

Validation of the UBC Powered Upper Limb Orthosis Simulator

by

Daryoush Sahebjavaheer

BASc., Simon Fraser University, 2009

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Master of Applied Science

in

THE FACULTY OF GRADUATE STUDIES
(Mechanical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

April 2012

© Daryoush Sahebjavaheer, 2012

Abstract

Paresis in the upper limb is a condition that leaves the users with flail or severely weakened arms, prohibiting them from being able to perform common everyday tasks such as reaching and grasping objects, eating, drinking, cooking, performing personal hygiene, etc. The UBC Powered Upper Limb Orthosis (UBC-PULO) is a wearable, highly-functional assistive device that supports and restores the functions necessary to one entire arm, enabling users to perform high-priority daily living tasks. Previous work towards achieving this goal has included: developing user specifications, identifying desired tasks and arm motions required to perform them, establishing design specifications and evaluation criteria, developing a control strategy along with hardware and software for the device, and completing the construction of an orthosis prototype with user control interfaces. Recently, a virtual reality simulation environment (VRSE) has been developed for the UBC-PULO which models the control system and electro-mechanical exoskeleton. The VRSE is a tool that can be used for optimization of the control system and testing current or new exoskeleton designs, and has the potential for screening and training of potential users but only if it properly mimics the real physical orthosis device. Therefore, the final objective of this research is to validate the VRSE against the physical prototype. The steps needed to achieve this final objective in the UBC-PULO project include reviewing the state of the current prototype, performing work to commission the device and interfaces, development and implementation of an experimental setup to allow for validation, development of representative control input

parameters for proper simulation of the physical device, and testing to validate the VRSE against the physical prototype. The research work described in this thesis responds to these needs by performing a review of the current electrical and mechanical state of the prototype, proposing and implementing a range of design improvements necessary for commissioning the prototype system, development and implementation of a completely new electronic platform along with the necessary software, and performing tests to validate the VRSE.

Table of Contents

Abstract	ii
Table of Contents	iv
List of Tables	ix
List of Figures	xi
Acknowledgements	xvii
Chapter 1 - Introduction	1
Chapter 2 - Literature Review	6
2.1 Introduction.....	6
2.2 Review of current assistive devices	8
2.2.1 Other differentiating aspects	10
2.2.2 Conclusion	12
2.3 Previous development of the UBC Powered Upped Limb Orthosis (UBC-PULO) 14	
2.3.1 Introduction	15
2.3.2 Design and development	19
2.4 Review of UBC-PULO Virtual Reality Simulation Environment (VRSE).....	33
2.4.1 Introduction	33
2.4.2 Implementation of the high level controller.....	35
2.4.3 Drive system and PID controllers	36
2.4.4 Physical model.....	37
2.5 Summary	39

Chapter 3 – Experimental Setup	42
3.1 Introduction	42
3.2 Mechanical review	44
3.2.1 Evaluating the PULO drive system.....	44
3.2.4 Mechanical review summary	56
3.3 Electrical review	56
3.3.1 Evaluating the PULO electrical system	56
3.3.2 Electrical review summary.....	59
3.4 Summary	60
Chapter 4 – Upgrading the UBC-PULO Hardware	61
4.1 Introduction	61
4.2 Mechanical improvements	61
4.3 Development of a new UBC-PULO Electrical system.....	62
4.3.1. User control interfaces	62
4.3.2. Feedback angle sensors	64
4.3.3. Design requirements	69
4.3.4. Design overview	70
4.3.5. Selection of SBC board and DAQ cards	75
4.3.6. Setup of the SBC board and DAQ cards for XPC Target.....	77
4.3.7. The new electrical system	78
4.4 Conclusion	80
Chapter 5 – Developing UBC-PULO Software in XPC Target	82

5.1 Introduction	82
5.2 UBC-PULO software	82
5.2.1 Task control algorithm	84
5.2.2 Inverse kinematics	88
5.2.3 Low level controller and plant	91
5.2.4 New UBC-PULO software features	93
5.3 Software for experimental setup	99
5.4 Summary	103
Chapter 6 – Validation of the VRSE	105
6.1 Introduction	105
6.2 Test results	106
6.2.1 Experiment 1- PID response	106
6.2.2 Experiment 2- performing tasks	109
6.3 Conclusion	126
Chapter 7 – Discussion and Limitations	127
7.1 Introduction	127
7.2 Limitations	128
7.2.1 Evaluating the UBC-PULO drive system	128
7.2.2 Modelling of the UBC-PULO prototype	129
7.2.3 Limitations of the physical prototype	131
7.2.4 Limitations of VRSE validation	132
7.3 Using the experimental setup as a platform for future development	133
7.4 Conclusion	134

Chapter 8 – Conclusion and Recommendations	136
8.1 Introduction.....	136
8.2 Conclusions	137
8.2.1 Literature review.....	137
8.2.2 Reviewing the UBC-PULO prototype	138
8.2.3 Experimental setup	138
8.2.4 Validating the VRSE.....	141
8.2.5 Research conclusions	141
8.3 Future recommendations.....	143
References	146
Appendix A– UBC-PULO Forward and Inverse Kinematics	150
A.1 Forward kinematics	150
A.2 Inverse kinematics.....	152
A.3 Parametric kinematics development in MuPAD.....	154
Appendix B– Flex Shaft Apparatus	161
B.1 Test apparatus	161
B.2 Flex shaft casing solutions	163
Appendix C– Further Assessment of the UBC-PULO	165
C.1 Shoulder.....	165
C.2 Forearm	167
C.3 Grasp	167
C.4 Weight and comfort.....	168

C.5 Forearm redesign.....	169
C.6 Prioritized list of mechanical improvements	171
C.7 Prioritized list of electrical improvements	172
Appendix D – Host and Target Computer Setup	173
D.1 Host computer.....	173
D.2 Target computer.....	176
Appendix E – Custom PCB Design.....	180
E.1 Selected electrical components for custom PCB.....	180
E.1.1 Motor driver IC	180
E.1.2 Level shifter	186
E.1.3 Current sensor IC small with no external components.....	187
E.1.4 Power regulators and batteries.....	188
E.1.5 LEDs.....	190
E.1.6 DIP switches.....	191
E.1.7 Connectors	191
E.2 PCB features and theory of operation	192
E.3 PCB design consideration and manufacturing	201
E.4 Custom Board schematic diagrams and PCB layouts	207
E.4.1 Motor driver board schematic diagram	207
E.4.2 Motor driver board PCB layout.....	213
E.4.3 Backplane I/O and Power board schematic.....	221
E.4.4 Backplane I/O and Power board PCB layout	224
E.8 Bill of material.....	231

List of Tables

Table 1. Annual US Statistics.....	7
Table 2. Summary of Exoskeleton Robots in Literature.	13
Table 3. Defined User Characteristics.....	15
Table 4. Top Priority Tasks [2]	16
Table 5. High Level Controller.....	23
Table 6. Electrical System Components.....	28
Table 7. Motor Torque Evaluation.....	47
Table 8. Joints Summary.....	49
Table 9. New Electrical System Design Requirements	70
Table 10. New Electrical System Features.....	80
Table 11. PID Constant Trade-off	106
Table 12. Sample Azimuth Step Response.....	107
Table 13. Joint Angle Limits Test	111
Table 14. End-effector Position Error Throughout Tasks 1 to 12	117
Table 15. Table Top Mode End-effector Position Error (Tasks 1 to 6)	119
Table 16. Functional Mode End-effector Position Error (Tasks 7 to 12).....	120
Table 17. Model End-effector Position Error Throughout Tasks 1 to 12.....	120
Table 18. Physical Device Joint Angle Errors Throughout Tasks 1 to 12.....	122
Table 19. Model Joint Angle Error Throughout Tasks 1 to 12	122
Table 20. Upper Bound Error between Active Joints of Model and Physical Prototype	123
Table 21. Orientation Mode Joint Angle Error (Physical Device).....	124

Table 22. Orientation Mode Joint Angle Error (Model)	124
Table 23. Upper Bound Error between Model and Physical Prototype.....	126
Table 24. DH Parameters.....	151
Table 25. Vendors for Flex Shaft with Casings	164
Table 26. Prioritized List of Mechanical Improvements	171
Table 27. Prioritized List of Electrical Improvements.....	172
Table 28. Important PCM 3362 Connectors	177
Table 29. Typical Current Consumption.....	189
Table 30. Bill of Material for Custom PCBs	231

List of Figures

Figure 1. Options Available to Patients with Upper Limb Paresis.....	1
Figure 2. Degrees of Freedom in the Human Arm	17
Figure 3. UBC-PULO Prototype	17
Figure 4. Control Interfaces.....	18
Figure 5. Neutral Zone	19
Figure 6. UBC-PULO Conceptual Design	21
Figure 7. Axis of Rotation for each DOF.....	22
Figure 8. Control System.....	22
Figure 9. UBC-PULO Modes of Operation State Machine Diagram.....	25
Figure 10. UBC-PULO Control Logic Flowchart	27
Figure 11. a) Head Interface Inclinometers b) Inclinometer Dimensions (inches)	29
Figure 12. a) Schematic of Shoulder User Interface at LOW, MID and HIGH Positions b) Photo of the Shoulder User Interface without Cover	30
Figure 13. a) Rotary Potentiometer b) Potentiometer Electrical Representation	31
Figure 14. a) High Level Control Board b) Low Level Control and Motor Driver Board	32
Figure 15. Electrical System.....	33
Figure 16. Overview of the UBC-PULO Virtual Reality Simulator.....	34
Figure 17. Sample Drive System and PID Controller Block [1].....	36
Figure 18. Second Order Motor Model [1].....	37
Figure 19. Solid Works Assembly of UBC-PULO Prototype [1].....	38
Figure 20. SimMechanics model of UBC-PULO Prototype [1].	39
Figure 21. Experimental Setup for VRSE Validation	43

Figure 22. Simplified Moment of Inertia Calculation	46
Figure 23. Test Apparatus Overview	51
Figure 24. Physical Test Apparatus.....	53
Figure 25. Screen Shot of the Test Apparatus User Interface.....	54
Figure 26. Sample Second-order Curve Fit.....	54
Figure 27. a) VTI Technologies SCA1020 b) Electrical Connection Diagram.....	63
Figure 28. a) Gravitational Acceleration Y-axis and Z-axis.....	64
Figure 29. a) AS5045 Electrical Pin-out b) Contactless Magnet Angle Detection	66
Figure 30. Encoder Housing Design	67
Figure 31. a) 1:1 Acetal Gears with Press Fit Magnet b) Encoder Housing (Top).....	68
Figure 32. XPC Target System	71
Figure 33. a) PC/104-Plus Form Factor b) PC/104 Modules Stacked.....	73
Figure 34. Electrical System Design Overview.....	74
Figure 35. PCB Design Overview.....	74
Figure 36. Selected SBC Board (on left) and DAQ Cards.....	76
Figure 37. a) New UBC-PULO Electrical System b) Assembled Backplane PCB Board and Motor Driver PCB Board	79
Figure 38. UBC-PULO Software Overview.....	84
Figure 39. Task Control Algorithm (Part 1).....	85
Figure 40. Task Control Algorithm (Part 2).....	87
Figure 41. Inverse Kinematics.....	89
Figure 42. Inverse Kinematics Calculations	90
Figure 43. Inverse Kinematics Stability and Physical Plant.....	91

Figure 44. Simplified Physical Plant	92
Figure 45. a) Sample Discrete PID Loop And Saturation Block b) The Motor Driver Block Which Commands The Motor Driver PCB	93
Figure 46. Condition Limit Block.....	96
Figure 47. User Adjustable Functional Mode	97
Figure 48. User Adjustable Functional Mode	98
Figure 49. Plant Simulation Overview	100
Figure 50. UBC- PULO Plant Choices.....	101
Figure 51. UBC-PULO Visual Representation.....	103
Figure 52. Azimuth Step Response.....	107
Figure 53. All Joint Step Responses	108
Figure 54. Switching Operation Modes Using Shoulder Interface	109
Figure 55. User Control Interfaces vs. System Control Signals.....	110
Figure 56. Physical Device Joint Angles vs. Condition Number	112
Figure 57. Test Tasks for Position Mode of Operation	114
Figure 58. End-effector Position vs. Time	116
Figure 59. Table Top Mode End-Effector Position (Tasks 1 to 6).....	118
Figure 60. Functional Mode End-effector Position (Tasks 7 to 12).....	119
Figure 61. Model vs. Physical End-effector Position (Tasks 1 to 12)	120
Figure 62. Joint Angles Throughout Tasks 1 to 12.....	121
Figure 63. Joint Angles Throughout Tasks 1 to 12.....	125
Figure 64. Test Apparatus LabVIEW Diagram	162
Figure 65. a) Harmonic Actuator [48] b) Harmonic Gear (Strain Wave Gearing) [49]	166

Figure 66. Grasp Mechanism	167
Figure 67. UBC-PULO prototype top-view	168
Figure 68. Considered Forearm Redesign Concepts	169
Figure 69. Proposed Forearm Design	170
Figure 70. Host PC TCP/IP Address	173
Figure 71. Screen Shot of XPC Explorer Window	175
Figure 72. Advantech PCM 3362 Connectors with Component Side on Left and Back Side on Right (for More Details Visit Datasheet)	176
Figure 73. PCM 3362 BIOS Screen Shot	179
Figure 74. Texas Instrument's PowerPad Technology Allows Heat Dissipation from the Bottom of the IC.	181
Figure 75. Simplified DRV8412 Connection Diagram with Output Stage Half-H-Bridges Marked in Red Boxes	182
Figure 76. MOSFET Full-H-bridge (Marked in Black) Composed of Two Half-H-Bridges (Marked in Red-black)	183
Figure 77. a) DRV 8412 Output PWM b) Input PWM Duty Cycle v.s. Output PWM Duty Cycle	184
Figure 78. a) Allegro Microsystems ACS712 b) Output Voltage V.S Sensed Current	188
Figure 79. a) Texas Instruments PTR08060W (33 Watts) b) TDK-LAMBDA AMERICAS INC CC6-0512SR-E (6 Watts)	190
Figure 80. WAGO 233 Series Terminal Blocks	191
Figure 81. Backplane I/O and Power PCB (Top View)	193
Figure 82. Motor Driver PCB Top Side (Top View)	194

Figure 83. Motor Driver PCB Bottom Side (Top View)	195
Figure 84. Schematic Diagram for The DRV 8412 Motor Driver IC	198
Figure 85. Level Shifter Schematic	199
Figure 86. Current Sensor Schematic	200
Figure 87. Power Regulator Schematic.....	201
Figure 88. Texas Instruments Recommends Solid Vias for PowerPAD	203
Figure 89. Screen Shot of Motor Driver Board PCB Layout Showing Top Side Components and All Six Layers	204
Figure 90. Screen Shot Backplane Board PCB Layout Showing Top Side Components and All Six Layers	205
Figure 91. PCB Layout of the Motor Driver Board's Top and Bottom Layer	206
Figure 92. Schematic Diagram for the Motor Driver Board -Sheet 1/5	208
Figure 93. Schematic Diagram for the Motor Driver Board -Sheet 2/5	209
Figure 94. Schematic Diagram for the Motor Driver Board -Sheet 3/5	210
Figure 95. Schematic Diagram for the Motor Driver Board -Sheet 4/5	211
Figure 96. Schematic Diagram for the Motor Driver Board -Sheet 5/5	212
Figure 97. Screen Shot of the Motor Driver Board PCB Layout showing Top Layer (Layer 1) with Top Side Components.....	213
Figure 98. Screen Shot of the Motor Driver Board PCB Layout showing Bottom Layer (Layer 6) with Bottom Side Components.....	214
Figure 99. Screen Shot of the Motor Driver Board PCB Layout showing Top Layer (Layer 1).....	215
Figure 100. Screen Shot of the Motor Driver Board PCB Layout showing Layer 2	216

Figure 101. Screen Shot of the Motor Driver Board PCB Layout showing Layer 3	217
Figure 102. Screen Shot of the Motor Driver Board PCB Layout showing Layer 4	218
Figure 103. Screen Shot of the Motor Driver Board PCB Layout showing Layer 5	219
Figure 104. Screen Shot of the Motor Driver Board PCB Layout showing Bottom Layer (Layer 6).....	220
Figure 105. Schematic Diagram for the Backplane Board Sheet 1/2	222
Figure 106. Schematic Diagram for the Backplane Board Sheet 2/2	223
Figure 107. Screen Shot of Backplane Board PCB Layout Showing Top Layer (Layer 1) and Top Side Components	224
Figure 108. Screen Shot of Backplane Board PCB Layout Showing Top Layer (Layer 1)	225
Figure 109. Screen Shot of Backplane Board PCB Layout Showing Layer 2.....	226
Figure 110. Screen Shot of Backplane Board PCB Layout Showing Layer 3.....	227
Figure 111. Screen Shot of Backplane Board PCB Layout Showing Layer 4.....	228
Figure 112. Screen Shot of Backplane Board PCB Layout showing Layer 5	229
Figure 113. Screen Shot of Backplane Board PCB Layout showing Bottom Layer (Layer 6).....	230

Acknowledgements

I would like to thank Dr. Douglas P. Romilly for his continuing guidance and efforts throughout the duration of this project. I would like to show my gratitude to Glenn Jolly for his excellent practical advice for PCB design. I would like to thank Terence Woo for his contributions in the design of the joint angle sensors, evaluation of the flex shafts and help towards the proposal of the new forearm design. Thanks also to Dr. Daniel Gelbart for sharing his practical knowledge, which was tremendously helpful for the forearm redesign. I am grateful for my brother Ramin who not only gave me great advice, but also made my overall university experience at UBC more pleasing through wonderful lunch breaks and squash sessions. And finally, and most importantly, a special thanks to my parents for their unconditional support throughout the years of my education.

Chapter 1 - Introduction

The incidence of upper limb disability in the US is in the hundreds of thousands every year. The victims suffer from partial or complete loss of arm function (i.e. paresis) due to disease or injury – e.g. from Stroke, Amyotrophic Lateral Sclerosis (ALS), Muscular Dystrophy (MD) and Spinal Cord Injury (SCI) being a few examples [2] and are unable to perform every-day activities [2], [3] such as:

- Reaching and picking up objects
- Personal hygiene
- Hobbies and crafts
- Eating and drinking
- Housework

Thus, people with a disability in their upper limbs are highly dependent, have less privacy, and are a social and economical burden to their friends and family. Overall, they have a *lower quality of life* and thus the need for an assistive device is evident. An assistive device is highly beneficial to the users by enabling them to perform basic every-day tasks such as eating, drinking and reaching for objects, allowing them to achieve a higher degree of independence. Figure 1 presents the four options available to a person with upper limb paresis:

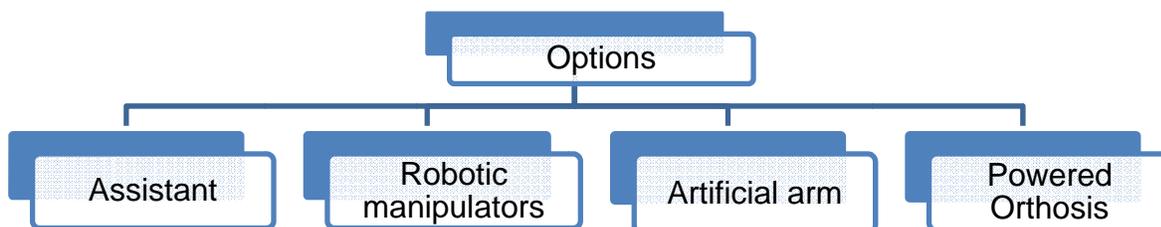


Figure 1. Options Available to Patients with Upper Limb Paresis

The first option is to hire a nurse or social worker as an assistant to do tasks for the physically-challenged person. Having an assistant is desirable to some patients as he or she also provides companionship; however, an assistant can also impose on privacy, is an economic burden to the health care system, and isolates the patients from the self-gratifying pleasure of doing tasks themselves. The second option is to acquire a robotic manipulator, i.e. a robotic unit that performs a set of programmed tasks requested by the user. Although robotic manipulators maintain the user's privacy, they still operate as a separate entity cutting off the user from the sense of self-gratification and reward in doing tasks themselves. The third option is an artificial arm, a robotic arm that physically and functionally replaces the arm and is directly controlled by the user. While an artificial arm is a good option for someone who has physically lost both arms, it is a disadvantageous for persons with upper limb paresis for two main reasons: First, it requires the amputation of the entire arm, and second, it deprives the users from utilizing their own tactile sensation (i.e. and all the benefits of an individual's personal touch). A powered orthosis is an electro-mechanical exoskeleton that externally supports the arm and allows users to directly control their upper limb through control interfaces. Finally, a powered orthosis allows users to utilize their own arm and remaining tactile sensation to perform tasks themselves. In addition, a powered orthosis also maintains the users' privacy and sense of self-gratification by enabling them to perform tasks themselves using their own arm. An added benefit is the continuous mobility of the existing arm which also increases circulation and maintains joint function, while decreasing atrophy of the existing tissues, muscles, bone mass within the affected limb.

A Powered Upper Limb Orthosis (PULO) is being developed at UBC with the goal of constructing a highly functional, portable, user-friendly assistive device that enables the user to perform daily living tasks via worn user control interface devices. The UBC-PULO is targeted at users with severely impaired or lost motor function in both upper limbs who are of adult size, have intact sensation, show full range of joint motion, demonstrate full cognitive abilities and are free of spasticity in their upper limbs. Previous work toward achieving this goal has included: developing user specifications, identifying desired tasks and arm motions required to perform these tasks, establishing design specifications and evaluation criteria, developing a control strategy along with hardware and software for the device, and completing the assembly of an orthosis prototype with user control interfaces [1], [2], [4–8].

In addition, a virtual reality simulation tool (VRSE) [1] was developed for the UBC-PULO prototype in parallel to this research work. The VRSE is capable of simulating the performance of the UBC-PULO prototype with respect to the overall control system and dynamic characteristics of the electro-mechanical exoskeleton to serve as a research platform that allows:

- Optimization and testing of the control system.
- Optimization and testing of the exoskeleton design.
- Screening and monitoring tool for potential users to practice operation of the device.
- Evaluation and testing of future designs before physical development.

Based on the existing control system and device model incorporated in the VRSE framework, to allow the model to properly mimic the performance of the physical device, the following research questions need be answered:

- What are the available parameters required as inputs to the VRSE that can be used to tune the system?
- What elements in the physical device are likely to be poorly represented by assumed values of these inputs in the existing VRSE model?
- What steps are needed to quantify these inputs for the identified elements?
- What steps are required to make the UBC-PULO prototype suitable for validation testing?
- What tests are needed to accomplish this validation?
- What are the Limitations of the VRSE in properly modelling the physical prototype?

The objective of this research is to validate the UBC-PULO virtual reality simulation model (VRSE) [1]. To achieve this objective, the following tasks were completed:

1. Literature review of similar devices.
2. Understanding and reviewing the current state of the orthosis.
3. Evaluating the experimental setup, identifying system parameters and determining the required high-priority improvements.
4. Upgrading the UBC-PULO hardware to: (1) satisfy the required high-priority improvements, (2) allow for characterization of the physical prototype and (3) to allow validation of the VRSE.

5. Developing the UBC-PULO software to allow for validation of the VRSE.
6. Commissioning the experimental setup.
7. Validating the UBC-PULO Simulator using the experimental setup.

The following thesis chapters describe in detail on how each of the identified tasks was completed and highlight its contribution to the UBC PULO project. Chapter 2 investigates both current and previously developed assistive devices and thoroughly explores the design and development of the UBC-PULO project. The review of the current state of the UBC-PULO (i.e. the experimental setup) and recommendations for the required high-priority hardware changes are discussed in Chapter 3. Chapter 4 presents the development and implementation of new hardware upgrades required to both commission the experimental setup and to validate the VRSE. Chapter 5 entails the development of the UBC-PULO software in the Simulink and XPC Target environment, which acts as a framework to allow validation of the VRSE. Results on validation of the VRSE and characterization of the physical system are presented in Chapter 6. Chapters 7 and 8 provide conclusions based on this research work and recommendations for future work on the UBC-PULO project, respectively.

Chapter 2 - Literature Review

2.1 Introduction

There are numerous cases of severe upper limb paresis occurring on an annual basis all over the world [3], [4], [9–14]. Table 1 depicts the number of annual cases of partial or full upper limb paresis due to disease or injury in the US alone. Every year more than 795,000 strokes occur in the United States [4] while there are more than 6.4 million stroke survivors in the population. Approximately 85% of stroke survivors experience upper limb paresis, with 40% facing chronic impairment [14]. According to Joel Stein *et al.* [13]:

“Although most patients regain their walking ability, 30–66% are no longer able to use the affected arm functionally. It has been found that only 5% of individuals who receive intensive therapy for severe upper-extremity weakness after stroke regain functional use of the upper extremity during the course of rehabilitation.”

Similarly, diseases such as Amyotrophic Lateral Sclerosis (ALS), Muscular Dystrophy (MD) and Multiple Sclerosis (MS), which eventually leave the person with non-functioning (i.e. flail) arms, account for more than 15000 annual cases of upper limb paresis in the US alone [3], [9–13]. Furthermore, C5, C6 or C7 spinal cord injury (SCI) results in partial or complete loss of arm function, which accounts for more than 2750 annual cases in the US.

Table 1. Annual US Statistics.

Disease/Injury	Description	(Annual incidence in the US)
Stroke	Rapid loss of brain function due to disturbance in blood supply to the brain	700,000
ALS	Amyotrophic Lateral Sclerosis- Weakens muscles, flail arm will always result	5000
MD	Muscular Dystrophy- affects upper limb by reduced shoulder and grip strength	500 – 600
SCI	Spinal Cord Injury – e.g. C6, C5 injury partial hand function is lost, neck, shoulder work. Sensation might be full, partial, or none.	11,000
MS	Multiple Sclerosis – overwhelming fatigue and poor coordination	9600

Hence, it is anticipated that a significant number of people with upper limb paresis are in need of an assistive device that enables them to do everyday tasks. Despite the need, there have been relatively few attempts made to develop a feasible assistive device, especially one that can provide powered upper limb assistance. Most of the devices currently available in the market or under development are designed for rehabilitative purposes rather than assistive purposes. The focus of this literary review is on the state-of-the-art of assistive exoskeleton devices rather than rehabilitative devices or prosthesis devices which target a different set of users.

This chapter serves to provide a comprehensive comparison between the UBC-PULO and the current state-of-the-art power assistive devices in terms of available technologies, functionality, target market and overall design. First, the review will explore current research and commercial products available and evaluate the suitability

for the defined user for the UBC-PULO (Section 2.2). Next, the goals, design, development and current state of the UBC-PULO will be described in more detail for comparison (Section 2.3). The knowledge gained in the literature review served as a guide in the design and development process of the of the UBC-PULO project. Key aspects which differentiate the UBC-PULO project form other available designs are also discussed in this section.

2.2 Review of current assistive devices

This section reviews the current assistive devices. For an excellent review on the state of the art in rehabilitative upper-limb robots refer to [15].

Early research on assistive and rehabilitative robot focused on end-effector (i.e., manipulator) robots. Although this type of robot may be simpler in design and implementation, the joints of the manipulator will not match that of the human limb, leading to uncontrolled load transfer between upper limb joints and a combination of undesired movements at the wrist, elbow and shoulder joints. Examples of end-effector robots for rehabilitation purposes are MIME [16] and GENTLE/s [17]. Jaco developed by Kinova Inc. (Quebec Canada) [18] is an example of a recent manipulator wheelchair mounted robot with 6 DOF and 3 fingers that is operated via a joystick.

Recently, robotic research for assistive and therapy has shifted towards exoskeleton robots. An Exoskeleton is a structure with joints and linkages that resemble and match the upper limb of a human in terms of joints and linkages. Exoskeletons are able to closely follow the motion of the human upper limb, and therefore can be attached to the upper limb – this allows the exoskeleton to more fully determine the

upper limb posture and to apply controlled torques to each joint (i.e. DOF) separately. According to Lo *et al.* [15] a wider range of motion may be possible with exoskeleton robots compared to end-effector robots.

At the moment, there exists no exoskeleton robot that can actively actuate all DOF of the upper limb [15]. Most researchers have developed upper limb exoskeletons which cover from shoulder to wrist joints without including DOF for finger movements. That being said, there have been number of commercially available powered upper limb robots. Most of them are designed for therapy purposes with some also capable of assistive function. Historically, the development of the first powered upper limb orthosis began in the 1960's at Rancho Los Amigos Hospital with the Rancho Los Amigos Orthosis (RLAO) by Nickel *et al.* [19]. The first wheel-chair mounted powered orthosis had six degrees-of-freedom (DOF) plus grasp thus targeting persons with severe upper limb paresis, and was capable of performing basic human arm motions except the wrist yaw. This device used a bi-directional tongue switch as the control interface to sequentially activate each joint. ArmeoPower is a 7 DOF robot based on the ARMin III exoskeleton [20] manufactured by Armeo products (Hocoma, AG, Switzerland). mPower armbrace (Mymo Inc., Cambridge, MA) [13] is a assistive single DOF portable arm brace. Assistive torques are generated for the elbow from electromyogram (EMG) signals measured from biceps and triceps muscles. Another commercially available robot is the Hand Mentor (Kinetic Muscles Inc., Tempe AZ), a 1 DOF wearable device for the rehabilitation of the wrist and fingers which provides force, position and EMG feedback and is actuated by an air muscle. The Robot Suit HAL-5 (CYBERDYNE Inc., Japan) is a full body exoskeleton [21], [22] which uses EMG signals from the user to

generate assistive torques. The full-body exoskeleton powers all the upper and lower limbs of the body using DC motors designed for either human performance augmentation (i.e. improving the physical capability of disabled person) and/or rehabilitation. Among many exoskeletons, the IntelliArm has implemented movements for shoulder, elbow and wrist [23]. Sarcos Master Arm (Sarcos, Inc., Salt Lake City, USA) [24] is a seven DOF hydraulically actuated exoskeleton robot arm. Its anthropomorphic design mimics the major seven DOFs of the human arm. The MGA [25] and ARMin III [20] have implemented an actuated DOF for shoulder elevation/depression. Johnson *et al.* [26] developed the wheelchair mounted five DOF Motorized Upper-Limb Orthotic System (MULOS), which employed a four DOF joystick as the control interface. IKO [27] is a wearable upper limb exoskeleton oriented to increasing human force during routine activity at the workplace. To allow for an overall more ergonomic exoskeleton design, some groups have opted to use passive DOF for the sternoclavicular joint [23], [28], [29]. Passive DOF allows the sternoclavicular joint to move freely but eliminates the ability to generate actuation forces at the joint.

2.2.1 Other differentiating aspects

2.2.1.1 Actuation method

Most developments for the upper limb exoskeletons are actuated using electromagnetic motors due to their ease of actuation. Pneumatic muscle actuators (PMA) have also been used by several groups [30–32]. They are inherently compliant and back-drivable, and therefore deemed safer. PMAs have high power to density ratio

compared to electric motors; however, response frequency of PMA (typically around 5 Hz) is lower than for electromagnetic motors. RUPERT IV for example, is a lightweight exoskeleton that is portable and uses unpaired PMA to provide movement for the five DOF of the upper limb [31]. However, the joints can only be actuated in one direction since only one PMA is used for each DOF. Smart materials have also been used for actuation of exoskeleton robots. For example, De Rossi *et al.* [33] have integrated electro-active polymeric materials into wearable garments. There have also been hydraulic actuation exoskeletons. Miniaturized hydraulic actuators were used by Schultz *et al.* [34] in a hybrid powered upper limb orthosis (FEShybrid) design for the elbow flexion. The FEShybrid orthosis combines the advantages of orthotics in mechanically stabilising joints together with the possibilities of Functional Electrical Stimulation (FES) for actively eliciting muscle contractions.

2.2.1.2 Alternate user interfaces

Several research groups have focused on alternate solutions other than joystick and EMG user interfaces. Kiguchi *et al.* [35] have developed an assistive exoskeleton robot that takes environmental information obtained from sonar sensors and a stereo camera into account. Baklouti *et al.* [36] proposed the use of face and mouth gestures as control commands for an exoskeleton. De Rossi *et al.* [33] have been developing wearable garments using electroactive polymers which allow strain sensing and actuation. Although these haptic devices are not exoskeletons, their features can be integrated into exoskeletons which can be used, for example, in a virtual reality environment.

2.2.1.3 Kinematic stability

Multi DOF robots may possess configurations which causes a DOF to be lost. This occurs in configurations where the axes of two rotary joints align with each other. Some of the singular configurations can be resolved by attaching the exoskeleton to the user's upper arm and forearm. However this approach cannot be used for the singular configurations of the exoskeleton shoulder joints. CADEN-7 [37], MGA [25] and MEDARM [38] exoskeletons have been designed so that the singular configurations of the shoulder joints occur at a posture that is least likely to interfere with performing rehabilitation tasks, however a singular configuration may still exist.

2.2.2 Conclusion

Table 2 shows a summary of the mentioned devices along with their advantages and disadvantages.

As it is evident from the literature review presented in Table 2, there are a very limited number of assistive exoskeleton powered upper limb devices that are (i) wearable (or at least portable), and (ii) do not rely on EMG signals. Examples of limiting factors include: RLAO [19], which uses a bi-directional tongue switch and a sequential joint-by-joint activation control which makes it tedious to operate. MULOS [26], for example, is heavy and needs to be mounted on a wheelchair. MULOS also requires a functional hand that can operate the joystick, thus making it unusable for persons with two disabled arms. IKO [27] relies on at least one functional hand and lacks a grasping functionality.

Table 2. Summary of Exoskeleton Robots in Literature.

Exoskeleton Robots	Active DOF	Assistive Rehab	User Interface	Actuation	Attachment
HAL-5 (CYBERDYNE Inc.) [21], [22]	19	Assistive	EMG	Electric motor	Wearable
IntelliArm [23]	8	Rehab	Haptic +EMG	Electric motor (cable drive)	Wall-mount
SUEFUL-7 [28]	7	Both	EMG	Electric motor	Wall-mount
CADEN-7 [37]	7	Both	EMG	Electric motor (cable driven)	Wall-mount
Sarcos Master Arm (Sarcos, Inc.) [24]	7	Assistive	Haptic	Hydraulic	Wall-mount
ARMin III [20]	6	Both	Haptic Joystick	Electric motor	Stationary base
MGA [25]	6	Both	Haptic	Electric motor	Wall-mount
RLAO [19]	6	Assistive	tongue switch	Electric motors	Wheelchair
MEDARM [38]	6	Rehab	EMG	Electric motor (cable drive)	Wall-mount
RUPERT IV [31]	5	Rehab	Operator driven	Pneumatic	Wearable
MULOS [26]	5	Assistive	Haptic Joystick	Electric motor	Wheelchair
IKO [27]	5	Assistive	Haptic	Electric + Pneumatic	Wearable
Hand Mentor (Kinetic Muscles Inc.)	1	Rehab	EMG	Pneumatic	Wearable
mPower (Mymo Inc.) [13]	1	Assistive	EMG	Electric motor	Wearable
FEShybrid [34]	1	assistive	EMG	Hydraulic	Wearable

While numerous examples of assistive and rehabilitative robots exist, a highly functional, user-friendly and portable powered assistive device capable of performing every day activities has yet to be developed. This is the aim of the UBC PULO program. In the following section we will introduce the UBC Powered Upper Limb Orthosis which operates based on an intuitive control interface specifically designed to enable users to do simple everyday tasks.

2.3 Previous development of the UBC Powered Upped Limb Orthosis (UBC-PULO)

The UBC-PULO concept started research and development at UBC in 1992. By studying the target users and kinematics involved in doing every-day tasks activities, previous research work at the UBC Biomechanics and Rehabilitation Engineering Laboratory (UBC - BioREL) has worked to narrow down the scope and capabilities of a defined user along with the requirements of the electro-mechanical design, has identified the user control interfaces, and established a unique control strategy to provide a user-friendly interface to control a highly-functional, high DOF device designed to perform daily living tasks. This section introduces the previously developed UBC-PULO prototype, and clarifies why certain design choices were made based on the project goal and target user assumptions. It also discusses the mechanical system, electrical system and control system of the prototype for the purposes of comparison with other existing devices.

2.3.1 Introduction

Previous research defined a set of concise user assumptions by determining the target user group, understanding their disability and recognizing their needs [9, 11-15]. This was a necessary step to narrow down the design requirements for both the electro-mechanical system and the control strategy. Section 2.1 includes an overview of potential disease or injuries which might inflict a potential orthosis user. The defined user characteristics for the UBC-PULO, which eventually shaped the design requirements of the first prototype, are listed in Table 3 below:

Table 3. Defined User Characteristics

-
- User has two completely paralyzed arms
 - User is not capable of any movement below the shoulder
 - User has full range of head motion (side-to-side/back-to-front)
 - User has vertical shoulder motion
 - User has intact hand sensation
 - User shows no signs of spasticity
 - User demonstrates full range of joint motion in the upper limbs
 - User has full cognitive abilities
 - User is adult sized
-

As identified in Chapter 1, the goal of the UBC-PULO project is to design a highly functional, user-friendly portable system that enables users to perform daily living tasks by restoring function to one entire arm. One of the most rewarding and valuable aspects of the orthosis is how well it will allow the end user to gain independence. Thus, it was essential to first identify the top priority everyday activities to properly optimize the UBC-PULO design [2]. Based on interviews and surveys conducted with potential users

through previous research [2], the top five priority daily activities were identified. Table 4 lists the top five tasks users would like to be able to perform by having their upper limb function restored:

Table 4. Top Priority Tasks [2]

Priority	Task
1	Reaching/picking up objects
2	Basic personal hygiene
3	Hobbies/crafts
4	Eating/drinking
5	House work

As shown in Figure 2, the human arm has seven DOF [39]. In an attempt to reduce the complexity of the mechanical system and control strategy, a study of human arm motion involved in doing the top priority tasks (Table 4) was conducted. Evaluating real 3D workspace data (acquired by using two video cameras and reflective tape) against a kinematic simulation program [2] indicated that five DOF are adequate for performing high-priority living tasks. More specifically, the shoulder elevation and wrist yaw degrees of freedom (marked in red in Figure 2) were eliminated from the design.

Figure 3 shows the first prototype which provides five DOF plus grasp. The plastic body cast is custom-made for each user to comfortably encircle their waist, sitting on their hips to support the weight of the electro-mechanical device. An assistant will initially help the user to put on the device, place their arm in the cuffs and strap it down using hook-and-loop fasteners. More details about the mechanical design and its features are described in Section 2.3.2.

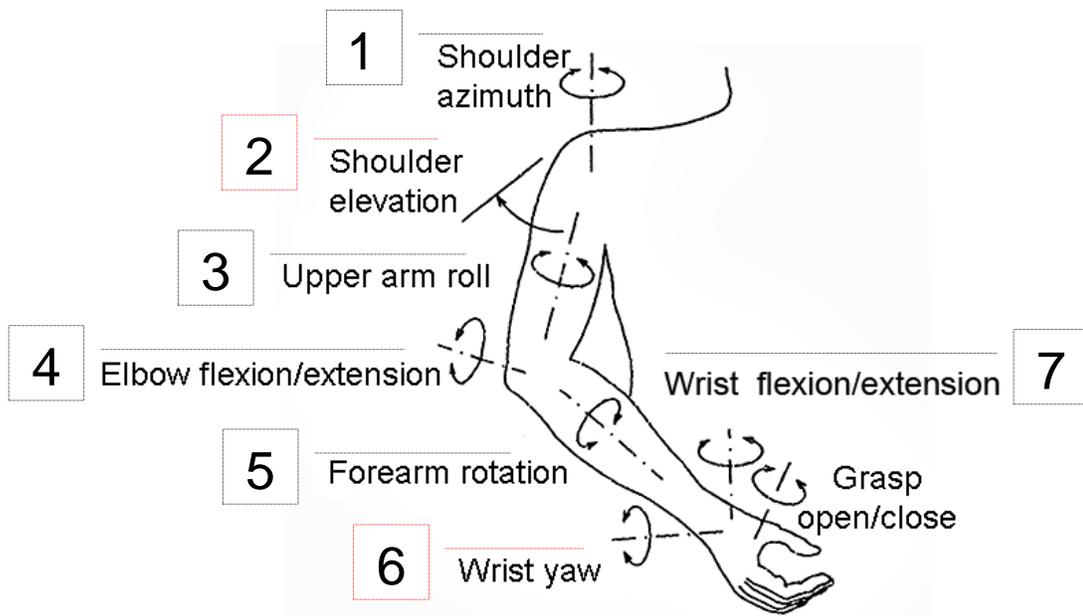


Figure 2. Degrees of Freedom in the Human Arm



Figure 3. UBC-PULO Prototype

One of the challenges for UBC-PULO development was creating a suitable user control interface for people with the assumed user characteristics – e.g. two flail arms.

Previous research work examined various control interfaces including, but not limited to, shoulder movement, EMG signals, eye movement, head movement, chin switches and voice commands. Based on a weighted point comparison system, the head and shoulder are clearly the most advantageous control interfaces for the target market [7]. Thus the UBC-PULO uses both a head controller and shoulder controller (see Figure 4), which serve to provide an intuitive and user-friendly interface for the UBC-PULO. Figure 4 demonstrates how the head tilt in the X and Y rotational axes produce a *proportional velocity vector* (for more details see Section 2.3.2.2) used for controlling the end-effector velocity. The Z axis can be controlled by performing extended shoulder shrugs (for more details see Section 2.3.2.2).

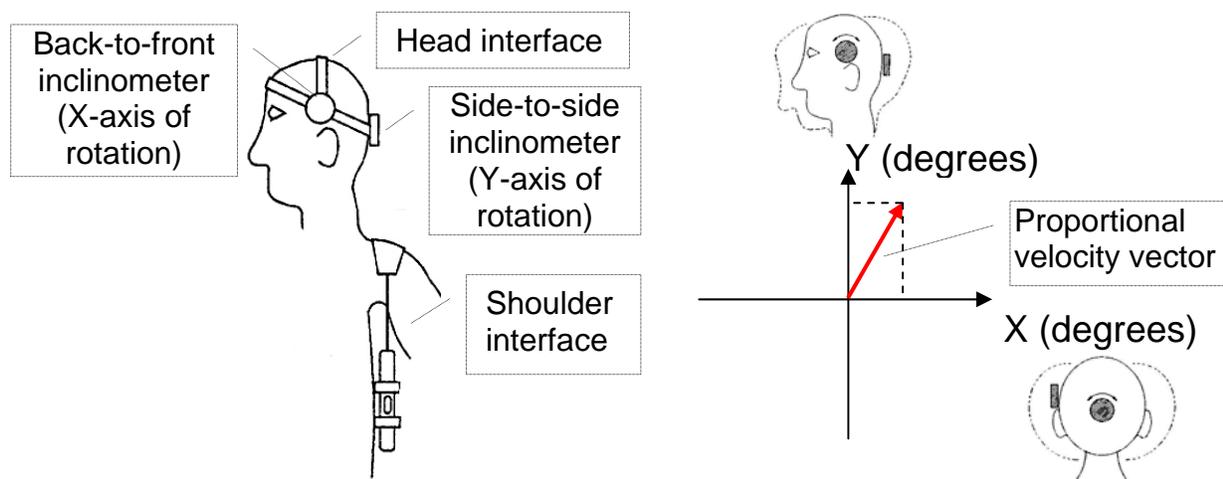


Figure 4. Control Interfaces

For comfort and safety reasons, a user-definable neutral zone permits a predetermined range of free head motion without resulting in any end-effector motion. As shown in Figure 5, any head movement that remains within the neutral zone will not generate any motion; however, once the head passes beyond the user-defined

threshold, controlled device motion is generated. More details about the head controller can be found in Section 2.3.2.2.

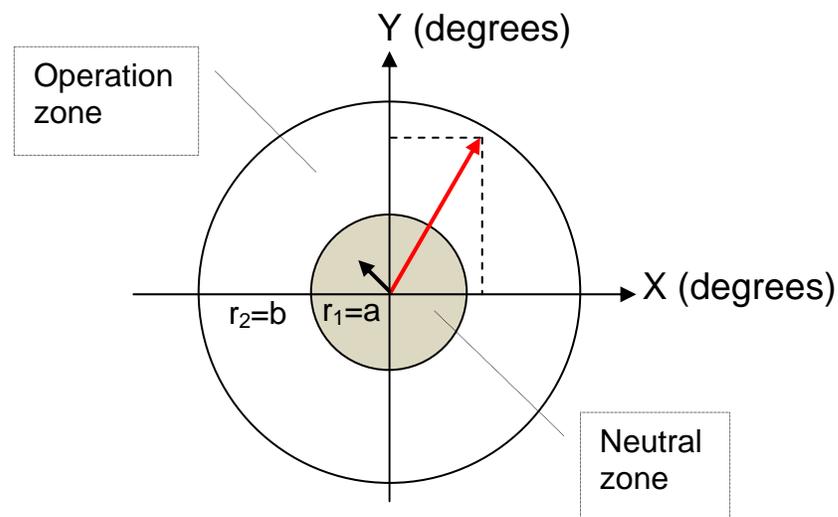


Figure 5. Neutral Zone

The shoulder controller is a custom-made three-level switch which tells the control system whether the vertical shoulder is at neutral (LOW), half-way (MED), or full vertical position (HIGH). The shoulder controller allows users to switch between the UBC-PULO control modes and selects the one most suitable for the task they are planning to perform. The next section (2.3.2) provides more detail about the control interfaces, control strategy, electrical systems and mechanical system.

2.3.2 Design and development

2.3.2.1 Mechanical system

Figure 6 shows one of the first conceptual designs and identifies the main components of the UBC-PULO system. The electro-mechanical exoskeleton has five

DOF plus grasp and is installed on a custom-made plastic body cast such that it sits comfortably on the user's hips. The user's arm resides in a series of cuffs where it is strapped down using hook-and-loop fasteners, thus providing stiffness and support to the otherwise flexible exoskeleton structure. Specific mechanical components in each segment of the exoskeleton were designed to be either adjustable or easily modified so that the unit could be easily custom fit to the user.

The electrical boards interpret all system inputs, including those from the user, and drive the electrical motors based on the device kinematic and control strategy computations. Next, the motor power is transmitted via a flexible drive shaft to each joint gearbox resulting in relative link motion at that joint. The azimuth and shoulder roll joints are driven directly via a motor and gearbox unit whereas all the other joints (including grasp) utilize a flex-shaft to transfer power to the gearbox and joint unit. While the orthosis unit needs to be as light as possible overall, it is also very important to minimize the weight of the moving robotic arm exoskeleton to both minimize power requirements, inertial loads, and to keep the weight distribution of the device concentrated on the user's waist for ergonomic reasons. For this reason, flexible shafts were used - allowing the motors to sit on the plastic body cast thus reducing the weight and bulkiness of the robotic exoskeleton.

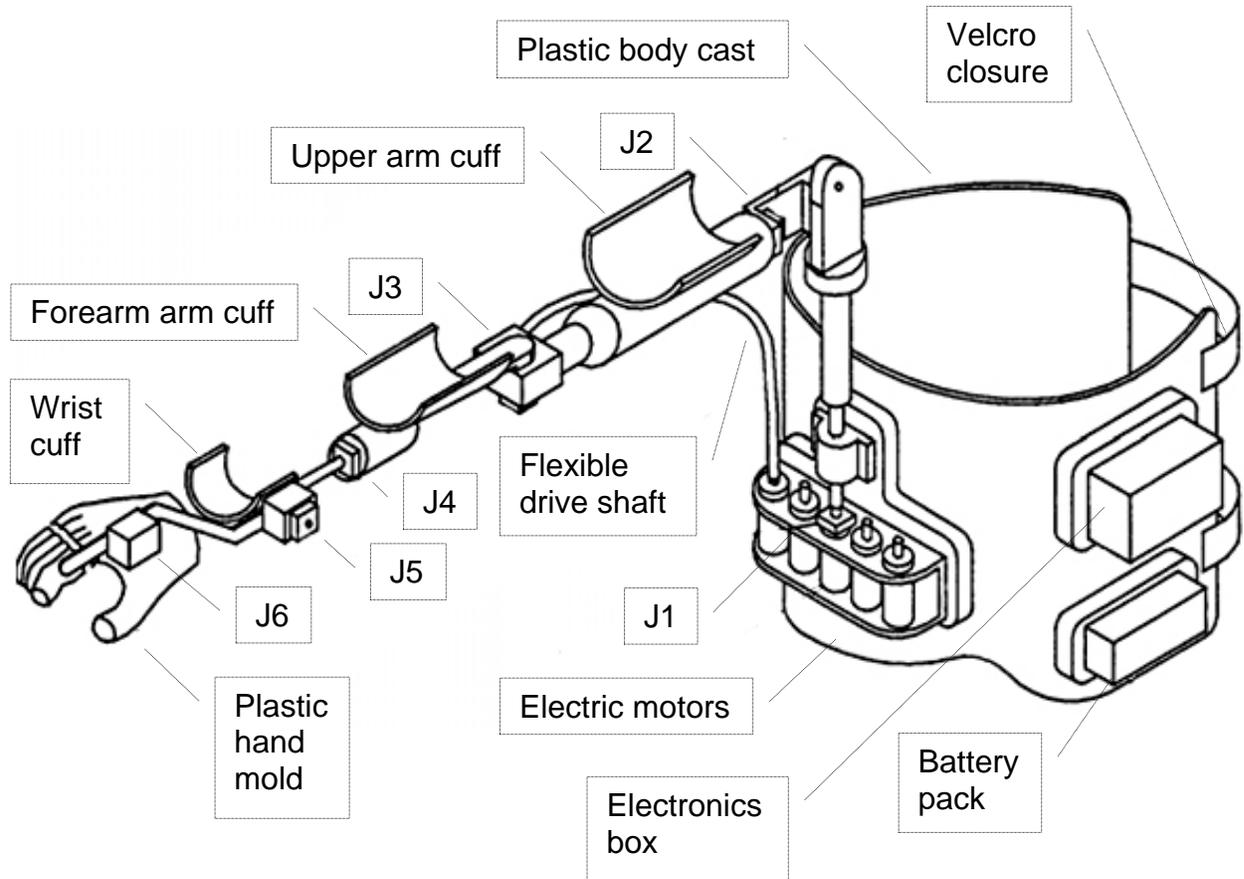


Figure 6. UBC-PULO Conceptual Design

After some modification to the initial concept design, the first UBC-PULO prototype was developed (see Figure 3). The location of each joint is highlighted with J1 to J6 representing the contact point between the gearbox's output to the next mechanical link. Angle sensors attached to each one of the joints provide angular position information to the control system. Figure 14 highlights the axis of rotation of each degree-of-freedom (more details are provided in [7]).

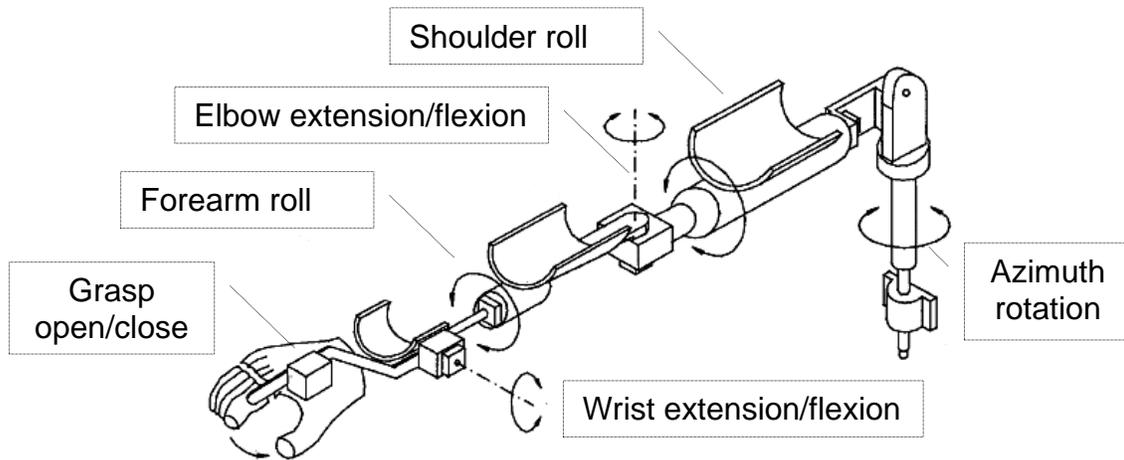


Figure 7. Axis of Rotation for each DOF

2.3.2.2 Control system

The UBC-PULO control system is composed of two levels: Low level control and high level control. As demonstrated in Figure 8, the job of the low level controller is to minimize error between the commanded joint angles (received from the high level controller) and real-time (i.e. current) joint angles (received from joint angle sensors) by outputting corresponding commands to the motor drive system. The UBC-PULO prototype low level controller is a collection of six parallel proportional integral derivative controllers (PID) minimizing all six joint angles errors simultaneously. More details of the low level controller are presented later in Section 2.3.2.3.

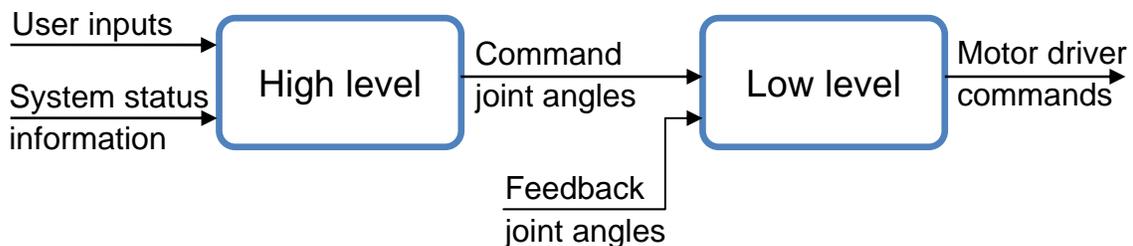


Figure 8. Control System

The high level controller, depicted in Figure 8, embeds the main control strategy thus enabling users to do everyday tasks by issuing the right commands to the low level controller. User inputs and the current status signals are interpreted to produce desired joint angles which are, in turn, sent to the low level controller as *command joint angles*. The *user inputs* in this case are the inputs received from the shoulder interface and head interface. The *system status information* is comprised of the current control mode, current joint angles, safety conditions, and other fault conditions (e.g. such as a low battery). Table 5 outlines the main functions of the high level controller responsible for computing the output command joint angles.

Table 5. High Level Controller

High Level control functions	Description
Neutral zone detection	<ul style="list-style-type: none"> • Determines if the head controller X-Y vector signals are outside of the neutral zone
Mode of operation	<ul style="list-style-type: none"> • Interprets shoulder controller signals to select the mode of operation
Safety	<ul style="list-style-type: none"> • Enforces preset software joint limits • Monitors the battery level • Detects unintentional user inputs (such as sneezing) for emergency shutdown
End-point control	<ul style="list-style-type: none"> • Implements the inverse kinematics to allow for direct control of the end-effector position by moving the joints concurrently

As shown in Figure 5, all head controller signals must exceed the neutral zone in order to produce a vectorized control input. The neutral zone ensures user safety and user comfort by allowing end-effector motion to occur only after the head controller

passes a predefined head tilt angle, thus avoiding motion due to minor involuntary head motion. Previous research [7] has assumed that the neutral zone is a circular shape with a radius $r_1=a$ (degrees head tilt). Equations 1 and 2 present the normalized control inputs u_x and u_y , where x and y are the head inclinometer inputs in the x and y head rotational axes directions respectively. It is assumed that the maximum head angle input is less than $r_2=b$ (degrees head tilt) which defines the outer limit of the operation zone. Note that any input vector with a magnitude greater than b will be saturated to the value of b . The vector \vec{U} presented in equation 3 is the resultant normalized head controller input.

$$\left. \begin{aligned} u_x &= \frac{x - a}{b - a} && ,\text{if } x > a \\ u_x &= 0 && ,\text{if } x \leq a \end{aligned} \right\} \text{eq.1}$$

$$\left. \begin{aligned} u_y &= \frac{y - a}{b - a} && ,\text{if } y > b \\ u_y &= 0 && ,\text{if } y \leq b \end{aligned} \right\} \text{eq.2}$$

$$\vec{U} = \frac{u_x \langle \vec{x} \rangle + u_y \langle \vec{y} \rangle}{\sqrt{2}} \text{eq.3}$$

For safety reasons and the user's comfort, the high level control system enforces user definable, pre-set joint limits such that none of the joints can exceed the user's comfort zone. The safety function has a built in emergency shutdown in case the user exerts an involuntary head controller input such as coughing or sneezing. This is

calculated by ensuring that the magnitude of the derivative of \vec{U} is below a pre-set threshold.

The high level controller is based on the concept of end-point control, which is one of UBC-PULO's key features. End-point control allows the user to directly control a specific end-point (in this case the user's hand) in any desired direction by producing concurrent joint motion. The end-point motion, which is a product of simultaneous joint angle computations, is directly proportional to the 3D input velocity vector \vec{U} acquired from the head and shoulder interfaces. By implementing the end-point control concept in the high level control system, the user may move their arm to any desired position in a simple, intuitive and efficient manner. Mathematically speaking, end-point control is achieved by solving all joint angles for a given end-point destination using inverse kinematics. In general, there are three ways of computing inverse kinematics: numerical methods, geometric solutions or algebraic solutions. Previous research work decided on using the inverse Jacobian method which is an algebraic way of solving inverse kinematics. Appendix A outlines the upgraded inverse Jacobian algorithm programmed into the UBC-PULO prototype along with the calculations.

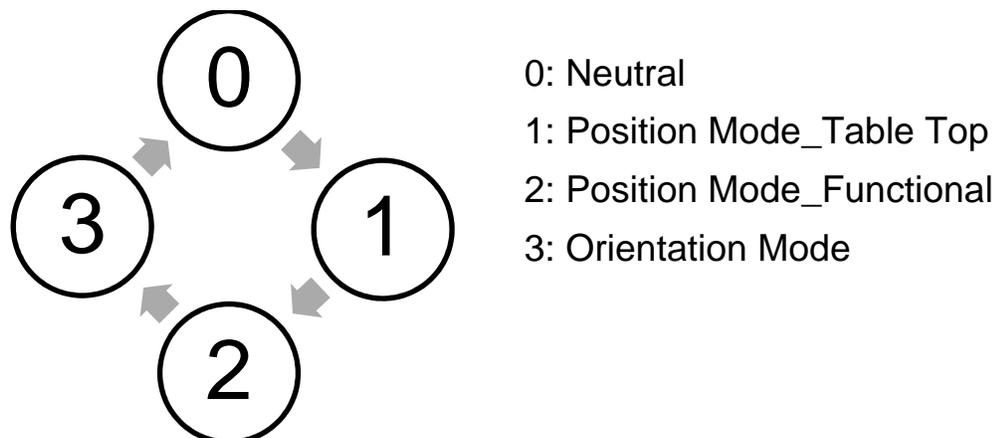


Figure 9. UBC-PULO Modes of Operation State Machine Diagram

The high level controller interprets the shoulder controller signals to allow the user to select the desired mode of operation. The high level controller distinguishes between a quick shoulder shrug and an extended shoulder shrug as input signals. A short shoulder shrug switches between the four modes of operation as presented in Figure 9. Depending upon the mode of operation, an extended shoulder shrug either controls the end-effector in the Z direction (direction perpendicular to X-Y plane) or controls hand grasp. In *position mode*, holding the shoulder controller in the MED position lowers the end-effector in the Z direction, while holding the shoulder controller in the HIGH position does the opposite. Similarly, in *orientation mode*, holding the shoulder controller in the MED position closes the grasp and does the opposite when in HIGH position. The flow chart in Figure 10 summarizes the control strategy for the UBC-PULO. In position mode, the end-effector reaches the desired X-Y-Z position based on concurrent motion of three joints: azimuth, shoulder roll and elbow joints.

In orientation mode, the head interface directly controls the wrist and forearm joints to take the user's hand to the desired orientation. Holding the shoulder interface in MED or HIGH position controls the grasp when in *orientation mode*.

The UBC-PULO utilizes two position modes derived as a result of the previous motion analysis research study [2], [5]: Table top mode and functional mode. In table top mode the user may manipulate the end-effector in either the horizontal plane (table-top) using the head interface or in the vertical direction using the shoulder interface, allowing the user to pick, place and move objects. The functional mode is designed for tasks requiring movement between the table-top and the user's head such as eating or drinking. In this mode, the end-effector moves on a pre-defined inclined plane between

the table-top and user's head position maneuvered by the head interface. In functional mode, holding the shoulder interface in the MID or HIGH position moves the end effector in a direction perpendicular to the incline plane (Z direction). Once the desired position is reached the user may select orientation mode by shrugging their shoulder and then achieve the desired orientation using the control interfaces (Figure 10), thus providing a user friendly control strategy.

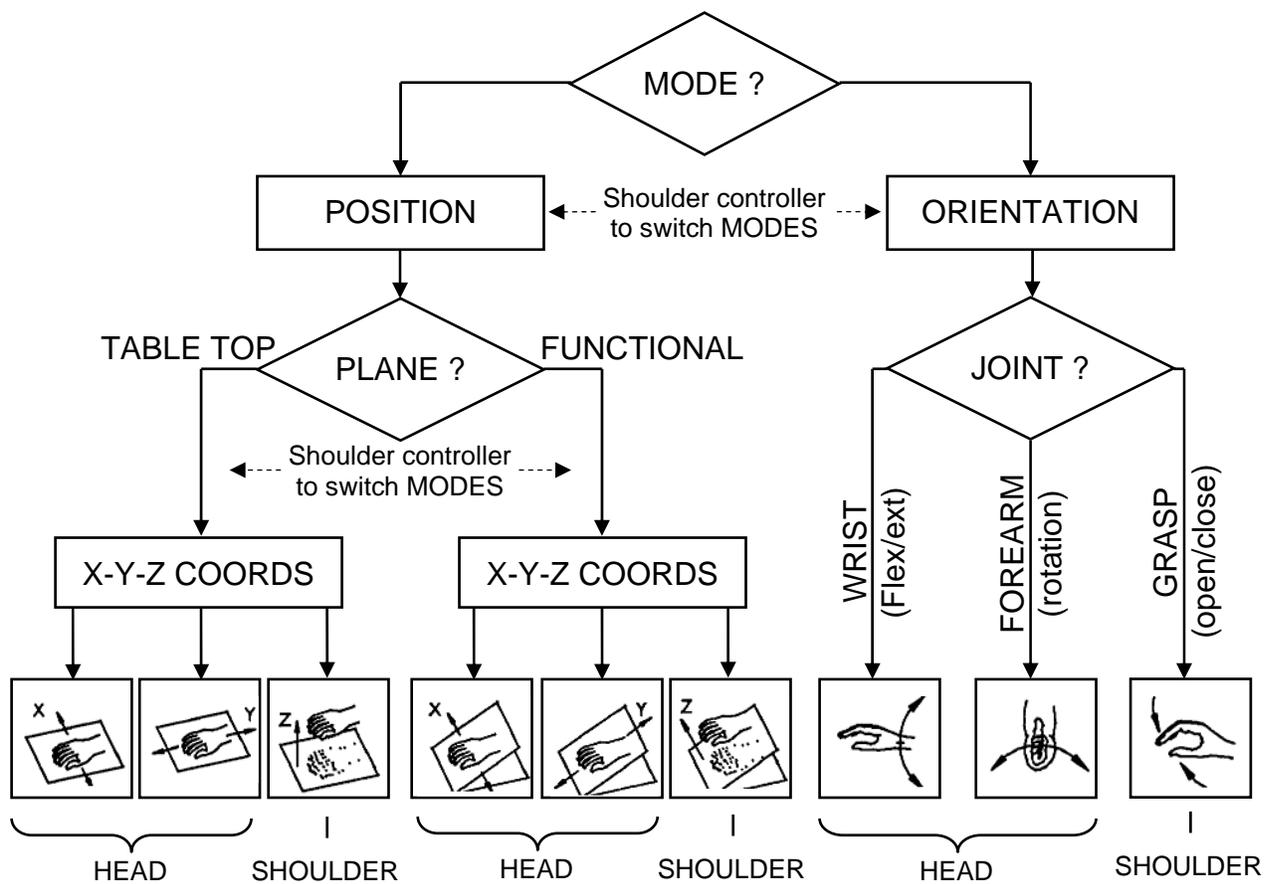


Figure 10. UBC-PULO Control Logic Flowchart

2.3.2.3 Electrical system

Table 6 lists the UBC-PULO electrical system components. The electrical boards process the user inputs and joint angle inputs to produce the desired motion by driving the motors. This section reviews each component of the electrical system and is essential to understanding the subsequent design modifications described in this thesis.

Table 6. Electrical System Components

Component	Description
Head interface	Two optical inclinometers output the front-to-back and side-to-side angles of the user's head
Shoulder interface	Outputs two digitally encoded signals to indicate a LOW, MID, or HIGH shoulder position
Joint angle sensors	Six 25 K Ω potentiometers at joint provide an analog output that is directly proportional to the joint angle
Electrical boards	The boards implement the functions of the high and low level controllers, and drive the electrical motors
Motors	Six brushed DC motors drive of the six joints of the prototype

2.3.2.3.1 Head interface

The head interface was made of two optical inclinometers (T2-7200, U.S. Digital Corp.) strapped on the side and back of the head as presented in Figure 11-a. Figure 11-b shows the physical dimensions of each inclinometer. Each encoder produces a digital two-channel quadrature encoder signal and is capable of capturing angles up to a 0.1° resolution.

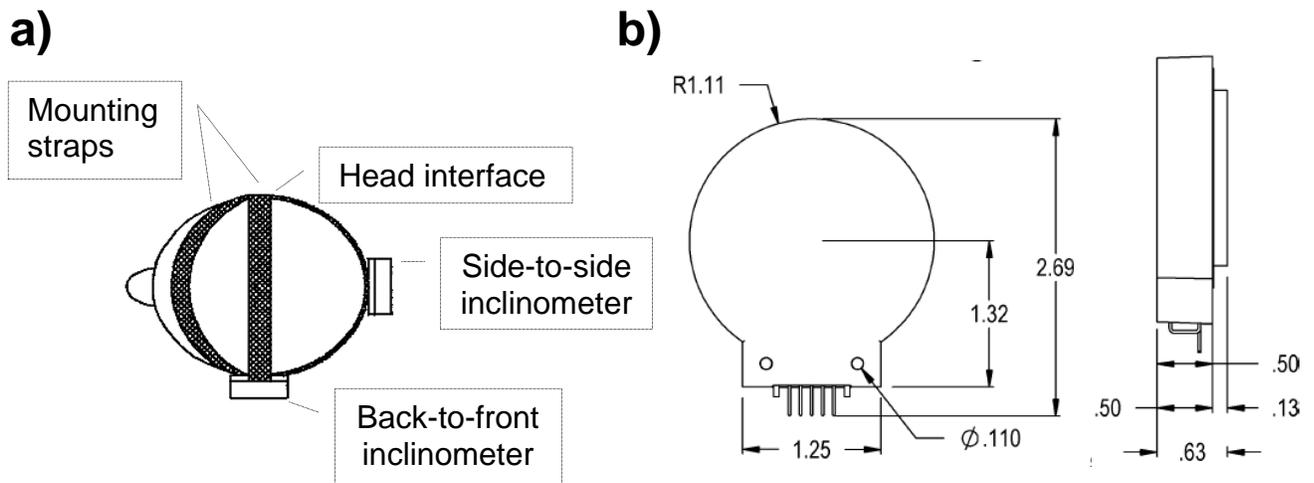


Figure 11. a) Head Interface Inclinometers b) Inclinometer Dimensions (inches)

2.3.2.3.2 Shoulder user interface

As shown in Figure 12-b, the shoulder user interface, developed previously in the UBC – BioREL laboratory, detects the vertical shoulder position via two optical switches and a slide bar mechanism. Since the slide bar is connected to the shoulder via a cable, the slide bar gets translated up and down along with the shoulder. A spring attached to the other end of the slide bar pulls it back down when the shoulder is in the relaxed (LOW) position. The optical switches distinguish the slide bar position by producing two encoded digital signals as shown in Figure 12-a. When the slide bar blocks an optical switch, the infrared light is blocked and it outputs a 0V digital signal (OFF). In contrast, when the slide bar passes the optical switch, the infrared beam can pass through, producing a 5V signal (ON).

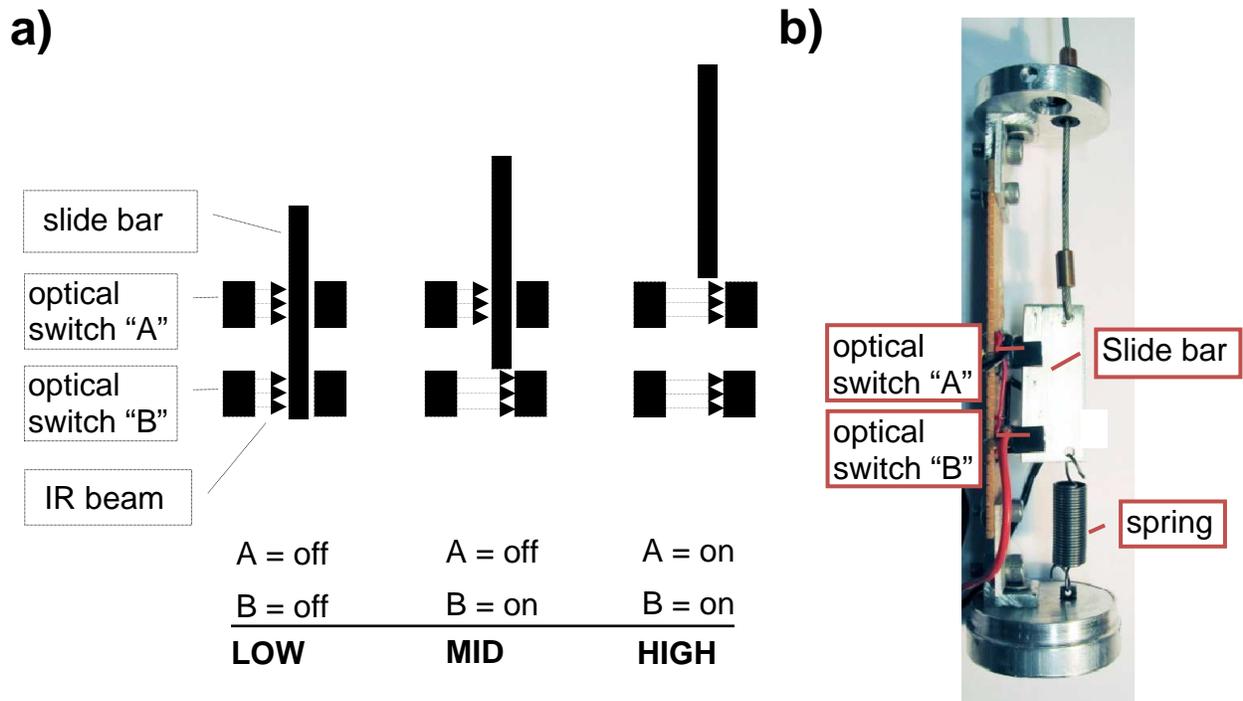


Figure 12. a) Schematic of Shoulder User Interface at LOW, MID and HIGH Positions b) Photo of the Shoulder User Interface without Cover

2.3.2.3.3 Joint angle sensors

The angle sensors consist of rotary potentiometers with their hub directly connected to each joint shaft. As illustrated in Figure 13-a, rotating the hub sweeps the wiper on top of the resistive material causing a change in resistance R1 and R2. By powering the potentiometer with a 5V source and measuring the output voltage, the angle may be calculated as shown in Figure 13-b. The dead-zone is a range where the output voltage goes to zero and therefore cannot be used for any measurements.

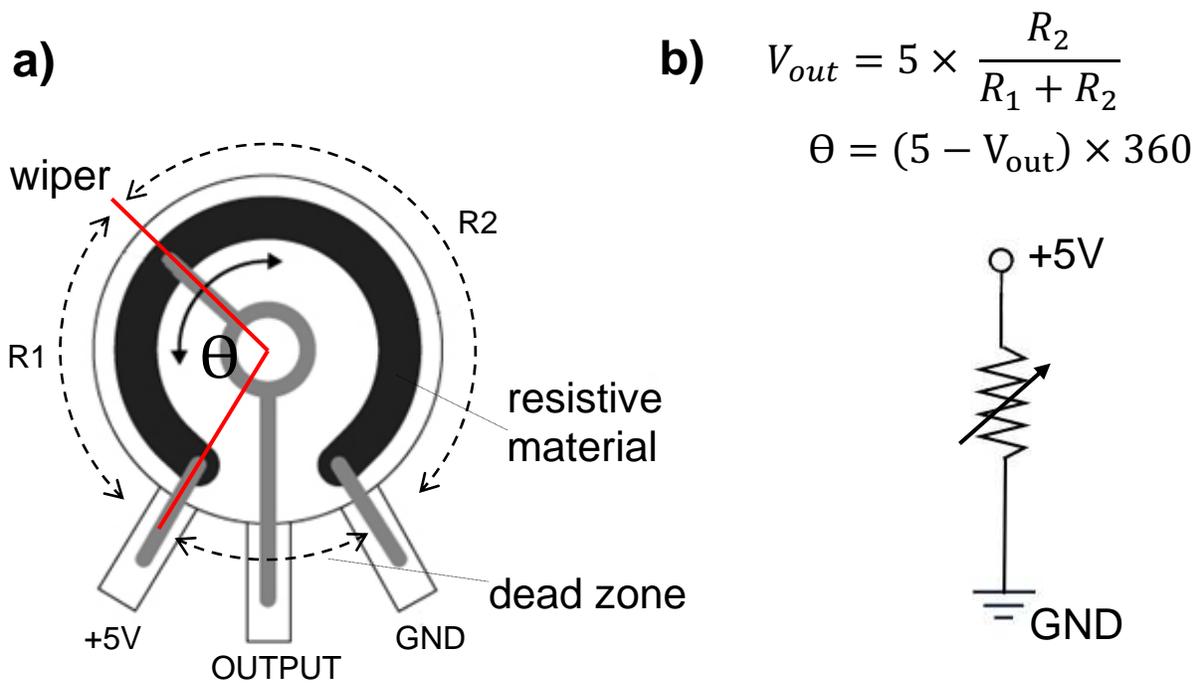


Figure 13. a) Rotary Potentiometer b) Potentiometer Electrical Representation

2.3.2.3.4 Electrical boards

Two electrical boards receive signals from the user interfaces and joint angle sensors to perform the control algorithm computations and then drive the six electric motors. The electrical boards shown in Figure 14-a incorporate the “high level controller” functions discussed in Section 2.3.2.2 using two microprocessors running in parallel. The board shown in Figure 14-b, on the other hand, performs the “low level control” computation using a microcontroller and drives the electric motors by six parallel class AB amplifiers- which are analog linear amplifiers.

a)



b)

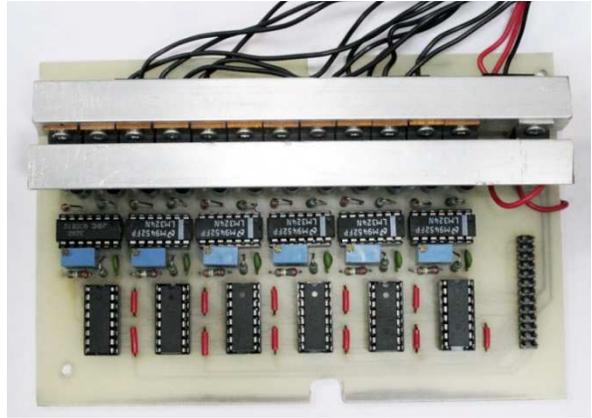


Figure 14. a) High Level Control Board b) Low Level Control and Motor Driver Board

In total, the electrical boards have three processors running in parallel (the control algorithm processor, inverse kinematics processor, and angle controller processor) leading to a faster computation time (Figure 15). The *control algorithm processor* receives the head interface and shoulder interface signals and uses these to calculate the desired incremental change in the end effector position (dx, dy and dz). The *inverse kinematics processor* produces the desired joint angles by computing the inverse Jacobian relative to the concurrent joint angles (for more detail regarding inverse kinematics see Appendix A). The *angle controller processor* receives the desired joint angles (from the inverse kinematics processor) and current joint angles (from the joint angle potentiometers) to run six parallel PID (Proportional Integrator Differentiator) loops that keep each joint at the desired position by feeding the right voltage to the motor driver amplifiers.

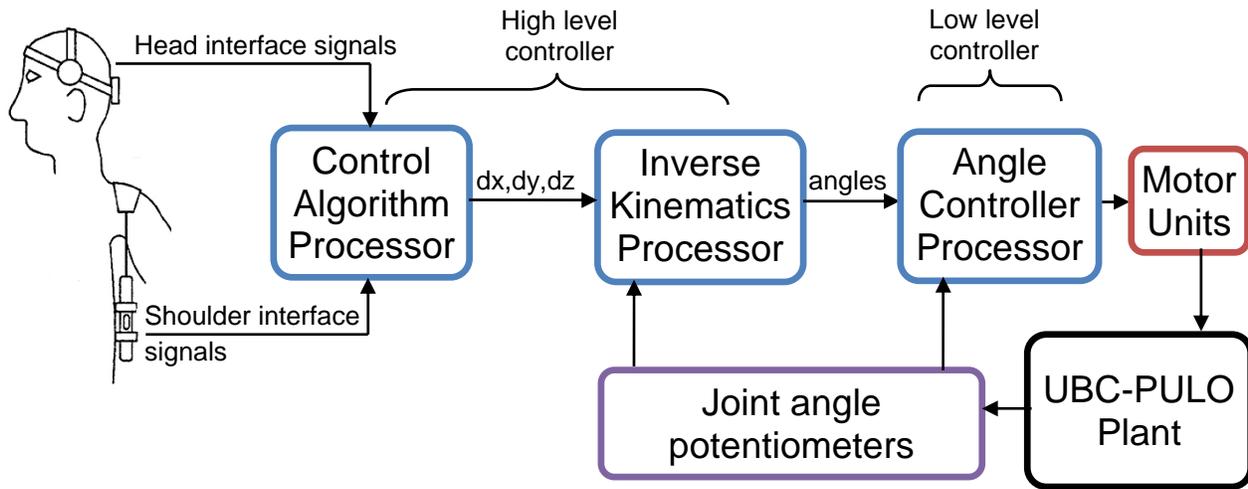


Figure 15. Electrical System

2.4 Review of UBC-PULO Virtual Reality Simulation Environment (VRSE)

2.4.1 Introduction

The Virtual Reality Simulation Environment (VRSE) was previously developed in the UBC – BioREL laboratory to allow for efficient optimization of the UBC-PULO control system [1]. By mimicking the behaviour of the UBC-PULO prototype, the virtual simulator can also be used for testing the overall control system, testing the dynamic behaviour of the prototype, as well as training or screening users [1]. The VRSE was developed using MATLAB's Simulink, SimMechanics and the Real-time tool boxes. Figure 16 presents an overview of the UBC-PULO VRSE.

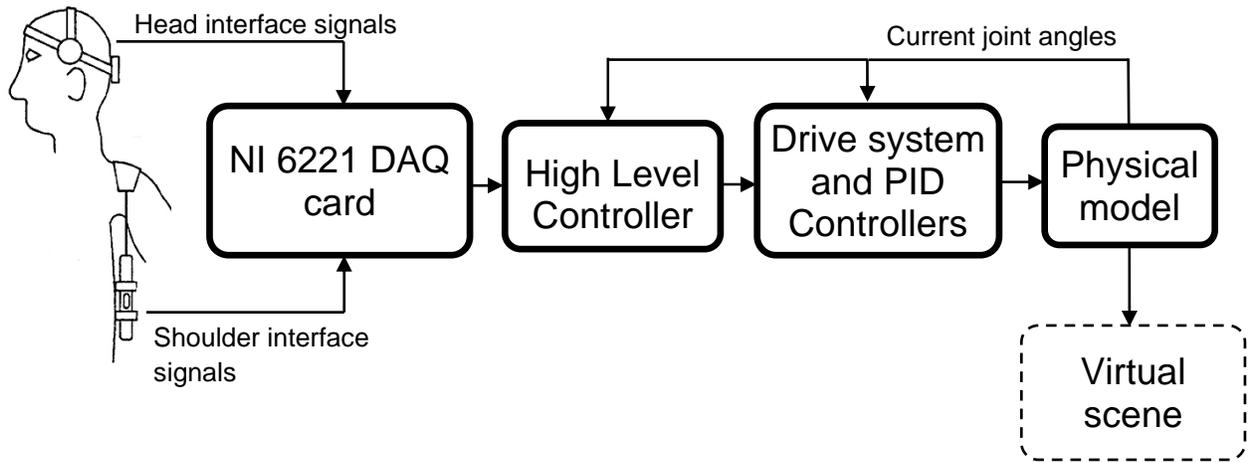


Figure 16. Overview of the UBC-PULO Virtual Reality Simulator

The simulator acquires the user interface signals using a National Instruments data acquisition card (NI DAQ 6221) and sends this information to the *high level controller*. The *high level controller* is responsible for simulating the behaviour of the UBC-PULO high level control board (see Section 2.3.2.3.4), which interprets the user inputs to produce the desired output joint angles. The six desired joint angles are sent to the *drive system and PID controller* block, which is composed of a model of the six DC motors and the six parallel PID controllers. The developed DC motor models are designed to simulate the actual response of the DC motors, and the PIDs minimize the error between the commanded joint angles (i.e. desired joint angles) and the physical model's current joint angles. The outputs of the DC motors drive the *physical model*, which simulates the dynamic behaviour of the UBC-PULO exoskeleton (i.e. joints and links). The coordinates of the *physical model's* joints, links and end-effector are sent to the *virtual scene* block which provides a virtual representation of the user's arm in a room environment (i.e. a room with a table, TV and other virtual objects). Thus, the

VRSE simulates the UBC-PULO prototype's behaviour and allows the user to control their virtual arm in a virtual scene using the UBC-PULO control interfaces. The following sections provide more details about the implementation of the *high level controller* block, *drive system and PID controller* block and *physical model* block. Further details about the development of the VRSE can be found in [1]. Since the overall goal of this research is to validate the VRSE, this review also highlights the relevant limitations of the VRSE with respect to the current UBC-PULO prototype.

2.4.2 Implementation of the high level controller

The *high level controller* block interprets the shoulder and head interface signals to determine the mode of operation, calculate the desired end-effector position change or orientation change, and to produce the desired joint angles (i.e. as explained in Section 2.3.2.2). During development of the VRSE, it was realized that the inverse Jacobian matrix is ill-conditioned due to the small offset between the axis of rotation of the exoskeleton's shoulder roll and forearm roll [1]. To avoid this singularity issue, the shoulder roll degree of freedom was removed and a shoulder elevation degree of freedom was introduced instead. As a result of this modification, the Jacobian matrix became well-conditioned and a new inverse kinematics and forward kinematics calculation were formulated. Thus, a limitation of the VRSE is that its shoulder degree of freedom is not a representation of the current UBC-PULO prototype.

2.4.3 Drive system and PID controllers

The drive system of the UBC-PULO simulator is composed of models of the DC motors and PID controllers as presented in Figure 17. To minimize the error between the reference joint angle (i.e. desired joint angle) and the current joint angles, the *drive system utilizes a PID controller block*.

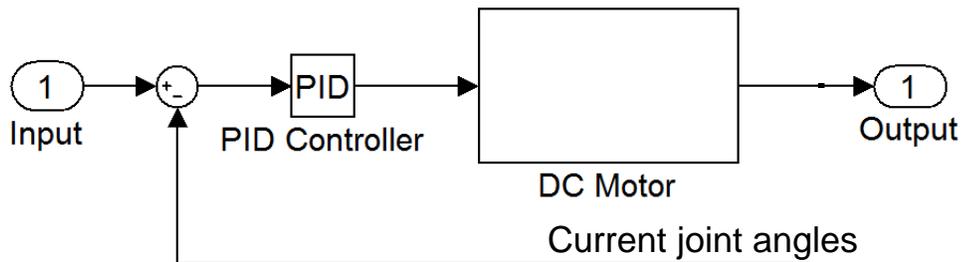
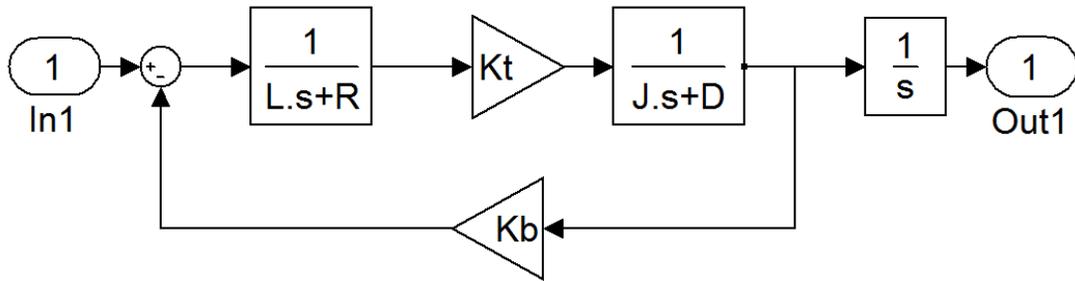


Figure 17. Sample Drive System and PID Controller Block [1]

The DC motors are modeled as second order s-domain transfer functions to describe the relationship between the input voltages and output positions as presented in Figure 18. The various motor parameters used by the DC motor models were assumed in the previous research work and implementation of suitable values (i.e. either from motor data sheet or characterization results) were recommended.



Where,

- R = Armature resistance
- L = Armature inductance
- Kt = Torque constant
- Kb = Back EMF Constant
- J = Inertia
- D = Damping constant

Figure 18. Second Order Motor Model [1]

2.4.4 Physical model

The physical model was created by using SimMechanics and represents the UBC-PULO exoskeleton's kinematic and dynamic properties [1]. The physical model consists of six orthosis arm segments connected via six revolute joints as presented in Figure 19. Each of the revolute joints is driven by the *drive system and PID controllers* block and the resulting simulated current joint angles are output from each joint. The physical model is created in two steps [1]:

1. A SolidWorks assembly model of the orthosis is developed, which includes the geometries, mechanical properties, and mechanical constrains of individual orthosis segments.
2. The SolidWorks assembly is converted into a SimMechanics model using the SolidWorks-to-SimMechanics Translator tool, which is a partnership product provided by SolidWorks and MathWorks.

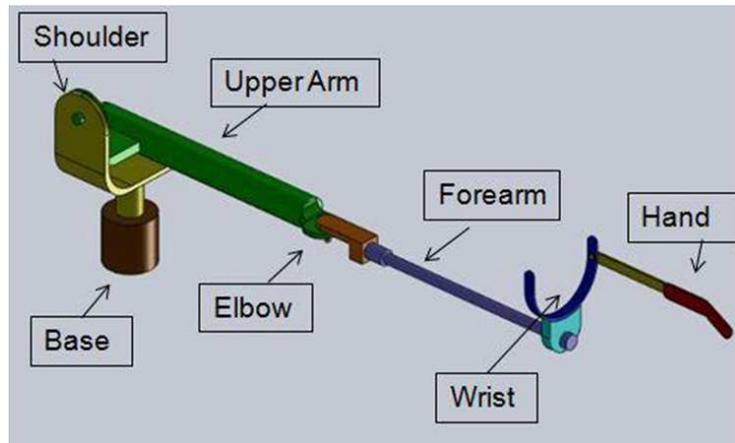
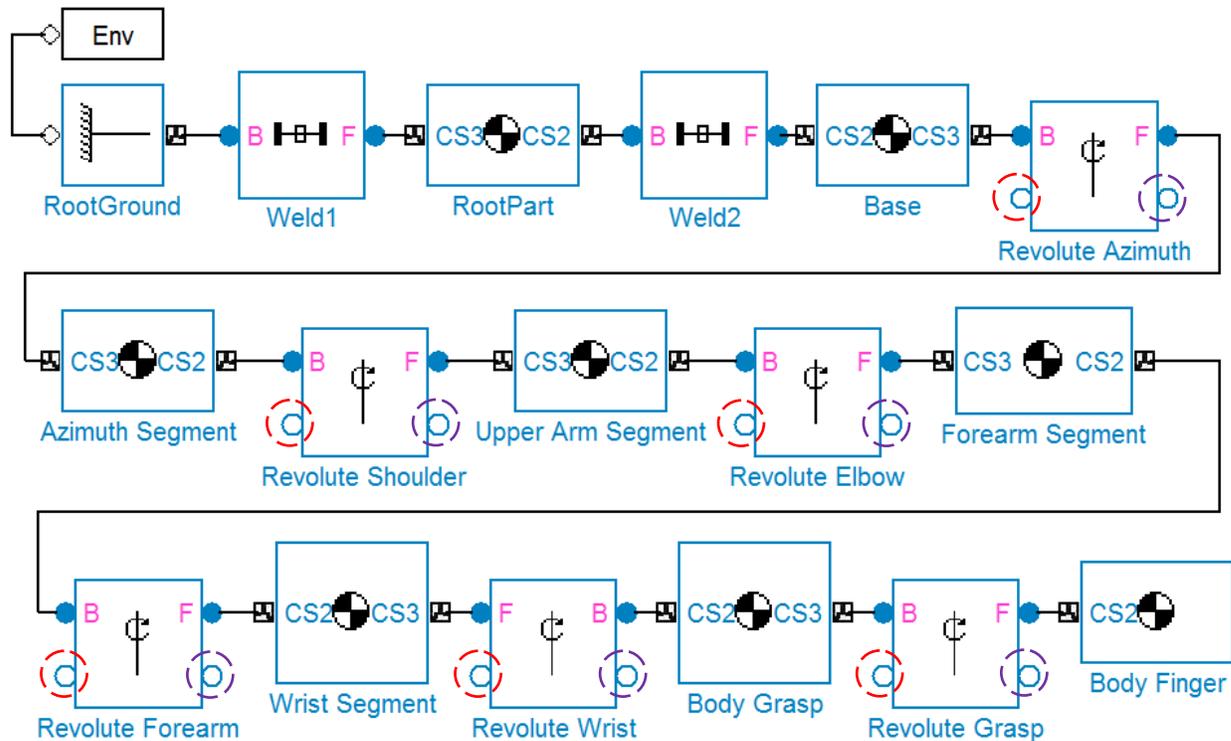


Figure 19. Solid Works Assembly of UBC-PULO Prototype [1].

The SolidWorks assembly of the UBC-PULO prototype, shown in Figure 19, was converted to a SimMechanics model (i.e. the physical model) as shown in Figure 20. The physical model is composed of a series of blocks representing the various orthosis segments (i.e. body blocks) and joints (i.e. revolute joint blocks) that represent the UBC-PULO exoskeleton. The Base block is constraint to the systems ground via a *weld* block. The azimuth, upper arm, forearm, wrist and grasp segments are connected via revolute joints, which allows for revolution around the corresponding joint degree of freedom. The control signals from the *model of the drive system and PID controllers block* are sent to the actuator blocks which drive the joints. The output of the *revolute joint* blocks are sent to sensor blocks, which measure the current joint angles and send them back to the PID controllers.



Where,

1. The *Env* block (i.e. the environment block) provides the mechanical environment for the model, including the simulation dynamics, gravity settings, motions analysis mode and visualization settings.
2. The *RootGround* and *RootPart* blocks define the defined base coordinate system.
3. The blocks marked with the \oplus symbol are body blocks (i.e. orthosis segments), which define the rigid bodies, their mass and inertia properties and coordinate systems.
4. The revolute joint blocks connect the body blocks and define axis of rotation for the degrees of freedom.
5. The red circles at the input of the revolute joints are connected to actuator blocks. The control signals from the *model of the drive system* and *PID controllers* block are sent to the actuator blocks which drive the joints.
6. The purple circles at the output of the revolute joints are connected to sensor blocks, which measure the current joint angles and send them back to the PID controllers.

Figure 20. SimMechanics model of UBC-PULO Prototype [1].

2.5 Summary

A brief demographic review demonstrated that there are numerous people affected with partial or full upper limb paresis. Clearly, an assistive device enabling

users to perform simple everyday activity tasks would be highly beneficial. Despite numerous attempts, there has been limited success in developing an easy to use and highly functional powered assistive device that is portable and suitable for persons with severe cases of upper limb paresis. The UBC-PULO project targets users who are completely paralyzed below the shoulder with the goal of restoring the functionality to perform the top priority daily living tasks (Table 4) to one of the upper limbs. Currently, the UBC Powered Upper Limb Orthosis is the only known assistive solution focusing on restoring daily-living-task arm function to users with two paralyzed upper limbs that is not EMG based. Other concurrent assistive devices either need a partly functional arm to be operated, can restore only limited function, are wheelchair mounted or are not user-friendly. One of the project's user assumptions (Table 3) is that the user has tactile sensation in their arm, realizing that there is no benefit in restoring mobility to a non-functional arm without sensation. An advantage of the UBC-PULO is that users can have the satisfaction of feeling objects (i.e. force, heat and texture) and the sense of self gratification of doing tasks themselves. To satisfy the design requirements stated in Section 2.3.1 while providing an intuitive and user-friendly control system for the UBC-PULO, the previous research team investigated different potential control methods, user interface designs, and the functionality of different device degrees-of-freedom. It was demonstrated that the high-priority daily living tasks (Table 4) can be satisfied by five DOF (Figure 2). The UBC-PULO incorporates both a head user interface and a shoulder user interface to provide the relevant joint motion needed for performing the specific high-priority tasks by means of end-point control [2], [7]. Although the UBC-PULO prototype has been designed specifically for extreme cases (Table 3), users with

partial paresis (who still satisfy the remaining user requirements stated in Table 3) may also benefit from the powered orthosis.

The next chapter addresses the current state of the UBC-PULO prototype and investigates its mechanical and electrical systems to determine the high-priority improvements and to characterize the UBC-PULO drive system.

Chapter 3 – Experimental Setup

3.1 Introduction

Figure 21 shows the experimental setup required for the validation of the VRSE. In this setup, three independent systems run in parallel while receiving the same user inputs. The goal is to compare the output of the physical prototype, the simulation model (i.e. VRSE) and the ideal output (i.e. the ideal model) under the same test conditions to allow for validation of the VRSE. Note that, the *ideal model* block is able to instantly follow any commanded inputs, producing the ideal case plant output. The three *UBC-PULO control system* blocks shown in Figure 21 are identical and can either run the UBC-PULO control algorithm (i.e. which produces a desired joint angle output after interpreting the user inputs and the current joint angles) or perform tests (i.e. joint position step response). This allows for the UBC-PULO physical prototype, simulation model (i.e. VRSE) and ideal model) to be tested under the same conditions. If the three *UBC-PULO control system* blocks are running the UBC-PULO control algorithm, then the three independent plant (i.e. physical prototype, simulated model and ideal model) outputs such as the end-effector position, velocity and acceleration can be compared with respect to the same user inputs. If the three *UBC-PULO control system* blocks are running a test algorithm such as a step response, then three independent plant outputs can be compared with respect to the same step response input. Thus, to validate the VRSE, the first step is to develop and implement this experimental setup. This chapter investigates the required improvements needed to commission the physical prototype. Chapter 4 provides details on the development and implementation of the required

UBC-PULO hardware upgrades that act as a framework for the experimental setup shown in Figure 21. The development and implementation of the *UBC-PULO Control System block* is discussed in Chapter 5. Using the developed experimental setup, the VRSE validation results are presented in Chapter 6.

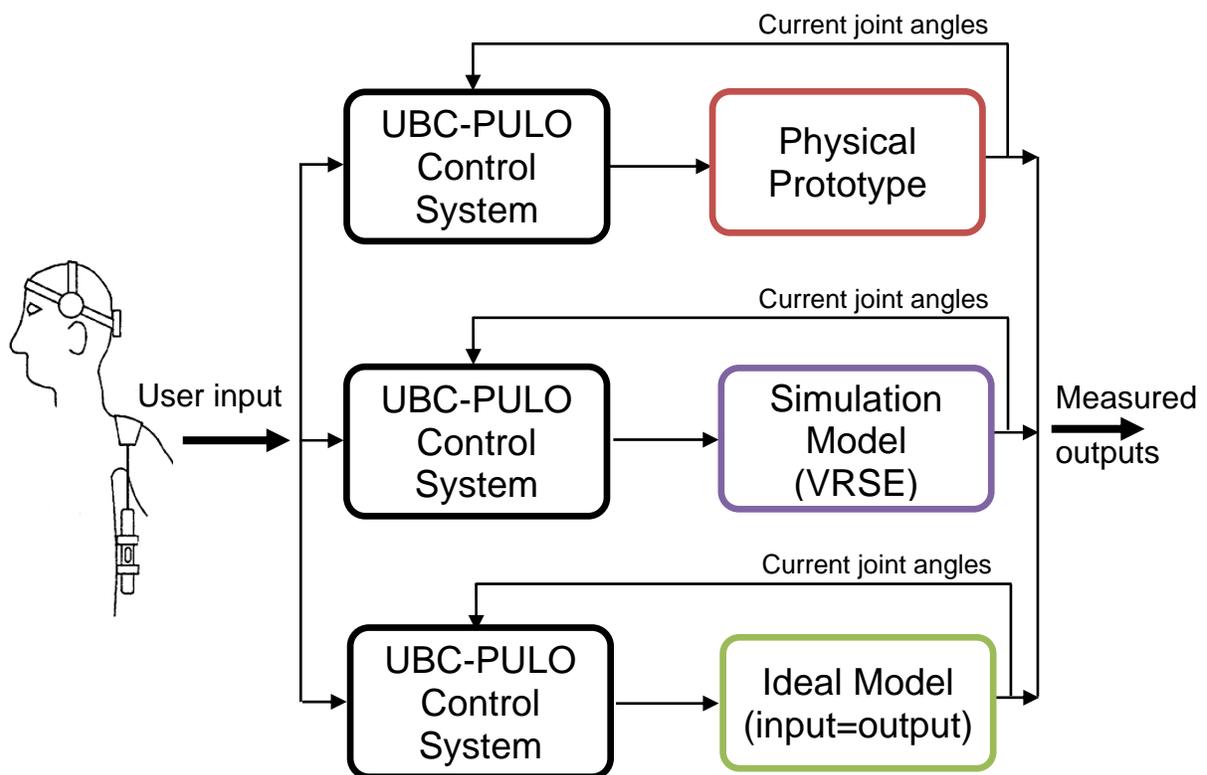


Figure 21. Experimental Setup for VRSE Validation

At the start of this Masters project, the first UBC-PULO prototype was not functional due to mechanical and electrical issues [5]. This chapter investigates the overall UBC-PULO prototype design to address inadequacies of the electro-mechanical system. After examination of all components, the high-priority modifications that were required to commission the prototype are identified. These examinations also led to the

discovery of various improvement opportunities for the UBC-PULO prototype, which are presented as a prioritized list of improvements in Appendix C.

In addition, the results obtained from characterization of the drive system will be presented to complement a research project that was being conducted parallel to this work – i.e. development of a Virtual Reality Simulator (VRSE) [1]. To properly simulate the UBC-PULO prototype, the VRS requires realistic modelling of the drive system, which is composed of the electrical motors, flex-shafts and joint gearboxes. Although the properties of the electrical motors and gearbox are readily available from their documentation (i.e. datasheets and design files), the behaviour of the flex-shafts was unknown. Therefore, a testing apparatus was designed and effort was placed on characterising the flex-shafts in terms of a second order model. Besides the integration of results into the VRS model, the results would also determine if the flex-shafts are suitable for this application and if there are any areas of improvement needed.

3.2 Mechanical review

3.2.1 Evaluating the PULO drive system

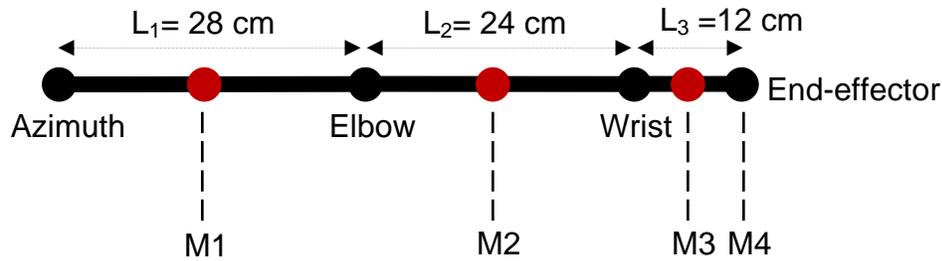
The UBC-PULO drive system consists of the electrical DC motors, gearboxes, joints and flexible drive shafts, which will be reviewed in this section.

3.2.1.1 Motors

The main goal of this section is to determine whether the DC motors utilized by the UBC-PULO meet minimum requirements and are able to provide sufficient torque to satisfy the desired dynamic system response. Since a more valuable and detailed dynamical response study can be performed in the future once the prototype is fully working, it was decided to take a simplistic approach in calculating the motor torque requirements. The problem was broken down into two steps: calculating the maximum angular acceleration that each joint can theoretically produce, and seeing if it satisfies desired angular acceleration requirements. Equation 4 presents the angular acceleration necessary to rotate a joint by 90 degrees in two seconds, which was deemed as a somewhat arbitrary yet reasonable minimum requirement criterion for the application.

$$\alpha = \theta'' \Rightarrow \iint \alpha dt = \iint \theta'' dt \Rightarrow \theta = \alpha \frac{t^2}{2} \Rightarrow \alpha = 2\theta/t^2 = \frac{2 \times (\pi/2)}{2^2} = 0.78(\text{rad/sec}^2) \quad \text{eq.4}$$

To estimate the maximum angular acceleration each joint has to offer, the calculation assume maximum leverage distance (i.e. the exoskeleton's positional configuration requiring the most amount of torque) and motor stall torque values obtained from the datasheet. The moment of inertia calculations were performed by simplifying the human arm and the PULO exoskeleton representation to a point mass system as shown in Figure 22. The effects of a one kilogram load carried by the end-effector were also taken into account. Provided that the motor torque is enough, the exoskeleton can manoeuvre to any workspace position in the desired time frame.



$M1 = 1.2 \text{ Kg (mass of PULO upper limb)} + 1.8 \text{ Kg (mass of average human upper limb [40])} = 3 \text{ Kg}$

$M2 = 0.43 \text{ Kg (mass of PULO forearm)} + 1.1 \text{ Kg (mass of average human forearm [40])} = 1.53 \text{ Kg}$

$M3 = 0.11 \text{ Kg (mass of PULO wrist/grasp)} + 0.4 \text{ Kg (mass of average human hand [40])} = 0.51 \text{ Kg}$

$M4 = 1 \text{ Kg mass}$

Figure 22. Simplified Moment of Inertia Calculation

Table 7 shows the estimate torque and force calculation results based on the motor stall torque (obtained from data sheet), motor gear box ratio, joint gearbox ratio, motor gear box efficiency (obtained from data sheet), and an estimate of the joint gear box efficiency [41]. Equations 5 through 14 demonstrate the calculation method for the presented values. Note that DC motors can only produce stall torque for short periods of time, or else they will overheat. The motors installed on UBC-PULO can continuously produce approximately 45% of their rated stall torque value without overheating according to their datasheet. To estimate the force each DOF has to offer, the calculation used the maximum leverage distance (see equations 10 through 14 for details), motor torque and measured torque due to static friction as presented in equations 8 and 9.

Table 7. Motor Torque Evaluation

Motor function	τ_{stall} (Nm)	N_M	N_J	η_M	η_J	τ_{P_out} (Nm)	L (cm)	τ_{sf} (Nm)	F_{max} (N)	F_{cont} (N)	α_{max} (rad/sec ²)
Grasp	0.011	19.2	30	0.7	0.8	3.52	6	0.72	93.6	46.8	- - -
Wrist	0.011	19.2	30	0.7	0.8	3.52	12	0.30	50.3	25.1	126.4 (eq.13)
Forearm	0.011	19.2	30	0.7	0.8	3.52	15	1.49	32.3	16.1	351.0 (eq.12)
Elbow	0.017	43	20	0.7	0.7	7.16	36	1.92	40.5	20.3	1.71 (eq.11)
Shoulder	0.017	43	15	0.7	0.8	6.14	36	0.34	29.1	14.5	-3.47 (eq.10)
Azimuth	0.011	369.6	1	0.49	1	2.0	64	0.85	5.0	2.5	2.27 (eq.9)

Where,

- τ_{stall} = Stall torque
- τ_{P_out} = Torque available at joint
- τ_{sf} = Torque due to static friction
- N_M = Motor gear box ratio
- N_J = Joint gear box ratio
- η_M = Motor gear box efficiency (manufacturer's ratings)
- η_J = Joint gear box efficiency (estimated based on gear type [49])
- L = Leverage distance
- F_{max} = Maximum static force for leverage distance
- F_{cont} = Continuous static force for leverage distance
- α_{max} = Maximum angular acceleration
- g = 9.81 m/sec²

$$\tau_{P_out} = \tau_{stall} \times N_J \times N_M \times \eta_M \times \eta_J \quad \text{eq.5}$$

$$\tau_{C_out} = 0.45 \times \tau_{P_out} \quad \text{eq.6}$$

$$\tau_{sf} = I_{measured} \times K_m \times N_M \times N_J \quad \text{eq.7}$$

$$F_{\max} = \frac{\tau_{stall} \times N_J \times N_M - \tau_{sf}}{L} \quad \text{eq.8}$$

$$F_{cont} = \frac{0.45 \times \tau_{stall} \times N_J \times N_M - \tau_{sf}}{L} \quad \text{eq.9}$$

Azimuth when arm outstretched horizontally:

$$\alpha_{\max} = \tau_{P_out} / \left(M_1 \left(\frac{L_1}{2} \right)^2 + M_2 \left(L_1 + \frac{L_2}{2} \right)^2 + M_3 \left(L_1 + L_2 + \frac{L_3}{2} \right)^2 + M_4 (L_1 + L_2 + L_3)^2 \right) \quad \text{eq.10}$$

Shoulder roll when elbow at 90° and arm lying in horizontal plane:

$$\alpha_{\max} = \left[\tau_{P_out} - g \left(M_2 \left(\frac{L_2}{2} \right) + M_3 \left(L_2 + \frac{L_3}{2} \right) + M_4 (L_2 + L_3) \right) \right] / \left(M_2 \left(\frac{L_2}{2} \right)^2 + M_3 \left(L_2 + \frac{L_3}{2} \right)^2 + M_4 (L_2 + L_3)^2 \right) \quad \text{eq.11}$$

Elbow flexion when upper arm and forearm in horizontal plane moving up:

$$\alpha_{\max} = \left[\tau_{P_out} - g \left(M_2 \left(\frac{L_2}{2} \right) + M_3 \left(L_2 + \frac{L_3}{2} \right) + M_4 (L_2 + L_3) \right) \right] / \left(M_2 \left(\frac{L_2}{2} \right)^2 + M_3 \left(L_2 + \frac{L_3}{2} \right)^2 + M_4 (L_2 + L_3)^2 \right) \quad \text{eq.12}$$

Forearm rotation while rotating a 30cm rod-shaped load about its center axis:

$$\alpha_{\max} = \tau_{P_out} / \left((M_2 + M_3) \times 0.05^2 / 2 + M_4 \times 0.3^2 / 12 \right) \quad \text{eq.13}$$

Wrist extension while lying in horizontal plane:

$$\alpha_{\max} = \left[\tau_{P_out} - g \left(M_3 \left(\frac{L_3}{2} \right) + M_4 (L_3) \right) \right] / \left(M_3 \left(\frac{L_3}{2} \right)^2 + M_4 L_3^2 \right) \quad \text{eq.14}$$

The calculations demonstrate that all static forces and angular accelerations are within a reasonable range with the exception of the shoulder roll (i.e. it has a negative angular acceleration for a 1 kg end-effector load). Recalculating the maximum shoulder roll acceleration for a 0.75 Kg load resulted in $\alpha_{\max}=1.17$ (rad/sec²). Therefore, the use of a more powerful motor or a gearbox with higher gear ratio is recommended for the shoulder roll joint. A full dynamic study should be performed once the UBC-PULO is functional; however, these preliminary calculations suggest that the motors meet expectations for up to a 0.75 Kg load.

3.2.1.2 Joints

This section outlines the conditions of all six UBC-PULO joints. The joints were inspected in terms of smoothness of motion and amount of backlash. To measure the amount of backlash, each joint was detached from the prototype and a high resolution optical encoder was connected to the output shaft. Using a vice, the input shaft was fixed and the no-load output shaft's play was recorded (encoder angles were recorded in LabVIEW using a National Instrument PCI 6221 DAQ card) while being manually moved. The procedure was repeated several times over various shaft positions and the maximum backlash value was determined. Table 8 summarizes the results. The wrist, forearm and shoulder roll joints demonstrated smooth motion throughout the full range with a backlash less than 1.6 degrees (see Table 8). Although the grasp joint was found to be smooth over most of the range with a maximum 1.5 degree backlash, it was found to have a high rotational friction/resistance. The elbow joint was found to be virtually unmoveable as assembled and had a high amount of backlash, thus needing immediate fixing or redesign.

Table 8. Joints Summary

Joint Function	Joint Gearbox Ratio	Type	Backlash Amount	Condition
Grasp open/close	30	Worm-gear	1.5°	Marginal
Wrist flex/ext	30	Worm-gear	1.6°	Good
Forearm rotation	30	Planetary gear	1.1°	Good
Elbow flex/ext	20	Worm-gear	4.4°	Poor
Shoulder roll	15	Planetary gear	1.2°	Good
Azimuth rotation	1	Directly driven	0°	Good

As presented in Figure 6, the azimuth joint supports the entire radial load due to the lever amplified weight of the users arm and PULO exoskeleton by a bearing block

(i.e. consisting of two ball bearings spaced 3 cm apart on the same axis of rotation). This is of concern since any load on the end-effector would largely affect the radial load on the bearing. Upon further investigation, a 2.9 Nm static torque loss was measured when a 0.5 Kg weight was attached to the end-effector. Thus, a more robust support structure is recommended for the azimuth joint to ensure proper operation under load. It is realized that the radial load on the azimuth joint ball bearings will affect the life expectancy of the ball bearings; however, the effect on ball bearing life expectancy was not quantified as it is outside the scope of this research work. Furthermore, for user safety reasons, mechanical rotational limits need to be implemented on all the joints in the future.

3.2.1.3 Flex shafts

The flex shafts are responsible for transmitting power from the electrical motors to five of the joints; however, due to their spring-damper like nature, they introduce complications in predicting the actual performance of the drive system, making the design of an effective controller more challenging. Since the flex shafts for each joint have a different length and are not guided (i.e. they are not attached to the orthosis structure and can easily bend/twist), the overall effects are unknown and need to be examined. Flex shafts are constructed of densely wrapped flexible metal spring coils; therefore, their performance characteristics may vary based on the direction of rotation. Ideally, an accurate model representation of the flex shafts can be formulated which could be used in UBC-PULO virtual reality simulator [1] and to optimize the control system.

In order to evaluate the flex shaft's response to an applied torque and to see if it is possible to model the flex shaft as a second order system, a test apparatus was designed and data was acquired with respect to various input parameters. Figure 23 and Figure 24 illustrate the test setup, which records the response of the flex-shaft to a step input. The testing process is as follows: A computer running LabVIEW outputs a step voltage using one of the analog channels of the DAQ card. The motor driver board drives the DC motor, which is directly connected to the flex-shaft under test. High resolution optical encoders sense the angular position, producing quadrature signals that are acquired via the DAQ card's counter inputs and recorded in LabVIEW in real-time. The flex shaft is attached to a rotary torque sensor via a shaft adaptor module. The torque sensor transmits torque information to LabVIEW using one of the DAQ cards analog input channels. Weights are attached / detached from to the rotary torque sensor's shaft allowing for an adjustable moment of inertia. Different support patterns on the base wood structure allow the sensors and flex shafts to be moved around to accommodate the testing of various flex-shaft lengths and bending configuration. To modify the response, a movable *flex shaft guide* structure was installed on selected points as needed along the flex shaft path to restrict non-axial motion.

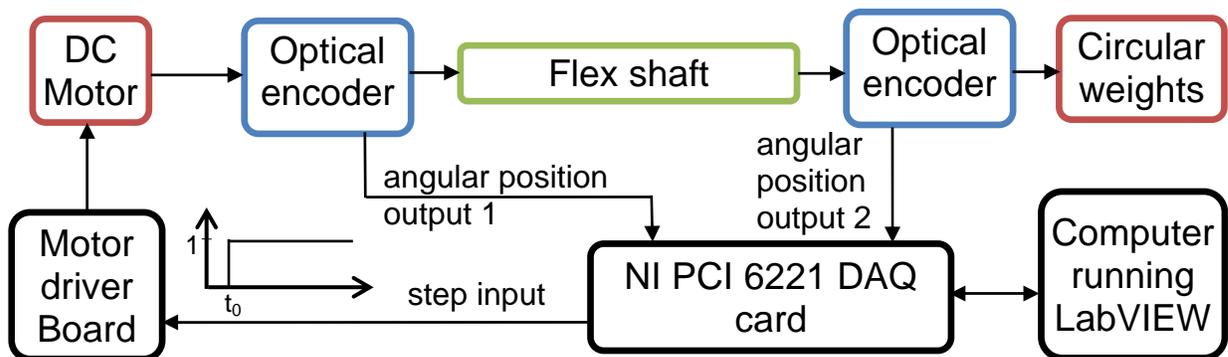


Figure 23. Test Apparatus Overview

The acquisition system was synchronized with a high-frequency external clock signal to make simultaneous data acquisition between the mixture of digital, analog and counter input and outputs possible (i.e. the system starts acquisition with rising edge of clock signal and remains in sync).

Figure 25 shows the LabVIEW user interface which allows the user select the frequency of step input re-occurrence, input step amplitude, direction of rotation, and data acquisition rate (default at 10 KHz). Measured data was saved in a tabulated format (i.e. with a “.LVM” file extension format) which made it easy to analyze in Excel or MATLAB. For further details regarding the test apparatus please refer to Appendix B.

The step response of the flex-shafts was measured for various weights, configurations (i.e. guided /not guided, bent /not bent, free / inside plastic tube casing), and input step amplitudes. The measurements were analyzed and curve fitted to the second order equation shown below (equation 15) by solving for amplitude, natural frequency and damping ratio coefficients. These values were obtained by using the MATLAB curve fitting toolbox`s least square minimization algorithm. The development of the curve fitting MATLAB code and the measurements are explained in reference [42].

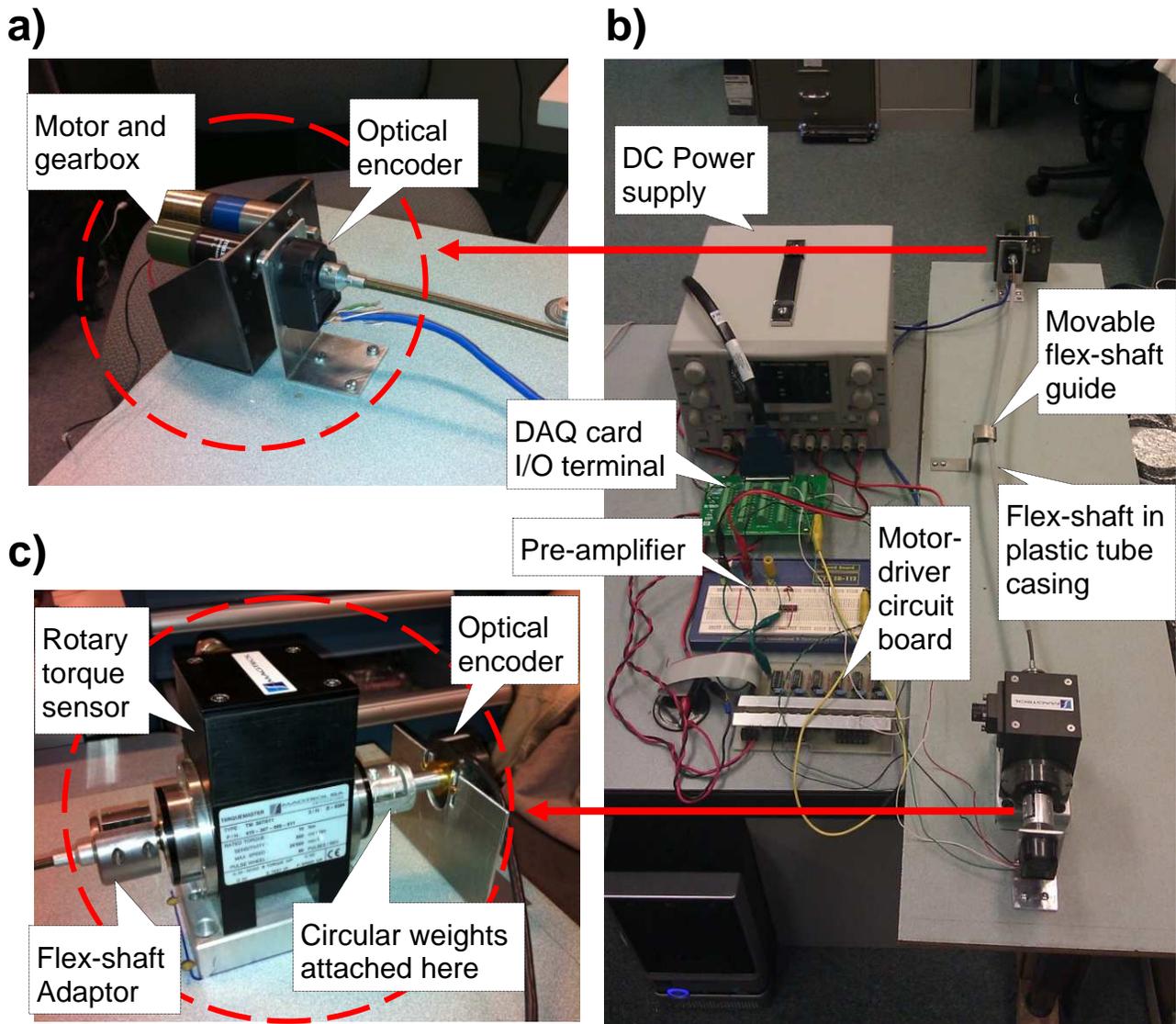


Figure 24. Physical Test Apparatus

$$\theta(t) = e^{-\zeta\omega_n t} \left(A \sin(\omega_n t \sqrt{1 - \zeta^2}) + B \cos(\omega_n t \sqrt{1 - \zeta^2}) \right) \quad \text{eq.15}$$

where,
 θ = angular position
 A, B = amplitude coefficients
 ω_n = natural frequency
 ζ = damping ratio
 t = time

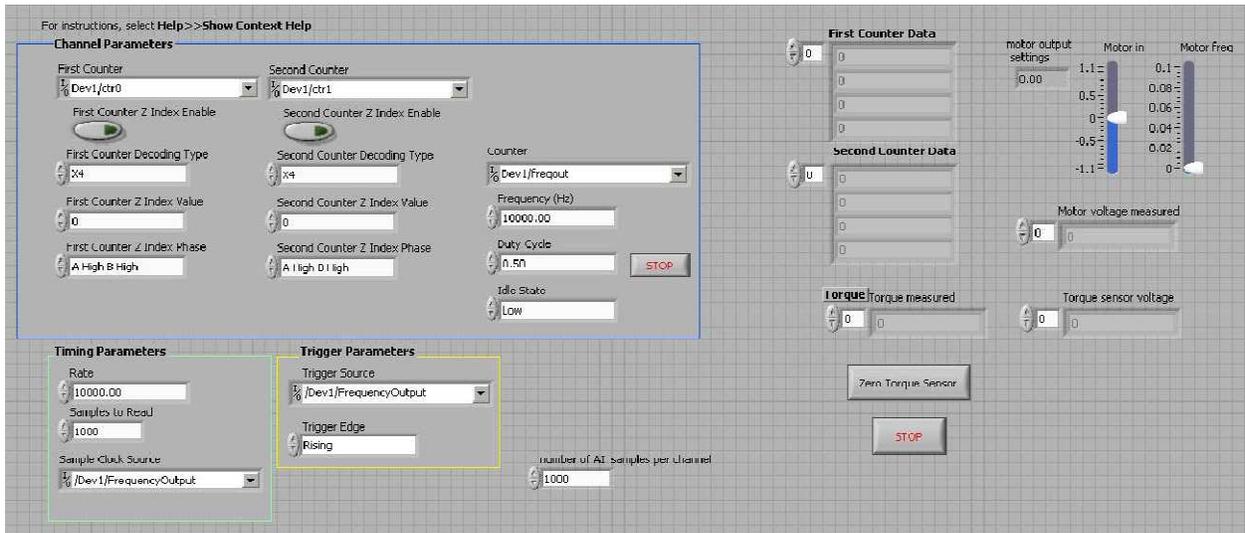


Figure 25. Screen Shot of the Test Apparatus User Interface

Figure 26 shows a sample flex-shaft step response data (in blue) along with the corresponding second order curve fit (in red). More details regarding data and the curve fitting algorithm are presented in reference [42].

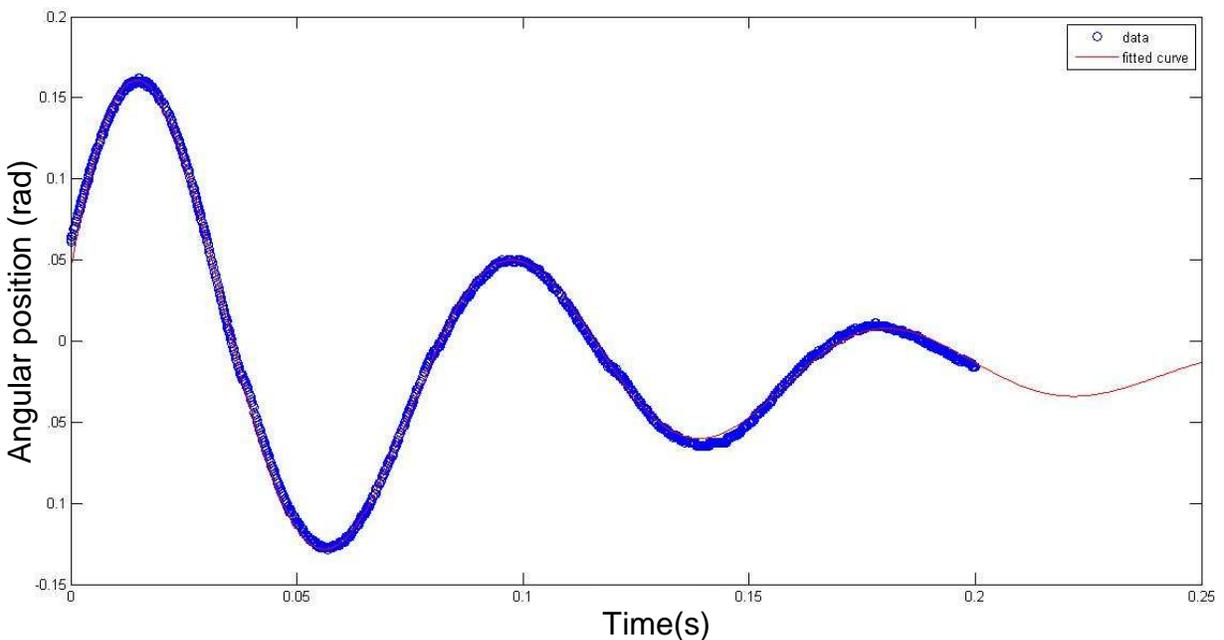


Figure 26. Sample Second-order Curve Fit

$$\omega_n = \sqrt{k/J} \quad \text{eq.16}$$

$$\zeta = c/2J\omega_n \quad \text{eq.17}$$

The study results show that, if the flex-shaft under test is not guided, the step responses significantly vary and are non-repeatable. To produce a consistent step response during orthosis operation, the flex shaft needs to be guided and use of tube casing is highly recommended (i.e. plastic tubes casing) to suppress non-axial rotational movements and vibrations throughout the length of the shaft. It was noted that the presence of bending in a flex-shaft that is not guided produces unpredictable results as well. Moreover, for the torque and weight ranges tested, the flex-shafts do not show any differences in directionality (i.e. same second order response in either direction). Despite the fact that the flex-shaft's step response can be approximated by a second order transfer function for a given moment of inertia or length, they exhibit non-linear behaviour when either moment of inertia or length is varied. According to equations 16 and 17, the natural frequency and damping ratio should decrease with an increase in moment of inertia; however, this relationship was not satisfied which can be explained by the non-linear performance characteristics of the flex-shafts. Due to time limits imposed on this research work and ongoing higher priority objectives such as commissioning of the prototype, the non-linear behaviour of the flex shafts was not further investigated.

3.2.4 Mechanical review summary

This section addressed the short comings of the UBC-PULO prototype's mechanical system through examination and testing of various components. The repair or redesign of the elbow joint was identified as a required high-priority improvement for commissioning the system. Further details with respect to the mechanical evaluation of the prototype and the resulting prioritized list of improvements are presented in Appendix C.

3.3 Electrical review

3.3.1 Evaluating the PULO electrical system

The UBC-PULO electrical system consists of the two user control interfaces, two circuit boards and feedback angle sensors, all of which will be reviewed in this section.

3.3.1.1 Control interfaces

The head user interface works well and is sufficient for a proof-of-concept design; however, the optical inclinometers are quite large and obtrusive (Figure 11) considering the fact that they sit on the user's head. Each encoder has four wires connected to it, making a total of eight wires running from the users head to the electrical box, which can be uncomfortably thick. Ideally, the head interface should be small, light and have as few wires as possible that could hamper use.

3.3.1.2 Processing and motor driver circuits

Initial tests of the electronics showed that the high level electrical circuit was non-functional. When searching for replacement parts it was found that the microprocessors, EEPROM and various other ICs used are obsolete. The fact that all the microprocessors were selected to be programmed in Assembly language to increase processing speed made any modifications to the UBC-PULO software very time consuming and inefficient, i.e. not a desirable feature during the UBC-PULO prototyping and testing cycle. Another issue found with the high level controller board was large processing delays in the inverse kinematics processor. The control algorithm processor scans the user control interfaces every 65 milliseconds and the inverse kinematics processor (Figure 15) requires 300 milliseconds of computation time to update and output the joint angles. This results in a 365 millisecond delay between the user command input and any associated reaction of the exoskeleton. The nominal human reaction time is 50-60 milliseconds [43] in controlling a closed loop system; therefore, ideally the lag between user command and endpoint motion should be less than this response time.

The low level control and motor driver board uses class AB amplifiers which are less efficient than the state of the art motor driving technologies due to their analog nature. The dissipated energy is emitted as heat from the transistors. In order to keep the transistor's junction temperature below the allowable limit, heat sinks are used, adding weight and bulk to the system. Additionally, the efficiency of the motor driving system is important as motors are one of the main sources of power consumption, affecting battery life. Each motor draws up to 1.1 Amps of current at 6 Volts, thus

demanding up to 39.6 Watts (i.e. $P=VI=6\text{V}\times 1.1\text{A}$) from the motor driving system. This is large source of power consumption, especially when compared to the power required for the rest of the electrical system, which consumes only 5 Watts. Therefore, a more efficient driving system would not only extend the operational time between battery recharging, but also help reduce weight and bulk (i.e. by requiring a smaller heat sink).

Based on this testing and review, it was decided that the entire electrical system had to be redesigned immediately due to a non-functional high level board which consisted of obsolete components and overall outdated technology. Furthermore, the inflexible software and hardware strictly limited the future development of the UBC-PULO prototype, so a more flexible technology was to be researched for replacement of this previous system.

3.3.1.3 Feedback sensors

Potentiometers need a stable supply voltage as any noise or voltage instability directly affects their output. Since the UBC-PULO is powered by a battery, sudden spikes of current needed for driving the motors causes fluctuations in the supply voltage due to the internal resistance of the battery. The produced noise caused by the large currents passing through the inductive traces wires and the fluctuations in the supply voltage will be directly reflected on all potentiometer outputs. The problem can be mitigated by monitoring the supply voltage in real-time and accounting for these fluctuations during the microprocessor calculations. Also, introducing large capacitors at

the motor inputs and implementing a voltage regulator will greatly enhance the stability of the supply voltage. The potentiometer outputs are analog signals and thus any noise from the environment or the UBC-PULO electrical circuits will affect the accuracy of the angular position readings (i.e. the long wire connecting the angle sensor to electrical box acts as an antenna picking up noise). Moreover, potentiometers are both subject to mechanical wear and are sensitive to environmental temperature. Besides being subject to the limitations mentioned above, the potentiometers installed on the UBC-PULO are obsolete and have demonstrated poor electrical contact between the wiper and resistive material surface (i.e. due to degradation of resistive material surface). The potentiometers should be replaced with better angle sensors that are compact, contactless, digital and immune to environmental conditions such as temperature and small supply voltage variations.

3.3.2 Electrical review summary

In this section the UBC-PULO prototype's electrical system, including the control interfaces, electrical boards, and angle sensors, were examined. The redesign of the electrical system is a required high-priority modification to allow for the commissioning and characterization of the prototype. In addition, the redesign and implementation of new joint angle sensors are recommended. The examination of the overall UBC-PULO electrical system also resulted in the identification of a prioritized list of improvements which is presented in Appendix C.

3.4 Summary

In this chapter the electrical and mechanical short-comings of the first UBC-PULO prototype were identified. The required high-priority improvements consist of the repair of the elbow joint and the redesign of the electrical system. Implementation of these high-priority improvements would allow for the commissioning and characterization of the first UBC-PULO prototype. The examination of the system also resulted in the identification of a prioritized list of mechanical and electrical improvements, which is presented in Appendix C in Table 26 and Table 27, respectively. In addition, the results obtained from characterization of the drive system, to complement the VRSE, were presented. Chapter 4 presents the process and results as to how the UBC-PULO hardware was upgraded to meet the required high-priority improvements and to allow for the integration of the VRSE with the hardware.

Chapter 4 – Upgrading the UBC-PULO Hardware

4.1 Introduction

This chapter provides details on the electrical and mechanical changes that lead to the commissioning of the UBC-PULO prototype. In addition, the development process and implementation of a new electrical system will be presented. This new electrical system will act as the framework required for the implementation of the experimental setup explained in Section 3.1 (Figure 21), i.e. the experimental setup required for validating the VRSE against the physical prototype.

4.2 Mechanical improvements

The repair of the elbow joint was identified as a high-priority improvement in Chapter 3. Upon inspection of the elbow joint it was determined that the worm gear was unable to rotate about its axis due to the combined effects of a broken spring pin and a bent shaft. The shaft was straightened and the spring pin was replaced, making the elbow joint functional. The joint is now functional; however, the current design's shaft is not robust enough to tolerate the forces applied on the joint (i.e. the reason it originally bent). Due to custom made components and other mechanical constraints it was not possible to replace the existing shaft with a thicker shaft and spring pin. Thus, the elbow joint was identified as requiring a redesign. The redesign of the elbow joint was not pursued due to time limitations and other priorities (i.e. design of a new electrical

system). In addition to this work, flex-shaft casing solutions and a finalized forearm re-design have been proposed in Appendix B and Appendix C respectively.

4.3 Development of a new UBC-PULO Electrical system

The electrical system of the UBC-PULO has been completely redesigned, including modifications to the electrical boards, user control interfaces and angles sensors. These enhancements will be presented in detail in this section.

4.3.1. User control interfaces

The shoulder user interface was malfunctioning due to a broken Printed Circuit Board (PCB) trace (Figure 12). A new PCB was assembled using new electrical components. The repaired shoulder interface was then tested and proved fully functional.

A new head interface was designed using a two-axis MEMS based inclinometer IC chip. The VTI Technologies SCA1020 IC, shown in Figure 27, is only 8.4 mm x 15.6 mm x 5.08 mm in dimension and can provide a two axis inclination angle with a $\pm 90^\circ$ range. The chip outputs a linear Y-axis, Z-axis analog output that is directly proportional to gravitational acceleration experienced by each rotational axis (i.e. Figure 28-a). Figure 27-c demonstrates the acceleration sensed by the chip in various spatial orientations. As shown in Figure 27-b, the electrical connections are simple with no need of external

components. Once the SCA1020 is powered-up, a corresponding Y-out and Z-out analog signal will result as the output signal.

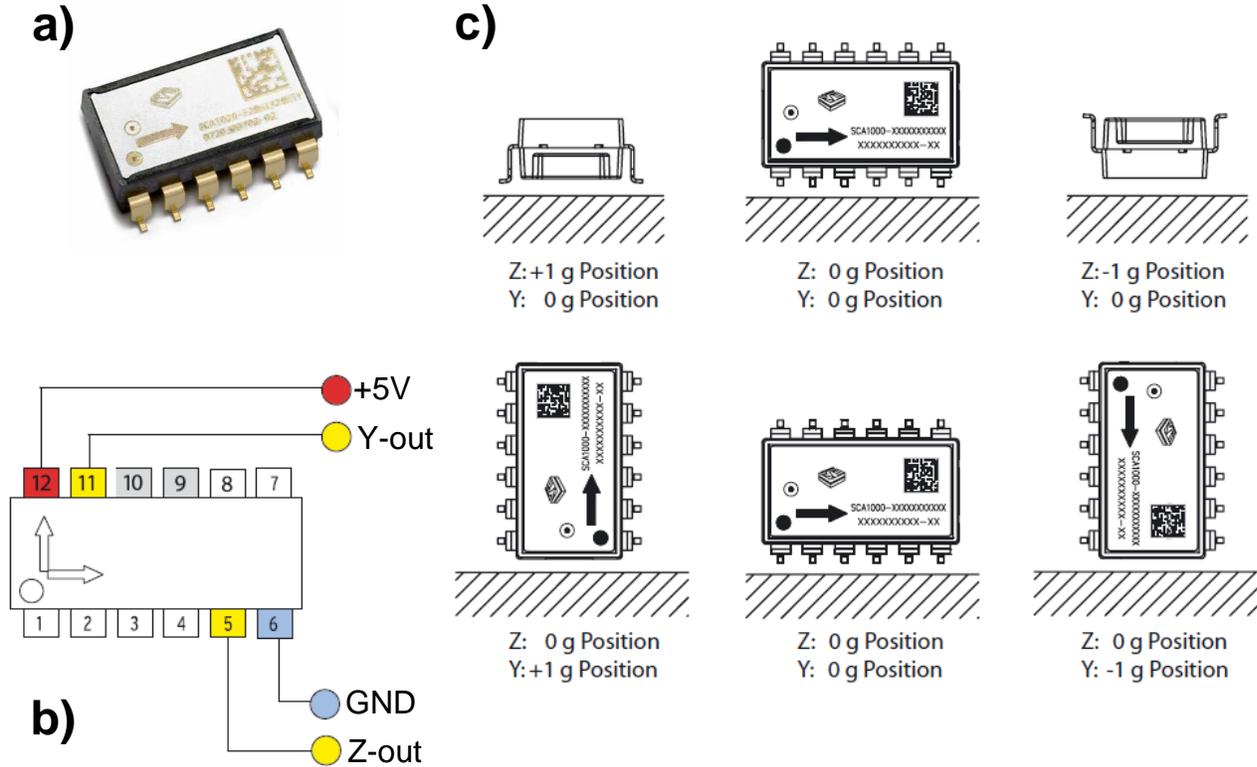
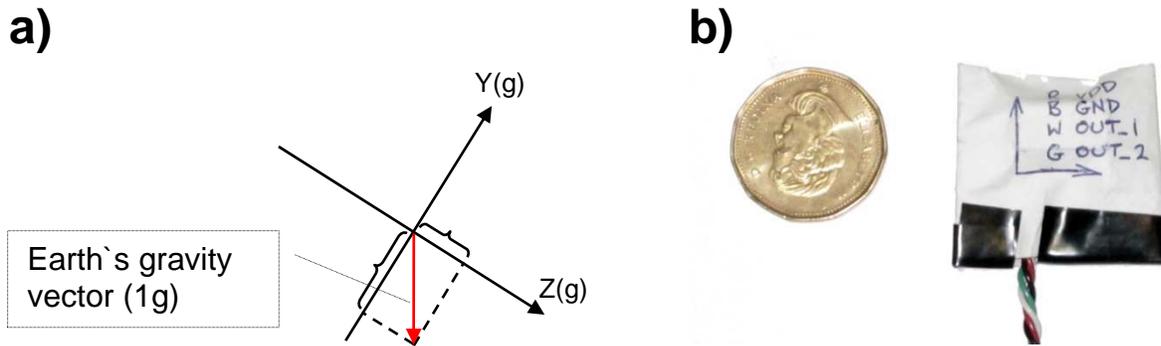


Figure 27. a) VTI Technologies SCA1020 b) Electrical Connection Diagram c) SCA1020 Z-Y Operation Illustration (1g = 9.81 m/sec²)

The angular inclination position in each axis can be calculated using the following equations:

$$\alpha_z = \arcsin\left(\frac{Z_{out} - Offset}{Sensitivity}\right) \quad \text{eq.18}$$

$$\alpha_y = \arcsin\left(\frac{Y_{out} - Offset}{Sensitivity}\right) \quad \text{, where sensitivity} = 1.2V/g \text{ and offset} = 2.5 V \quad \text{eq. 19}$$



**Figure 28. a) Gravitational Acceleration Y-axis and Z-axis
b) New Head Interface Assembly**

Figure 28-b demonstrates the new *head user interface* assembly. In addition to being light and compact, the new head user interface is very power efficient, requiring only 20 mW nominally (i.e. 4mA nominal supply current) which makes it 25 times more efficient than the old design. For better noise immunity, the SCA1020's Serial Peripheral Interface (SPI) can be used in the future as the cable connecting the new head interface to the electrical box is vulnerable to noise.

4.3.2. Feedback angle sensors

New angle sensors were designed and installed on the UBC-PULO prototype. The new design is based on Austria Micro System's AS5045 high resolution magnetic encoder chip which is able to sense the angular position of a contactless rotating magnet located directly above the chip (i.e. Figure 29-b). This technology was selected for the following main reasons:

- It is extremely compact (currently claimed to be the smallest angular encoder in the world)
- It is contactless and not subject to wear
- It has a high resolution (i.e. 0.0879°)
- It is customizable to fit the UBC-PULO application, especially for future designs where it can be built as part of the joint unit (i.e. as opposed to an externally attached component)
- It provides a variety of digital output including quadrature encoded digital signals, and Pulse Width Modulated (PWM) signals
- It has a high common mode rejection of external magnetic field (i.e. external magnetic interference do not affect output)
- It is affordable at \$7 per chip

Since each of the UBC-PULO joints already have an output shaft designed for the potentiometers, it was decided that the new angle sensor design should be through-hole mounted as well. Pulse Width Modulation (PWM) is an industry standard fixed-frequency variable-width digital signal as shown in Figure 29-c. The PWM output type was selected as it has the advantage of sending data over one wire and being immune to noise (due to its digital nature). The AS5040 was electrically connected as shown in Figure 29-a. Angular information can be retrieved by the electrical system by calculating the duty cycle (i.e. $\text{angle} = \text{duty cycle} \times 360^\circ/100$).

One of the challenges in the design of the new angle sensor was minimizing the alignment error between the center of the magnet and the magnetic encoder chip. As shown in Figure 29-d, a maximum of $R_d = 0.25$ mm is recommended by the

manufacturer as a larger misalignment will cause non-linearity in the output. For example, a 0.7 mm misalignment will have a ± 1 degree of non-linearity error. Also, to have the optimal magnetic field strength, the magnet should be hovering between 0.5 to 1.5 mm above the chip.

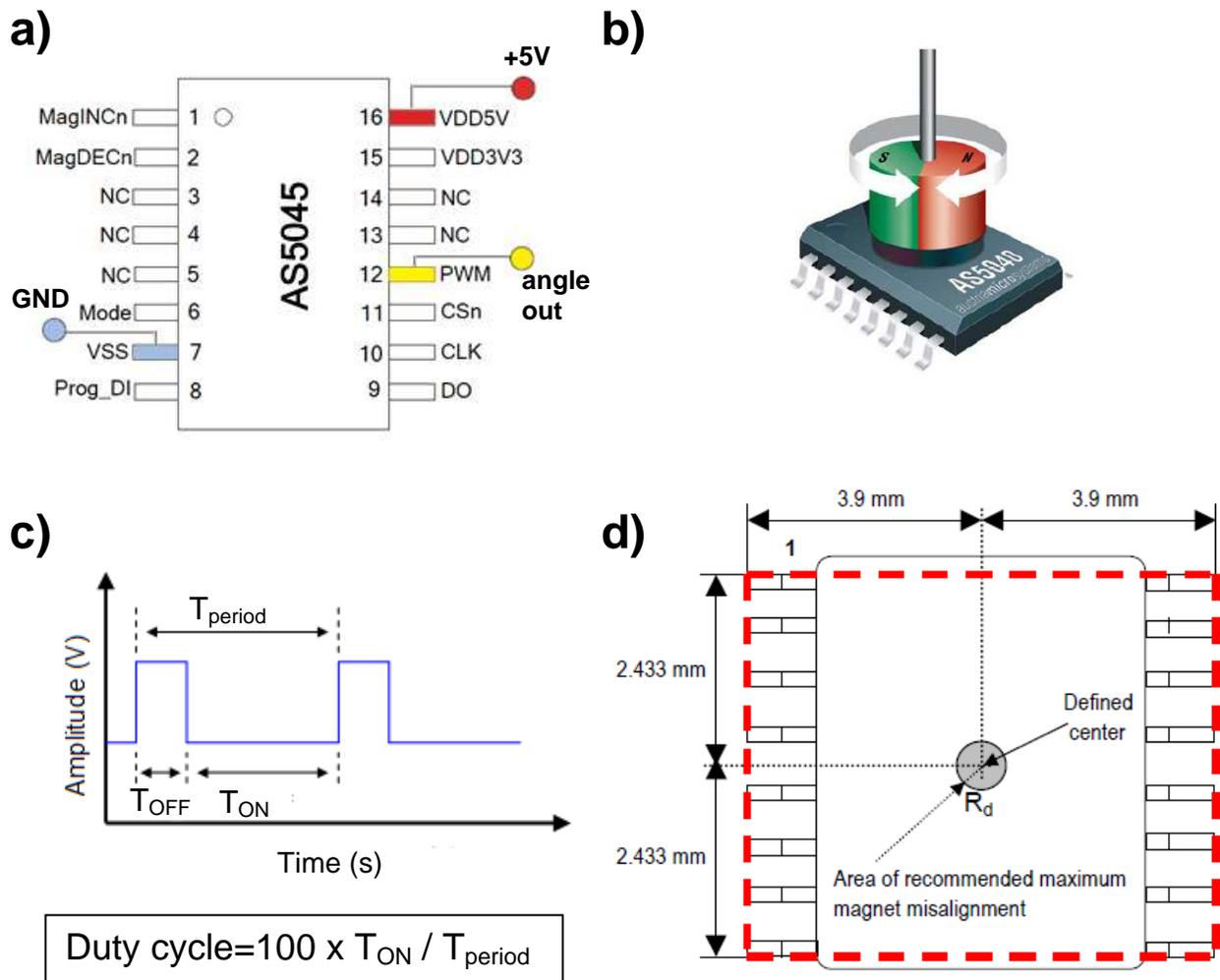


Figure 29. a) AS5045 Electrical Pin-out b) Contactless Magnet Angle Detection c) Pulse Width Modulation d) Recommended Magnet Misalignment

The new angle sensor design, which adheres to all the mechanical constraints mentioned above while maintaining a through-hole and compact design, is shown in Figure 30. A new encoder housing was also designed and constructed. Due to time and

budget limitations, other design parameters were low cost and ease-of-manufacturability; therefore, use of off-the-shelf products and low machining time were preferred. The use of 1:1 Acetal gears allowed for the through-hole mounting so that the potentiometers could be readily replaced by the new angle sensors.

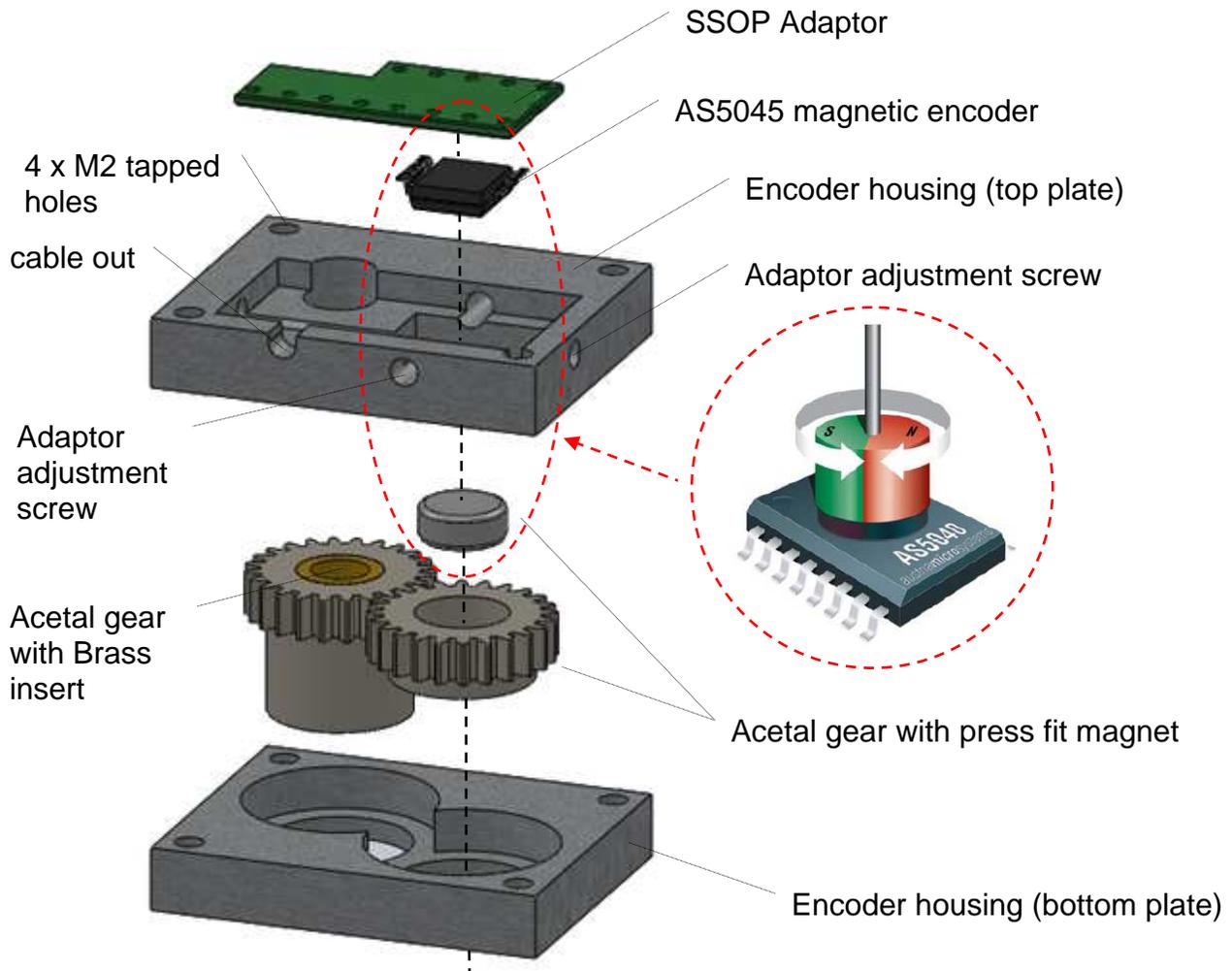
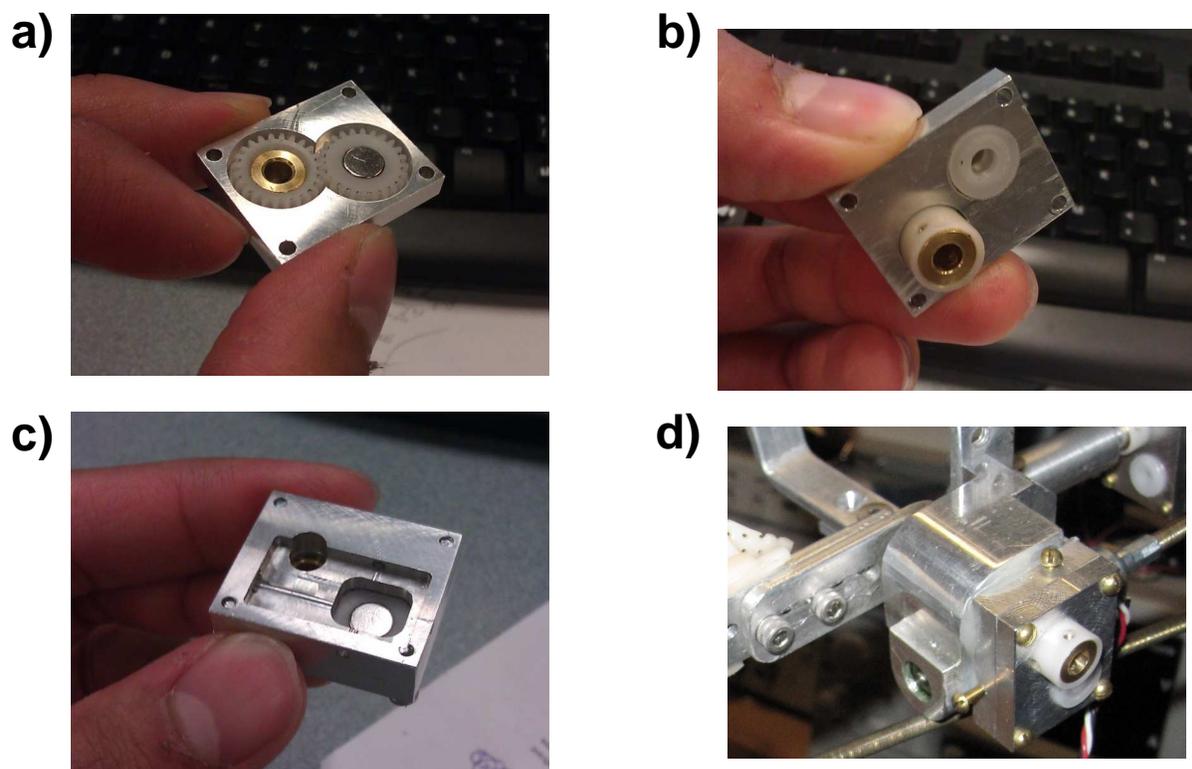


Figure 30. Encoder Housing Design

The off-the-shelf PCB adaptor plate (SSOP adaptor) provides a secure base for the magnetic encoder to be affixed by soldering, and also electrically connects the magnetic encoder to the output cable. Since the machining and assembly of

components was performed manually, manufacturing error was inevitable. To satisfy the recommended magnet misalignment tolerances, four adjustment screws allow for fine-tuning of the encoder positioning. Figure 31 shows the final assembly and a sample implementation of the new joint angles sensors on the prototype's wrist joint. All assembled angle sensors were tested, demonstrating robust and accurate angular position readings.



**Figure 31. a) 1:1 Acetal Gears with Press Fit Magnet b) Encoder Housing (Top)
c) Assembled Angle Sensor d) Sample Angle Sensor Implementation**

To minimize misalignment error, a specialized encoder positioning technique was developed. Using the setup explained in Section 3.2.1.3, the encoders were coupled

with a DC motor running at constant speed while real-time angular velocity graphs were monitored. Next, the encoder position adjustment screws were fine-tuned such that minimal fluctuations were observed on the velocity graph.

4.3.3. Design requirements

A main goal of the electrical system redesign, in addition to commissioning the UBC-PULO, was to provide a framework on which the experimental setup (see Figure 21) could be implemented. It was also realized that building a highly customized electrical system during the prototyping stages of the UBC-PULO would be ineffective as it discourages efficient development of the system's software and hardware. For development cost and time-efficiency purposes, a flexible electrical platform is in the best interest of the project, realizing that a highly-customized design should be left for the product development cycle as it requires a finalized version of the overall system to optimize for consumer production. Table 9 summarizes the design requirements for the new electrical system. The system requires a minimum of eight analog inputs to be able to realize the current sensor (see Appendix E) and the new head interface inputs. The number of required digital signals are 24 as 6 digital inputs are required by the new angle sensors, 4 by the previous head interface (i.e. in case the new design didn't work well), 2 inputs for the shoulder interface and 12 I/O to drive for the drives ICs (see Appendix E). Having an electrical box that was larger than the previous electrical system (25cm x15 cm x10 cm) was deemed undesirable. Additionally it is essential that the new hardware is able to directly run MATLAB, Simulink and SimMechanics (i.e. or to

convert these to understandable code for real-time execution) in order to be integrated with the VRSE.

Table 9. New Electrical System Design Requirements

Requirements	Description
Flexible hardware	<ul style="list-style-type: none"> • At least 8 analog inputs, 24 digital I/O • Preferably a modular design where parts can be added/replaced as needed • Can drive at least six 3 Watt electric motors
Flexible software	<ul style="list-style-type: none"> • High-level software for time-efficient programming • Integrated with VRSE (i.e. MATLAB, Simulink and SimMechanics)
Portable	<ul style="list-style-type: none"> • Battery powered and portable with overall dimensions less than 25x15x10 cm

4.3.4. Design overview

Various real-time solution products were explored and MathWork's XPC Target product was selected as the best fit to the UBC-PULO electrical design requirements. XPC Target is a MATLAB toolbox which allows real-time execution of Simulink and Stateflow models on a target computer. As shown in Figure 32, a typical XPC Target based system is composed of three sub-systems:

- The *host computer* running XPC Target, Simulink, Real-Time Workshop, and a C compiler

- The *target computer* running a real-time XPC Target operating system named XPC Kernel
- The hardware which is composed of other electrical systems that interface with the target computer



Figure 32. XPC Target System

XPC Target allows the host computer Simulink and Stateflow models to be downloaded and executed in real-time on the target computer. The target computer can be connected to other systems such as data acquisition cards (DAQ cards) to provide physical inputs and outputs (I/O) that interface with the hardware. A large library of drivers is available through XPC Target allowing for real-time modules to be easily interfaced with the system in a plug-and-play fashion.

Further XPC Target features include:

- Stand alone operation using XPC Target Embedded Option (i.e. no host computer needed)
- Tuning and viewing of parameters before or during real-time execution
- Real-time data analysis and visualization on an external monitor provided that the target computer has a graphics card
- Record and log data on the host and/or target computer

The *host computer* can be selected from the variety of computers types ranging from personal computers to industrial grade computers. An embedded Single Board Computer (SBC), which is a full computer built on a single Printed Circuit Board (PCB), was selected as the *host computer* because of both compactness and high performance benefits. SBCs come in many different form factors (i.e. the specifications of the motherboard such as dimensions, types of ports, number of ports, etc), such as CompactPCI, PXI, PICMG, PC/104, PC/104+, EPIC, EBX, etc. It was decided that the PC/104+ form factor would be the most suitable for the UBC-PULO application. PC/104+ is a standard board format used in the embedded computer industry where mobility, stability and real time control are essential. Each PC/104+ board can be stacked in a plug-and-play fashion to any other PC/104 or PC/104+ module as presented in Figure 33. PC/104+ modules incorporate both a PCI bus and an ISA bus, thus exhibiting superior performance compared to PC/104 boards which only use the ISA bus (i.e. PCI bus is about sixty times faster than the ISA bus). Any number of desired PC/104+ DAQ card modules can be simply stacked onto the host computer to provide I/O. If the DAQ cards fall within the predefined XPC Target library, their Simulink

library *input* and *output* blocks will be readily available (i.e. no need to develop driver code).

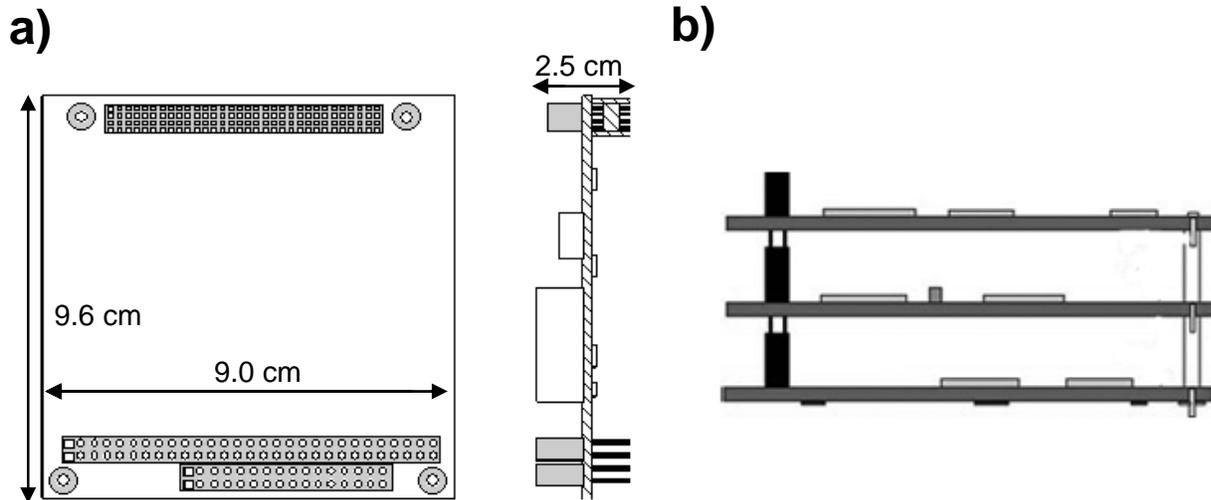


Figure 33. a) PC/104-Plus Form Factor b) PC/104 Modules Stacked

Figure 34 shows the new electrical system design with XPC Target at its core. The UBC-PULO electrical box is composed of a SBC board, two DAQ cards and a custom made PCB. The SBC board interacts with the host computer and the DAQ cards. This allows the UBC-PULO software, developed in MATLAB and Simulink, to be downloaded and executed on the PC/104+ SBC board which is stacked together with the PC/104+ DAQ cards to send and receive I/O. The DAQ cards interact with the custom-made PCB board via various I/O to drive the motors and receive signals from the head and shoulder user interfaces, and the joint angle sensors. Additionally, current sensors were implemented to measure real-time current passing through each motor, and thus are also passed as I/O to the DAQ cards.

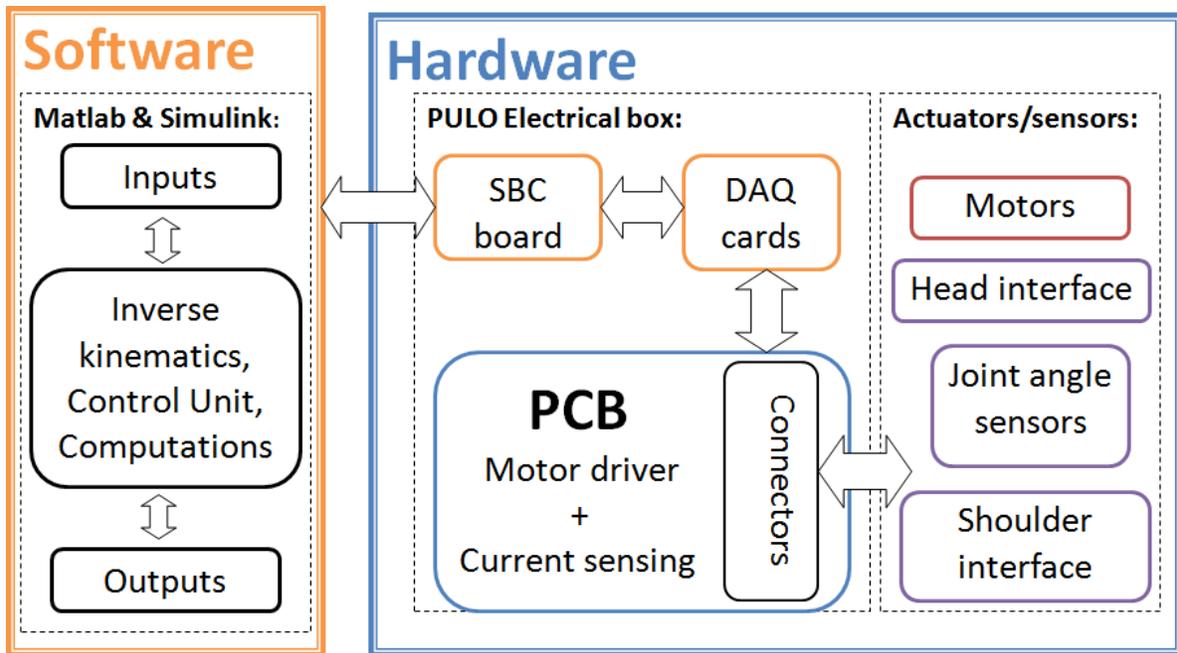


Figure 34. Electrical System Design Overview

Figure 35 presents the custom-made PCB in more detail, showing that the DAQ card signals go through *I/O connector 1* to control the motor driver circuit, while signals from the *sensor connector* and current sensor circuitry go directly through *I/O connector 1* and are received by the DAQ cards.

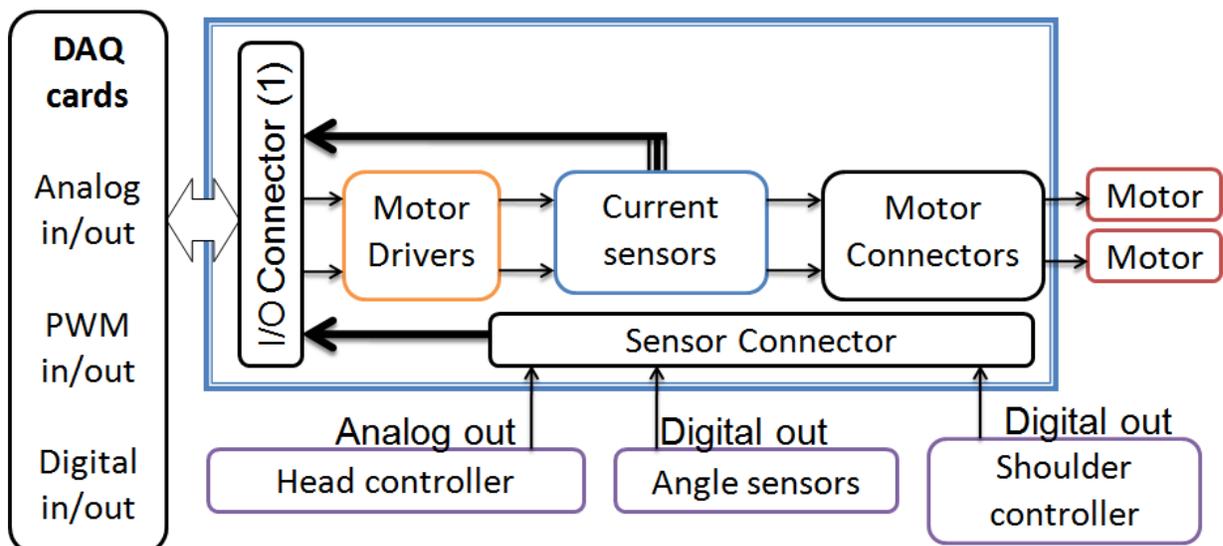


Figure 35. PCB Design Overview

The process of component selection and PCB design is an iterative cycle where many interdependent electrical and mechanical constraints have to be satisfied. For example, the overall size of the PCB would only be known once all of the components are known and positioned in their optimal location. The optimized location depends on the type of I/O (i.e. analog, digital, high power or noise-sensitive signals), location of other components, as well as other requirements such as heat dissipation. As a result, changing any one component usually leads to changing several other components. Therefore, the total number of I/O and type of required I/O could only be determined once all the components were selected and their PCB design finalized. For details with respect to the development process and implementation of the custom PCB boards please refer to Appendix E.

4.3.5. Selection of SBC board and DAQ cards

Any x86 computer, such as an Intel 386/486/Pentium or AMD K5/K6/Athlon, is XPC Target compatible. In addition, the host computer's ethernet chipset has to be supported by XPC Target to allow Ethernet connection between the host and target computer. A list of compatible Ethernet chipsets can be found on the MathWorks website¹. In addition, larger CPU cache sizes and faster CPU speeds lead to higher

¹ http://www.mathworks.com/products/xpctarget/supported-hardware/xPC_Target_Supported_Ethernet_Chipsets.pdf

performance. The Advantech PCM 3362 PC/104+ SBC board was selected with its main features listed shown in Figure 36.

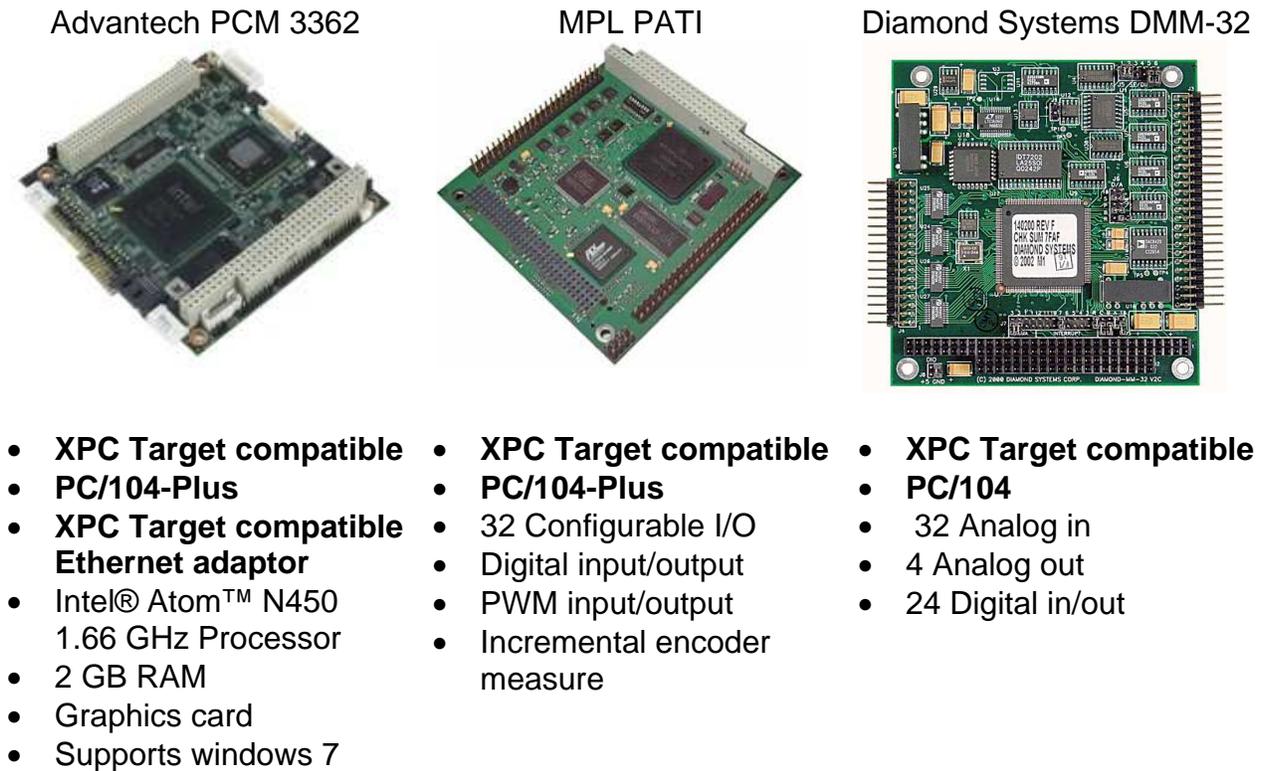


Figure 36. Selected SBC Board (on left) and DAQ Cards

The selected DAQ cards and their main features are presented in Figure 36, where both have a PC/104 format and are XPC Target compatible allowing for plug-and-play operation. The type and number of required I/Os satisfy the custom PCB design. The DAQ cards were selected once the final custom PCB design was known. For a full understand of their functions and reason for selection, please refer to Appendix E.

4.3.6. Setup of the SBC board and DAQ cards for XPC Target

This section explains a brief overview of the steps involved in setup of the XPC Target based system. This includes the setup of the PCM-3362 SBC board, the host computer and the DAQ cards.

The PCM-3362 SBC board had to be setup to run the MATLAB based XPC Kernel 5.0 operating system. This setup involved performing PCM-3362 SBC's Jumper settings (i.e. electric conductors/switches used for initial setup and configuration of the board), installing a 2 Giga-byte DDR2 RAM, and connecting a power supply, mouse, keyboard, monitor and external CD-ROM device to the system. Next, the target computer was powered up and various BIOS settings were performed to allow the system to recognize the CD-ROM and USB devices that are connected to it. Next, the XPC Target operating system (i.e. XPC Kernel 5.0) was installed and various BIOS setting were performed to ensure high performance real-time operation. The last step involved the removal of the keyboard, mouse and external CD-ROM device. At that point, the SBC board would start up in the XPC Kernel 5.0 operating system ready to communicate with the host computer using an ethernet crossover cable- allowing the UBC-PULO software to be downloaded and executed on the system. More details with respect to the setup of the target computer is presented in Appendix D.

The host computer setup involved the installation of MATLAB, Simulink, XPC Target and a C compiler. In addition, initializations and required adjustments were

performed on XPC Explorer (i.e. the XPC Target user interface), Simulink, and the host computer's network settings. These adjustments would not only allow for the UBC-PULO software to be developed in the host computer environment, but would also enable the host computer and target computer to communicate via a crossover ethernet cable- thus, the UBC-PULO software could be developed on the host computer and download/executed on the target computer. Once the host computer setup was final, a "bootDISK" was created on a USB memory device which included the installation files for the XPC Kernel 5.0 operating system and the host computer information – this was required for the setup of the target computer. More details with respect to the setup of the host computer is presented in Appendix D.

Due to their PC/104 format, both the DMM-32 and MPL-PATI DAQ cards can be simply stacked onto the SBC board and be readily used in a plug-and-play fashion. The default manufacturer jumper settings are suitable for this application and no initial setup was necessary on either board. More information regarding the DAQ cards can be found on their corresponding datasheet.

4.3.7. The new electrical system

The custom PBCs, which are explained in detail in Appendix E, were ordered through Sierra Proto Express and assembled by hand in-house. Figure 37 demonstrates the final assembly of the new UBC-PULO electrical system. The capabilities were tested and demonstrated full functionality meeting all expectations. By

meeting the requirements listed in Table 9, the newly developed UBC-PULO electrical system serves as a platform for the experimental setup with the main features as listed in Table 10.

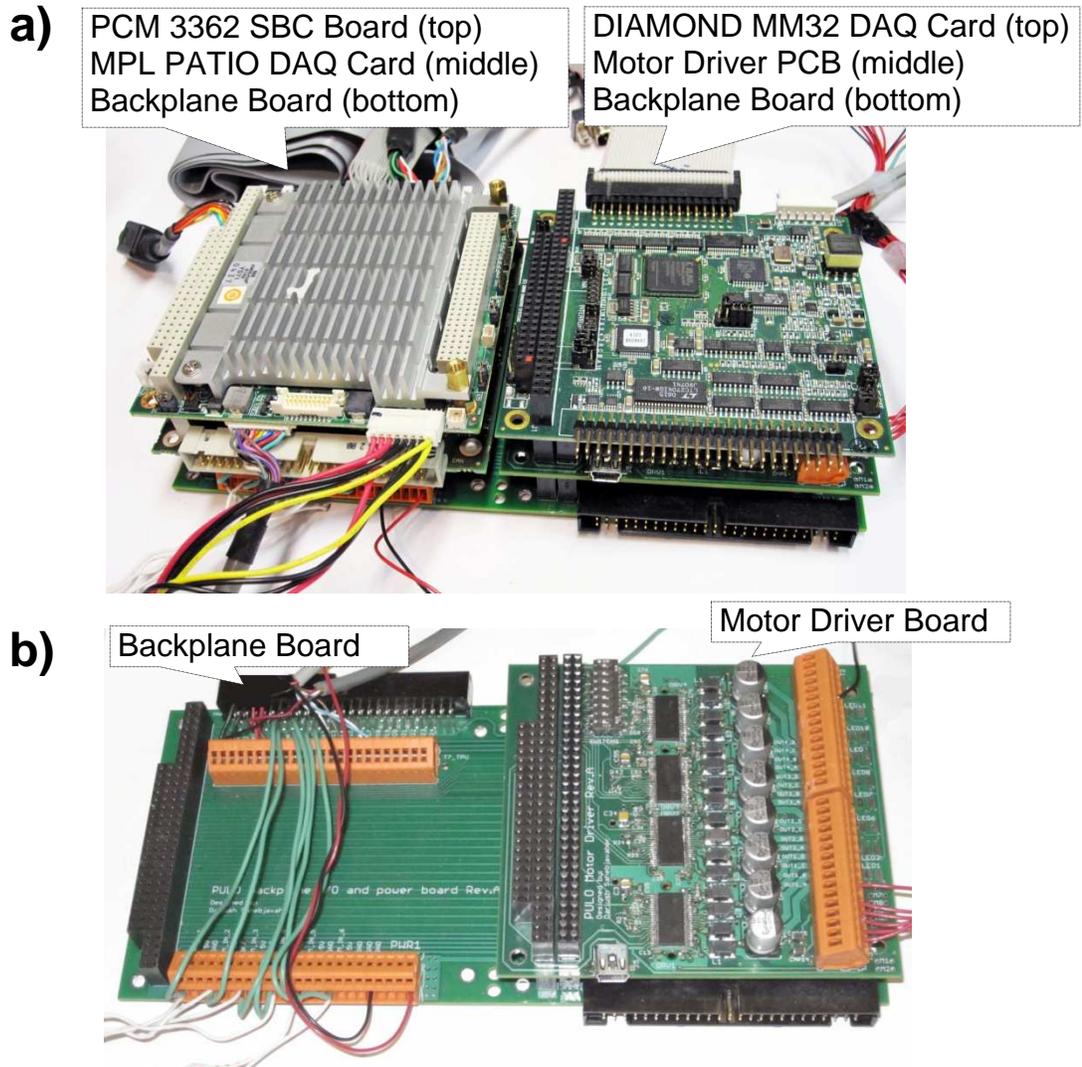


Figure 37. a) New UBC-PULO Electrical System b) Assembled Backplane PCB Board and Motor Driver PCB Board

Table 10. New Electrical System Features

Features	Description
Flexible hardware	<ul style="list-style-type: none">• Stackable plug and play PC/104 modules• Various digital and analog I/O readily available via backplane PCB board• Modular design allows UBC-PULO modules to be added/replaced as needed• High processing power allows for fast sampling rate and execution of complex models
Flexible software	<ul style="list-style-type: none">• MATLAB/Simulink/XPC Target
Portable	<ul style="list-style-type: none">• Battery powered and portable system
Efficient motor driving circuitry	<ul style="list-style-type: none">• PWM based motor driver• Power efficient and compact motor driver board design capable of driving up to 8 motors
Other	<ul style="list-style-type: none">• New angle sensor design• New head interface design

4.4 Conclusion

This chapter provided details on the electrical and mechanical modifications that lead to the commissioning of the UBC-PULO hardware. The electrical system was redesigned to meet all requirements and the elbow joint was repaired, making the system hardware-ready. Since the new electrical system is capable of executing MATLAB, Simulink and SimMechanics models in real-time, the experimental setup can be implemented on the system, which was the main objective of the electrical redesign besides commissioning the physical prototype.

The next chapter explains how the UBC-PULO software, in addition to the other new capabilities, was integrated into the new UBC-PULO hardware leading to a fully operational experimental setup.

Chapter 5 – Developing UBC-PULO Software in XPC Target

5.1 Introduction

The UBC-PULO software was developed in MATLAB and Simulink using XPC Target. Once Simulink and MATLAB were setup as instructed in Appendix D, Simulink models can be readily downloaded and executed on the host computer. Since the selected DAQ cards are XPC Target compatible, their input and output blocks are available within the XPC Target Simulink Library and can be readily used (please refer to Section 4.3 for further details). The overall goal of this chapter is to highlight the development and implementation of the experimental setup's software that enables the validation of the UBC-PULO VRSE – i.e. running the physical prototype, the simulation model (VRSE) and ideal model in parallel (see Figure 21). First, details about the UBC-PULO software are presented- i.e. this software is required for the commissioning of the physical prototype. Then, the experimental setup's software, which integrates the UBC-PULO control system, the simulation model (VRSE), the ideal model and the physical prototype, is presented (see Figure 21).

5.2 UBC-PULO software

Figure 38 presents an overview of the UBC-PULO software developed for the new electrical hardware. The *PULO hardware block* (i.e. the “plant” block marked in red) incorporates all the XPC target I/O blocks thus enabling it to communicate with the

motors driver board, angle sensors, user interfaces and various other electrical components by sending and receiving analog, digital and PWM signals. The *Task Control Algorithm block* receives the shoulder interface, head interface and joint angle information in order to perform neutral zone detection, determine the UBC PULO operation mode, monitor the overall system status, ensure user safety and produce the desired change in end-effector position as requested by the user (refer to Section 2.3.2.2 for further details). The *Inverse Kinematics Block* receives the real-time joint angles and recalculates the Inverse Jacobian (see Appendix A). The desired change in end-effector position received from the *Task Algorithm block* and recalculated Inverse Jacobian is used to produce the desired joint angles. The *PID controller block* incorporates six parallel PID loops to minimize the angular position error between the desired joint angles and the real-time measured joint angles. The output of the *PID controller block* is sent to the *PULO hardware block* where it is converted to a PWM signal driving the six PULO motors through the motor driver board. The *Constant Tuning Parameters* block contains many adjustable parameters that determine the overall properties of the software and hardware. This chapter will explain the implementation of each of the identified blocks in Simulink and also reveal new capabilities added to the UBC-PULO software.

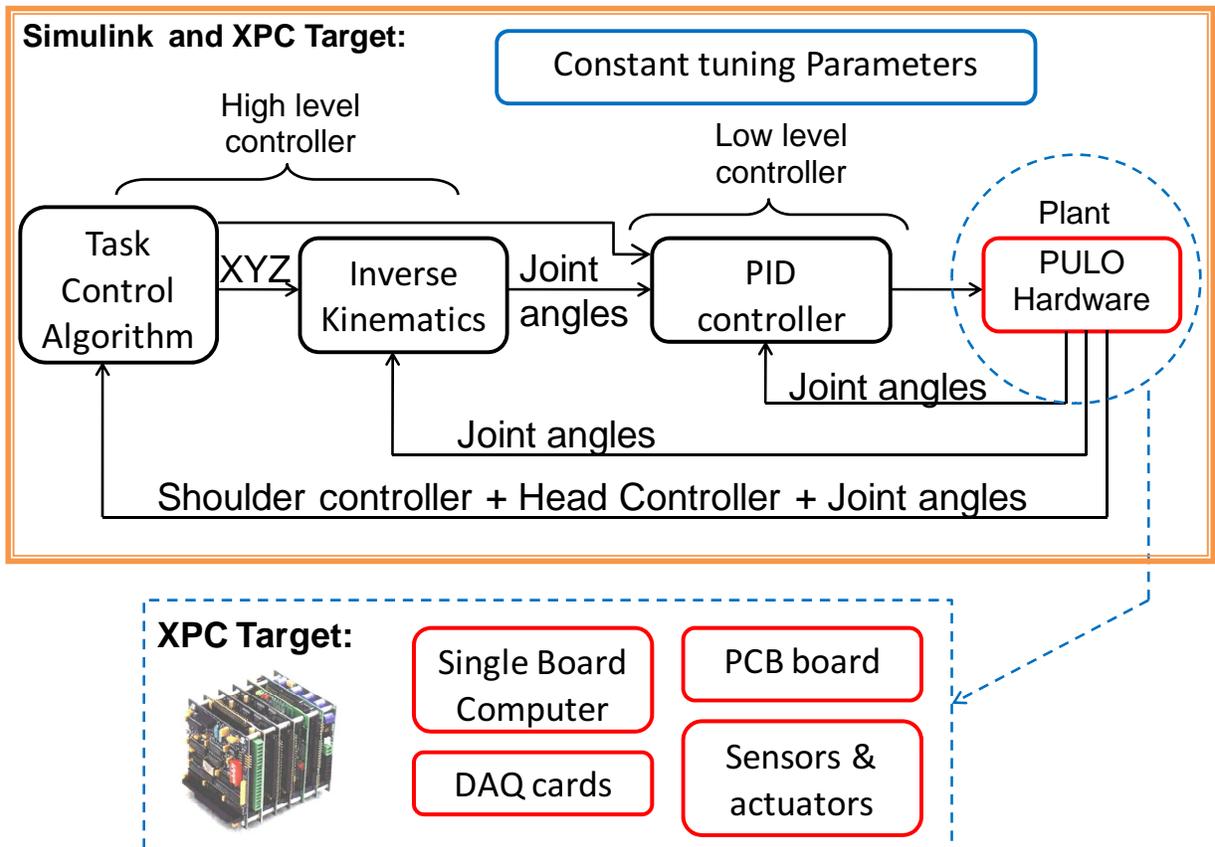


Figure 38. UBC-PULO Software Overview

5.2.1 Task control algorithm

Figure 39 and Figure 40 provide screen shots of the Simulink blocks that form the UBC-PULO *task control algorithm* (see Section 2.3 for task control algorithm theory of operation). The *head controller* block acquires the head interface signals and outputs the head tilt in the X and Y axis in degrees.

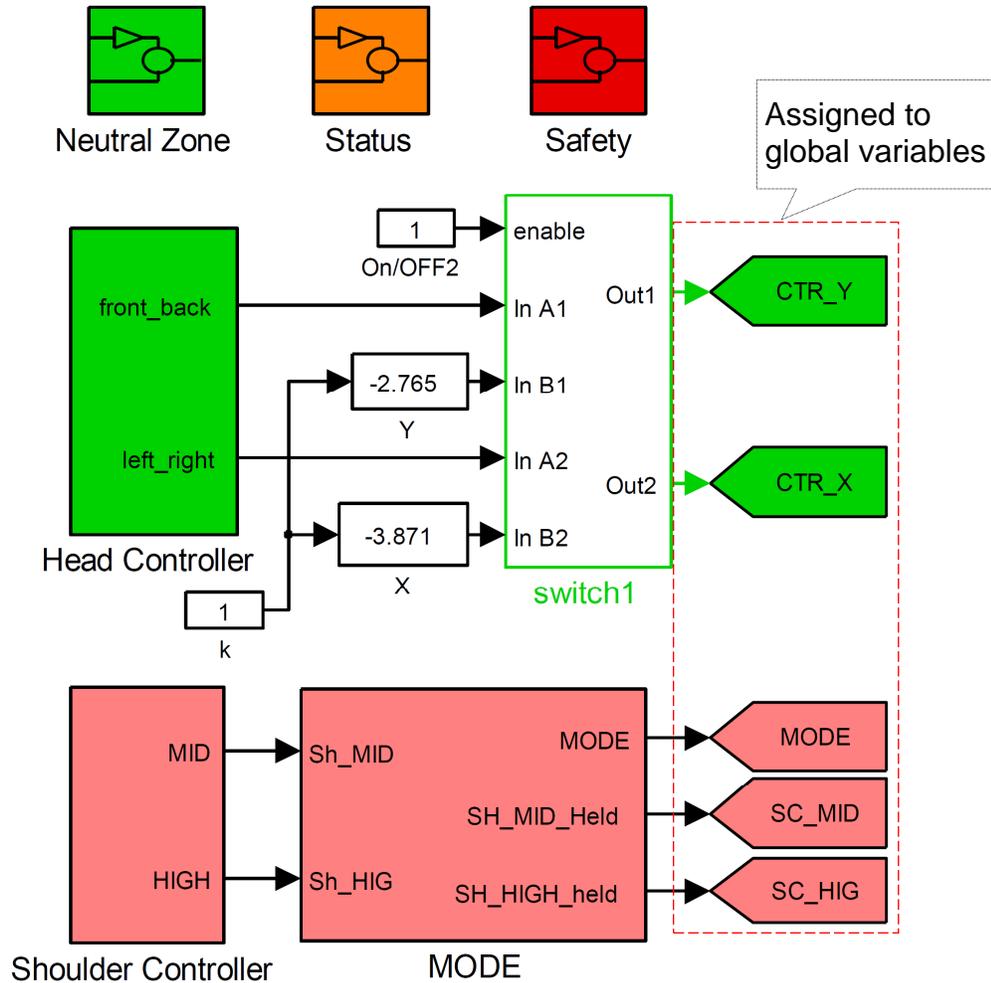


Figure 39. Task Control Algorithm (Part 1)

Switch blocks were introduced in various parts of the system to allow for the flexibility of switching between manual inputs and real-time inputs during the testing stage of the UBC-PULO software. In this case, enabling the *switch* block (i.e. a value of “1” assigned to the “enable” input of *switch* block) assigns the real-time X and Y head angles to the *CTR_X* and *CTR_Y* global variables. However, if the *switch* block is disabled (i.e. a value of “0” assigned to the “enable” input of *switch* block), a manual head angle is assigned to the *CTR_X* and *CTR_Y* global variables.

The *Shoulder Controller* block acquires the two optical switch shoulder interface digital inputs and forwards this information to the *Mode* block (see Section 2.3.2.3.2 for details regarding the shoulder interface). The *Mode* block differentiates between a quick shoulder shrug and an extended shoulder shrug. A short shoulder shrug updates the global mode variable to the next operation mode (see Section 2.3.2.2 for details on operation modes). An extended shoulder shrug in the *MID* position assigns a value of “1” to *SC_MID* global variables as long as the shoulder controller is held in the middle position. Similarly, an extended shoulder shrug in the *HIGH* position assigns a value of “1” to *SC_HIGH* global variables as long as the shoulder controller is held in the high position. Otherwise, the *SC_HIGH* and *SC_MID* global variables are set to a value of “0”.

The *Neutral Zone* block detects whether the user’s head is within the neutral zone or not and forwards this information to various system blocks. The *status* block monitors the UBC-PULO battery level, motor currents and detects if the motor driver board is in an Over Temperature or Fault condition (see Appendix E for details regarding the motor driver board). The *Safety* block enforces a user defined soft limit on the UBC-PULO joints (i.e. the commanded joint angles cannot exceed a predefined range of motion). Additionally, the head interface’s X and Y angular velocity and angular acceleration are calculated by the *Safety* block by computing the derivative and double derivative of the *CTR_X* and *CTR_Y* signals. If the magnitude of the head interface’s angular velocity or angular acceleration exceeds a predefined limit, the head interface values will be ignored to ensure user safety.

Figure 40 shows the blocks responsible for performing the Neutral mode, table-top mode, functional mode and orientation mode calculations (see Section 2.3 for details on operation modes). These blocks receive the *Mode* block and *head controller* block output global variables (designated by a red box in Figure 39) and produce the desired output. When the neutral mode is activated, the *neutral mode* block simply acquires a sample of the current joint angles and holds the same six joint angles values at the output. This will essentially hold the exoskeleton in the same position.

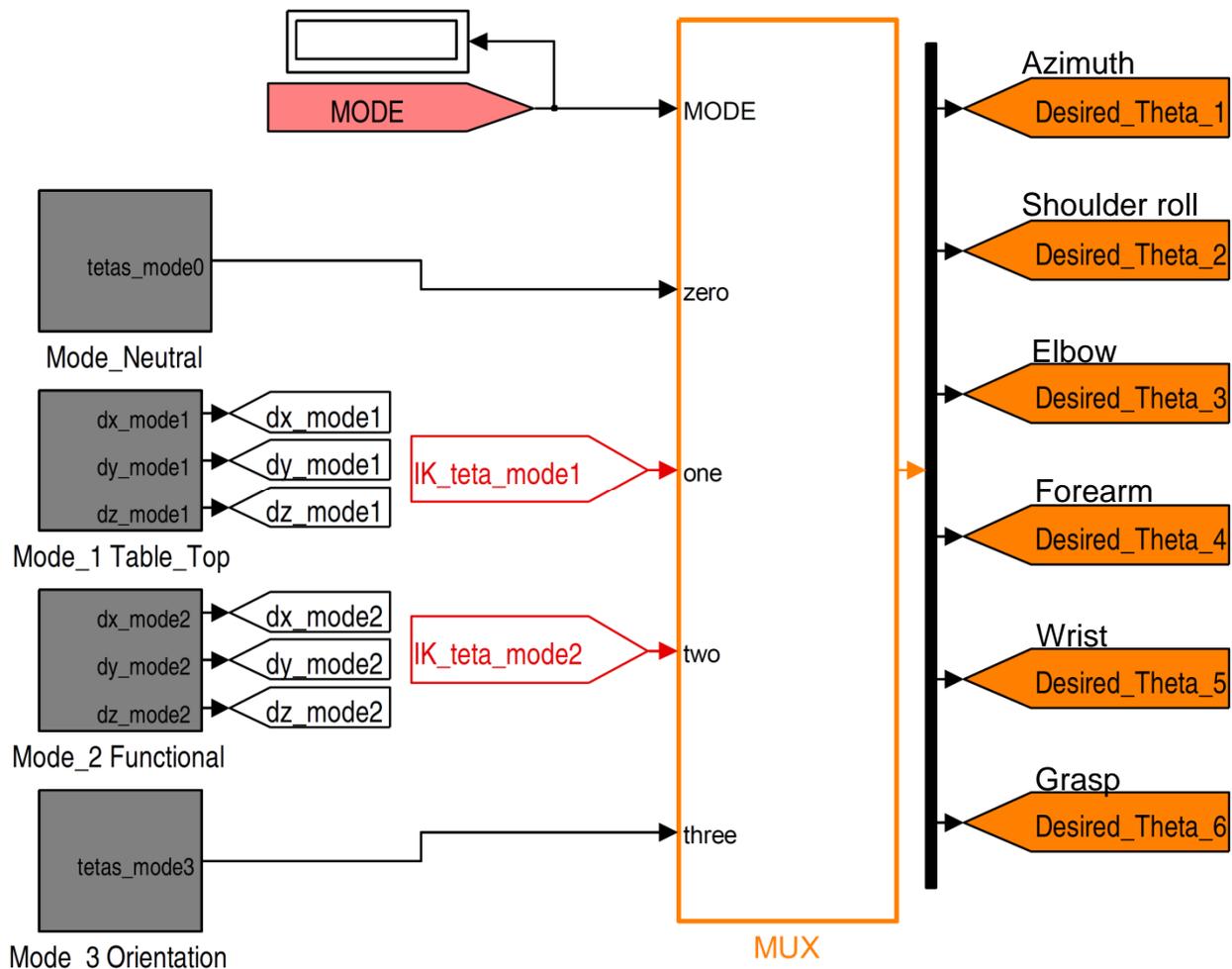


Figure 40. Task Control Algorithm (Part 2)

Similarly, the *orientation mode* block outputs the desired output angles based on user input. The *Table Top mode* block interprets the real-time user inputs to produce a proportional incremental change in the end-effector X, Y and Z position and assigns them to dx_mode1 , dy_mode1 and dz_mode1 global variables respectively. The inverse kinematics block accepts the dx_mode1 , dy_mode1 and dz_mode1 variables, calculates the desired joint angles and outputs them to the IK_teta_mode1 global variable. Similarly, the *Functional mode* block produces the incremental dx_mode2 , dy_mode2 and dz_mode2 variables and sends them to the inverse kinematics block which outputs the calculated the desired joint angle outputs through the IK_teta_mode1 global variable. The *Multiplexer* block (*MUX* block) selects one of the desired joint angles depending on the mode of operation and assigns them to the six *Desired_Theta* global variables which are sent to the *low level controller and plant block*.

5.2.2 Inverse kinematics

As shown in Figure 41, a *MUX* block is responsible for forwarding the incremental end-effector position change (i.e. ∂x , ∂y and ∂z) from the corresponding *Mode* block to the *Inverse Kinematics* block based on the mode of operation. The *Inverse Kinematics* block receives the current azimuth, shoulder roll, elbow angles, and the incremental change in end-effector position (i.e. ∂x , ∂y and ∂z) to calculate the desired output angles. Since the end-effector position can be controlled by the first three joints (i.e. azimuth, shoulder roll and elbow), the forearm, wrist and grasp angles remain constant during table top and functional operation modes. The output of the Inverse Kinematics

calculations consists of the six joint angles which get assigned to *IK_teta_mode1* and *IK_teta_mode2* global variables.

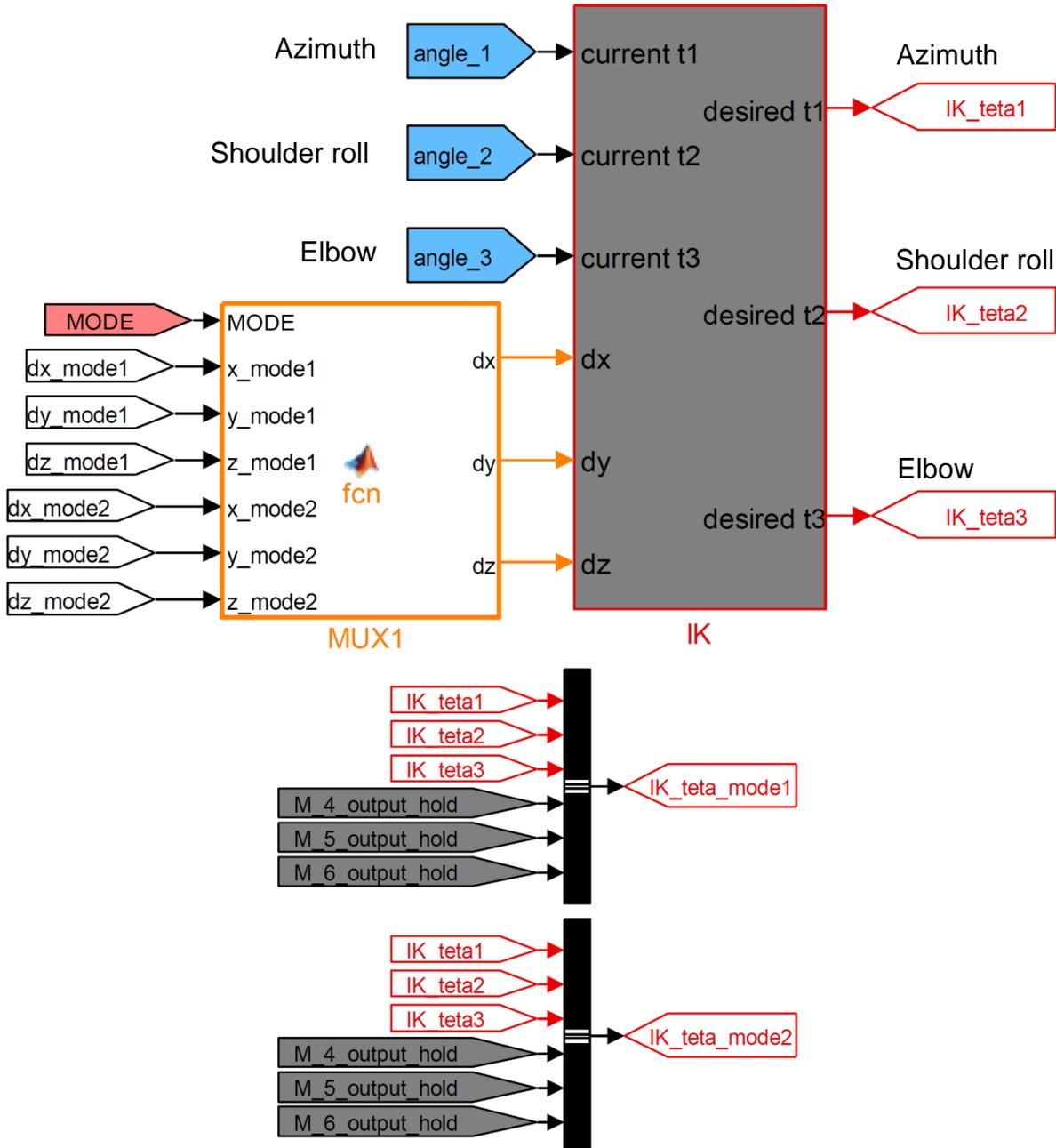


Figure 41. Inverse Kinematics

Figure 42 shows the sub blocks that make up the *Inverse kinematics* block. The joint angle inputs are converted to radians allowing the *Jacobian* block to calculate the Jacobian matrix for the current angles. Next, the inverse Jacobian is calculated and multiplied by the desired end-effector incremental change vector. The output of the *Matrix Multiply* block is the desired joint angle changes in radians. The desired joint angle changes are converted to degrees and added to the current joint angles to produce the desired joint angles, which are the *Inverse Kinematics* block's output. Additionally, the Jacobian's condition number [46], which indicates the overall inverse kinematic stability, is assigned to the global variable. Note that the *Jacobian* block can be parametrically updated for any Denavit-Hartenberg table (DH table) through the MuPAD script as mentioned in Appendix A.

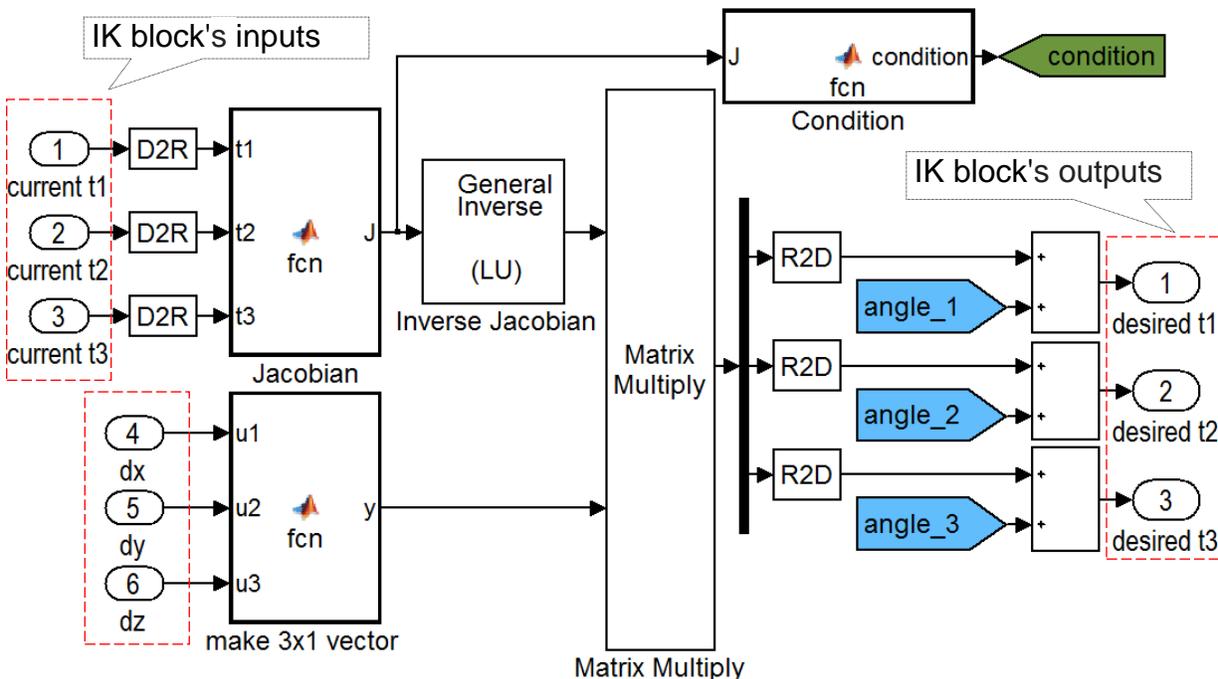


Figure 42. Inverse Kinematics Calculations

5.2.3 Low level controller and plant

The *Plant* block (see Figure 67) receives the six desired joint angles as the input to drive the manipulator to the target position. In addition, the *Plant* block acquires the current joint angles and assigns them to *angle_x* global variables at the output. A *switch* block allows choosing between a manual angle entry for testing purposes (i.e. step input) or calculated desired angles (i.e. *Desired_Theta_x*) during operation. The *Safety and IK stability* block ensure overall inverse kinematic stability by monitoring the condition number [46] and enforcing software joint limits that prohibit the UBC-PULO to reach a singular inverse kinematics configuration. The *Safety and IK stability* block also limits the joint range of motion, which is user-definable, to ensure user safety. If the inputs pose no problem, they will simply be output to the *plant* block. However, if an input causes the inverse kinematic to enter a singular or near singular configuration, the previous output is held until the next input is evaluated. Section 5.2.4.3 explains the *Safety and IK stability* in further details.

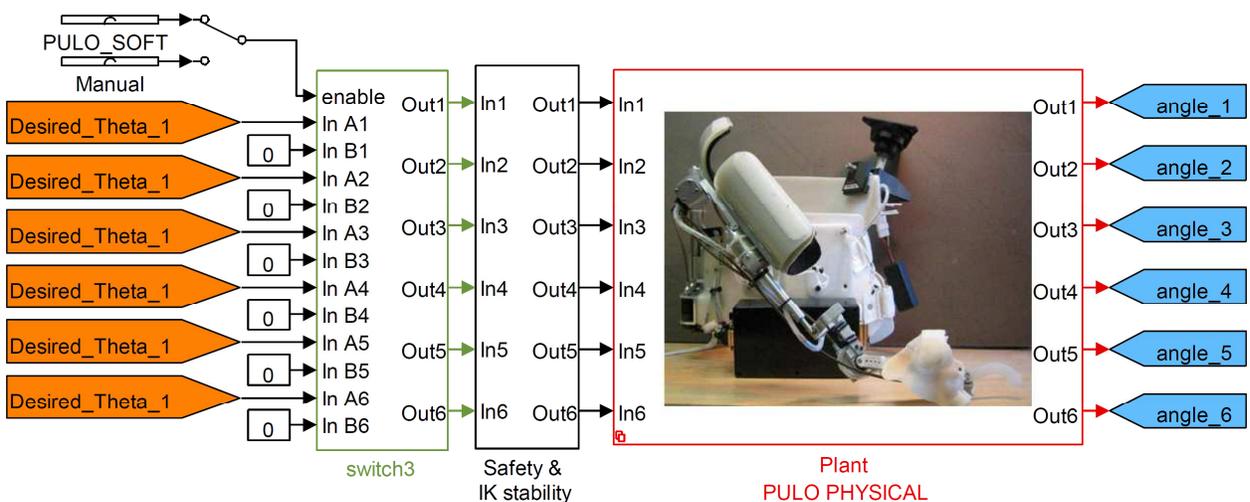


Figure 43. Inverse Kinematics Stability and Physical Plant

Figure 44 shows the simplified subsystem diagram of the *UBC-PULO Physical Plant* block. The joint angle sensor signals are acquired and converted to degrees in the *joint angle acquisition* block. This block also incorporates a joint angle initialization algorithm to define the zero position of each angle sensor during installation. The real-time joint angles are sent to the *Plant* block output (where they are made available to the high level controller) and the *PID Controller* block.

The *PWM_Motor_out* block, also shown in Figure 45-b, drives the UBC-PULO motors by commanding the motor driver PCB board. The inputs units are in terms of PWM duty cycle ranging from -100% to +100%. For example a +100% drives the motor full power forwards, a -50% will drive the motor half power backwards and a 0% input will not move the motor. Thus, the six signals received from the *PID Controller* block directly drive the UBC-PULO motors.

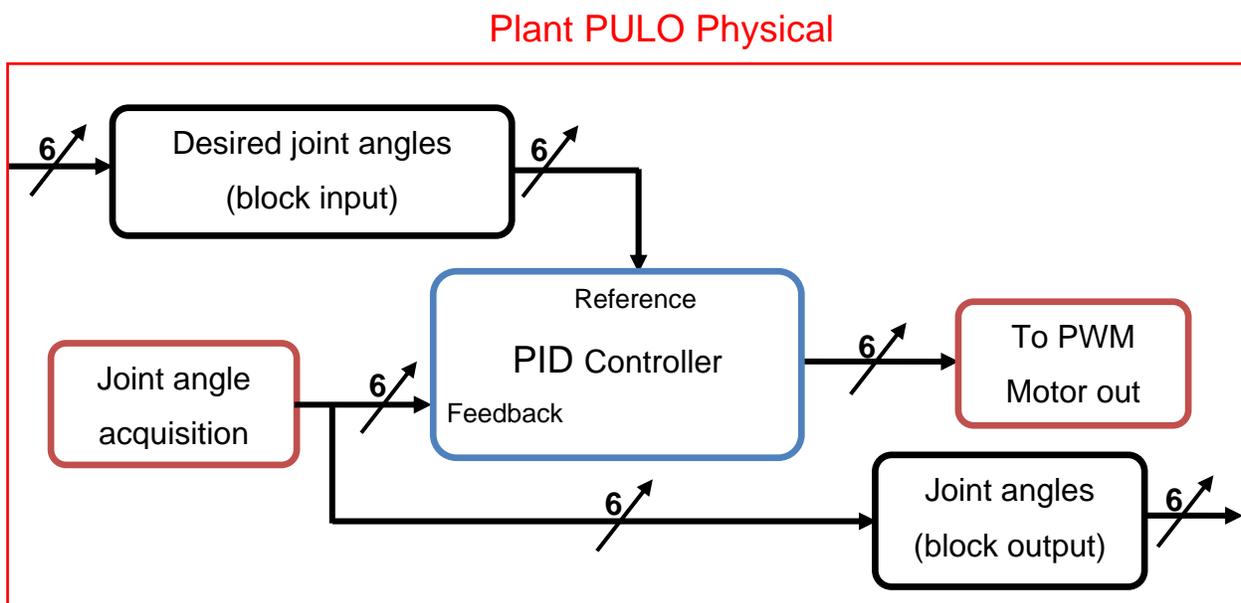


Figure 44. Simplified Physical Plant

The *PID controller* block is composed of six parallel discrete PID loops which minimize the error between the desired joint angles (reference) and current joint angle (feedback) inputs. As shown in Figure 45-a, a saturation block follows each discrete PID controller to limit the output to $\pm 100\%$ (i.e. required by the *PWM_motor_out* block). All six PID loops have adjustable PID gain values which are accessible as a *constant tuning parameter*.

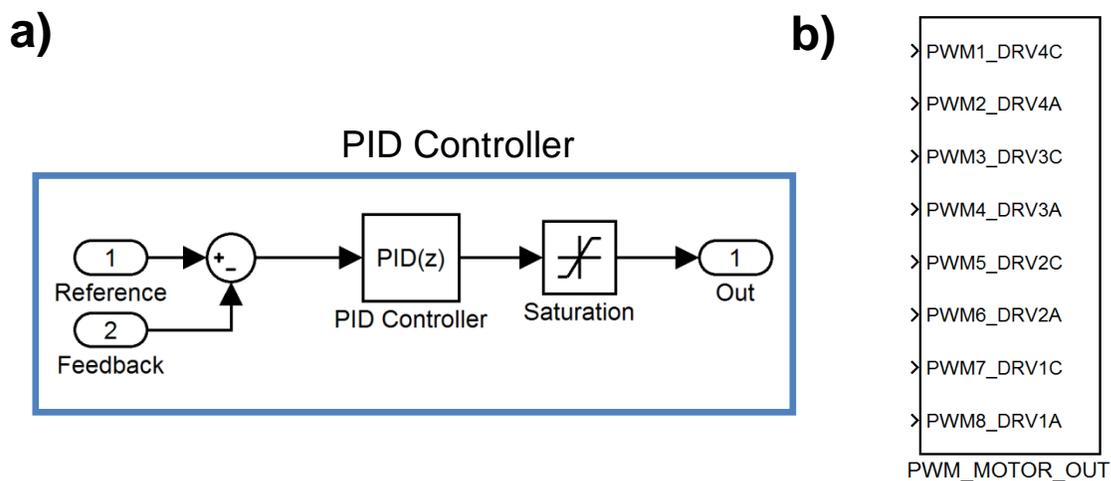


Figure 45. a) Sample Discrete PID Loop And Saturation Block b) The Motor Driver Block Which Commands The Motor Driver PCB

5.2.4 New UBC-PULO software features

5.2.4.1 Current sensing

The *current sense* block, which is located under the *status* block in Figure 39, outputs the real-time motor currents going through each motor. Since the motor currents signals are quite noisy, low pass filters are installed in series with each output to filter

out high frequency noise. Furthermore, the motor current readings can be converted to an estimate motor torque reading. Knowing the gear ratios and lever lengths, joint torques and force estimates can also be calculated (i.e. if friction is low). Post processing of torque, force, position, velocity and acceleration can provide valuable information in regards to over-all dynamic performance, selecting suitable motors and power consumption.

5.2.4.2 Initialization sequence

Once the UBC-PULO is turned on, the following power-up sequence occurs in order:

1. All variables are reset and UBC_PULO enters neutral mode. The motor driver board is enabled with all motor drivers receiving a 0% duty cycle.
2. After 2 seconds the PID block is enabled and the exoskeleton is driven to the initial angle (initial angles are user-definable).
3. After 5 seconds the system is ready and accepts user inputs.

The initialization sequence ensures a safe and stable boot up sequence for the UBC-PULO. If this initialization sequence is not in place, the system will become unstable for the following reasons:

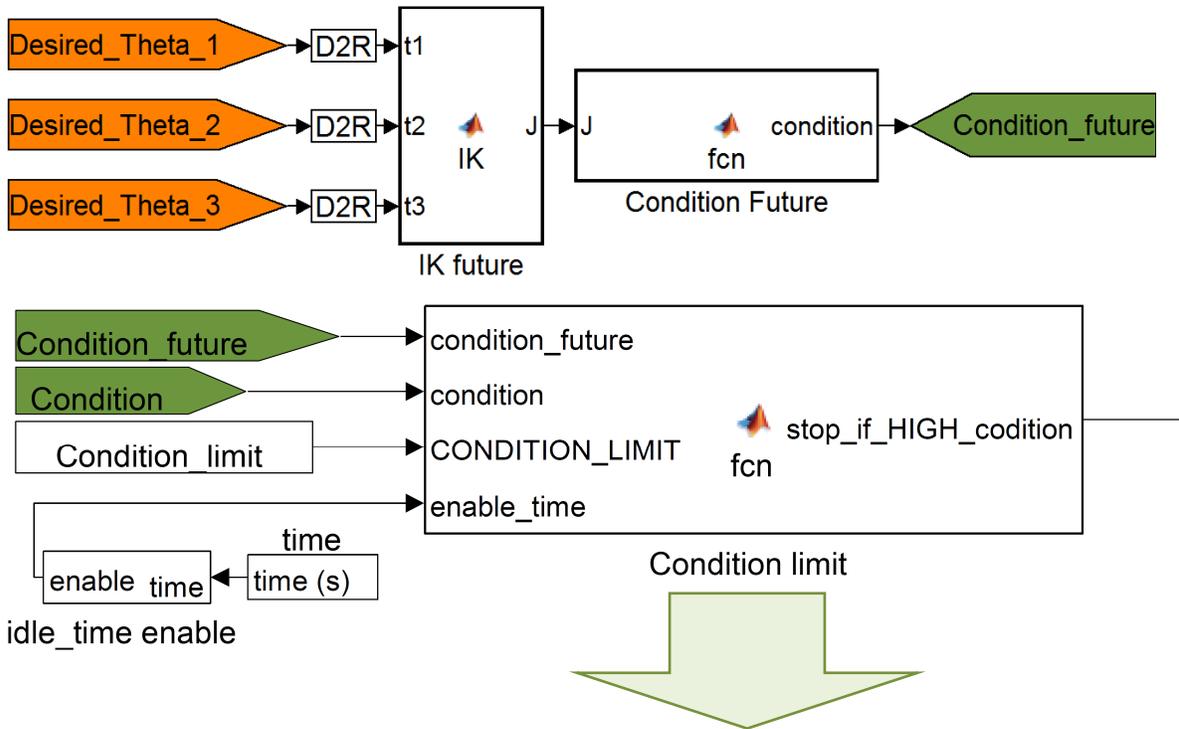
- The initialization sequence avoids a current rush into the electrical system by not driving motors for 2 seconds. This permits sufficient time for the power regulators to stabilize. In the contrary case, the voltage may drop below required levels.
- The PID block is a time sensitive low level controller and needs to be disabled until the motor driver board is online. Since the motors are not controllable during

the initial two seconds, the PID integral term provides a false value to the feedback loop (i.e. The integral of angle position error over two seconds times the Integral constant) leading to system instability.

5.2.4.3 Inverse kinematics stability

The *Safety and IK Stability* block shown in Figure 67, prevents the manipulator from entering a singular or close to singular position. This block is crucial to the overall stability of the device and was deemed necessary after a number of early tests. Early tests demonstrated that the manipulator becomes kinematically unstable (i.e. reaches a singular configuration) when certain positions in the workspace are reached (for example when the elbow joint is close to zero degrees), causing undesirable rapid movements. The *Safety and IK Stability* block is based on calculating the condition number [46]. As demonstrated in Figure 46, the algorithm remains one step ahead of the system by evaluating the future condition number and comparing it with the current condition number. The manipulator is only allowed to move if either the current condition number is below the `CONDITION_LIMIT` (which is pre-definable as a *constant tuning parameter*) or if the current condition is less than the future condition number. This ensures that the condition number is always below the `CONDITION_LIMIT` or if somehow the system has passed the `CONDITION_LIMIT` it can only move toward a lower condition number (i.e. if “stop_if_HIGH_codition” is set to “1”, the manipulator is allowed to move in user-commanded direction; and, if “stop_if_HIGH_codition” is set to “0” it will remain stationary not responding to user commands). The *condition limit* block

is activated via the *idle_time_enable* block after 5 seconds as this is the time when user inputs are first accepted by the control system.



```

if (enable_time ==1)

    if (condition_future <= CONDITION_LIMIT || condition_future < condition)

        stop_if_HIGH_codition = 1; %IK is OK

    else

        stop_if_HIGH_codition =0; %IK is reaching singular position
    end

else
stop_if_HIGH_codition =1; %IK is OK
end

```

Figure 46. Condition Limit Block

5.2.4.4 User adjustable functional mode

The functional operation mode (see Section 2.3.2.2) algorithm has been modified to move between two user-defined start and end end-effector positions. Currently, the functional mode is adjusted to be suitable for eating. In this mode, the mouth and table top position have been previously defined and are adjustable by the user as shown in Figure 47. The user is able to move the end-effector toward their mouth by tilting their head backwards past the neutral zone. Similarly, tilting their head forwards will move the end-effector toward the predefined table position. Additionally, the end-effector can be moved in the negative or positive X axis direction (i.e. left or right) by a left or right head tilt past the neutral zone. Holding the shoulder controller in the MID or HIGH position moves the end-effector perpendicular to the functional plane shown in red.

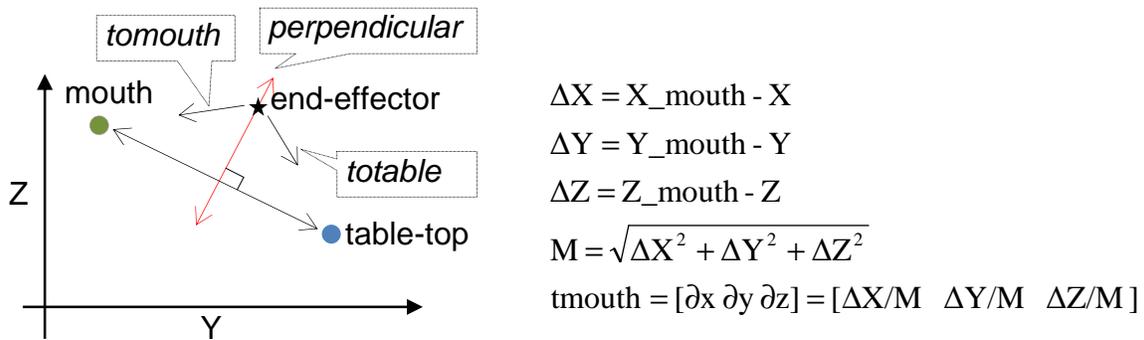


Figure 47. User Adjustable Functional Mode

To produce the desired motion, the *Functional mode* block (see Figure 40) calculates the incremental end-effector values dx_{mode2} , dy_{mode2} and dz_{mode2} and sends them to the *Inverse kinematics* block (see Figure 41). The *Normalized direction vector* block shown below is a subsystem of the *Functional mode* block and is responsible for calculating the $\vec{tomouth}$, $\vec{totable}$ and $\vec{perpendicular}$ vectors. Depending

on the user's intentions, one of these vectors is selected and multiplied by a proportional tilt-angle gain before being assigned to the incremental end-effector values dx_mode2 , dy_mode2 and dz_mode2 .

The *Forward Kinematics* block calculates the end-effector position based on the current joint angles and assigns it to the $FK_position$ global variable. Knowing both the current end-effector position and end-point position, the normalized direction vectors $\vec{tomouth}$, $\vec{totable}$ and $\vec{perpendicular}$ can be calculated by the *Normalized direction vector* block as demonstrated in Figure 48. A sample calculation for the $\vec{tomouth}$ vector is shown in Figure 47.

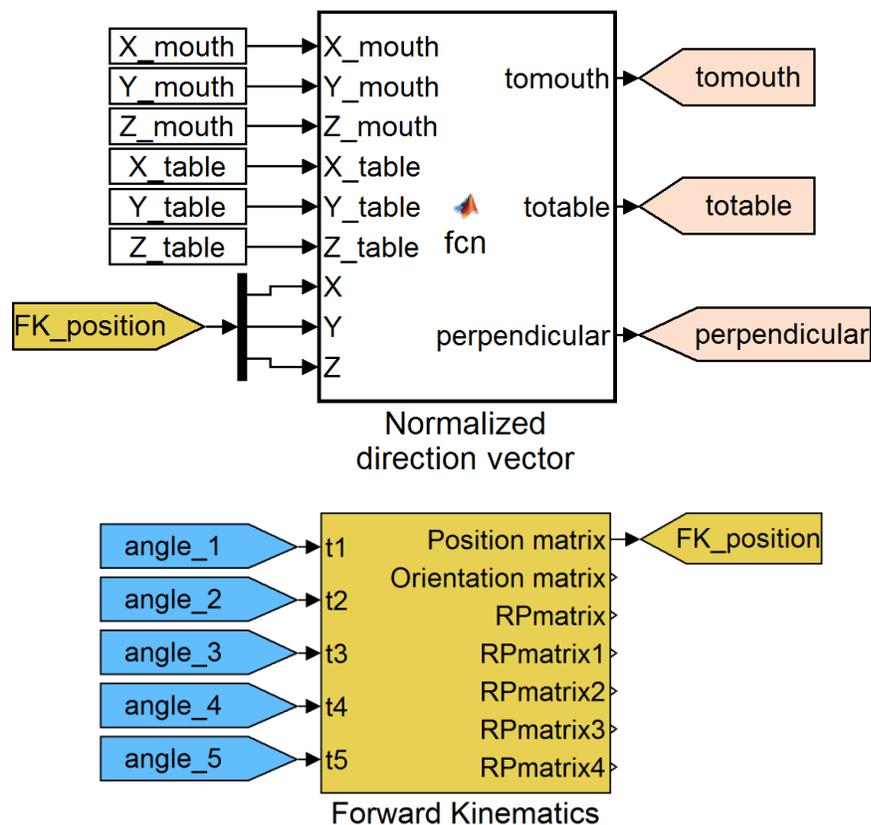


Figure 48. User Adjustable Functional Mode

5.2.4.5 Hardware in the loop testing

During external simulation mode, the host computer simulation interacts with the UBC-PULO embedded hardware (i.e. physical plant) by sending and receiving real-time data – also known as Hardware in the Loop testing. During external simulation, parameters can be tuned and data can be monitor or acquired in real-time. Additionally, real-time data may be graphed or monitored directly on an external monitor using XPC Target scope blocks, since the target computer has its own LVDS LCD interface. This provides a powerful test platform, where the effects of parameter changes can be monitored and evaluated during or after execution. Furthermore, any number of signals or variables can be acquired and stored for post-processing on the host computer.

5.2.4.6 Parametric software design

All system variables of interest (such as PID constants, DH table constants, sample time, initialization angles, initialization times, joint limit constants, the condition limit constant and user preferences) are stored in the “variables.m” M-file allowing quick access to all the main system parameters.

5.3 Software for experimental setup

One of the powerful tools available through the new UBC-PULO software is having multiple *plant* block options. As shown in Figure 49, the *plant* block can be switched to a model representation of the UBC-PULO prototype, allowing for real-time simulation and

virtual visualization of the system. During simulation, parameters can be changed (manually or parametrically) and data can be viewed or logged (for post processing) as needed. This serves as a testing, development and model validation tool. The *PULO Physical plant* drives the physical prototype device as mentioned in Section 5.2.3 and Figure 43. The plant block can also be switched to an ideal plant block to represent the ideal case, where the plant is able to instantly follow any commands – i.e. the desired input joint angle will immediately become the output joint angle. Thus, the setup presented on Figure 21 can be satisfied by performing a real-time execution of three independent instances of the UBC-PULO software (i.e. one using the simulation plant block, one using the ideal model plant block and the other using the physical plant block) in parallel using the new UBC-PULO electrical system.

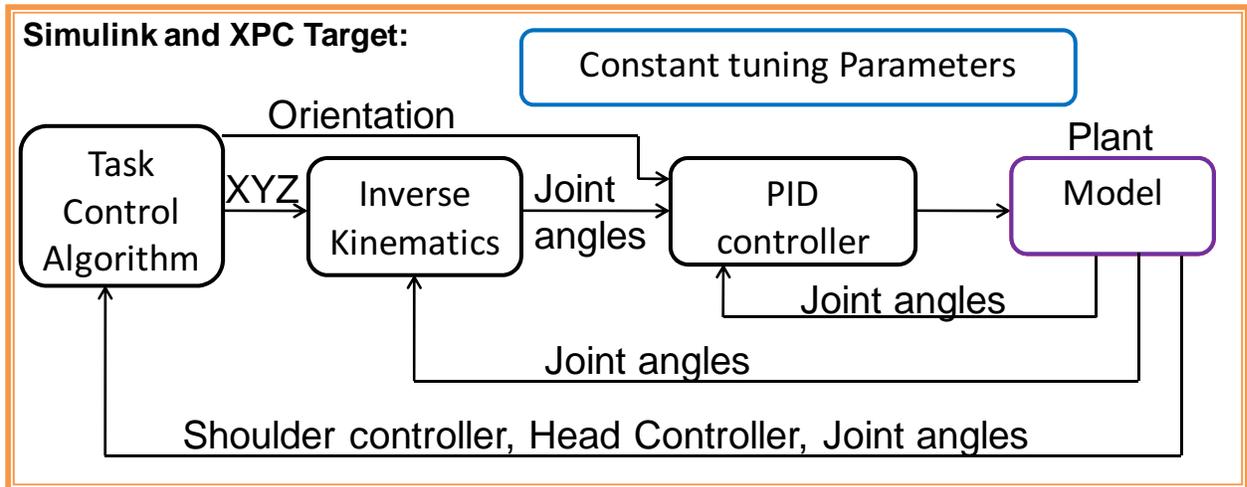
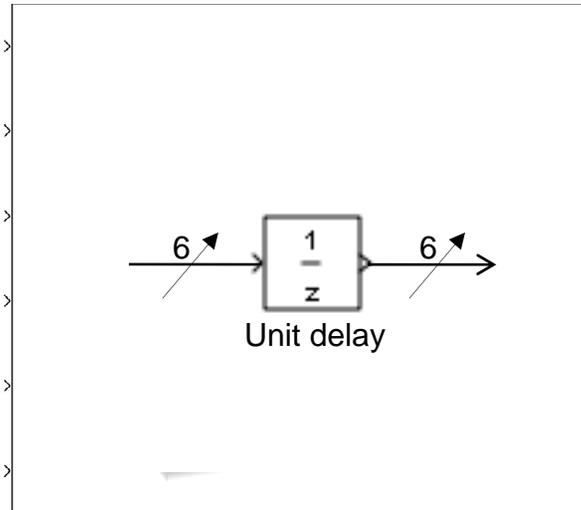


Figure 49. Plant Simulation Overview

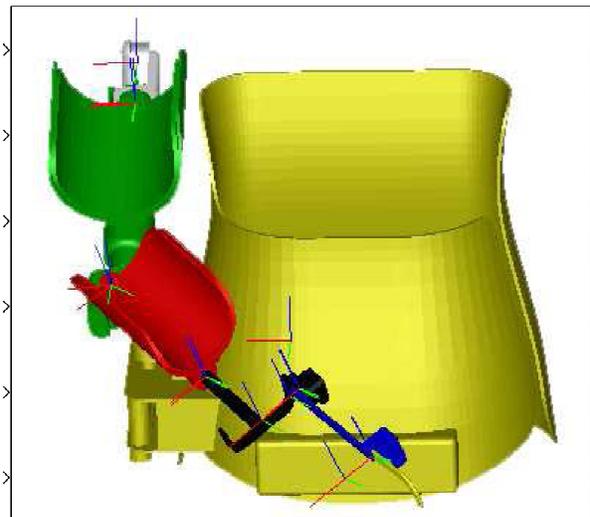
The available *plant* block choices are presented in Figure 50, where they can be swapped by right clicking on the *plant* block and choosing the desired “block choice”.



Ideal drive System



PULO PHYSICAL



Model of drive system and exoskeleton

Figure 50. UBC- PULO Plant Choices

The “*ideal drive system*” *plant* block represents the ideal case, where the UBC-PULO responds to the desired input joint angle with one unit delay (i.e. one sample time), and immediately jumps to the desired position. Although this block utilizes the same SimMechanics model as the “*model of drive system and exoskeleton*” *plant* block presented in Figure 51, the SimMechanics joint actuator is set to directly position the joints to the block’s input angle. Thus, the “*ideal drive system*” *plant* block only facilitates

a visual representation of UBC-PULO with the input and output angles of this block being always the same. This model of the UBC-PULO is useful for simulating, visualizing, testing and analyzing the ideal behaviour of the system with respect to the control inputs which is discussed in Chapter 6.

The “*model of drive system and exoskeleton*” *plant* block provides contains an upgraded VRSE representation of the UBC-PULO. To reflect a more accurate representation of the current UBC-PULO, the previous model was upgraded in this research work and an accurate representation of the UBC-PULO exoskeleton assembly was prepared in SolidWorks. Then, this SolidWorks assembly was converted into a SimMechanics model and imported into Simulink as explained in Section 2.3. The resulting six DOF (i.e. five DOF plus grasp) 3D UBC-PULO visualization tool is presented in Figure 51. The “*model of drive system and exoskeleton*” *plant* block consists of six parallel discrete PID loops, a model of the drive system and a SimMechanics representation of UBC-PULO in order to closely simulate the behaviour of the physical device. This *plant* block option would ideally produce the same dynamic response as the physical prototype once all the parameters are tuned, making it a tool for:

1. Observing the dynamic behaviour of the system
2. Acquiring and analyzing data
3. Optimizing the control system
4. Validating the UBC-PULO Virtual Reality Simulator [1]
5. Performing tests on future designs and prototypes before physical production

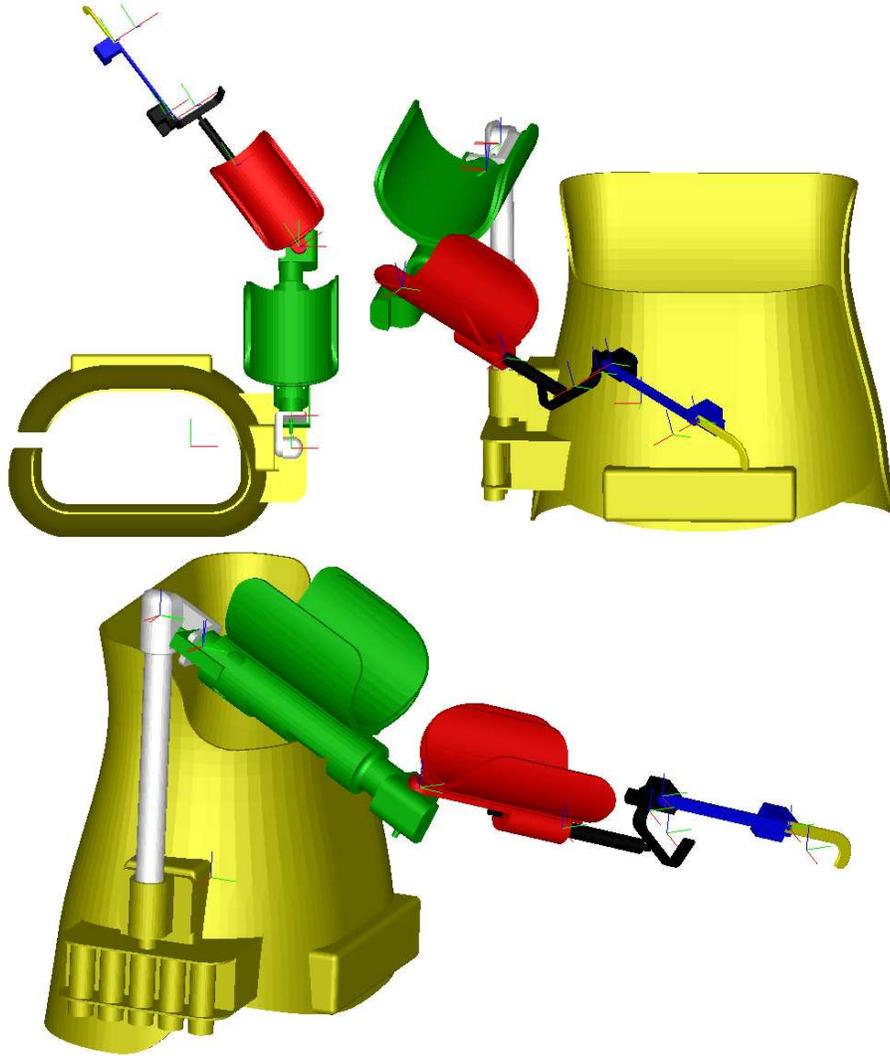


Figure 51. UBC-PULO Visual Representation

5.4 Summary

This chapter provided details related to the new UBC-PULO software and its new capabilities. It was only through powerful features such as the modeling and visualization; HIL testing; and, time-efficient Simulink, MATLAB and XPC Target programming that the commissioning of the UBC-PULO was made possible in the

available time-frame. In addition, the experimental setup's software was developed, which allows for validation of the VRSE. The next chapter demonstrates results on the validation of VRSE.

Chapter 6 – Validation of the VRSE

6.1 Introduction

This chapter examines the performance of the commissioned UBC-PULO prototype against that of the VRSE by performing two sets of experiments using the experiment setup running in real-time on the UBC-PULO electrical hardware as explained in Section 5.3. As illustrated in Figure 21, the experimental setup allows for the comparison of the VRSE simulation model, the physical prototype device and ideal plant model outputs for the identical input conditions.

The first experiment tests the closed loop joint response for each individual joint for the same input step response. The results of this experiment will demonstrate how closely the closed loop joint responses of the simulation model and physical prototype are matched (i.e. the closed loop drive system and joint can be validated).

Before performing the second experiment, the UBC-PULO physical device needed to be tested to demonstrate full functionality with respect to the original control algorithm discussed in Section 2.3, i.e. performance of for the overall control algorithm with respect to user safety, the inverse kinematics stability and the correct interpretation of user inputs received via the user control interfaces.

The second experiment tests the overall performance of the manipulator with respect to the user performing various pre-defined tasks over the full range of motion. In this experiment the three parallel systems in the experimental setup (see Figure 21) receive the same user input and thus the output results allow for:

1. Comparison / validation of joint position data when all joints are actuated
(i.e. validation of all joints working at the same time)
2. Comparison / validation of the end-effector positions
(i.e. validation of overall closed loop drive system coupled with the mechanical properties of the manipulator's links)

6.2 Test results

6.2.1 Experiment 1- PID response

During initial testing, the PID constants were tuned for each joint on the physical UBC-PULO prototype to optimize between rise time, overshoot and settling time. At first a Ziegler-Nichols [45] tuning method was implemented; however, this resulted in excessive overshoot and a long settling time. The final PID constants were tuned by performing a parametric sweep of the PID constants using HIL testing (See Section 5.2.4.5 for more details on HIL testing). Knowing the trade-offs presented in Table 11, the step response graphs were visually analysed and the most suitable PID constants selected. A similar procedure was repeated for tuning the *“model of drive system and exoskeleton”* plant block constants (See Section 5.3) such that the model step response closely agreed with the physical device.

Table 11. PID Constant Trade-off

PID Constant	Rise Time	Overshoot	Settling Time	Steady State Error
K_p	Decrease	Increase	Small Change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_d	Small Change	Decrease	Decrease	Small Change

Figure 52 presents a sample azimuth joint response of the ideal model block, the simulation model (i.e. model of drive system and exoskeleton) and physical device for a 30 degree step input. As demonstrated in Table 12, the physical device and the simulation model have reasonably close responses (i.e. a settling time difference of 0.22 seconds, overshoot difference of 35% and rise time difference of 0.06 seconds for a 30 degree step input).

Table 12. Sample Azimuth Step Response

	Rise Time	Overshoot	Settling Time	Steady State Error
Ideal model	0	0	0	0
Simulation model	0.31	5.4%	0.93	0.21
Physical PULO	0.25	8.3%	0.71	0.45

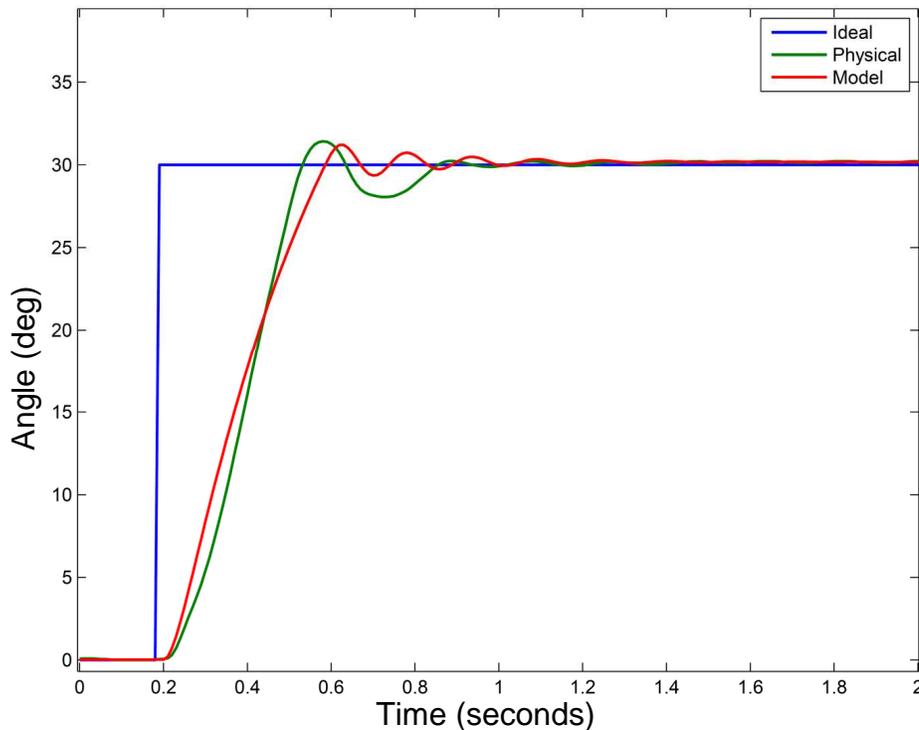


Figure 52. Azimuth Step Response

Figure 53 illustrates the results on a sample 30 degree step response for all the joints. The step responses for all the joints are similar to the results shown for the azimuth, thus allowing the system to reach the steady state position in about one second. Additionally, the simulation model's joint step responses are reasonably close to that of the physical device for all six joints, with rise times, settling times, overshoots and steady state errors similar to values shown in Table 12.

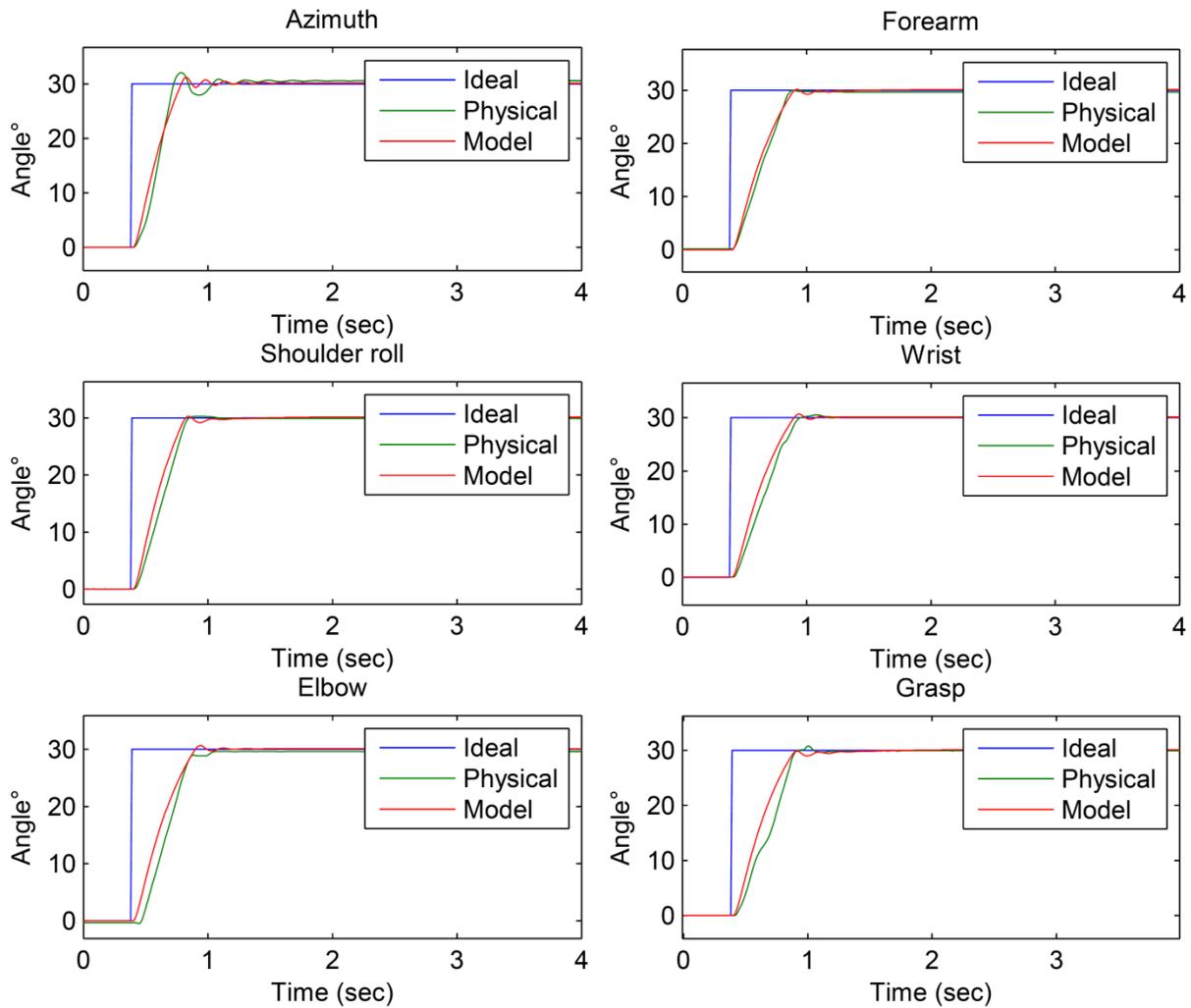


Figure 53. All Joint Step Responses

6.2.2 Experiment 2- performing tasks

6.2.2.1 Testing the full functionality of the UBC-PULO control algorithm

6.2.2.1.1 User interface and task control algorithm input signals

This section presents test results with regards to the user interfaces and task control algorithm input signals. Using the shoulder interface, the user can switch between the UBC-PULO modes of operation by performing a shoulder shrug as shown in Figure 54.

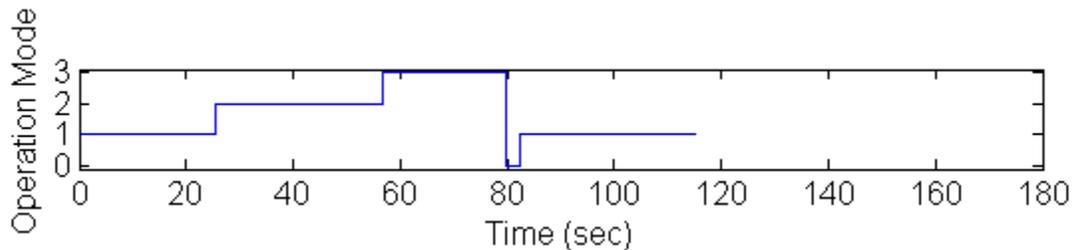


Figure 54. Switching Operation Modes Using Shoulder Interface

As presented in Figure 55, extended shoulder shrugs or head tilts extending past the neutral zone boundary produce the desired *normalized user input* (See Section 2.3.2.2 for more details). The *normalized user input* signal is sent to the *task control algorithm* block (See Section 5.2.1), which depending on the mode of operation and overall system status, produces the *incremental inverse kinematics vector* (See Section 5.2.2) thus driving the end-effector in the desired direction. In this demonstration, the neutral zone boundary was defined to be at 15 degrees and the outer limit of the operation zone was defined as 45 degrees (i.e. $a=15$ and $b=45$ in Equation 1 and Equation 2). Shoulder shrugs longer than 65 ms were defined as extended shoulder

shrug, producing a logic “1” for the MID and /or HIGH position as explained in Section 5.2. Thus, if the *neutral zone* logic state is “1”, *normalized user inputs* will produce a non-zero incremental signal to the Inverse Kinematics, which in turn produces the desired end-effector motion. In this demonstration, the head interface’s X-axis position, Y-axis position and extended shoulder shrugs were activated separately for clarity.

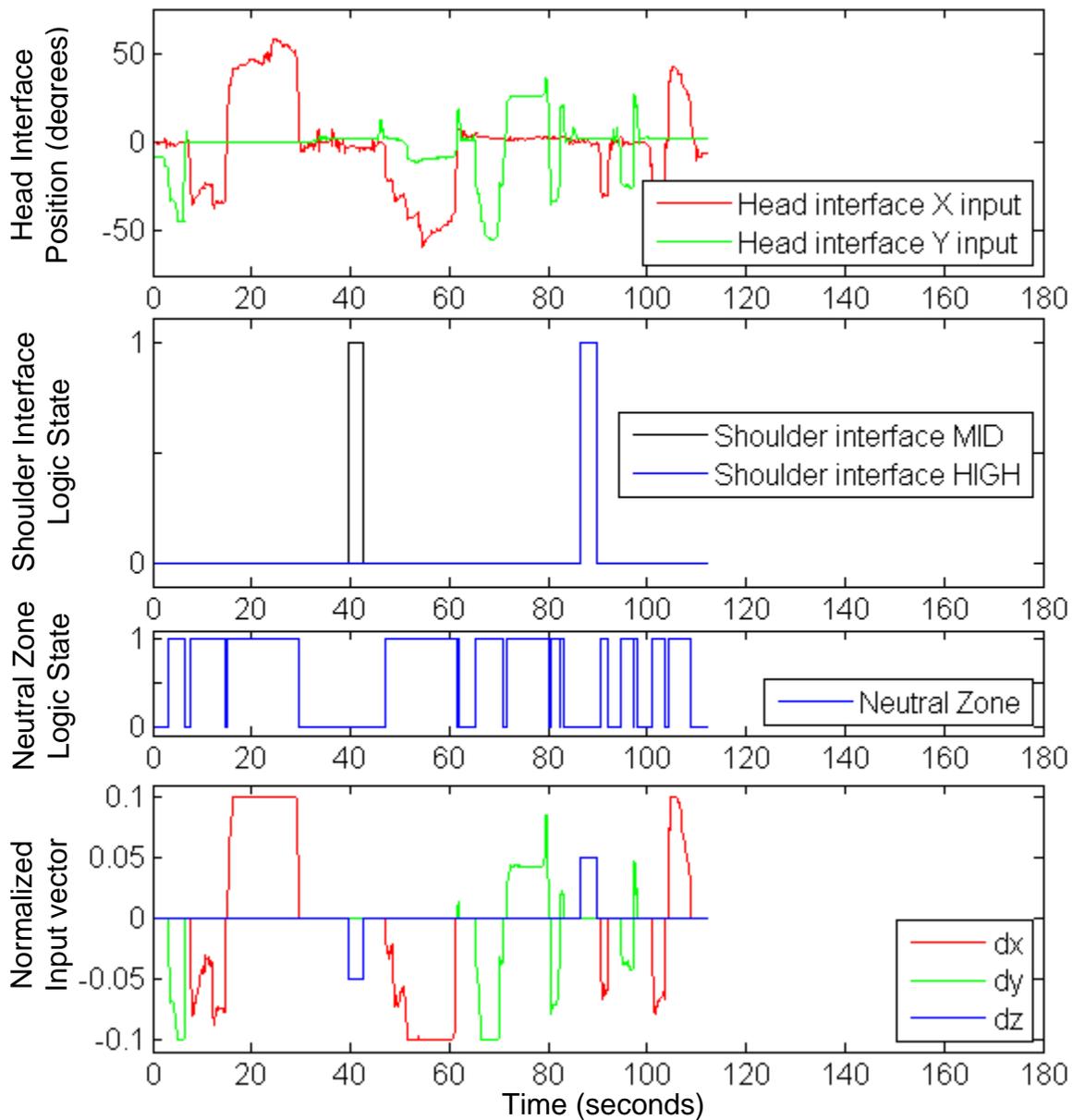


Figure 55. User Control Interfaces vs. System Control Signals

6.2.2.1.2 User safety and inverse kinematics stability

To test how well the manipulator performs to keep within the predefined joint limit, it was rapidly manoeuvred around during all operation modes. Table 13 presents the maximum and minimum recorded joint angles with respect to the allowable limits.

Table 13. Joint Angle Limits Test

Joint	Allowed lower limit (degrees)	Allowed upper limit (degrees)	Measured minimum angle (degrees)	Measured maximum angle (degrees)
Azimuth	-90	40	-90.52	41.54
Shoulder roll	-70	70	-70.24	70.32
Elbow	15	105	26.36	106.26
Forearm	-40	30	-40.31	30.11
Wrist	-45	45	-45.12	45.08
Grasp	-70	70	-70.07	70.03

In addition, tests confirm that excessively fast head interface inputs (i.e. user sneezing) do not produce any end-effector motion, meeting the original design requirements mentioned in Section 2.3.

Figure 56 presents the condition number, azimuth, shoulder and elbow joint angles during table top mode and functional mode. The manipulator starts at a condition number of 8.2 and is only allowed to move in a decreasing direction, as expected – i.e. only user input that reduces the condition number can produce end-effector motion.

The condition number remains below the predefined threshold value of 6 for the remainder of the experiment, ensuring overall inverse kinematics stability. Whenever the condition number reaches the threshold value, all joints freeze until an acceptable user input is commanded- i.e. user commands that reduces the condition number.

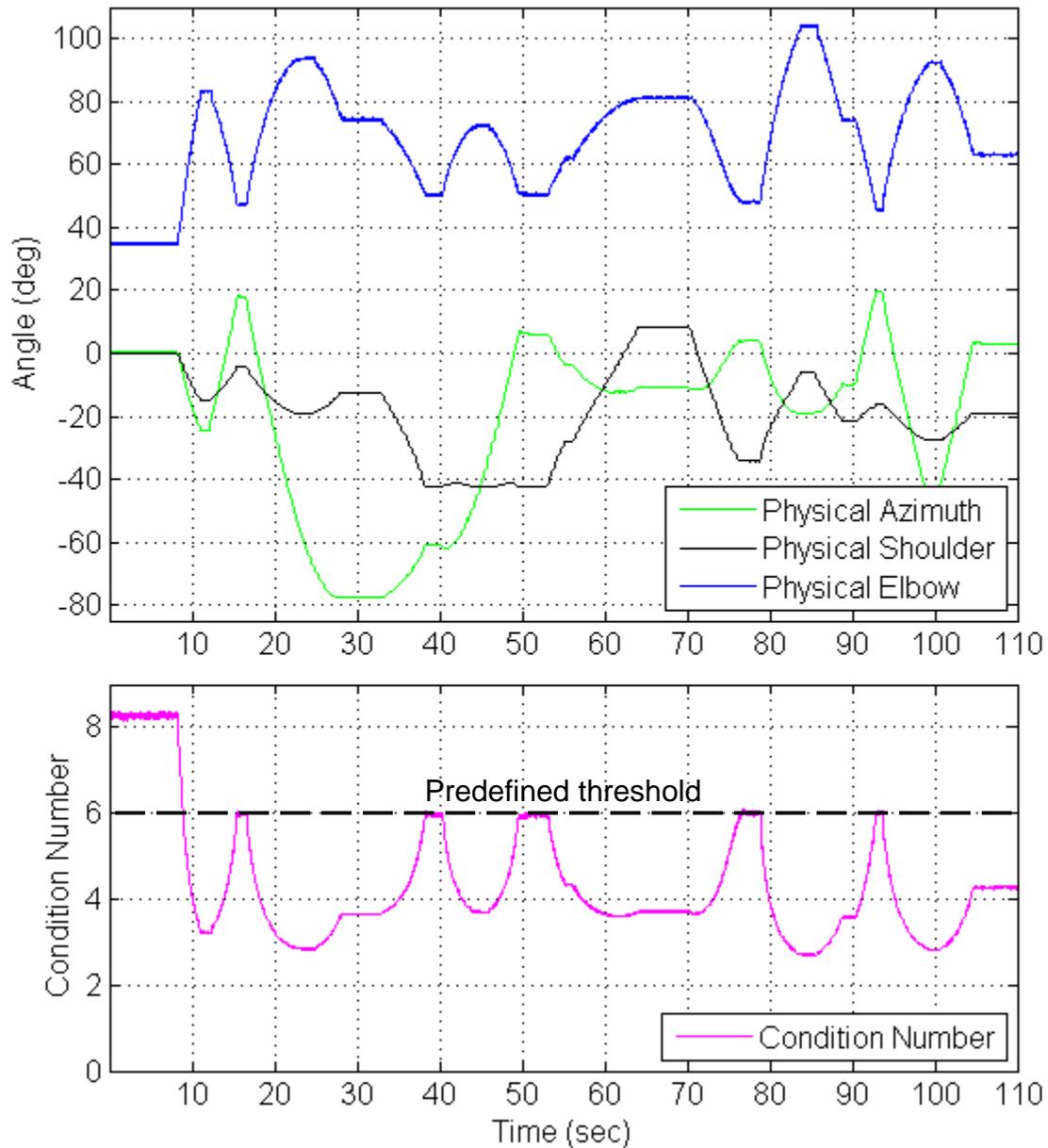


Figure 56. Physical Device Joint Angles vs. Condition Number

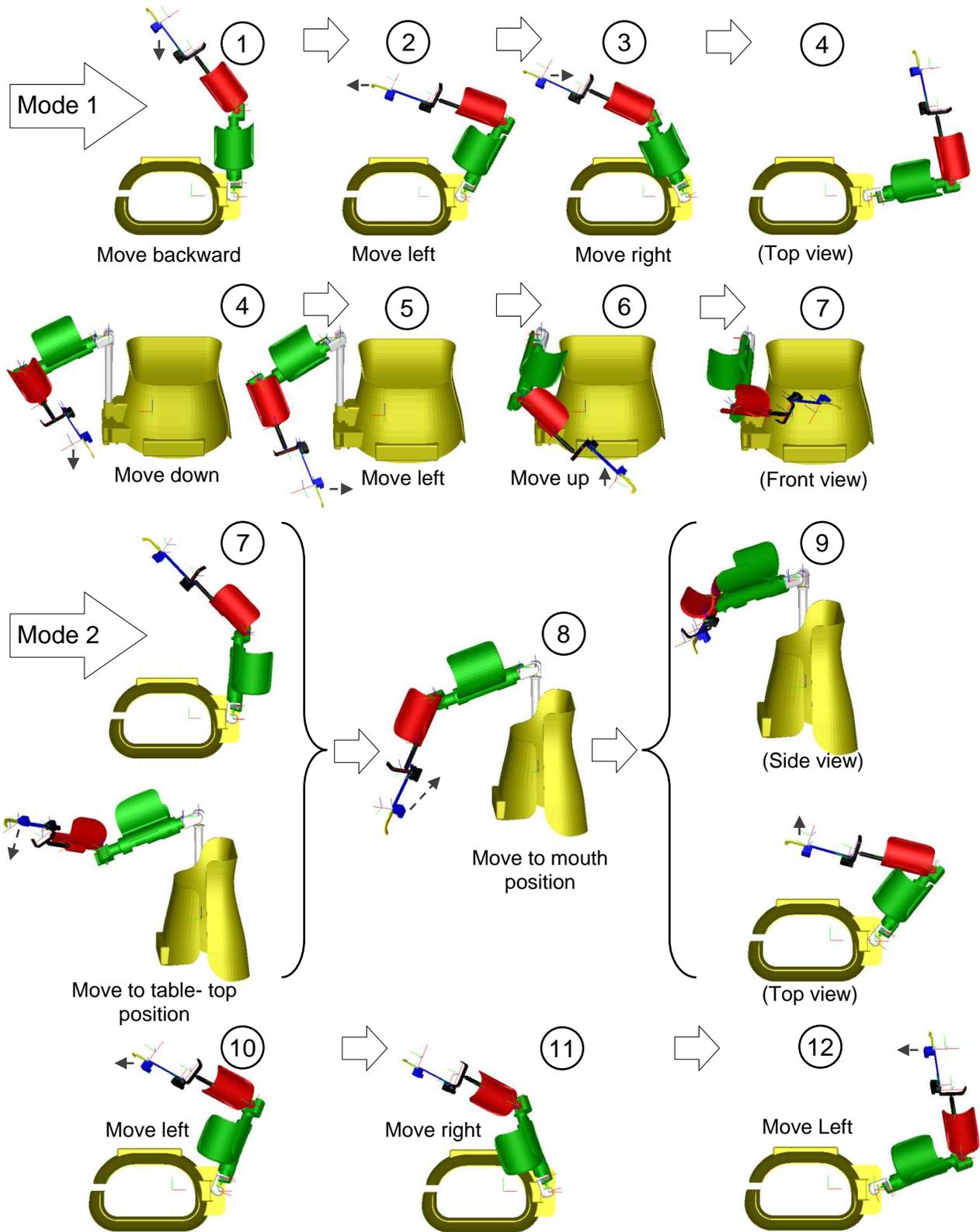
6.2.2.2 Performing tasks using experimental setup

6.2.2.2.1 Position mode of operation (Table Top and Functional mode)

This section presents the results obtained during testing of the *table-top* and *functional* operational modes of operation. The objective was to test the functionality of

the control algorithm and the overall performance of the manipulator over the full range of motion (i.e. as far as the joint limits and inverse kinematics safety algorithms would allow). To achieve the objective, the following set of tasks were completed (See Figure 57):

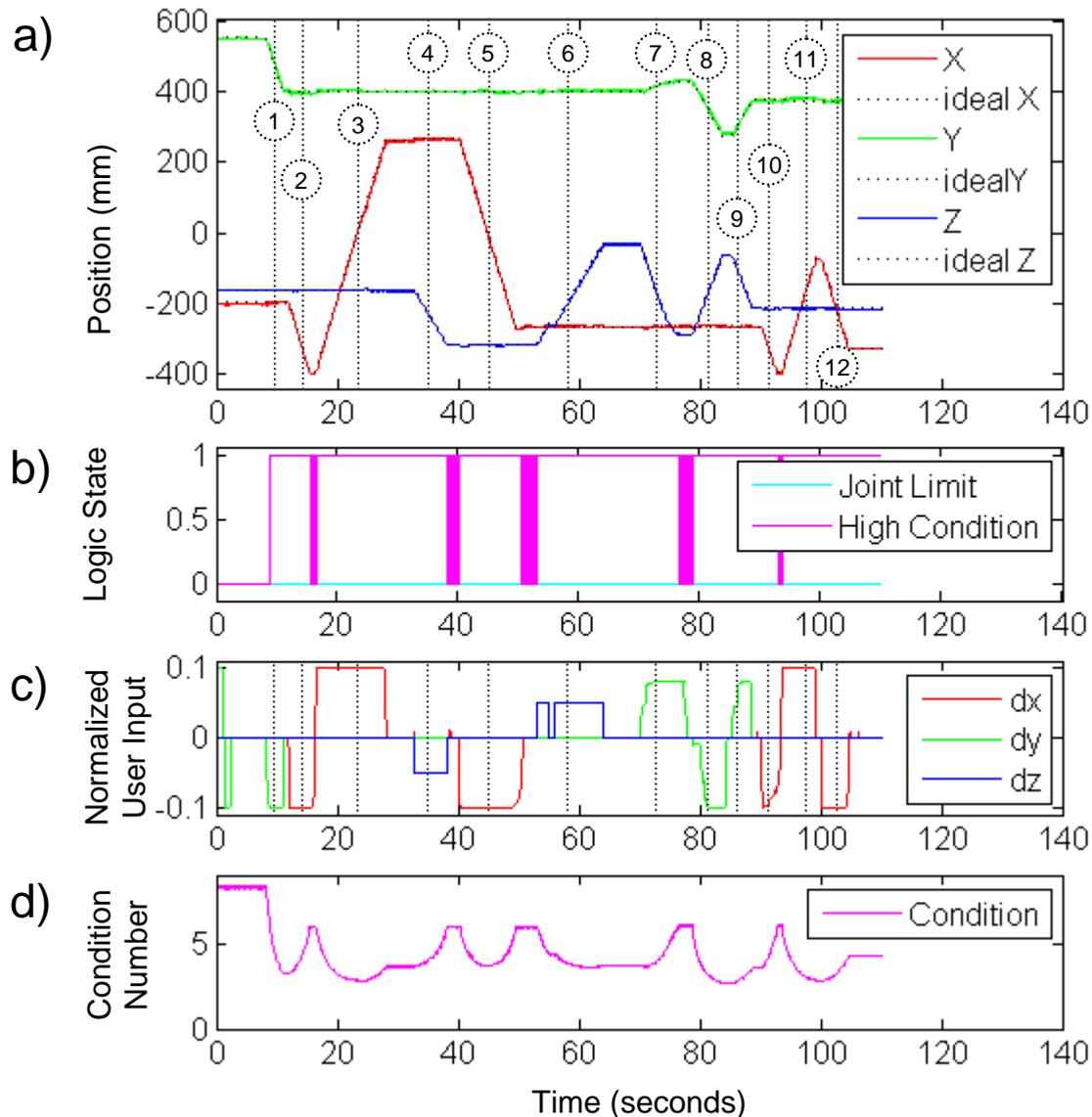
1. Start in *neutral mode*. Switch to *table-top* mode. Tilt head backwards to move end-effector backwards.
2. Tilt head to the left pass the neutral zone and move the end-effector as far as possible to the left.
3. Tilt head to the right pass the neutral zone and move the end-effector to a far right position.
4. Perform an extended MID position shoulder shrug to move the end-effector down the z-axis.
5. Tilt head to the left pass the neutral zone and move the end-effector to the left until the end-effector reaches the centre position.
6. Perform an extended HIGH position shoulder shrug to move end-effector up.
7. Switch to mode 2. Tilt head forwards pass the neutral zone until the end-effector reaches table-top position.
8. Tilt head backward pass the neutral zone until the end-effector reaches mouth position.
9. Tilt head forwards pass the neutral zone until the end-effector reaches the centre position.
10. Tilt head to the left pass the neutral zone and move the end-effector as far as possible to the left.
11. Tilt head to the right pass the neutral zone and move the end-effector to a far right position.
12. Tilt head to the left pass the neutral zone and move the end-effector the centre position.



**Figure 57. Test Tasks for Position Mode of Operation
(Mode 1 = Table Top Mode, Mode 2 = Functional Mode)**

Figure 58 shows the physical prototype and ideal *end-effector position* during compilation of all the listed objectives against the *normalized user input*. If the safety and inverse kinematic stability conditions are satisfied, the user input will drive the end-effector in the desired direction. Figure 58-b shows the controller's internal *joint limit* and *high condition* logic signals. If all joints are within the predefined limits and the condition number is below a predefined value (in this case a value of "6"), the robot will operate under a normal condition with "joint limit = 0" and "high condition = 1". If the manipulator reaches one of the joint limits, the end-effector will freeze (i.e. "joint limit = 1") and only a user command driving the manipulator back to the allowed joint range would be accepted. Similarly, in case of a high condition (i.e. "high condition = 0") the end-effector will freeze and only a user command resulting in a decrease in the condition number can produce end-effector motion until the condition number reaches the predefined allowed limit. Figure 58 shows that the overall UBC-PULO control system worked as expected producing end-effector motion that closely matched the ideal expectation. During Task 1, the end-effector doesn't move when the device is in neutral mode even though user inputs are non-zero. Once the device is switched to table-top mode a backward head tilt past the neutral zone (i.e. negative d_y) produces a decrease in the end-effector Y-axis position. Note that the manipulator started at a condition number of 8.2 (i.e. "high condition = 0") but since the user input caused the manipulator to move towards a decreasing condition number, the manipulator was allowed to move. The condition number remains below the predefined value of 6 throughout the remainder of the tasks as expected –ensuring inverse kinematics stability. During table top mode (Tasks 1 through 6), a side-to-side head tilt past the neutral zone produces

desired d_x directly manipulating the end-effector X-axis position while the Y-axis and Z-axis positions remain constant. Similarly, a front-to-back head tilt past the neutral zone (d_y) affects only the end-effector Y-axis position and an extended shoulder shrug (d_z) only affects the end-effector Z-axis.



**Figure 58. a) End-effector Position vs. Time
b) Joint Limit and High Condition Logic Signals vs. Time
c) Inverse Kinematics Incremental Input Vector vs. Time
d) Condition Number vs. Time**

During Task 7, a forward head tilt past the neutral zone decreases the end-effectors Z-axis position while simultaneously increasing the Y-axis position. Note that, the Y-axis motion is limited during task 7 because the condition limit is reached (i.e. the manipulator gets to a near-singular position when the elbow is close to being straight). In functional mode (i.e. Tasks 7 through 12), a side-to-side head tilt affects only the X-axis end-effector position. In contrast, a back-to-front head tilt affects both the Y-axis and Z-axis end-effector position in order to move the manipulator to the predefined mouth position or table top position desired (see Section 5.2.4.4).

Table 14 presents the measured X-axis, Y-axis and Z-axis end-effector position error as recorded during the testing. The measurements suggests that the mean X-axis, Y-axis and Z-axis position absolute error of the physical device is within 7.34 mm, 6.05 mm and 4.59 mm respectively (95% confidence interval).

Table 14. End-effector Position Error Throughout Tasks 1 to 12

Axis	Maximum absolute error(mm)	Mean absolute error (mm)	Standard deviation (mm)
X (physical)	14.10	2.54	2.40
Y (physical)	13.97	2.29	1.88
Z (physical)	8.97	1.77	1.53

Figure 59 illustrates both the physical end-effector path and the ideal end-effector path as viewed from the top, rear and left side while performing table top mode tasks (i.e. Tasks 1 through 6). This shows that the prototype is able to closely follow the desired path to perform table top tasks as per the original design specifications. Table 15 presents the end-effector error statistics during completion of the table top mode tasks.

Similarly, Figure 60 shows the physical end-effector path and ideal end-effector path while operating in functional mode (Tasks 7 through 12). The results demonstrate that the prototype is able to closely follow the desired path to perform desired functional tasks as per the original device design specifications. Table 16 presents the end-effector error statistics during completion of the functional top mode tasks

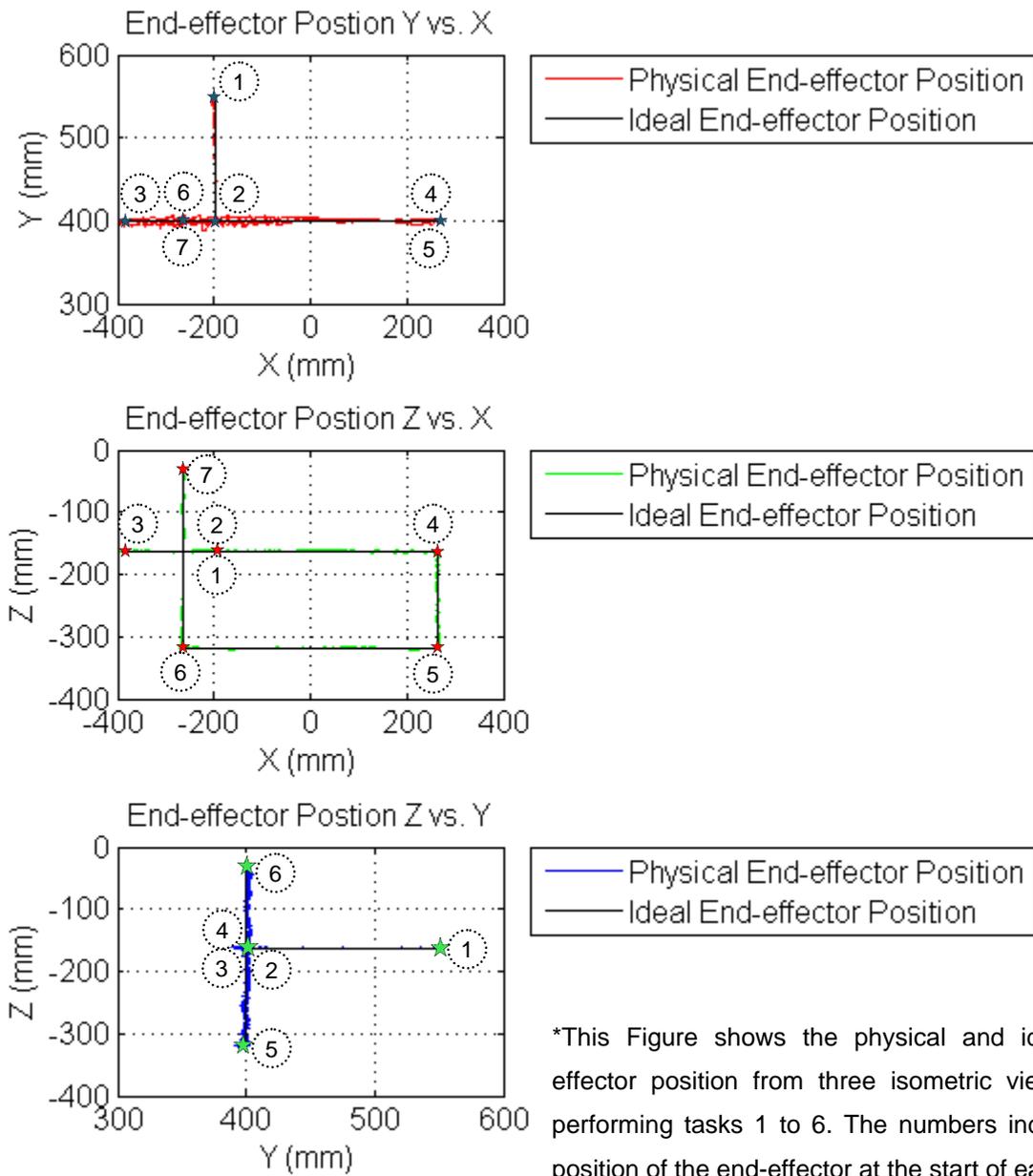


Figure 59. Table Top Mode End-Effector Position (Tasks 1 to 6)

Table 15. Table Top Mode End-effector Position Error (Tasks 1 to 6)

Axis	Maximum absolute error (mm)	Mean absolute error (mm)	Standard deviation (mm)
X (physical)	14.09	3.10	2.65
Y (physical)	12.46	2.01	1.58
Z (physical)	6.27	1.52	1.39

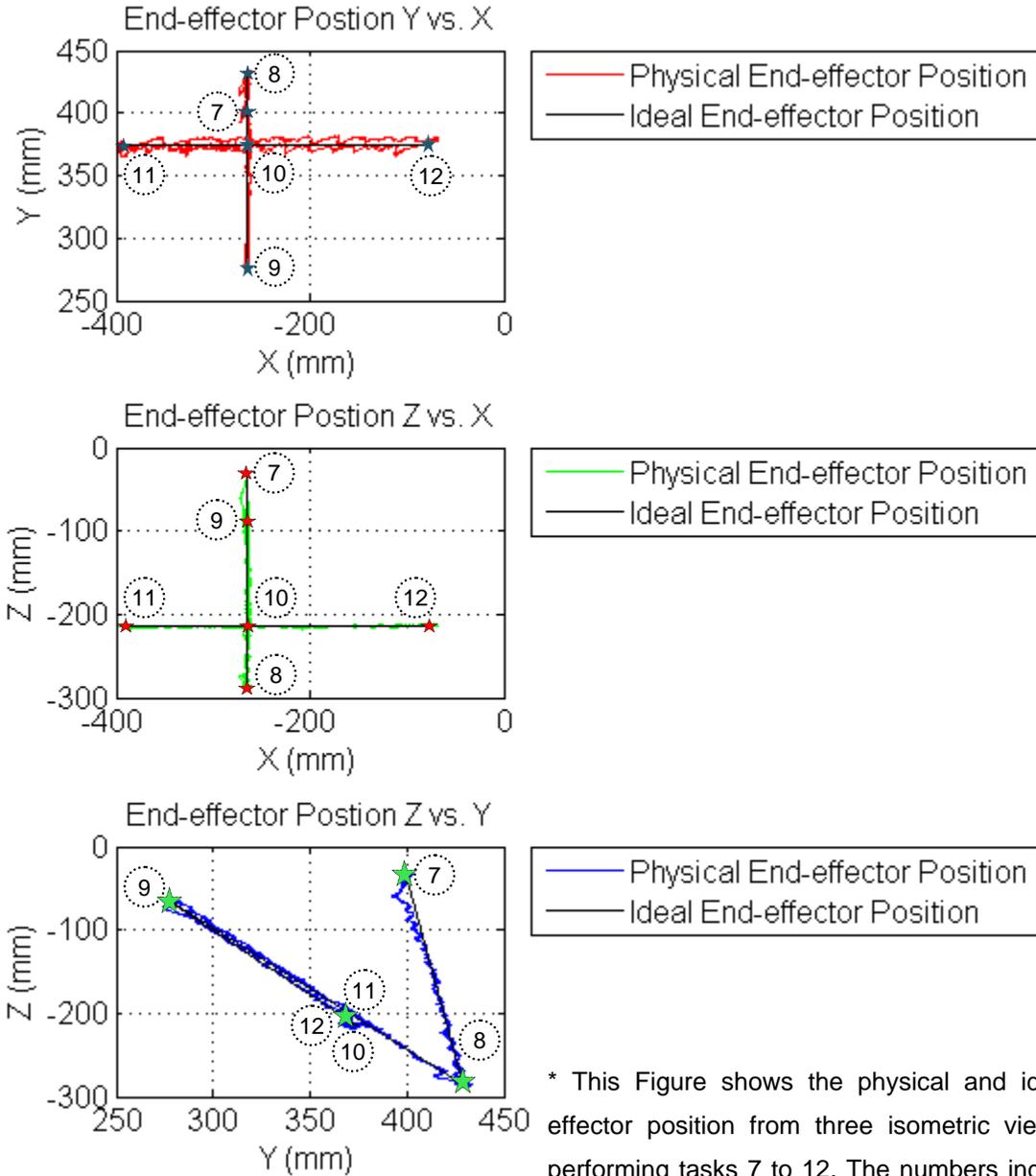


Figure 60. Functional Mode End-effector Position (Tasks 7 to 12)

Table 16. Functional Mode End-effector Position Error (Tasks 7 to 12)

Axis	Maximum absolute error (mm)	Mean absolute error(mm)	Standard deviation (mm)
X (physical)	11.62	1.71	1.66
Y (physical)	13.97	2.71	2.19
Z (physical)	8.97	2.13	1.65

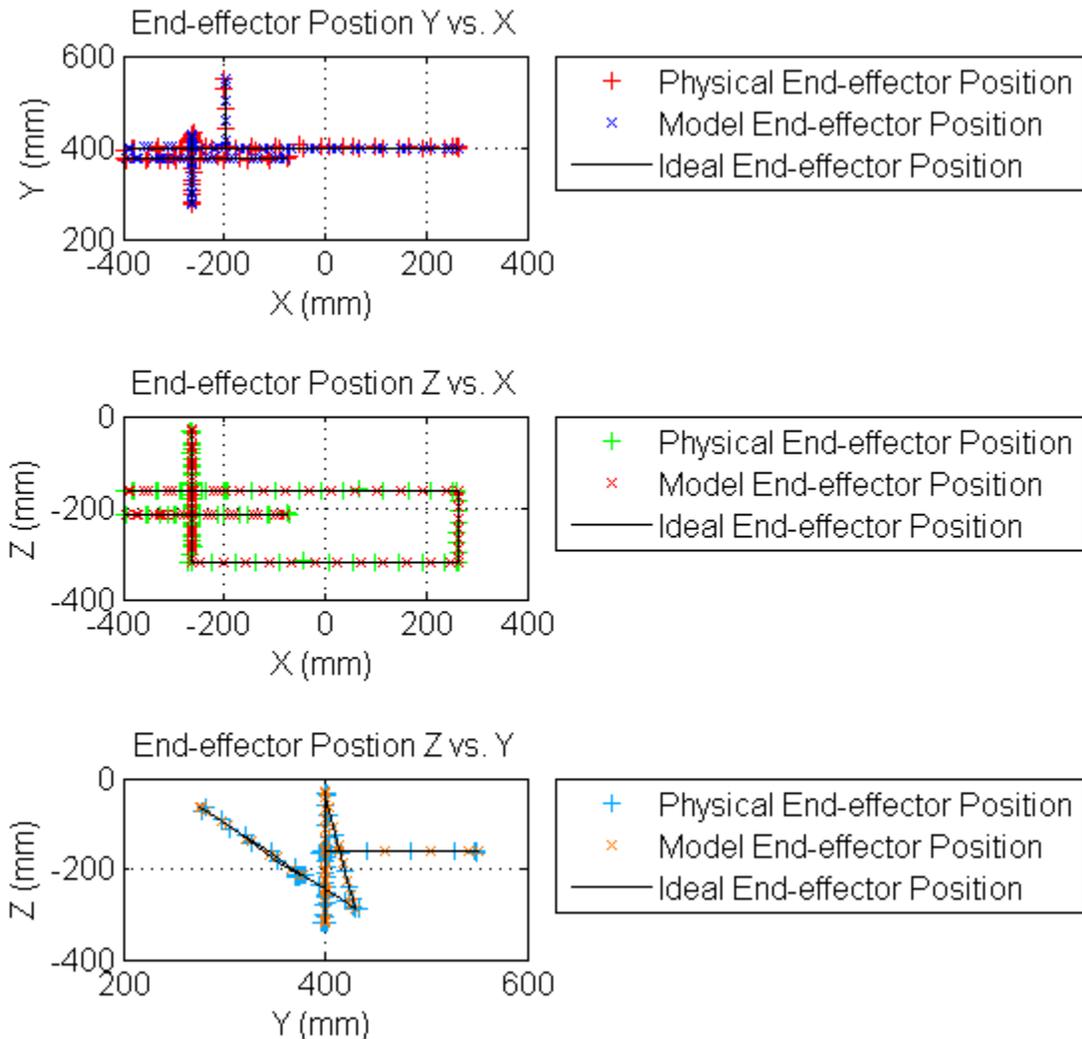


Figure 61. Model vs. Physical End-effector Position (Tasks 1 to 12)

Table 17. Model End-effector Position Error Throughout Tasks 1 to 12

Axis	Maximum absolute error (mm)	Mean absolute error (mm)	Standard deviation (mm)
X (Model)	22.87	7.60	10.21
Y (Model)	23.34	1.74	4.34
Z (Model)	21.29	3.94	6.42

Figure 61 compares the simulation model, ideal model and the physical prototype end-effector path results during competition of Tasks 1 through 12 (i.e. for the same exact user inputs). The end-effector paths are a close match, with Table 17 documenting the model's end-effector position error statistics which are close to the values presented in Table 14 with the upper bound maximum error and upper bound mean error being 37.44 mm and 10.1 mm, respectively, for the X, Y or Z axis.

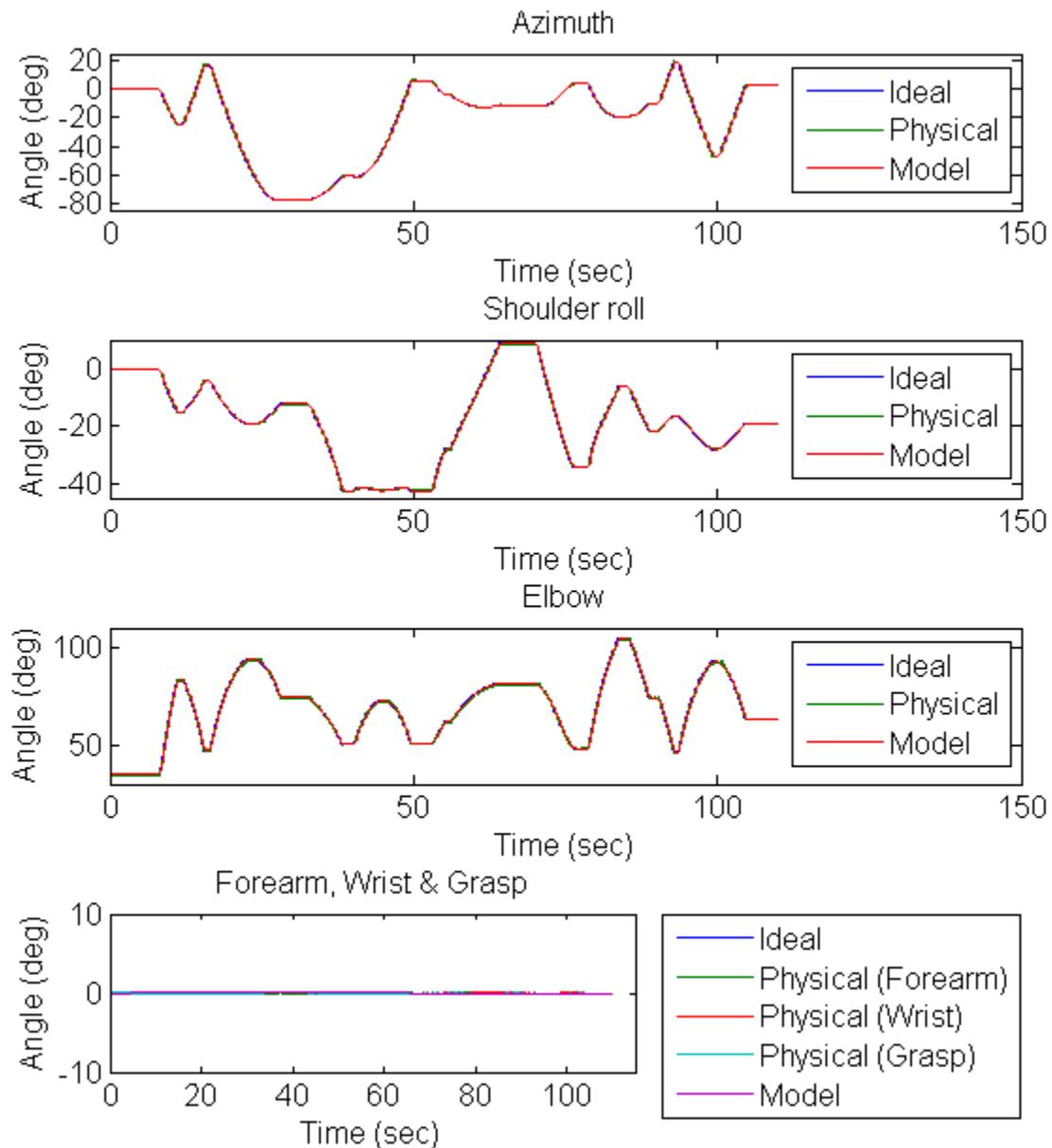


Figure 62. Joint Angles Throughout Tasks 1 to 12

Figure 62 presents all the UBC-PULO joint angles for the ideal case, the simulation model and physical device during completion for all the specified tasks. As shown in the figure, the azimuth, shoulder roll and elbow joints are able to closely follow the ideal joint angles and move concurrently to produce the desired end-effector motion. The forearm, wrist and grasp remain constant (zero in this case) since these joints are not active during the table top or functional operation modes. Table 18 and Table 19 document the joint angle error statistics for both the physical device and the simulation model, which verify their similarity to the desired ideal joint angles. Table 20 illustrates the upper bound joint angle error between the physical prototype and the simulation model for the active joints during completion tasks 1 to 12.

Table 18. Physical Device Joint Angle Errors Throughout Tasks 1 to 12

Joint	Maximum absolute error (degrees)	Mean absolute error (degrees)	Standard deviation (degrees)
Azimuth	2.11	0.43	0.31
Shoulder roll	1.18	0.37	0.19
Elbow	2.78	0.63	0.42
Forearm	0.82	0.14	0.11
Wrist	1.23	0.13	0.13
Grasp	0.61	0.03	0.04

Table 19. Model Joint Angle Error Throughout Tasks 1 to 12

Joint	Maximum absolute error (degrees)	Mean absolute error (degrees)	Standard deviation (degrees)
Azimuth	2.77	0.83	0.89
Shoulder roll	1.78	0.45	0.46
Elbow	3.96	0.87	0.87
Forearm	0.21	0.00	0.00
Wrist	0.09	0.00	0.00
Grasp	0.04	0.00	0.00

Table 20. Upper Bound Error between Active Joints of Model and Physical Prototype

Joint	Maximum absolute error (degrees)	Mean absolute error (degrees)
Azimuth	4.88	1.26
Shoulder roll	2.96	0.82
Elbow	6.74	1.50

6.2.2.2.2 *Orientation mode of operation*

This section presents the results obtained during testing of the *orientation* mode of operation. The objective was to test the functionality of the control algorithm and the performance of the exoskeleton with respect to user inputs for both the UBC-PULO prototype and the simulation model. To achieve this objective, the following tasks were completed:

13. Switch UBC-PULO to orientation mode. Tilt head forwards past the neutral zone to perform wrist extension (i.e. increasing angular direction).
14. Tilt head backwards past the neutral zone to perform wrist flexion (i.e. decreasing angular direction).
15. Tilt head left past the neutral zone to perform forearm pronation (i.e. decreasing angular direction).
16. Tilt head right past the neutral zone to perform forearm supination (i.e. increasing angular direction).
17. Perform extended MID shoulder shrug to close grasp (i.e. increasing angular direction).
18. Perform extended HIGH shoulder shrug to open grasp (i.e. decreasing angular direction).

Figure 63 presents the ideal case, physical device and the simulation model joint angles during completion of the listed tasks. The results suggest that the user can control the orientation and grasp of the manipulator as outlined, which also agrees with the original specification mentioned in Section 2.3.2.2. As expected, the azimuth, shoulder roll and elbow joint positions remain constant. Table 21 and Table 22 show statistics in regards to joint angle error for the physical prototype and the simulation model during the completion of the *orientation mode* tasks. Table 23 illustrates the upper bound error of the joint angle errors during the completion of orientation mode tasks.

Table 21. Orientation Mode Joint Angle Error (Physical Device)

Joint	Maximum absolute error (degrees)	Mean absolute error (degrees)	Standard deviation (degrees)
Azimuth	0.22	0.09	0.02
Shoulder roll	0.11	0.01	0.01
Elbow	0.35	0.07	0.05
Forearm	1.05	0.33	0.27
Wrist	1.23	0.43	0.30
Grasp	1.39	0.25	0.32

Table 22. Orientation Mode Joint Angle Error (Model)

Joint	Maximum absolute error(degrees)	Mean absolute error (degrees)	Standard deviation (degrees)
Azimuth	0.22	0.09	0.02
Shoulder roll	0.11	0.01	0.01
Elbow	0.35	0.07	0.05
Forearm	2.03	0.47	0.57
Wrist	2.26	0.34	0.58
Grasp	1.94	0.36	0.53

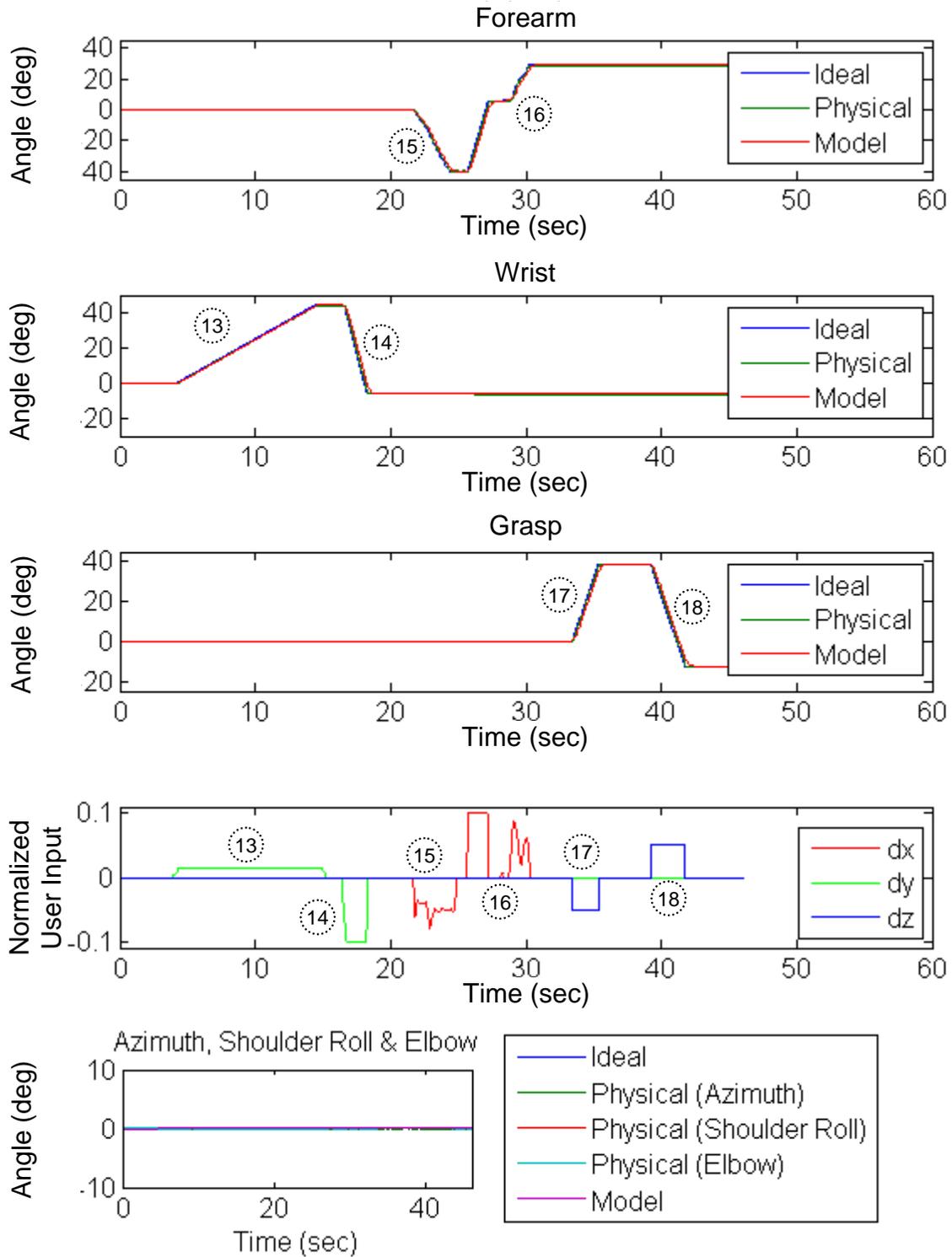


Figure 63. Joint Angles Throughout Tasks 1 to 12

Table 23. Upper Bound Error between Model and Physical Prototype

Joint	Maximum absolute error (degrees)	Mean absolute error (degrees)
Azimuth	0.44	0.18
Shoulder roll	0.22	0.02
Elbow	0.70	0.14
Forearm	3.08	0.80
Wrist	3.49	0.77
Grasp	3.33	0.62

6.3 Conclusion

This chapter demonstrated the simulation model's response to an angular step input at each joint, with the resulting end-effector position compared to that of the physical prototype using the experimental setup previously described as part of the process to validate the VRSE. The results demonstrate a reasonably close performance (i.e. mean end-effector upper bound error of ± 10.1 mm during tasks in question); however, upper bound maximum errors exist (maximum end-effector upper bound error of ± 37.44 mm) which need to be alleviated in the future. The reasons for existence of these differences between the physical device and the VRSE are primarily due to limitations present in the modelling of the drive system, the modeling of the mechanical properties of the robotic exoskeleton (i.e. link masses and inertias), and the existence of non-linearity in the physical drive system (i.e. due to joint backlash and flex-shafts). These limitations will be discussed in more detail in Chapter 7. It was also demonstrated that the commissioned UBC-PULO is now fully functional with respect to the original design and control algorithm specifications. The physical device exhibited good accuracy (i.e. upper bound mean end-effector error of ± 10.1 mm) while completing various tasks in table top mode, functional mode and orientation mode.

Chapter 7 – Discussion and Limitations

7.1 Introduction

Previous research on the UBC-PULO project resulted in the development of user specifications, identifying desired tasks and arm motions required to perform them, establishing design specifications and evaluation criteria, developing a unique control strategy, and completing the construction of an orthosis prototype with user control interfaces. The main objective of this research was to validate the VRSE against the physical prototype. Ideally, to validate the VRSE results are compared with respect to the same inputs and the same control conditions. Thus, to allow for validation of the VRSE an experimental setup was proposed (see Figure 21). This required a functional physical device; however, upon initiation of this work the UBC-PULO prototype was not operational due to mechanical and electrical issues. Thus a full mechanical and electrical review was conducted which lead to the identification of high-priority required improvements outlining the system's shortcomings which were useful for commissioning the physical device. A new electrical system was developed and implemented which allowed for the integration of physical device and the VRSE into one electrical system-serving as a framework required for the experimental setup. Next, the experimental setup's software was developed which was primarily based on new UBC-PULO software using MATLAB, Simulink, XPC Target and SimMechanics. Finally, the UBC-PULO Virtual Reality Simulator [1] was upgraded and its performance was validated against the physical device using the experimental setup. This chapter discusses the

limitations imposed on this work and discusses the experimental setup's capability for serving as platform for future development.

7.2 Limitations

7.2.1 Evaluating the UBC-PULO drive system

The calculation used to provide the motor torque requirement for each joint was based on a simplifying assumption, i.e. that the UBC-PULO electro-mechanical exoskeleton and attached human arm would be a three-body point-mass dynamic system. The calculations ignored the effects of friction, damping and axial forces that are imposed on the system by the user's arm. In addition, calculation of the maximum angular acceleration for each joint assumed maximal leverage distance, and estimate gear efficiency based on the type of gear used [41] and values stated in the corresponding motor-gearbox datasheets. The main goal at this point was to perform a preliminary evaluation of the electrical motors to ensure they could provide sufficient torque to satisfy a desirable system response. It was realized that a full dynamic response study could be performed much more efficiently once the UBC-PULO prototype was commissioned.

Since the UBC-PULO Virtual Reality Simulator [1] was being developed in parallel to the prototype evaluation stage of this work, it was decided to see if it was possible to characterize the flex shafts in terms of a second-order system. After development of a test apparatus and implementation of a least-squares curve fitting tool, it was concluded that the flex shaft could not be accurately modeled as a second-order system as they

exhibit non-linear behaviour. It should be noted that the conclusions reached in regards to the flex-shafts were based on only a limited number of available weights, flex-shaft lengths and torque range (i.e. a torque range that was specific to UBC-PULO). Also, the flex-shaft test apparatus has its limitations in terms of accuracy. Due to time limitation, the non-linear behaviour of the flex shafts was not investigated.

7.2.2 Modelling of the UBC-PULO prototype

The “*model of drive system and exoskeleton*” plant block, which consists of the upgraded UBC-PULO simulator, is primarily based on previous research work [1] and has, therefore, the same limitations. The main modification made to the UBC-PULO Virtual Reality Simulator was upgrading the SimMechanics model to more accurately represent the UBC-PULO prototype (see Section 5.3) with respect to degrees of freedom (i.e. for more details visit Section 2.4) and link mechanical properties (i.e. dimensions, mass and inertia). The UBC-PULO model has the following limitations:

- The model of the flexible drive shafts was not included in the simulator drive system.
- The effects of gear backlash were ignored in the model.
- The mechanical bodies were assumed to be infinitely rigid.
- The mechanical properties of the exoskeleton arm segments were assumed to be entirely either aluminum or plastic, i.e. the materials of the major parts.

- The model geometry and dimensions of the SimMechanics model were constructed based on the original UBC-PULO design documentation and may vary from the physical prototype due to inevitable manufacturing errors.
- The weight and dynamics of the human arm is not included in the simulator.
- The electric motor parameters and joint friction have not been experimentally identified.

The UBC-PULO simulator is a tool that allows assessment of current /or new control algorithm's performance; testing of manipulator work space movements with respect to range of motion and user-friendliness in performing tasks; evaluation of the control algorithm's user friendliness; to serve as a cost-efficient and time-efficient development tool for incorporation and testing of new designs; and, to serve as a training tool to users. The results from Chapter 6 demonstrated that the simulation model response is close to that of the physical device, however, an upper end-effector error bound of ± 37.44 mm exists which can be explained by the limitations listed above. However, it is realized that by ignoring the effects of flex shaft, the gear backlash and assuming infinitely rigid mechanical bodies, the simulation model is unable to simulate the exact response of the physical prototype. Since the simulation model is not validated in open loop, it cannot be used to optimize the low level controller response of the physical prototype (i.e. PID tuning). Although, the weight and dynamics of the human arm are not currently included in the simulation model, the weight of arm segments can be incorporated in the model and validated using the experimental setup in the future to allow for more accurate simulation of the device when worn by the user.

7.2.3 Limitations of the physical prototype

The UBC-PULO prototype was commissioned by implementing the high-priority electrical and mechanical improvements suggested in Chapter 3. Thus, the lower priority items, which are summarized in Table 26 and Table 27 in Appendix C, still do limit the overall performance of the physical prototype. Besides the limitations imposed on the system by lack of implementation of the lower priority items, the commissioned UBC-PULO prototype also has the following limitations:

- Backlash in the elbow joint – although the elbow joint was repaired, the joint needs to be redesigned. After repairing the elbow joint, the backlash and overall functionality of the elbow joint was significantly improved; however, the current design's shaft is not robust enough to tolerate the forces applied on the joint which causes a larger backlash over time.
- Existence of joint cross distortion - the movement of joints affect one another in terms of position, velocity, acceleration and torque. Ideally the mechanical exoskeleton and low level controller would be designed in a way to minimize cross distortion.
- Limitation in initial starting position accuracy – there is a need for a calibration jig in the future to ensure the initial starting position is accurately calibrated the first time the device is programmed.
- Limitation in selecting the optimal PID constants for the low level controller - the PID constants were adjusted by performing HIL testing; however, the optimal PID constants depend on weight, geometry, friction and moment of inertia

experienced by each joint due to the exoskeleton, the users arm, and the end-effector load.

These limitations, especially the existence of elbow backlash, reflect on the validation results by introducing error in joint angles and end-effector positions as the simulation model does not take these non-ideal characteristics into account.

7.2.4 Limitations of VRSE validation

The tests performed on the UBC-PULO prototype demonstrated the full functionality of the prototype with respect to the original control algorithm, and presented data on the physical performance of the commissioned system. However, it is recognized that the results were limited by the following factors:

- Validations or tests performed did not include the weight of the human arm and end-effector loads.
- The UBC-PULO Virtual Reality Simulator was compared with the physical prototype only with respect to angular position, end-effector position and position step response – the model joint angular velocity, joint angular acceleration, joint torque, end-effector velocity and end-effector acceleration were not compared to that of the physical device.

7.3 Using the experimental setup as a platform for future development

One of the main accomplishments of this work was developing a platform for future development. The flexibility of the new system allows for a very efficient design and prototyping cycle. As summarized in Table 10, the electrical hardware has a large number of analog and digital I/Os which are readily available through terminal blocks (see Figure 80). Thus, new user interface devices or user feedback devices can be easily integrated with the system if desired. In addition, the motor driver circuit can drive up to eight 20 Watt electrical motors each with a separate supply voltage (see Appendix E). Thus, the DC motors can be replaced with more powerful motors if desired and it is also possible to add two more motors to the system. In addition, the software combines powerful development tools such as MATLAB, Simulink, SimMechanics, MuPAD, Virtual Reality toolbox and XPC Target that make the following possible:

- Efficient software development
- Modeling, visualization and simulation of the system
- HIL testing, real-time data monitoring and logging data for post analysis
- Experimental setup for validating VRSE

The UBC-PULO software (presented in Chapter 5) was made as modular and parametric as possible to promote efficient additions and modifications to the software. By having multiple *plant* block options (see Section 5.3), the UBC-PULO software is capable of either simulating the prototype's behaviour or to be downloaded/executed in real-time on the physical prototype. Thus any changes made to the UBC-PULO software can be simulated on the host computer or downloaded onto the prototype for

physical testing. In addition, the UBC-PULO software can accommodate any mechanical design changes by updating the SimMechanics model (i.e. The UBC-PULO SolidWorks model can be updated and imported as a SimMechanics model [1]) or the drive system model [1]. Therefore, in case of a redesign or future upgrades, the simulation tools can be upgraded to match the new design, allowing the system to be evaluated in terms of work space range, joint angle range, user friendliness, control algorithm, user safety and performance during everyday tasks before the physical modifications are made. Since the VRSE system can be validated using the experimental setup, the simulation test results guarantee accuracy of the results within the measured validation error bounds. Once the simulations exhibit promising results, the physical modifications can be carried out on the physical device. Using tools such as HIL testing or real-time data monitoring parameters can be effectively tuned and the upgraded prototype can be readily tested. Since the system is portable, clinical trials and patient studies can also be performed in the future. Therefore, for the various reasons discussed, the experimental setup has created an effective platform for future development within the UBC-PULO project, contributing to a time and cost efficient prototyping cycle.

7.4 Conclusion

This chapter discussed the main limitations of this research work and highlighted how the experimental setup can be used as a platform for future development. The

following chapter summarizes the conclusions reached by this research work and makes recommendations for future work.

Chapter 8 – Conclusion and Recommendations

8.1 Introduction

Every year there are numerous instances of severe upper limb paresis all over the world [3], [4], [9–14]. The affected people face partial or complete loss of arm function, making them unable to perform everyday activities and significantly lowering their quality of life. The goal of the UBC-PULO project is to develop a highly functional, portable and user friendly assistive device that restores function to an entire upper limb, and to enable the user to perform high priority every day task activities via worn user control interfaces. Previous work has interviewed potential users, identified the desired high-priority daily tasks, established design specifications, developed a unique control strategy along with user interfaces and completed the construction of a five DOF orthosis prototype. The objective of this research was to validate the UBC-PULO virtual reality simulation model (VRSE) [1], which required the completion of the following tasks:

1. Literature review of similar assistive devices.
2. Understanding and reviewing the current state of the orthosis.
3. Evaluating the experimental setup, identifying system parameters and determining the required high-priority improvements.
4. Upgrading the UBC-PULO hardware to: (1) satisfy the required high-priority improvements, (2) allow for characterization of the physical prototype and (3) to allow validation of the VRSE.
5. Developing the UBC-PULO software to allow for validation of the VRSE.

6. Commissioning the experimental setup.
7. Validating the UBC-PULO Simulator using the experimental setup.

This chapter focuses on the conclusions reached upon completion of each task which led to the fulfilment the thesis objective. This chapter will also present recommendations for future work.

8.2 Conclusions

8.2.1 Literature review

An understanding of the state of the art relative to upper limb assistive devices and a thorough understanding of the UBC-PULO development provided the fundamental knowledge necessary for carrying out this research work. A number of current assistive and rehabilitative devices were reviewed and it was concluded that a highly functional, user-friendly and portable powered assistive device capable of performing every day activities has yet to be developed. The UBC Powered Upper Limb Orthosis is the only known assistive solution focusing on restoring daily-living-tasks arm function to users with two paralyzed upper limbs – i.e. the people in the most need of such a device. Besides being intuitive and user friendly, one of the great advantages of UBC-PULO is that it allows the user to take pleasure in feeling objects and to enjoy the self gratifying ability of doing tasks themselves. Despite being designed specifically for the extreme case of total upper limb dysfunction (Table 3), users with only partial

paralysis (who still satisfy the remaining user requirements stated in Table 3) may also benefit from use of the powered orthosis.

8.2.2 Reviewing the UBC-PULO prototype

Due to both mechanical and electrical issues, the UBC-PULO prototype was not operational at the time this research work was initiated. Thus, a full mechanical and electrical review was conducted to identify the high-priority improvements needed to commission the prototype. The investigation also resulted in identification of a prioritized list of improvements which serves as a reference guideline for future development of the project presented and is presented in Appendix C.

8.2.3 Experimental setup

Chapter 4 provides details on the electrical and mechanical improvements that lead to the commissioning of the UBC-PULO prototype. In addition, the development and implementation of the necessary hardware changes to allow the integration of the experimental test setup for validation of the simulator (see Figure 21 and Section 3.1) was presented. The elbow joint was repaired and the electrical system of the UBC-PULO was completely redesigned, which included modifications to the electrical boards, user control interfaces and joint angles sensors. The research work performed to develop a new UBC-PULO electrical system included:

- Understanding and reviewing the previous UBC-PULO electrical system.
- Establishing the design requirements for the new electrical system.
- Proposing, designing and implementing a new improved solution for the user head interface and joint angle sensors.

- Exploring possible solutions for designing the experimental setup and choosing to use XPC Target in combination with the PC/104+ form factor.
- Performing research with respect to selecting electrical components, how to interconnect them and to design a PCB.
- Selecting the PCB components, DAQ cards and SBC board after completing the iterative cycle of schematic design, PCB component selection, PCB optimization (i.e. component positioning/space constraints, heat-dissipation constraints, noise considerations and flexibility considerations). The process of component selection and PCB design is an iterative cycle where many interdependent electrical and mechanical constraints had to be satisfied. For example, the overall size of the PCB was only known once all of the components were known and positioned in their optimal location. The optimized location depends on the type of I/O, location of other components, as well as other requirements such as heat dissipation. As a result, changing any one component usually led to changing several other components.
- Completing a final PCB design along with the assembling and testing of it.

The UBC-PULO electrical system acts as an experimental setup to not only allow for validation of the VRSE, but also promote a time-efficient and cost-efficient future development of the UBC-PULO project. Thus, a main feature of the new fully functional UBC-PULO electrical system is that it acts as a “platform for future development” by supporting a flexible hardware and software that is integrated with the experimental setup.

The UBC-PULO software was developed in MATLAB, Simulink, MuPAD, SimMechanics and XPC Target as described in Chapter 5. Besides the original control algorithm, the implementation of a new inverse *kinematic stability* features ensures overall inverse kinematic stability at all times, which has tremendously contributed to the full functionality of the device. Besides incorporating the experimental setup software used for validating the VRSE, the new UBC-PULO software fully complements the concept of “platform for future development.” Overall, the UBC-PULO software provides a *framework* to make the following possible:

- Efficient implementation of new UBC-PULO algorithms/software concepts via an overall modular and high-level software design.
- Efficient testing of new UBC-PULO algorithms/software concepts via simulation and visualization tools.
- Simulation, visualization and testing of new or finalized mechanical design concepts via an upgradable UBC-PULO SimMechanics model (i.e. VRSE).
- HIL testing, real-time data monitoring and parameter tuning during and after execution.
- Real-time execution of the experimental setup to allow for validation of VRSE.

Therefore, the new electrical system and software have created an effective platform for future development, for the various reasons discussed, contributing to a time and cost efficient UBC-PULO prototyping cycle.

8.2.4 Validating the VRSE

Using the experimental setup, the simulation model performance was validated against the physical prototype by performing two sets of experiments, which were explained in detail in Chapter 6. The results demonstrated that while performing a set of tasks over the full range of manipulator motion, the upper bound mean end-effector error (i.e. between the simulation model and physical prototype) was ± 10.1 mm for the X, Y or Z axis. This shows that the VRSE can be used to simulate the performance of the physical device in virtual reality with relatively good accuracy in the tasks in question.

8.2.5 Research conclusions

This section highlights the findings of this work with respect to the original research questions.

The electric motor model constants (see Figure 18) were identified from the data sheet. Although a good degree of accuracy is expected from the values provided in the motor datasheets, the physical motor would most likely behave differently from the stated values. Characterizations of the actual electrical motors are recommended.

The VRSE drive system model does not take the effects of the flex shaft into account. In an attempt to formulate a second order model for the flex shafts, a test apparatus was designed and implemented to allow for characterization of the flex shafts. It was demonstrated that the flex-shafts have a non-linear behaviour and that

they cannot be modeled by a second-order system. Future characterization of the flex-shafts for a non-linear model is recommended.

The VRSE joints assume a linear model and require the gear ratio as an input parameter. Thus, non-linear effects of the physical system such as gear backlash are ignored by the joint model. The parameters input into the joint models were based on the original UBC-PULO design specifications and the physically measured gear ratios were not identified.

To model of the exoskeleton, the VRSE uses six mechanical components which were modeled in SolidWorks and imported in SimMechanics. The components were assumed to be either plastic or aluminum (i.e. the material used on the physical prototype). Although, the geometrical representations of the exoskeleton segments are relatively accurate, the masses may vary from the physical device. To more accurately model the physical device, the weights of the exoskeleton's segments need to be identified and input to the model.

To validate the model, an experimental setup was proposed in Section 3.1 (see Figure 21), which allowed for output data comparison of the simulation mode and physical model under the same input conditions. The experimental setup was developed and implemented and validation tests were performed by two experiments. The first experiment tested the closed loop joint response for each individual joint for the same input step response. The second experiment tested the overall performance of the manipulator with respect to the user performing various pre-defined tasks over the full range of motion. The experiment results demonstrated that the VRSE the simulation model and the physical prototype have a reasonably close response while performing

the tasks in question (i.e. mean end-effector upper bound error of ± 10.1 mm). However, upper bound maximum errors exist (maximum end-effector upper bound error of ± 37.44 mm) due to the differences that exist between the physical prototype and the VRSE model. The reasons for the existence of these differences between the physical device and the VRSE are primarily due to limitations present in the modelling of the drive system, the modeling of the mechanical properties of the robotic, and the existence of non-linearity in the physical drive system, which were discussed in Chapter 7.

8.3 Future recommendations

Part of this research work required investigation of electrical and mechanical components which led the identification of a prioritize list of improvement listed in Appendix C. Besides the recommendations described in the prioritized list of improvement (see Table 26 and Table 27), the following are recommended for the future development of the UBC PULO project:

- The experimental setup should be used to perform more VRSE validation tests.
- The physical device should be further characterized to identify more accurate simulation parameters.
- The open loop response of the system should be validated.
- The physical prototype and simulation performance should be evaluated while carrying the weight of the human arm and an end-effector load.

- The *platform for future development* should be utilized to evaluate or optimize the design of future improvements to the mechanical components, control algorithm or software by performing simulations. Tests can include work space studies, evaluation of performance on predefined tasks, user-friendliness and overall drive system performance. Once the design is final it can be implemented on the physical prototype.
- Clinical evaluations need to be performed against pre-defined high-priority tasks, overall user-friendliness, user control input devices and control algorithm on both the VRSE and physical prototype.
- The forearm mechanical components need to be redesigned to alleviate the issues caused by an offset rotation axis between the user's forearm and exoskeletons forearm. A design has been recommended in Appendix C.
- The prototype's shoulder design needs to be redesigned such that it has the same axis of rotation and translational movements as the user's shoulder. Additionally, the new shoulder design needs to be more robust against friction when under load.
- The grasp mechanism needs to be redesigned to allow users to securely grasp objects of different shapes and sizes.
- The casings for the flex-shafts need to be installed and the flex shafts need to be attached at several positions along the orthosis structure for guidance. The flex-shaft length for each joint need to be optimized to avoid entanglement or shortage over the joint range of motion.
- The elbow joint needs to be redesigned to alleviate the backlash issues.

- The shoulder roll torque generation capability needs to increase to assure proper operation under load.
- Mechanical joint limits need to be implementing for user safety.
- Smaller arm cuffs need to be installed to allow greater range of elbow joint motion.
- User comfort need to be improved and the prototype's weight needs to be optimized (see Section 3.2.3).
- The electrical systems power consumption and batteries need to be optimized.
- Additional modes of operations and alternate control strategies should be implemented.

References

- [1] J. Zhao, "A virtual reality simulator for the UBC powered upper limb orthosis," University of British Columbia, 2011.
- [2] C. Anglin, "A functional task analysis and motion simulation for the development of a powered upper-limb orthosis," University of British Columbia, Vancouver, Canada, 1993.
- [3] "Multiple Sclerosis Statistics." [Online]. Available: <http://multiple-sclerosis.emedtv.com/multiple-sclerosis/multiple-sclerosis-statistics.html>. [Accessed: 15-Apr-2012].
- [4] D. Lloyd-Jones, R. J. Adams, T. M. Brown, M. Carnethon, S. Dai, G. De Simone, T. B. Ferguson, E. Ford, K. Furie, C. Gillespie, A. Go, K. Greenlund, N. Haase, S. Hailpern, P. M. Ho, V. Howard, B. Kissela, S. Kittner, D. Lackland, L. Lisabeth, A. Marelli, M. M. McDermott, J. Meigs, D. Mozaffarian, M. Mussolino, G. Nichol, V. L. Roger, W. Rosamond, R. Sacco, P. Sorlie, R. Stafford, T. Thom, S. Wasserthiel-Smoller, N. D. Wong, and J. Wylie-Rosett, "Heart Disease and Stroke Statistics—2010 Update A Report From the American Heart Association," *Circulation*, vol. 121, no. 7, p. e46–e215, Feb. 2010.
- [5] D. P. Romilly, C. Anglin, R. G. Gosine, C. Hershler, and S. U. Raschke, "A functional task analysis and motion simulation for the development of a powered upper-limb orthosis," *IEEE Transactions on Rehabilitation Engineering*, vol. 2, no. 3, pp. 119–129, 1994.
- [6] C. A. Stanger, C. Anglin, W. S. Harwin, and D. P. Romilly, "Devices for assisting manipulation: a summary of user task priorities," *Rehabilitation Engineering, IEEE Transactions on*, vol. 2, no. 4, pp. 256–265, Dec. 1994.
- [7] D. Brown, "Control system development for a powered upper-limb orthosis," 1996.
- [8] R. M. Mahoney, "Robotic products for rehabilitation: Status and strategy," in *Proceedings of ICORR*, 1997, vol. 97, pp. 12–22.
- [9] "Statistics about Muscular Dystrophy - RightDiagnosis.com." [Online]. Available: http://www.rightdiagnosis.com/m/musdys/stats.htm#medical_stats. [Accessed: 15-Apr-2012].
- [10] "Polio - Symptoms, Causes, Treatments - Better Medicine." [Online]. Available: <http://www.localhealth.com/article/polio>. [Accessed: 15-Apr-2012].
- [11] "Amyotrophic Lateral Sclerosis - Symptoms, Causes, Treatments - Better Medicine." [Online]. Available: <http://www.localhealth.com/article/amyotrophic-lateral-sclerosis>. [Accessed: 15-Apr-2012].
- [12] "SPINALCORD: Spinal Cord Injury Information Network." [Online]. Available: <http://www.spinalcord.uab.edu/>. [Accessed: 15-Apr-2012].

- [13] J. Stein, K. Narendran, J. McBean, K. Krebs, and R. Hughes, "Electromyography-controlled exoskeletal upper-limb-powered orthosis for exercise training after stroke," *Am J Phys Med Rehabil*, vol. 86, no. 4, pp. 255–261, Apr. 2007.
- [14] *Wearable Robots: Biomechatronic Exoskeletons*. Wiley Online Library, 2008.
- [15] H. S. Lo and S. Q. Xie, "Exoskeleton robots for upper-limb rehabilitation: State of the art and future prospects," *Medical Engineering & Physics*, 2011.
- [16] C. G. Burgar, P. S. Lum, P. C. Shor, and H. F. Machiel Van der Loos, "Development of robots for rehabilitation therapy: the Palo Alto VA/Stanford experience," *J Rehabil Res Dev*, vol. 37, no. 6, pp. 663–673, Dec. 2000.
- [17] R. Loureiro, F. Amirabdollahian, M. Topping, B. Driessen, and W. Harwin, "Upper Limb Robot Mediated Stroke Therapy—GENTLE/s Approach," *Autonomous Robots*, vol. 15, no. 1, pp. 35–51, 2003.
- [18] "Kinova Robotics | Robotic arm for research and rehab." [Online]. Available: <http://kinovarobotics.com/index.html>. [Accessed: 14-Apr-2012].
- [19] T. J. Engen, "Development of Externally Powered Upper Extremity Orthotic Systems," *J Bone Joint Surg Br*, vol. 47-B, no. 3, pp. 465–468, 1985.
- [20] Nef, M. Guidali, and R. Riener, "ARMin III—arm therapy exoskeleton with an ergonomic shoulder actuation," *Applied Bionics and Biomechanics*, vol. 6, no. 2, pp. 127–142, 2009.
- [21] A. Tsukahara, Y. Hasegawa, and Y. Sankai, "Standing-up motion support for paraplegic patient with Robot Suit HAL," in *Rehabilitation Robotics, 2009. ICORR 2009. IEEE International Conference on*, 2009, pp. 211–217.
- [22] A. Tsukahara, R. Kawanishi, Y. Hasegawa, and Y. Sankai, "Sit-to-stand and stand-to-sit transfer support for complete paraplegic patients with robot suit HAL," *Advanced robotics*, vol. 24, no. 11, pp. 1615–1638, 2010.
- [23] Y. Ren, H.-S. Park, and L.-Q. Zhang, "Developing a whole-arm exoskeleton robot with hand opening and closing mechanism for upper limb stroke rehabilitation," in *Rehabilitation Robotics, 2009. ICORR 2009. IEEE International Conference on*, 2009, pp. 761–765.
- [24] M. Mistry, P. Mohajerian, and S. Schaal, "Arm movement experiments with joint space force fields using an exoskeleton robot," in *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, 2005, pp. 408–413.
- [25] C. Carignan, J. Tang, and S. Roderick, "Development of an exoskeleton haptic interface for virtual task training," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 3697–3702.
- [26] G. R. Johnson, D. A. Carus, G. Parrini, S. Scattareggia Marchese, and R. Valeggi, "The design of a five-degree-of-freedom powered orthosis for the upper limb," *Proc Inst Mech Eng H*, vol. 215, no. 3, pp. 275–284, 2001.
- [27] F. Martinez, I. Retolaza, A. Pujana-Arrese, A. Cenitagoya, J. Basurko, and J. Landaluze, "Design of a five actuated DoF upper limb exoskeleton oriented to

- workplace help,” in *Biomedical Robotics and Biomechatronics, 2008. BioRob 2008. 2nd IEEE RAS EMBS International Conference on*, 2008, pp. 169 –174.
- [28] R. A. R. C. Gopura, K. Kiguchi, and Y. Li, “SUEFUL-7: A 7DOF upper-limb exoskeleton robot with muscle-model-oriented EMG-based control,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 1126 –1131.
- [29] A. H. A. Stienen, E. E. G. Hekman, F. C. T. van der Helm, and H. van der Kooij, “Self-Aligning Exoskeleton Axes Through Decoupling of Joint Rotations and Translations,” *Robotics, IEEE Transactions on*, vol. 25, no. 3, pp. 628 –633, Jun. 2009.
- [30] D. G. Caldwell, N. G. Tsagarakis, S. Kousidou, N. Costa, and I. Sarakoglou, “‘soft’ exoskeleton for upper and lower body rehabilitation - design, control and testing,” *International Journal of Humanoid Robotics*, vol. 04, no. 03, p. 549, 2007.
- [31] S. Balasubramanian, R. Wei, M. Perez, B. Shepard, E. Koeneman, J. Koeneman, and J. He, “RUPERT: An exoskeleton robot for assisting rehabilitation of arm functions,” in *Virtual Rehabilitation, 2008*, 2008, pp. 163 –167.
- [32] J.-F. Zhang, C.-J. Yang, Y. Chen, Y. Zhang, and Y.-M. Dong, “Modeling and control of a curved pneumatic muscle actuator for wearable elbow exoskeleton,” *Mechatronics*, vol. 18, no. 8, pp. 448–457, Oct. 2008.
- [33] D. De Rossi, F. Carpi, F. Lorussi, E. P. Scilingo, and A. Tognetti, “Wearable kinesthetic systems and emerging technologies in actuation for upperlimb neurorehabilitation,” in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, 2009, pp. 6830 –6833.
- [34] S. Schulz, C. Pylatiuk, A. Kargov, I. Gaiser, O. Schill, M. Reischl, U. Eck, and R. Rupp, “Design of a Hybrid Powered Upper Limb Orthosis,” in *World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany*, vol. 25/9, O. Dössel, W. C. Schlegel, and R. Magjarevic, Eds. Springer Berlin Heidelberg, 2009, pp. 468–471.
- [35] K. Kiguchi and M. Liyanage, “A study of a 4DOF upper-limb power-assist intelligent exoskeleton with visual information for perception-assist,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008, pp. 3666 –3671.
- [36] M. Baklouti, E. Monacelli, V. Guitteny, and S. Couvet, “Intelligent Assistive Exoskeleton with Vision Based Interface,” in *Smart Homes and Health Telematics*, vol. 5120, S. Helal, S. Mitra, J. Wong, C. Chang, and M. Mokhtari, Eds. Springer Berlin / Heidelberg, 2008, pp. 123–135.
- [37] J. C. Perry, J. Rosen, and S. Burns, “Upper-Limb Powered Exoskeleton Design,” *Mechatronics, IEEE/ASME Transactions on*, vol. 12, no. 4, pp. 408 –417, Aug. 2007.
- [38] S. J. Ball, I. E. Brown, and S. H. Scott, “MEDARM: a rehabilitation robot with 5DOF at the shoulder complex,” in *Advanced intelligent mechatronics, 2007 IEEE/ASME international conference on*, 2007, pp. 1 –6.

- [39] A. L. O'Connell and E. B. Gardner, *Understanding the scientific bases of human movement*. Williams & Wilkins Co., 1972.
- [40] C. E. Clauser, J. T. McConville, and J. W. Young, *Weight, volume, and center of mass of segments of the human body*. Aerospace Medical Research Laboratory, Aerospace Medical Division, Air Force Systems Command, 1969.
- [41] "Robot Gears Tutorial." [Online]. Available: http://www.societyofrobots.com/mechanics_gears.shtml. [Accessed: 15-Apr-2012].
- [42] T. Woo, "Flex-shaft Response," University of British Columbia, Vancouver, Canada, Sep. 2011.
- [43] "Human Reaction Times as a Response to Delays in Control Systems." [Online]. Available: <http://www.measurepolis.fi/ENindex.html>. [Accessed: 15-Apr-2012].
- [44] J. P. Merlet, "Jacobian, Manipulability, Condition Number, and Accuracy of Parallel Robots," *Journal of Mechanical Design*, vol. 128, no. 1, pp. 199–206, 2006.
- [45] Microstar Laboratories, Inc., "Ziegler-Nichols Tuning Rules for PID," 2009. [Online]. Available: <http://www.mstarlabs.com/control/znrule.html>. [Accessed: 15-Apr-2012].
- [46] N. Klopčar, M. Tomšič, and J. Lenarčič, "A kinematic model of the shoulder complex to evaluate the arm-reachable workspace," *Journal of Biomechanics*, vol. 40, no. 1, pp. 86–91, Jan. 2007.
- [47] D. L. Jones, "PBC Design Tutorial," 2004.
- [48] "Products | Actuators | Custom actuators:: Harmonic Drive LLC." [Online]. Available: <http://www.harmonicdrive.net/products/actuators/customactuators/>. [Accessed: 15-Apr-2012].
- [49] "Power Transmission Engineering July 2006," *The Basics of Harmonic Drive Gear*. [Online]. Available: <http://www.powertransmission.com/issues/0706/harmonic.htm>. [Accessed: 15-Apr-2012].
- [50] T. Woo, "Forearm Redesign Proposal," University of British Columbia, Vancouver, Canada, Sep. 2011.
- [51] "Power PAD Made Easy," 2004.
- [52] "PC/104 Specification - Version 2.3," 1996.

Appendix A– UBC-PULO Forward and Inverse Kinematics

The UBC-PULO utilizes the Denavit-Hartenberg method, which models the manipulator as a series of rigid links connected in series, to calculate the forward kinematics and inverse kinematics of the manipulator [1], [5]. The forward kinematic calculations are essential for determining the manipulator's end-effector (i.e. user's hand) position and orientation at all times. Once the manipulator's joint angles are known (i.e. real-time angular data acquired from joint angle sensors), the forward kinematics solution determines the end-effector position and orientation by calculating the DH homogeneous transformations. In contrast, the inverse kinematics is essential for determining the required joint angles that drive the manipulator to a desired end-effector position. The details on the forward kinematics and inverse kinematics axes layout, coordinates system definition and theory of operation have been thoroughly explained in previous research [1], [5].

This section serves to provide details on the calculations performed using the most up to date system parameters and to elaborate on the parametric kinematics development which is utilized by the current UBC-PULO software.

A.1 Forward kinematics

The up to date DH parameters, which are used for computing the forward kinematics for the UBC-PULO, are listed in Table 24:

Table 24. DH Parameters

Joint number	θ_i (degrees)	α_i (degrees)	a_i (mm)	d_i (mm)
$i=1$	t_1	$\alpha_1 = -110$	$a_1 = 0$	$d_1 = 0$
$i=2$	t_2	$\alpha_2 = 90$	$a_2 = 10$	$d_2 = 278.5$
$i=3$	t_3	$\alpha_3 = -90$	$a_3 = 0$	$d_3 = 10$
$i=4$	t_4	$\alpha_4 = 90$	$a_4 = -26$	$d_4 = 240$
$i=5$	t_5	$\alpha_5 = -90$	$a_5 = -120$	$d_5 = 0$

Thus, the overall transformation matrix can be calculated by:

$$A_{i-1,i} = \begin{bmatrix} \cos(t_i) & -\sin(t_i)\cos(\alpha_i) & \sin(t_i)\sin(\alpha_i) & a_i \cos(t_i) \\ \sin(t_i) & \cos(t_i)\cos(\alpha_i) & -\cos(t_i)\sin(\alpha_i) & a_i \sin(t_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{05} = A_{01}A_{12}A_{23}A_{34}A_{45} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ 0 & 1 \end{bmatrix}$$

Where,

$A_{i-1,i}$ = The homogeneous transformation matrix for each joint number

T_{05} = The overall transformation matrix describing the current end-point position (i.e. P matrix) and orientation (i.e. R matrix) based on current joint angles.

$$r_{11} = -\cos(t_1) \sigma_{13} - \sigma_2 \sin(t_1) \sigma_9 - \sigma_1 \sin(t_1) \sigma_4$$

$$r_{12} = \sigma_2 \sin(t_1) \sigma_5 - \cos(t_1) \sigma_6 + \sigma_1 \sin(t_1) \sin(t_3) \sin(t_4)$$

$$r_{13} = -\cos(t_1) \sigma_{12} - \sigma_2 \sin(t_1) \sigma_8 - \sigma_1 \sin(t_1) \sigma_3$$

$$P_x = \cos(t_1) \sigma_{10} - \sigma_2 \sin(t_1) \sigma_7 - \sigma_1 \sin(t_1) \sigma_{11}$$

$$r_{21} = \sigma_2 \cos(t_1) \sigma_9 - \sin(t_1) \sigma_{13} + \sigma_1 \cos(t_1) \sigma_4$$

$$r_{22} = -\sin(t_1) \sigma_6 - \sigma_2 \cos(t_1) \sigma_5 - \sigma_1 \cos(t_1) \sin(t_3) \sin(t_4)$$

$$r_{23} = \sigma_2 \cos(t_1) \sigma_8 - \sin(t_1) \sigma_{12} + \sigma_1 \cos(t_1) \sigma_3$$

$$\begin{aligned}
P_y &= \sin(t_1) \sigma_{10} + \sigma_2 \cos(t_1) \sigma_7 + \sigma_1 \cos(t_1) \sigma_{11} \\
r_{31} &= \sigma_1 \sigma_9 - \sigma_2 \sigma_4 \\
r_{32} &= \sigma_2 \sin(t_3) \sin(t_4) - \sigma_1 \sigma_5 \\
r_{33} &= \sigma_1 \sigma_8 - \sigma_2 \sigma_3 \\
P_z &= \sigma_1 \sigma_7 - \sigma_2 \sigma_{11}
\end{aligned}$$

$$\begin{aligned}
\sigma_1 &= \sin(7\pi/18) \\
\sigma_2 &= \cos(7\pi/18) \\
\sigma_3 &= \cos(t_3) \cos(t_5) - \cos(t_4) \sin(t_3) \sin(t_5) \\
\sigma_4 &= \cos(t_3) \sin(t_5) + \cos(t_4) \cos(t_5) \sin(t_3) \\
\sigma_5 &= \cos(t_2) \cos(t_4) - \cos(t_3) \sin(t_2) \sin(t_4) \\
\sigma_6 &= \cos(t_4) \sin(t_2) + \cos(t_2) \cos(t_3) \sin(t_4) \\
\sigma_7 &= \sin(t_2) \sigma_{14} - 10 \sin(t_2) + \cos(t_2) \sigma_{15} \\
\sigma_8 &= \sin(t_2) \sigma_{16} + \cos(t_2) \sin(t_4) \sin(t_5) \\
\sigma_9 &= \sin(t_2) \sigma_{17} - \cos(t_2) \cos(t_5) \sin(t_4) \\
\sigma_{10} &= 10 \cos(t_2) + \sin(t_2) \sigma_{15} - \cos(t_2) \sigma_{14} \\
\sigma_{11} &= \cos(t_3) \sigma_{19} - \sin(t_3) \sigma_{18} + 278.5 \\
\sigma_{12} &= \cos(t_2) \sigma_{16} - \sin(t_2) \sin(t_4) \sin(t_5) \\
\sigma_{13} &= \cos(t_2) \sigma_{17} + \cos(t_5) \sin(t_2) \sin(t_4) \\
\sigma_{14} &= \cos(t_3) \sigma_{18} + \sin(t_3) \sigma_{19} \\
\sigma_{15} &= 26 \sin(t_4) - 120 \cos(t_5) \sin(t_4) + 10 \\
\sigma_{16} &= \cos(t_5) \sin(t_3) + \cos(t_3) \cos(t_4) \sin(t_5) \\
\sigma_{17} &= \sin(t_3) \sin(t_5) - \cos(t_3) \cos(t_4) \cos(t_5) \\
\sigma_{18} &= 26 \cos(t_4) - 120 \cos(t_4) \cos(t_5) \\
\sigma_{19} &= 120 \sin(t_5) + 240
\end{aligned}$$

A.2 Inverse kinematics

As presented in Section 5.2.2, the UBC-PULO calculates the required incremental change in joint angles to drive the end-effector to the desired end-effector position based on user inputs. The inverse kinematics can be solved once the inverse Jacobian is calculated for the concurrent joint angle values. As presented on equation 21, the

required incremental change in joint angles is calculated by multiplying the desired incremental end-effector position with the inverse Jacobian matrix.

$$\partial\theta = J^{-1}\partial P \quad \text{Where, } \theta = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad \text{eq.21}$$

Since the Jacobian matrix is a function of joint angles, the UBC-PULO inverse kinematics algorithm evaluates the inverted Jacobian using the concurrent joint angle values during each task execution cycle (i.e. time required by the SBC to process all calculations and acquire all data). The symbolic Jacobian calculations are presented below for the given DH parameters listed in Table 25:

$$J = \begin{bmatrix} \frac{\partial P_x}{\partial t_1} & \frac{\partial P_x}{\partial t_2} & \frac{\partial P_x}{\partial t_3} \\ \frac{\partial P_y}{\partial t_1} & \frac{\partial P_y}{\partial t_2} & \frac{\partial P_y}{\partial t_3} \\ \frac{\partial P_z}{\partial t_1} & \frac{\partial P_z}{\partial t_2} & \frac{\partial P_z}{\partial t_3} \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$a = -\sin(t_1) \sigma_3 - \sigma_1 \cos(t_1) \sigma_4 - \sigma_2 \cos(t_1) \sigma_6$$

$$b = \cos(t_1) \sigma_4 + \sigma_1 \sin(t_1) \sigma_3,$$

$$c = \sigma_2 \sin(t_1) \sigma_7 - \cos(t_1) \cos(t_2) \sigma_5 - \sigma_1 \sin(t_1) \sin(t_2) \sigma_5$$

$$d = \cos(t_1) \sigma_3 - \sigma_1 \sin(t_1) \sigma_4 - \sigma_2 \sin(t_1) \sigma_6$$

$$e = \sin(t_1) \sigma_4 - \sigma_1 \cos(t_1) \sigma_3$$

$$f = \sigma_1 \cos(t_1) \sin(t_2) \sigma_5 - \sigma_2 \cos(t_1) \sigma_7 - \cos(t_2) \sin(t_1) \sigma_5$$

$$g = 0$$

$$h = -\sigma_2 \sigma_3$$

$$i = \sigma_1 \sigma_7 + \sigma_2 \sin(t_2) \sigma_5$$

$$\sigma_1 = \sin(7\pi/18)$$

$$\sigma_2 = \cos(7\pi/18)$$

$$\sigma_3 = 10 \cos(t_2) + \sin(t_2) \sigma_8 - \cos(t_2) \sigma_7$$

$$\begin{aligned}\sigma_4 &= \sin(t_2) \sigma_7 - 10 \sin(t_2) + \cos(t_2) \sigma_8 \\ \sigma_5 &= \cos(t_3) \sigma_{10} - \sin(t_3) \sigma_9 \\ \sigma_6 &= \cos(t_3) \sigma_{10} - \sin(t_3) \sigma_9 + 278.5 \\ \sigma_7 &= \cos(t_3) \sigma_9 + \sin(t_3) \sigma_{10} \\ \sigma_8 &= 26 \sin(t_4) - 120 \cos(t_5) \sin(t_4) + 10 \\ \sigma_9 &= 26 \cos(t_4) - 120 \cos(t_4) \cos(t_5) \\ \sigma_{10} &= 120 \sin(t_5) + 240\end{aligned}$$

The UBC-PULO uses this symbolic Jacobian representation and evaluates it for the current joint angles during each task execution cycle and computes the Inverse Jacobian by the LU decomposition method as shown in Figure 42 in Section 5.2.2.

A.3 Parametric kinematics development in MuPAD

A parametric code has been developed in MuPAD which calculates a simplified symbolic representation of inverse kinematic and forward kinematic equations that are utilized by the UBC-PULO software. Once the DH parameters are entered, the symbolic calculations can be performed by clicking on the MuPAD's "evaluate all" button. The code also automatically generates the corresponding inverse kinematics and forward kinematics in the "IK.m" and "FK.m" MATLAB files (M-file) respectively under the file path "C:\\Users\\Darius\\Desktop\\XPC_Orthosis_2". The contents of the "IK.m" M-file can be directly copied to the UBC-PULO software's *Jacobian* block (see Figure 42) to fully update the inverse kinematics based on the entered DH parameters. Similarly, the contents of the "FK.m" M-file can be directly copied to the UBC-PULO software's *Forward Kinematic* block (see Figure 48) to update the system. The MuPAD code is presented below:

```
alp1:=-110*((PI)/180); alp2:= PI/2;alp3:=-PI/2;alp4:=PI/2;alp5:=-PI/2
```

$$-\frac{11\pi}{18}$$

$$\frac{\pi}{2}$$

$$-\frac{\pi}{2}$$

$$\frac{\pi}{2}$$

$$-\frac{\pi}{2}$$

```
a1:=0;a2:=10;a3:=0;a4:=-26;a5:=120
```

0

10

0

-26

120

```
d1:=0;d2:=278.5;d3:=10;d4:=240;d5:=0
```

0

278.5

10

240

0

```
DH := matrix([[t1,t2,t3,t4,t5],[d1,d2,d3,d4,d5],
[alp1,alp2,alp3,alp4,alp5],[a1,a2,a3,a4,a5]])
```

$$\begin{pmatrix} t1 & t2 & t3 & t4 & t5 \\ 0 & 278.5 & 10 & 240 & 0 \\ -\frac{11\pi}{18} & \frac{\pi}{2} & -\frac{\pi}{2} & \frac{\pi}{2} & -\frac{\pi}{2} \\ 0 & 10 & 0 & -26 & 120 \end{pmatrix}$$

```
A01:=matrix([[cos(t1),-sin(t1)*cos(alp1),sin(t1)*sin(alp1),a1*cos(t1)],
[sin(t1),cos(t1)*cos(alp1),-cos(t1)*sin(alp1),a1*sin(t1)],
[0,sin(alp1),cos(alp1),d1],[0,0,0,1]])
```

$$\begin{pmatrix} \cos(t1) & \sigma_2 \sin(t1) & -\sigma_1 \sin(t1) & 0 \\ \sin(t1) & -\sigma_2 \cos(t1) & \sigma_1 \cos(t1) & 0 \\ 0 & -\sigma_1 & -\sigma_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\sigma_1 = \sin\left(\frac{7\pi}{18}\right)$$

$$\sigma_2 = \cos\left(\frac{7\pi}{18}\right)$$

```
A12:=matrix([[cos(t2), -sin(t2)*cos(alp2), sin(t2)*sin(alp2), a2*cos(t2)],
[ sin(t2), cos(t2)*cos(alp2), -cos(t2)*sin(alp2), a2*sin(t2)],
[0, sin(alp2), cos(alp2), d2], [0, 0, 0, 1]])
```

$$\begin{pmatrix} \cos(t2) & 0 & \sin(t2) & 10 \cos(t2) \\ \sin(t2) & 0 & -\cos(t2) & 10 \sin(t2) \\ 0 & 1 & 0 & 278.5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
A23:=matrix([[cos(t3), -sin(t3)*cos(alp3), sin(t3)*sin(alp3), a3*cos(t3)],
[ sin(t3), cos(t3)*cos(alp3), -cos(t3)*sin(alp3), a3*sin(t3)],
[0, sin(alp3), cos(alp3), d3], [0, 0, 0, 1]])
```

$$\begin{pmatrix} \cos(t3) & 0 & -\sin(t3) & 0 \\ \sin(t3) & 0 & \cos(t3) & 0 \\ 0 & -1 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
A34:=matrix([[cos(t4), -sin(t4)*cos(alp4), sin(t4)*sin(alp4), a4*cos(t4)],
[ sin(t4), cos(t4)*cos(alp4), -cos(t4)*sin(alp4), a4*sin(t4)],
[0, sin(alp4), cos(alp4), d4], [0, 0, 0, 1]])
```

$$\begin{pmatrix} \cos(t4) & 0 & \sin(t4) & -26 \cos(t4) \\ \sin(t4) & 0 & -\cos(t4) & -26 \sin(t4) \\ 0 & 1 & 0 & 240 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
A45:=matrix([[cos(t5), -sin(t5)*cos(alp5), sin(t5)*sin(alp5), a5*cos(t5)],
[ sin(t5), cos(t5)*cos(alp5), -cos(t5)*sin(alp5), a5*sin(t5)],
[0, sin(alp5), cos(alp5), d5], [0, 0, 0, 1]])
```

$$\begin{pmatrix} \cos(t5) & 0 & -\sin(t5) & 120 \cos(t5) \\ \sin(t5) & 0 & \cos(t5) & 120 \sin(t5) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T05:=A01*A12*A23*A34*A45
```

$$\begin{aligned} & [[-\cos(t1) \sigma_{13} - \sigma_2 \sin(t1) \sigma_9 - \sigma_1 \sin(t1) \sigma_4, \sigma_2 \sin(t1) \sigma_5 - \cos(t1) \sigma_6 \\ & + \sigma_1 \sin(t1) \sin(t3) \sin(t4), -\cos(t1) \sigma_{12} - \sigma_2 \sin(t1) \sigma_8 - \sigma_1 \sin(t1) \sigma_3, \cos(t1) \sigma_{10} - \sigma_2 \sin(t1) \sigma_7 \\ & - \sigma_1 \sin(t1) \sigma_{11}], [\sigma_2 \cos(t1) \sigma_9 - \sin(t1) \sigma_{13} + \sigma_1 \cos(t1) \sigma_4, -\sin(t1) \sigma_6 - \sigma_2 \cos(t1) \sigma_5 \\ & - \sigma_1 \cos(t1) \sin(t3) \sin(t4), \sigma_2 \cos(t1) \sigma_8 - \sin(t1) \sigma_{12} + \sigma_1 \cos(t1) \sigma_3, \sin(t1) \sigma_{10} + \sigma_2 \cos(t1) \sigma_7 \end{aligned}$$

$$+ \sigma_1 \cos(t1) \sigma_{11}], [\sigma_1 \sigma_9 - \sigma_2 \sigma_4, \sigma_2 \sin(t3) \sin(t4) - \sigma_1 \sigma_5, \sigma_1 \sigma_8 - \sigma_2 \sigma_3, \sigma_1 \sigma_7 - \sigma_2 \sigma_{11}],$$

$$[0, 0, 0, 1]]$$

where

$$\sigma_1 = \sin\left(\frac{7\pi}{18}\right)$$

$$\sigma_2 = \cos\left(\frac{7\pi}{18}\right)$$

$$\sigma_3 = \cos(t3) \cos(t5) - \cos(t4) \sin(t3) \sin(t5)$$

$$\sigma_4 = \cos(t3) \sin(t5) + \cos(t4) \cos(t5) \sin(t3)$$

$$\sigma_5 = \cos(t2) \cos(t4) - \cos(t3) \sin(t2) \sin(t4)$$

$$\sigma_6 = \cos(t4) \sin(t2) + \cos(t2) \cos(t3) \sin(t4)$$

$$\sigma_7 = \sin(t2) \sigma_{14} - 10 \sin(t2) + \cos(t2) \sigma_{15}$$

$$\sigma_8 = \sin(t2) \sigma_{16} + \cos(t2) \sin(t4) \sin(t5)$$

$$\sigma_9 = \sin(t2) \sigma_{17} - \cos(t2) \cos(t5) \sin(t4)$$

$$\sigma_{10} = 10 \cos(t2) + \sin(t2) \sigma_{15} - \cos(t2) \sigma_{14}$$

$$\sigma_{11} = \cos(t3) \sigma_{19} - \sin(t3) \sigma_{18} + 278.5$$

$$\sigma_{12} = \cos(t2) \sigma_{16} - \sin(t2) \sin(t4) \sin(t5)$$

$$\sigma_{13} = \cos(t2) \sigma_{17} + \cos(t5) \sin(t2) \sin(t4)$$

$$\sigma_{14} = \cos(t3) \sigma_{18} + \sin(t3) \sigma_{19}$$

$$\sigma_{15} = 26 \sin(t4) - 120 \cos(t5) \sin(t4) + 10$$

$$\sigma_{16} = \cos(t5) \sin(t3) + \cos(t3) \cos(t4) \sin(t5)$$

$$\sigma_{17} = \sin(t3) \sin(t5) - \cos(t3) \cos(t4) \cos(t5)$$

$$\sigma_{18} = 26 \cos(t4) - 120 \cos(t4) \cos(t5)$$

$$\sigma_{19} = 120 \sin(t5) + 240$$

```
WRITEPATH := "C:\\Users\\Darius\\Desktop\\XPC_Orthosis_2"
"C:\\Users\\Darius\\Desktop\\XPC_Orthosis_2"
```

```
Pos:= T05 [1..3,4]:
```

```
generate::MATLAB(Pos)
```

```
" Pos = zeros(3,1); Pos(1,1) = cos(t1)*(cos(t2)*1.0e1+sin(t2)*(sin(t4)*2.6e1-cos(t5)*sin(t4)*1.2e2+1.0e1)-cos(t2)*(cos(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+sin(t3)*(sin(t5)*1.2e2+2.4e2)))-cos(pi*(7.0/1.8e1))*sin(t1)*(sin(t2)*-1.0e1+sin(t2)*(cos(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+sin(t3)*(sin(t5)*1.2e2+2.4e2))+cos(t2)*(sin(t4)*2.6e1-cos(t5)*sin(t4)*1.2e2+1.0e1))-sin(pi*(7.0/1.8e1))*sin(t1)*(cos(t3)*(sin(t5)*1.2e2+2.4e2)-sin(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+2.785e2); Pos(2,1) = sin(t1)*(cos(t2)*1.0e1+sin(t2)*(sin(t4)*2.6e1-cos(t5)*sin(t4)*1.2e2+1.0e1)-cos(t2)*(cos(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+sin(t3)*(sin(t5)*1.2e2+2.4e2)))+cos(pi*(7.0/1.8e1))*cos(t1)*(cos(t3)*(sin(t5)*1.2e2+2.4e2)+cos(t2)*(sin(t4)*2.6e1-cos(t5)*sin(t4)*1.2e2+1.0e1))+sin(pi*(7.0/1.8e1))*cos(t1)*(cos(t3)*(sin(t5)*1.2e2+2.4e2)-sin(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+2.785e2); Pos(3,1) = sin(pi*(7.0/1.8e1))*sin(t2)*-1.0e1+sin(t2)*(cos(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+sin(t3)*(sin(t5)*1.2e2+2.4e2))+cos(t2)*(sin(t4)*2.6e1-cos(t5)*sin(t4)*1.2e2+1.0e1))-cos(pi*(7.0/1.8e1))*(cos(t3)*(sin(t5)*1.2e2+2.4e2)-sin(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+2.785e2); "
```

```
fprint(Unquoted, Text, "FK.m", generate::MATLAB(Pos)):
```

```
P := (T05 [1..3,4])
```

$$\begin{pmatrix} \cos(t1) \sigma_5 - \sigma_2 \sin(t1) \sigma_3 - \sigma_1 \sin(t1) \sigma_4 \\ \sin(t1) \sigma_5 + \sigma_2 \cos(t1) \sigma_3 + \sigma_1 \cos(t1) \sigma_4 \\ \sigma_1 \sigma_3 - \sigma_2 \sigma_4 \end{pmatrix}$$

where

$$\sigma_1 = \sin\left(\frac{7\pi}{18}\right)$$

$$\sigma_2 = \cos\left(\frac{7\pi}{18}\right)$$

$$\sigma_3 = \sin(t2) \sigma_6 - 10 \sin(t2) + \cos(t2) \sigma_7$$

$$\sigma_4 = \cos(t_3) (120 \sin(t_5) + 240) - \sin(t_3) \sigma_8 + 278.5$$

$$\sigma_5 = 10 \cos(t_2) + \sin(t_2) \sigma_7 - \cos(t_2) \sigma_6$$

$$\sigma_6 = \cos(t_3) \sigma_8 + \sin(t_3) (120 \sin(t_5) + 240)$$

$$\sigma_7 = 26 \sin(t_4) - 120 \cos(t_5) \sin(t_4) + 10$$

$$\sigma_8 = 26 \cos(t_4) - 120 \cos(t_4) \cos(t_5)$$

`J := linalg::jacobian(P, [t1, t2, t3])`

$$\begin{aligned} & \left[\left[-\sin(t_1) \sigma_3 - \sigma_1 \cos(t_1) \sigma_4 - \sigma_2 \cos(t_1) \sigma_6, \cos(t_1) \sigma_4 + \sigma_1 \sin(t_1) \sigma_3, \sigma_2 \sin(t_1) \sigma_7 \right. \right. \\ & \quad \left. \left. - \cos(t_1) \cos(t_2) \sigma_5 - \sigma_1 \sin(t_1) \sin(t_2) \sigma_5 \right], \right. \\ & \left[\cos(t_1) \sigma_3 - \sigma_1 \sin(t_1) \sigma_4 - \sigma_2 \sin(t_1) \sigma_6, \sin(t_1) \sigma_4 - \sigma_1 \cos(t_1) \sigma_3, \sigma_1 \cos(t_1) \sin(t_2) \sigma_5 \right. \\ & \quad \left. - \sigma_2 \cos(t_1) \sigma_7 - \cos(t_2) \sin(t_1) \sigma_5 \right], \\ & \left[0, -\sigma_2 \sigma_3, \sigma_1 \sigma_7 + \sigma_2 \sin(t_2) \sigma_5 \right] \end{aligned}$$

where

$$\sigma_1 = \cos\left(\frac{7\pi}{18}\right)$$

$$\sigma_2 = \sin\left(\frac{7\pi}{18}\right)$$

$$\sigma_3 = 10 \cos(t_2) + \sin(t_2) \sigma_8 - \cos(t_2) \sigma_7$$

$$\sigma_4 = \sin(t_2) \sigma_7 - 10 \sin(t_2) + \cos(t_2) \sigma_8$$

$$\sigma_5 = \cos(t_3) \sigma_{10} - \sin(t_3) \sigma_9$$

$$\sigma_6 = \cos(t_3) \sigma_{10} - \sin(t_3) \sigma_9 + 278.5$$

$$\sigma_7 = \cos(t_3) \sigma_9 + \sin(t_3) \sigma_{10}$$

$$\sigma_8 = 26 \sin(t_4) - 120 \cos(t_5) \sin(t_4) + 10$$

$$\sigma_8 = 26 \sin(t_4) - 120 \cos(t_5) \sin(t_4) + 10$$

$$\sigma_9 = 26 \cos(t_4) - 120 \cos(t_4) \cos(t_5)$$

$$\sigma_{10} = 120 \sin(t_5) + 240$$

```
generate::MATLAB(J)
```

```
" J = zeros(3,3); J(1,1) = -sin(t1)*(cos(t2)*1.0e1+sin(t2)*(sin(t4)*2.6e1-cos(t5)*sin(t4)*1.2e2+1.0e\
1)-cos(t2)*(cos(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+sin(t3)*(sin(t5)*1.2e2+2.4e2)))-cos(pi*(\
7.0/1.8e1))*cos(t1)*(sin(t2)*-1.0e1+sin(t2)*(cos(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+sin(t3)\
*(sin(t5)*1.2e2+2.4e2))+cos(t2)*(sin(t4)*2.6e1-cos(t5)*sin(t4)*1.2e2+1.0e1))-sin(pi*(7.0/1.8e1))*c\
os(t1)*(cos(t3)*(sin(t5)*1.2e2+2.4e2)-sin(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+2.785e2); J(1\
,2) = cos(t1)*(sin(t2)*-1.0e1+sin(t2)*(cos(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+sin(t3)*(sin(t5)\
)*1.2e2+2.4e2))+cos(t2)*(sin(t4)*2.6e1-cos(t5)*sin(t4)*1.2e2+1.0e1))+cos(pi*(7.0/1.8e1))*sin(t1)*\
(cos(t2)*1.0e1+sin(t2)*(sin(t4)*2.6e1-cos(t5)*sin(t4)*1.2e2+1.0e1)-cos(t2)*(cos(t3)*(cos(t4)*2.6e1\
-cos(t4)*cos(t5)*1.2e2)+sin(t3)*(sin(t5)*1.2e2+2.4e2))); J(1,3) = -cos(t1)*cos(t2)*(cos(t3)*(sin(t5)\
)*1.2e2+2.4e2)-sin(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2))+sin(pi*(7.0/1.8e1))*sin(t1)*(cos(t3)\
*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+sin(t3)*(sin(t5)*1.2e2+2.4e2))-cos(pi*(7.0/1.8e1))*sin(t1)*\
sin(t2)*(cos(t3)*(sin(t5)*1.2e2+2.4e2)-sin(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)); J(2,1) = cos\
(t1)*(cos(t2)*1.0e1+sin(t2)*(sin(t4)*2.6e1-cos(t5)*sin(t4)*1.2e2+1.0e1)-cos(t2)*(cos(t3)*(cos(t4)*\
2.6e1-cos(t4)*cos(t5)*1.2e2)+sin(t3)*(sin(t5)*1.2e2+2.4e2)))-cos(pi*(7.0/1.8e1))*sin(t1)*(sin(t2)*\
-1.0e1+sin(t2)*(cos(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+sin(t3)*(sin(t5)*1.2e2+2.4e2))+cos(t\
2)*(sin(t4)*2.6e1-cos(t5)*sin(t4)*1.2e2+1.0e1))-sin(pi*(7.0/1.8e1))*sin(t1)*(cos(t3)*(sin(t5)*1.2e2\
+2.4e2)-sin(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+2.785e2); J(2,2) = sin(t1)*(sin(t2)*-1.0e1+\
sin(t2)*(cos(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+sin(t3)*(sin(t5)*1.2e2+2.4e2))+cos(t2)*(sin\
(t4)*2.6e1-cos(t5)*sin(t4)*1.2e2+1.0e1))-cos(pi*(7.0/1.8e1))*cos(t1)*(cos(t2)*1.0e1+sin(t2)*(sin(t\
4)*2.6e1-cos(t5)*sin(t4)*1.2e2+1.0e1)-cos(t2)*(cos(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)+sin(\
t3)*(sin(t5)*1.2e2+2.4e2))); J(2,3) = -cos(t2)*sin(t1)*(cos(t3)*(sin(t5)*1.2e2+2.4e2)-sin(t3)*(cos(t\
4)*2.6e1-cos(t4)*cos(t5)*1.2e2))-sin(pi*(7.0/1.8e1))*cos(t1)*(cos(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t\
5)*1.2e2)+sin(t3)*(sin(t5)*1.2e2+2.4e2))+cos(pi*(7.0/1.8e1))*cos(t1)*sin(t2)*(cos(t3)*(sin(t5)*1.2\
e2+2.4e2)-sin(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)); J(3,2) = -sin(pi*(7.0/1.8e1))*(cos(t2)*1.\
0e1+sin(t2)*(sin(t4)*2.6e1-cos(t5)*sin(t4)*1.2e2+1.0e1)-cos(t2)*(cos(t3)*(cos(t4)*2.6e1-cos(t4)*c\
os(t5)*1.2e2)+sin(t3)*(sin(t5)*1.2e2+2.4e2))); J(3,3) = cos(pi*(7.0/1.8e1))*(cos(t3)*(cos(t4)*2.6e\
1-cos(t4)*cos(t5)*1.2e2)+sin(t3)*(sin(t5)*1.2e2+2.4e2))+sin(pi*(7.0/1.8e1))*sin(t2)*(cos(t3)*(sin(t\
5)*1.2e2+2.4e2)-sin(t3)*(cos(t4)*2.6e1-cos(t4)*cos(t5)*1.2e2)); "
```

```
WRITEPATH := "C:\\Users\\Darius\\Desktop\\XPC_Orthosis_2"
```

```
"C:\Users\Darius\Desktop\XPC_Orthosis_2"
```

```
fprint(Unquoted, Text, "IK.m", generate::MATLAB(J)):
```

Appendix B– Flex Shaft Apparatus

B.1 Test apparatus

The test apparatus was developed to investigate the behaviour of the UBC-PULO flex shafts as explained in Section 3.2.1.3. The LabVIEW program communicates with the NI PCI 6221 DAQ card to send /receive the required analog and digital I/O - this enables the system to control the motor and acquire the measured step response data. The real-time angular position information from the two optical encoders, the torque-sensor measurements and the motor voltage are acquired by the system and recorded in tabulated format for post-analysis via Excel or MATLAB.

Figure 64 shows the block diagram for the test apparatus's software developed using LabVIEW. The diagram has been divided into numbered sub-sections using red dashed boxes. The function of each of these sub-sections with respect to each box number is listed below:

1. This sub-section is responsible for the first optical encoder acquisition setup and utilizes one of the DAQ card's counter input. The angular position is determined by performing X4 decoding of the optical encoder's quadrature signal.
2. This sub-section's function is the same as box 1 repeated for the second optical encoder (i.e. the one close to the torque sensor).
3. This sub-section is responsible for acquiring and interpreting the two analog input voltages received for the torque sensor and the motor voltage input.
4. This sub-section is responsible for sending the square wave signal of desired magnitude and re-occurrence frequency to the motor driver board utilizing one of

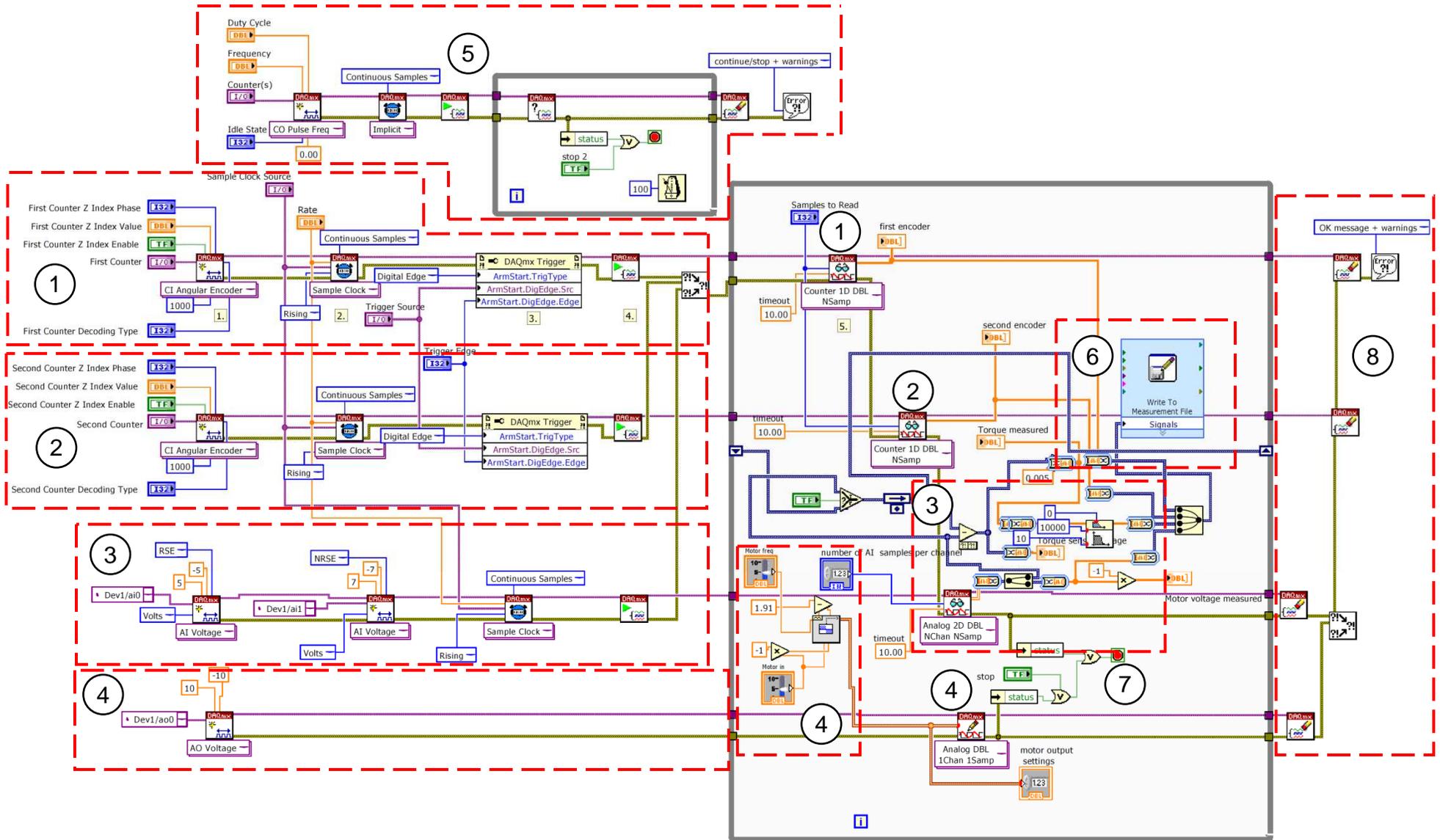


Figure 64. Test Apparatus LabVIEW Diagram

the DAQ card's analog out channel. The motor driver board amplifies this signal and drives the motor.

5. This sub-section is responsible for generating a external high-frequency clock using the DAQ cards *DEV1 frequency output* channel. This is a global signal required for synchronizing the acquisition of the various types of I/O used by the test apparatus (i.e. Digital inputs, counter inputs, analog inputs and analog outputs).
6. This sub-section is responsible for recording all the acquired data in a tabulated ".LVM" format.
7. This sub-section is responsible for stopping the software in case of an error or if the user presses the *STOP* button.
8. This sub-section clears the temporarily system variables and resets the DAQ card once the acquisition is terminated (i.e. user clicks on *STOP*).

B.2 Flex shaft casing solutions

The use of flex with casing is highly recommended to suppress non-axial rotational movements and vibrations throughout the length of the shaft on the UBC-PULO. Research work was performed to narrow down a list of manufacturer's that sell suitable flex shaft. The recommended manufacturers are listed in Table 25:

Table 25. Vendors for Flex Shaft with Casings

<i>Company</i>	<i>Description</i>
S.S. White Technologies	<p>The Ready-Flex® standard shafts are available with casings and come with either Ball Bearing or Oilite Bearing couplings. Custom solutions are available.</p> <p>Website: http://www.sswt.com/standard.htm</p>
HASPA GmbH	<p>The shafts are available with metal, rubber, or plastic tube type casings. Custom solutions are available.</p> <p>Website :http://www.haspa-gmbh.de/Main.aspx?lang=en-US</p>
Elliott Manufacturing	<p>The shafts are similar to the ones currently used on the prototype. Further inquiries need to be made.</p> <p>Website: http://www.elliottmfg.com/index.php?id=3</p>

Appendix C– Further Assessment of the UBC-PULO

C.1 Shoulder

Currently, the UBC-PULO prototype's makes use of the one degree-of-freedom shoulder azimuth rotation mechanism to perform adduction and abduction of the users shoulder. However, the shoulder involves other complex motions, such as horizontal and vertical translations, during horizontal adduction/abduction [46]. Additionally, the prototype's azimuth rotation axis is currently offset from the user's shoulder axis. The complex shoulder motions and the existence of an offset in the azimuth axis pose a problem in terms of user comfort, ergonomics and range of motion.

The shoulder elevation was originally omitted from the current orthosis for reducing complexity of the mechanical design, simplifying the control algorithm and optimizing the overall weight [2]. One of the main arguments provided for eliminating shoulder elevation was that adding this DOF would introduce extra bulk and increase power consumption due to motor torque requirements that are 3 times higher than any other joint [2]. Unfortunately this research work did properly identify the fact that fixing the shoulder elevation would adversely affect some relatively less important tasks such as "pouring from a pitcher, reaching for a doorknob, flipping a light switch from a seated position or combing hair." Furthermore, fixing the shoulder elevation would cause the UBC-PULO to produce less natural arm movements. Thus, exclusion of the shoulder elevation limits the user in performing a range of every-day tasks and causes less natural arm motion and further research with respect to reinstating the shoulder DOF is recommended.

Harmonic gears (also known as strain wave gears) are a good candidate for the shoulder elevation redesign as they are light, compact, have no backlash, and yet come with high gear ratios capable of handling high torque loads [47]. Figure 65 shows a sample harmonic actuator and harmonic gear. According to [47], high gear reduction ratios are possible in a small volume - a ratio of 100:1 is possible in the same space in which planetary gears typically only produce a 10:1 ratio. For this reason, harmonic gears are typically used in many robotics and space applications and have also been used for Hal-5's (Section 2.2) lower limb actuator. Using harmonic gears it is possible to restore the shoulder elevation DOF to the user's upper limb, and produce more natural arm motion with little added cost (i.e. weight and power consumption); therefore, future UBC-PULO prototypes or shoulder redesigns should investigate this idea. In addition, a future design needs to incorporate an axis of rotation that coincides with the user's shoulder, and would ideally accommodate the complex translational shoulder joint movements as well.

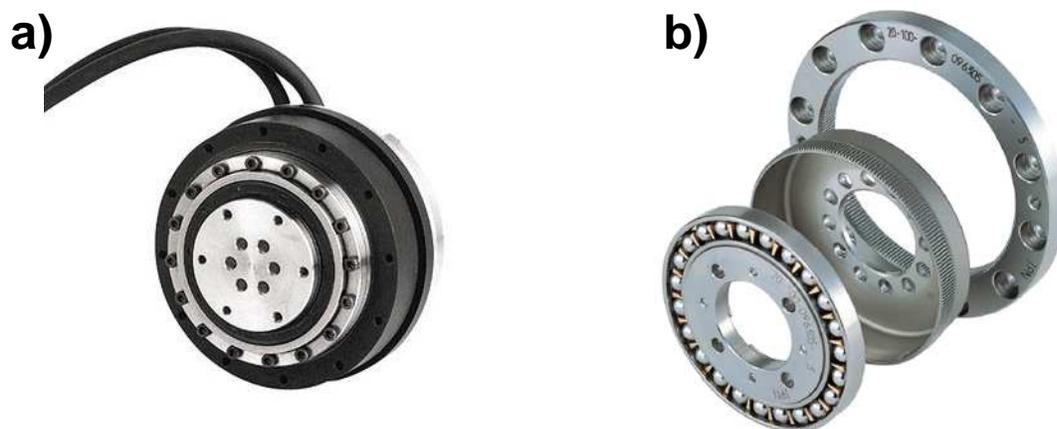


Figure 65. a) Harmonic Actuator [48] b) Harmonic Gear (Strain Wave Gearing) [49]

C.2 Forearm

The UBC-PULO prototype's forearm axis of rotation is offset from the user's forearm axis of rotation [1], thus twisting the forearm and limiting its rotational range-of-motion. This issue needs to be addressed in the future. A finalized design, including final sketches, a bill of materials (BOM) and a manufacturing method has been proposed in reference [50] and Appendix C5.

C.3 Grasp

As illustrated in Figure 66, currently the opening and closing of the grasp is activated by the grasp joint, which moves the user's fingers (fixed inside the splinting mold designated by label 1) relative to their thumb (designated by label 2). This mechanism limits the UBC-PULO grasp to a limited number of object shapes. A good grasping mechanism would involve the hand fitting the shape of the part of object being held as grabbing a book, glass of water, door handle or a spoon is quite different. In the future, improvements to the grasp mechanism are strongly recommended.

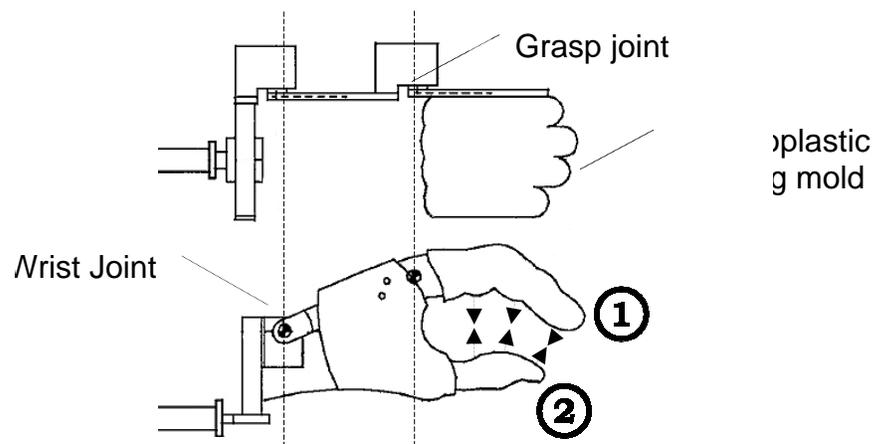


Figure 66. Grasp Mechanism

C.4 Weight and comfort

Ideally the overall device should be as light as possible with its center-of-gravity as close as possible to the middle of user`s lower waist so that it can comfortably sit on their hip without fatiguing back muscles. As shown in Figure 67, the center-of-gravity of the device is offset from the ideal case mainly due to the weight and positioning of the exoskeleton, electrical box, battery and drive system. While reducing the overall UBC-PULO weight is a long term goal, the distribution of various components should also be adjusted to move the center-of-gravity to the ideal center of gravity location as presented in Figure 67. For example, the battery could be moved to the lower left corner (top-view) of the device to balance out the weight of the exoskeleton. Optimizing the exoskeleton weight and its distribution through material and component selection also contributes to moving the center of gravity to a desirable position.

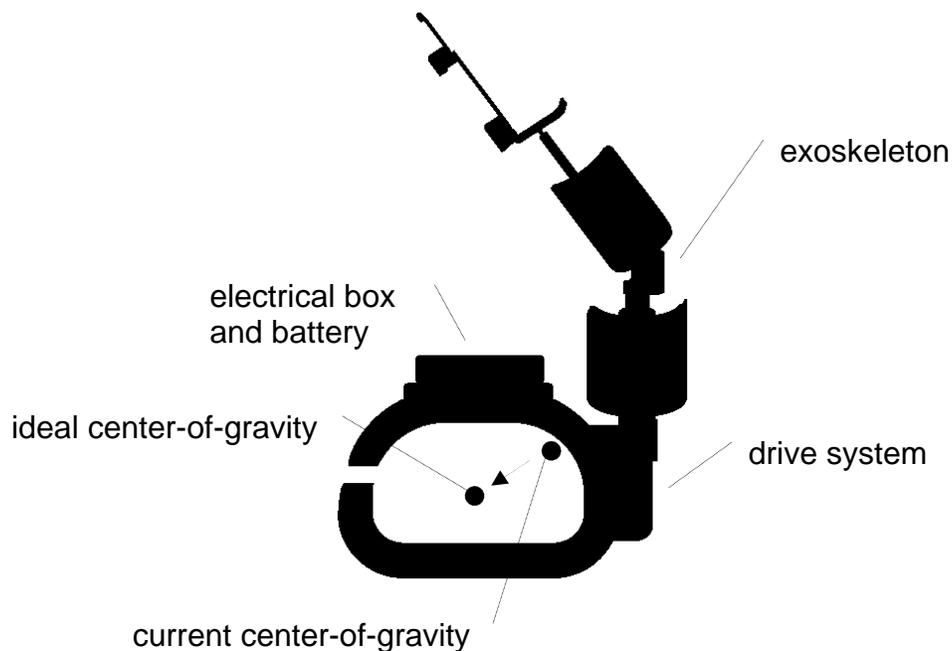
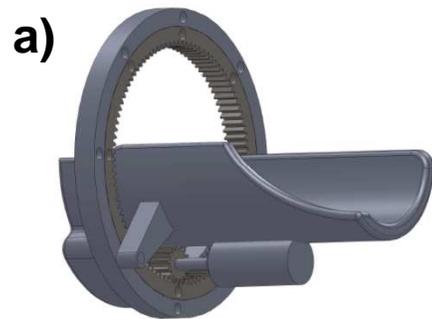


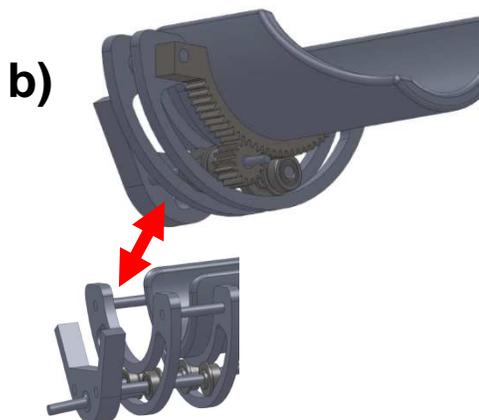
Figure 67. UBC-PULO Prototype Top-view

C.5 Forearm redesign

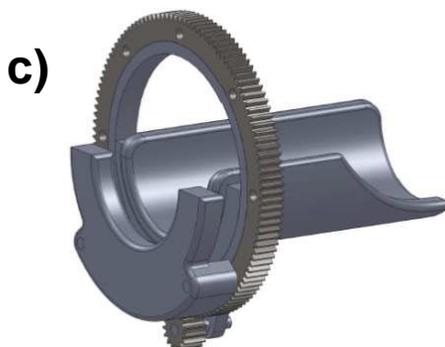
Possible forearm mechanics redesigns concepts for the UBC-PULO were considered as illustrated in Figure 68. However, these designs concepts are bulky and intrusive to the user and alternative solutions were explored.



The slewing bearing with internal gears has the same axis of rotation as the forearm can be driven by the flex shafts attached to a spur gear. The wrist and grasp mechanics can be rigidly supported by multiple attachment points on the slewing bearing.



The ball bearing on track mechanism provides guidance for rotation about the forearm axis of rotations and provides support for the attached forearm, wrist and grasp exoskeleton. The flex shaft is attached to a spur gear sitting on a geared track, driving the guided ball bearing mechanism.



This forearm mechanism is similar to design concept shown in part “a” with the difference of incorporating external gears. This design is slightly less bulky as a smaller slewing bearing mechanism is used.

Figure 68 Considered Forearm Redesign Concepts

A forearm design has been proposed in reference [50], where a plastic base is allowed to slide on a thin steel guide track to provide forearm rotation (see Figure 69). The design utilizes a pulley system to drive the plastic base about the forearm axis of rotation by having a steel cable wound around the drive pulley that is attached to either end of the existing forearm brace. Use of PTFE-Filled Delrin plastic for the base and use of steel for the guide track as these materials have low sliding friction while also providing the required rigidity. The pulley system also provides approximately a 30:1 gear ratio and, therefore, allows for the elimination of the current gear box. The main advantages of this design are its overall compactness, lightness, ease of manufacturing, low cost and the elimination of gearbox. More details about the design are presented in reference [50].

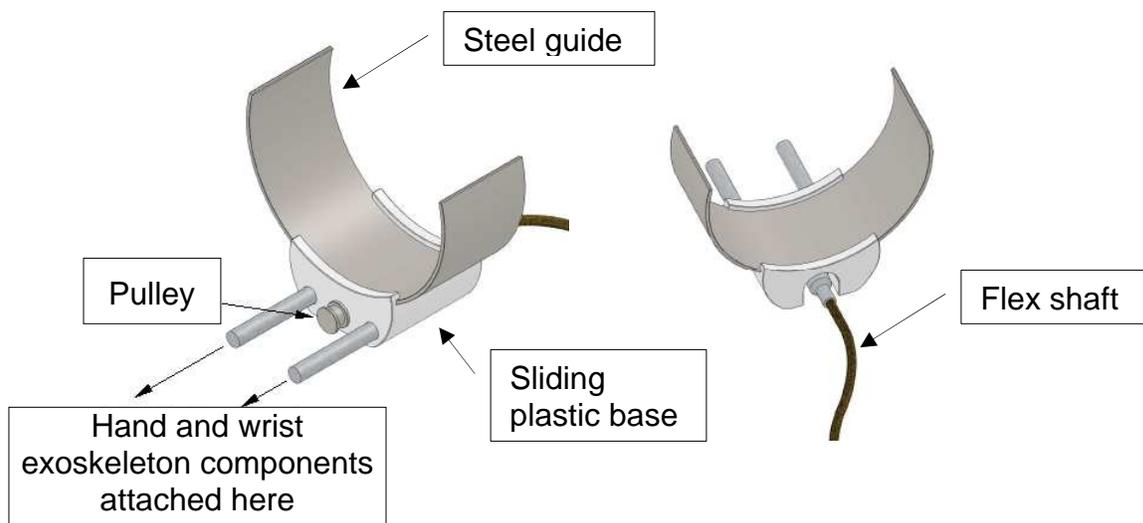


Figure 69 Proposed Forearm Design

C.6 Prioritized list of mechanical improvements

Table 9 provides a prioritized list of improvements, with the high-priority items being those needing immediate attention for proper commissioning of the prototype system. Medium-priority items limit the UBC-PULO operation in terms of controllability, range of motion and manoeuvrability during execution of daily task activities. Medium-priority items need to be improved in the near future. Low-priority improvements do not affect the functionality of the device, but should be long term goals as they affect the feasibility of the UBC-PULO.

Table 26. Prioritized List of Mechanical Improvements

Priority	Description
High	<ul style="list-style-type: none"> • Fix elbow joint
Medium	<ul style="list-style-type: none"> • Redesign forearm mechanism • Redesign shoulder mechanism • Redesign grasping mechanism • Find and Implement flex-shaft casing solutions • Redesign elbow joint • Improve shoulder roll torque performance • Redesign azimuth joints to be more robust against friction when under load
Low	<ul style="list-style-type: none"> • Improve user comfort and reduce weight • Implementing mechanical safety limits

C.7 Prioritized list of electrical improvements

Table 27 presents the prioritized list of improvements that was created based on this examination, with the high-priority items being the ones needing immediate attention for commissioning the existing prototype system. Items needing attention in the near future are listed as medium-priority as they affect the performance and feasibility of the UBC-PULO. Low-priority items need to be carried out in the far-future (i.e. long term goal) as it is most effective to implement them during the final prototyping stages or during the product re-development cycle. This ensures that all design factors are known before investing in the development of a highly customized electrical system. Items on the medium-priority list need to be implemented earlier-on than low-priority items as they can be tested and optimized during the prototyping cycle.

Table 27. Prioritized List of Electrical Improvements

Priority	Description
High	<ul style="list-style-type: none">• Redesign UBC-PULO electrical system
Medium	<ul style="list-style-type: none">• Redesign angle sensors• Redesign head interface• Redesign shoulder interface• Power optimization of low level control board (motor driver board)
Low	<ul style="list-style-type: none">• Power optimization of high level board• Implementing power management routines in software• Optimize size and weight• Battery selection

Appendix D – Host and Target Computer Setup

D.1 Host computer

First MATLAB, XPC Target and a C compiler (note that MathWorks recommends Visual studio or Watcom) were installed on the host computer. Note that XPC Target 5.0 is only compatible with 32-bit operating systems; therefore, the 32-bit version of MATLAB has to be installed.

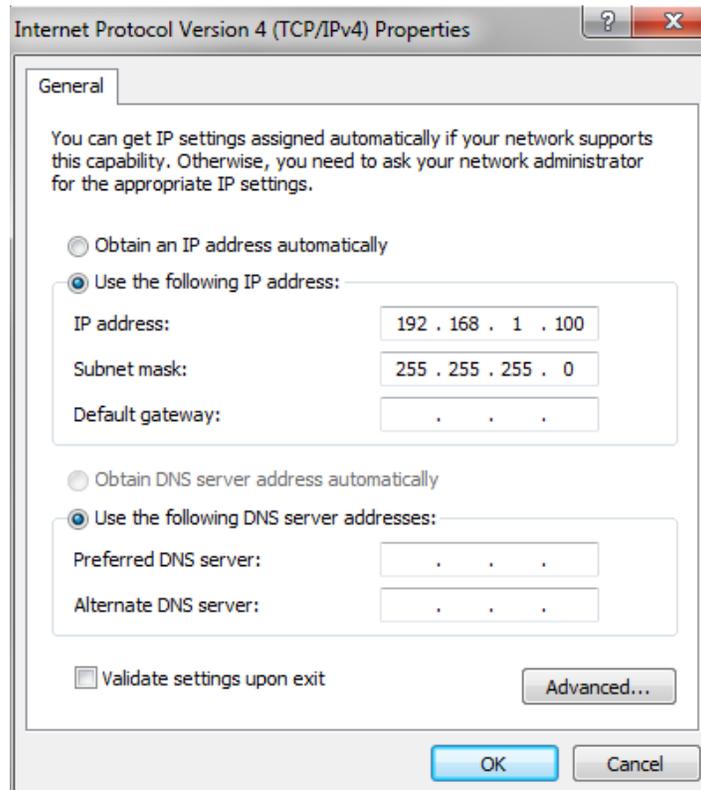
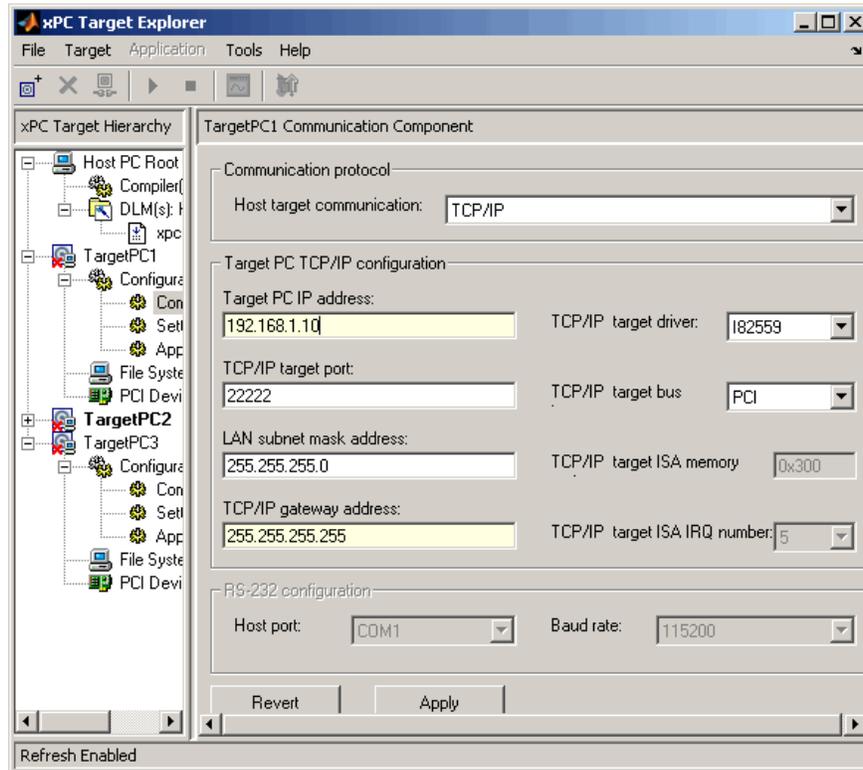


Figure 70. Host PC TCP/IP Address

Next, the network connection on the host computer has to be setup as static with an available arbitrary IP address as shown in Figure 70. The subnet mask has to be set to “255.255.255.0”. On Windows 7 this can be performed by going to Control Panel, Network and Internet, Network Connections and changing IP settings as shown in Figure 70.

Next, the XPC Explorer window was opened in MATLAB. Under the *XPC Target Hierarchy* the compiler configuration setting Visual C was selected in the “select C compiler” menu, and the “Compiler Pathway” was set to the path where Microsoft Visual Studio was installed. Subsequently, the TargetPC1's communication configuration, the settings were changed to the values shown in Figure 71-a. The “TCP/IP target driver” was set to I82559 because an Ethernet crossover cable was used to connect the host and target computer. As presented in Figure 71-b, after all the mentioned settings were applied, the “DOSloader” option was selected under the TargetPC1's configuration and a “bootDISK” was created on a USB memory device. The host computer was then ready to be connected to TargetPC1 (i.e. the UBC-PULO target computer) and the next step was to setup the target PC. For additional information refer to “XPC Target 5 Getting Started Guide” document on the MathWorks website.

a)



b)

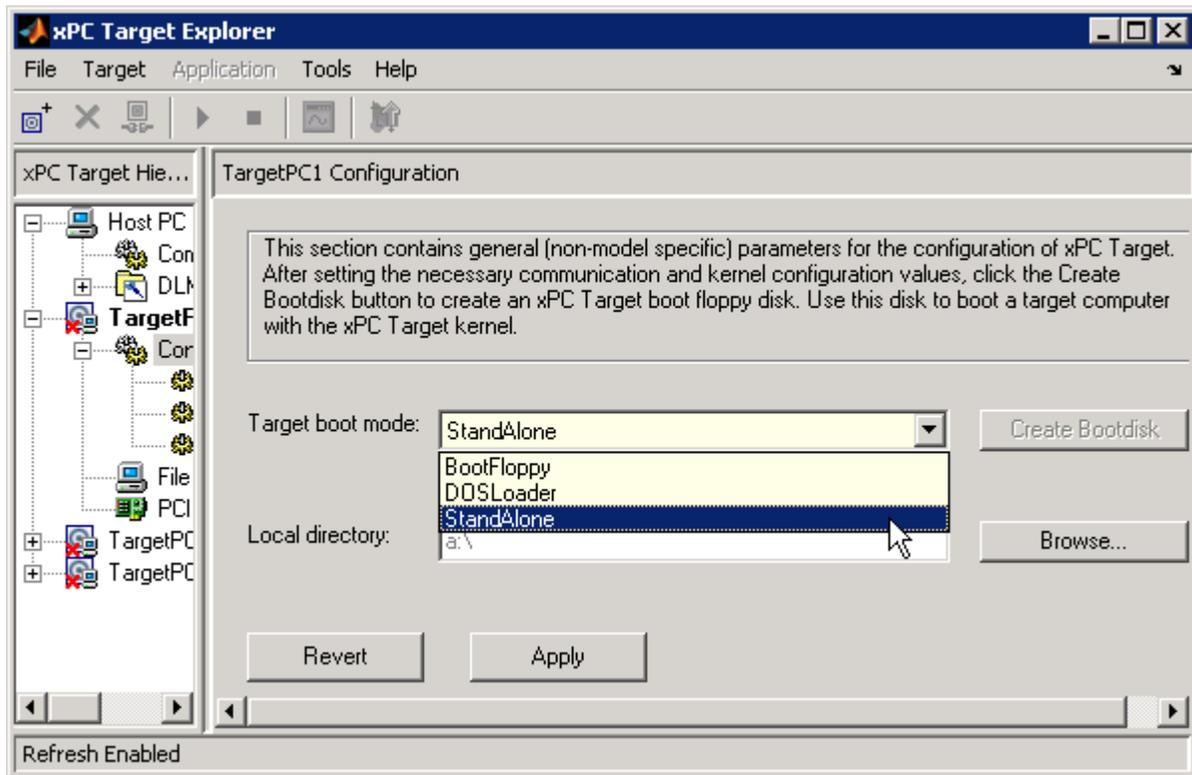


Figure 71. Screen Shot of XPC Explorer Window

D.2 Target computer

Figure 72 shows all the PCM 3362 SBC board connectors with Table 28 listing the important connectors and their functions. Before powering the unit, a 2 Giga-byte DDR2 RAM was pushed into the 200-pin SODIMMs socket. Using the provided adaptor cables the SBC board was connected to a monitor, keyboard and mouse.

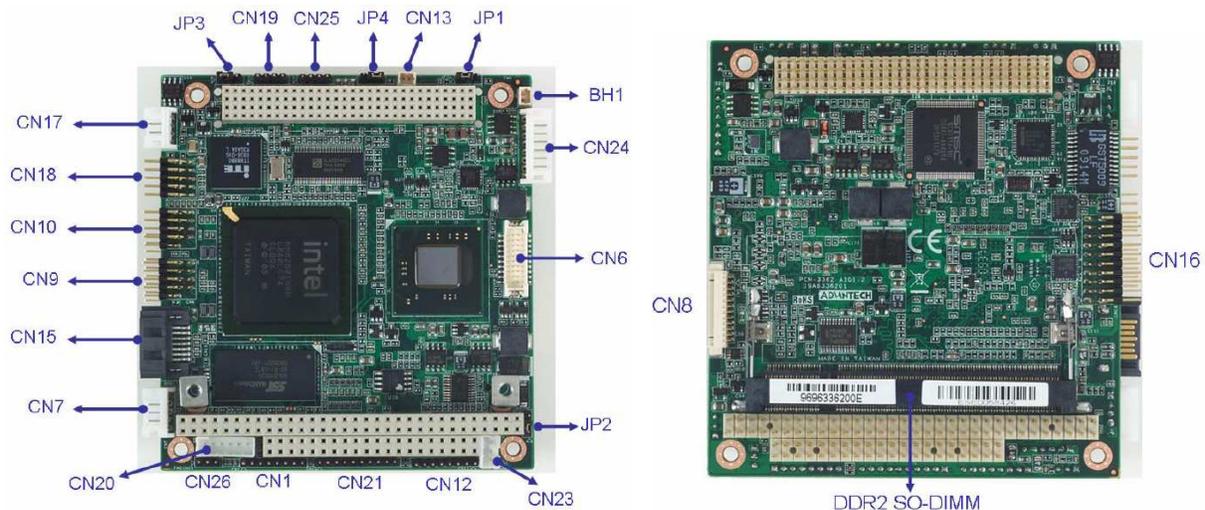


Figure 72. Advantech PCM 3362 Connectors with Component Side on Left and Back Side on Right (for More Details Visit Datasheet)

A standard ATX (Advanced Technology eXtended) power supply was used for the setup procedure (i.e. connected to CN19). An Ethernet crossover cable was connected to CN18 and the host computer. An external CD-ROM device was attached to the USB connection loaded with an Ubuntu bootable CD (i.e. Ubuntu is a free open source operating system).

Table 28. Important PCM 3362 Connectors

Connector	Function
DDR2 SO-DIMM	Small Outline Dual In-line Memory Modules (RAM connector)
CN8*	VGA connector
CN9* & CN10*	USB connectors
CN13	Buzzer connector
CN18	Giga LAN Connector
CN20*	Keyboard and PS/2 mouse connector
CN23	ATX Power in connector

* requires adaptor cable

Next, the device was powered on and the *delete* button on the keyboard was held while the system was booting up to enter the BIOS setup as shown in Figure 73. The following modifications were made to the default BIOS settings:

1. The “Hotplug USB FDD Support” option and “BIOS EHCI hand-off” option were enabled under the *Advanced > USB Configuration menu > Hotplug USB FDD Support* menu.
2. The “4 ports” option was selected under the *Chipset > South bridge chipset configuration > USB Function* menu.
3. The “USB 2.0 controller” was enabled under the *Chipset > South bridge chipset configuration > USB 2.0 controller* menu.
4. The BIOS setting were saved and the target computer was restarted.
5. The “Hotplug USB FDD” was option was selected for the “1st Boot Device” under the *Boot > Boot device priority > 1st Boot Device* menu.
6. The BIOS setting were saved and the target computer was restarted.

7. The bootable Ubuntu operating system CD was inserted in the external CD-ROM device and the target computer was restarted.
8. The system was rebooted and the Ubuntu operating system was loaded. Next, the USB memory device containing the “bootDISK” data was attached to the target computer’s USB plug and its contents were copied to C:\.
9. The target computer was restarted and the bootable DOS 6.22 CD was inserted into the external CD-ROM device.
10. The system rebooted and the DOS 6.22 operating system was loaded. The command “format C:/s” was executed. This installed the DOS 6.22 on the target computer which is a requirement for loading XPC Kernel operating system based on the “DOSloader” option.
11. The target computer was restarted. The system loaded the XPC Target 5.0 operating system ready to be connected to the host computer.
12. The USB memory device and External CD-ROM device were removed.

To optimize the real-time performance of the system the following BIOS setup changes were performed:

1. The “Power management” option was disabled under the *Advanced > AMP Configurations > Power Management* menu (i.e.If the power management option is enabled it will significantly affect the real-time performance of the system).
2. The “Legacy USB Support” option was disabled under the *Advanced > USB Configuration > Legacy USB Support* menu. The “USB 2.0 controller” and “USB Function” was disabled under the *Chipset > South Bridge Configuration* menu.

This disabled the USB chipsets which significantly enhance the real-time performance of the target computer.

3. The “No” option was selected under the *PCIPnP*> *Plug and Play OS* menu since the XPC Kernel 5.0 is not a PnP system.

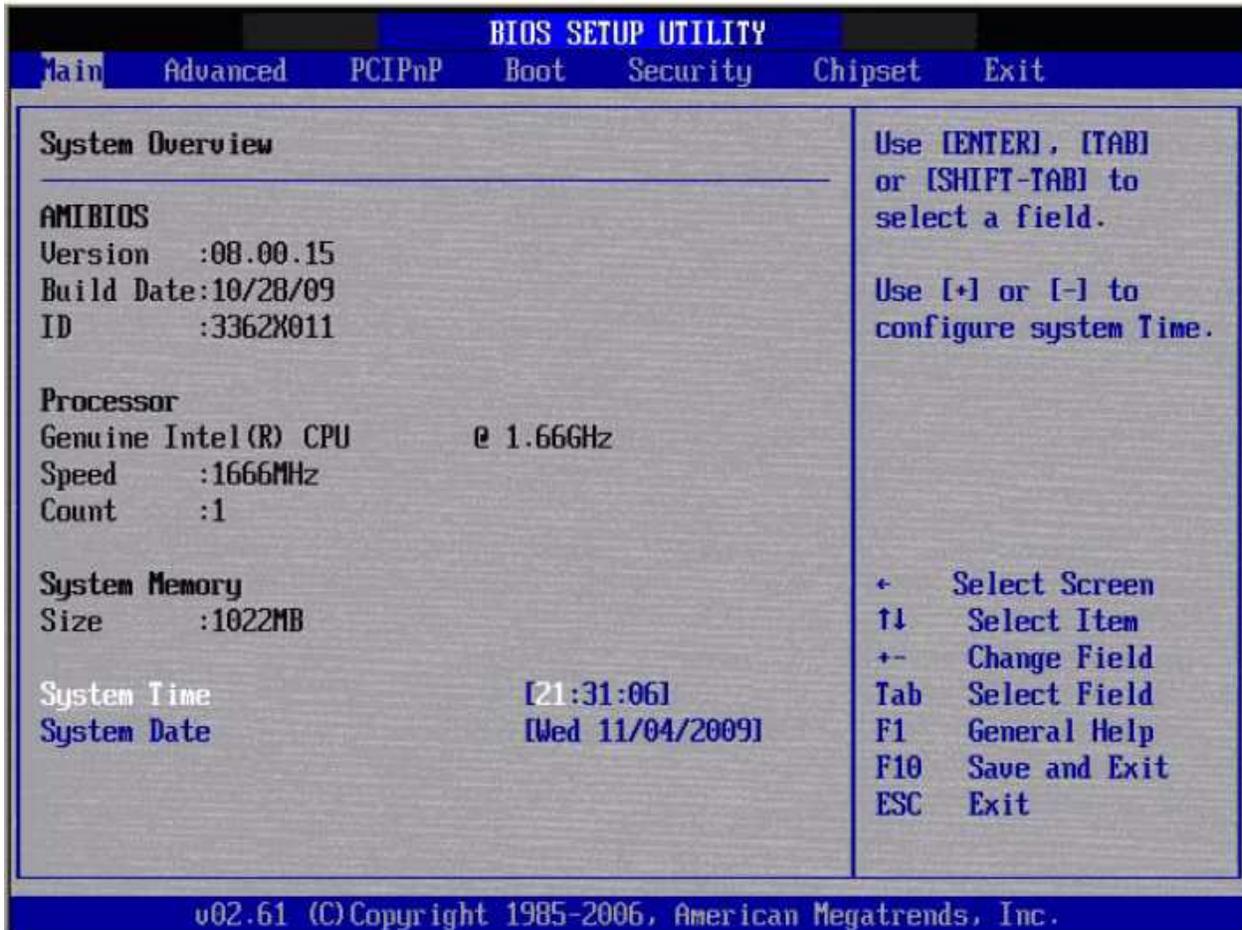


Figure 73. PCM 3362 BIOS Screen Shot

Additional information regarding the PCM 3362 board, such as jumper settings, is available in its datasheet. For more information about XPC Kernel and the various settings used in this work please refer to “XPC Target 5 Getting Started Guide” document on MathWorks website.

Appendix E – Custom PCB Design

E.1 Selected electrical components for custom PCB

This section elaborates on the features and properties of the selected electrical components for the custom PCB.

E.1.1 Motor driver IC

The Texas Instruments (TI) DRV8412 chip was chosen due to its compactness and high efficiency. This motor driver is only 8x16 mm in size, capable of driving two DC motors with efficiencies up to 97%. The DRV8412 is rated at 3 Amps of continuous output current per DC motor at up to 50 Volts (i.e. 150 Watt total output power). Due to its high efficiency and TI's PowerPad technology (see Figure 74), no external heat sink is required. The thermal pad is exposed at the bottom providing extremely low thermal resistance between the junction and the exterior of package. The PowerPad (i.e. thermal pad) can be soldered directly to the PCB using the PCB as a heat sink. Special heat sink designs can be incorporated on the opposing side of the PCB for optimal heat transfer. Additionally, the DRV8412 requires a small number of external components and incorporates internal self-protection circuits for protection against over-current, over-temperature and short-circuit faults.

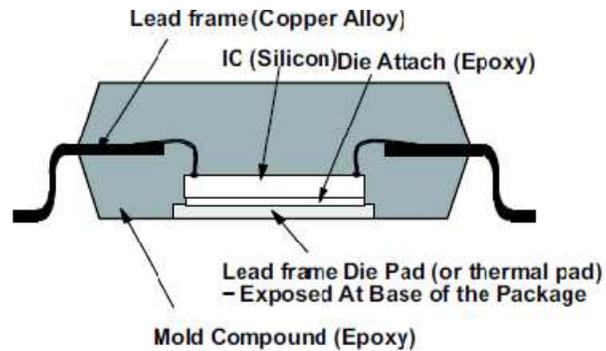


Figure 74. Texas Instrument's PowerPad Technology Allows Heat Dissipation from the Bottom of the IC.

Figure 75 shows a simplified diagram of this motor driver chip where all the gate drive voltage supply pins (i.e. GVDD_X) have to be connected to a 12V supply. The DRV8412 has various *output modes* which can be programmed by setting each M1, M2 or M3 by pin to a digital 1 (i.e. connected to VREG pin) or 0 (i.e. connected to AGND pin). The Over Temperature Warning pin (or \overline{OTW}) outputs a digital 0 if the junction temperature exceeds 125° and a digital 1 under normal conditions. Under normal conditions the \overline{FAULT} pin outputs digital 1; however, in case of output-short-circuit, over-current or a junction temperature exceeding 150°, the \overline{FAULT} pin outputs digital 0 and all motor outputs are shut down. So under normal condition both \overline{FAULT} and \overline{OTW} pins are in a digital 1 state. A programmable resistor connected between the OC_ADJ pin and AGND sets the desired Over Current Value. The DRV8412 receives four PWM signals (i.e. PWM_A, PWM_B, PWM_C and PWM_D) from the controller to drive the motors using PWM. Depending on the selected *output mode*, the four input PWM signals the can drive one, two or four motors.

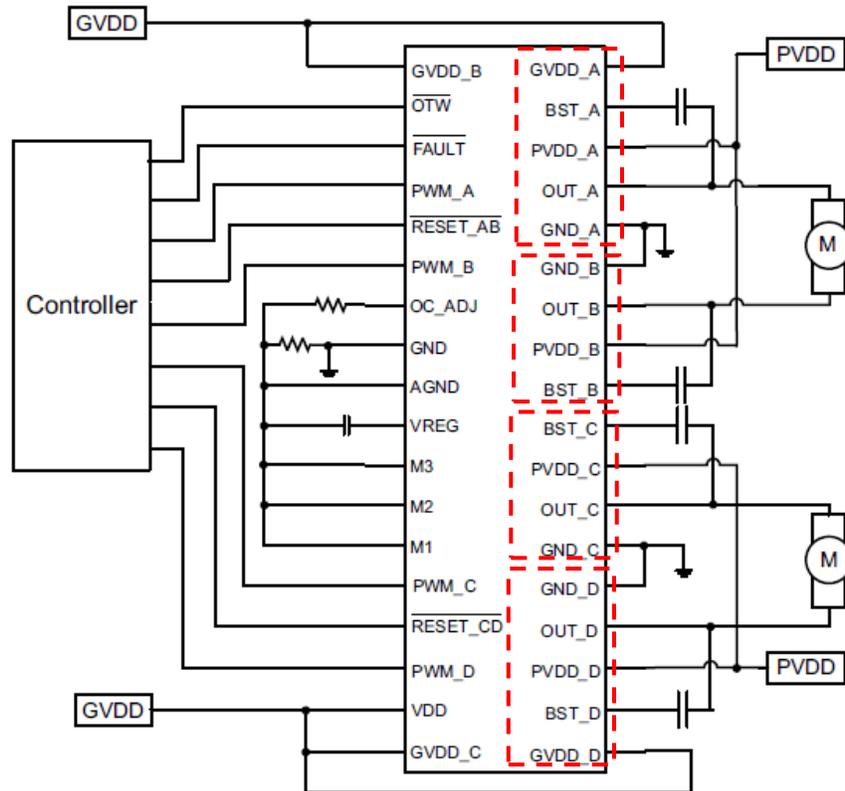


Figure 75. Simplified DRV8412 Connection Diagram with Output Stage Half-H-Bridges Marked in Red Boxes

The output stage of the DRV8412 chip is composed of four Half-H-bridge circuits internally. A Half-H-bridge can drive a motor in only one direction. To drive a motor in both directions, two Half-H-bridges are required. In this example, Half-H-bridge A and B drive the first motor and Half-H-bridge C and D drive the second motor. Figure 76 presents the basic principle of operation of the DRV8412s output stage. Each Half-H-bridge is essentially composed of two MOSFETs (Metal Oxide Semiconductor Field Effect transistor) controlled by the internal control signals. In simple terms, each MOSFET acts as a digital switch which can be turned ON or OFF. By setting the DRV8412 output configuration to *complementary Full Bridge* mode (i.e. M1=1, M2=1

and $M3=0$), MOSFET 1 and 4 will complement MOSFET 2 and 3. So, if MOSFET 1 and 4 are ON, MOSFET 2 and 3 will be OFF (i.e. *state A*) and vice versa (i.e. *state B*). As presented in Figure 76, the direction of the output current passing through the motor and thus direction of rotation depends on the state: during *state A*, the current through the electric motor goes in a counter clockwise direction; and, during *state B*, the current through the electric motor goes in a clockwise direction.

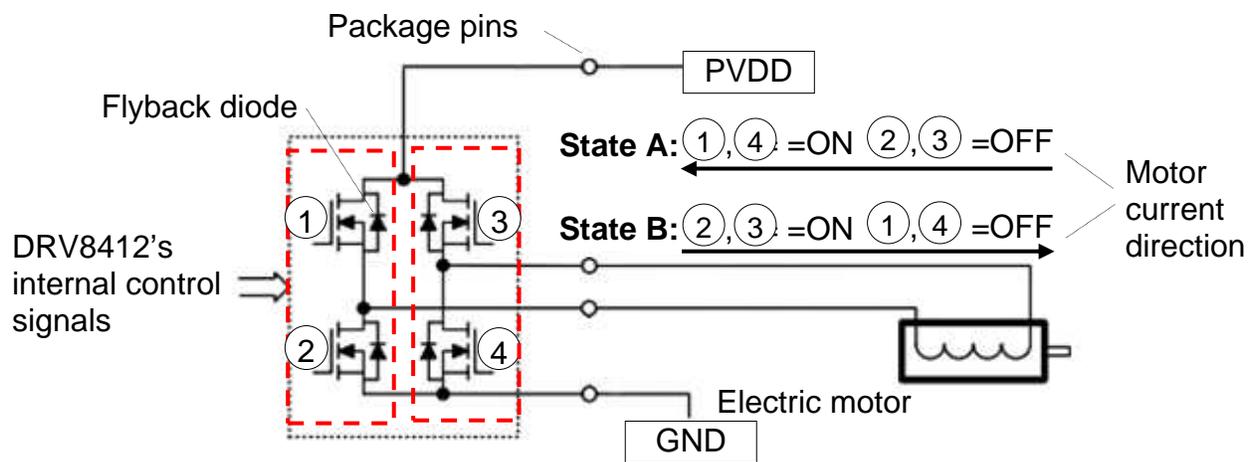


Figure 76. MOSFET Full-H-bridge (Marked in Black) Composed of Two Half-H-Bridges (Marked in Red-black)

In the *complementary Full Bridge* output configuration mode, the output stage is either in *state A* or *state B*. The output PWM duty cycle is defined in Figure 77-a, and the relationship between the input PWM duty cycle and the output PWM duty cycle is presented in Figure 77-b. At 100% input duty cycle the output will only be in *state A*, whereas at 0% input duty cycle the output will be only in *state B*. At a 50% input duty cycle there will be no current flowing in the motor as *state A* and *B* will be active for equivalent amounts of time. With the output PWM being typically above 20 KHz, the

average output voltage, which is linearly proportional to input PWM duty cycle, drives the motor at desired speed and direction.

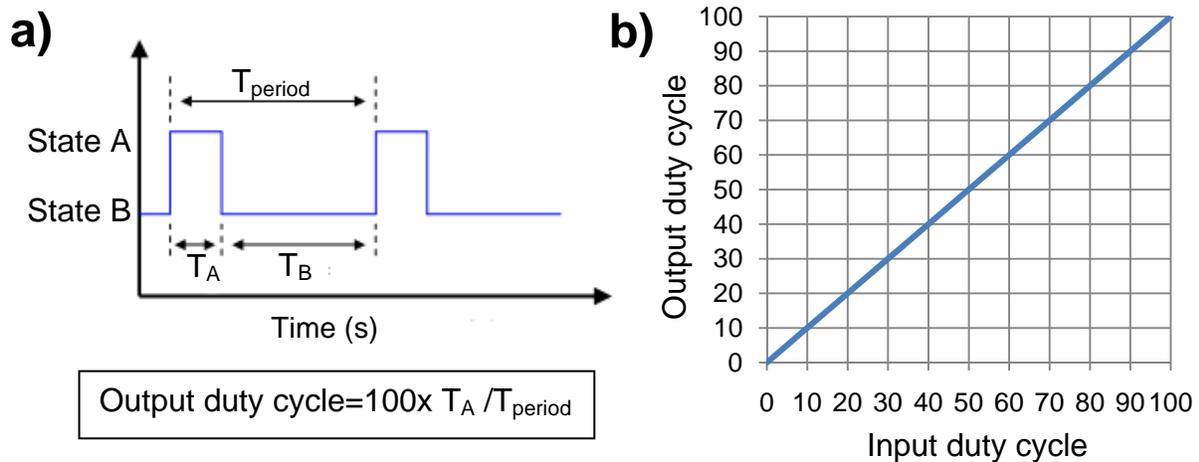


Figure 77. a) DRV 8412 Output PWM b) Input PWM Duty Cycle v.s. Output PWM Duty Cycle

Asserting the $\overline{\text{RESET_AB}}$ or $\overline{\text{RESET_CD}}$ pins low (i.e. digital 0) will force the corresponding Half-H-Bridges into a high-impedance state, disabling the motors. When driving the motors, the outputs need to be enabled by having $\overline{\text{RESET_AB}}$ and $\overline{\text{RESET_CD}}$ set high (i.e. digital 1). Furthermore, if a fault or over-temperature condition occurs, the outputs get disabled. To re-enable outputs both the $\overline{\text{RESET_AB}}$ and $\overline{\text{RESET_CD}}$ pins need to first asserted low and then set back to high again.

Moreover, the DRV 8412 incorporates internal fly back diodes on all Half-H-bridges thus no external ones are necessary (see Figure 76). As shown in Figure 75, each Half-H-Bridge has its own PVDD and GND pin allowing for each motor to be driven at the desired voltage.

The DRV8412 package has a 25 °C/W junction-to-ambient thermal resistance under the recommended PowerPad layout. Since the maximum allowed junction temperature is 150 °C, the maximum continuous power dissipation at the junction is 5 Watts assuming a 25 °C ambient temperature as presented below:

$$T_{\text{Junction (max)}} = 150 \text{ °C} = P_{\text{dissipated}} \times 25 \text{ °C/W} + 25 \text{ °C (ambient)} \Rightarrow P_{\text{dissipated}} = 5 \text{ Watts}$$

Since the DRV8412 is 97% efficient,

$$P_{\text{dissipated}} = 5 \text{ Watts} = \frac{100-97}{100} \times P_{\text{out}} \Rightarrow P_{\text{out}} = 167 \text{ Watt}$$

Thus, each Full-H-bridge can output 83 Watts assuming there is equal power being output from both Full-H-bridge outputs. For a 6V PWM output, the maximum continuous current output per Full-H-bridge is $I_{\text{max}} = P/V = 83/6=13.88$ Amps. The UBC-PULO DC motors are driven at 6 Volts and have a stall current of 1.1 Amps. Thus, each DRV 8412 can drive two UBC-PULO DC motors even at stall current with no need for external heat sinks. The worst case calculation is shown below with the assumption of 70 °C ambient temperature and 93% efficiency:

$$150 \text{ °C} = P_{\text{dissipated}} \times 25 \text{ °C/W} + 70 \text{ °C (ambient)} \Rightarrow P_{\text{dissipated}} = 3.2 \text{ Watts}$$

$$P_{\text{dissipated}} = 3.2 \text{ Watts} = \frac{100-93}{100} \times P_{\text{out}} \Rightarrow P_{\text{out}} = 45.7 \text{ Watt} \Rightarrow P_{\text{out_AB}} = 22.85 \text{ Watt}$$

$$I_{\text{max}} = \frac{22.86}{6} = 3.81 \text{ Amps}$$

This worst case calculation demonstrates that even in extreme circumstances, each DRV8412 Full-H-Bridge would be sufficient to drive the UBC-PULO motors. For more details about the DRV8412, please refer to the manufacturer's data sheet¹.

E.1.2 Level shifter

The Integrated Device Technology Inc. IDT74ALVC chip is a 16-bit 5V to 3.3V level shifting transceiver. This level shifter was selected to convert the 5V PWM signal produced by the MPL PATI DAQ card to a 3.3V PWM signal compatible with DRV8412's input PWM requirements. Since four DRV8412 motor driver chips were used in the PCB design, a 16-bit convertor was selected providing 16 PWM level conversions. The IDT74ALVC is available in a 48-TSSOP PCB package, which is quite compact. This device is typically used for PCI Interface applications so there were no concerns regarding speed, slew rates and reaction times (i.e. PCI bus is in MHz range which is much faster than the 20 KHz used for PWM motor driving).

¹ <http://www.ti.com/lit/ds/symlink/drv8432.pdf>

E.1.3 Current sensor IC small with no external components

The Allegro ACS712 current sensing IC was selected mainly because of its compact size (i.e. standard SOIC-8), accuracy, implementation simplicity and adjustable output bandwidth. This IC is able to sense current in the range of -5 to +5 amps and has a linear analog output with 185 mV/A sensitivity as shown in Figure 78-b. Since this IC utilizes a linear Hall Effect circuitry to sense the current, it only introduces a 1.2 mΩ resistance in the conductor path under test, resulting in an extremely low power loss. Figure 78-a demonstrates the electrical connection diagram where the current path “IP” under test is connected in series to the IP+ and IP- pins. An external capacitor C_F adjusts the output bandwidth. An adjustable output bandwidth is a desirable feature for the UBC-PULO motor current sensing application since high-frequency transient response is not of interest. More detail can be found in the corresponding data sheet¹.

¹ www.allegromicro.com/~Media/Files/.../ACS712-Datasheet.ashx

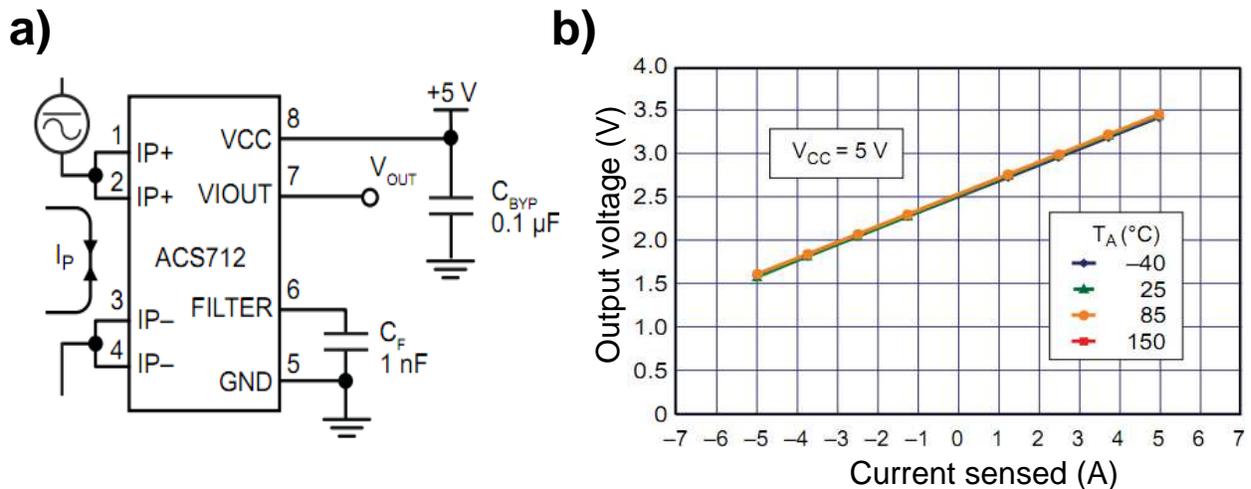


Figure 78. a) Allegro Microsystems ACS712 b) Output Voltage V.S Sensed Current

E.1.4 Power regulators and batteries

The UBC-PULO electrical system has number of voltage level requirements. The DAQ cards, SBC board, angle sensors, shoulder interface and head interface require a 5V supply. Additionally, a 12V supply is required for the DRV8412 motor driver IC and a 3.3V supply is needed for the level shifter IC. The DC motors are rated at 6 V. These voltage levels need to be regulated to prevent damage to electrical components, reduce noise and ensure proper operation of analog and digital circuits. Table 29 shows the expected current consumption of all electrical components. Thus, the power regulators have to meet at least the following power requirements (i.e. calculated by $P=V \times I_{total}$):

$$P_{5V} = 5V \times 3.8A = 19 W$$

$$P_{3.3V} = 3.3V \times 0.021A = 0.069 W$$

$$P_{12V} = 12V \times 0.202A = 2.42 W$$

Table 29. Typical Current Consumption

Voltage supply	DAQ cards (A)	SBC (A)	Current sensors (mA)	Motor driver ICs (mA)	Level shifter (mA)	LEDs (mA)	Angle sensor (mA)	Head interface (mA)	Shoulder interface (mA)
+3.3 V	0	0	0	0	1	1x20	0	0	0
+ 5 V	2x0.5	2.37	8x4	0	1	10x20	6x16	5	22
+12 V	0	0	0	4x48	0	1x20	0	0	0

The new UBC-PULO electrical system uses two separate batteries: one for powering the DC motors and the other for the rest of the electrical system. Having a separate battery for the motors is desirable for the following reasons:

- Due to their highly inductive nature, DC motors introduce a lot of undesirable noise and voltage spikes into the electrical circuit. A separate battery isolates the noisy DC motor circuit from the rest of the electrical system.
- DC motors introduce a relatively high amount of current fluctuation into the circuit (i.e. Up to 6.6 Amps if all six motors are stalling). Since batteries have an internal resistance, current fluctuation will directly reflect on their output voltage. If only one battery is used, a rapid current change caused by the DC motors will result in a rapid voltage change, affecting the rest of the electrical system's supply voltage. The use of power regulators would only reduce these voltage fluctuations and not eliminate the source of the problem.
- Having a separate battery adds the flexibility of changing to DC motors that have a different voltage rating.

The Texas Instruments PTR08060, shown in Figure 79-a, was selected to provide the 5V supply. The TDK CC6-0512SR-E, shown in Figure 79-b, was selected to provide the 12V supply. Both regulators are compact and efficient board-mounted switching regulators that convert a 6 to 9V battery voltage to well-regulated 5V or 12V supplies as required by the various electrical components. Additionally, the Texas Instruments TL1963A-33DCYR linear regulator was selected to convert the 5V provided by the PTR08060 to a 3.3V supply required for the level shifter IC.

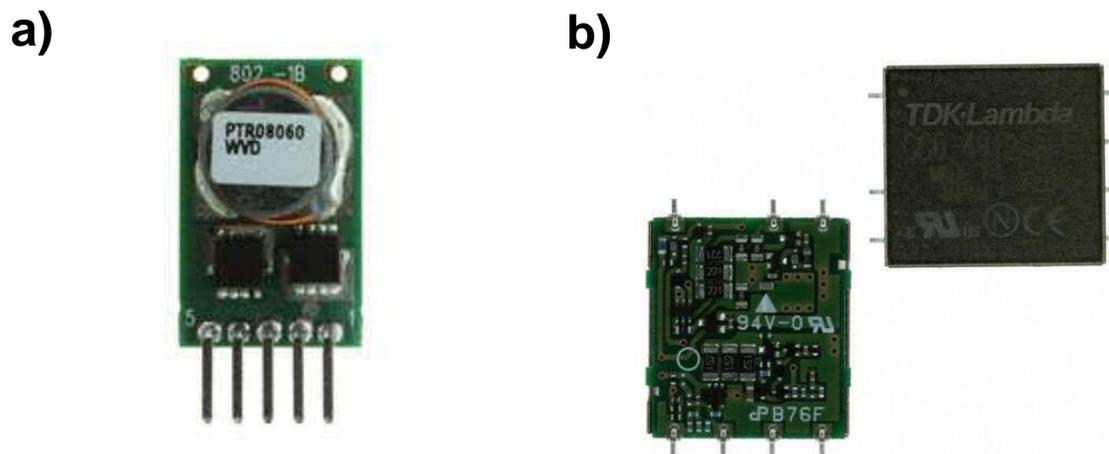


Figure 79. a) Texas Instruments PTR08060W (33 Watts) b) TDK-LAMBDA AMERICAS INC CC6-0512SR-E (6 Watts)

E.1.5 LEDs

A number of blue, green, yellow and red LED (Light Emitting Diodes) devices were implemented to provide feedback as to whether the UBC-PULO PCB is under normal operation or under fault conditions. These LEDs are compact and are surface mounted to best satisfy PCB space limitations (i.e. standard 3225 metric PCB package).

E.1.6 DIP switches

Compact eight-position top-actuated Dual In-line Package (DIP) switches were used to add flexibility to the over-all design. For example, the switches can be configured to enable the use of both DRV8412's Full-H-bridge outputs in parallel, making the option of driving a more powerful DC motor possible.

E.1.7 Connectors

Standard 50-pin and 35-pin insulation displacement connectors (IDC) and standard PC/104 connectors were implemented to connect the DAQ cards to the custom PCB (see Figure 34). To promote flexibility and ease of connection while still maintaining a compact design, the WAGO 233 series terminal blocks, presented below, were selected for connecting the custom PCB to UBC-PULO's various sensors and actuators (see Figure 34).

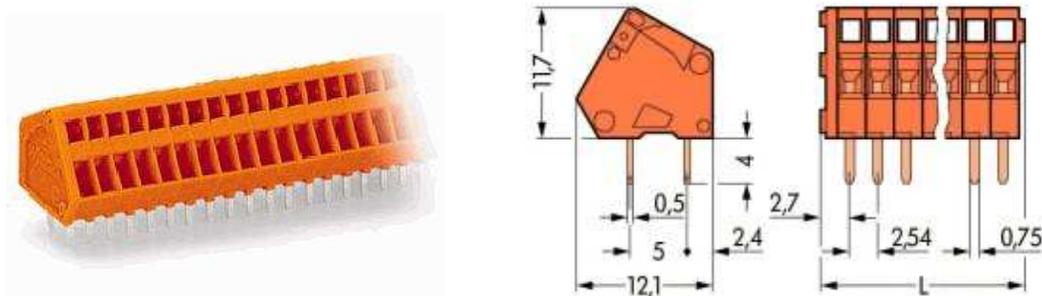


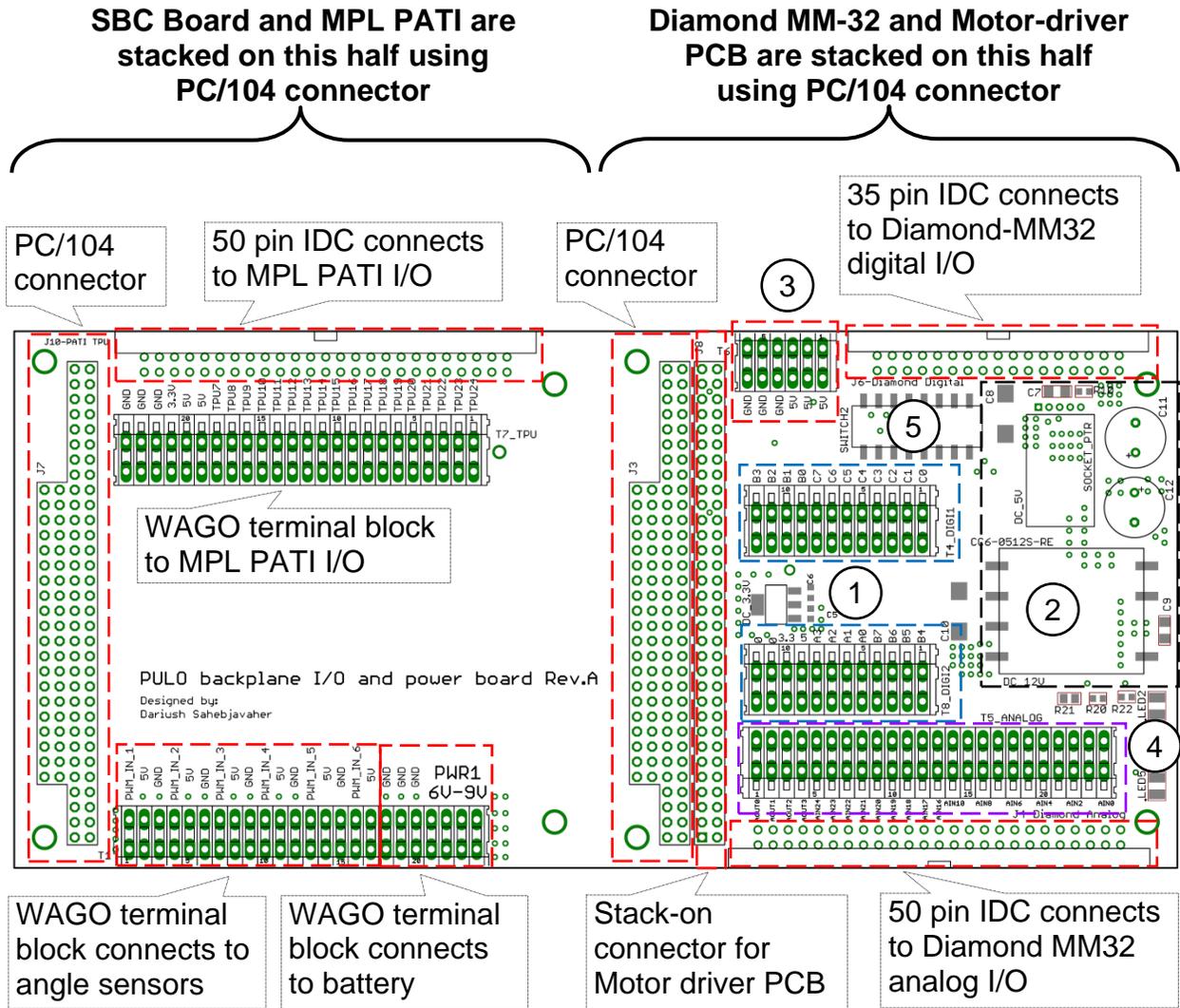
Figure 80. WAGO 233 Series Terminal Blocks

E.2 PCB features and theory of operation

The final design consists of two custom PCB boards: the “Backplane I/O and Power board”, and the “Motor Driver” board. Figure 81 shows the final backplane PCB design, which has the following main functions:

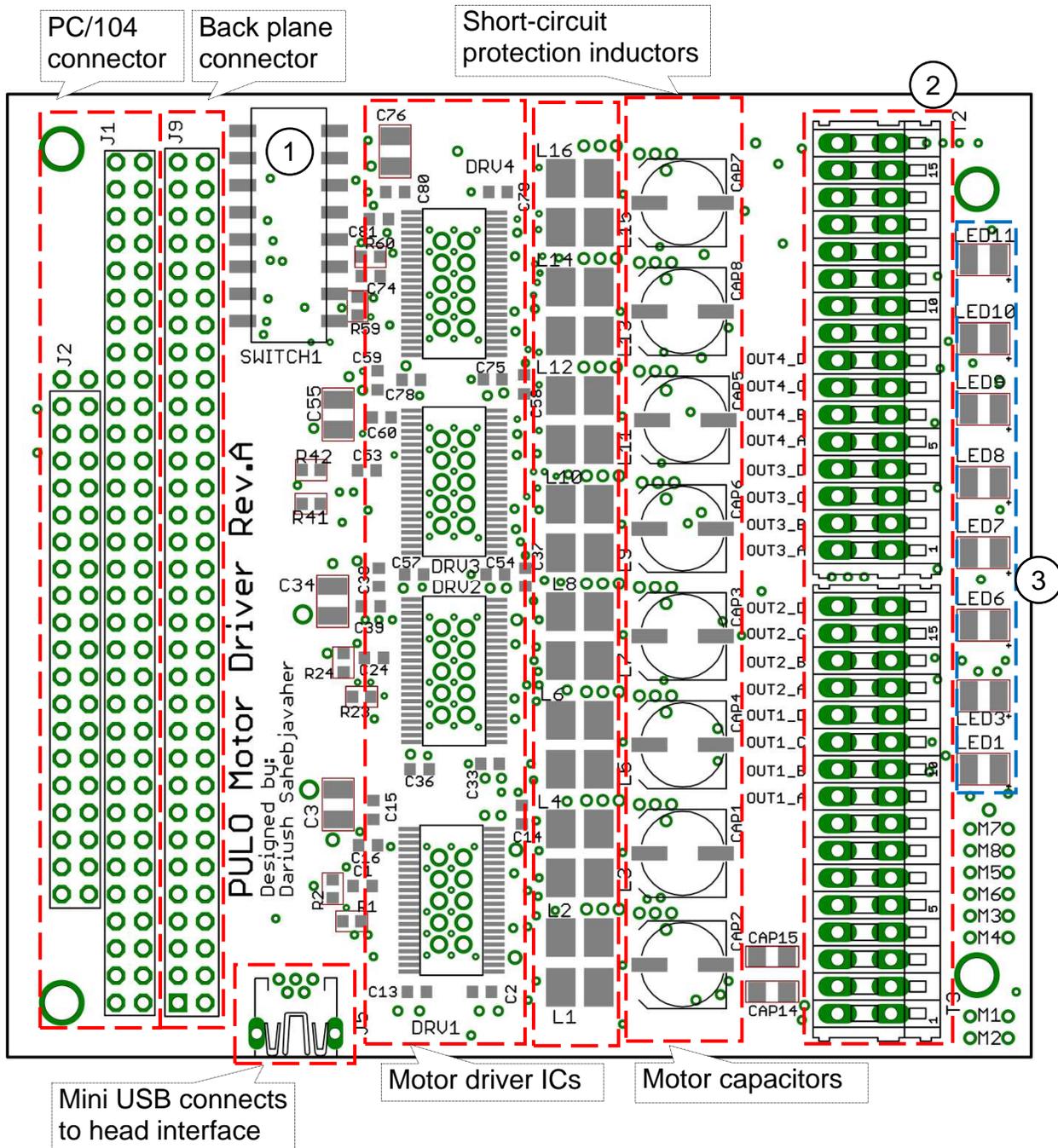
- It acts as a solid base and expansion module for other PC/104 boards to be stacked on to.
- It makes all DAQ card I/Os accessible via WAGO terminal blocks.
- It converts and regulates battery power to supply the entire electrical system.

The design allows for the PCM 3362 SBC board and MPL PATI to be stacked on the left side. Similarly, the Motor Driver PCB and Diamond MM-32 are stacked on the right side. The IDC connectors are located directly below the DAQ card IDC connectors allowing for one-to-one connections via ribbon cables. Any digital or analog I/O through the electrical system is attached to this board, with the exception of motor power and motor output. The power regulators provide clean +3.3V, +5V and +12V while also protecting the electrical system against short-circuit (i.e. in case of a short circuit the regulators immediately shut down) or under voltage (i.e. when battery voltage is below 6V, the power regulators shut down).



1. WAGO terminal blocks highlighted in blue provide Diamond-MM32 digital I/O. WAGO terminal blocks highlighted in purple provide Diamond-MM32 analog I/O.
2. Power regulation circuitry converting battery power to +3.3V, +5V and +12V that supply the entire electrical system.
3. A +5V supply output for the PCM-3362 SBC board.
4. Three LEDs indicate if +3.3V, +5V and +12V supply are online.
5. DIP switch provides a configurable OTW and FAULT reporting.

Figure 81. Backplane I/O and Power PCB (Top View)



1. DIP switch allows for enabling the DRV8412 parallel output mode option.
2. Supports eight DC motor outputs. Each DC motor needs a separate voltage supply.
3. Yellow LEDs lit on in case of motor driver IC OTW (Over Temperature Warning). Red LEDs lit on in case of motor driver IC Fault condition.

Figure 82. Motor Driver PCB Top Side (Top View)

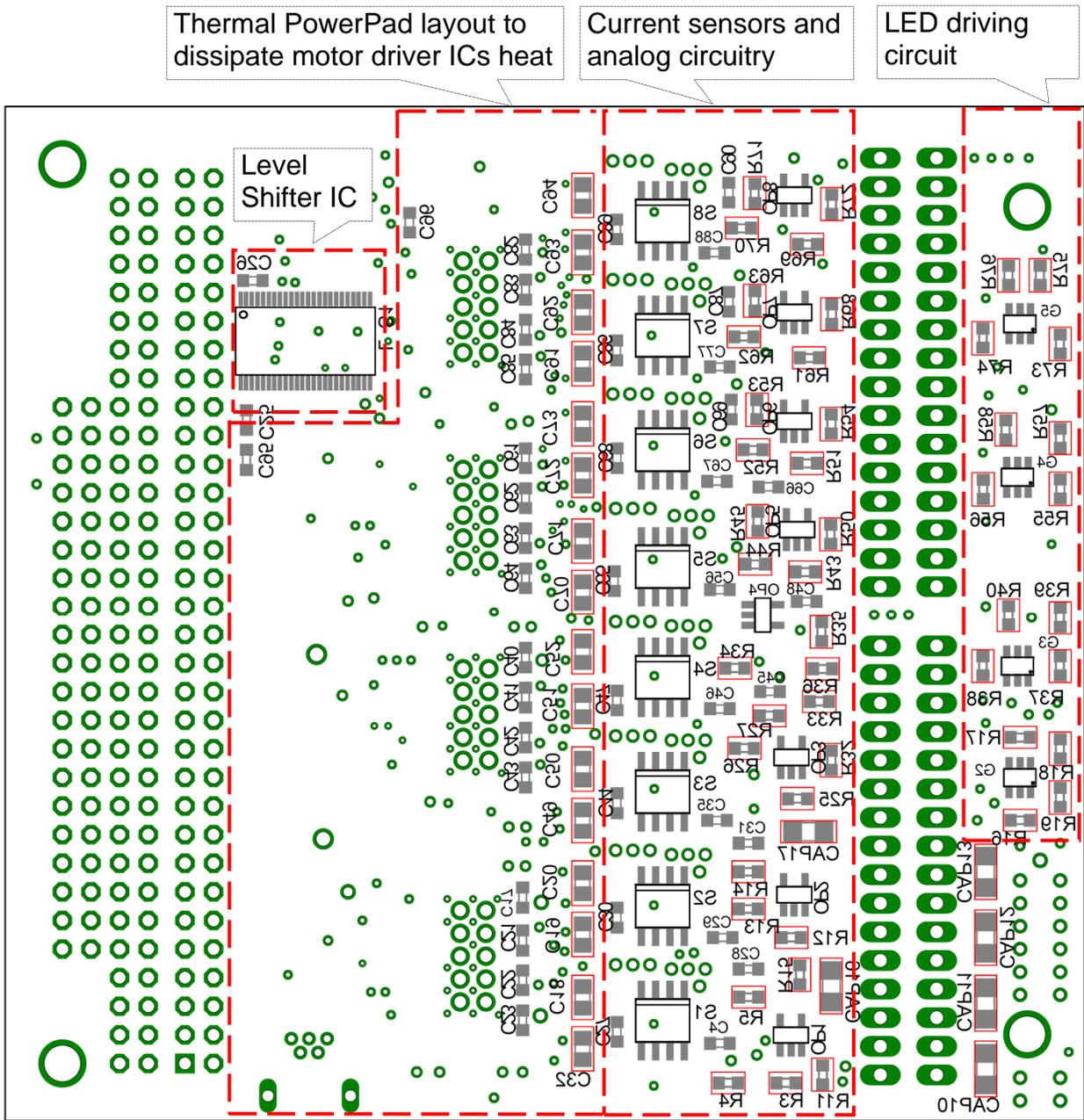


Figure 83. Motor Driver PCB Bottom Side (Top View)

The final Motor Driver PCB design, with top component side shown in Figure 82 and bottom component side shown in Figure 83, is a PC/104 compatible board capable of driving eight motors while sensing real-time current passing through each motor. A

DIP switch allows for configuring each DRV 8412 motor driver IC to operate in normal mode or parallel mode. In parallel mode the DRV 8412's two Full-H-Bridges work in parallel making it possible to drive motors requiring twice the output power.

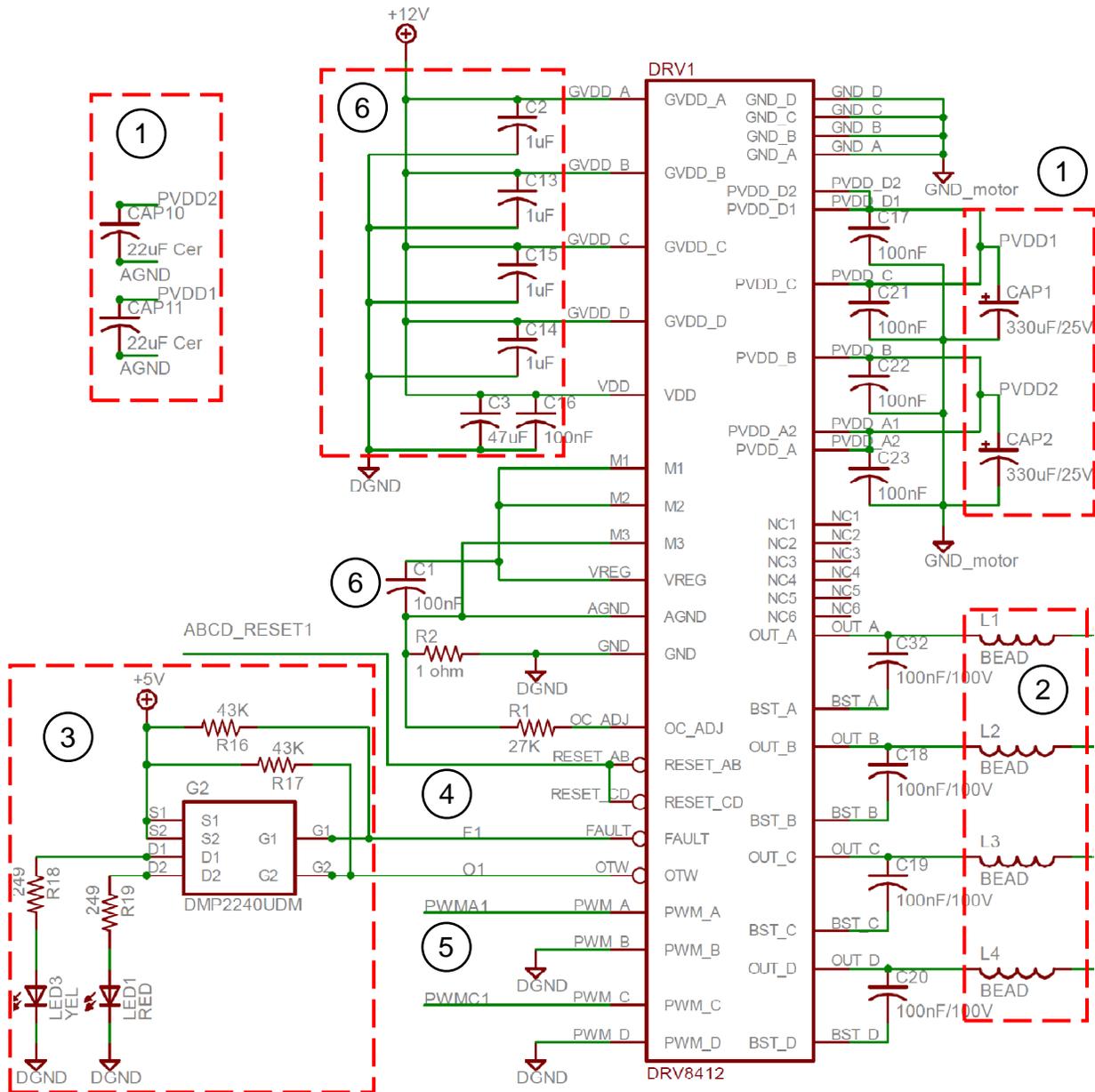
Additionally, each motor driver IC's condition is readily known through feedback LEDs. Yellow LEDs light in case of Over Temperature Warning and red LEDs light in the case the motor driver IC undergoes a Fault condition (i.e. over-current, short-circuit, over temperature shutdown, under-voltage). More details about the PCB Layout and design considerations are presented in Appendix E.3 and E4.

Figure 84 shows a sample connection diagram for one of the DRV 8412 motor driver ICs. The output configuration has been set to dual complementary, Full-H-Bridge mode allowing a PWM_A signal received from MPL PATI to control a motor connected to OUT_A and OUT_B. Similarly, a PWM_C signal received from MPL PATI will control a motor connected to OUT_C and OUT_D. The short-circuit inductors implemented at the output stage were calculated based on recommendations by Texas Instruments. These inductors limit the output slew rate in the case of a short circuit, and give the device enough time to shut-down before any damage occurs. PVDD1 and PVDD2 power the AB-Full-H-Bridge and CD-Full-H-Bridge respectively. Thus, the two motors can each be driven at the desired PVDD voltage. To mitigate the effects of transient current induced by the DC motors on the PVDD line, two 330 μ F general purpose electrolytic motor capacitors and two 22 μ F ceramic capacitors were connected in parallel to each PVDD. This prevents the current surge into the DC motors to translate

to the rest of the circuitry. When the motor transients draw large currents, the currents come from these capacitors instead of coming directly through the battery. Because these transient currents can be quite large, without the capacitors, the entire path from the battery to the motor will become noisy due to the inductive nature of the conductor. The capacitors minimize the distance travelled by the transient currents (i.e. path between the capacitors and motors is significantly shorter than path from the battery and motors), resulting in a significantly less noisy circuit. The 330 μF capacitors have a larger time constant ($\tau = RC$) and thus provide buffering for lower-frequency, yet larger in-rush currents. On the other hand, The 22 μF capacitors have a smaller time constant and provide buffering of higher-frequency, but smaller inrush currents.

The RESET, OTW and FAULT pins are connected to the Diamond MM-32 digital I/O through the backplane connector (i.e. J9 in Figure 82). Therefore, the UBC-PULO controller can be aware of OTW and FAULT conditions only if the backplane DIP switch is in off position (see Figure 81). The RESET pin is also available to UBC-PULO controller, allowing the motors to be immediately disabled in the case of an emergency or to enable the motor driver IC's after a FAULT condition. Moreover, a dual P-Channel MOSFET was selected to drive the two OTW and FAULT indication LEDs. The resistor calculations are provided in equation 20. Since the selected LEDs had a 2V forward voltage and were driven with a 12 mA current, a suitable resistor value of $R = \frac{5-2}{0.012} = 250\Omega$ was selected.

$$R = \frac{V_s - V_f}{I_{LED}} \quad \text{eq. 20}$$



1. Motor-power capacitors
2. Half-H-bridge outputs A,B,C and D with short-circuit protection inductors
3. LED driving circuit indicating OTW or FAULT condition
4. FAULT,OTW and RESET pins are connected to the Diamond MM-32 DAQ card
5. PWM_A and PWM_C are connected to the MPL PATI card
6. Supply noise reduction capacitors recommended by TI

Figure 84. Schematic Diagram for The DRV 8412 Motor Driver IC

The 27 KΩ resistor between the OC_ADJ pin and ground sets the Over Current settings to 9.7 A. Thus, the DRV 8412 will go into a FAULT condition if any output current exceeds 9.7 A. This resistor is a 1206 standard PCB package and can be changed to any desired value later if necessary. The rest of the external components were selected and connected in accordance with the data sheet instructions.

Figure 85 shows the Level shifter IC schematic which converts the eight 5 Volt MAPL PATI PWM signals going to the motor driver ICs to 3.3 Volts. A DIP switch allows for the shorting of the two PWM signals together for parallel mode operation by sending the same control signal to both Full-H-Bridges.

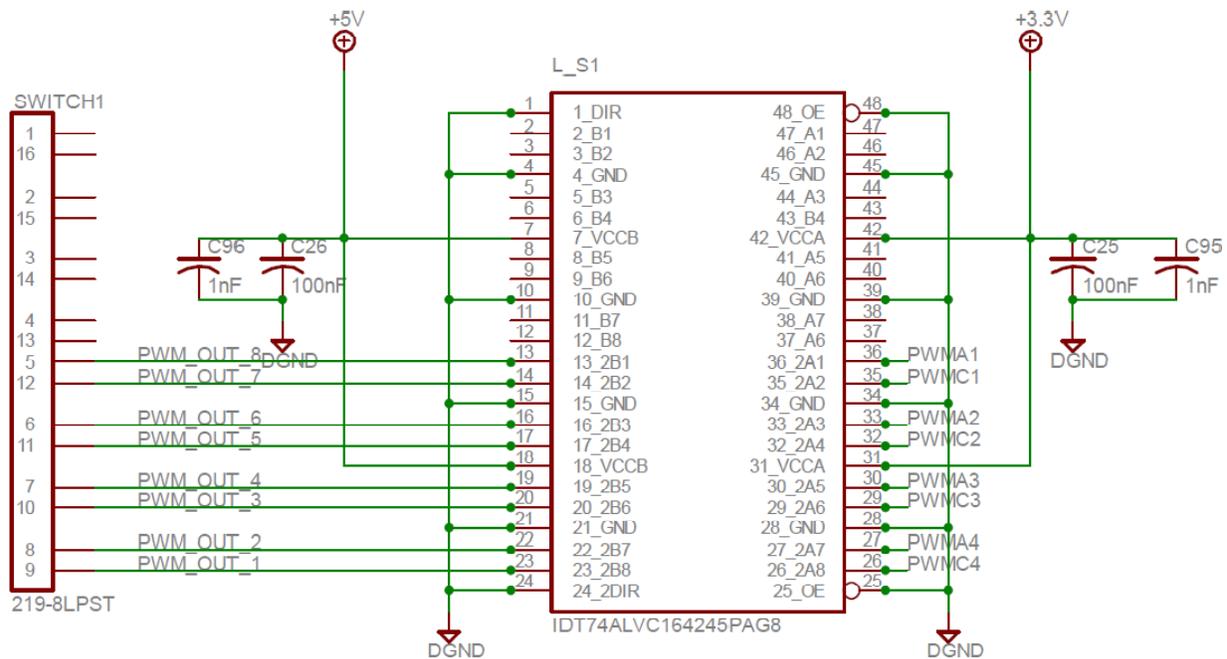


Figure 85. Level Shifter Schematic

Figure 86, shows a sample current sensor connection diagram which is connected in series with each Full-H-Bridge motor output. The current sensor produces a linear 185mV/A analog output (V_{out}). This analog output is amplified by an inverting Operational Amplifier (Op Amp) circuit producing $V_{current} = 2.5 - (R_5 / R_4) \times V_{out}$. In this case $V_{current} = 2.5 - (R_5 / R_4) \times V_{out} = 2.5 - 3.3 \times V_{out}$, increasing the current output sensitivity by 3.3 times. All eight current sensor outputs are amplified in the same manner and the analog $V_{current}$ signals are sent out to the Diamond MM-32 Analog I/O channels through the backplane connector (see Figure 82) and 50 pin IDC connector (see Figure 81).

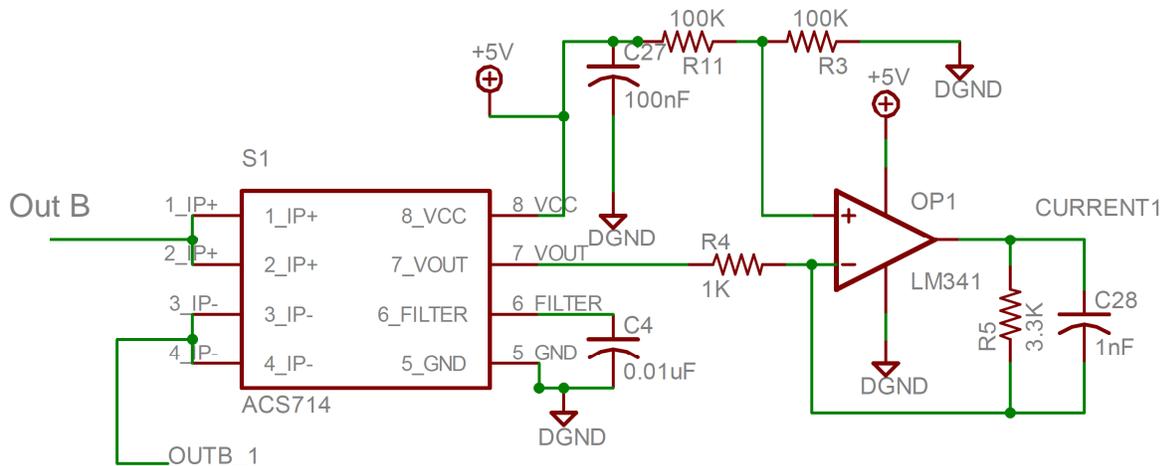


Figure 86. Current Sensor Schematic

Figure 87 presents the connection diagram for the various power regulators located on the backplane board. The external components were selected and connected in accordance with the corresponding data sheet instructions. The LED resistance values were calculated using equation 20.

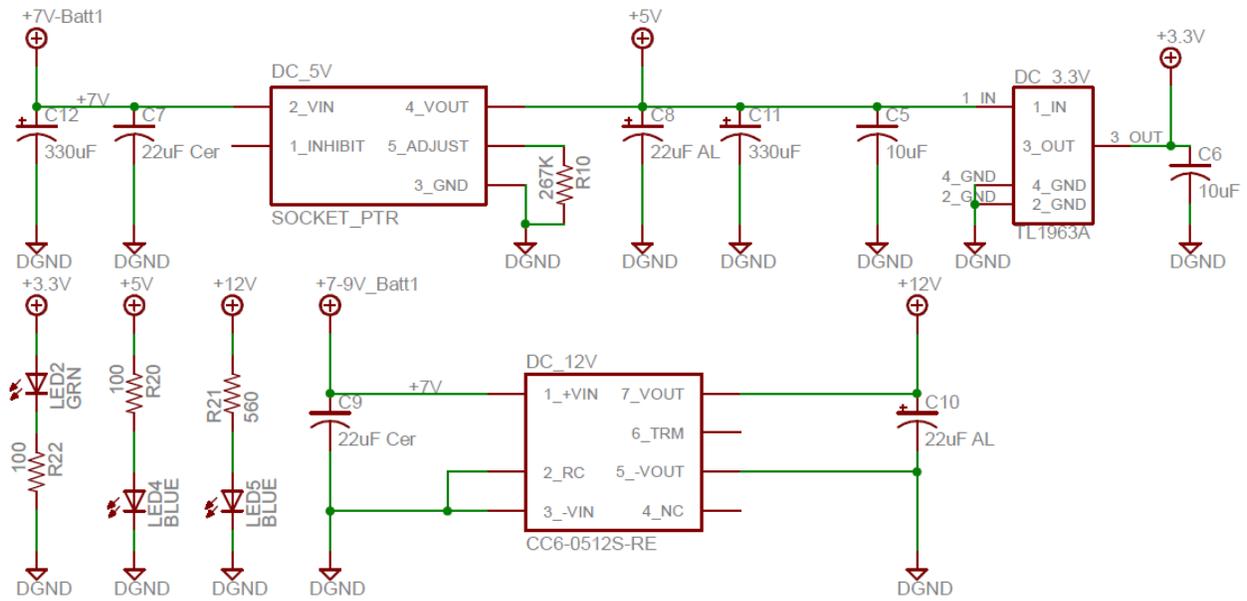


Figure 87. Power Regulator Schematic

For further details regarding a detailed connection schematic and PCB layout please refer to Appendix E.4.

E.3 PCB design consideration and manufacturing

The PCB was designed in CadSoft Eagle 5.10 by carefully following all datasheet instructions and “rules of thumb” provided in the “PCB Design Tutorial” by David L. Jones [47]. Both the backplane PCB board and the motor driver PCB are six-layer PCB circuits because of the overall complexity introduced by the number of components and noise isolation requirements. In addition to the electrical noise, thermal, trace width, space limitations and component position constraints, physical three-dimensional mechanical constraints were met in order to satisfy the PC/104 form factor standards. Many components (namely the DRV8412 and power regulators) required a custom library package design which included the design of the devices’ PCB footprint and

corresponding schematic representation. Figure 89 and Figure 90 show the final six layer design of these PCB boards. A detailed layer by layer representation of the PCB layouts along with the corresponding schematic diagram is presented in Appendix E.4.

The main design considerations are listed below:

- Motor capacitors (see Figure 82) had to be as close as possible to the corresponding PVDD pin to minimize noise.
- Bypass capacitors [47] had to be placed as close as possible to the corresponding component pin. This smoothens out the power going to that particular device.
- The motor output traces had to be as wide as possible to minimize path impedance (See Figure 91).
- Ground plane layers had to be incorporated to enable short paths to ground, allowing low impedance ground connection. This would allow pins to be connected to ground by use of vias (i.e. vias are plated holes that connect traces on different PCB layers by making a vertical electrical connection).
- If high currents or high frequency signals were expected through a path, the traces had to be wider and multiple vias had to be utilized. Increasing the number of vias, size of vias or the trace width would decrease the path's inductance and resistance.
- The motor ground plane, digital ground plane and analog ground plane, which are in order of most noisy to least noisy, had to be separated to isolate the more sensitive circuits from noise. The grounds had to be connected at one point close

to the battery using a star configuration. These ground planes had not to be overlapped as noise could be inherited through the capacitive nature of two overlapped layers [47]. See Appendix E.4 for more details.

- The IDC connectors had to ideally be placed at the board edges to provide a one-to-one connection with corresponding DAQ card I/O.
- Different signal types (i.e. digital signals, PWM signals and analog signals) had to be routed not too closely to keep them isolated; this is especially true for PWM signals which would cause a high Electro Magnetic Interference (EMI).
- The bottom side of the DRV8412 motor driver ICs needed at least a $34\text{mm}^2/\text{IC}$ area of clearance to satisfy the manufacturer's recommended thermal requirements (See Figure 83). Use of Solid vias for the DRV8412 PowerPads were highly recommended by Texas Instruments (See Figure 88) as they provide a significantly lower thermal resistance [51].

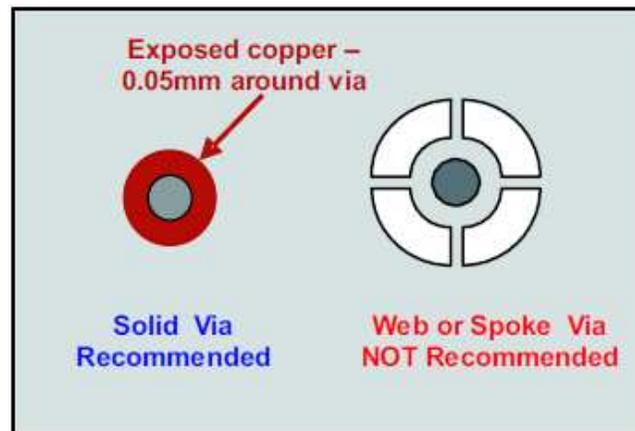


Figure 88. Texas Instruments Recommends Solid Vias for PowerPAD

- Ground planes block EMI radiation and can provide a good isolation between analog circuits and digital circuits located on the top and bottom PCB sides. The analog circuit on the bottom side of the motor driver PCB shown in Figure 83 was

isolated from the noisy top layer's output stage in this fashion by use of two ground layers (please refer to Appendix E.4 for details).

- Components with a height more than 2.5 mm had to be placed on the top side in order to comply with the PC/104 form factor standards [52]. The maximum height of any top side component could not exceed 11 mm for the same reason [52].

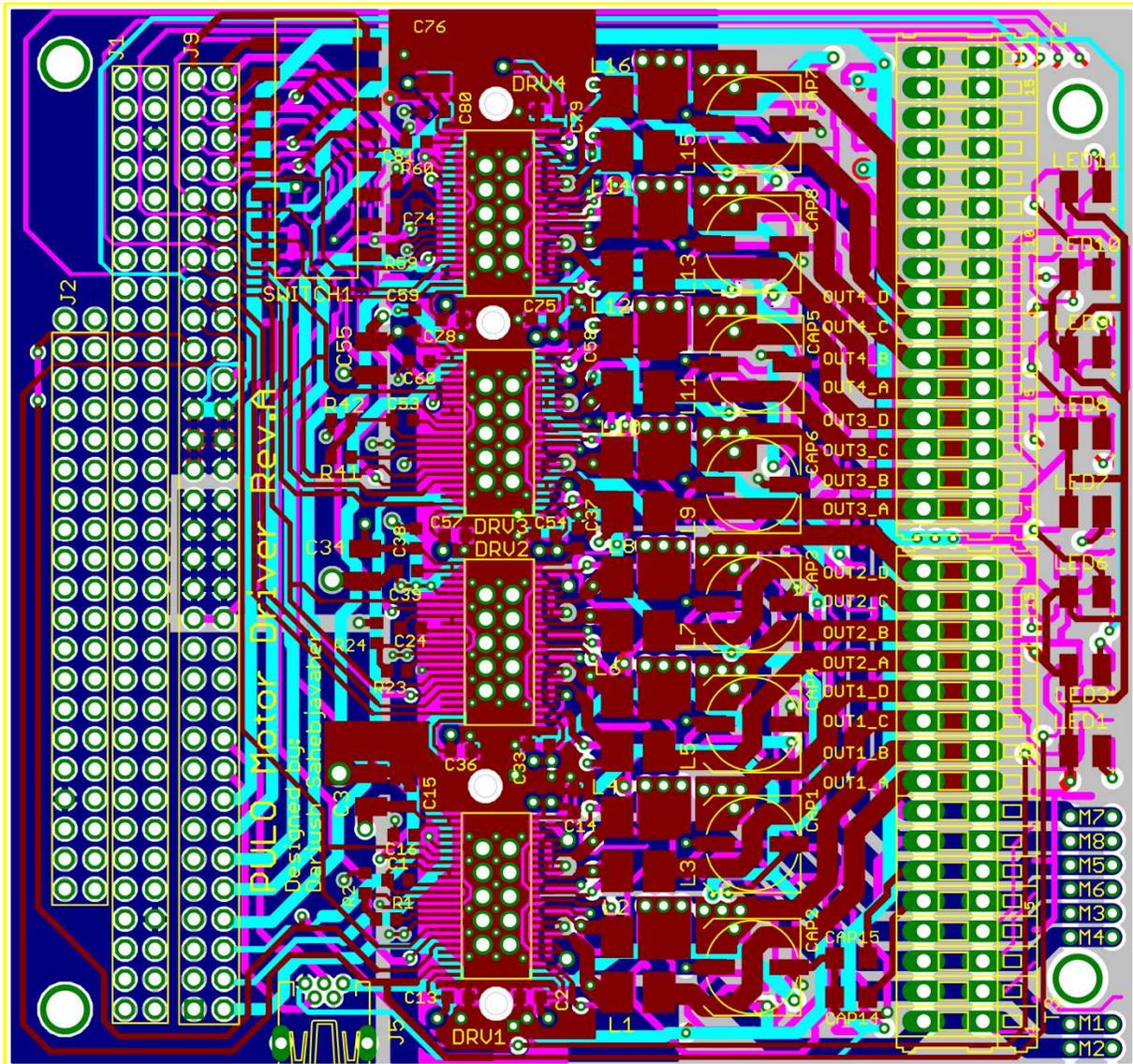


Figure 89. Screen Shot of Motor Driver Board PCB Layout Showing Top Side Components and All Six Layers

DRV8412 motor output traces as wide as possible. Each motor output trace goes to the bottom layer where the current sensing circuitry is located. Multiple vias used to minimize path impedance.

Motor output traces return to top layer via multiple vias and the wide traces go to the WAGO terminal block

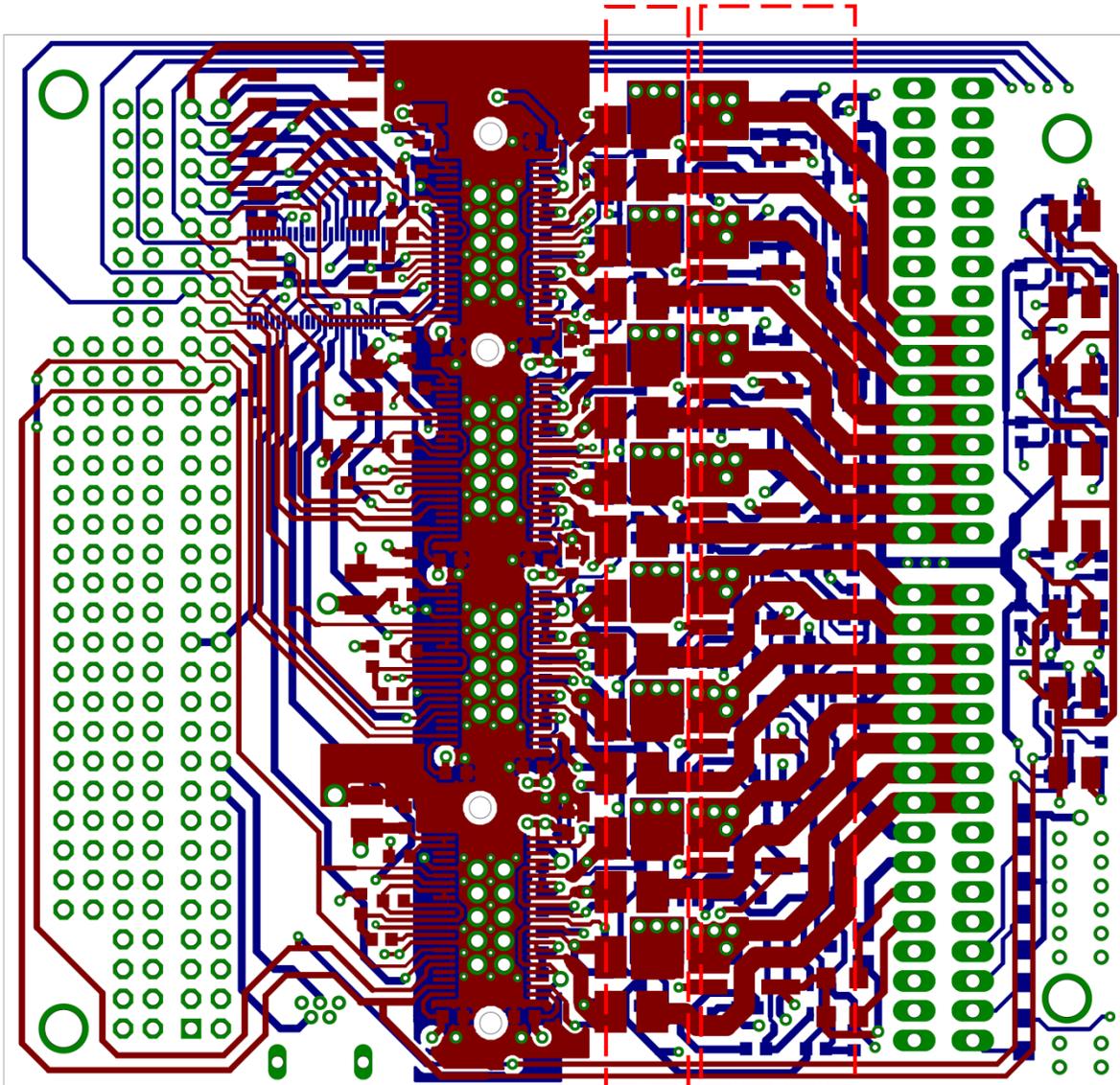
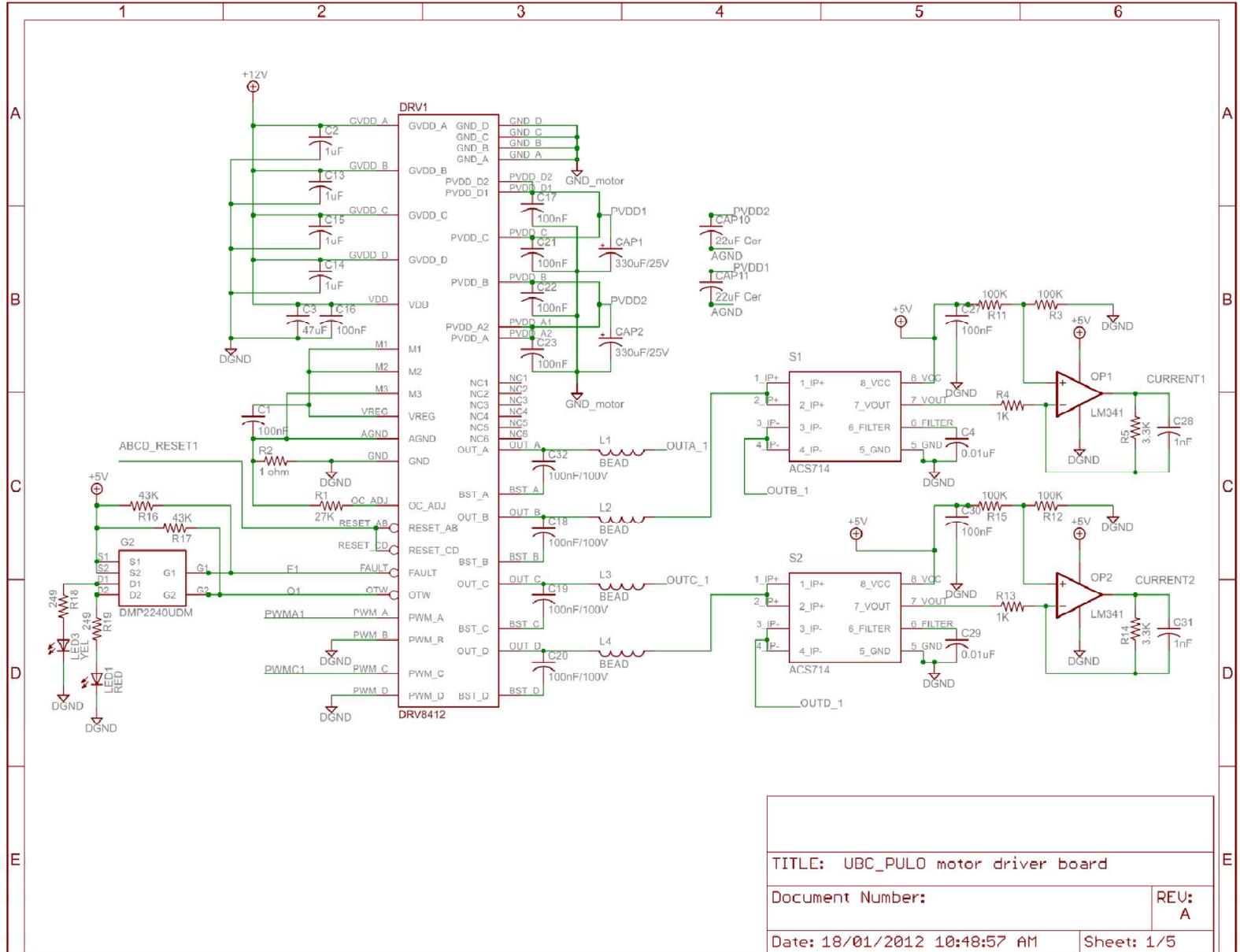


Figure 91. PCB Layout of the Motor Driver Board's Top and Bottom Layer

E.4 Custom Board schematic diagrams and PCB layouts

E.4.1 Motor driver board schematic diagram

Figure 92. Schematic Diagram for the Motor Driver Board -Sheet 1/5



TITLE: UBC_PULO motor driver board	
Document Number:	REV: A
Date: 18/01/2012 10:48:57 AM	Sheet: 1/5

Figure 93. Schematic Diagram for the Motor Driver Board -Sheet 2/5

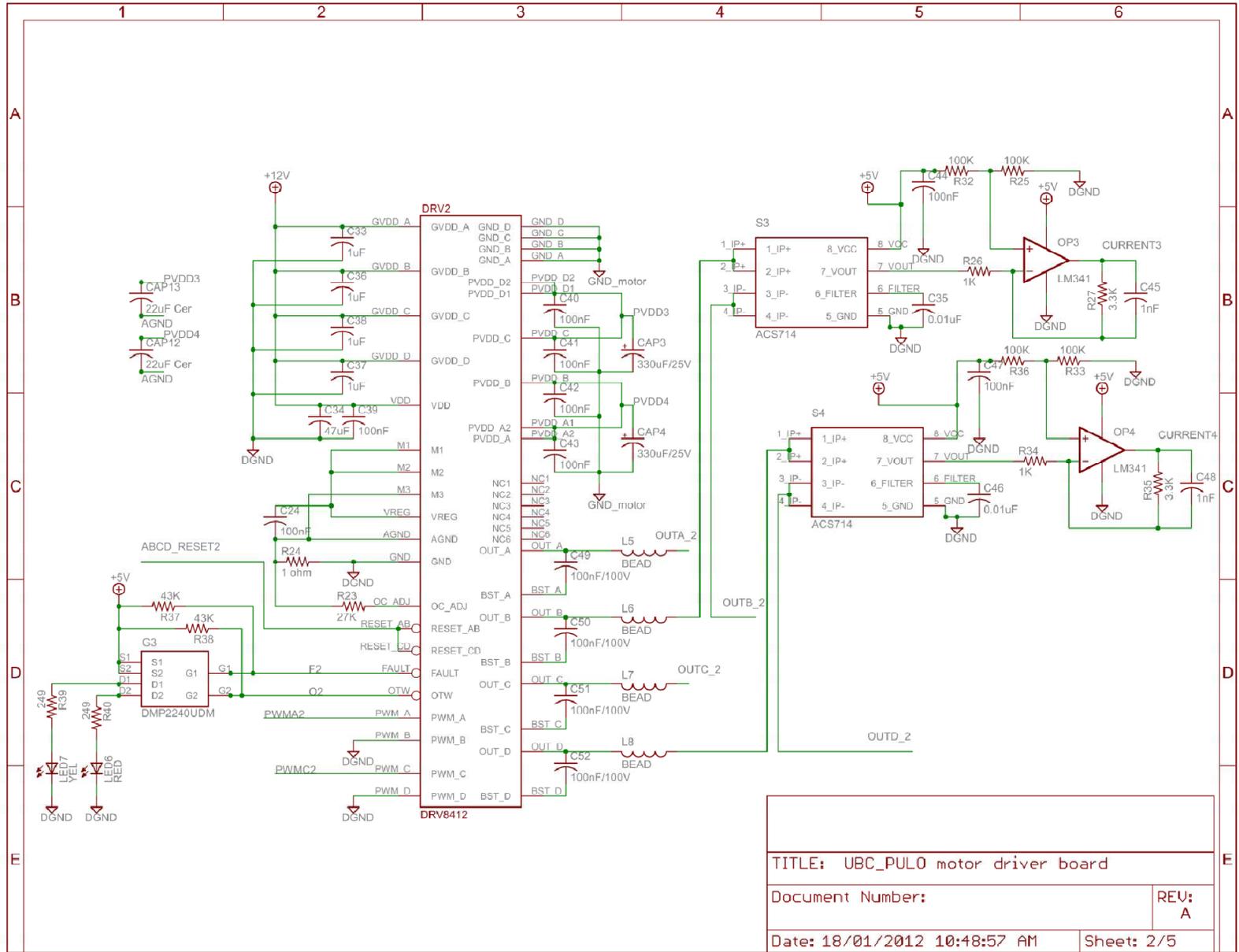


Figure 94. Schematic Diagram for the Motor Driver Board -Sheet 3/5

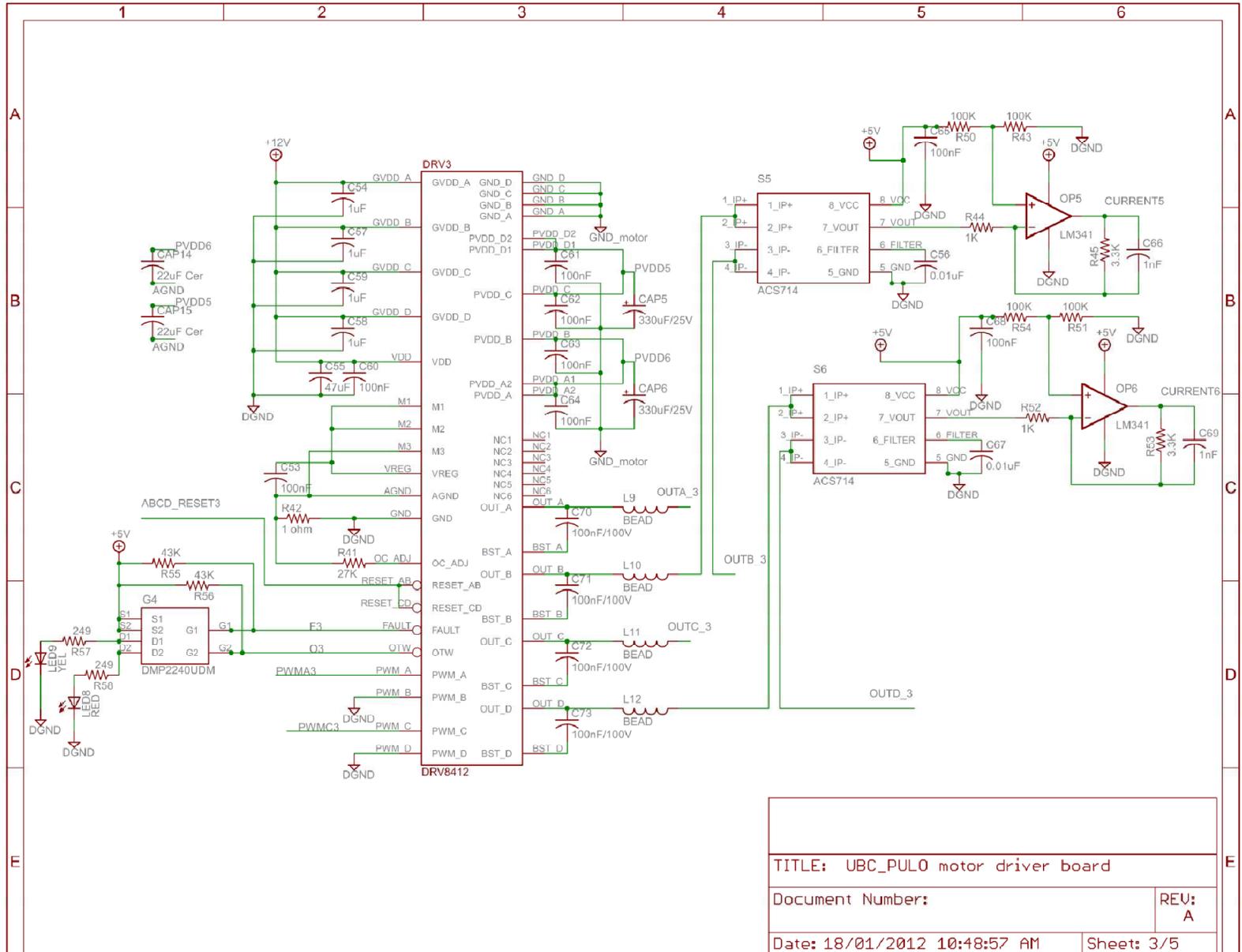


Figure 95. Schematic Diagram for the Motor Driver Board -Sheet 4/5

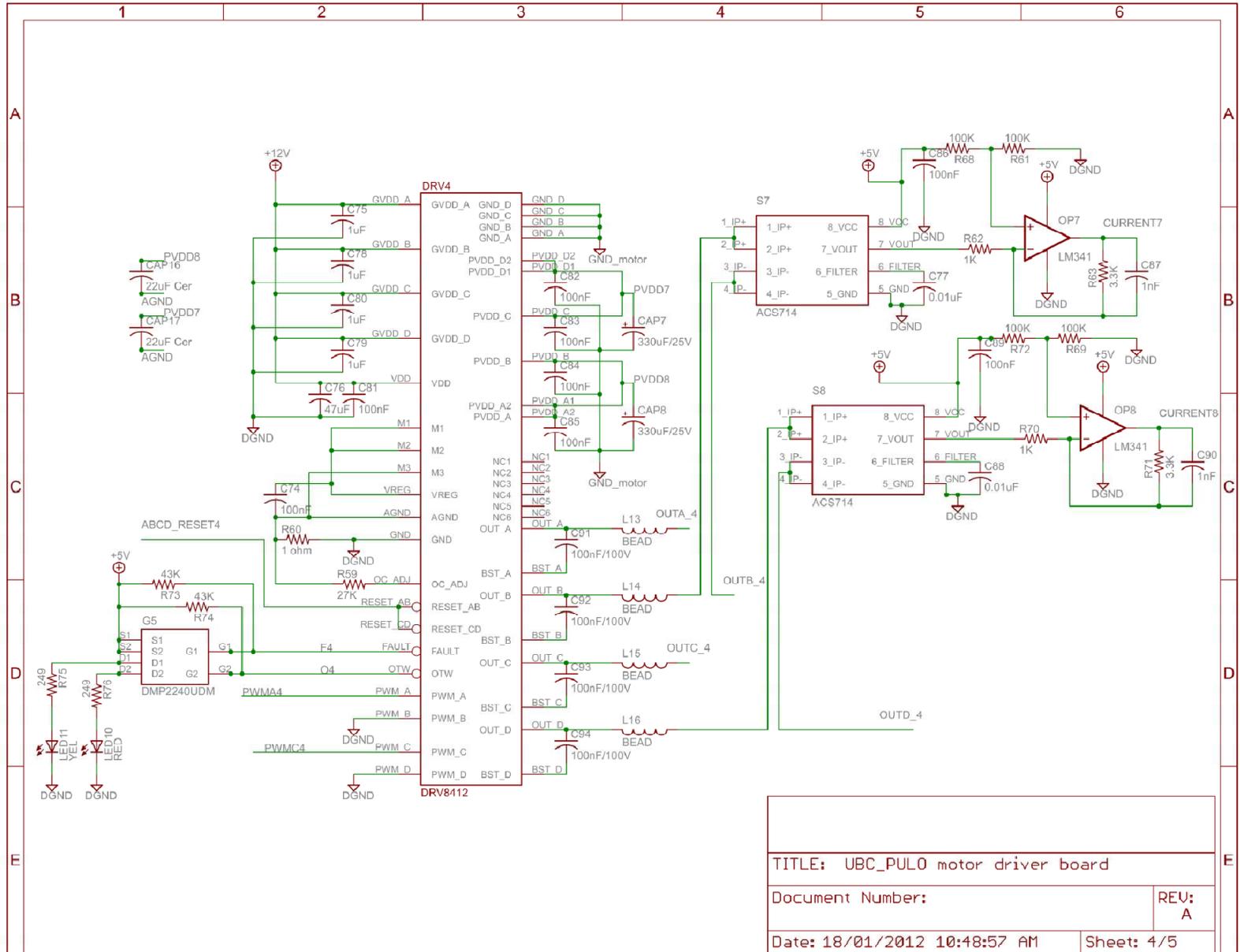
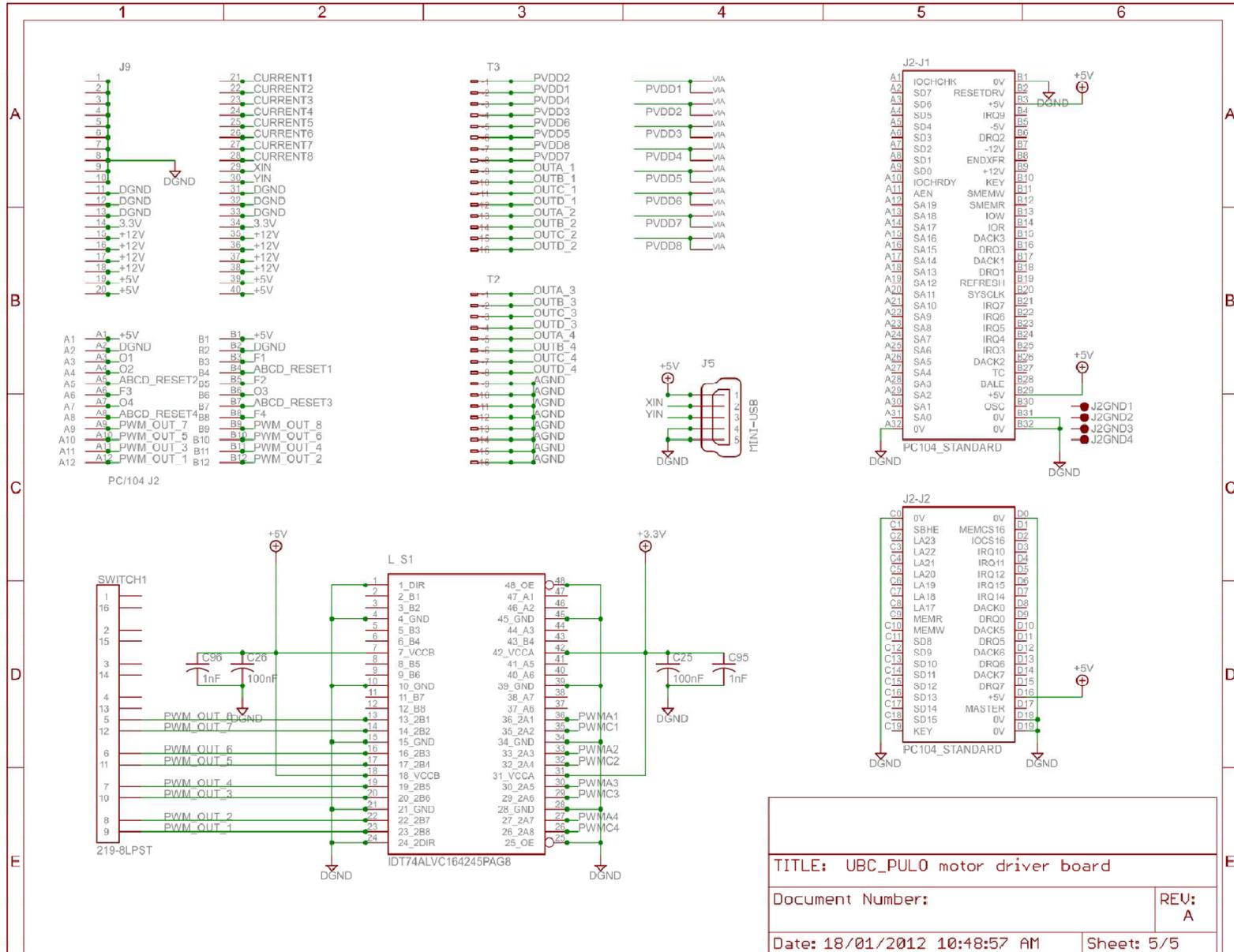


Figure 96. Schematic Diagram for the Motor Driver Board -Sheet 5/5



E.4.2 Motor driver board PCB layout

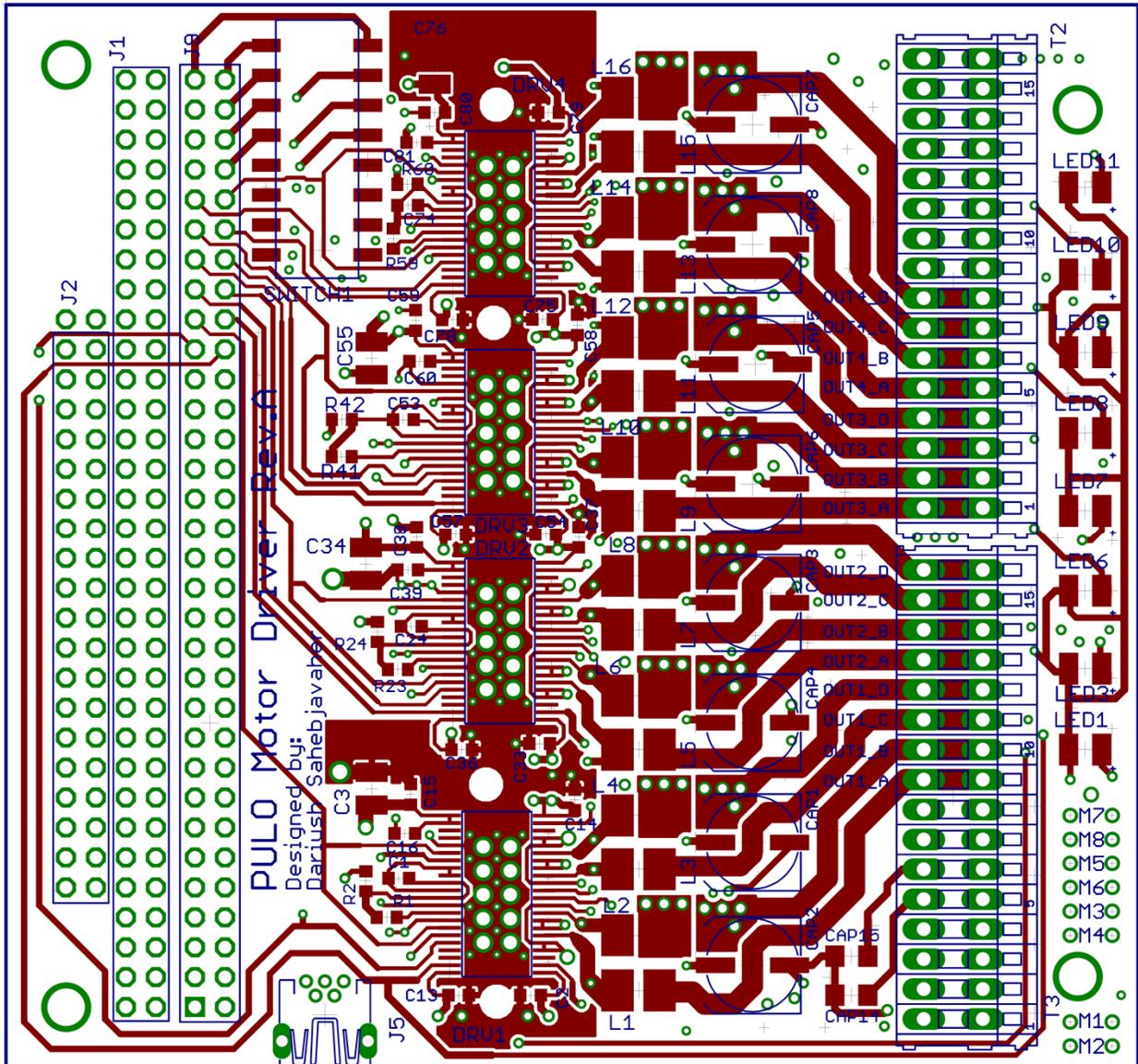


Figure 97. Screen Shot of the Motor Driver Board PCB Layout showing Top Layer (Layer 1) with Top Side Components

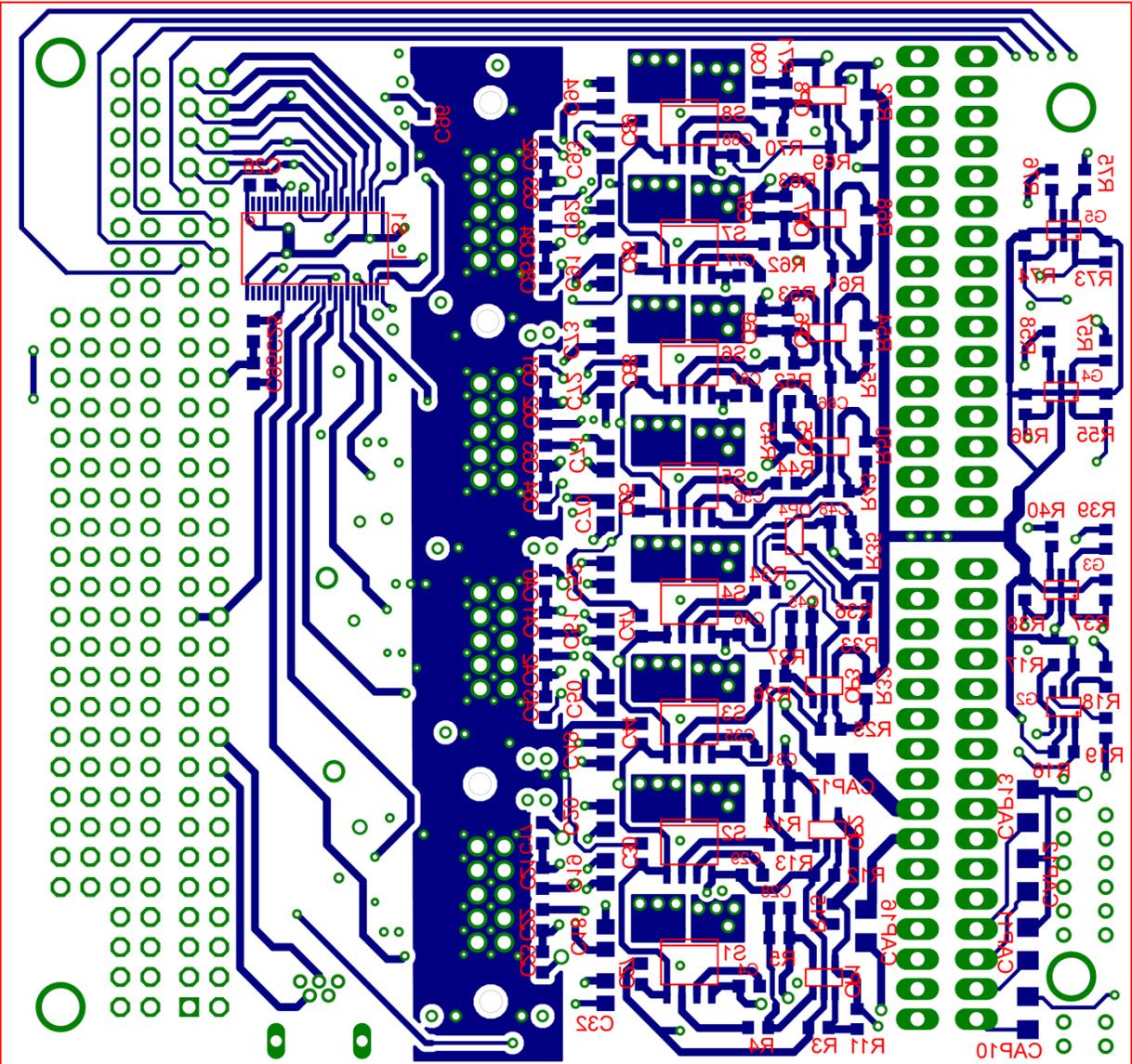


Figure 98. Screen Shot of the Motor Driver Board PCB Layout showing Bottom Layer (Layer 6) with Bottom Side Components

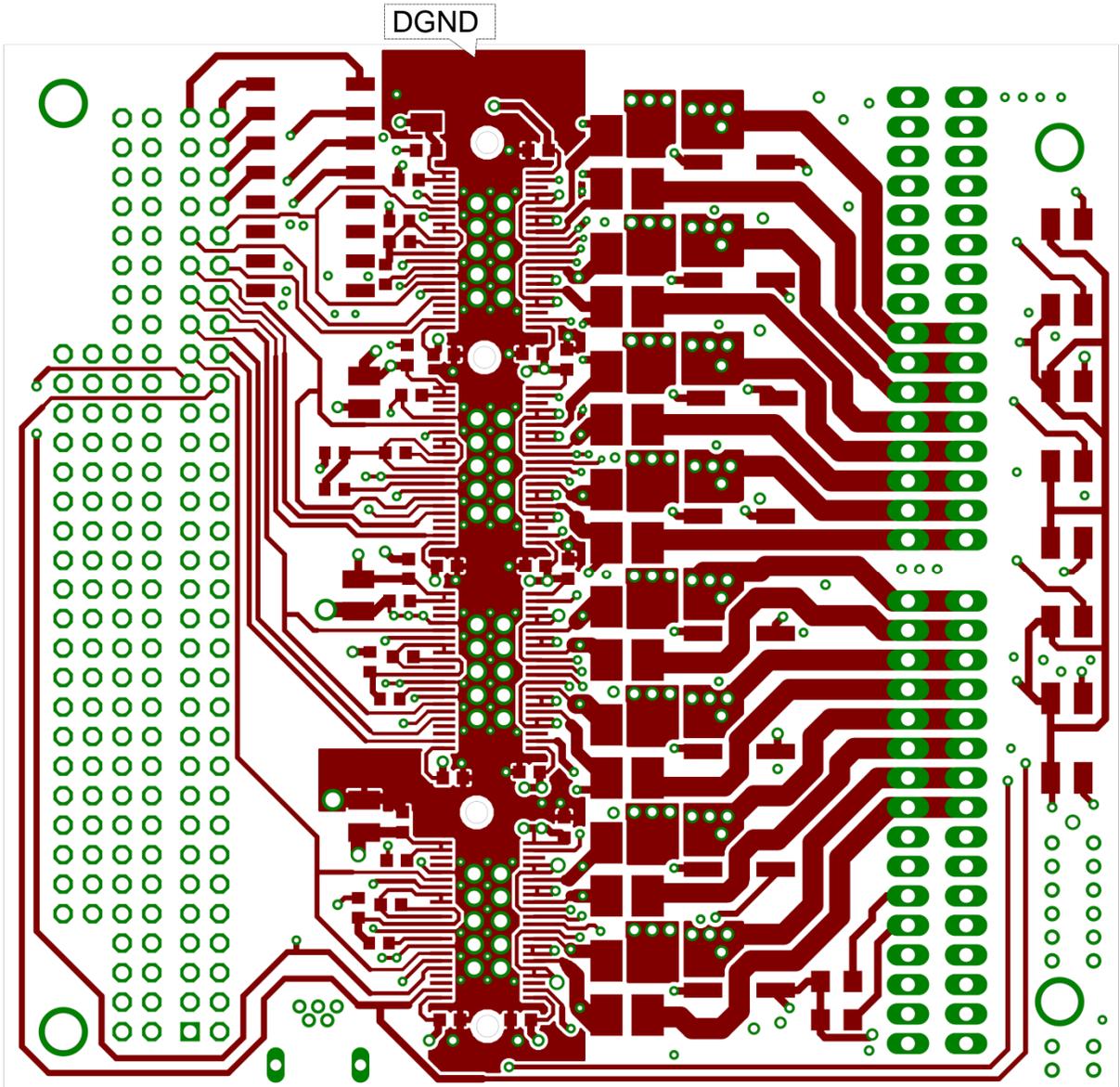


Figure 99. Screen Shot of the Motor Driver Board PCB Layout showing Top Layer (Layer 1)

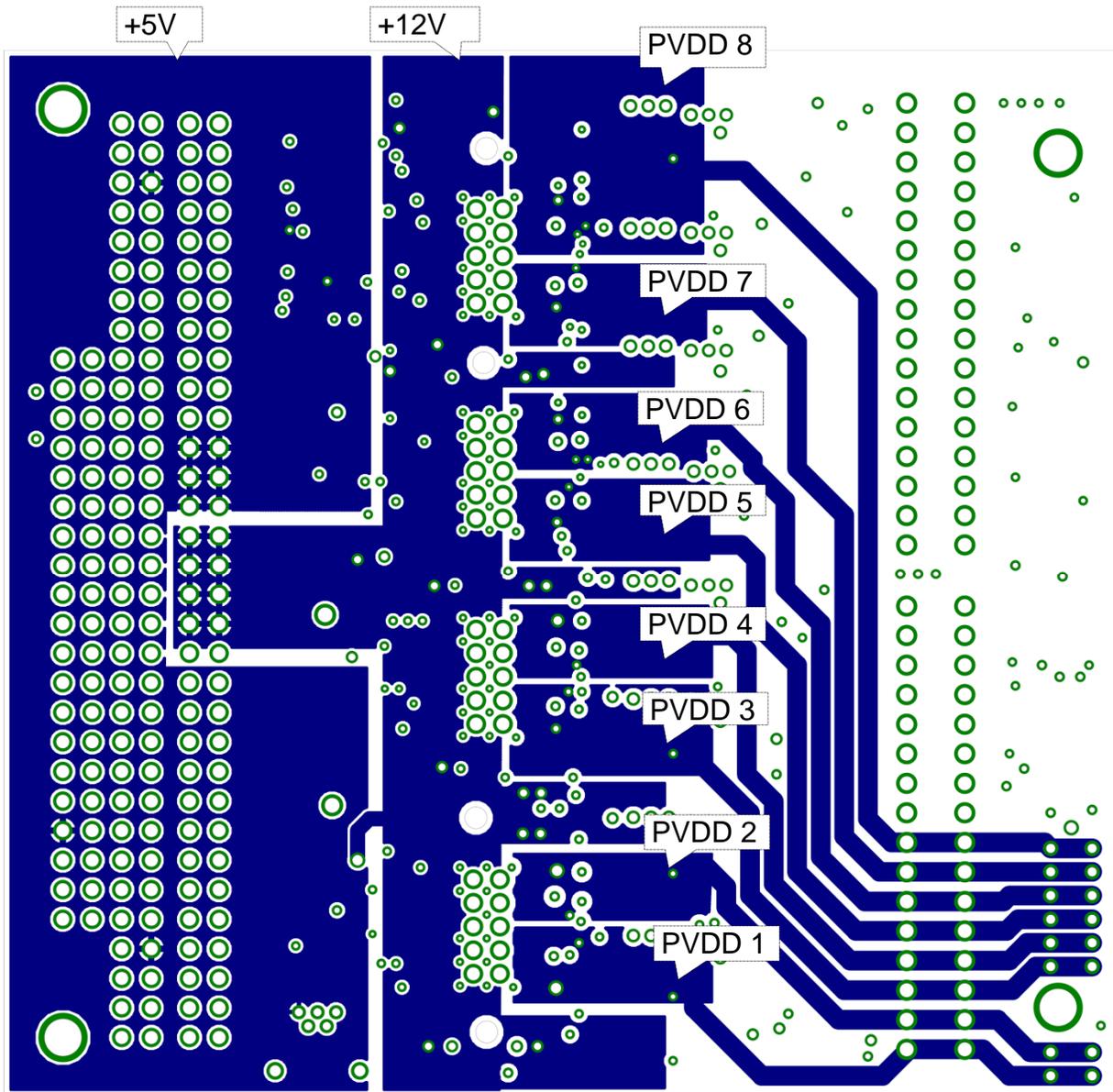


Figure 100. Screen Shot of the Motor Driver Board PCB Layout showing Layer 2

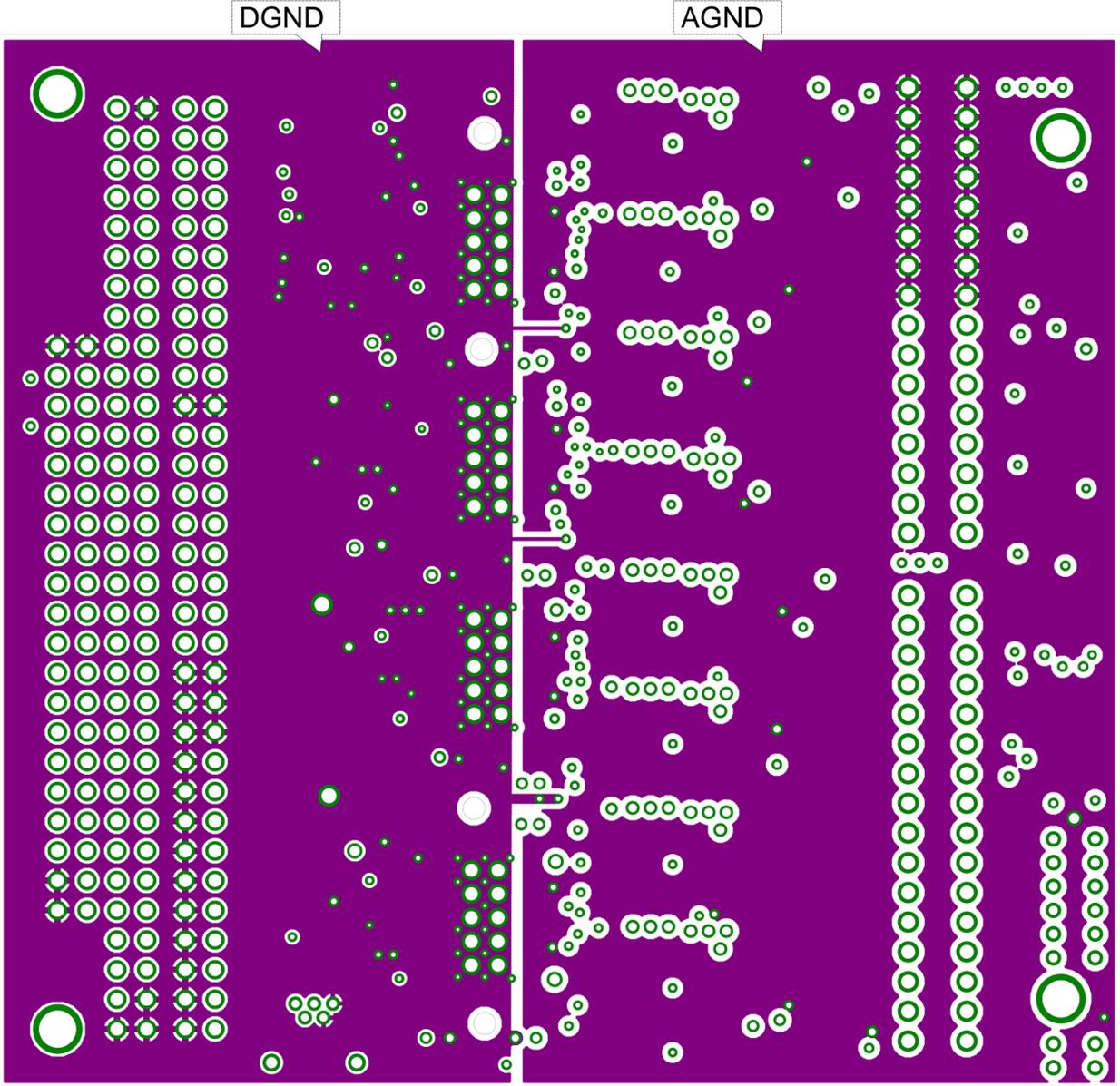


Figure 101. Screen Shot of the Motor Driver Board PCB Layout showing Layer 3

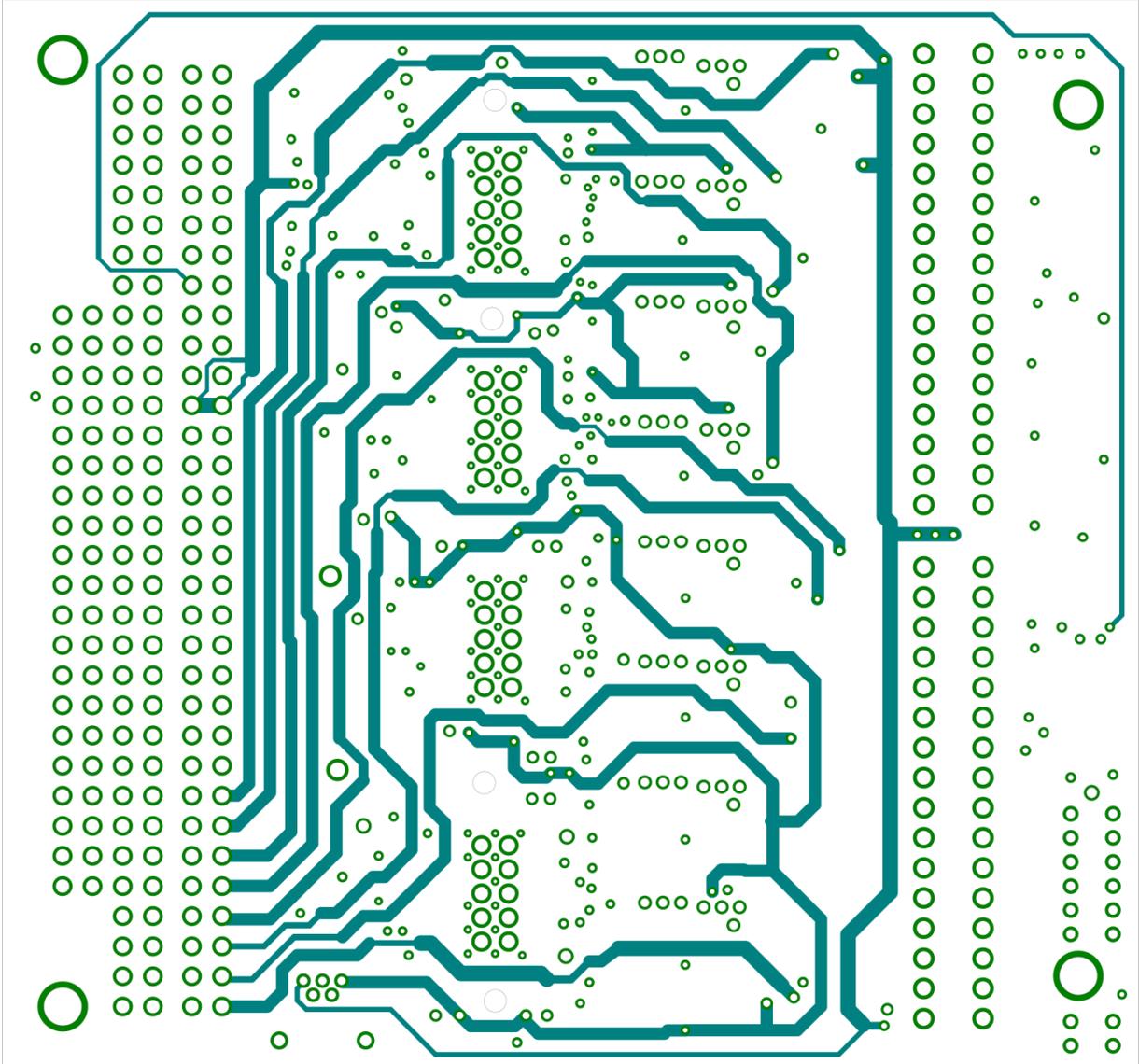


Figure 102. Screen Shot of the Motor Driver Board PCB Layout showing Layer 4

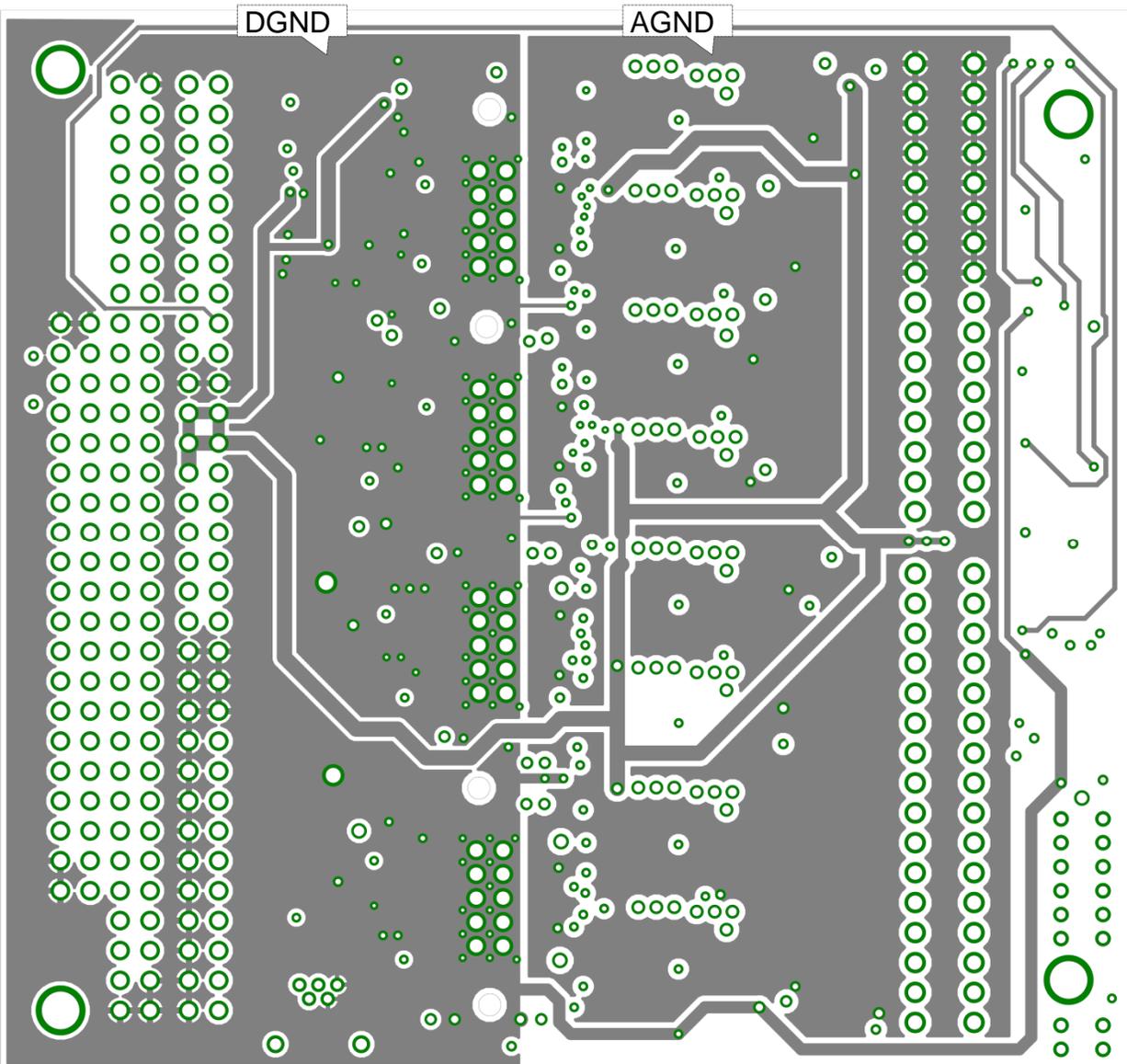


Figure 103. Screen Shot of the Motor Driver Board PCB Layout showing Layer 5

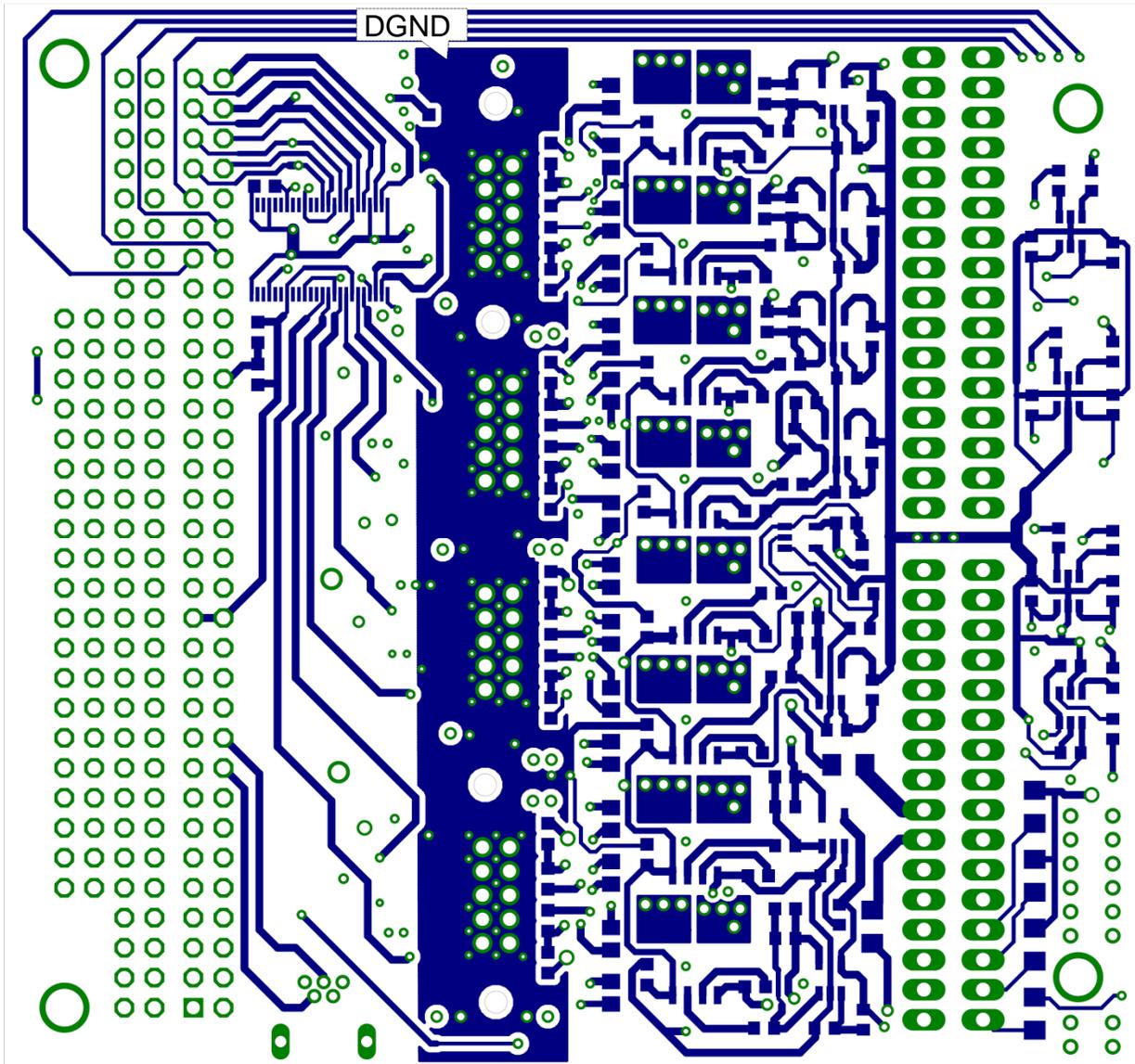


Figure 104. Screen Shot of the Motor Driver Board PCB Layout showing Bottom Layer (Layer 6)

E.4.3 Backplane I/O and Power board schematic

Figure 105. Schematic Diagram for the Backplane Board Sheet 1/2

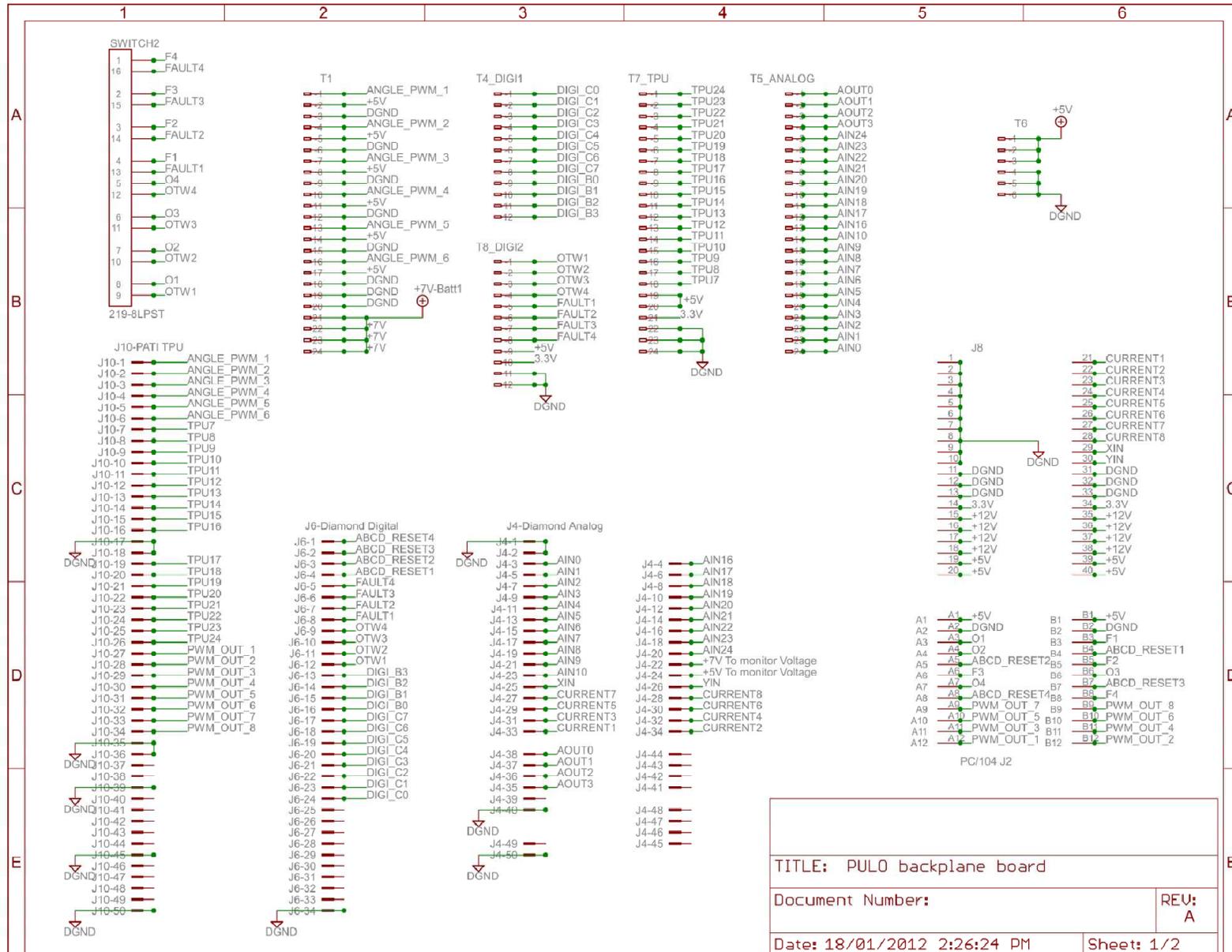
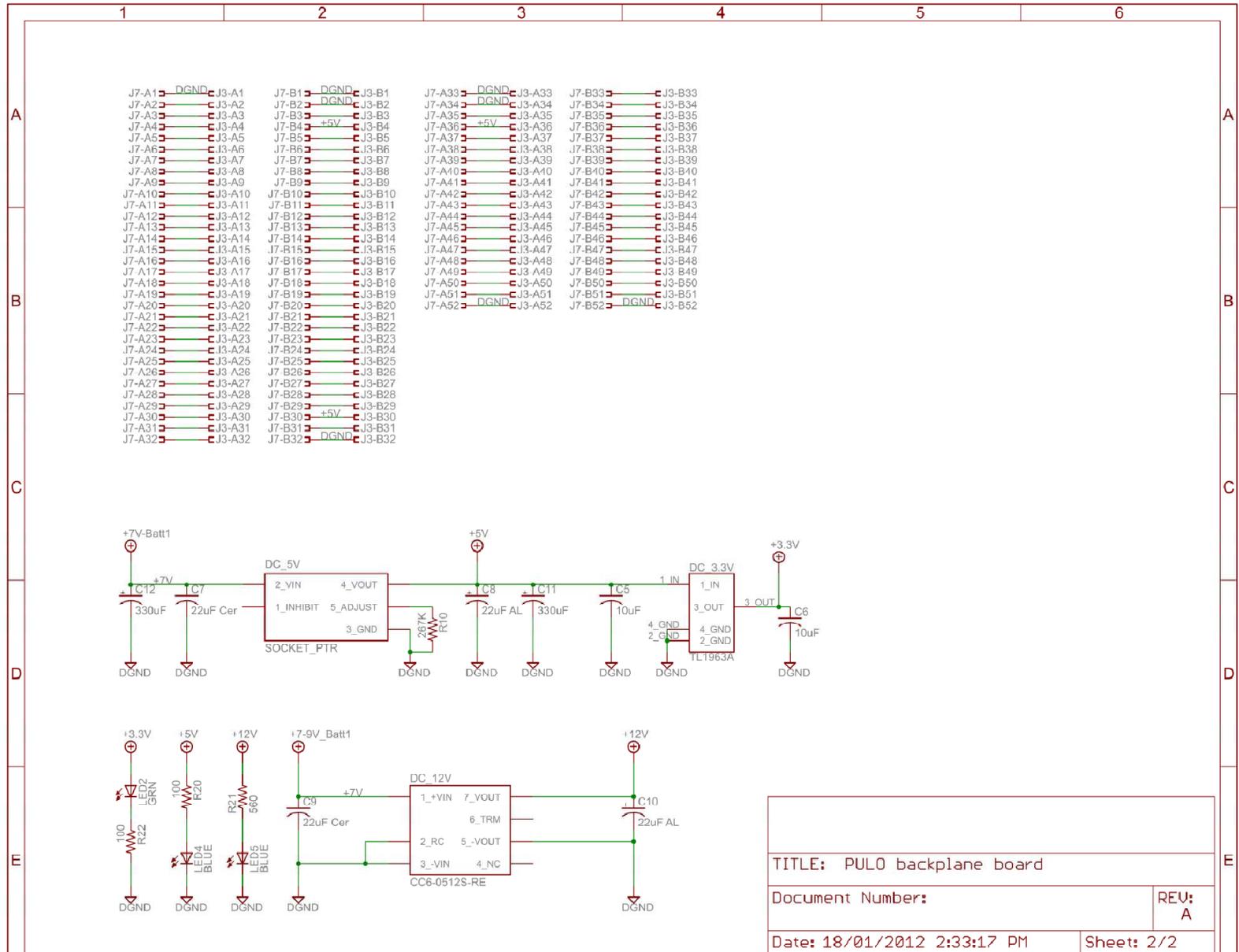


Figure 106. Schematic Diagram for the Backplane Board Sheet 2/2



E.4.4 Backplane I/O and Power board PCB layout

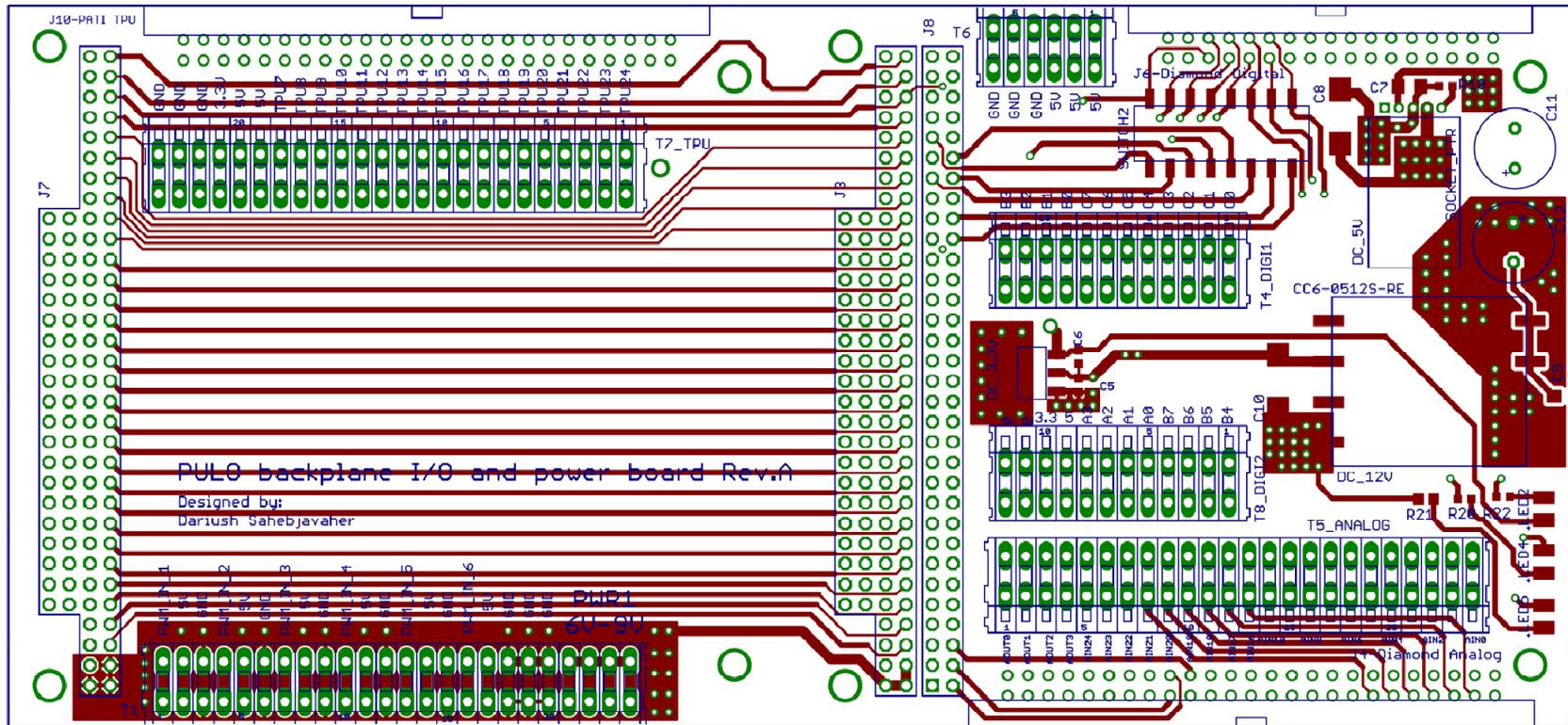


Figure 107. Screen Shot of Backplane Board PCB Layout Showing Top Layer (Layer 1) and Top Side Components

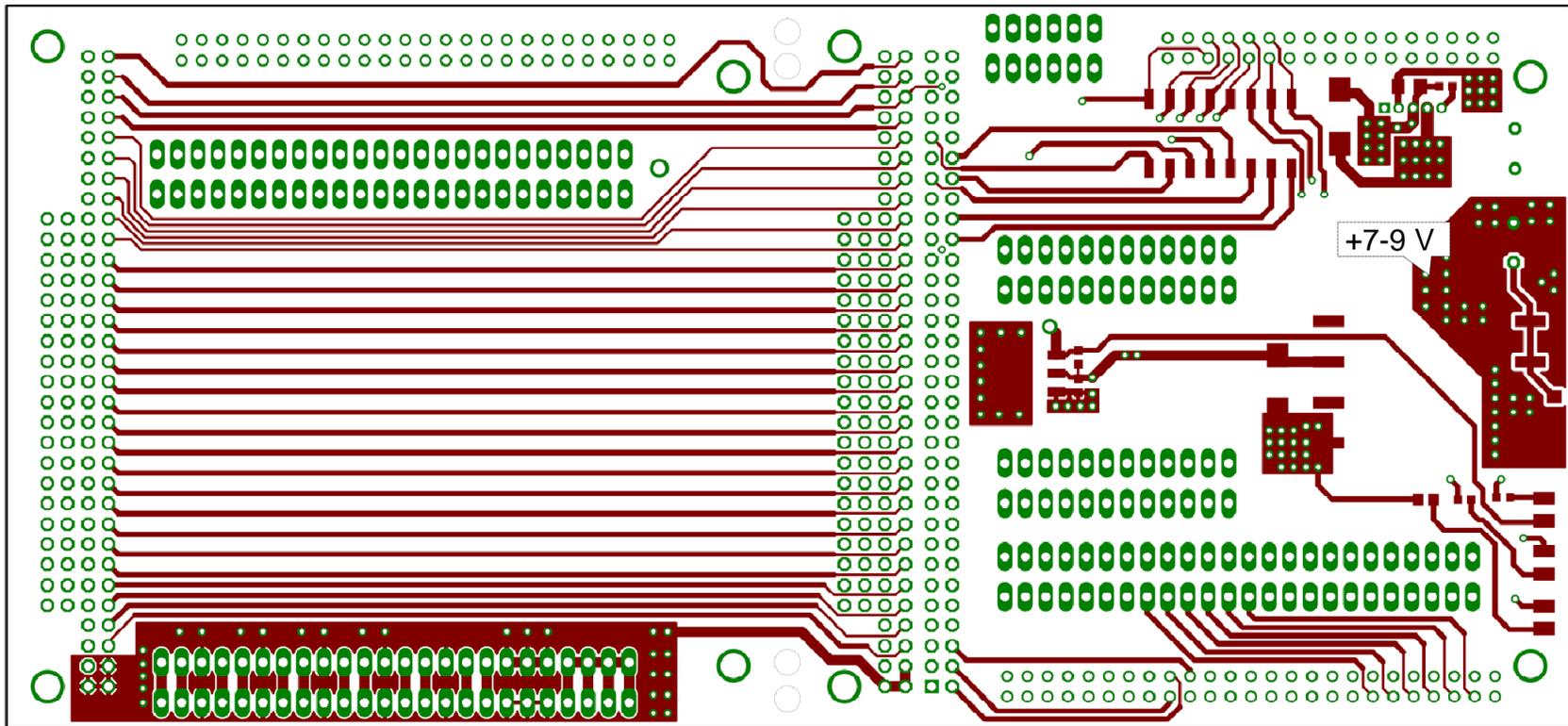


Figure 108. Screen Shot of Backplane Board PCB Layout Showing Top Layer (Layer 1)

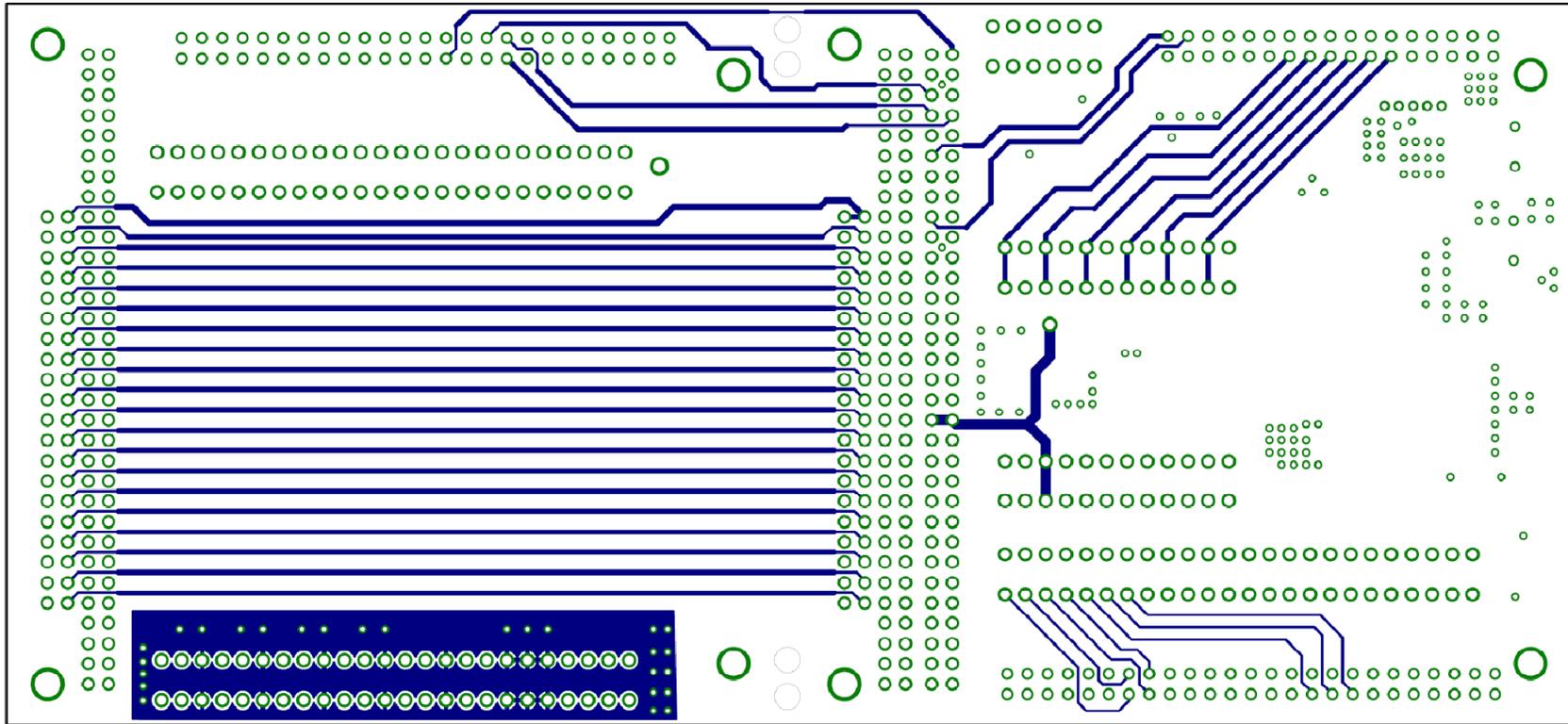


Figure 109. Screen Shot of Backplane Board PCB Layout Showing Layer 2

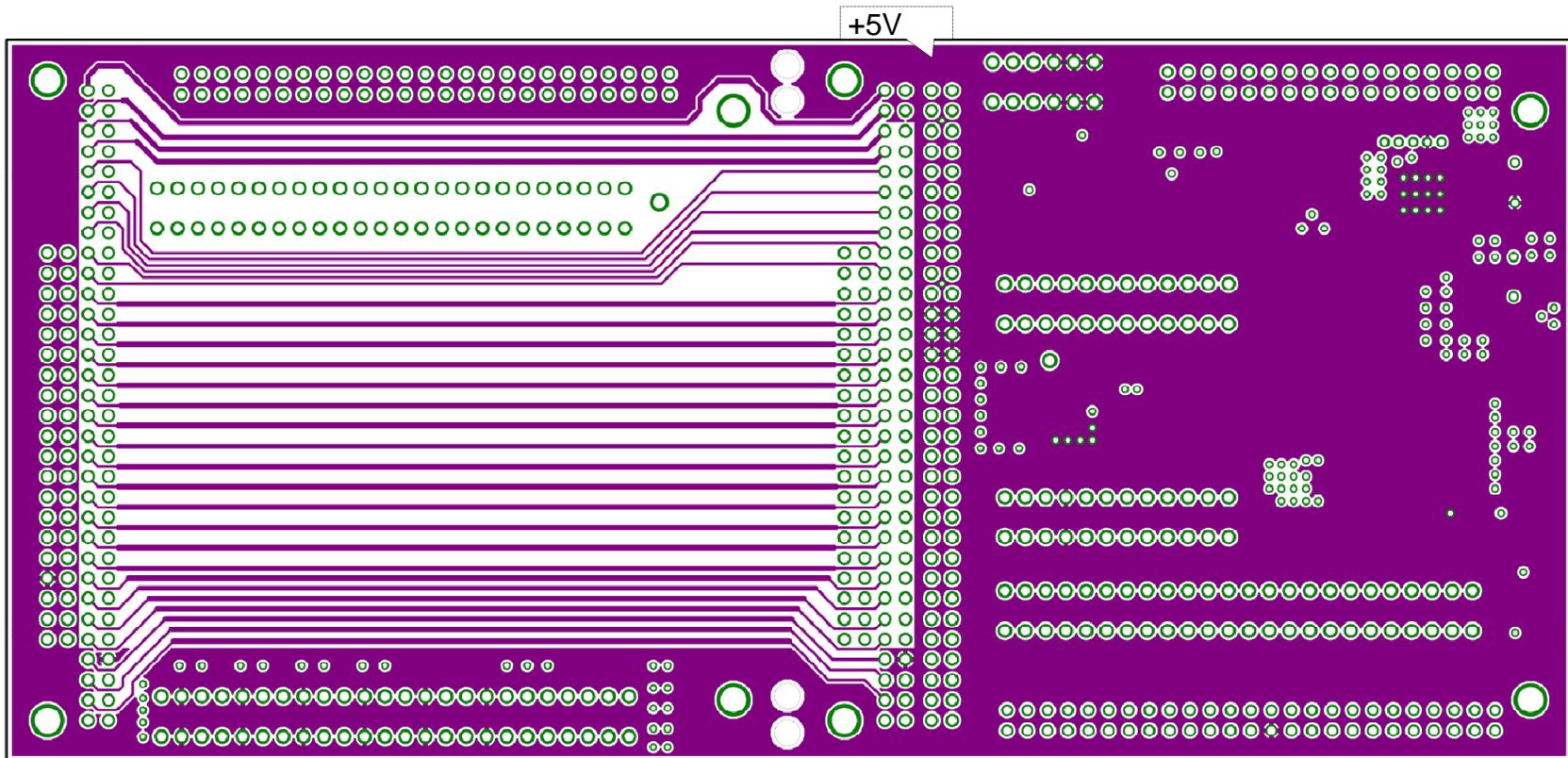


Figure 110. Screen Shot of Backplane Board PCB Layout Showing Layer 3

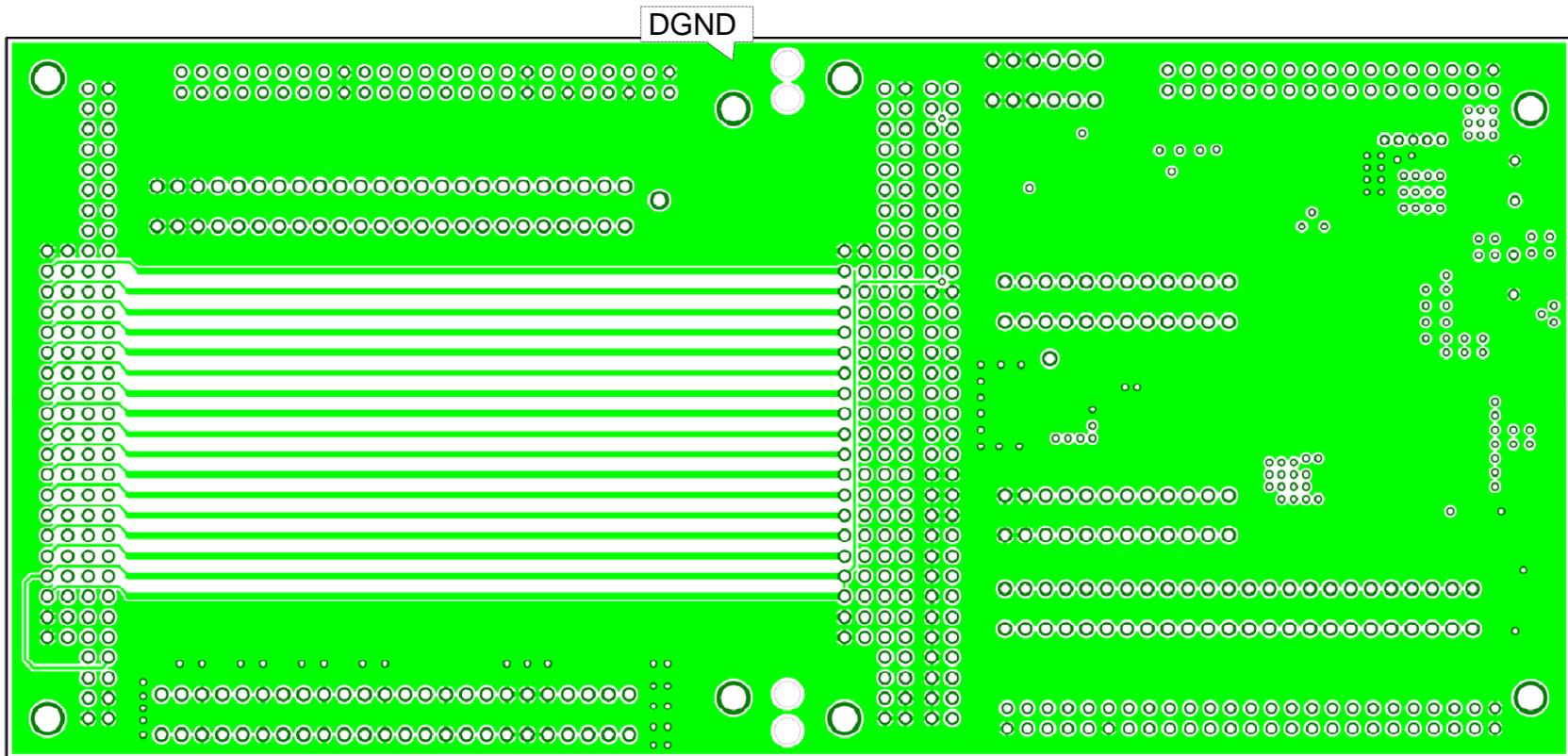


Figure 111. Screen Shot of Backplane Board PCB Layout Showing Layer 4

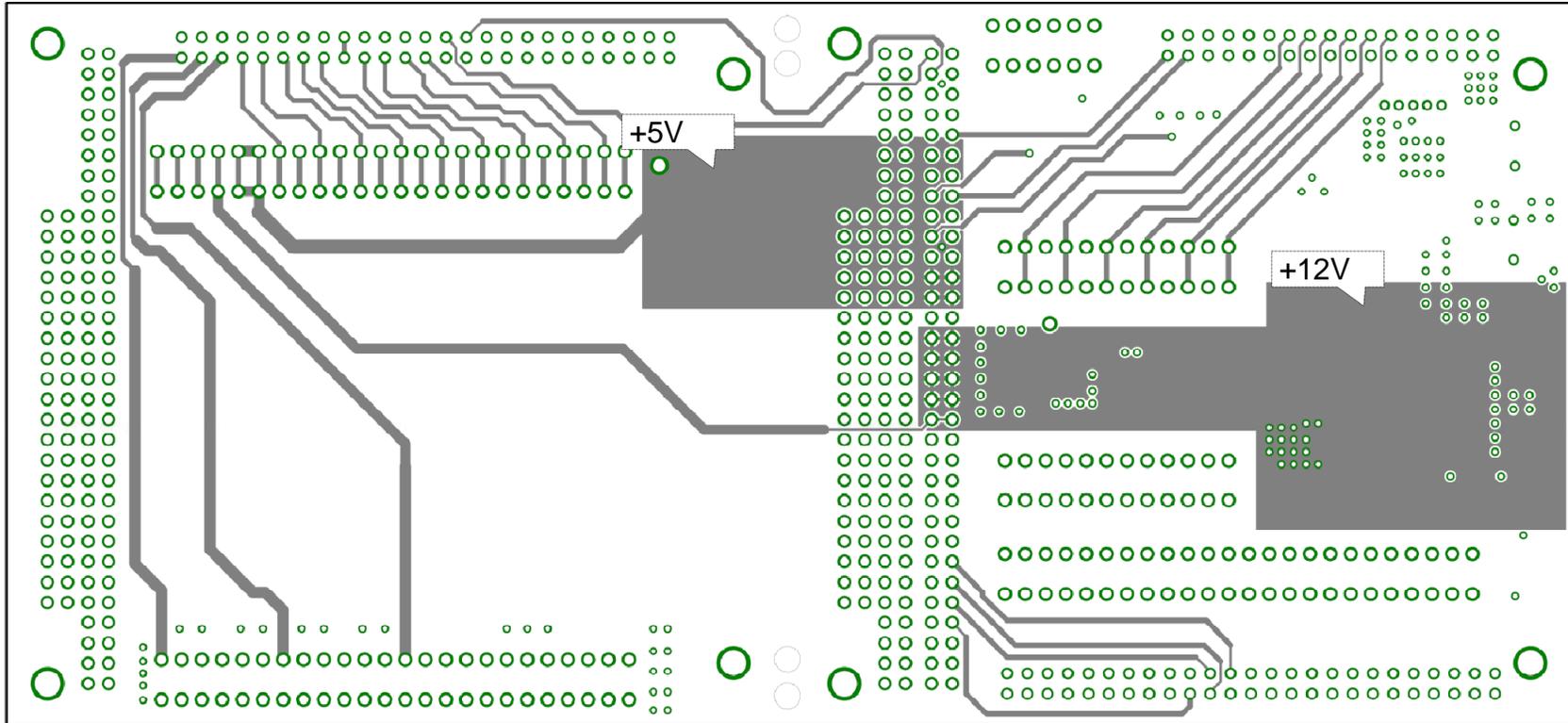


Figure 112. Screen Shot of Backplane Board PCB Layout showing Layer 5

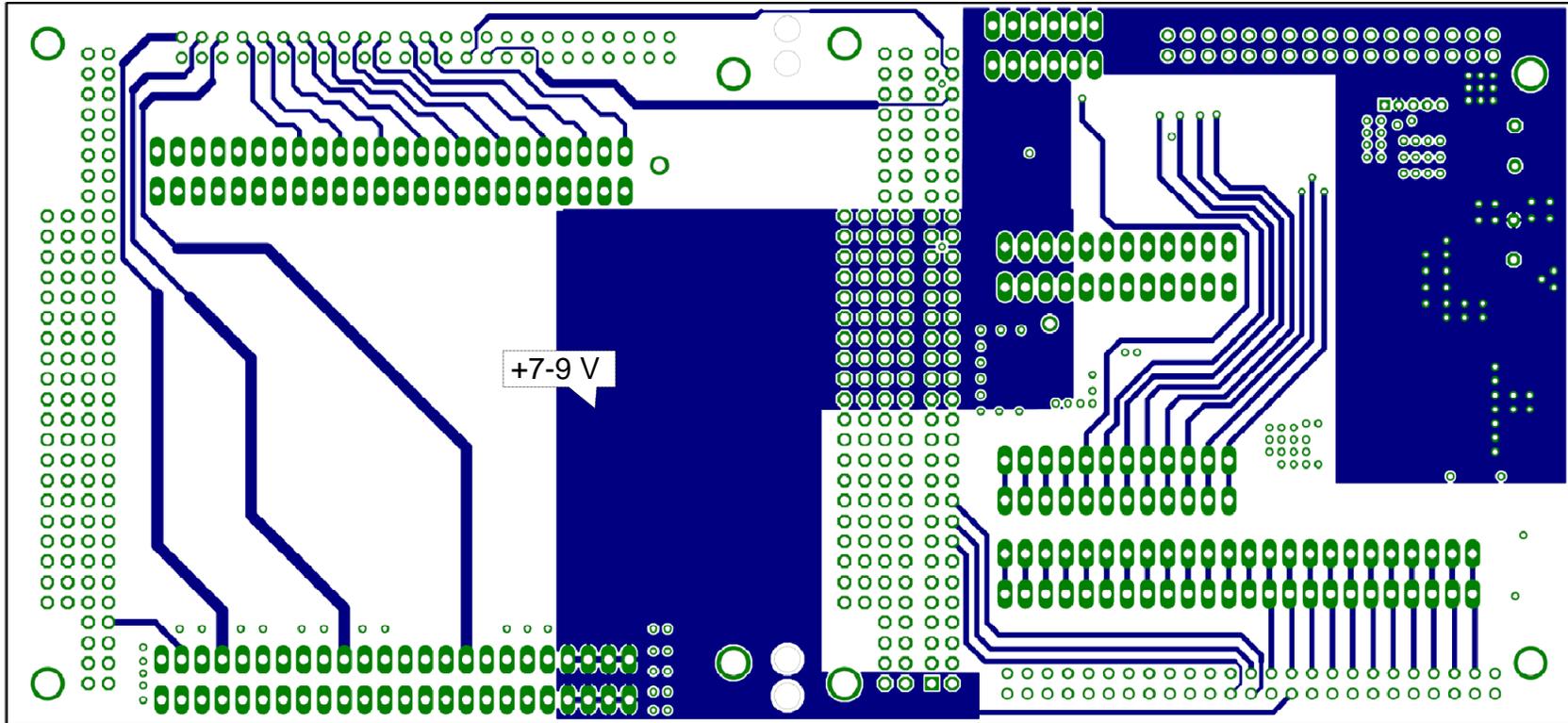


Figure 113. Screen Shot of Backplane Board PCB Layout showing Bottom Layer (Layer 6)

E.8 Bill of material

Table 30. Bill of Material for Custom PCBs

Number	Part Number	Description	Manufacturer	Quantity	Price per unit	Total price	Distributor
Samples from manufacturer							
1	DRV8412	Motor Driver	TI	0	\$ 11.00	\$ -	Digikey
2	PTR08060W	33 Watt power regulator-5V	TI	0	\$ 13.28	\$ -	Digikey
3	AS5040	Magnetic encoder sensor	Austriamicrosystems	0	\$ 6.51	\$ -	
4	AS5040 magnets	Magnets	Austriamicrosystems	0	\$ 2.97	\$ -	
FROM UBC ELECTRICAL SHOP							
	34 WIRE RIBBON CABLE						
	50 WIRE RIBBON CABLE						
	THERMAL PASTE						
	SOLDER PASTE						
	HEAT GUN/SOLDER						
DIGIKEY							
1	ACS714ELCTR-05B-T	Current Sensor	Allegro	8	\$ 4.94	\$ 39.52	Digikey
2	IDT74ALVC164245PAG8	Level shifter	IDT	1	\$ 1.99	\$ 1.99	Digikey
3	GRM188R71E104KA01D	CAP CER .1UF 25V 10% X7R 0603	Murata Electronics North America	100	\$ 0.02	\$ 2.21	Digikey
4	ESR03EZPF2702	603 resistor 27K 1% 0.2W	Rohm Semiconductor	10	\$ 0.14	\$ 1.35	Digikey
5	CRCW06031R00JNEAHP	RES 1.0 OHM .25W 5% 0603 SMD	Vishay/Dale	10	\$ 0.29	\$ 2.94	Digikey
6	GRM188R61C105KA93D	CAP CER 1.0UF 16V 10% X5R 0603	Murata Electronics North America	50	\$ 0.07	\$ 3.51	Digikey
7	F951C476MBAAQ2	CAP TANT 16V 47uF SMD 1210	Nichicon	4	\$ 2.62	\$ 10.48	Digikey
8	35F0121-OSR-10	FERRITE 10A 42 OHMS @ 10 MHZ SMD 1612	Laird-Signal Integrity Products	20	\$ 0.27	\$ 5.36	Digikey
9	SML-LXL1209SIC-TR	LED 636NM SUP RED DOME CLR SMT	Lumex Opto/Components Inc	5	\$ 0.59	\$ 2.95	Digikey

Number	Part Number	Description	Manufacturer	Quantity	Price per unit	Total price	Distributor
DIGIKEY (continued)							
10	LTST-C930KSKT	LED Yellow CLEAR 1210 SMD 2V	Lite-On Inc	5	\$ 0.52	\$ 2.60	Digikey
11	RNCP0603FTD249R	RES 249 OHM 1% 0603 SMD	Stackpole Electronics Inc	10	\$ 0.04	\$ 0.43	Digikey
12	LTST-C930KGKT	LED Green CLEAR 1210 SMD 2V	Lite-On Inc	3	\$ 0.52	\$ 1.56	Digikey
13	LTST-C930TBKT	LED BLUE CLEAR 1210 SMD 3.4V	Lite-On Inc	3	\$ 0.59	\$ 1.77	Digikey
14	LM321MF/NOPB	OP AMP SOT23	LM321MF/NOPB	8	\$ 0.89	\$ 7.12	Digikey
15	ESR03EZPF1003	RES 100K OHM 1/5W 1% 0603 SMD	Rohm Semiconductor	20	\$ 0.14	\$ 2.70	Digikey
16	ESR03EZPJ102	RES 1.0K OHM 1/5W 5% 0603 SMD	Rohm Semiconductor	10	\$ 0.09	\$ 0.94	Digikey
17	RGH1608-2C-P-332-B	RES 3.3K OHM 1/6W 0.1% 0603 SMD	Rohm Semiconductor	10	\$ 0.61	\$ 6.08	Digikey
18	ESR03EZPJ622	RES 6.2K OHM 1/5W 5% 0603 SMD	Rohm Semiconductor	10	\$ 0.09	\$ 0.94	Digikey
19	ESR03EZPJ912	RES 9.1K OHM 1/5W 5% 0603 SMD	Rohm Semiconductor	10	\$ 0.09	\$ 0.94	Digikey
20	ESR03EZPF1202	RES 12.0K OHM 1/5W 1% 0603 SMD	Rohm Semiconductor	10	\$ 0.14	\$ 1.35	Digikey
21	ESR03EZPJ153	RES 15K OHM 1/5W 5% 0603 SMD	Rohm Semiconductor	10	\$ 0.09	\$ 0.94	Digikey
22	CC0603KRX7R8BB103	CAP CERAMIC .010UF 25V X7R 0603	Yageo	10	\$ 0.02	\$ 0.24	Digikey
23	RC0603FR-07267KL	RES 267K OHM 1/10W 1% 0603 SMD	Yageo	10	\$ 0.02	\$ 0.16	Digikey
24	EMK316BJ226ML-T	CAP CER 22UF 16V X5R 1206 - input cap	Taiyo Yuden	10	\$ 0.73	\$ 7.30	Digikey
25	EEU-FC1E331	CAP 330UF 25V ELECT FC RADIAL - one at in one at out	Panasonic - ECG	2	\$ 0.57	\$ 1.14	Digikey
26	ECASD61C226M030K00	CAP AL POLY CHIP 22UF 16V SMD-out	Murata Electronics North America	2	\$ 3.43	\$ 6.86	Digikey
27	TL1963A-33DCYR	Linear REG 1.5A 6in to 3.3V out SOT223-4	ti	1	\$ 3.50	\$ 3.50	Digikey
28	LMK107BJ106MALTD	CAP CER 10UF 10V X5R 20% 0603 - one at in one at out	Taiyo Yuden	2	\$ 0.68	\$ 1.36	Digikey
29	CC6-0512SR-E	DC/DC 6W SNGL 5Vin 12Vout SMD	TDK	1	\$ 17.06	\$ 17.06	Digikey
30	DMP2240UDM-7	MOSFET P-CH DUAL 20V 2A SOT-26	Diodes Inc	4	\$ 0.53	\$ 2.12	Digikey
31	ECH-U1C102JX5	CAP .001UF 16V PPS FILM 0603 5%	Panasonic - ECG	10	\$ 0.53	\$ 5.30	Digikey
32	C2012X7R2A104K	CAP CER .10UF 100V X7R 10% 0805	TDK Corporation	20	\$ 0.24	\$ 4.76	Digikey

Number	Part Number	Description	Manufacturer	Quantity	Price per unit	Total price	Distributor
DIGIKEY (continued)							
33	RNCP0603FTD100R	RES 100 OHM 1% 0603 SMD	Stackpole Electronics Inc	6	\$ 0.06	\$ 0.36	Digikey
34	ERJ-P06J561V	RES ANTI-SURGE 560 OHM 5% 0805	Panasonic - ECG	6	\$ 0.26	\$ 1.56	Digikey
35	SFH213-PPPC-D25-ID-BK	CONN SOCKET IDC 50POS W/STR GOLD	Sullins Connector Solutions	4	\$ 1.77	\$ 7.08	Digikey
36	1658621-8	CONN IDC SKT 34POS W/POL 15 GOLD	TE Connectivity	2	\$ 2.96	\$ 5.92	Digikey
37	1-5103310-0	CONN HEADER LOPRO R/A 50POS GOLD	TE Connectivity	2	\$ 4.45	\$ 8.90	Digikey
38	5103310-7	CONN HEADER LOPRO R/A 34POS GOLD	TE Connectivity	1	\$ 3.34	\$ 3.34	Digikey
39	54819-0519	CONN USB RECEPTACLE 5POS RT ANG	Molex Inc	1	\$ 1.82	\$ 1.82	Digikey
40	UX40-MB-5P	CONN PLUG MINI USB2.0 5POS	Hirose Electric Co Ltd	1	\$ 1.45	\$ 1.45	Digikey
41	219-8LPST	SWITCH TAPE SEAL 8 POS SMD	CTS Electrocomponents	2	\$ 1.40	\$ 2.80	Digikey
42	ERJ-3GEYJ433V	PULL-UP-RES 43K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	10	\$ 0.02	\$ 0.20	Digikey
43	UCL1E331MNL1GS	CAP ALUM ELECT 330UF 25V SMD	Nichicon	10	\$ 0.99	\$ 9.94	Digikey
44	ED90488-ND	SOCKET SLDRTAIL SHRINK SIP 21POS (for DC/DC)	Mill-Max Manufacturing Corp.	1	\$ 7.68	\$ 7.68	Digikey
45	M20-6113205	CONN HDR FEMALE .100" 64POS AU (BACKPLANE)	Harwin Inc	1	\$ 5.69	\$ 5.69	Digikey
46	M20-6103205	CONN HDR FEMALE .100" 64POS AU (stackthrough)	Harwin Inc	1	\$ 6.85	\$ 6.85	Digikey
47	SMD291AX-ND	SOLDER PASTE NO-CLEAN 63/37 5CC	Chip Quik Inc	1	\$ 13.51	\$ 13.51	Digikey
48	SCA100T-D02	INCLINOMETER DUAL 1G DIL12 SMD	VTI Technologies	1	\$ 68.46	\$ 68.46	Digikey
49	ETH	TIP REPLACEMENT SCREWDRVR .031"	Apex Tool Group	1	\$ 4.03	\$ 4.03	Digikey
NEWARK							
1	233-406	TERMINAL BLOCK, PCB, 6POS, 28-20AWG	WAGO	1	\$ 2.14	\$ 2.14	Newark
2	1375796-4	PC 104 ISA (J1) 104 pin (for backplane)	TE CONNECTIVITY / AMP	2	\$ 13.55	\$ 27.10	Newark
3	1375795-2	PC 104 ISA (J1) 104 pin (board)	TE CONNECTIVITY / AMP	1	\$ 19.81	\$ 19.81	Newark

Number	Part Number	Description	Manufacturer	Quantity	Price per unit	Total price	Distributor
NEWARK (continued)							
4	233-412	TERMINAL BLOCK, PCB, 12POS, 28-20AWG	WAGO	12	\$ 4.16	\$ 49.92	Newark
5	AMP - 1445251-1	PC 104 spacer J1	AMP	4	\$ 2.23	\$ 8.92	Newark
MCMMASTER							
1	92005A004	1 pack of Machine Screws M1.6		1	\$ 5.86	\$ 5.86	McMAster
2	90591A109	1 pack of M1.6 NUTS		1	\$ 8.10	\$ 8.10	McMAster
OTHER							
1	SIPA06-1	SIP Adapter - 6 pin (for inclenometer)	proto-advantage	1	\$ 0.99	\$ 0.99	proto-advantage