if present, return the value stored under the rover specific key (e.g. under *key*:*rover*). If there is no rover specific value stored, the *CMS* will return the general value. This is all transparent to the client, allowing for identical software to run on different rovers with different configuration values.

6.1.2 Problems Encountered

6.1.2.1 Restarting Server Required to Update Values

The *CMS* is designed to provide a framework for storing general and rover-specific configuration values outside the code of ArgoSoft programs. It does this by reading in a comma separated database of entries at startup and serving the values to connected clients. During development and deployment, this allows algorithm parameters to be modified without recompilation of the code. Updating values does require restarting the *CMS* and the client program. Since the *CMS* is integrated into *ArgoServer.exe*, this means that every running server module must be simultaneously restarted. This is still preferred to manually handling rover differences and recompiling code, but given the connection problems of some devices in the autonomous control system (§5.6.2.1, 5.7.2.1) it is not ideal.

6.1.3 Evaluation

The *Configuration Manager Server (CMS)* successfully removes the complications of dealing with rover differences from a common code-base (§6.1.1.1). While updating values during rover operation can be problematic (§6.1.2.1), it has rightfully become integral to the design of the autonomous control system. Continued use and development of the *CMS* for ongoing and future work is highly recommended.

6.2 *Musketeer Control Server (MCS)*

The *Musketeer Control Server (MCS)* is an algorithm server that provides clients with a single interface for high-level rover control. Its primary function is to implement a low-level digging controller [24] around which high-level excavation controllers can be designed (§6.6, §6.7, §6.8). The *MCS* also provides high-level traverse algorithms for positioning the rover to arbitrary locations.

To aid evolved, discrete high-level excavation controllers such as *C1* (§6.6) and *C2* (§6.7), the *MCS* also provides discretized position data from the *Kalman Filter Server (KFS)* (§6.3) and discrete moving and digging commands. The low-level digging controller is designed to create a continuous excavation pass from multiple discrete dig commands.

6.2.1 Successes

6.2.1.1 Low-Level Digging Controller

The ultimate success of the ASSP tasks of the ISRU project depends on the digging capabilities of the Musketeer rovers. The rovers must be able to create a level and flat surface from unpredictable ground conditions and profiles so that landing pads and roads can be constructed from a series of excavation passes. The design of the digging controller is complicated by the use of discrete high-level controllers. These controllers will attempt to make a continuous excavation pass by issuing successive short dig commands. In order to simplify their design, the low-level controller must be created so that these repeated digging commands are equivalent to a single long pass. The low-level digging controller requires tuning to specific soil conditions but was demonstrated before and during the Mauna Kea deployment to be successful. It can create flat, continuous passes from multiple or single client commands (Figure 6.1).

Figure 6.1: *Athos* testing the digging controller during the Mauna Kea deployment. Photograph courtesy of Kenneth Law.

6.2.1.2 Single Interface for High-Level Digging Controllers

The modular design of the autonomous control system maximizes portability between platforms but requires independent connections to each device server from controllers. This can become cumbersome to design, troubleshoot and maintain as well as difficult to use manually in the field. The *MCS* simplifies this by creating a single interface to the group of devices that constitute a Musketeer rover. This decreases the design complexity of the high-level excavation controllers. It also allows operators to teleoperate the rover using the same command set as the high-level controllers. This proved invaluable during development and the Mauna Kea deployment to both test and demonstrate functionality.

6.2.1.3 Repeatable Discrete Actions

The discrete nature of the evolved excavation controllers requires that the rover be able to repeatably perform a set of basis behaviours. The basis behaviours consisted of driving forward or backwards 1 metre, turning left or right 90°, digging forward 1 metre and unloading the blade. The *MCS* was responsible for providing these behaviours to connected clients through the modified application of its driving and digging algorithms. The *MCS* was demonstrated to be able to repeatably and reliably execute these basis behaviours.

6.2.1.4 Robust Driving Controller

Rover tasks depend on having a reliable driving algorithm that is robust to disturbances. Placing the algorithm in the *MCS* makes it available to all connected high-level controllers as well as operators in the field. The driving controller is designed for nonholonomic vehicles and attempts to arrive at the goal location in the correct orientation. It is a proportional controller depending on three variables [25]. The distance between rover and goal (ρ), the angle from the rover to ρ (α) and the angle from ρ to the goal heading (β) (Figure 6.2, Equations 6.1-6.2). To facilitate the future integration of low-level collision avoidance, the DirectedPerception PTU-D47 (§5.18) tracks ahead of the rover in order to scan for upcoming obstacles.

$$v = k_\rho \cdot \rho \tag{6.1}$$

$$\omega = k_\alpha \cdot \alpha + k_\beta \cdot \beta \tag{6.2}$$

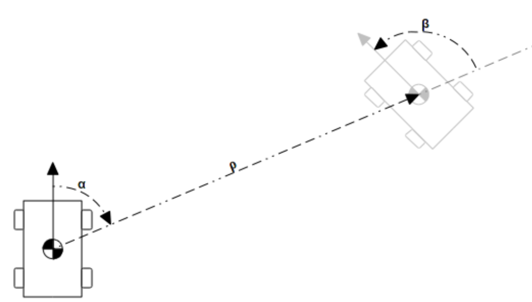


Figure 6.2: Diagram of the driving controller [25].

6.2.2 Problems Encountered

6.2.2.1 Aggressive Excavation Passes

The *MCS* server takes relatively large cuts of earth in each excavation pass. This increases the load on the motors as well as decreases the finish of the cut. Reducing the cut depth would improve the cut finish; however, is not possible considering the limitations of the load cells (§3.1.2.5, §3.1.2.6, §3.1.2.7) and the noise problems experienced with the SCADA Box (§3.3.2.4, §3.3.2.5, §3.3.2.6, §3.3.2.7).

6.2.2.2 Command Priority Conflicts

The *MCS* is designed to handle multiple connected clients issuing conflicting commands by using a priority system. In theory, this allows for emergency stops and other overrides to be issued by programs or operators in the field. The current implementation has some conditions under which higher priority commands are erroneously ignored. These situations do not present themselves in the standard operating procedures followed during the Mauna Kea deployment, however they exist and should be fixed. They do not effect the ability of the *MCS* to handle multiple commands from a single client, multiple clients silently connected or commands that are issued when the rover is idle.

6.2.2.3 Loss of Ground Contact During Excavation

The low-level digging controller has no sensors to measure the ground height in front of the blade. It uses load cells to make an initial estimate and then uses geometry to integrate the estimate forward for the remainder of the excavation pass. If the initial ground measurement is too high, the resulting excavation pass will do no work and simply keep the blade level in the air as it drives. During the Mauna Kea deployment, this result occurred repeatedly as a result of sensor error (§3.1.2.6, §3.1.2.7) and widely varying ground profiles. This is an expected limitation of the solution method and will be remedied when the SICK LMS111 (§5.19) is integrated into the digging controller (§7.2.4).

6.2.2.4 Lack of Low-Level Collision Avoidance

The autonomous control system planned to use a two-level collision avoidance system. Preliminary collision avoidance was to be provided by high-level excavation controllers that make intelligent scheduling decisions to reduce the risk of collision. A low-level collision avoidance algorithm running in the *MCS* would then scan for unexpected obstacles during operation and prevent the rover from driving into them. This provides collision avoidance even when the rover is operated manually. The lack of a completely integrated SICK LMS111 (§5.19) prevented its implementation. This means that there are no software modules specifically preventing the rover from driving into obstacles.

6.2.2.5 Unexplained DirectedPerception PTU-D47 Interference

For an in-depth discussion of the interference observed during the Mauna Kea deployment, see §5.18.3.2.

6.2.2.6 Inconsistent Data Flow (Orientation)

The autonomous control system was to use a UKF (§6.3) to provide an optimal estimate of the rover position and orientation. Since the *KFS* was unfinished for the Mauna Kea deployment (§6.3.2.1), the *MCS* was redesigned to access the MicroStrain 3DM-GX2 AHRS (§5.14) directly so it could obtain a higher data rate than available from the incomplete *KFS*. The *MCS* should get all its pose information from the *KFS* once it is finished.

6.2.2.7 Inconsistent Logging

The ArgoSoft framework contains a powerful and flexible logging system called *Logger* (§6.11). It was designed so that asynchronous data from a variety of sources could be recorded easily with a common timestamp. It creates consistent log files in a common directory that are ready for analysis in MATLAB or GNU Octave. It was designed to be the logging solution for every server that is a part of the autonomous control system. The *MCS* does not implement *Logger*, instead choosing to handle its own logging requirements in an arbitrary format and location. This means that operators must read another data format when analyzing experimental results. This increases the time required to analyze data as well as places the data at risk of being lost or ignored.

6.2.2.8 Incomplete Implementation of GPR Mode

In addition to digging algorithms for excavation controllers, the *MCS* contains mobility algorithms that are useful to any control client. In order to allow access to these algorithms when the blade payload (§3.1) is not attached, the *MCS* disables the low-level digging controller. This is referred to as the *GPR mode* as it was initially designed for use with the GPR payload (§3.2). Development of the *GPR mode* in the *MCS* is not complete and does not function correctly, requiring *GPRSurvey* to duplicate code in order to be functional (§6.5.2.3).

6.2.3 Evaluation

The *Musketeer Control Server (MCS)* was designed to give high-level access to low-level algorithms for various controllers and clients (§6.2.1.2). It was successful in providing low-level digging (§6.2.1.1) and driving (§6.2.1.4) controllers as long as the blade payload (§3.1) was attached (§6.2.2.8). It also provided discretized actions that could successfully be combined together to approximate continuous actions (§6.2.1.3) as required by the discrete controllers *C1* and *C2* (§6.6, §6.7).

Developers are aware of a bug in the priority handling (§6.2.2.2) and design decisions inconsistent with the ArgoSoft design framework (§6.2.2.6, §6.2.2.7). It is recommended that these be resolved when time is available. Remaining problems with the digging controller (§6.2.2.1, §6.2.2.3) and low-level collision avoidance (§6.2.2.4) depend on the completed integration of the SICK LMS111 Laser Measurement System (§5.19). Once it is implemented, new digging and driving controllers can be developed (§7.2.4, §7.2.2). Continued use and development of the *MCS* is recommended for ongoing and future work.

6.3 *Kalman Filter Server (KFS)*

The *Kalman Filter Server (KFS)* is an algorithm server that provides clients with estimates of the rover position and orientation. When completed the *KFS* will implement an UKF to combine the various sensors into a best estimate of the state variables of the rover. Specifically, it will acquire encoder values from the RoboteQ AX2850 (§5.1), accelerations and orientations from the MicroStrain 3DN-GX2 AHRS (§5.14) and absolute positions from the Trimble GPS Pathfinder ProXRT (§5.16) and the u-blox EVK-5H GPS (§5.15). When the blade payload (§3.1) is attached, the prediction model will incorporate forces read from the Omega OMB-DAQ-3001 (§5.2) to better estimate wheel slippage during digging.

Until the UKF is completed, the *KFS* passes the sensor information directly to connected clients. This provides clients with positional data in a local Cartesian frame and will facilitate adoption of the UKF.

6.3.1 Successes

6.3.1.1 Single Interface for Pose Information

The KFS provides a single location for all Musketeer rover pose information. For an analogous discussion on the benefits of a unifying server, see §6.2.1.

6.3.1.2 Positional Information in a Local Cartesian Frame

The ISRU project requires positional data to be exchangeable between Neptec, Xiphos and UTIAS. This data must share a common coordinate system and origin. The agreed standard was to use a local Cartesian frame. The *KFS* handles the conversion of the latitude and longitude values from the GPS receivers to the local Cartesian frame using GeographicLib [26]. Having the entire autonomous control system operate in the local Cartesian coordinate system aids debugging and increases interoperability between project groups.

6.3.2 Problems Encountered

6.3.2.1 Incomplete Implementation

As a result of time and resource constraints, a functioning UKF was never implemented into the *KFS*. The *KFS* only functions as a unified connection point and a converter from latitude and longitude to a local Cartesian coordinate system. As a result, the *KFS* is limited to run at 1 Hz, the rate of the slowest device. This has required some servers (§6.2, §6.5) to bypass the *KFS* to connect directly to higher-bandwidth devices (§6.2.2.6, §6.5.2.4). The designed *KFS* framework will allow the UKF to be installed transparently to client applications.

6.3.2.2 Lack of Manual Local Cartesian Conversion

The local Cartesian frame is the common coordinate system for the Mauna Kea deployment. The *KFS* converts position information from connected devices to the coordinate system automatically. It does not provide a way to convert arbitrary values. During the Mauna Kea deployment it was frequently necessary to survey specific points. This was done manually using a Trimble GPS Pathfinder ProXRT (§5.16) which gives latitude and longitude values. Converting these to the local Cartesian frame was difficult; however, given a properly exposed *KFS* client command, this functionality could be implemented through *CaT* (§6.4).

6.3.2.3 No Data Logging

The KFS does not implement any logging. For an analogous discussion on the importance of having a consistent logging structure, see §6.2.2.7.

6.3.3 Evaluation

The *Kalman Filter Server (KFS)* is incomplete. While it provides a single interface for rover pose information (§6.3.1.1) in a local Cartesian frame (§6.3.1.2), the Unscented Kalman Filter (UKF) was not implemented as planned (§6.3.2.1). It also does not implement *Logger* (§6.3.2.3) or allow client access to its coordinate transformations (§6.3.2.2). Continued use and development of the existing *KFS* framework is recommended for ongoing and future work while the UKF is implemented. The framework is in place to allow the UKF to be inserted transparently to client programs.

6.4 *Calibration and Testing (CaT)*

Calibration and Testing (CaT) was designed as a way to test and calibrate device servers and aid in the development of algorithms but grew to be the default Command-Line Interface (CLI) for Musketeer rovers. It was used during development and the Mauna Kea deployment to position the rover, query its status and in general provide field operators with manual control over the rover when required.

6.4.1 Successes

6.4.1.1 Simple System Control

The ArgoSoft framework is modular to promote interoperability between hardware platforms as well as software controllers. This independent nature can lead to dispersed and inconsistent user interfaces for the various software modules. *CaT* was invaluable in providing a single unified control point for operators to the majority of the autonomous control system. Actuating the blade, moving a rover or polling a sensor use different device servers and hardware yet were all available to the operator in one location. This functionality was instrumental to the completion of the Mauna Kea deployment.

6.4.2 Problems Encountered

6.4.2.1 Inconsistent Menus

One of the advantages of having a unified CLI to the Musketeer rovers lies in providing a consistent interface to various algorithms and sensors despite their disparate design and functionality. This only occurs as a result of conscious effort. As development wore on, some new menus in *CaT* became either inconsistent or illogically arranged. This is especially noticeable when the exit or halt commands vary between devices. This decreases the advantages of using a centralised program like *CaT* as a common CLI.

6.4.2.2 Incomplete Implementation

When implemented properly, the menu structure of *CaT* allows operators unfamiliar with a specific device server or algorithm to quickly find and operate it in the field. Again the wear of development lead some devices and algorithm servers to not be fully implemented in *CaT*. The unpredictable nature of experimental field robotics means that the operators may need unexpected functions of a specific device or algorithm server. Having some not implemented in *CaT* decreases its functionality; as the operators have to search for other ways to accomplish specific manual functions.

6.4.2.3 Stranded Algorithms

The goal of UTIAS in the ISRU project is to develop an autonomous control system for ASSP that requires minimal human interaction. Any algorithms that are implemented in *CaT* are only accessible by human operators and therefore are not a true part of the autonomous control system. They should be placed in more appropriate locations, like algorithm servers, where control clients as well as human operators can execute them.

6.4.3 Evaluation

CaT was invaluable to the development of the autonomous control system and its use during the Mauna Kea deployment. Having a single control point to the majority of the device and algorithm servers (§6.4.1.1) made controlling the rovers easier for operators in the field and prevented time consuming mistakes. The rapid development cycle however has seen a decrease in interface consistencies (§6.4.2.1, §6.4.2.2) that if left uncorrected will reduce *CaT*'s usability. These problems, coupled with the existence of algorithms (§6.4.2.3), reinforce the need for regular upkeep to assure that *CaT* maintains its core functionality as a simple CLI to the Musketeer rovers. Continued use and development of *CaT* for ongoing and future work is recommended with regular maintenance.

6.5 *GPRSurvey*

The first autonomous step in landing pad construction (Figure 1.2) is performing a subsurface Ground Penetrating Radar (GPR) scan. This is accomplished by *GPRSurvey*, a control client that runs autonomous, multiagent GPR surveys by connecting directly to the Sensors & Software OEM Noggin NIC GPR (§5.3). The area to be scanned is divided into sub-surveys for each rover and then entered into the controller. *GPRSurvey* then evenly divides the area east-west and meticulously drives the rovers in north-south lines acquiring GPR traces as per the ISRU GPR Server Interface Control Document (ICD) [27]. After each line is finished the controller drives the rover to the next starting location which is offset one line-spacing to the east of the previous line. Once the survey is finished, the collected data is sent to Xiphos Technologies Inc. where it is post-processed and analyzed.

6.5.1 Successes

6.5.1.1 Evenly Spaced Straight Survey Lines

The data collected by *GPRSurvey* must meet the requirements specified in the ICD by Xiphos so that it can be properly post-processed. The ICD requires that the lines be collected north to south and be evenly spaced and sequential east to west. The individual lines must be straight and parallel [27]. Meeting these requirements depends on both the positional information available and *GPRSurvey*. The final version of the controller used during the Mauna Kea deployment was able to place the rover at the starting positions within the accuracy of the positioning system (Figure 6.3). *GPRSurvey* was then able to maintain straight lines while driving over terrain with varying slopes (Figure 6.3). While the variations in topography were minor, over the length of the scan line, they were large enough to disrupt the course of the rover when operated without the controller. The result was evenly spaced straight lines that satisfied the requirements of the ICD.

Figure 6.3: GPR survey during the Mauna Kea deployment. Photograph courtesy of Francis Frenzel.

6.5.2 Problems Encountered

6.5.2.1 Memory Corruption and Leaks

The current implementation of *GPRSurvey* shows significant memory leaks and memory corruption. Each GPR trace gets a new memory location that is never released. This means that the resources used by *GPRSurvey* will incrementally grow, taking limited computing resources away from other processes. There is also at least one known instance of memory corruption. Its source is unknown but it has caused otherwise inexplicable run-time changes to configuration values. Further debugging will be required to isolate the problem and repair it.

6.5.2.2 Inconsistent Hardware Connection (Sensors & Software OEM Noggin NIC GPR)

The ArgoSoft framework serves hardware devices to connected clients like the *GPRSurvey* control client via device servers (§4.5.1). This creates device modularity between hardware and software as well as between platforms. *GPRSurvey* does not follow this design framework and instead connects directly to the Sensor & Software OEM Noggin NIC GPR (§5.3). This was acceptable given the time constraints of the project, but given the size of the autonomous control system code base creates an untenable inconsistency long-term that will effect code maintenance. Implementation of a GPR device server will also help deal with known hardware issues when reconnecting to the Noggin NIC (§5.3.2.1).

6.5.2.3 Inconsistent Design (Algorithms)

Having algorithms in only one location is a key to a maintainable code-base. It assures that all software uses the same version of the algorithm and facilitates bug-fixes and updates. Since the *GPR mode* of the *MCS* (§6.2) was not finished (§6.2.2.8), *GPRSurvey* had to replicate driving algorithms from the *MCS*. *GPRSurvey* should use the versions of the algorithms in the *MCS* as soon as the *GPR mode* is completed (§7.1.7).

6.5.2.4 Inconsistent Data Flow (Orientation and Position)

As a result of the incomplete KFS (§6.3), *GPRSurvey* accesses the MicroStrain 3DM-GX2 AHRS (§5.14) and RoboteQ AX2850 (§5.1) directly. For analogous discussion see §6.2.2.6.

6.5.2.5 Single Raster Scan

GPR surveys construct a three-dimensional image of the subsurface from one-dimensional measurements of boundary transitions. Sensors & Software recommends that to achieve a reliable three-dimensional image at least two perpendicular raster scans are used [28]. This prevents the long, unsurveyed areas that exist between scan lines of a single raster scan. *GPRSurvey* is capable of doing perpendicular raster scans; however, Xiphos specifies in the ICD that surveys must be limited to a single raster scan [27]. Using a parallel raster scan would be a simple way to increase the resolution of the survey.

6.5.3 Evaluation

The *GPRSurvey* control client was capable of running a survey autonomously that met the ICD requirements for both line orientation and spacing (§6.5.1.1). This allowed the data to be post-processed successfully by Xiphos. The autonomous GPR survey was a success.

Once the *Musketeer Control Server (MCS)* (§6.2) and the *Kalman Filter Server (KFS)* (§6.3) are finished, *GPRSurvey* should be designed to use them instead of direct connections to device servers (§6.5.2.4) and replicated code (§6.5.2.3). To maintain ArgoSoft framework standards, a device server should be designed for the Sensors & Software OEM Noggin NIC GPR (§5.3) and *GPRSurvey* should use it instead of a direct hardware connection (§6.5.2.2). A concerted effort must be made to find and repair the memory leaks and corruption (§6.5.2.1). Continued use and development of the *GPRSurvey* control client for ongoing or future work can only be recommended once the memory problems are solved.

6.6 *C1*: Artificial Neural Tissue (ANT) Controller

The primary high-level, multiagent excavation controller is *C1*, a discrete evolved controller based on the Artificial Neural Tissue (ANT) controller developed at UTIAS [29]. By using an evolved controller, it is hoped that behaviours will emerge that demonstrate novel solutions to the ISRU problem. *C1* is evolved using genetic algorithms to solve the landing pad and road construction problem in a simulated grid world on a cluster of computers. A measure of each controllers success, deemed its fitness, is used to guide the evolution towards a solution. Once a suitably optimal solution is found in simulation, *C1* is ported to physical hardware and the real experimental environment.

On physical hardware, *C1* continues to operate in a discretized grid world. Its inputs are the discretized topology data provided by the TriDAR, rover position and rover orientation. All discretized information and actions are available through the *MCS* (§6.2). The topographical map is provided over the ArgoSoft protocol *SharedMap* from the *WinGCS* (§6.10).

6.6.1 Successes

6.6.1.1 Success in Simulation

An evolved controller must first find a solution to the given problem in simulation. This is not simple, as the high-dimensional solution space is not guaranteed to be continuous. *C1* demonstrated excellent success at the simulated landing pad and road construction problem. It was able to clear and prepare an area in 250 discrete moves. While it is unrealistic to expect this level of efficiency in the real world, the simulation results are positive enough to merit trying the controller on real hardware.

6.6.2 Problems Encountered

6.6.2.1 Insufficient Testing Time

Comprehensive real-world testing with the autonomous control system and accompanying hardware is required to test *C1*. As a result of various internal and external pressures, including scheduling constraints, shortfalls and software errors, there was insufficient time to completely test *C1* during the Mauna Kea deployment. Evolved controllers are not conducive to partial completion as their strengths depend on stochastically emergent behaviour. Aborting the test makes it difficult to predict the final success of the controller.

6.6.3 Evaluation

C1 performed well enough in simulation (§6.6.1.1) to merit real-world testing. Unfortunately there was insufficient testing time to evaluate *C1* on real hardware (§6.6.2.1). In a results-based project such as the Mauna Kea deployment, it is difficult to justify the complexity and time spent developing, evolving and deploying an evolved controller such as *C1* when hand-coded controllers, such as *C3* (§6.8) show more appreciable results with less work. As a result, continued use and development of *C1* cannot be recommended for future and ongoing deployment work. This report makes no attempt to evaluate the research value of ANT.

6.7 *C2*: Alternative Neural Network (ANN) Controller

The secondary high-level, multiagent excavation controller is *C2*, a discrete evolved controller based on a simplified feed-forward network developed at UTIAS for the project named the Alternative Neural Network (ANN). By using an evolved controller, it is hoped that behaviours will emerge that demonstrate novel solutions to the ISRU problem. *C2* is evolved using genetic algorithms to solve the landing pad and road construction problem in a simulated grid world on a cluster of computers. A measure of each controllers success, deemed its fitness, is used to guide the evolution towards a solution. Once a suitably optimal solution is found in simulation, *C2* is ported to physical hardware and the real experimental environment.

On physical hardware, *C2* continues to operate in a discretized grid world. Its inputs are the discretized topology data provided by the TriDAR, rover position and rover orientation. All discretized information and actions are available through *MCS* (§6.2). The topographical map is provided via the *SharedMap* framework from the *WinGCS* (§6.10).

6.7.1 Successes

6.7.1.1 Success in Simulation

C2 was able to clear and prepare a simulated work area in 3000 discrete moves and successfully solved 10 of 10 problems presented. While simulation differences make comparing these numbers to those of *C1* impossible, an analogous discussion on their meaning is presented in §6.6.1.1.

6.7.1.2 Implementation of an Evolved Controller on Physical Hardware

It is a significant challenge to implement a controller evolved in simulation on real-world hardware and get similar behaviours. *C2* demonstrated that the design of the autonomous control system and specifically the *MCS* provided the necessary framework. *C2* was run successfully on the Musketeer rovers during the Mauna Kea deployment with observed behaviours and decisions similar to those seen in simulation. The evolved controller made appropriate decisions and actions towards the goal of a creating a landing pad.

6.7.2 Problems Encountered

6.7.2.1 Collision Avoidance

Evolved multiagent controllers are expected to make decisions to reduce the risk of collisions. In order to facilitate the emergence of this behaviour, *C2* used hand-specified occlusion zones around each rover. When zones overlapped, the affected rovers execute a predefined set of manoeuvres to position themselves away from potential collisions. On real hardware it proved difficult to properly specify the size of these occlusion zones. Too small and the uncertainty of experimental robotics could allow the rovers to collide. Too large and the rovers would become trapped; unable to move more than a few spaces in any direction and spending a majority of their time executing the collision avoidance manoeuvres. In practise, the occlusion zones had to be chosen from observed behaviours in the field and other heuristic metrics.

6.7.2.2 Inadequate Simulation Fidelity

It is impractical to have a simulation capture all the complexities present in experimental robotics. Complex simulations would be prohibitively expensive and make the genetic algorithm process more difficult and computationally expensive. Instead, the complexity of the real-world task is split between low-level controllers and the high-level evolved controller.

The most pertinent features of the hardware and the task are modelled in simulation. This creates an environment that encompasses the major problems of the task without the complexity of the minor ones. A controller evolved on this simulation will be a solution to the conceptualized task. The unmodelled complexities are then dealt with by low-level controllers. In the autonomous control system, these are the basis behaviours of the *MCS* (§6.2). The combination of these two controllers should result in a complete real-world solution to the specified task.

The Mauna Kea deployment showed that for *C2* this was not the case. Significant real-world complexities were observed that were either unexpected or not accounted for in the simulation.

These included the amount of positional and behavioural noise and the soil conditions.

The existence of noise was not unexpected but *C2* appeared to have problems dealing with its level. Rover positions would vary with the accuracy of the Trimble GPS Pathfinder ProXRT (§5.16). The addition of error in the *MCS* basis behaviours made the result of manoeuvres inconsistent. Commands meant to move the rover one grid-square forward could result in a move of too far forward, forward and sideways or many other combinations. The feed-forward nature of ANN only uses current data for decisions, leaving *C2* unaware of the result of its action. This should have limited the effect of noise to a loss in efficiency. This appeared to be true except when near other rovers or boundaries.

Between rovers, the position and behaviour noise meant that collisions were possible. This required an increase in the the size of the occlusion zone surrounding the rovers (§6.7.2.1). When near the work area boundaries, the noise meant that rovers could find themselves outside the work area. This was a condition the controller could not deal with as it had evolved to never cross the boundary. Solving this problem required additional code akin to the collision avoidance that gave the rovers a series of manoeuvres to return them to the work area.

The soil conditions resulted in unexpected and unmodelled complications to the landing pad construction task. The simulation had assumed that moving the rover had no effect on the ground. During the Mauna Kea deployment, this was observed to be untrue. The rovers left clear, significant tracks in the loose volcanic soil as a result of their skid-steer design. These effects were most pronounced when turning on the spot, a behaviour frequently used. Since the rovers spent a lot of time positioning themselves for new excavation passes, their previous passes became indiscernible under tire marks. It could appear that the rovers were making no headway towards the targeted landing pad.

There was insufficient testing time (§6.7.2.4) to tell if this lack of simulation fidelity would ultimately prevent task completion or merely increase the solution time. It remains a clear concern.

6.7.2.3 Asynchronous Behaviour Execution

C2 was evolved in a world where rovers acted simultaneously. In the real world the basis behaviours take varying lengths of time to finish. Further variance is added by network delays encountered when attempting to distribute data to other rovers following a completed move. If desired, synchronous behaviour must be enforced in software. Because it was not, *C2* and the rovers acted asynchronously. Rovers could execute multiple, fast moves (e.g. turning) in the time it took another rover to execute a single one (e.g. digging). Since rovers only updated their position at the end of a move, this meant rovers were basing decisions on invalid data. Coupled with unmodelled positional noise (§6.7.2.2), this promoted interagent antagonism by increasing collision potential as well as disrupting any emergent cooperative behaviour that may have existed in *C2*.

6.7.2.4 Insufficient Testing Time

For an analogous discussion on the effect of aborted tests, see §6.6.2.1.

6.7.3 Evaluation

C2 was developed in approximately the same time as *C3*, a hand-designed traditional controller (§6.8). The simulated results (§6.7.1.1) showed enough promise to merit real-world testing; however, time constraints and other failures prevented the controller from being tested to completion (§6.7.2.4). Testing did reveal the success of basis behaviours (§6.7.1.2) and the difficulties of ensuring collision avoidance (§6.7.2.1), finding proper simulation fidelity (§6.7.2.2) and the necessity of synchronous behaviours (§6.7.2.3). Further testing would help evaluate the magnitude of these problems and their effect on task completion. In a results-based project such as the Mauna Kea deployment, it is difficult to justify the complexity and time spent developing, evolving and deploying an evolved controller when hand-coded controllers, such as *C3* show more recognizable results with less development time. As a result, continued use and development of *C2* cannot be recommended for future and ongoing deployment work. This report makes no attempt to evaluate the research value of ANN.

6.8 *C3*: Model-Based Control

The tertiary high-level, multiagent excavation controller is *C3*, a model-based controller. It works in a continuous world to create a level landing pad with a semicircular berm around it or create a straight access road. It uses the same low-level digging controller implemented in the *MCS* (§6.2) as the evolved controllers.

The scheduling algorithm used for *C3* prioritizes levelling the highest distinct points ("hills") in the work area. It does this by selecting one hill for every active rover and directing them to the hills that minimize the total distance travelled by the team. The rovers then execute excavation passes from the hill radially to the perimeter of the landing pad where the soil is deposited as a berm. The rovers return to the hill 4 times, slightly offsetting their starting point each time. The topology information is provided by the same discretized map as *C1* and *C2* (§6.6, §6.7) which comes from the TriDAR via the *WinGCS* (§6.10) over *SharedMap*.

6.8.1 Successes

6.8.1.1 Creation of Perimeter Berm

To increase efficiency, *C3* always finishes excavation passes on the perimeter so the load can be used to create the berm. Starting construction of the berm at the beginning of the excavation process meant that a semicircular berm began to emerge quickly around the landing pad. By the end of the limited testing period available for *C3* (§6.8.2.4), the protective berm was continuous and deemed sufficiently high (Figure 6.4).

Figure 6.4: Landing pad berm constructed during the Mauna Kea deployment. Photograph courtesy of Kenneth Law.

6.8.1.2 Prioritization of Highest Points

The successful construction of a landing pad requires a surface levelled with respect to the gravitational frame. *C3* attempts to accomplish this by selecting the highest points in the work area as measured in the local Cartesian frame that is aligned to the Earth ellipsoid. The algorithm successfully identified the highest hills and directed the rovers to excavate them.

6.8.1.3 Even Distribution of Starting Points

The scheduling algorithm in *C3* attempted to select distinct high points as starting positions for excavation passes. It did this by placing an occlusion zone around local maxima that prevented the selection of any high points included in it. Once hills were identified, the rovers were assigned such that the total distance travelled by the team was minimized. During the

Mauna Kea deployment this helped disperse the rovers around the work area, maximizing the appearance of progress and decreasing the amount for interagent antagonism.

6.8.2 Problems Encountered

6.8.2.1 Collision Avoidance

C3 depends on its scheduling algorithm to avoid collisions. The hill assignment method not only minimizes the distance travelled by rovers but also reduces the opportunity for rover collision. The Mauna Kea deployment showed that while this was true under most operating conditions, it was insufficient. The nonholonomic nature of the driving algorithm (§6.2.1.4) does not guarantee a straight path towards a goal. If the end and starting orientation are sufficiently different, the rover may take a large, circuitous route to arrive at its next starting point. This increases the chance for collisions and the rover leaving the workspace. The scheduling algorithm also had problems when rovers were removed or added from the working group. As a result of the motor issues (§2.2.2.6), this can occur frequently. These problems were solved during the Mauna Kea deployment with human spotters on the ground, but a better high-level collision avoidance scheme, coupled with a working low-level collision avoidance system based on the SICK LMS111 Laser Measurement System (§7.2.2) is necessary.

6.8.2.2 Static Scheduling Algorithm

The work area of the Mauna Kea deployment had a continuous slope down from the north to the south. This meant that the majority of the high points selected by the scheduling algorithm in *C3* were located near the north boundary. As this was also the location of the semicircular berm, the resulting excavation passes were quite short and clustered *C3*'s efforts in one area. This is not ideal as excavation loads from further south will have to pass over the northern areas to reach the berm. This demonstrated a need for the controller to make more intelligent choices about the starting points of the rovers. This was temporarily accomplished by preventing *C3* from selecting hills that were too close to the northern boundary. This solved the problem during the Mauna Kea deployment but would have become problematic if there was sufficient time to complete the task (§6.8.2.4) as eventually only hills inside the northern occlusion zone would remain. This demonstrates the need for a second generation scheduling algorithm that is more dynamic in its hill selection.

6.8.2.3 Insufficient Final Landing Pad Smoothness

Not only does the final landing pad need to be level to the gravitational field but it also needs to be smooth. The smoothness of individual excavation passes is handled by the low-level digging controller (§6.2) while the overall smoothness is handled by the high-level excavation controller. *C3* attempts to create smoothness simply by targeting the highest points above the goal height for excavation (§6.8.1.2). While *C3* was not given time to finish its run (§6.8.2.4), it was not creating a smooth landing pad (Figure 6.5). Not only did digging passes leave windrows in the work area, but driving rovers left significant tire marks. It is unclear if continued operation would have been sufficient to remove these or if a separate control stage is necessary for smoothing the landing pad.

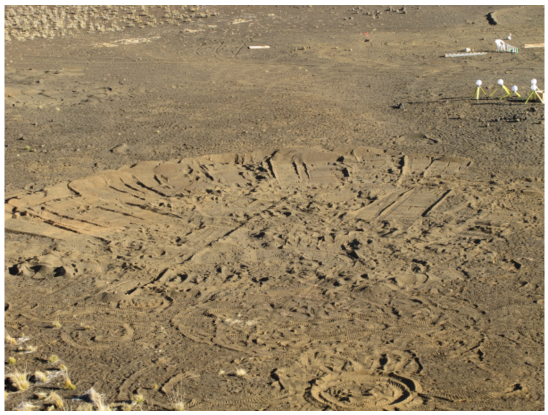


Figure 6.5: Landing pad constructed during the Mauna Kea deployment. Photograph courtesy of the author.

6.8.2.4 Insufficient Testing Time

As a result of various internal and external pressures, including scheduling constraints, shortfalls and software errors, only two days were available during the Mauna Kea deployment to test *C3*. This is insufficient time to test the effectiveness of its design and its final results. Longer testing is required.

6.8.3 Evaluation

C3 was developed as a benchmark for the evolved controllers *C1* and *C2* (§6.6, §6.7). *C3* proved by far to be the most visibly successful controller tested during the Mauna Kea deployment. While problems with the scheduling algorithm may have diminished the final quality of the landing pad (§6.8.2.2, §6.8.2.3), there was insufficient testing time to find out (§6.8.2.4). In the limited time that was available, *C3* showed the ability to recognize the highest points in the work area (§6.8.1.2), distribute the rovers accordingly (§6.8.1.3) and very quickly start to create a perimeter berm (§6.8.1.1). Problems with collision avoidance (§6.8.2.1) will be aided by the eventual integration of the SICK LMS111 Laser Measurement System (§7.2.2) but a redesign of the scheduling algorithm may also be beneficial. In a results-based project such as the Mauna Kea deployment, the quick development cycle and immediately appreciable results of *C3* make it the most suitable choice for a high-level excavation controller. Continued use and development of *C3* is recommended for future and ongoing deployment work. *C3* can also continue to provide a benchmark controller for future research on evolved controllers.

6.9 *ThermalMon*

In order to protect the motors from thermal damage (§2.2.2.6), temperature sensors were installed on the motor casings (§5.17). *ThermalMon* is a control server that monitors these temperatures from *thermal* (§5.17.1), logs them and stops the rover when they rise above a user-specified value. The rover remains paused until the motors cool below a second designated value. During the Mauna Kea deployment, values of 80 °C and 60 °C respectively were used.

6.9.1 Successes

6.9.1.1 Motor Thermal Protection

The limitations of the motors (§2.2.2.6) necessitated that steps be taken to limit their rate of failure. Temperature was identified as the underlying cause of physical damage to the motors and was used to enforce periods of rest. *ThermalMon* was designed solely for this purpose. It achieves this by connecting to the RoboteQ AX2850 (§5.1) and issuing a high-priority stop command. As long as *ThermalMon* is connected and enforcing the stop command, all other RoboteQ commands will be ignored. The motors remain stopped until they cool down at which point *ThermalMon* releases its lock on the RoboteQ.

6.9.2 Problems Encountered

6.9.2.1 Other Actuators Remaining Operable

The late addition of the motor thermal protection did not allow time to design a method to notify the rest of the autonomous control system in the event of a thermal pause. *ThermalMon* uses the existing command priority framework in ArgoSoft to override all other commands to the RoboteQ AX2850 (§5.1). This allows controllers to continue operate and continue to send commands to rover devices. *ThermalMon* makes no attempt to pause any server other than the RoboteQ. While the rover cannot drive during thermal protection events, it can still operate the blade or GPR payload (§3.1, §3.2) and move the DirectedPerception PTU-D47 (§5.18). This partial behaviour may be problematic to the original controller and confuses the visualization of the thermal protection.

6.9.2.2 Redundant Logging

ThermalMon implements its own logging solution in addition to using *Logger* (§6.11), unnecessarily increasing the SSD throughput (§5.10.3.1). For an analogous discussion on the importance of having a consistent logging structure, see §6.2.2.7.

6.9.3 Evaluation

ThermalMon was added to the autonomous control system late in the development cycle to deal with recognized problems with the motors (§2.2.2.6). Despite incomplete (§6.9.2.1) and inconsistent (§6.9.2.2) implementation, *ThermalMon* accomplishes its main task of protecting motors from thermal damage by preventing their operation when they are overheated (§6.9.1.1). Addressing the listed problems will increase the robustness of the solution and help integrate it into the autonomous control system. Continued use and development of *ThermalMon* is recommended for ongoing and future work.

6.10 *WinGCS*

The Ground Control Station (GCS) provides operators with remote access to the autonomous control system on each rover. *WinGCS* is a GUI for the GCS. It provides graphical representation of rover information, including position, status and current basis behaviour, as well as providing the operator with pause and resume commands. *WinGCS* also triggers the TriDAR provided by Neptec Design Group Ltd. and discretizes the output topography for distribution to rovers via the *SharedMap* protocol. Implemented in MFC, the GUI can be run on any computer running a Microsoft Windows operating system.

6.10.1 Successes

6.10.1.1 Unified Graphical Display of Rover Status

Autonomous rovers are equipped with multiple sensors and actuators. While these exist predominantly for use by the control algorithms, unusual values or sudden changes can provide insight to an attentive operator. *WinGCS* was fundamental in collecting the information available from the device and algorithm servers and presenting them in a central location on the GCS. This helps identify and troubleshoot potential problems. Common examples included drive motor temperature readings and rover battery levels.

6.10.2 Problems Encountered

6.10.2.1 Pause Commands Not Resuming

In addition to presenting rover status graphically in a single location, *WinGCS* was designed to provide operators with high-level pause and resume commands. Given network connectivity, these were meant to function as a remote, software-based emergency stop. Currently, the pause button works but the resume button does not, rendering a remote emergency stop possibly but making resuming work more difficult. Further investigation is necessary to isolate the problem.

6.10.2.2 Communication of Discretized TriDAR Map

WinGCS served as the map master server for the discretized topography map used by the digging controllers *C1*, *C2* and *C3* (§6.6, §6.7, §6.8). This was accomplished using the *SharedMap* protocol in the ArgoSoft framework. Once the rovers reported that they were ready, *WinGCS* would trigger the TriDAR for a new scan. Upon scan completion, *WinGCS* would process the data into a discrete map where the height of each discrete grid was the highest point found in that area and distribute it to the rovers. During the Mauna Kea deployment problems developed in this process. There were issues providing correct handshake protocols as well as periodic problems with timing and triggering the TriDAR scans. Further investigation is necessary to identify and resolve the problem.

6.10.3 Evaluation

As a result of time and resource limitations, *WinGCS* remains an immature piece of software. While it successfully presents rover status graphically (§6.10.1.1) it suffers bugs in its control functionality (§6.10.2.1) and map distribution (§6.10.2.2) that are typical of its developmental maturity. *WinGCS* still provides a necessary service and continued use and development of *WinGCS* for ongoing and future work is recommended.

6.11 *Logger*

Logger is a flexible logging system designed to be a logging solution for the entire autonomous control system. It allows asynchronous data from a variety of sources to easily be recorded with a common timestamp. It creates consistent log files that are ready for analysis in MATLAB or GNU Octave in a common directory regardless of the source.

6.11.1 Successes

6.11.1.1 Unified Logging Solution

The autonomous control system is a collection of hardware and software running at various frequencies with various internal clocks. These different sources make it difficult to collect and compare data. *Logger* successfully provided a common and flexible logging solution for all the devices. This consistency makes data analysis faster and easier.

6.11.2 Problems Encountered

6.11.2.1 Incomplete Log Files on Improper Shutdown

Closing *Logger* requires waiting for it to write the remainder of its buffer to file. A large number of log files from Mauna Kea show that *Logger* was not allowed to close cleanly and was instead terminated. This truncates the last line and requires minor manual intervention before the data can be loaded into MATLAB or GNU Octave.

6.11.2.2 High Data Throughput

Autonomous control systems have limited computing resources. One known bottleneck of the ASUS S101 Eee PC (§5.10) is the limited write speed of its SSD (§5.10.3.1). Running at 10 Hz, *Logger* writes a significant amount of data, so reducing the log file size will help reduce the SSD bottleneck. This can be accomplished by using an light-weight compression algorithm in the process of writing the logs to the SSD.

6.11.2.3 Incomplete Implementation

The benefits of *Logger* depend on its universal implementation as a logging solution. Having servers implement their own logging system (Table 6.1) results in inconsistent file formats and locations throughout the autonomous control system. While this is the responsibility of the individual servers, it impacts the functionality of *Logger*.

6.11.2.4 Logging Limited to ArgoServers

Logger was envisioned to be a common logging solution to all of the ArgoSoft framework. The client/server design makes this impossible, it limits *Logger* to collecting data from servers as a client polling sensor values. This leaves control clients, such as digging controllers, without a common logging solution. Experience debugging systems during the Mauna Kea deployment reinforce the need to have logs of the control algorithms. Client logging was accomplished independently for each piece of software without the benefit of the *Logger* framework.

Table 6.1: Logging solutions organized by server.

Device or Algorithm Server	Logging Solution
RoboteQ AX2850	*Logger*
Omega OMB-DAQ-3001	*Logger*
ASUS S101 Eee PC	*Logger*
MicroStrain 3DM-GX2 AHRS	*Logger*
u-blox EVK-5H GPS	*Logger*
Trimble GPS Pathfinder ProXRT	*Logger* (Incomplete)
QualityKits VK011 Serial Temperature Sensor	*Logger*, *ThermalMon*
SICK LMS111 Laser Measurement System	None
DirectedPerception PTU-D47	*Logger*
Microsoft LifeCam VX-3000 Camera	N/A
Musketeer Control Server	*Musketeer Control Server*
Kalman Filter Server	None

6.11.3 Evaluation

Logger was designed to provide an easy-to-implement self-contained logging solution for the autonomous control system. While its limitation to ArgoServers (§6.11.2.4) is serious, it did provide these servers with a unified, consistent logging solution (§6.11.1.1). Inconsistent implementation is almost predominantly the result of design choices by the individual developers (§6.11.2.3), not a result of limitations of *Logger*. While experience during the Mauna Kea deployment have exposed some minor areas for improvement (§6.11.2.1, §6.11.2.2), *Logger* was ultimately successful in the majority of its goals. Continued use and development of *Logger* is recommended for ongoing and future work while it is extended to including logging for client programs.

6.12 *HealthMon*

Designed prior to the *WinGCS* (§6.10) *HealthMon* is a CLI with auditory warnings to collect and present rover health information to operators.

6.12.1 Successes

HealthMon worked as designed to present rover health related information to the operator.

6.12.2 Problems Encountered

No problems were encountered with *HealthMon*.

6.12.3 Evaluation

HealthMon functioned as designed to provide a CLI for operators to various health-related information about the rover. Continued use and development of *HealthMon* is recommended for ongoing and future work where necessary.

6.13 *CameraMon*

CameraMon is a client for *ArgoVis* (§5.20.1), the device server used for the Microsoft LifeCam VX-3000 (§5.20) and other cameras. It displays live images from the camera on the same rover or over the network. It also allows operators to change camera specific settings on some camera models. The rover cameras allow operators to examine the work area remotely as well as document interesting events.

6.13.1 Successes

CameraMon worked as designed to provide live streaming video from the rovers.

6.13.2 Problems Encountered

No problems were encountered with *CameraMon*.

6.13.3 Evaluation

CameraMon is a mature piece of software and functioned as designed to control and present cameras over the network. Continued use and development of *CameraMon* is recommended for ongoing and future work.

Chapter 7

Recommendations

The completion of the Mauna Kea deployment is a major milestone in the ISRU project. It provides an opportunity to review accomplishments and adjust future priorities and resources. This report recommends that in order to maximize the progress to date concerted effort be placed in the following areas.

7.1 Areas Requiring Immediate Attention

Required work is listed in perceived order of importance. It should be viewed as an addendum to the continued refinement that occurs naturally when using devices or software. The following areas are the most pressing subset of the problems laid out in this report. They demonstrate the highest possibility to negatively impact future work and should be dealt with immediately.

7.1.1 Solve the Locomotion System Problems

The basis of any autonomous mobile robotic system is its locomotion system (§2.2). It must be reliable and robust; capable of performing repeatable actions. This has never been the case with the rovers as a series of ongoing problems has lead to time consuming fixes and outstanding issues. Currently, the locomotion system is functional but not robust enough to comfortably accommodate algorithm development. It uses underpowered motors that fail regularly (§2.2.2.6). The RoboteQ AX2850 controller demonstrates random motion (§5.1.3.2, §5.1.3.3) and fails when switched under load (§5.1.3.5) which is the only emergency stop procedure available that does not place operators in danger. The encoder connections are intermittent and unreliable (§2.2.2.2). The following steps are recommended:

1. Source and install replacement motors (§2.2.2.6),

2. Fix problems with the RoboteQ AX2850 (§7.1.2),

3. Finish replacing the encoder wire harnesses (§2.2.2.2).

7.1.2 Redesign the RoboteQ AX2850 Circuit

The autonomous control system accesses the entire locomotion system (§2.2) through the RoboteQ AX2850 motor controller (§5.1). Previous experience with the device has shown it should be capable of this task, however this has not been the case during the project. The RoboteQ has demonstrated frightening instabilities, including random driving surges (§5.1.3.3) and runaway conditions (§5.1.3.2). It loses motor power on some chassis when performing tight turns under R/C (§5.1.3.4). The controller cannot be stopped without entering the workspace or risking its failure from switching under load (§5.1.3.5). Given the clear inadequacies of the accompanying circuit (§5.1.3.6), it is difficult to isolate the source of these problems. The following steps are recommended:

1. Redesign the accompanying circuit to incorporate the design requirements provided by RoboteQ Inc. (Figure 5.1) and to provide a remote emergency stop (Figure 7.1). Specifically,

    i. Remove the erroneous debounce circuits,

    ii. Replace the SPST main relay with a Single-Pole, Double-Throw (SPDT) relay that grounds the power input of the RoboteQ AX2850 whenever the rover is powered off (Figure 7.1a),

    iii. Add a fuse to the RoboteQ power input and include a diode to allow for regenerated energy to be returned to the battery if the fuse is blown (Figure 7.1b),

    iv. Control the RoboteQ control line directly via a R/C operated relay (Figure 7.1c),

    v. Switch the RoboteQ control line with a SPDT switch that assures that the RoboteQ can be powered off at any time by grounding the control line and that the RoboteQ is held off whenever power is disconnected (Figure 7.1d),

    vi. Control the input to the RoboteQ directly via a SPDT switch (Figure 7.1e),

    vii. Disconnect pin 7 from the D-SUB 15 pin connector (Figure 7.1f),

    viii. Assure the motor outputs are tied to ground if the hardware version of the controller requires it (Figure 7.1g).

    ix. Assure that the connection of a computer to the serial port does not create unwanted connections between otherwise independent circuits. The connected serial device must share some common electrical reference with the RoboteQ circuit or be optically isolated.

2. Reevaluate the suitability of the RoboteQ AX2850.

Separate switching of the control and the input lines removes the need for the automatic restart circuit on the control line. Switching the RoboteQ control input will require the operator to manually shut down the RoboteQ, switch the control line and restart the RoboteQ. This is a minor workflow change but it should have powerful effects on the stability of the RoboteQ installation. It gives operators complete control of the power state of the RoboteQ under all conditions. This should prevent the RoboteQ from turning on in an indeterminate state while under load and stop devices failures from switching control inputs. Including the recommended grounding options should help return regenerated power to ground and further improve the stability of the RoboteQ installation.

7.1.3 Redesign the Supervisory Control and Data Acquisition (SCADA) Box

The SCADA Box (§3.3) was designed under fundamental misunderstandings of the component specifications. The OMB-DAQ-3001 (§5.2) is not an isolated device (§5.2.3.1), making the attempt to build an isolated circuit around it fundamentally unstable (§3.3.2.4, §3.3.2.5, §3.3.2.6). The active range of the load cells do not coincide well with the working range of the rover (§3.1.2.5). The actuators and load cell readings were susceptible to noise and offsets (§3.3.2.7). The actuators themselves were damaged (§3.1.2.4) by unknown electrical events. It is recommended that prior to solving these problems that the OMB-DAQ-3001 is first examined in case it is the cause. The following steps are recommended:

1. Decide whether or not an isolated circuit is desired,

2. If designing an isolated circuit, purchase a new DAQ and redesign the circuit taking extra care that connecting a computer to the SCADA Box does not bridge the individual circuits,

3. If designing a nonisolated circuit, purchase a new DAQ or redesign the circuit with a proper understanding of the limitations of the OMB-DAQ-3001.

If at all possible, it is recommended that the OMB-DAQ-3001 be replaced regardless of the desired circuit design. The OMB-DAQ-3001 limits the operating system choices of the autonomous control system (§5.2.3.2) as well as places an inordinate amount of load on the ASUS S101 Eee PC (§5.2.3.3). Upon repairing the circuit around the DAQ, the following steps are recommended:

1. Test to see if the observed problems with sensor readings have been resolved,

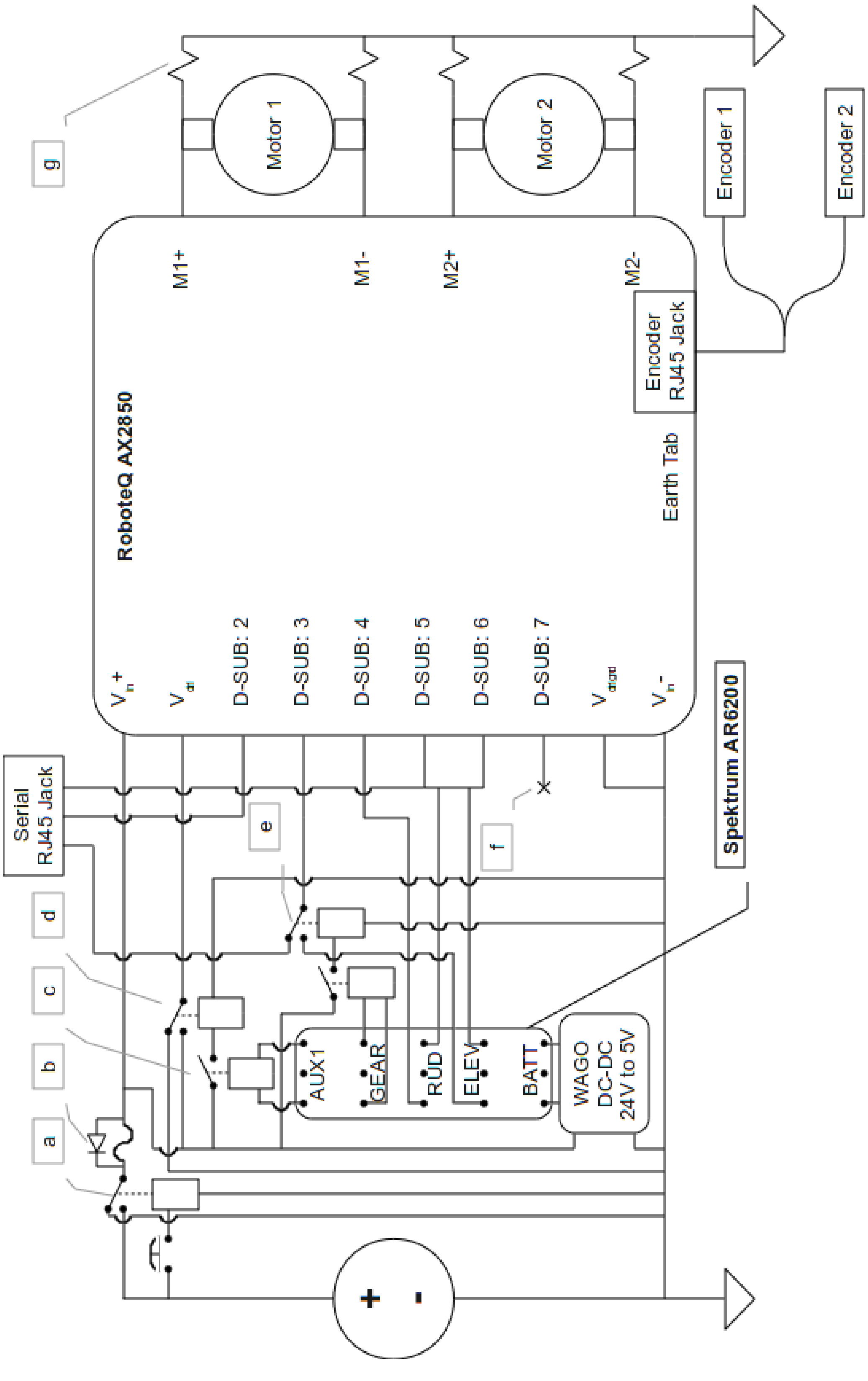


Figure 7.1: Simplified electrical schematic of a proposed solution to the problems experienced with the RoboteQ AX2850.

2. If the active range of the load cells is still a poor match for tractive effort of the rover, add amplifier circuits between the load cell controllers and the DAQ,

3. If the load cells or actuators continue to show noisy or offset values, add filters circuits between the sensors and the DAQ.

7.1.4 Finish the *Kalman Filter Server (KFS)*

An autonomous mobile robotic platform can only operate as well as the information provided to it. No information is more important than its position and orientation. The first step towards improving the positional knowledge of the autonomous control system is to finish the *KFS* (§6.3). The following steps are recommended:

1. Implement a basic UKF using the encoders, a primary positional sensor, such as the Trimble GPS Pathfinder ProXRT (§5.16), and heading sensor, such as the MicroStrain 3DM-GX2 AHRS (§5.14).

2. Add additional sensors to the UKF including secondary positional sensors such as the u-blox EVK-5H GPS (§5.15), integrated accelerations calculated from the AHRS or any other available exteroceptive sensors.

3. Make the encoder models dynamic depending on the current state of the rover. If the rover is traversing or surveying, assume less wheel slippage than if the rover is digging. Make the slippage during digging dependent on the position and loading of the blade.

7.1.5 Find a Permanent Positioning System

The Trimble GPS Pathfinder ProXRT (§5.16) with OmniSTAR corrections is not a suitable long-term primary positioning system. The per-unit subscription cost (§5.16.3.1) is recurring and prohibitive. It is susceptible to interference (§5.16.3.2) and has a slow update rate (§5.16.3.7). It also limits where experiments can be undertaken by relying on external satellites (§5.16.3.8). Implementing the UKF (§6.3) will fix some of these problems by combining the absolute positioning of the GPS with the relative position of the encoders. However, problems like losing satellites or OmniSTAR corrections remain a serious concern. There are two recommended solutions:

1. Short term: Implement the UKF to make full use of the available accuracy. Work around periods of inactivity caused by lost satellites and correction information, accepting that these will delay work and dilute results.

2. Long term: Develop a self-contained positioning system independent of external sources. Possibilities include radio triangulation systems (Sapphire DART [30]), a TriDAR provided by Neptec Design Group Ltd. or image based localization systems.

7.1.6 Resolve the Critical System Error

Having rovers prone to random, irrecoverable failures is not conducive to developing an excavation platform. The critical system error created by either the Perle IOLAN SDS4 Device Server or the Lantronix UBox 2100 (§5.6.2.3) must be identified, fixed or removed immediately. The following steps are recommended:

1. Identify the offending device by running the rovers locally and examining the Stop Error Screen,
2. Update firmware and drivers to try and fix the problem,
3. If the problem persists, redesign the interface topology to remove the offending device.

A possible solution would be to transition from an Ethernet-based system to a USB2.0-based system (Figure 7.2). This will require careful design as USB hubs operate at the speed of the slowest connected device [31]. USB1.0 devices will have to be kept on separate hubs from USB2.0 devices.

7.1.7 Refactor *Musketeer Control Server (MCS)*

The *MCS* (§6.2) has become the central algorithm server in the Musketeer autonomous control system. It has outstanding issues that are impeding the development of other software. Despite containing driving algorithms that are required by *GPRSurvey* and general operators, it does not function properly without a blade payload (§6.2.2.8) and has priority conflicts that prevent the proper handling of multiple commands (§6.2.2.2). The current version behaves rationally, but not as intended. The following actions are recommended:

1. Implement the *GPR mode* so that it does not require a SCADA server (§6.2.2.8),
2. Fix issues resolving multiple commands (§6.2.2.2).

7.1.8 Fix Memory Problems in *GPRSurvey*

The implementation of *GPRSurvey* is unacceptable. It has serious memory leaks as well as unexplained memory corruption (§6.5.2.1). This decreases confidence in the collected GPR data. Before any further GPR surveys are undertaken, these problems must be fixed. The following steps are recommended:

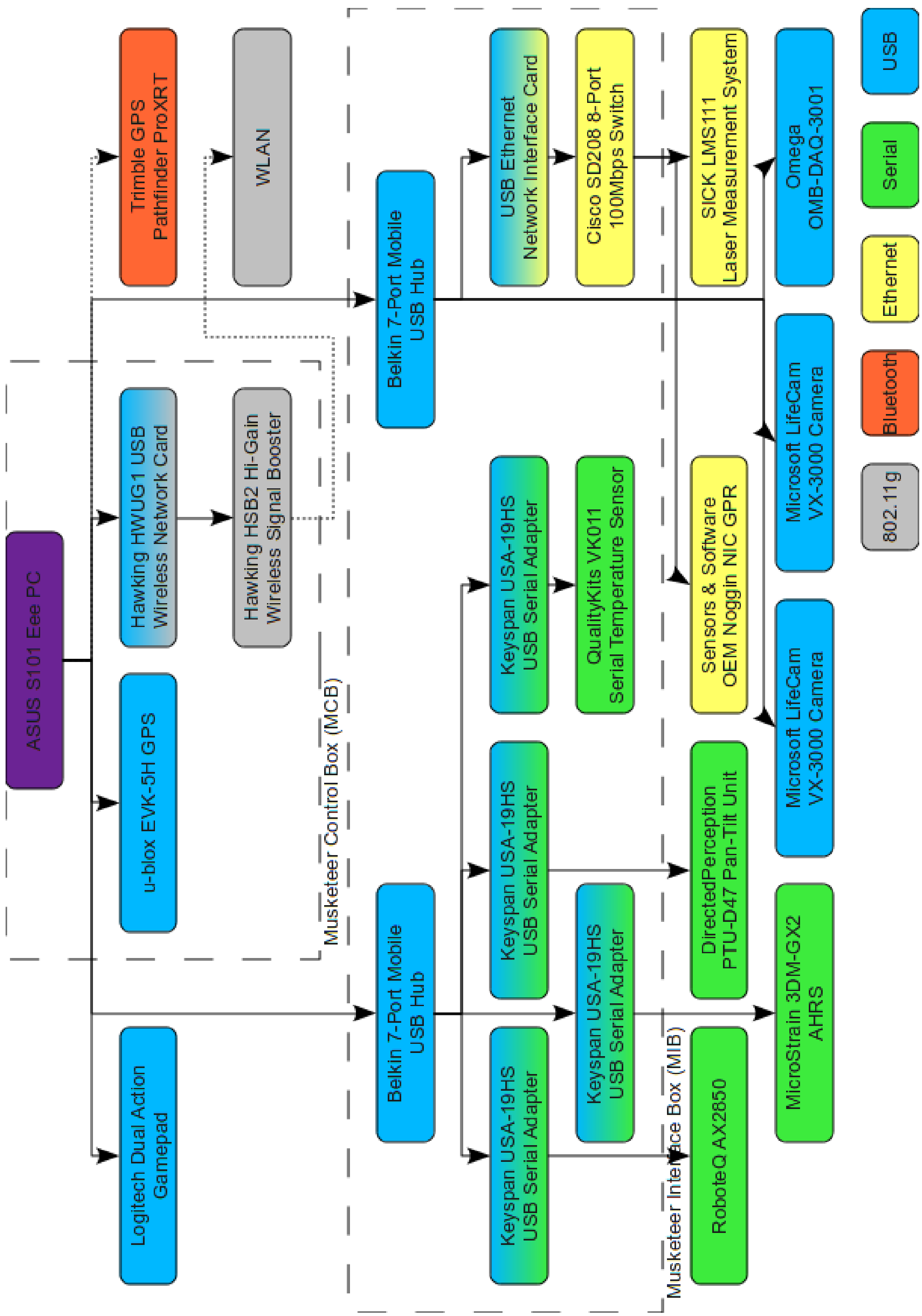


Figure 7.2: Alternative interface topology removing the Perle IOLAN SDS4 and the Lantronix UBox 2100.

1. Preserve the control algorithms,
2. Redesign the controller architecture,
   i. Design following ArgoSoft design standards (§6.5.2.2, §6.5.2.3, §6.5.2.4),
   ii. Implement knowledge gained during the Mauna Kea deployment,
   iii. Implement a clear memory handling system to release allocated memory,
3. Test the new controller.

7.1.9 Create a Logging Framework for Clients

Successful autonomous control development depends on being able to analyze data and the resulting decisions. This requires a consistent and thorough logging scheme. *Logger* (§6.11) provides this for device and algorithm servers but there is no equivalent for control clients (§6.11.2.4). This leaves a significant gap in the logging structure, requiring individual client modules to log data for post-experimentation analysis. One of the following two solutions is recommended, however the first is preferred:

1. Modify *Logger* to log arbitrary data, not just data available from servers. This may be possible by reversing the client/server relationship and making *Logger* a server to which clients send data to be logged.
2. Create a separate logging solution that is equivalent to *Logger* for software that is not an ArgoServer.

7.2 Areas for Rapid Advancement

Required work is listed in perceived order of importance. It should be viewed as an addendum to the continued refinement that occurs naturally when using devices or software. The following areas are those that demonstrate a strong potential for rapid and significant gains. Working on these areas should be secondary to the areas that require immediate attention (§7.1).

7.2.1 SICK LMS111 Laser Measurement System Integration

Integration of the SICK LMS111 Laser Measurement System (§5.19) into the autonomous control system is incomplete. This is the result of shortages in time and resources, no other obstacles are known that should prevent its completion. Finishing it will open up multiple design and research paths, including low-level collision (§7.2.2) and enhanced digging controllers (§7.2.4).

7.2.2 Low-Level Collision Avoidance

Low-level collision avoidance is dependent on the integration of the SICK LMS111 Laser Measurement System (§7.2.1). Possible low-level collision avoidance solutions vary from simple stop commands to the integration of potential field controllers. A potential field controller would apply an increasing virtual force on the rover the closer it got to an obstacle. This smoothly, if somewhat unpredictably, steers the rover away from objects in the work area.

7.2.3 Advanced Ground Penetrating Radar (GPR) Scans

The quality of the GPR survey can be improved by allowing more flexibility in the collection of the data if Xiphos Technologies Inc. can alter the ICD (§6.5.2.5). The ability to process an arbitrary collection of traces would allow for more advanced scanning techniques, from the already implemented but unused parallel raster scan to dynamic survey controllers that attempt to cluster measurements around areas of interest.

7.2.4 Enhanced Digging Controller

The integration of the SICK LMS111 Laser Measurement System (§7.2.1) into the autonomous control system would permit the development of next-generation digging controllers. These advanced controllers would be capable of measuring the height of the ground in front of the rover and adapting their control states. Effort spent preparing the existing digging controller framework for the addition of the SICK will expedite the development of the new controllers when it becomes available. Possible future enhancements include predicative controllers and on-the-fly path determination.

7.2.5 Clean Up Code

Multiproject code-bases require clear organization to remain functional and manageable long term. Having multiple developers with varying programming backgrounds contribute code is not conducive to maintaining the necessary clarity. The completion of a major milestone like the Mauna Kea deployment provides an opportunity to review the work completed and make changes to guarantee its continued functionality. It is highly recommended that the complete code-base to date be audited to catalogue and address outstanding issues such as incomplete functionality (§6.2.2.8, §6.3.2.1), inconsistent implementations (§6.2.2.6, §6.2.2.7, §6.3.2.3, §6.4.2, §6.5.2, §6.9.2.2, §6.11.2.3) and general bad coding practises.

7.2.6 Make *Logger* More Efficient

Logger writes a significant amount of data to the SSD (§5.10.3.1). Write throughput is a limitation of SSDs like the ones used in the ASUS S100 Eee PC (§5.10). Redesigning *Logger* to compress data in-line before storage will reduce the bottleneck on the SSD without increasing the risk of data loss. Redirecting the data flow to a separate drive, such as the integrated removable SD card or another external drive would both isolate the throughput from the rest of autonomous control system as well as make it more portable.

7.2.7 Reevaluate the Operating System Requirements

Some of the devices in the autonomous control system restrict the selection of the operating system (Table 7.1). If there comes a time when these devices are no longer part of the autonomous control system, either as a result of changes recommended in this report or in project focus, it is recommended that the autonomous control system be moved away from Microsoft Windows towards a more light-weight operating system. Such an operating system would release computing overhead to the autonomous control algorithms. This would reduce the bottlenecks that are already beginning to appear when running intensive services such as streaming video (§6.13). Possibilities include customized Linux installations or real-time operating systems such as QNX.

Table 7.1: Autonomous control system devices requiring Microsoft Windows.

Device
Omega OMB-DAQ-3001
Sensors & Software OEM Noggin NIC GPR
Lantronix UBox 2100
Trimble GPS Pathfinder ProXRT
Microsoft LifeCam VX-3000 Cameras

Chapter 8

Conclusions

Developing multiagent Autonomous Site Selection and Preparation (ASSP) was an ambitious goal in a very limited time frame. The In-Situ Resource Utilization (ISRU) project made great progress developing the necessary platform and algorithms and successfully tested them during an analogue deployment on Mauna Kea, Hawai‘i from January 24^{th} to February 13^{th}, 2010.

While some devices were problematic, the majority have functioned as desired with only minor problems. Once identified, these problems were avoidable with careful design and operation. This has left the project with a mostly-solid system of hardware devices for future work (Table 8.1).

The software of the autonomous control system was based around the ArgoSoft framework developed by UTIAS for previous platforms. It has continued to prove to be a flexible and robust solution to the needs of an autonomous mobile platform. Its modular nature allowed for the seamless implementation of three different digging controllers as well as a GPR controller. Of the three digging controllers tested during the Mauna Kea deployment, the hand-coded controller *C3* demonstrated the best results. The research value of the evolved controllers *C1* and *C2* cannot be discounted however and the framework remains in place to test them. *GPRSurvey* succeeded functionally; however, close work with Xiphos Technologies Inc. will be needed to redesign the data requirements to be more realistic of autonomous surveying.

Altogether, the work done preparing for and during the Mauna Kea deployment leaves the ISRU project in an excellent position for future research. While there remains some areas that do require attention and reconsideration, this can be expected considering the ambitious scope and short time line. Complete multiagent autonomous mobile excavation platforms were designed and constructed in approximately a year.

The work has successfully proved the feasibility of using multiagent rover teams to perform ASSP tasks. It provides the CSA and project partners with the technology and opportunity to solidify a unique advanced foothold in the area of ISRU and ASSP. It should lead to exciting future results in multiagent mobile robotics, ISRU, ASSP and many other fields.

Table 8.1: Summary of autonomous control system device evaluations.

Device	**Evaluation**	**Section**
RoboteQ AX2850	Conditional Yes	5.1
Omega OMB-DAQ-3001	No	5.2
Sensors & Software OEM Noggin NIC GPR	Conditional Yes	5.3
Musketeer Interface Box	Yes	5.4
Cisco SD208 Ethernet Switch	Strong Yes	5.5
Perle IOLAN SDS4	Conditional Yes	5.6
Lantronix UBox 2100	Conditional Yes	5.7
Musketeer Control Box	Yes	5.9
ASUS S100 Eee PC	Strong Yes	5.10
Hawking HWUG1 Wireless Network Card	Conditional Yes	5.11
Hawking HSB2 Wireless Booster	Strong Yes	5.11
MicroStrain 3DM-GX2 AHRS	Yes	5.14
u-blox EVK-5H GPS	Yes	5.15
Trimble GPS Pathfinder ProXRT	Conditional Yes	5.16
QualityKits VK011 Serial Temperature Sensor	Conditional Yes	5.17
SICK LMS111 Laser Measurement System	Yes	5.19
DirectedPerception PTU-D47	Yes	5.18
Microsoft LifeCam VX-3000 Camera	Yes	5.20

Bibliography

[1] G. B. Sanders, W. E. Larson, K. R. Sacksteder, and C. A. McLemore, "NASA In-Situ resource Utilization (ISRU) project - development & implementation," in *AIAA SPACE 2008 Conference*. American Institute of Aeronautics and Astronautics, 1801 Alexander Bell Drive, Suite 500, Reston, VA, 20191-4344, USA,, 2008.

[2] R. P. Mueller and I. Townsend, "Lunar regolith simulant feed system for a hydrogen reduction reactor system," in *47 th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition(Disc 1)*. American Institute of Aeronautics and Astronautics, 1801 Alexander Bell Drive, Suite 500, Reston, VA, 20191-4344, USA,, 2009.

[3] G. Neff, *Apollo 12 Video Library*, National Aeronautics and Space Administration (NASA), 1999, http://history.nasa.gov/alsj/a12/.

[4] M. Viel, *Memorandum Regarding Rover Motor Problems*, NORCAT, 1545 Maley Drive, Sudbury Ontario, Canada, 2009.

[5] *Telephone conversation with the author*, NPC Robotics, 4851 Shoreline Drive, PO Box 118, Mound MN, 55364, November 17 2009.

[6] *WAVESERIES WAS5 PRO Bridge Datasheet*, Weidmüller Interface GmbH & Co. KG, Klingenbergstraße 16 D-32758 Detmold Germany.

[7] *SCADA Distribution Box Electrical Drawing*, NORCAT, 1545 Maley Drive, Sudbury Ontario, Canada, May 2009.

[8] *Design Guide & Applications Manual For Maxi, Mini, Micro Family DC-DC Converter and Accessory Modules*, Vicor Corporation, 25 Frontage Road Andover, MA, USA 01810.

[9] M. Viel, *Memorandum Regarding Repairing Isolation Problem on the Plow Electronics*, NORCAT, 1545 Maley Drive, Sudbury Ontario, Canada, November 20 2009.

[10] *RoboteQ AX2850 Digital Motor Controller User's Manual*, v1.9b ed., RoboteQ, 8426 E. Shea Blvd. Scottsdale, AZ 85260 - USA, June 2007.

[11] *RoboteQ AX2850 Read Me: Cabling Instructions*, V91013 ed., RoboteQ, 8426 E. Shea Blvd. Scottsdale, AZ 85260 - USA.

[12] *Rover Chasis Electrical Drawing*, NORCAT, 1545 Maley Drive, Sudbury Ontario, Canada, May 2009.

[13] *OMB-DAQ-3000 Series User's Manual*, rev 1.1 ed., OMEGA Engineering Inc., 976 Bergar, Laval Quebec, H7L 5A1, Canada.

[14] *Telephone conversation between Francis Frenzel and Adam Fazzari*, Sensors & Software Inc., 1040 Stacey Court Mississauga, Ontario, Canada.

[15] *1450 Case Specifications*, Pelican Products, Inc., 23215 Early Avenue, Torrance, CA 90505.

[16] *Lantronix UBox EOL Notice*, Lantronix, 15353 Barranca Parkway, Irvine, CA 92618, USA, October 2009.

[17] *GPS Pathfinder ProXRT receiver Datasheet*, Trimble Navigation Limited, 10355 Westmoor Drive, Suite 100, Westminster, CO 80021, USA.

[18] *FCC Online Table of Frequency Allocations*, Federal Communications Commission Office of Engineering and Technology Policy and Rules Division, 445 12th Street SW, Washington, DC 20554.

[19] *DS18S20 High-Precision 1-Wire Digital Thermometer*, Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, California, 94086, April 2008.

[20] *Conputer Controlled Pan-Tilt Unit Model PTU-D47 User's Manual*, 2nd ed., DirectedPerception, 890C Cowan Road, Burlingame, California 94010,, March 2007.

[21] *Laser Measurement Systems of the LMS100 Product Family Operating Instructions*, SICK AG Waldkirch, Nimburger Straße 11, 79276 Reute, Germany, 01.

[22] *Telephone conversation between the author and Ron Stahl*, SICK, Inc., 6900 West 110th Street Minneapolis, MN 55438 USA, December 18 2009.

[23] *Microsoft LifeCam VX-3000 Technical Data Sheet*, 0908th ed., Microsoft Corporation, 2009.

[24] K. Law, "Development of a test platform and control system to demonstrate autonomous excavation for space exploration applications," Master's thesis, University of Toronto Institute for Aerospace Studies, 2010.

[25] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots.* A Bradford Book, The MIT Press, 2004.

[26] *GeographicLib*, http://geographiclib.sourceforge.net/.

[27] *ISRU GPR Server ICD*, Xti-1176-1041 ed., Xiphos Technologies Inc., 3981 boul. St-Laurent, Suite 500, Montréal, QC H2W 1Y5, May 2009.

[28] *Noggin GPR Course*, Sensors & Software Inc.

[29] J. Thangavelautham, A. Smith, D. Boucher, J. Richard, and G. D'Eleuterio, "Evolving a scalable multirobot controller using an artificial neural tissue paradigm," in *2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 77–84.

[30] *Sapphire Technology Datasheet*, Zebra Enterprise Solutions, 1000 Broadway, Ste. 150, Oakland, CA 94607 USA.

[31] *Universal Serial Bus Specification*, 2nd ed., April 2000.