

**A VIRTUAL REALITY SIMULATOR FOR THE
UBC POWERED UPPER LIMB ORTHOSIS**

by

Jidong Zhao

B.A.Sc., Baker College of Flint, 2007

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)

August 2011

© Jidong Zhao, 2011

ABSTRACT

An orthosis, an assistive device to be worn by the user, has been proposed as an option for regaining arm function necessary for performing high priority daily living tasks. Since the 1960s, researchers and engineers have been developing assistive devices in a variety of different forms to help people with weak or flail arms. The goal of the UBC Powered Upper Limb Orthosis (PULO) is to design a highly functional orthotic device to support and restore arm function to one entire arm of a person who has severe weakness or paralysis in both upper limbs. A significant quantity of research work was conducted previously by researchers and students under the supervision of Dr. Douglas Romilly. Currently, a prototype exists, but requires further optimization in both the mechanical and control systems.

The objective of this research is to model, test and optimize the current UBC PULO control system via the development of a virtual simulator. To accomplish this goal, required tasks include software selection, Simulink-based modeling of the current control system, physical modeling of the mechanical component and creation of a virtual scene. The developed simulator allows the user to provide input commands through the newly developed user interface devices, with the output motion of the prototype orthosis displayed in a Virtual Reality (VR) environment. The specifically designed simulation has indicated a high level of potential suitability of using the virtual simulator for testing and optimizing the current control system, screening and training of potential users for the UBC PULO.

TABLE OF CONTENTS

ABSTRACT	ii
TABLE OF CONTENTS.....	iii
LIST OF TABLES	vi
LIST OF FIGURES	vii
ACKNOWLEDGEMENTS.....	ix
CHAPTER 1: INTRODUCTION	1
CHAPTER 2: LITERATURE REVIEW	5
2.1 INTRODUCTION.....	5
2.2 DESIGN OF EXISTING POWERED UPPER LIMB ORTHOSES.....	5
2.2.1 SUMMARY	12
2.3 UBC POWERED UPPER LIMB ORTHOSIS.....	13
2.3.1 TARGET USER GROUP	14
2.3.2 IDENTIFICATION OF THE DESIRED TASKS	15
2.3.3 SELECTION OF THE UBC PULO DEGREE OF FREEDOM	16
2.3.4 CURRENT STATUS OF THE UBC PULO.....	18
2.4 DEVELOPMENT OF THE CONTROL SYSTEM.....	18
2.4.1 END-POINT CONTROL	19
2.4.2 USER CONTROL INTERFACES	19
2.4.3 CONTROL STRATEGY.....	23
2.4.4 MICROCONTROLLER	25
2.4.5 SUMMARY	28
2.5 SUMMARY.....	29
CHAPTER 3:	
DEVELOPMENT OF A VIRTUAL SIMULATOR FOR UBC PULO.....	31
3.1 INTRODUCTION.....	31
3.2 SOFTWARE TOOL ASSESSMENT.....	33
3.3 REALIZATION OF THE USER INTERFACE DEVICE INPUT SIGNALS.....	35
3.4 MICROCONTROLLER.....	36

3.4.1 USER CONTROL INTERFACES ALGORITHM	37
3.4.2 CONTROL STRATEGY.....	40
3.4.3 INVERSE KINEMATICS	42
3.5 DRIVE SYSTEM	49
3.6 PHYSICAL MODEL.....	51
3.7 VIRTUAL ENVIRONMENT.....	54
3.8 SUMMARY.....	57
CHAPTER 4: SIMULATION.....	59
4.1 INTRODUCTION.....	59
4.2 DESIGN OF THE VIRTUAL TASK.....	59
4.3 SIMULATION OF THE VIRTUAL TASK.....	60
4.3.1 SIMULATION STAGE ONE	61
4.3.2 SIMULATION STAGE TWO	68
4.3.3 SIMULATION STAGE THREE	72
4.4 SUMMARY.....	76
CHAPTER 5: POTENTIAL OPTIMIZAITON	77
5.1 INTRODUCTION.....	77
5.2 INTERPRETATION OF USER'S INTENTION.....	78
5.3 TESTING OF THE INVERSE KINEMATICS ALGORITHM.....	89
5.4 TUNING OF THE PID CONTROLLER.....	92
5.5 SUMMARY.....	97
CHAPTER 6: DISCUSSION	99
6.1 INTRODUCTION.....	99
6.2 ASSUMPTIONS	99
6.3 LIMITATIONS.....	101
6.4 CAPABILITY OF INTEGRATING FUTURE HARDWARE DESIGNS....	102
6.5 SUMMARY.....	105
CHAPTER 7: CONCLUSIONS AND RECOMMENDATIONS.....	107
7.1 INTRODUCTION.....	107
7.2 REVIEW OF POWERED UPPER LIMB ORTHOSIS	107

7.3 DEVELOPMENT OF THE VIRTUAL SIMULATOR	108
7.4 SIMULATED TASKS	110
7.5 OPTIMIZATION FOR THE UBC PULO CONTROL SYSTEM.....	111
7.6 CAPABILITY OF INTEGRATING THE FUTURE HARDWARE DESIGNS	112
7.7 SUMMARY	113
7.8 RECOMMENDED FUTURE WORK.....	113
REFERENCES	115
APPENDIX: UBC PULO FORWARD KINEMATICS	119

LIST OF TABLES

Table 2.1: Desired Tasks.....	15
Table 3.1: Potential Software Tools.....	34
Table 3.2: Control Features of the Head and Shoulder Controllers.....	37
Table 3.3: Definition of D-H Parameters.....	45
Table 3.4: DC Motor Parameters.....	50
Table 3.5: Definition of the Important SimMechanics Blocks [19].....	53
Table 4.1: User's Inputs and Corresponding Arm Motions in Orientation Mode.....	63
Table 4.2: Simulated Control Features in Stage One.....	67
Table 4.3: Simulated Control Features in Stage Two.....	72
Table 5.1: Effect of the Counter Frequency Setting.....	80
Table 5.2: Experimental Results for Defining Neutral Zone Forward/Backward.....	82
Table 5.3: Experimental Results for Defining Neutral Zone Left/Right.....	82
Table 5.4: Experimental Results for Defining Operation Zone Forward/Backward.....	83
Table 5.5: Experimental Results for Defining Operation Zone Forward/Backward.....	84
Table 5.6: Points for Defining Neutral and Operation Zones.....	84
Table 5.7: Error of the Head Controller Algorithm.....	88
Table 5.8: Error of the Inverse Kinematics.....	91
Table 5.9: P, I, D Gains and Response Characteristics.....	96
Table Appendix: D-H Parameters.....	119

LIST OF FIGURES

Figure 2.1: Rancho Los Amigos Orthosis	6
Figure 2.2: Hugh MacMillan Orthosis.....	7
Figure 2.3: MULOS.....	8
Figure 2.4: CADEN-7	10
Figure 2.5: ARMin III.....	11
Figure 2.6: UBC PULO DOF.....	17
Figure 2.7: UBC PULO Prototype	17
Figure 2.8: Head Controller	21
Figure 2.9: Shoulder Controller Prototype	22
Figure 2.10: Shoulder Controller Inputs	23
Figure 2.11: Control Strategy of the UBC PULO	24
Figure 2.12: Schematic of UBC PULO Control Components.....	29
Figure 3.1: Overall Schematics of the Virtual Simulator.....	32
Figure 3.2: Schematic of the DAQ set-up	36
Figure 3.3: Original Design of the Neutral and Operation Zones	39
Figure 3.4: Defined Shoulder Switch Signal	40
Figure 3.5: State Machine Diagram of the Simulator	41
Figure 3.6: Model of the PULO Arm Prototype	43
Figure 3.7: D-H Coordinate System Layout of the Simulator	45
Figure 3.8: Simulink Model of the DC Motor	50
Figure 3.9: Simulink Model of PID Controller	51
Figure 3.10: SolidWorks Model of the Orthosis Arm	52
Figure 3.11: SimMechanics Model of the Orthosis Arm	52
Figure 3.10: Virtual Scene of the Simulator	56
Figure 4.1: Initial Position of the End-Point in Virtual Environment.....	60
Figure 4.2: An Example of Desired User Input	62
Figure 4.3: Simulated Virtual Arm Motion in Stage One	64
Figure 4.4: Simulated End-Point Path in Stage One.....	65
Figure 4.5: Simulated Joint Angles in Stage One	66
Figure 4.6: End-Point Velocity	66
Figure 4.7: Simulated Virtual Arm Motion in Stage Two	70
Figure 4.8: Simulated End-Point Path in Stage Two.....	71
Figure 4.9: Simulated Joint Angles in Stage Two	71
Figure 4.10: Simulated Virtual Arm Motion in Stage Three.....	74
Figure 4.11: Simulated End-Point Path in Stage Three	75
Figure 4.12: Simulated Joint Angles in Stage Three.....	75
Figure 5.1: Scheme of Over-all Optimization Process	78
Figure 5.2: Quadrature Signals of the Inclometers.....	79
Figure 5.3: Original Design of the Neutral and Operation Zones	81
Figure 5.4: Defined Neutral and Operation Zones	85
Figure 5.5: End-Point Velocity vs. Head Tilt Magnitude.....	86
Figure 5.6: Schematic of the Head Controller Optimization	87

Figure 5.7: Results of the Head Controller Algorithm Optimization.....	88
Figure 5.8: Schematic of the Inverse Kinematics Optimization.....	90
Figure 5.9: Results of the Inverse Kinematics Optimization.....	91
Figure 5.10: Schematic of the Drive System Optimization.....	92
Figure 5.11: Guideline for Tuning P Gain	93
Figure 5.12: Guideline for Tuning I Gain.....	93
Figure 5.13: Guideline for Tuning D Gain	94
Figure 5.14: PID Tuner GUI.....	95
Figure 5.15: Tuned PID Controller Response.....	97
Figure 6.1: DC Motor Model	104

ACKNOWLEDGEMENTS

I wish to thank Dr. Douglas P. Romilly for his continuing support and guidance. I could not finish my study without his patience and understanding. I would also like to thank Dr. Ryozo Nagamune for all his assistance and encouragement. Finally, I would like to give my gratitude to my family members, especially, to my wife, Ally, for taking care of our daughter alone and giving up things to support me to finish this thesis.

CHAPTER 1: INTRODUCTION

Due to disease or injury, people can lose motor function in their upper limbs, making activities that require the use of their arms impossible to perform without external help. Options for assistance can come from either the attendance of caretakers or the use of medical assistive devices. Although the presence of caretakers offers companionship that no medical devices can provide for the user, long term home nursing could be potentially costly and make a person become less independent. Thus, the need for using medical devices to restore arm functions becomes apparent. In contrast to a prosthesis that requires amputation of an existing limb, or a robotic manipulator operated away from the user that isolates them from conducting tasks by using their own limbs, an orthosis as an assistive device attached externally to the affected limb allows users to control the supported limb directly through control interfaces. The value of an orthosis is not only from the functionality that allows users to perform desired tasks, but also the operability that brings comfort, ease of use and joy to users from daily living activities. Among various orthosis designs, a powered upper-limb orthosis has been under development at University of British Columbia (UBC). The program goal is to design and build a highly functional, portable and user-friendly orthosis which is capable of supporting one entire arm of a person with severely impaired or lost motor functions in both upper limbs to perform daily living tasks via user control interface devices.

Based on the design purpose, a powered upper-arm exoskeleton can be categorized into two major groups: therapeutic devices and assistive devices. A few systems from each group will be discussed in terms of the mechanical system and control design. Then, the UBC Powered Upper Limb Orthosis (PULO) will be introduced, and the previous work towards the development of the UBC PULO prototype will be reviewed in Chapter 2. Currently, the prototype exists, however subsequent design challenges have been found in both of the control system and the mechanical system. The interest of this research work focuses on the optimization process of the control system for the UBC PULO. Due to the potential failure of the mechanical components, it is extremely difficult to test the control system for the UBC PULO prototype using the conventional laboratory testing. Therefore, the development of a virtual simulator for the UBC PULO was proposed.

The project objective was therefore defined: to develop a simulator in the virtual reality environment for the current UBC PULO prototype, which provides:

- 1) a design tool for researchers to model, test and optimize the current control system,
- 2) a platform with the capability to integrate future hardware designs,
- 3) a screening and training tool for potential users to practice the control operation.

As a design tool it is essential that the virtual simulator includes and properly models the control algorithms and the control strategy to properly simulate the performance of the UBC PULO. Having the capability to adapt new

hardware designs is a necessary feature for the simulator because an upgrade on the mechanical components and the electronics of the current UBC PULO prototype is expected as an outcome of another current research project. Finally, the researchers can also use the simulator as a screening tool to select suitable users for the UBC PULO, and as a training tool for these potential users to learn the device operation in a safe and cost effective manner

The current project objective was achieved through the completion of the following tasks:

1. Review of the design of previously developed devices
2. Assessment of software tools
3. Modeling of the control algorithms
4. Creation of a physical model of the orthosis
5. Design of a virtual scene

Generally, a comprehensive review of previously developed devices, including the UBC PULO, was deemed necessary to the research to define the state of the art in the orthosis designs, and for the author to better understand the development process of the UBC PULO and its design specifications. In order to develop the simulator, selecting a feasible software tool that meet the design requirements for this particular project was also an essential task. The control algorithms and the control strategy designed for UBC PULO would be modeled and tested using the selected software. The expected outcome of this task was that the model of the control system in the simulator would include all the functional features designed for the UBC PULO control system. Additionally,

the physical model of the system, including properties of the essential structural components (such as joints and links), was also required to optimize the control system. Finally, the simulated motion of the UBC PULO prototype was to be displayed in a virtual scene while the user provides real time input commands. The detail of each task is included in Chapter 3.

The simulated results of the important features in the control system design and the outcomes while performing a sample task are demonstrated in Chapter 4. The potential orthosis control system optimization process is then discussed. The assumptions made during the simulator design are outlined, and the limitations of the current work are discussed in Chapter 5. Finally, the conclusions from this research and recommendations for future work are discussed in Chapter 6.

In the next chapter, the devices that have been developed or are currently under development are reviewed, and the development process of the UBC PULO is introduced.

CHAPTER 2: LITERATURE REVIEW

2.1 INTRODUCTION

Powered upper limb orthoses represent a subset of assistive robots and have been used in medical applications since the 1960s. In recent years, the design of a powered upper limb orthosis has focused on the needs of both rehabilitation and power assist for physically challenged people. The challenges of designing an acceptable orthosis include both the mechanical system where the concerns are to balance functionality, weight and portability, and the control system where the concerns are to provide ease of operation for the user. To better understand these design challenges and become better informed of the upper limb orthosis technologies currently available prior to this work, a review of the state-of-the-art was conducted and reported here.

To best clarify the information reviewed and work up to the current research, the discussion of this literature review has been divided into three sections: 1) the design of existing orthoses (not including the UBC device), 2) the development of the UBC Powered Upper Limb (PULO) prototype, and more specifically 3) the development of the control system for the UBC PULO.

2.2 DESIGN OF EXISTING POWERED UPPER LIMB ORTHOSES

The first attempt at developing a powered upper limb orthosis was the Rancho Los Amigos Hospital orthosis developed by Nickel et al. in 1960s [1,11,12]. This electrically-powered orthosis shown in Figure 2.1 was wheel-chair mounted, and aimed to provide assisted arm motions to severely paralyzed

people. With six degrees-of-freedom (DOF) plus grasp, the Rancho orthosis was capable of producing all basic motions of the human arm except wrist yaw.



Figure 2.1: Rancho Los Amigos Orthosis [11]

Each degree of freedom was controlled by a separate bidirectional tongue switch, which made the operation very tedious. Simple tasks such as pick up an object would require a lot of motion of the tongue. Due to the high cost and the frequency of breakdown of the device, the patient rejection was high [1]. Safety was also a concern because of the lack of sensory feedback among most of the patients.

In contrast to the multi-degree of freedom wheel-chair mounted orthosis, a simpler less functional portable orthosis with only one DOF plus grasp was developed by the Hugh MacMillan Rehabilitation Centre (HMRC) in 1987 [12]. The primary goal of this device was to assist people with severe upper limb weakness to eat. This orthotic device was controlled by electromyogram (EMG) signals from the frontalis muscles (located above the eyebrows) and the

supported elbow motion was actuated through a powered timing belt-driven winch unit (see Figure 2.2).

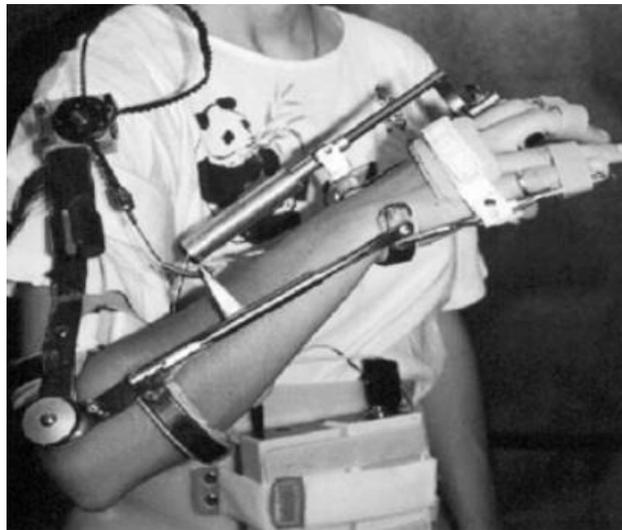


Figure 2.2: Hugh MacMillan Orthosis [12]

A portable device such as the HMRC orthosis allowed the user to take the advantage of having power assisted arm motion without being confined to a wheelchair. However, the control method employed in HMRC orthosis, i.e. myoelectric control with signals from the forehead frontalis muscles alternately being used to activate the hand or the arm, had the major drawback because the system was difficult and tiring to use reliably [33]. Additionally, since the orthosis was designed primarily to allow the user to eat, the device functionality as well as the difficulties in the control system limited its widespread use.

In 1997, GR Johnson et al at the University of Newcastle-upon-Tyne in the United Kingdom developed a Motorized Upper-Limb Orthotic System (MULOS) shown in Figure 2.3 [3,4,13]. The design of MULOS concentrated on the development of three control modes:

1. As an assistive device attached directly to the arm to provide controlled movement for people with severe disability.
2. As a therapeutic robot with continuous passive motion to treat injured joints.
3. As an exercise device for elderly people or those recovering from injury or surgery to strengthen motor functions.

A mechanical design aspect of the MULOS which differentiated it from previous devices was the kinematical equivalent mechanism at the orthosis shoulder joint having a series of three revolute joints with intersecting axes. This shoulder mechanism design aligned the center of rotation of the exoskeleton shoulder coincident with that of the human shoulder. The drive system was built by using cable drives to provide a high-stiffness, low-mass power transmission at the shoulder, smaller size motors and bevel gearbox to reduce design complexity at elbow joint, and a self-contained assembly for pronation/supination motion [3,13].

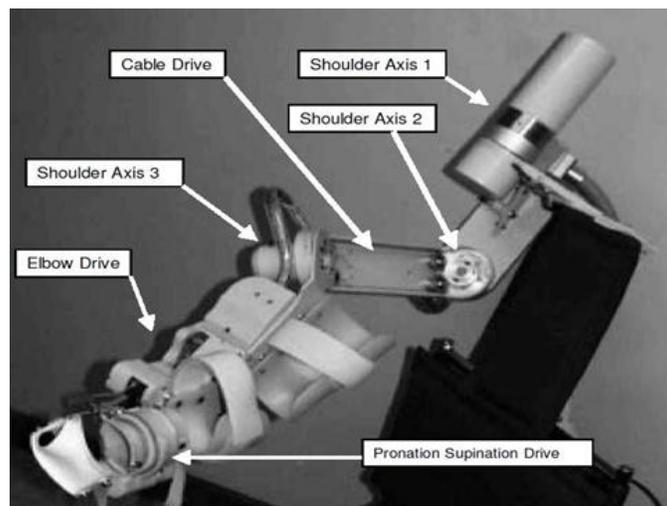


Figure 2.3: MULOS [13]

The operation of the MULOS is achieved using a joystick for user to provide input commands and a controlling computer mounted on the back of the wheelchair. The controlling computer provides both low-level proportional-integrated-derivative controller (PID) of the electric motors and necessary kinematics computations. A four DOF joystick is utilized as the user control interface for the MULOS, and located on the opposite side of the orthosis arm. Since one of the goals of MULOS is to assist people with severe disabilities, and who might have two paralyzed arms, then a hand-operated joystick can not be utilized as an interface. Therefore, speech-operated control was investigated as the potential control interface. Unfortunately, further development of MULOS with a speech-operated controller has not yet been reported.

In 2006 the CADEN-7, (i.e. Cable-Actuated Dexterous Exoskeleton for Neurorehabilitation (CADEN)), a frame mounted orthosis [6,32,38] shown in Figure 2.4, was designed by Perry and Rosen with seven DOF for use by disabled subjects suffering from various neurological disabilities, such as stroke, spinal cord injury, muscular dystrophies, and other neurodegenerative disorders. The mechanical design of CADEN-7 focused on a mechanical human-machine interface (mHMIs) with open human-robot attachment for both upper and lower arm segments, which allows user to be easily attached to the device. A cable-driven design was selected due to its strength in transmitting loads over distance without the friction or backlash. To ensure the user's safety during operation,

safety precautions were implemented on three levels including mechanical stoppers, electrical shut-down switches and software programs [32].

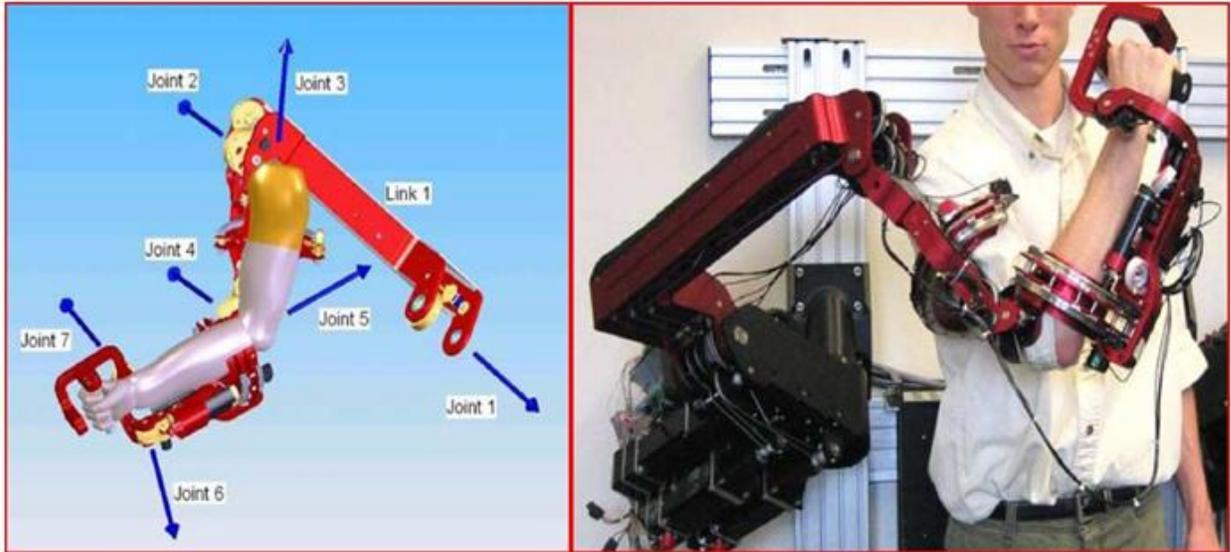


Figure 2.4: CADEN-7 [32]

The control system of CADEN-7 processes the obtained surface EMG signals and sends the information to the myoprocessor, (i.e. a set of computational models of a human muscle predicting joint torques in real-time). The exoskeleton predicts the torque and selects appropriate gain factors for the necessary joints to amplify the user's arm motion [6]. Currently, CADEN-7 is under laboratory testing and the long-term goal is to evaluate the device performance in clinical tests with disabled subjects.

In 2003, an arm rehabilitation robot (ARMin), was developed by Nef and Reiner in Zurich for arm therapy applicable to the task-oriented training of daily living activities in clinics [17,24,25,28,29,30,34]. The ARMin III is the third prototype version of ARMin robot and is equipped with three DOF at shoulder, one DOF at elbow, and two DOF for forearm pronation/supination and wrist

flexion/extension. The ARMin III was designed with a semi-exoskeleton mounted on the frame. In the development of ARMin, a major design challenge was to design a shoulder mechanism such that the exoskeleton could compensate for the humerus head translational movement which occurs during arm elevation motion. Thus, the researchers developed a shoulder actuation principle which described the relationship between the arm elevation angle and vertical translation of the humerus head [17,25,28]. Based on this principle, a shoulder kinematic structure was designed and implemented in the ARMin III to provide comfort for the user during the operation of the system. In the ARMin III shoulder mechanism, a motor was placed at a pre-selected location away from the humerus head, so that when arm elevation motion occurred, the robot humerus head produced a circular movement to approximate the natural motion of the human shoulder. This design allows for the vertical motion of the user's shoulder complex, which is not included in any of the other exoskeleton robots discussed here [29].

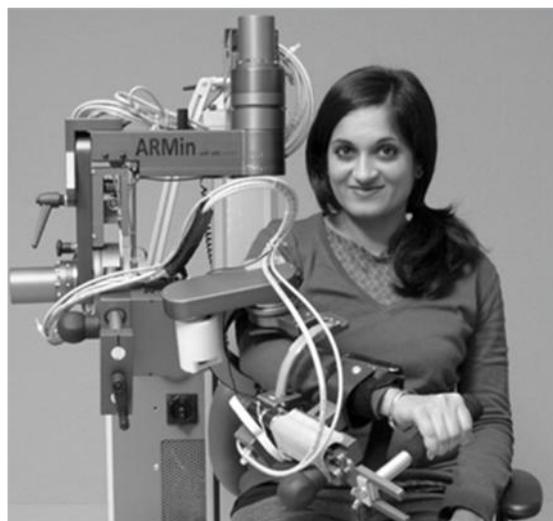


Figure 2.5: ARMin III [17]

The ARMin III provides different control modes including the pre-recorded trajectory mode, the predefined motion therapy mode, the point and reach mode and the patient guided force supporting mode. The user interface ran on a windows machine (ARMin Host). In addition, different training scenarios were designed virtually, and user could look at the graphical display to perform predefined training tasks. Currently, five ARMin III devices have been installed in hospitals in Switzerland and the United States, and are being used on chronic stroke subjects [14,17,34].

2.2.1 SUMMARY

The previously developed powered upper limb orthoses including the Rancho Los Amigos, the HMRC orthosis and the MULO's resulted in limited success because the device was either too complicate to control or suffered from a lack of functionality to assist a user in the wide variety of daily living activities. Although more recently developed devices such as CADEN-7 and ARMin III, improved the ergonomic and controllability aspects of the orthosis design, the added mechanical components greatly increased the weight of the system. Despite the need for a powered upper limb orthosis by a wide variety of users, an acceptable one with functionality and ease of use for home use has yet to be developed. Therefore, the design of the UBC PULO focused on the need of people with both paralyzed arms, and aimed to develop a device with functionality, portability and operability to assist the user to perform the daily living tasks.

2.3 UBC POWERED UPPER LIMB ORTHOSIS

In 1992, Dr. Douglas Romilly and Dr. Cecil Hershler initiated the UBC powered upper-limb orthosis project, and since then, it has been an ongoing a research project within the BioRel laboratory at UBC [1,2,10,35,36,37]. The goal of the research project is to develop a highly functional and wearable powered upper limb orthosis with ease of control to provide motion assist to one arm of people with both severely weakened or paralyzed upper limbs. Over the past years of this project, many researchers, graduate students and staff from University of British Columbia (UBC) and the British Columbia Institute of Technology (BCIT), as well as other medical professionals have made great strides in the development of the current prototype device.

The previous research work for this project includes defining the target users, identifying the higher priority functional tasks, selection of the degree of freedom of the device based on motion analysis studies of the high priority tasks, design and construction of a device prototype, design of the unique control strategy, control system and customized user-control interfaces [34,35]. During the initial laboratory testing phase of the prototype, problems were found in both the mechanical and control systems which prevented this testing from being completed on the physical device [10]. These combined problems and the lack of a properly functioning physical prototype led the researchers to separate the future redesign and optimization initiatives into two research areas: 1) redesign and optimization of the control aspect utilizing a "virtual" device, and 2) redesign and performance quantification of the mechanical system utilizing known

predefined programmed representative task motions. The focus of the current work is related to the first research area dealing with the control issues. The research objective of this work is to develop a simulator in virtual reality environment as a tool to virtually simulate the current UBC PULO design, and to further optimize the control system. To better understand the challenges of this research work and establish the current status of the UBC PULO prototype device, it is essential to briefly review the relevant previous research and development process steps and results, and understand how the control system of the UBC PULO is designed to work.

2.3.1 TARGET USER GROUP

The first important development task was to define the scope of the UBC PULO design by identifying a concise set of user characteristics which ultimately defined the specifications for both the mechanical and control system design. Certain diseases and injuries causing disability in the upper limbs were studied in terms of their medical background and the consequences related to user's abilities [1,37]. The objective in this task was to understand their motion limitations and to recognize the remaining functional capabilities which could make the use of an assistive device possible to them. The potential users of this device as defined previously included people suffering from head trauma, spinal cord injury, muscular dystrophy (MD), muscular sclerosis (MS), post-polio, or stroke [1,37]. Based on the dysfunction expected from these conditions, the specific characteristic descriptions of the target user were further defined for use in the development of UBC PULO as following [1]:

1. two completely flail arms,
2. intact sensation (temperature, pressure, texture),
3. no spasticity,
4. a full range of motion of the joints,
5. full cognitive abilities and,
6. adult-sized.

2.3.2 IDENTIFICATION OF THE DESIRED TASKS

The overall value of the UBC PULO is in its ability to allow the user to gain independence by being able to perform daily living activities that are important to them. Therefore, it was critical to identify the priority daily living tasks which then became the required tasks for the UBC PULO to perform. To identify these tasks, the researchers conducted surveys and interviews with potential users, and five tasks involving the use of arm movement were prioritized based on their importance [1,37,41] (see Table 2.1)

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

Table 2.1: Desired Tasks

Because reaching and picking up an object is integral to many activities, it has the highest priority. The key factors to successful performance of these tasks

are the functional mechanical design and efficient user-friendly operation of the device.

2.3.3 SELECTION OF THE UBC PULO DEGREE OF FREEDOM

To select the optimal degrees of freedom to balance the functionality and design complexity for the UBC PULO, a motion analysis study was conducted to obtain data on the arm motion required to perform the specified tasks. Real motion data obtained by user performing high priority tasks was compared to simulated prototype motion created by a specially developed kinematic simulation program in order to evaluate the required DOF necessary for completion of the tasks. The evaluation result indicated that fixing more than two degree of freedom (i.e. shoulder elevation and wrist yaw) would severely restrict the device functionality and not allow satisfactory completion of the high priority tasks [1,37]. Therefore, the current prototype device was designed and built with fixed shoulder elevation and wrist yaw. The remaining DOF of the UBC PULO are shown in Figure 2.6

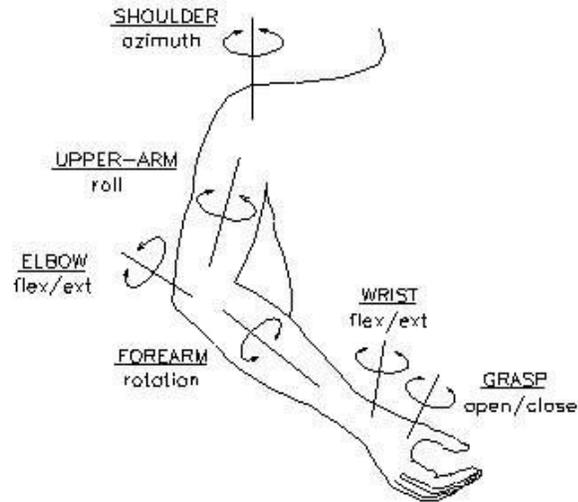


Figure 2.6: UBC PULO DOF [1,2]

The current UBC PULO prototype design has five DOF plus the capability of grasp, i.e. the researchers selected to eliminate the identified two DOF to reduce the complexities of both the mechanical and control systems while still maintaining the required functionality necessary to perform the defined tasks. Upon the DOF for the UBC PULO were selected, development work on both the mechanical design and control system proceeded resulting in construction of the physical prototype for UBC PULO (see Figure 2.7).

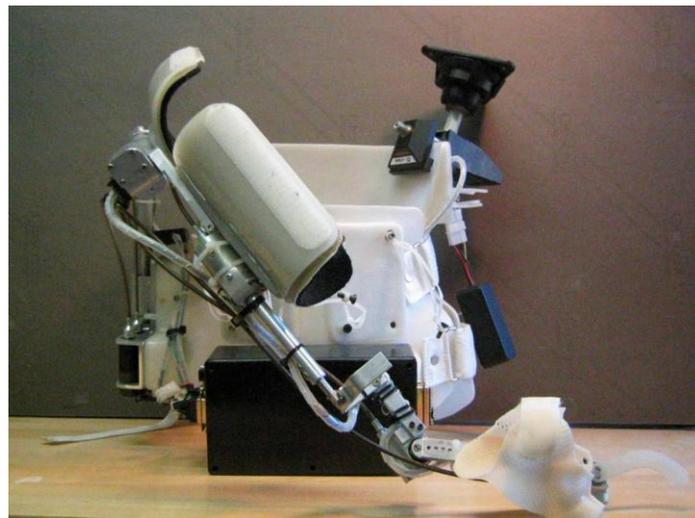


Figure 2.7: UBC PULO Prototype [2,10]

2.3.4 CURRENT STATUS OF THE UBC PULO

After completion of the UBC PULO prototype, initial laboratory testing identified various issues with both the mechanical and electrical system designs. According to the evaluation results [10,36], the issues within the mechanical system existed in the power system, the drive system and the arm supporting structure. Due to the nature of the issues present in the mechanical components, evaluation of the control system was made difficult as these issues typically interfered with the performance evaluation of the control system, leaving the identification of the problem issues of the control system unidentified. Thus the objective tasks of the current research work is to model, test, evaluate and further optimize the control system through the development of a virtual simulator thereby eliminating the initial need for a physical orthosis system. To achieve this objective, it is first necessary to review the development of the UBC PULO control system.

2.4 DEVELOPMENT OF THE CONTROL SYSTEM

To provide an intuitive and user-friendly control system for the UBC PULO, the previous research team investigated different potential control methods and user interface designs, and developed the current UBC PULO control system which includes an intuitive control methodology, suitable user control interfaces, and an efficient control strategy to provide the relevant motion needed for performing the specific high priority tasks [1,2,10,35,36,37].

2.4.1 END-POINT CONTROL

The concept of end-point control in this research work is one in which a specific end-point (in this case the center of the user's hand) is the focus of the control system, being directed in a desired vectored direction and at a varied but controlled speed [2,36]. During the operation to perform a task, the user provides the direction and speed for the hand to move via one of the interface devices, but predominantly the head interface device. The microcontroller computes the required speed for each individual joint based on the input commands. Thus, the joints move concurrently to ensure that the hand moves towards the desired destination. The implementation of end-point control in the UBC PULO design provides the user with efficient control by moving the joints concurrently, and ease of control through simple and intuitive input commands via the user control interfaces, as described in the next section.

2.4.2 USER CONTROL INTERFACES

A major consideration in the design of the control system for the UBC PULO was the development of suitable user control interfaces for people with two flail arms. The previous research team investigated a variety of potential methods including using shoulder motion, muscle activity (i.e. EMG), vision systems, and the head movement for providing input signals [2,35,36]. It was determined from that investigation that head and shoulder movement provide the most suitable input signal source for this device. The potential users of the UBC PULO have a full range of head motion which is unrestricted by the disease or injury. Although the ability of moving the shoulder for the patients varies with

diseases or injuries, most potential users of this device are capable of producing sufficient shoulder motion for the control purpose [2]. Utilization of the head and shoulder movement from the users provides non-invasive and reliable control sources to generate usable control input signals. Thus, head and shoulder control interfaces were subsequently developed to transfer the user's body motion into input commands for PULO operation.

2.4.2.1 HEAD CONTROLLER

The head controller consists of two incremental inclinometers mounted to the side and back of the head via a series of elastic straps and buckles (see Figure 2.8). This user interface device provides the input signal used to control movement of end-point within the X-Y plane (see Figure 2.11) with respect to the world coordinate system, and the rotation of the end-point to achieve the desired orientation. During operation to reposition the end-point of the arm, the head is tilted in the desired X-Y direction of motion, at which time the two digital-based encoder inclinometers monitor the magnitude of the head tilt on both the X and Y axis. The encoder produces a two channel quadrature output, from which the alternating lead and lag between the two channels determines whether the rotation is clockwise or anti-clockwise. The reading of the encoders counts (i.e. the number of the square wave pulses produced), and the speed of pulse generation are proportional to the angle and speed of the head tilting motion. The combined reading of the encoders produces a vector for direction and speed reflecting the user's intention, and is interpreted by the microcontroller for subsequent engaging of the appropriate motors for joint motion [36].

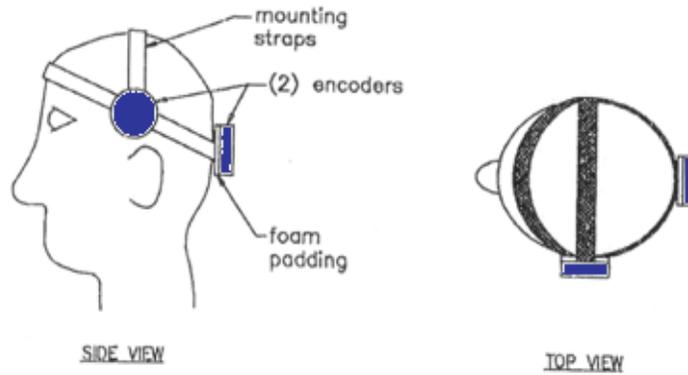


Figure 2.8: Head Controller [2]

2.4.2.2 SHOULDER CONTROLLER

The goal of introducing another control interface is to provide additional control signals through another available body movement from the user to enhance the functionality of the device. Thus, the shoulder controller has been developed to control the end-point in the Z direction as well as to provide additional functional controls [2,10,36]. The shoulder controller measures the vertical motion of the opposite arm (to the one the orthosis is mounted) during a shoulder shrug. The current prototype shoulder controller (see Figure 2.9) consists of two optical switches mounted on a Printed Circuit Board (PCB) and positioned a predetermined distance relative to the range of the user's shoulder motion. A thin aluminum plate is positioned between the ports of the optical switches.

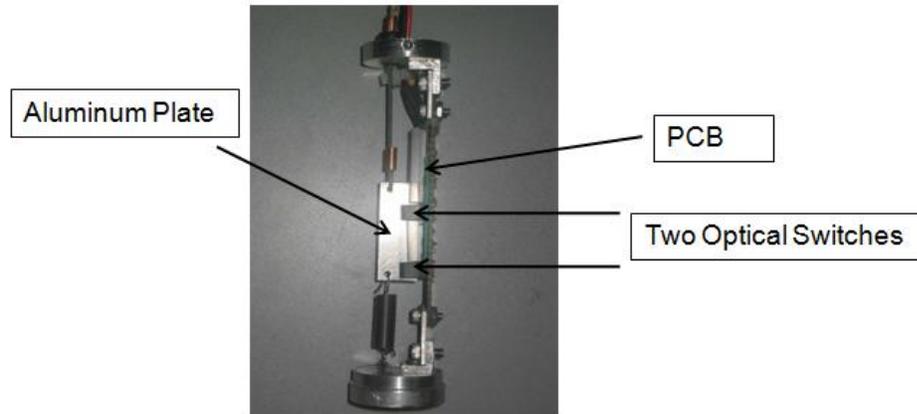


Figure 2.9: Shoulder Controller Prototype [10]

When the user shrugs the shoulder, the switches are activated as the translating plate travels between the switch ports and triggers the optical switches. Although more control signals can be provided by implementing additional optical switches and corresponding thresholds, a lessened travel distance between the switches greatly increases the difficulty for the user to accurately control the input signals, as well as maintain and differentiate between the two levels. Due to the decreased range of shoulder movement in potential users, the shoulder control interface has been limited to signals of two raised states (i.e. HIGH or MED) and the relaxed state (LOW). As a function of the shoulder position, the first (MED) signal is produced when the shoulder is raised above an adjustable threshold and the second (HIGH) signal is generated when the shoulder is raised to nearly full vertical motion (see Figure 2.10).

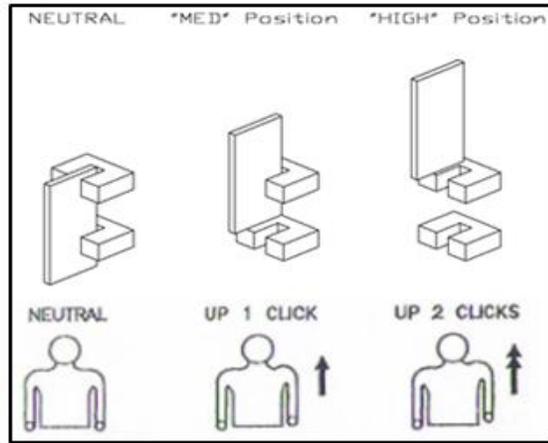


Figure 2.10: Shoulder Controller Inputs [2]

The combination of head and shoulder control interfaces provide control input signals which are utilized in an efficient control strategy as described in the next section.

2.4.3 CONTROL STRATEGY

The overall control strategy developed for the UBC PULO was to provide a means to achieve a high level of device functionality with reduced user complexity in controlling the device to perform the desired high priority tasks. The control strategy is summarized in the flow chart (Figure 2.11) which highlights the development of two device operational modes: i.e. position and orientation. The position mode involves the utilization of three arm joints to provide coarse end-point positioning. As a result of the concurrent motion of these joints (i.e. shoulder azimuth, upper-arm roll and elbow flexion/extension) to the desired position, the hand may not always be in the correct orientation for grasp. Thus, switching to the orientation mode allows the user to control various individual

joints (i.e. forearm, wrist and grasp) for fine adjustments of the end-point to achieve a comfortable hand orientation prior to grasp to perform the desired task.

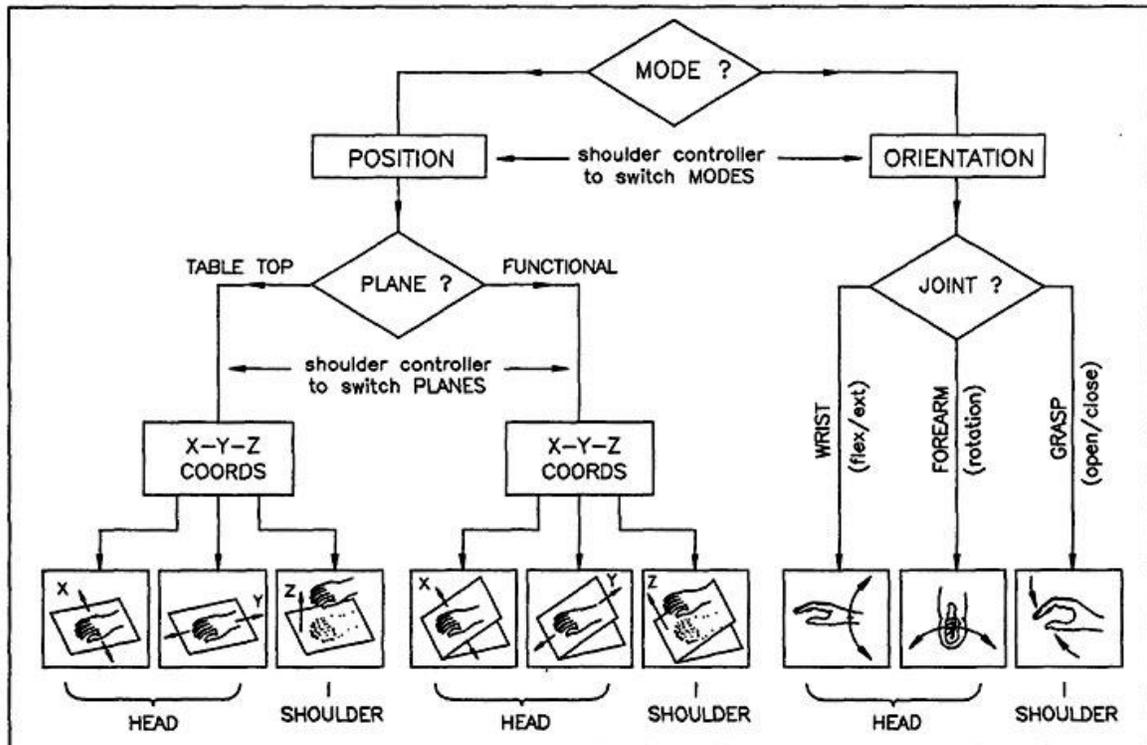


Figure 2.11: Control Strategy of the UBC PULO [2,36]

Based on the results of the previous motion analysis study [1,37], it was found that implementing a strategy of using both horizontal and inclined planes was of significant benefit in performing the identified priority tasks. Therefore, two operational planes: the horizontal (or table-top) plane and the inclined (or functional) plane were created within the control strategy for the UBC PULO design, both of which have advantages during the performance of particular tasks. The tasks requiring the use of horizontal plane are the ones that involve moving objects around on the table surface and lifting an object up vertically from the table surface. The inclined plane is the plane defined between the table surface and the user's head. The inclined angle from the table surface is different

from user to user and is programmable, but is generally orientated at between 30 and 45 degrees. Using an inclined plane is more efficient than using a horizontal plane to perform tasks such as drinking and eating where the repetitive motion of bringing items from the table to the mouth occurs.

The planar movements in position mode defined as X-Y motion are controlled by the head control interface while the perpendicular movement defined as Z motion is controlled by the shoulder control interface. The two user-selectable operating planes are implemented to simplify the user control by reducing the user's effort in switching to Z movement to perform the daily living tasks. Once the desired position is achieved, a quick shoulder shrug motion triggers the shoulder control interface to switch the orthosis from position mode to orientation mode. Under the orientation mode, instead of controlling the end-point in X-Y-Z motion, the head and shoulder control interfaces provide direct control of the rotation in the individual joints.

Although each end-point motion is clearly assigned in the control strategy, the signals generated from the user control interfaces need to be further processed to be utilized as control inputs for the drive system, which is done in the microcontroller.

2.4.4 MICROCONTROLLER

Generally, the microcontroller in the UBC PULO interprets the user's intention by converting the signals from the two user control interfaces into the desired end-point motion which is then translated into the required joint motion through inverse kinematics computation and feedback control.

Two micro-processors with custom-developed software were previously designed for the UBC PULO to link the input signals from user control interfaces and to subsequently carry out the process of device end-point control utilizing the concept of inverse kinematics [2,35]. The first micro-processor, termed “System Input Processor”, contains the two main control algorithms to perform: 1) head/shoulder control signals processing and, 2) the required inverse kinematics. The second micro-processor, termed the “Position Processor”, contains the position control algorithm and utilize a PID (proportional-integrated-derivative) controller to minimize the error between the desired input (in this case, the calculated joint angles) and observed output (in this case, the actual produced joint angles).

As noted above, the System Input Processor contains head and shoulder controller processing algorithms and the differential kinematics algorithm. The head and shoulder algorithm interprets the user-input signals and performs any/all of the following tasks [2]:

- Calculates the desired changing position of the end-point: dx , dy , dz (when in position mode)
- Provides bidirectional control of an individual joint (when in orientation mode)
- Toggles between control modes (Position or Orientation)
- Toggles between operational planes (Table top or functional)
- Deactivates the orthosis (when intentional/unintentional rapid head motion is detected)

- Reactivates the orthosis
- Disables the input signals(when the head movement is within the neutral zone)

When in position mode, the changing position of the end-point is used as the input to the second algorithm.

The differential kinematics algorithm calculates the required angle of each joint to move the end-point to the desired position based on the user's command. This algorithm includes the formulation of the homogeneous transformation matrix and the inverse Jacobian matrix. The homogeneous transformation matrix is computed by using the Denavit-Hartenberg (D-H) method, a commonly used method in the robotics field for forward kinematics computation [1,2]. Generally, the Denavit-Hartenberg method computes the sine and cosine of current joint angles to produce a 4x4 matrix which contains the position and orientation of the end-point in terms of X, Y and Z axis with respect to the base coordinate system. Thus, the current spatial information of the hand is known at all times during the operation. In addition, the position vectors from the homogeneous transformation matrix are used for the computation of the inverse Jacobian matrix. The inverse Jacobian matrix is the mapping between the rate of change in each joint and the rate of change of the end-point. Based on the nature of the inverse Jacobian matrix, the desired changing position of the end-point calculated from the head and shoulder algorithm is sent to the computation as the input, and the algorithm outputs the corresponding joint angles required to move the end-point to the

target position. The calculated joint angles as the reference signals are then sent to the second processor for the position control.

The Position Processor is implemented with the angle control algorithm using PID error control to regulate the voltage sent into the motors for producing the desired orthosis motion. In order to feedback the angular position information from the device, potentiometers are mounted at each joint. By tuning the control gains from the PID controller, the error between the reference joint angle calculated from the differential kinematics algorithm and the actual joint angle measured from the potentiometers are minimized.

2.4.5 SUMMARY

In the UBC PULO control system (see Figure 2.12), the method of end-point control is utilized to allow the user to control the hand to the desired destination intuitively and directly through simple input commands. The user input commands are generated through use of the user's head and shoulder motion, and acquired by head and shoulder control interfaces which translate the user's head inclination and vertical shoulder motion into the corresponding electronic signals to control the device. To enhance the functionality and efficiency in controlling the device, a unique control strategy was developed. First, a microcontroller interprets the signals from user control interfaces to realize the user's intention. Then, the inverse kinematics algorithm computes the angular position of the joints required to move as the reference inputs to the drive system. The potentiometers feedback the angular position actually

achieved at each joint to compare with the reference, and the error is minimized via PID control.

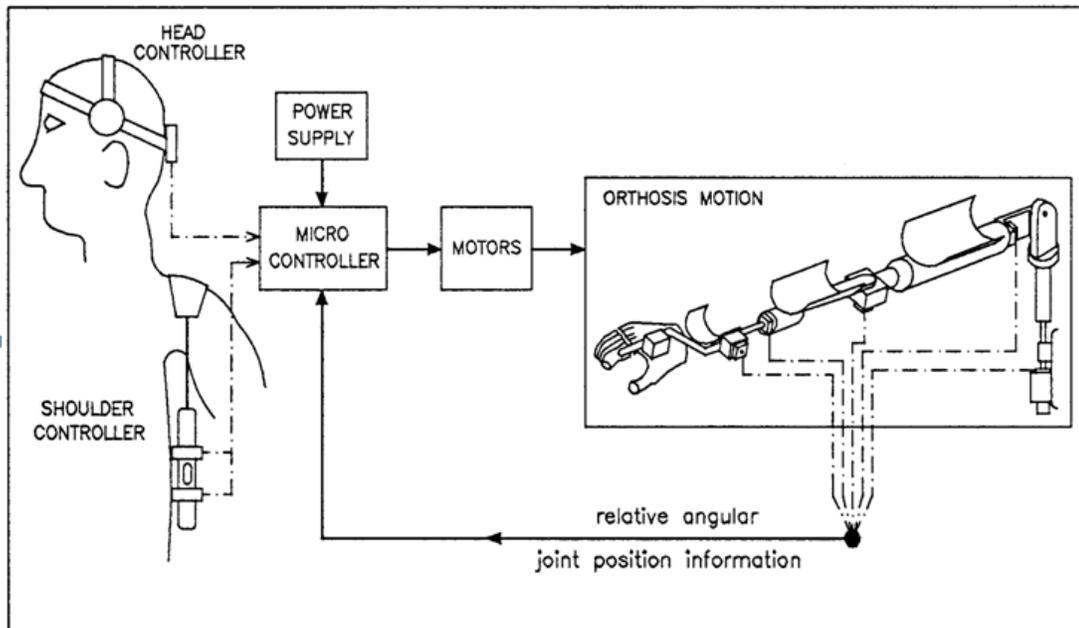


Figure 2.12: Schematic of UBC PULO Control Components [2]

2.5 SUMMARY

This literature review covered previously developed powered upper limb orthoses and those currently under development. While significant effort has been made in the development of a powered upper limb orthosis by researchers in the field, there is still a clear need to design a powered upper limb orthosis with high functionality, portability and ease of control to assist people having both flail arms to do daily living activities. Because of that need, the UBC Powered Upper Limb Orthosis program was initiated. The goal of the PULO program and the development of the PULO prototype have been outlined and discussed. Currently, the PULO prototype requires optimization in various aspects of both the mechanical system design and control system areas. In order to carry on the

current research work to optimize the current control system, the previously designed control concept, strategy and developed user control interfaces were reviewed.

In the following chapter, the detail of the required tasks for the development of a virtual simulator tool to assist in this optimization process is reported.

CHAPTER 3:
DEVELOPMENT OF A VIRTUAL SIMULATOR FOR UBC PULO

3.1 INTRODUCTION

The physical UBC PULO prototype was built based on the design developed by earlier researchers under the supervision of Dr. D.P. Romilly [1,2,10,35,36,37,41]. To evaluate the performance of the prototype, the device was previously tested on the laboratory bench. According to the results, hardware problems associated with both the mechanical and electrical components were identified, which made the testing of the PULO control system extremely difficult to perform due to these complicating issues. Thus, a new testing method to evaluate the current PULO control system without the interference of these hardware problems needed to be developed. The current research work focuses on the development of a virtual simulator to test and optimize the control system, as well as to provide a tool for candidate screening and user training purposes.

This chapter documents the development process of the virtual simulator for the UBC PULO, which included the following tasks:

1. Assessment of available software tools
2. Realization of the user input signals
3. Modeling of the microcontroller
4. Modeling of the drive systems and PID controllers
5. Physical modeling of the orthosis arm

6. Development of the virtual scene

The overall schematic of the UBC PULO virtual simulator shown in Figure 3.1 identifies the required tasks and indicates the role of each task in the overall development of the simulator. The defined inputs to the simulator are the signals of the user control interfaces which are triggered by the motions of the user's neck tilting and the shoulder shrug. The virtual simulator is expected to simulate the virtual arm motion controlled by the user in real-time.

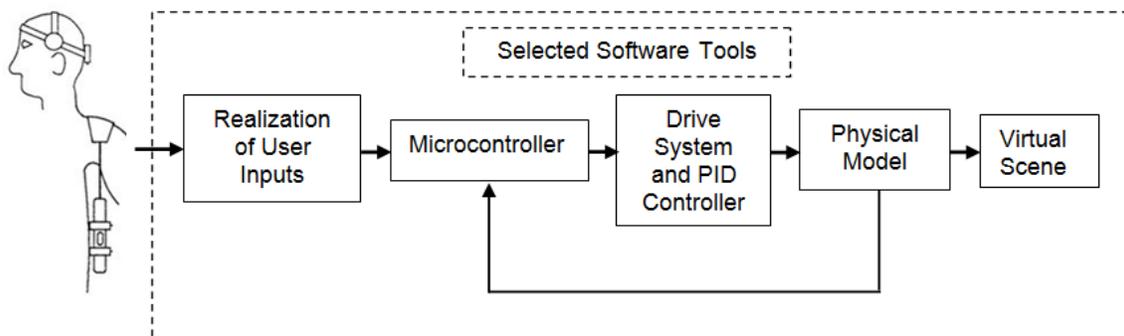


Figure 3.1: Overall Schematics of the Virtual Simulator

Using the selected software tools, a method was developed to obtain the real-time input signals from the user control interfaces for the computer to process. The model of the microcontroller contains the algorithms needed to quantify and interpret the user's input signals and to perform the necessary inverse kinematics to compute the desired joint angles. The desired joint angles (used as the reference input signals) are sent to both the PID controller and the model of the DC motors to minimize the error between the desired reference inputs and the actual outputs. Then, the physical model of the PULO is driven by the outputs of the drive system. The physical model of the orthosis, a graphical representation of the system containing the relationship of the mechanical

components, produces the virtual motion based on the motion of the DC motors. The virtual motion is then displayed in a virtual scene. In order to perform these tasks, the available software tools first needed to be investigated and selected for this work.

3.2 SOFTWARE TOOL ASSESSMENT

There are a variety of software tools that are capable of developing a Virtual Reality Environment (VRE); however, a specific software tool was needed to meet the design requirements for this project. Based on the description of the required tasks, this software tool required the following essential features:

1. A modeling feature that allows for modeling control elements such as a PID controller, and mechanical components such as DC motors, links and joints.
2. A capability to design a virtual reality environment that permits the virtual display of the orthosis motion in an appropriate setting.
3. Compatibility with receiving real-time user's input signals.
4. Ease of use, especially due to the author's limited programming background.

Considering the desired software features, software tools were identified, with a few potential software tools short-listed, and investigated in more detail. The evaluation results for the short-listed set are shown in Table 3.1.

Software Tools	Modeling & Control Design	Virtual Reality Design	Real-Time Simulation	Ease of Use
MapleSim	★★★	☆☆☆	★★☆	★★★
Vizard	★★☆	★★★	★★☆	★★☆
Robotic Studio	★★☆	★★☆	★★☆	★★☆
Matlab & Simulink	★★★	★★☆	★★★	★★★

Table 3.1: Potential Software Tools

(★: with desired features; ☆: without desired features)

MapleSim, developed by Maplesoft, provides the best features in terms of modeling the physical system and control system design when compared with the other reviewed software tools [16]. MapleSim's modeling process is achieved by dragging the desired components from the library browser to the model window, thus it requires no programming background. However, the drawback is that it does not support real-time simulation within the Virtual Reality Environment [16]. On the other hand, Vizard, invented by WorldViz, provides the best software tool for developing the Virtual Reality Environment. However, Vizard does not have the features needed for developing the required mechanical and control system models for this project. Additionally, it requires the developer to have significant programming skills using Python to write scripts [43]. Another reviewed software tool is the Microsoft Robotics Developer Studio (MSRDS) software designed to build robot applications using Visual Programming Language. MSRDS includes programming models to support a wide range of commercial robots. It also simulates and tests robots using a 3D physics-based simulation tool. The supported hardware interfaces for real-time simulation include a desktop joystick, the Xbox input controller, and various

speech based interfaces, but it does not have any built-in support needed for the user control interfaces of this project [23]. Finally, the combination of Matlab & Simulink from MathWorks is another great software tool which provides multi-domain simulation and model-based design for a dynamic system [21]. Similarly to MapleSim, the system model can be built based on dragging and placing the blocks from the library browser into Matlab & Simulink. The advantage of these software tools is that they contain different toolboxes to support a variety of design requirements. For instance, the Real Time Windows Target toolbox is compatible with data acquisition (DAQ) devices which can obtain real time input signals from the user control interfaces [18]. Additionally, another toolbox, SimMechanics, has the feature useful for modeling the mechanical system for this project and simulating its dynamic behaviour [19]. The simulated motion can then be visualized in the Virtual Reality Environment developed using the Simulink 3D Animation toolbox [15,20]. This toolbox only supports computer monitors as the display interface, but it is more suitable for use in this project because it greatly reduces the chance of cyber sickness for the user, as well as being more cost effective [5,42].

As a result of this investigation, Matlab & Simulink, as well as their associated toolboxes, were ultimately selected and purchased as the software tools to develop the virtual simulator for the UBC PULO.

3.3 REALIZATION OF THE USER INTERFACE DEVICE INPUT SIGNALS

Once the software tool was selected, the first task of the simulator development process was to establish the means of acquiring input data from

the user interface devices. Data acquisition (DAQ) is the process of measuring an electrical or physical phenomenon such as voltage, current, temperature, pressure, or sound [27]. In this case, the objective of the data acquisition process was to measure the user's head and shoulder position displacement through the monitoring of sensors from the head and shoulder controllers. In order to transmit the user's input command into the Matlab & Simulink environment, the National Instrument PCI-6221 (68-Pin) was used as the DAQ hardware interface between the computer and the signals obtained from the user control interfaces. The physical DAQ setup is shown in Figure 3.2

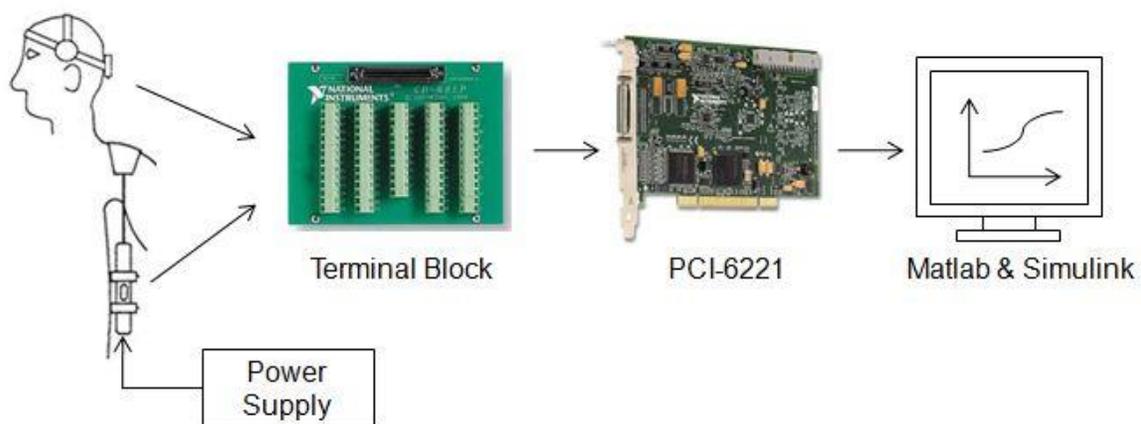


Figure 3.2: Schematic of the DAQ set-up

In order to utilize the signals produced from the head and shoulder controllers as the user control inputs, a microcontroller containing the necessary control algorithms was implemented on the UBC PULO prototype as defined in the following section.

3.4 MICROCONTROLLER

A microcontroller was designed and mounted to the UBC PULO prototype as the major control unit. The goal of the current work in this particular task is to

model the functional features of the actual microcontroller for use in the virtual simulator in order to mimic its performance in the virtual environment. The PULO control functions of the microcontroller include running the user control interfaces algorithm, the selection the user control mode and the performing the inverse kinematics computations.

3.4.1 USER CONTROL INTERFACES ALGORITHM

The user control interfaces algorithm in the simulator is a model of the control features which were previously designed for the head and the shoulder controllers. The control features for the user control interfaces are listed in Table 3.2.

Head Controller	Shoulder Controller
<ul style="list-style-type: none"> • User defined neutral zone & operation zone • Speed monitor • End-point speed • End-point changing position in X-Y plane under position mode • End-point rotation: direct control to the specific joints under orientation mode 	<ul style="list-style-type: none"> • User defined short pulse • Operation mode switching • End-point changing position in Z direction • End-point rotation: direct control to the specific joints under orientation mode • Emergency shutdown (Not included in the simulator)

Table 3.2: Control Features of the Head and Shoulder Controllers

The neutral zone and speed monitor are both safety features designed for the head controller to prevent accidental end-point movement from undesired neck motions [2,10,36]. The neutral zone (see Figure 3.3 below) is a user defined range of neck motion, which allows the user to move the head freely without producing any end-point movement. The speed monitor constantly measures the speed of the user's neck motion during the operation and disables the end-point motion if the speed of the neck motion exceeds the predetermined threshold.

These two safety features in the head controller were implemented in the simulator. In order to define the neutral zone, the subject moved the head to simulate the undesired neck motion for end-point movement, by turning the head to left or right, as if for a conversation or for nodding. Meanwhile, the algorithm recorded the readings from the inclinometers and approximated the range of the neutral zone. The same method was used to define the boundary of the operation zone where it contains a sufficient range of neck motion for the control of the end-point and maintains the user's eye contact to the hand at all times during the operation. To model the speed monitor, a subject wore the head controller and simulated the action of sneezing, and the speed of the signal generation from the inclinometers was recorded and set as the speed threshold.

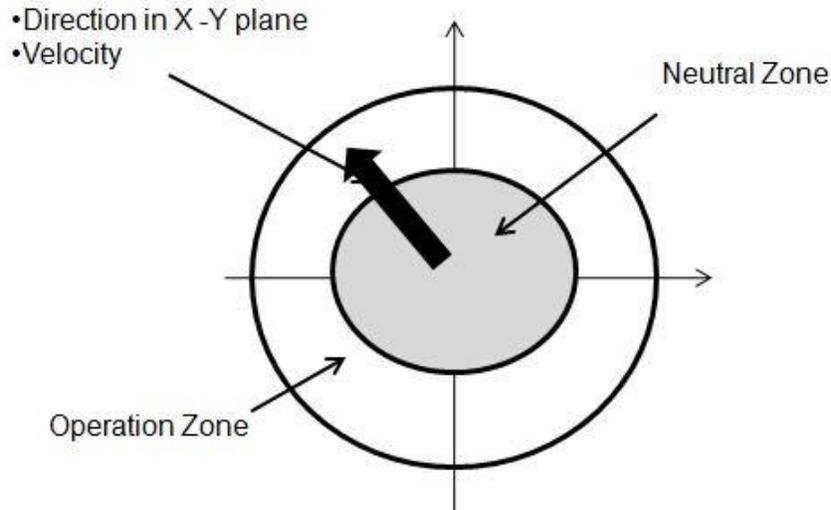


Figure 3.3: Original Design of the Neural and Operation Zones

As illustrated, the active command zone between the neutral zone and the boundary of the operation zone is where the active user commands are generated from the head controller. In position mode, the algorithm was coded as Matlab Embedded Functions in Simulink to convert the readings of each inclinometer into the desired changing position of the end-point in X-Y direction. In the orientation mode, the left, right, forward and backward tilting motion of the neck were utilized to produce the input signals for direct control of the specific joints in both directions. The magnitude of the readings from both inclinometers was computed and converted to the desired end-point speed accordingly. The issue of cross-sensing from the inclinometers was also considered when the algorithm was coded, so that the algorithm could accurately interpret the user's neck motion and generate the corresponding input signals.

As reviewed in Chapter 2, the two optical switches designed into the shoulder controller produce digital signals in the form of either "ON" or "OFF". By inducing a time variable that describes the duration of the signal when it is on,

more control signals can be generated as a function of time. A user defined time variable defines the signal as a short pulse or a long pulse as shown in Figure 3.4 [2,36]. A short pulse generated from a quick shoulder shrug motion switches the operation mode; a long pulse generated from a steady shoulder raise motion controls the end-point motion.

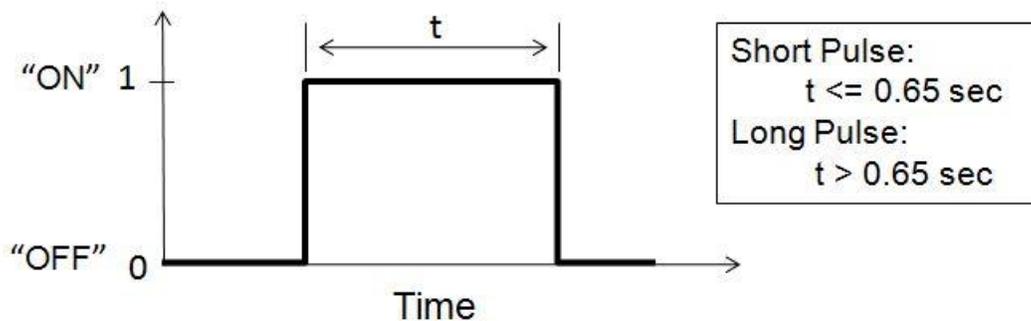


Figure 3.4: Defined Shoulder Switch Signal

3.4.2 CONTROL STRATEGY

The design of the control strategy for the UBC PULO was implemented in the simulator by creating a finite state machine. The finite state machine is a representation of an event-driven system (in this case, the PULO control system) that makes the transition from one state (in this case, the operation mode) to another if the defined condition is true [21]. The user defined short pulse signal was used for switching operation modes. A counter was designed in Simulink to count the falling edge of the short pulse signal and to reset the counting once the threshold was reached. In this case, the threshold was set to be 3. The flow diagram of the state machine is shown in Figure 3.5, with each number representing an operation mode.

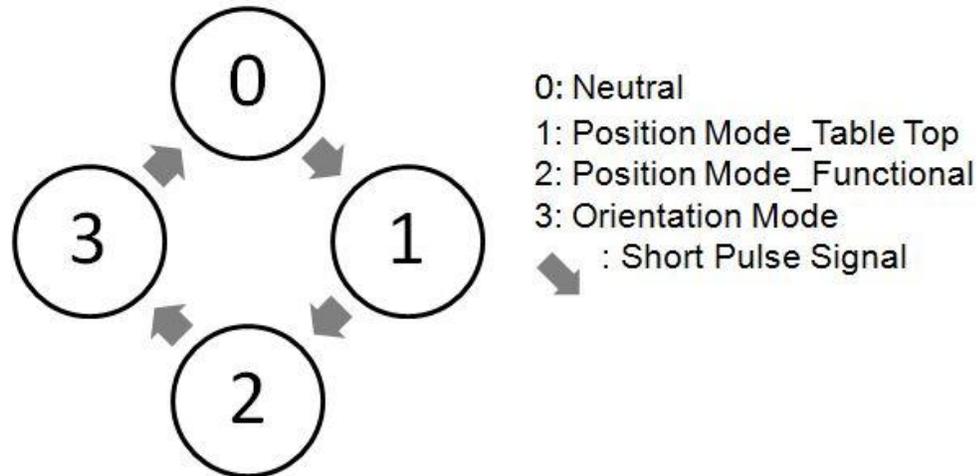


Figure 3.5: State Machine Diagram of the Simulator

As demonstrated, each additional short pulse signal generated from the quick shoulder shrug motion allows the user to switch to the next operation mode in a sequence of neutral, position mode (table top plane), position mode (functional plane), and orientation mode. The sequential loop design for selecting the mode is preferred rather than the direct mode switching in this project because additional body motion from the user needs to be introduced and characterized in order to define more control signals, which makes the control strategy less intuitive and requires more effort from the user.

The different operation modes were structured as subsystems of the control strategy model. In each subsystem, the necessary algorithm was coded to perform what each mode was designed to do. For instance, the orientation mode allows the user to have direct control for the motion of forearm roll, wrist flexion/extension and grasp. The algorithm for the orientation mode sends the signals from the head and the shoulder controllers to the corresponding motors. The predetermined joint limits are set as the positional reference for the motors

to move, and the speed of the motion is controlled by the first order low pass filter which is described in Laplace transform notation. In the position mode, the inverse kinematics algorithm is used to move the joints concurrently to achieve the desired end-point position.

3.4.3 INVERSE KINEMATICS

The inverse kinematics developed for the UBC PULO converts the user's input commands (in terms of a desired end-point position) to combined joint motion (in terms of the joint angles), and includes the algorithms of forward kinematics and the inverse Jacobian. The forward kinematics computes the position and orientation of the end-point relative to the reference frame based on the configuration of the orthosis arm [7]. The forward kinematics computation outputs a 4X4 transformation matrix used for the formulation of the inverse Jacobian. The inverse Jacobian in this project is a 3X3 matrix which specifies a mapping between the rate of change of the end-point position and the angular velocity of the joints [7,8,39]. Previously, the inverse kinematics algorithm was developed based on the current prototype. Throughout the current work, some configuration problem issues in the prototype were observed, thus to avoid these problems, and some modifications were made to the inverse kinematics algorithm.

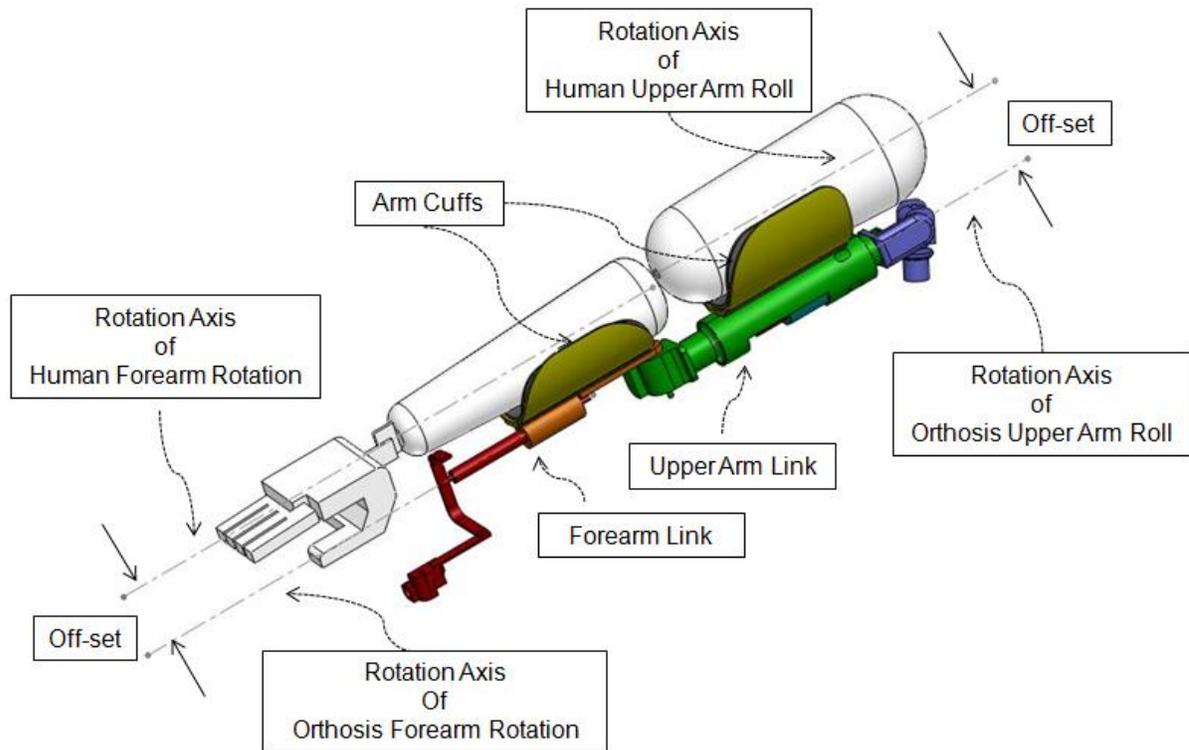


Figure 3.6: Model of the PULO Arm Prototype

The configuration of the UBC PULO prototype is shown in Figure 3.6. The arm cuffs are rigidly fixed to the upper arm and forearm links of the orthosis and provide both static and dynamic support to the user's arm. One issue of the current structural design is that an off-set distance exists between the center of the rotation of the robotic arm and the human arm. Another issue observed is that the small off-set between the center of the rotation of the orthosis upper arm and forearm at the elbow joint makes the orthosis arm a nearly singular configuration [8,22]. As a result of the inverse Jacobian matrix formulated based on this configuration being ill-conditioned, the end-point is uncontrollable at the nearly singular location.

In order to simulate how the UBC PULO was supposed to be designed to perform, a modification was made to the orthosis shoulder joint. Previously, two

degrees of freedom were selected for the shoulder joint: shoulder azimuth and upper arm roll. In the current work, the shoulder elevation was introduced in the simulator to replace the upper arm roll. As a result, the issue of the miss alignment between the centers of rotation of the human upper arm and orthosis upper arm was eliminated, and Jacobian matrix for the prototype became well-conditioned. Based on this new orthosis configuration, the forward kinematics and inverse Jacobian were then formulated.

The UBC PULO consists of several rigid links which are connected by revolute joints. One end of the links is fixed to the plastic body fixture which is considered as the base; while the other end, the hand, is free to move and is referred as the end-point. To define the orientation and the position of the end-point with respect to the base during the operation, the Denavit-Hartenberg (D-H) method is used to compute the necessary forward kinematics equations. The D-H method is commonly used in the robotics field to systematically locate the coordinate frame at each link and to compute the coordinate transformations between two consecutive links [1,8,39]. The coordinate frame at each joint is established based on the following rules [8]:

- 1: The z_{i-1} axis lies along the axis of rotation of the i^{th} joint
- 2: the x_i axis is assigned to be perpendicular and to intersect the z_{i-1} axis
- 3: the y_i axis completes the right-handed coordinate system as required.

The above “rules of thumb” for assigning the frames to the links results in a simplified method for computing the forward kinematics transformation T_{o-n} of a

robotic system. The D-H coordinate system layout of the UBC PULO simulator is shown in Figure 3.7.

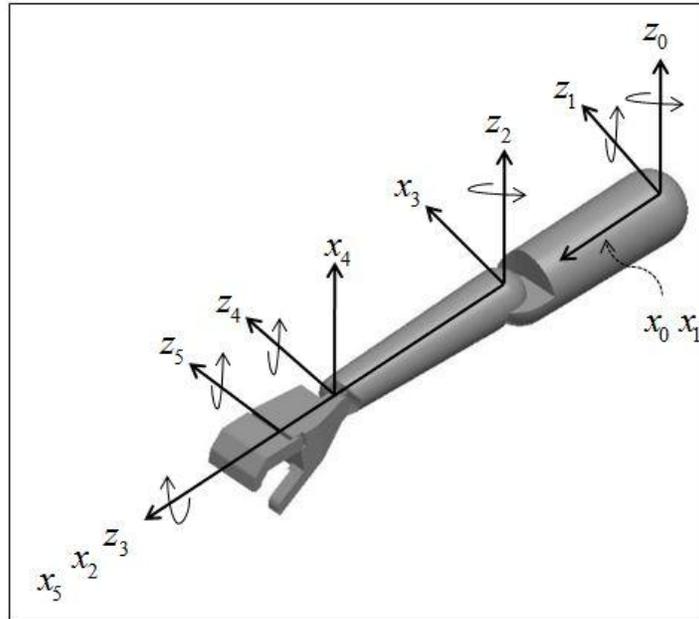


Figure 3.7: D-H Coordinate System Layout of the Simulator

Based on the established frames, the necessary parameters are defined to formulate the transformation matrices. The definition of the D-H parameters are described in Table 3.3

Parameter	Axis	Description
θ_i	z_{i-1}	Joint angle: variable for revolute joint
d_i	z_{i-1}	Link offset: variable for prismatic joint
a_i	x_i	Link length: constant perpendicular distance between z_{i-1} and z_i
α_i	x_i	Link twist: constant angles between z_{i-1} and z_i

Table 3.3: Definition of D-H Parameters

Any displacement between two coordinate frames attached to two links connected by a revolute joint can be represented by the following transformation matrix:

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

The overall transformation from the end-effector frame back to the base frame is:

$$T_{05} = A_{01} \cdot A_{12} \cdot A_{23} \cdot A_{34} \cdot A_{45}$$

This overall transformation matrix provides the information for the end-point position and orientation in the work space at all times. The detailed computation of the matrix, T_{05} , is included in Appendix. The positional information of the end-point is further utilized to generate the relationship between the joint angles and the end-point changing position by the method of differential kinematics.

The goal of the differential kinematics calculations is to find the functional relationship between the joint velocities and the end-effector linear and angular velocities [38]. In this project, the target position of the hand is calculated by the combination of realizing the user's intention and computing the current hand position. To ultimately move the hand, it is desired to express the joint velocities as a function of the end-point linear velocity by inverting the Jacobian matrix, which can be done by using the following relation:

$$q = J^{-1} \cdot p$$

Where:

$p = (p_x, p_y, p_z)^T$, describes the final position of the hand with respect to the base frame; J is a 3x3 matrix derived from the position vector of the homogenous transformation matrix; $q = (\theta_1, \theta_2, \theta_3)^T$ where $\theta_1, \theta_2, \theta_3$ are the desired joint angles.

In order to derive the inverse Jacobian, the differentiation method was used in the current work and the following computation were performed:

- a. The position vector was obtained from the D-H matrix where the last column of the final matrix, T_{05} , represents the spatial location of the end-point with respect to the base coordinate system, located at the shoulder. Specifically:

$$T_{05} = \begin{bmatrix} R^{3 \times 3} & P_x \\ & P_y \\ & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{4 \times 4}$$

where the position vector is a function of the joint angles:

$$P^{3 \times 1} = \begin{bmatrix} P_x(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \\ P_y(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \\ P_z(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \end{bmatrix}$$

- b. P_x, P_y, P_z were differentiated symbolically with respect to $\theta_1, \theta_2, \theta_3$, and the results of the partial derivative were assigned as letters from a to i .

$$a(\theta_1, \theta_2, \theta_3) = \frac{\partial\{P_x(\theta_1, \theta_2, \theta_3)\}}{\partial\theta_1}$$

$$b(\theta_1, \theta_2, \theta_3) = \frac{\partial\{P_x(\theta_1, \theta_2, \theta_3)\}}{\partial\theta_2}$$

$$c(\theta_1, \theta_2, \theta_3) = \frac{\partial\{P_x(\theta_1, \theta_2, \theta_3)\}}{\partial\theta_3}$$

$$d(\theta_1, \theta_2, \theta_3) = \frac{\partial\{P_y(\theta_1, \theta_2, \theta_3)\}}{\partial\theta_1}$$

$$e(\theta_1, \theta_2, \theta_3) = \frac{\partial\{P_y(\theta_1, \theta_2, \theta_3)\}}{\partial\theta_2}$$

$$f(\theta_1, \theta_2, \theta_3) = \frac{\partial\{P_y(\theta_1, \theta_2, \theta_3)\}}{\partial\theta_3}$$

$$g(\theta_1, \theta_2, \theta_3) = \frac{\partial\{P_z(\theta_1, \theta_2, \theta_3)\}}{\partial\theta_1}$$

$$h(\theta_1, \theta_2, \theta_3) = \frac{\partial\{P_z(\theta_1, \theta_2, \theta_3)\}}{\partial\theta_2}$$

$$i(\theta_1, \theta_2, \theta_3) = \frac{\partial\{P_z(\theta_1, \theta_2, \theta_3)\}}{\partial\theta_3}$$

c. In this project, the Jacobian is calculated as a 3X3 matrix, where each row represents the X, Y, Z coordinate; and the number of columns represents the number of joints. The results of the partial derivative were mapped to formulate the Jacobian matrix:

$$J = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

- d. The numerical determinant of the Jacobian, $\det[J]$, was computed. It is very important that $\det[J] \neq 0$; otherwise, the Jacobian inversion is undefined. Additionally, if $\det[J] = 0$, it will result in a singular configuration, and there will be no unique solution for the joint velocity to achieve the target end-point position.
- e. The inverse Jacobian was calculated symbolically as follows:

$$[J]^{-1} = \begin{bmatrix} \frac{(e \cdot i) - (f \cdot h)}{\det[J]} & \frac{(c \cdot h) - (b \cdot i)}{\det[J]} & \frac{(b \cdot f) - (c \cdot e)}{\det[J]} \\ \frac{(f \cdot g) - (d \cdot i)}{\det[J]} & \frac{(a \cdot i) - (c \cdot g)}{\det[J]} & \frac{(c \cdot d) - (a \cdot f)}{\det[J]} \\ \frac{(d \cdot h) - (e \cdot g)}{\det[J]} & \frac{(b \cdot g) - (a \cdot h)}{\det[J]} & \frac{(a \cdot e) - (b \cdot d)}{\det[J]} \end{bmatrix}$$

Because the Jacobian is a square matrix and $\det[J] \neq 0$, the relationship between the end-point linear velocity and the joint velocity can be exactly determined. In another words, a unique solution of the joint velocity to achieve the end-point linear velocity exists. The incremental change of the joint angles as the reference input signals were then sent to the model of the drive system for use in position control.

3.5 DRIVE SYSTEM

The drive system of the UBC PULO simulator consists of the Simulink models of the DC motors and PID controller. The DC motor, being an electromechanical component, has both electrical and mechanical behaviours which must be characterized [9,31]. In this work, the behaviour of the DC motors was mathematically modeled by transfer functions used to describe the

relationship between the electrical input, i.e. voltage, and mechanical output, i.e. angular displacement. The Simulink model of the DC motors is shown in Figure 3.8, and the parameters involved in the DC motor model are listed in Table 3.4

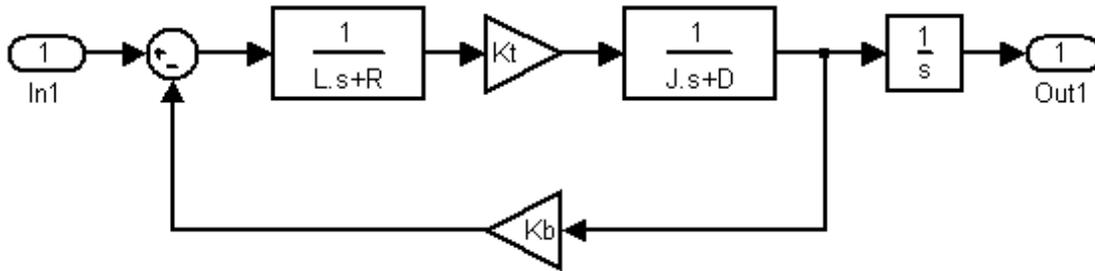


Figure 3.8: Simulink Model of the DC Motor

DC Motor Parameters	Description
R	Resistance
L	Inductance
Kt	Torque constant
Kb	Back EMF constant
J	Inertia
D	Damping constant

Table 3.4: DC Motor Parameters

In order to minimize the error between the produced joint angle and desired joint angle, a PID controller is implemented for each joint within the UBC PULO. The PID control algorithm was modeled in Simulink as shown in Figure 3.9.

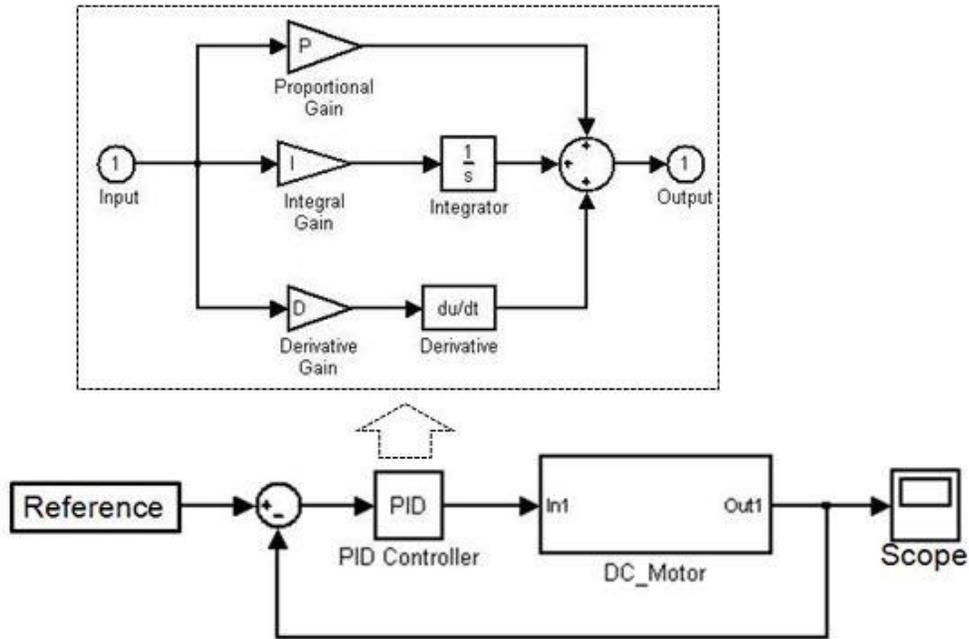


Figure 3.9: Simulink Model of PID Controller

The detailed tuning process of the P, I, D gains is discussed in Chapter 5. After modeling the drive system, the next step is to utilize the signals generated from the DC motors as the input signals to drive the physical model of the UBC PULO.

3.6 PHYSICAL MODEL

Generally, a physical model represents the physical structure of a machine, the mass properties, and geometric and kinematics relationships of its component bodies [19]. In this project, a physical model of the UBC PULO was created by using SimMechanics, which models the orthosis arm segments connected by revolute joints and the potentiometers implemented on the UBC PULO prototype at these joints. The process of creating the physical model is categorized in the following major steps:

a. Each orthosis arm segment was modeled in SolidWorks as individual parts based on their geometric and mechanical properties. Then, the parts were connected by revolute joints as an assembly model to represent the kinematic relationships for the components based on the prototype assembly. The SolidWorks model of the UBC PULO simulator is shown in Figure 3.10

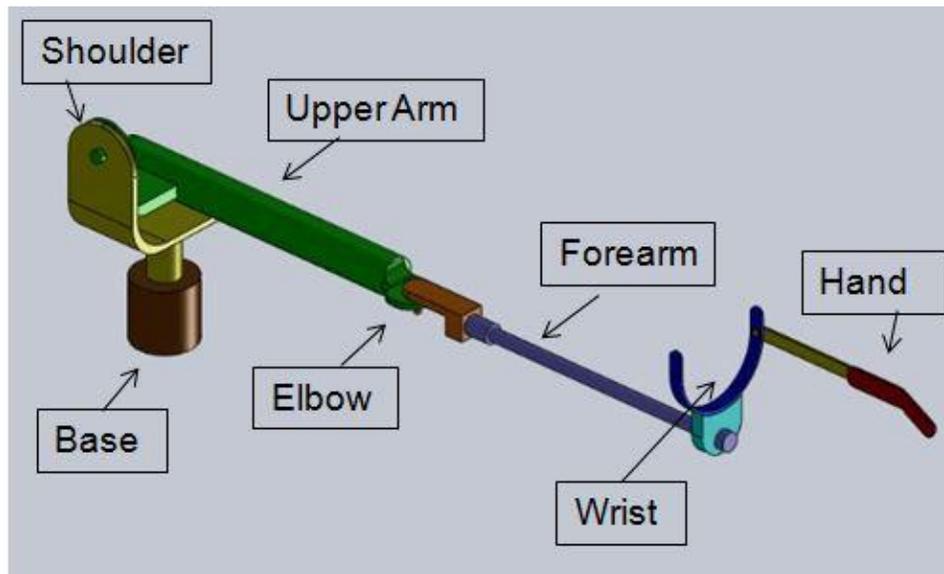


Figure 3.10: SolidWorks Model of the Orthosis Arm

b. The SolidWorks assembly was converted into the physical model as SimMechanics model of the UBC PULO shown in Figure 3.11 .

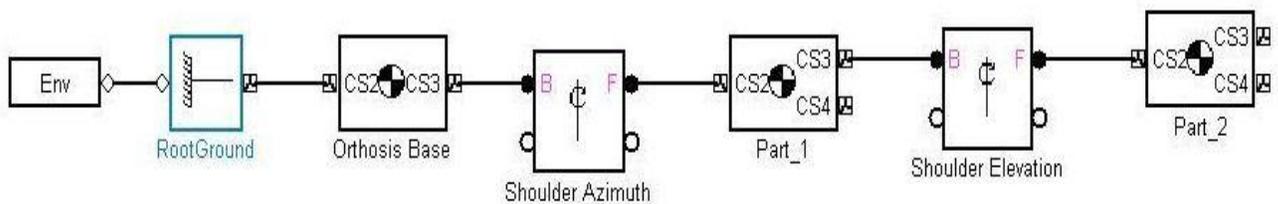


Figure 3.11: SimMechanics Model of the Orthosis Arm

The physical model of the UBC PULO is represented by a series of SimMechanics blocks. The role of the most important blocks are listed in the following table.

Block Name	Description
Environment	It provides the mechanical environment for the connected machine, including simulation dynamics, gravity setting, motion analysis mode and visualization.
Root Ground	It provides the base or ground for the machine, which enables the connected bodies to define the relative coordinate system.
Body	It models the rigid body of the machine components and specifies the mass property, the coordinate systems and the connector ports of the rigid bodies.
Joint	It connects the bodies and creates degree of freedom.

Table 3.5: Definition of the Important SimMechanics Blocks [19]

c. The control signals from the model of the drive system were sent to the SimMechanics model by adding the actuator blocks; the current joint angles were measured and sent back to the PID controller by adding the sensor blocks.

The physical model of the UBC PULO was created as a SimMechanics model based on the procedures described above. Since the creation of the physical model of the UBC PULO starts from the CAD model, it has the option of modeling the orthosis components graphically as well as configuring the material

properties. As the PULO prototype is further optimized (in terms of material selection for weight reduction), the SimMechanics model can be automatically updated from the CAD model to integrate the mechanical hardware modification. Additionally, the feature that allows the physical model to be controlled by the Simulink model for simulation provides a great tool for observing the dynamic behaviour of the system, as well as acquiring and analyzing the necessary output data.

However, in order to simulate the whole operational process of using the UBC PULO to complete a desired task in a real-time manner, a virtual reality environment is necessary as the interface to be provided for the user to perform any pre-defined tasks virtually. Thus, a virtual environment was developed by using Simulink 3D Animation toolbox as described in the next section.

3.7 VIRTUAL ENVIRONMENT

By definition, virtual reality is a user-computer interface that involves real-time simulation and interactions through multiple sensorial channels [5,40,42]. The history of virtual reality technology can be traced back more than 50 years, and with rapid development of the computer technology, virtual reality development has been prominent in a wide variety of applications including military and medical training, rehabilitation, and entertainment [5,40]. The implementation of a virtual reality environment in this project is motivated by its key feature of interactivity. The real time interactivity enables the instantaneous response by the virtual objects resulting from the user's inputs, which is a required function of the UBC PULO simulator for potential user training and

screening purposes. Therefore, a virtual environment was created for the UBC PULO simulator to visualize the output motion of the orthosis arm.

In this project, the virtual environment provides an interactive environment on the computer screen for the user to perform functional tasks. Because the UBC PULO is designed for daily living tasks, a home-like environment was chosen for the design. The process of creating the virtual environment is discussed below using a series of defined steps:

- a. The virtual objects were created by using the SolidWorks. The most important virtual object is the arm that represents the user's arm supported by the UBC PULO in the real world. The objects, such as the walls, the windows and the floor, were modeled to define the room environment. Additional objects such as the TV, the lamp and the table were modeled to help the user to relate with the environment.
- b. The virtual scene was created by exporting the virtual objects from the CAD program to the Simulink 3D Animation Toolbox in the Virtual Reality Modeling Language (VRML) format. The built-in VRML editor from the toolbox was used to edit and group the virtual objects in a hierarchical tree structure to represent its functionality in the scene. Additional information such as background, light condition and view points were also added into the scene to enhance the visualization. A screen shot of the virtual scene is shown in Figure 3.10

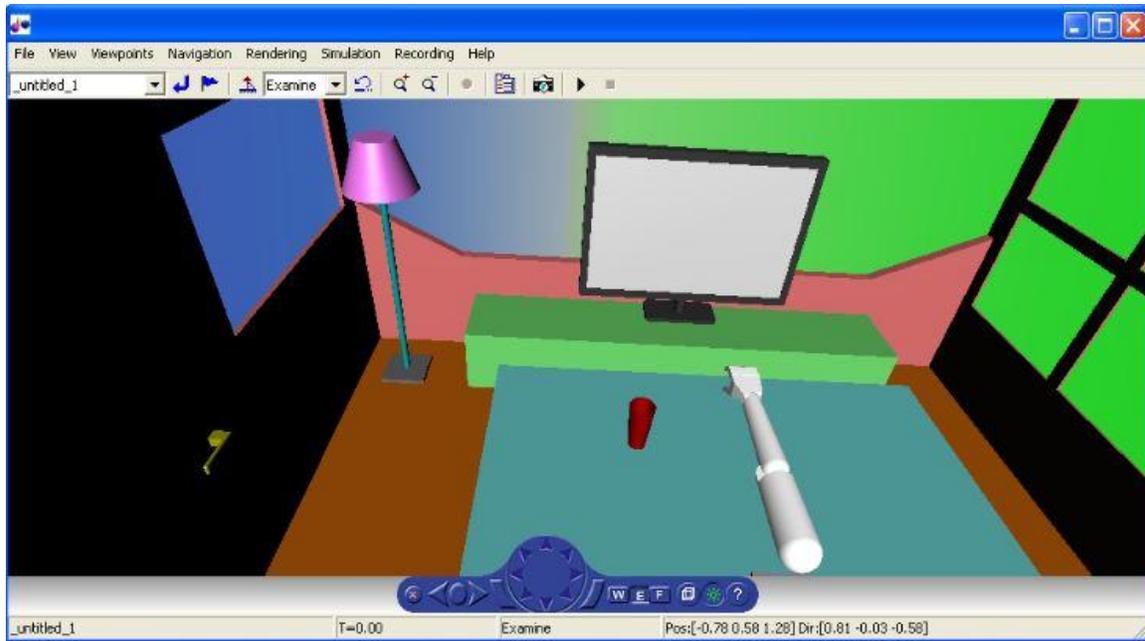


Figure 3.10: Virtual Scene of the Simulator

- c. The virtual arm motion was created by importing the necessary signals from the physical model. The motion of the virtual arm in the scene is driven by the displacement of each arm segment. The changing location of each arm segment is described by its incremental change in translation and rotation at the center of gravity. Therefore, in the physical model, the translation and rotation information of each orthosis arm segment was obtained and sent to the corresponding virtual arm segments in the virtual scene.

By completing the above steps, the virtual reality environment was developed to display the virtual arm motion driven by input signals obtained from physical model.

3.8 SUMMARY

The tasks involved in the development of the UBC PULO simulator are summarized below:

1. Starting from the software tool selection, Matlab & Simulink as well as the necessary Simulink Toolboxes were chosen to model the control system and the mechanical system of the UBC PULO.
2. The model of the control system started from the realization of the user's input by utilizing the National Instrument data acquisition hardware to interface between the user and the simulator. Then, the obtained signals from head and shoulder controllers were quantified, analyzed, and utilized to defined all the control features that were previously designed for the UBC PULO prototype.
3. The PULO microcontroller was modeled to interpret the signals from the user control interfaces, which contains the model of the control strategy, the formulation of the forward kinematics and the inverse Jacobian matrix. The model of the microcontroller outputs the desired joint angles.
4. The desired/calculated joint angles were used as the input to the model of the drive system for position control. The PID controller was modeled to minimize the error between the reference input and the actual output.
5. The physical models of the mechanical components of the prototype were created by using SolidWorks and SimMechanics.

6. The virtual environment was designed and developed to visualize the simulated arm motion. The motion of the virtual arm was driven by the signals from the physical model.

The virtual reality simulator for the UBC PULO was developed through the completion of the above tasks. Subsequently, simulation of the orthosis performing the defined tasks was conducted with the results discussed in Chapter 4.

CHAPTER 4: SIMULATION

4.1 INTRODUCTION

One of the research objectives of this work is to optimize the current control system for the UBC PULO. In order to do so, the current control system needs to be evaluated based on the performance of either the prototype or a virtual simulator. Due to the hardware issues that exist in the UBC PULO prototype, the current work approached the process of analyzing, testing and optimizing the control system through the development of a virtual simulator. The development procedure for the UBC PULO simulator was discussed in the previous chapter. Upon the completion of the simulator, a specific virtual task was designed to test the performance of the control system. This chapter documents the results obtained from the developed simulator for the subject performing the designed virtual task in real-time.

4.2 DESIGN OF THE VIRTUAL TASK

To test the simulator's ability, a specific virtual task was designed in this research. One of design requirements for the evaluation is that the virtual task must utilize every control component of the control system, so that each control feature can be simulated, tested and evaluated. Another design requirement of the virtual task is that it must be selected from the pre-determined priority tasks which the UBC PULO are required to perform in the home environment, so that the simulator can realistically be used as both a screening and a training tool for

the potential users in the future. Meanwhile, the complexity of the task also has to be considered if the task is to be simulated for training new users.

Due to the above considerations, a virtual eating/drinking task was selected and developed. This task involves the reaching/picking up an object, moving the object along an angled plane. Based on both required arm motion and the necessary user's input commands during the task, the completion process can be split into three stages. In each stage, a specific goal is set for the user to achieve by following the necessary operational procedures. The next section outlines and discusses the details of the objectives in each stage of the task, the required user's input commands, as well as the expected and simulated arm motion in order to highlight the specific aspects of the designed task and evaluation process.

4.3 SIMULATION OF THE VIRTUAL TASK

The specifically designed virtual task was split into three stages to more easily facilitate simulation.

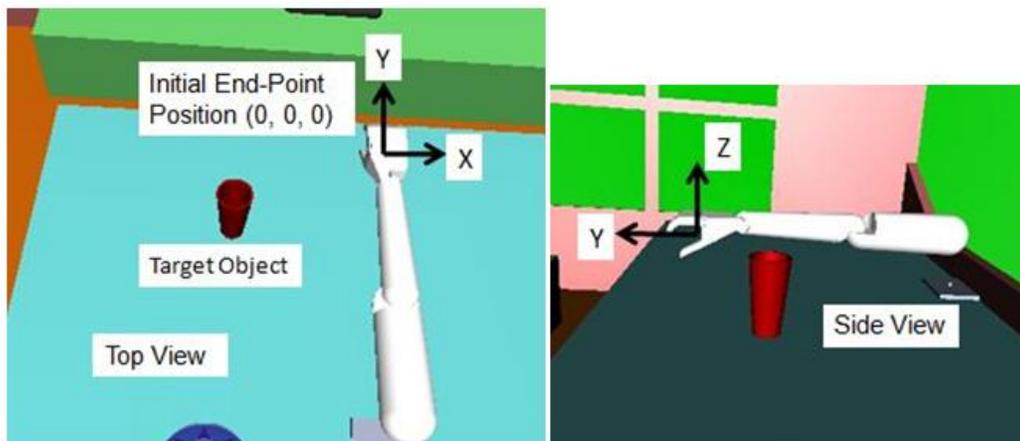


Figure 4.1: Initial Position of the End-Point in Virtual Environment

The details of the simulation in first stage of the task are provided below. Note that the initial position of the end-point (i.e. center of the hand) prior to the task is shown in Figure 4.1.

4.3.1 SIMULATION STAGE ONE

Objective: To reach and pick up the target object from the table.

Operational Procedures (Required User Inputs):

1. A short shoulder shrug to switch the operation mode of the simulator from "Neutral" to "Position: Table Top" mode while maintaining the initial head position.
2. Tilt the head slowly in the direction of the target object relative to the end-point. An example of the desired user's inputs is shown in Figure 4.2.

Expected arm motion: The virtual arm moves from the initial position toward the target object along the table surface plane (i.e. the X-Y plane)

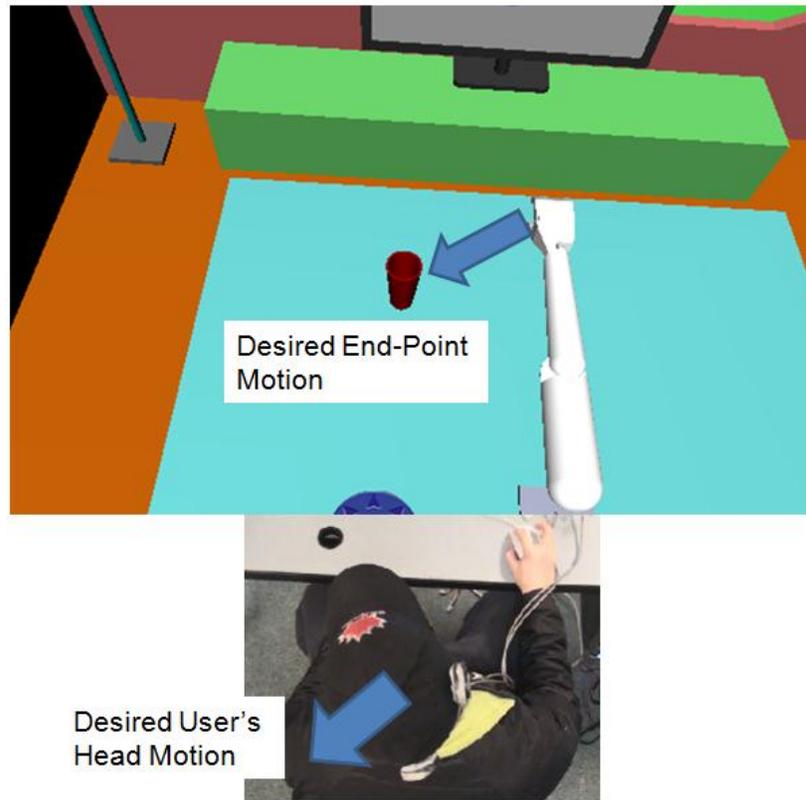


Figure 4.2: An Example of Desired User Input

3. Adjust the head's pointing direction slowly to correct the end-point path as it moves toward the target object.

Expected Arm Motion: The path of the arm is slightly altered based on the pointing direction of the head.

4. Adjust the tilt of the head to change the speed of the hand/end-point, as a greater tilt will increase the velocity of the end-point motion.

Expected Arm Motion: The velocity of the hand movement changes accordingly.

5. Restore the head to the initial "neutral" position when the hand reaches a position close to the target object.

6. Raise the shoulder sufficiently high to activate the "MEDIUM" switch on the shoulder controller, which lowers the hand in downward (-Z) direction. Release the shoulder back to the initial position while the hand is on the table surface.

Expected Arm Motion: The hand moves in the Z direction toward table surface, and the distance is 0.14m.

7. Repeat steps from 3 to 7 to move the hand to the position where it is close enough to grab the target object if necessary.

8. two short shoulder shrugs to switch the simulator from "Position: Table Top" to "Orientation" mode.

9. Perform the user's input commands outlined in Table 4.1 as desired to achieve a comfortable hand orientation to grasp the target object.

"Orientation" Mode	
User's Input Commands	Corresponding Arm Motions
Tilt head forward	Forearm rotation (supination)
Tilt head backward	Forearm rotation (pronation)
Tilt head to left	Wrist flexion
Tilt head to right	Wrist extension
Shrug shoulder to activate the "MEDIUM" switch of shoulder controller	Grasp to close
Shrug shoulder to activate the "HIGH" switch of shoulder controller	Grasp to open

Table 4.1: User's Inputs and Corresponding Arm Motions in Orientation Mode

By following the listed operational procedures, the objective of the stage one was accomplished. A series of screen images of the simulated arm motion are shown in Figure 4.3

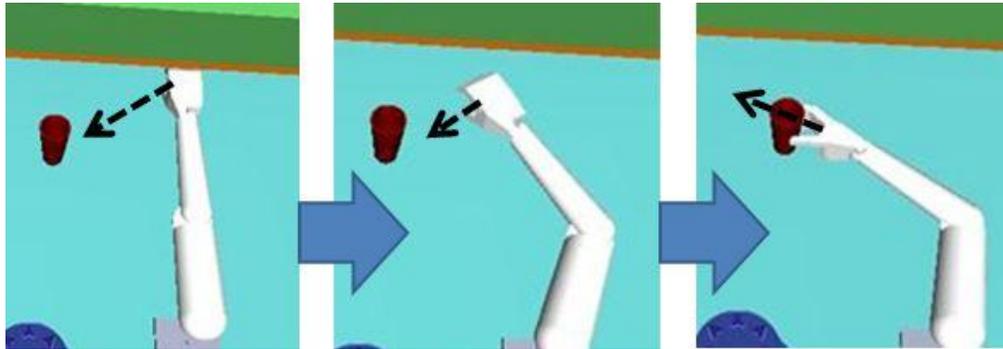


Figure 4.3: Simulated Virtual Arm Motion in Stage One

The simulated results in terms of the arm motion and joint angles that correspond to the user's input commands are shown in Figure 4.4 and Figure 4.5. The simulated arm motion corresponding to procedures 1 to 6 are as expected, showing that: a) while the end-point moves along the X-Y plane controlled by the head controller, the end-point maintains its vertical position (i.e. with respect to Z axis); and b) while the hand moves along the Z axis controlled by shoulder controller, the end-point maintains its position with respect to the X-Y plane. Corresponding to the procedure 4, Figure 4.6 demonstrates that the hand moves at different speeds depending on the magnitude of the user's head tilt. Corresponding to procedures 8 and 9 where the orientation of the hand is adjusted under "Orientation" mode, the end-point maintains its current position. Figure 4.5 shows the joint angles during the operation of this stage. The concurrent movement of the shoulder and elbow joints ensures the desired end-point path in "Position: Table Top" mode. When the "Orientation" mode is

selected by the user, the joints of forearm rotation, wrist flexion/extension, and grasp are controlled based on the user inputs. Meanwhile while in this mode, the shoulder and elbow joints maintain their current joint angles as expected to hold the end-point position in the X-Y plane.

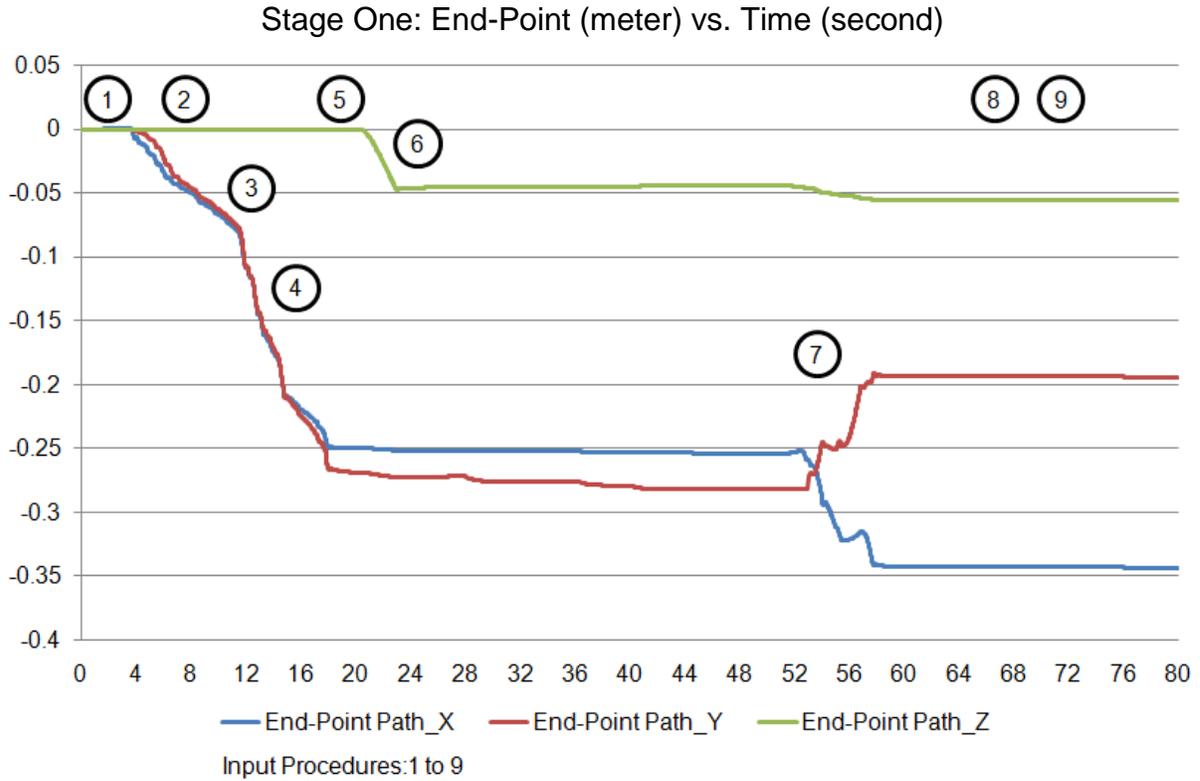


Figure 4.4: Simulated End-Point Path in Stage One

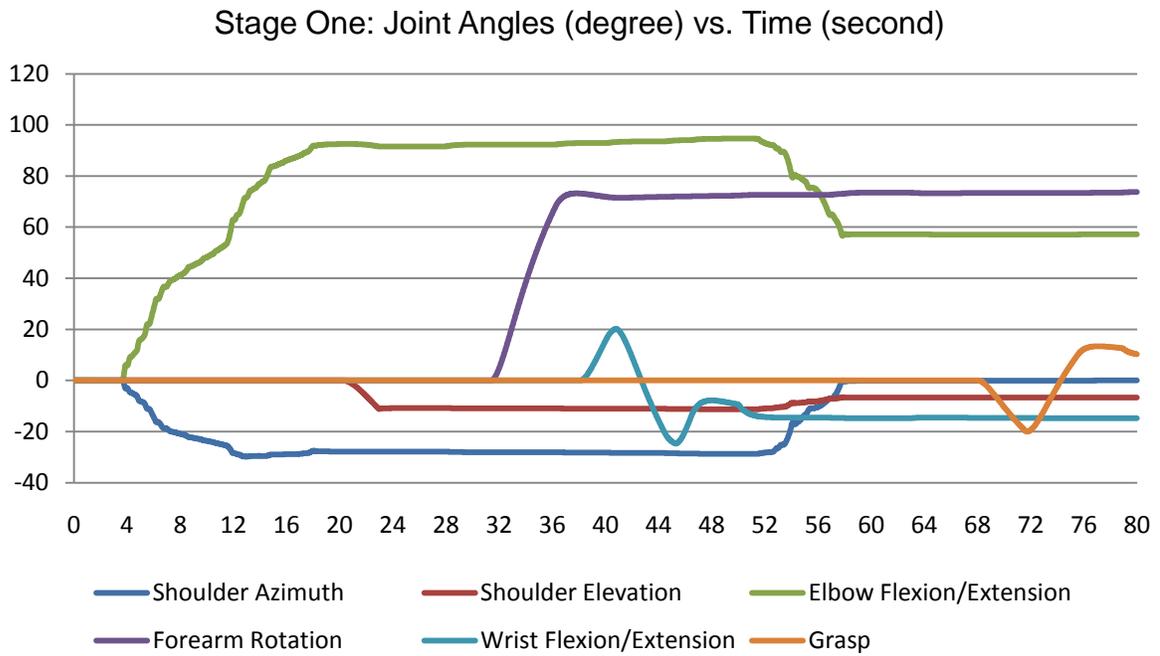


Figure 4.5: Simulated Joint Angles in Stage One

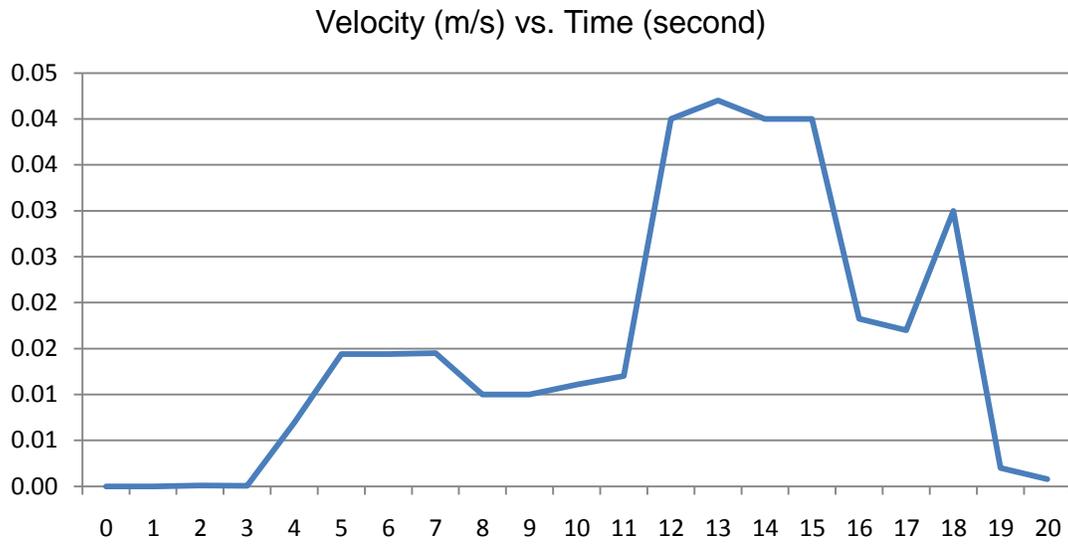


Figure 4.6: End-Point Velocity

In this stage of the task, the following control features are simulated.

Control Features Simulated in Stage 1 of the Task
<ul style="list-style-type: none">• Mode switching algorithm• Position: Table Top mode:<ul style="list-style-type: none">○ Head controller to control the hand on X-Y plane○ Head controller to adjust the speed of the hand○ Shoulder controller to move the hand in Z direction• Orientation mode: the combination of head and shoulder controllers to control the hand's orientation for picking up an object.

Table 4.2: Simulated Control Features in Stage One

The next stage of the task is designed to further test the end-point movement based on the referenced user's input commands, and also to test the safety features of the control system. The safety features implemented in the simulator include the neutral zone and the head tilting speed monitor that prevents accidental end-point motion from undesired user head tilt motion. The details of the safety features designed for the UBC PULO were reviewed in Chapter 2. The simulation conducted in this stage 2 differs from the previous stage in the aspect of the user's input commands. In stage 1, the direction of the end-point movement was selected, controlled and adjusted by the user. Therefore, the path from the initial end-point position to the target object was not unique. However, in stage 2, the user's input commands are given for simulating the expected end-point movement.

4.3.2 SIMULATION STAGE TWO

Objective: To move the target object to the left/right (along the X axis), forward/backward (along the Y axis) on the table surface (the X-Y plane), raise the target object up/down from the table surface (in the Z axis), and practice the safety features designed in the head controller.

Operation Procedures (Required User Inputs):

10. Switch the current operation mode to "Position: Table Top" by providing necessary numbers of short shoulder shrug motions.

11. Move the grasped object on X axis by tilting the head to left or right.

Expected Arm Motion: The hand should maintain the grasp of the target object, and move the object along the X axis only. The object should hold both the Y and the Z positions.

12. Move the grasped object along the Y axis by tilting the head forward or backward.

Expected Arm Motion: The hand should maintain the grasp of the target object, and move the object along the Y axis only. The object should hold both the X and the Z positions.

13. Move the grasped object along the Z axis by activating the shoulder controller

Expected Arm Motion: The hand should maintain the grasp of the target object, and move the object on Z axis only. The object should hold both the Y and the Z positions.

14. Test the design of the neutral zone by turning the head to left/right, or nodding.

15. Test the design of the head tilting speed monitor by simulating the action of sneezing.

Expected Arm Motion: No hand motion should be observed as the user's head motion is occurring in the neutral zone, and the speed of the user's tilting head motion exceeds the tilting speed threshold.

16. The procedure 15 and 16 can be performed anytime during this stage of the task.

17. Restore the head and the shoulder to the initial positions when the object is placed back on the table surface without releasing the grasp.

By finishing the above procedures, the stage two of the task was completed. A series of images during the simulation was captured as shown in Figure 4.7.

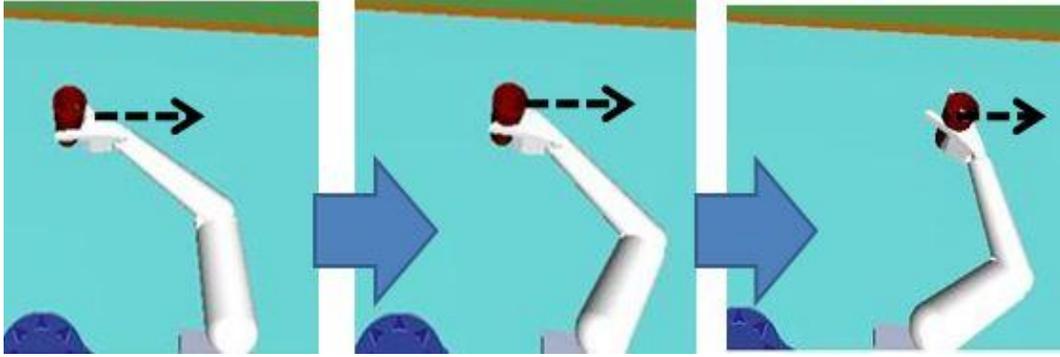


Figure 4.7: Simulated Virtual Arm Motion in Stage Two

The simulation results in terms of the end-point path and joints angles are shown in Figure 4.8 and Figure 4.9. Corresponding to the user's input procedures 11 to 13, the simulated motion of the end-point is as expected. Ideally, as the end-point moves along one of the X, Y, Z axis in "Position: Table Top" mode, the end-point should maintain its position on the other two axis. The Figure 4.8 shows that as the end-point moves along the X axis, there is a minor fluctuation in the end-point's path along the Z axis. The magnitude of this fluctuation is $\pm 1.5\text{mm}$, which is not observable during the simulation. Therefore; the fluctuation of the end-point on its path that is not observable during simulation is considered acceptable in this work. The joint angles experienced during this stage of the task are shown in Figure 4.9, the results demonstrating the capability of the inverse kinematics algorithm for computing the necessary joint angles of the shoulder and elbow to ensure the proper end-point path. Since the "Orientation" mode was not selected during this stage of the task, the forearm, wrist and grasp maintained their previous joint angles. Corresponding to the user input procedures 14 and 15, the end-point held its current position as

required, which means that the safety features of the control system as tested are tested are effective.

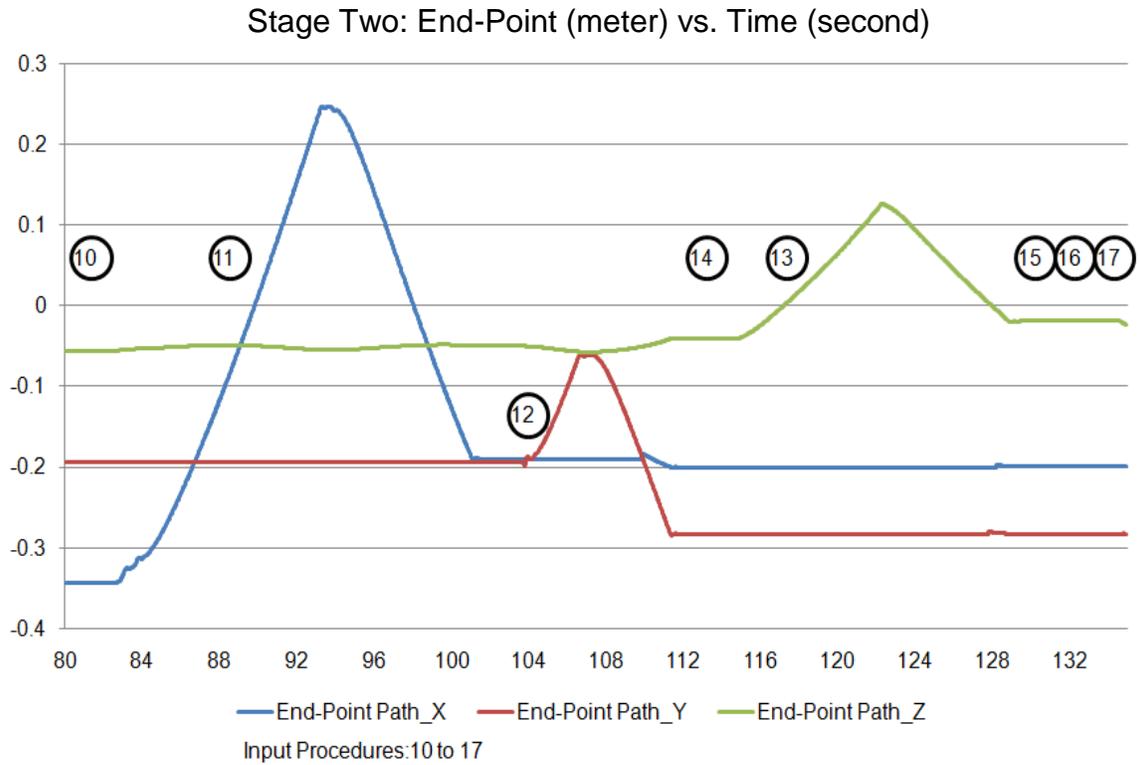


Figure 4.8: Simulated End-Point Path in Stage Two

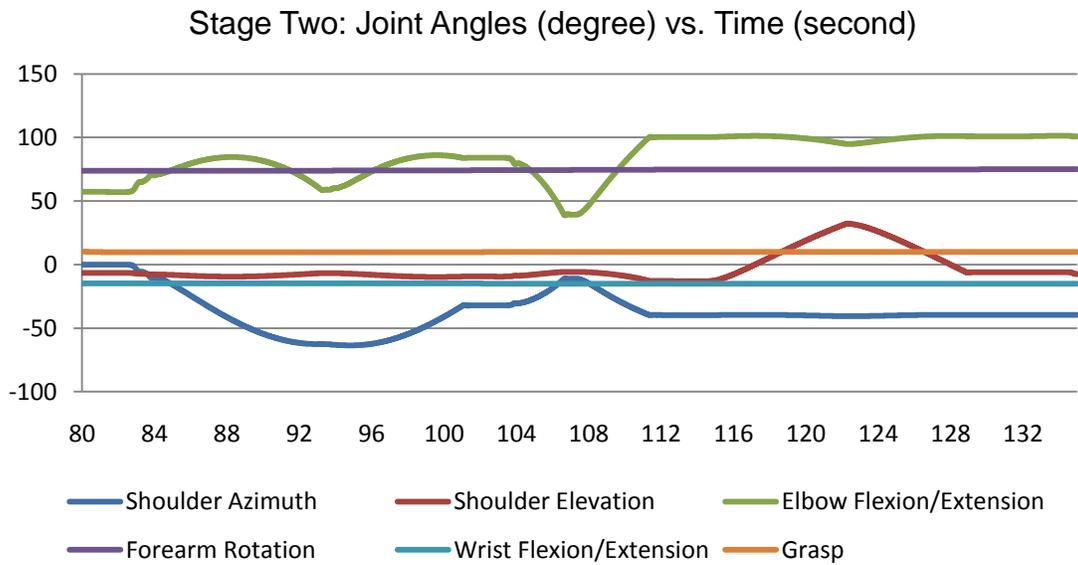


Figure 4.9: Simulated Joint Angles in Stage Two

In this stage of the task, the following control features are evaluated:

Control Features Included in Stage 2 of the Task
<ul style="list-style-type: none">• Inverse kinematics algorithm: Moving the hand along a single axis requires the concurrent movement of revolute joints from shoulder to elbow, where the unique combination of the joint angles is computed throughout the operation from the inverse kinematics algorithm.• The safety features in the head controller<ul style="list-style-type: none">○ Neutral zone○ Head tilt speed monitoring algorithm

Table 4.3: Simulated Control Features in Stage Two

The last stage of the virtual task was designed to test the control of the repetitive motions of the hand which are found in many of the prioritized tasks, such as drinking/eating and personal hygiene.

4.3.3 SIMULATION STAGE THREE

Objective: To move the grasped object from any location on the table surface to the user's mouth, and then move it back to the table surface.

Operational Procedures (Required User Inputs):

18. Switch the current operational mode to "Position: Functional" by providing necessary numbers of short shoulder shrug motions.
19. Activate the "HIGH" switch on the shoulder controller and hold the shoulder position to bring the target object toward the user's mouth.

Release the shoulder back to the initial position once the object is in the desired position.

Expected Arm Motion: The hand brings the target object to the user's mouth at a constant speed, and as the user's shoulder returns to the initial position, the object stays in the current position.

20. Activate the "MEDIUM" switch on the shoulder controller and hold the shoulder position to return the target object back to the table surface. Return the shoulder back to the initial position once the object is in the desired position.

Expected Arm Motion: The hand returns the target object back onto the table surface at a constant speed, and then the object stays in the current position.

21. Repeat steps 19 and 20 to simulate the repetitive motion involved in drinking/eating.

Stage 3 of the task is specifically designed to test the pre-programmed repetitive motions of the hand in the "Position: Functional" mode, i.e. the user-determined operational mode to enhance the functionality for the UBC PULO. In this case, the joint angles of the arm to bring the target object to user's mouth and the speed of the hand movement are pre-programmed, so that only the use of shoulder controller is needed to accomplish the task. Therefore, the head controller signals are disabled in the "Position: Functional" mode in the current simulation. Only using the shoulder controller as the required user inputs during

drinking/eating has the advantage of allowing the user's head to move freely to eat and drink. The simulated arm motion was captured as shown in Figure 4.10.

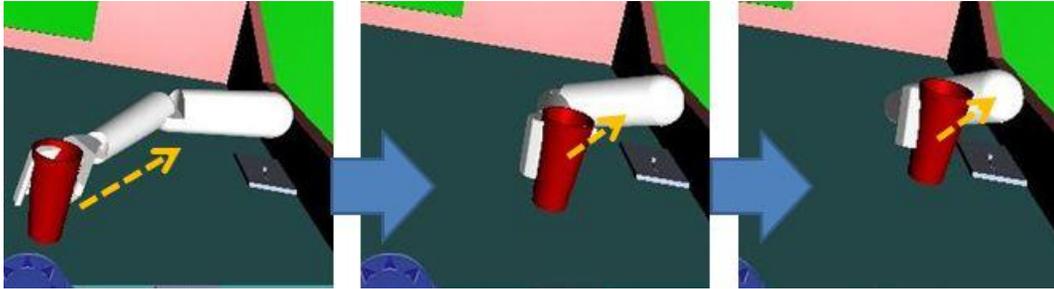


Figure 4.10: Simulated Virtual Arm Motion in Stage Three

The simulated results of this stage of the task are shown in Figure 4.11 and Figure 4.12. Corresponding to the user's input command procedures 18 to 21, the observed end-point motion is as expected. The end-point path results indicate the path of the repetitive end-point motion in this simulation, showing that:

- a) the decrease of the end-point path in the X and Y axes and the increase of the end-point path in the Z axis simulate the arm motion needed for bringing the target object toward the user's mouth;
- b) as the path of end-point motion along the X and Y axes moves from its low peak point to the high peak point, it simulates the motion of the hand bringing the target object back to the table surface.

Based on the user's preference, the speed of the joint under "Position: Functional" mode is pre-determined in this work.

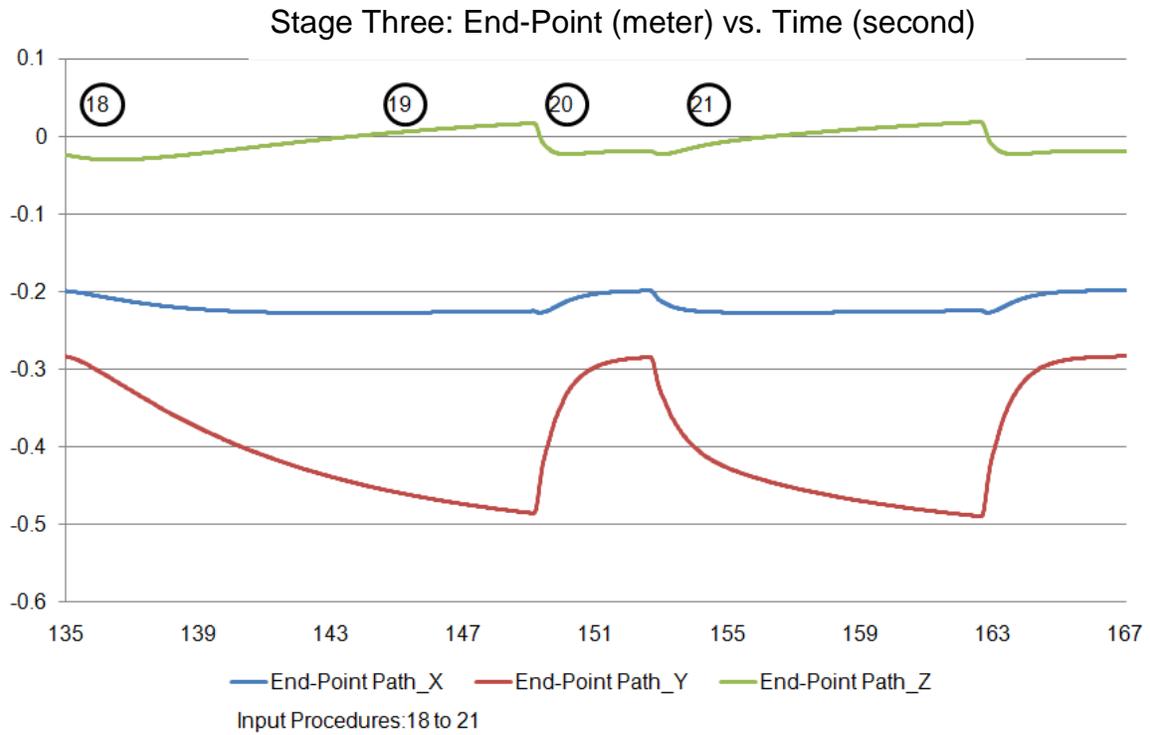


Figure 4.11: Simulated End-Point Path in Stage Three

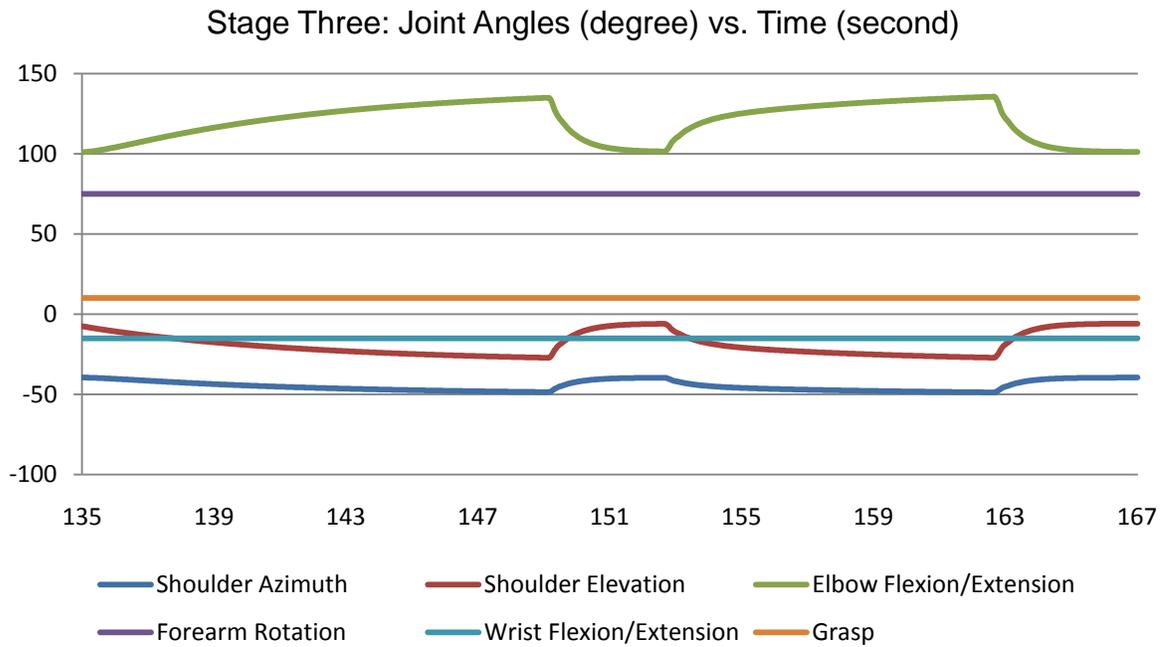


Figure 4.12: Simulated Joint Angles in Stage Three

4.4 SUMMARY

To utilize the virtual simulator for testing the performance of the UBC PULO control system and training the potential users, a specific virtual task of reaching/picking up the target object, moving the object, and drinking/eating was designed to be simulated in real-time. The inputs to the simulation were the user's head and shoulder movements that were acquired via the head and shoulder controller. The end-point path and the joint angles during the simulation have been presented and discussed. The simulation results indicate a high level of potential suitability of using the virtual simulator for screening and training of potential users for the UBC PULO based on its ability to quantify and visualize the users inputs to the PULO system. Additionally, throughout the simulation, the control system that was previously designed for the UBC PULO were tested. The results indicate that the design of the control system for the UBC PULO, as well as the head and shoulder controllers developed as the user control interfaces to provide control inputs, are quite capable of accomplishing the pre-defined daily-living tasks.

However; due to the fact that the simulation results may differ because of variation of different user's head and shoulder motions in real-time, it is necessary to tune the individual control component with referenced inputs to ensure the overall performance of the control system. Therefore, optimization for the UBC PULO control system is conducted and discussed, with the detailed procedure documented in the next chapter.

CHAPTER 5: POTENTIAL OPTIMIZATION

5.1 INTRODUCTION

As discussed in the previous chapter, the overall performance of the UBC PULO control system was tested through real-time simulation of a designed task operation. The control system of the UBC PULO contains a series of control components (see Figure 5.1), and each of the functional control features contributes to the overall performance of the control system. The goal of the optimization process in this work was to create the inputs for some control components, and tune the control algorithm by testing the corresponding outputs.

As demonstrated in Figure 5.1, this chapter documents the optimization process for the following control components:

1. Head controller algorithm: for interpreting the user's intention in "Position: Table Top" mode.
2. Inverse kinematics algorithm: for calculating the necessary joint angles to ensure the desired end-point motion.
3. PID controller: for minimizing the error between the referenced joint angles and the actually produced joint angles.

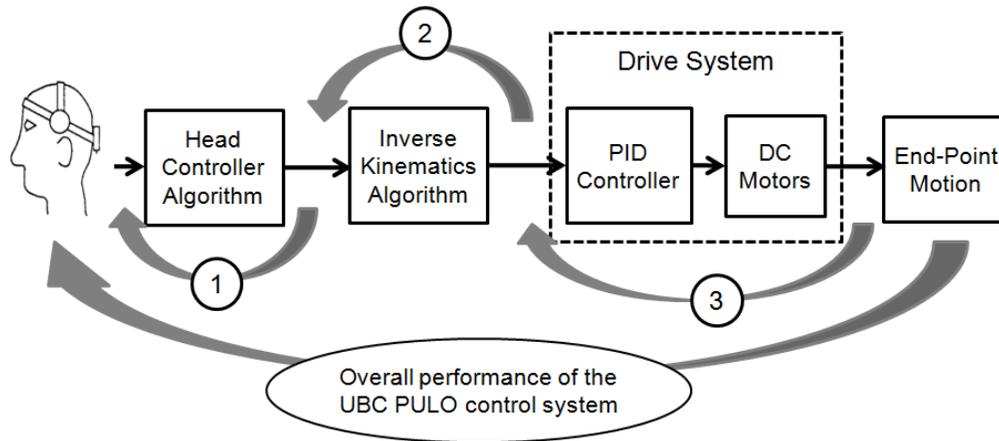


Figure 5.1: Scheme of Over-all Optimization Process

The completion of the above optimization processes ensures that the ultimate output of the UBC PULO control system, end-point motion, follows the control of the user's input.

5.2 INTERPRETATION OF USER'S INTENTION

In order to provide functional end-point control for the UBC PULO to perform the pre-defined priority daily living tasks, it first requires interpretation of the user's intention as provided through the developed user interface devices. As the user provides the needed body motions of head tilting and shoulder shrugging, the angular rotation of the head is measured around two axes by the inclinometers mounted on the head controller, and the translation of the shoulder joint activates the optical switches in the shoulder controller respectively. As a result, the input signals are first monitored to quantify the user's intention. The accuracy of the input signals is significantly affected by the setting of the parameters in the control algorithms of the head and shoulder controllers.

The head controller algorithm interprets the signals acquired from the inclinometers and produces corresponding control input signals. The inclinometers currently implemented for the head controller produce quadrature signals in terms of square waves as shown in Figure 5.2.

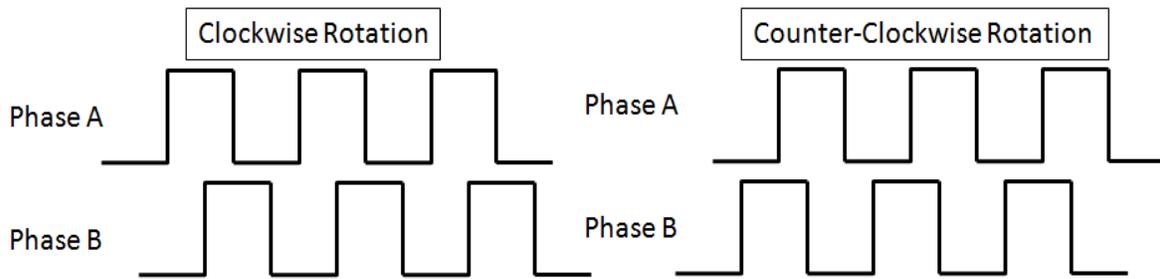


Figure 5.2: Quadrature Signals of the Inclinometers

The number of the square waves is proportional to the angle change that is observed from the head tilting motion. Thus, a counter was utilized in the Simulink model of the head controller algorithm, which outputs both the number of the square waves and the direction, with clockwise rotation set as positive. The key parameter in the counter setting is the frequency. The setting of the counter frequency in the model of the head controller algorithm determines how the readings from the inclinometers are being quantified as the input signals. The effects of the frequency setting are outlined in Table 5.1

Effects of the Counter Frequency Setting	
Frequency Setting	Effects
High	<ul style="list-style-type: none"> • Advantage: Maximize the sensitivity of the inclinometers in measuring the angles changes, as a result, the readings have a higher resolution to better detect small head movements.

	<ul style="list-style-type: none"> • Disadvantage: The sensitive response requires the user's head to maintain a nearly still posture to produce stable readings, which increases the difficulty of use.
Low	<ul style="list-style-type: none"> • Advantage: Less sensitivity means it is easier for the user to produce stable readings from the head controller. • Disadvantage: Less responsive to the user's head motion, consequently, the head controller produces discrete-time signals that may fail to measure some head motions.

Table 5.1: Effect of the Counter Frequency Setting

To balance the effects of the frequency setting, a frequency of 50HZ was utilized in the model of the head controller algorithm, which is capable of providing ease of use for the operator and sufficient sensitivity of signals for representing the user's head motion.

Another important control feature in the head controller algorithm is the design of the neutral zone and the head tilt speed monitor for detecting unintentional head motion, which will prevent undesired arm movement. The original design of the head controller algorithm was seen as a circle for the neutral and operation zones defined by points: a, b, c, d, A, B, C, D (shown in Figure 5.3); however, it was not realistic as determined from the experimental testing in this research.

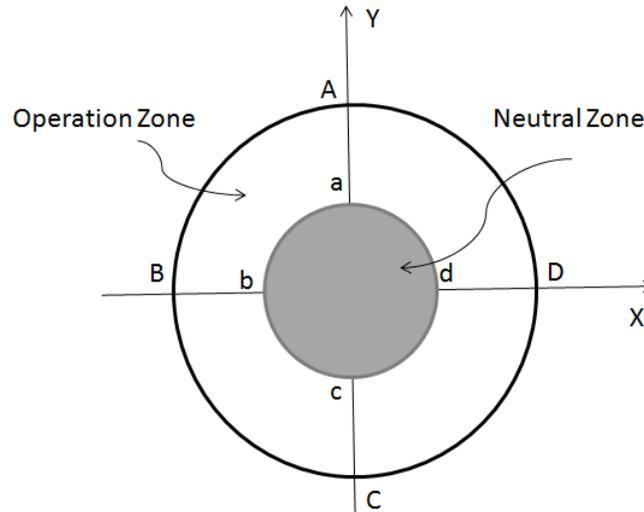


Figure 5.3: Original Design of the Neutral and Operation Zones

The experimental testing for defining the neutral and operation zones was discussed in Chapter 3. A user was wearing the head controller to simulate the undesired user input actions such as sneezing and nodding, meanwhile the corresponding readings from the inclinometers were recorded and used for defining the neutral zone. The operation zone was also determined based on the recorded readings of the inclinometers while the user's head tilted forward/backward and left/right without leaning the body. Based on one particular subject's head motion, the following testing results for defining the neutral and operation zones were obtained.

Neutral Zone:

- Simulated user inputs: head tilts forward/backward, as if for nodding
- Corresponding inclinometer: The one mounted on the side of the head controller
- Experimental results:

Forward/Backward		
Trial	Maximum	Minimum
1	145	-60
2	161	-47
3	147	-53
4	160	-70
5	169	-57
6	147	-100
7	135	-75
8	143	-68
9	127	-80
10	143	-90
Average	148	-70

Table 5.2: Experimental Results for Defining Neutral Zone Forward/Backward

- Simulated user inputs: head tilts left/right, as if for a conversation
- Corresponding inclinometer: The one mounted on the back of the head controller
- Experimental results:

Left/Right		
Trial	Maximum	Minimum
1	44	-45
2	44	-67
3	44	-43
4	37	-40
5	45	-49
6	51	-45
7	55	-60
8	47	-60
9	60	-46
10	65	-60
Average	47	-51

Table 5.3: Experimental Results for Defining Neutral Zone Left/Right

Operation Zone:

- Simulated user inputs: head tilts forward/backward, as if for controlling the end-point to move forward/backward.
- Corresponding inclinometer: The one mounted on the side of the head controller
- Experimental results:

Forward/Backward		
Trial	Maximum	Minimum
1	280	-148
2	262	-159
3	229	-141
4	232	-121
5	295	-142
6	269	-136
7	269	-149
8	256	-159
9	290	-134
10	275	-119
Average	266	-141

Table 5.4: Experimental Results for Defining Operation Zone Forward/Backward

- Simulated user inputs: head tilts left/right, as if for controlling the end-point to move left/right.
- Corresponding inclinometer: The one mounted on the back of the head controller
- Experimental results:

Left/Right		
Trial	Maximum	Minimum
1	231	-217
2	207	-214
3	201	-213
4	219	-223
5	233	-228
6	213	-216
7	239	-230
8	231	-205
9	225	-221
10	208	-221
Average	222	-219

Table 5.5: Experimental Results for Defining Operation Zone Forward/Backward

The numerical values of points A, B, C, D, a, b, c, d are shown in Table

5.6

	X	Y
a	0	148
b	-52	0
c	0	-70
d	49	0
A	0	266
B	-219	0
C	0	-141
D	221	0

Table 5.6: Points for Defining Neutral and Operation Zones

Therefore, the neutral and operation zones for this particular tested subject were determined as shown in Figure 5.4.

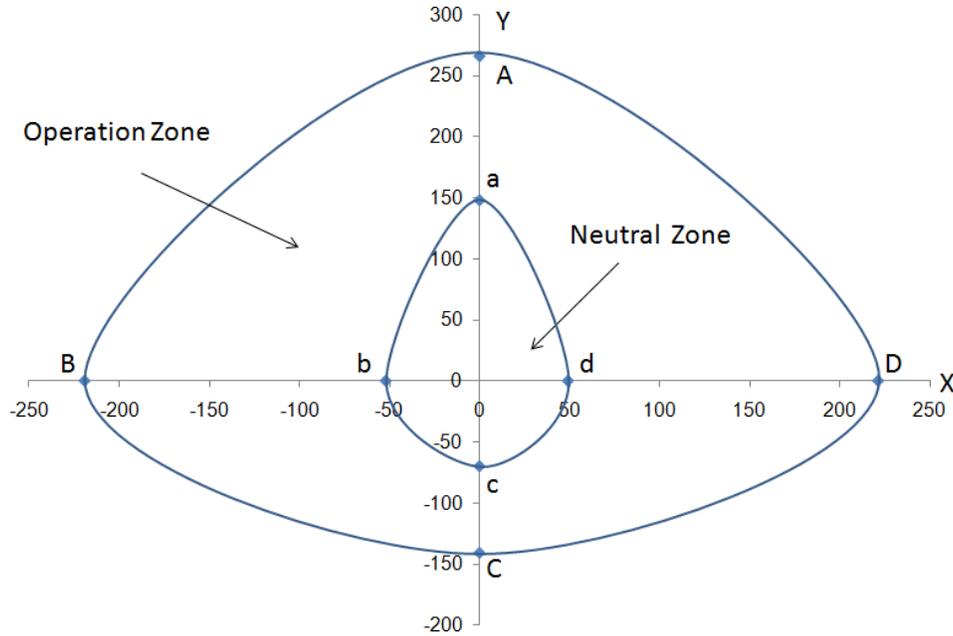


Figure 5.4: Defined Neutral and Operation Zones

Although the shape of the operation and neutral zones varies for different users, the result obtained from this experiment points out some general information about the user's head motion, which might be applicable to most of the users. A person has less range of neck motion in tilting the head backward than the range of neck motion in tilting the head forward, while maintaining a condition of eye sight fixed on the right hand. Without the additional algorithm to optimize the distance between the boundaries of operation zone and neutral zone, the magnitude of the user's head tilting backward is far less than the magnitude of the head motion to left/right. Since the desired speed of the end-point is designed proportional to the magnitude of the user's head tilting angles, consequently, the user would not be able to control the end-point backward with the same speed as to move the end-point to other directions. Thus, a scaling algorithm was added according to the desired end-point speed.

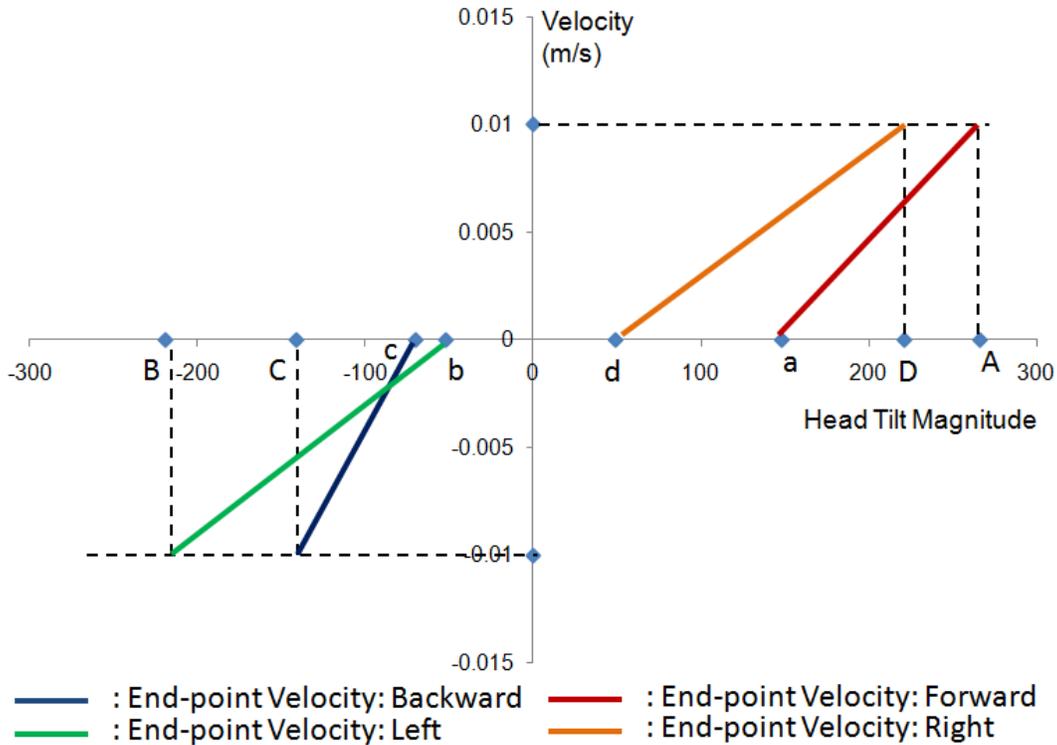


Figure 5.5: End-Point Velocity vs. Head Tilt Magnitude

As demonstrated in Figure 5.5, the horizontal axis represents the user's head tilt magnitude in terms of the inclinometer readings. Points a, A, b, B, c, C, d, D were plotted on this axis with positive pointing to right. The vertical axis represents the end-point velocity. This figure shows that when the user's input head motion is in the neutral zone, the corresponding end-point velocity is zero. While the user's head motion approaches to the boundary of the operation zone, the maximum allowed end-point velocity is produced in regard to the specific proportionality set for that head motion. As a benefit, scaling the range of head motion with the maximum end-point velocity enables the user to control the end-point on any desired speed within the safety limits to all directions in "Position: Table Top" mode. In this operation mode, the output of the head controller

algorithm is the desired incremental change of the end-point position on X and Y axis.

In order to generate the desired end-point movement that fully represents the user's intention, the head controller algorithm in "Position: Table Top" mode was further tested with referenced user inputs. In this work, the experiment was set as shown in Figure 5.6.

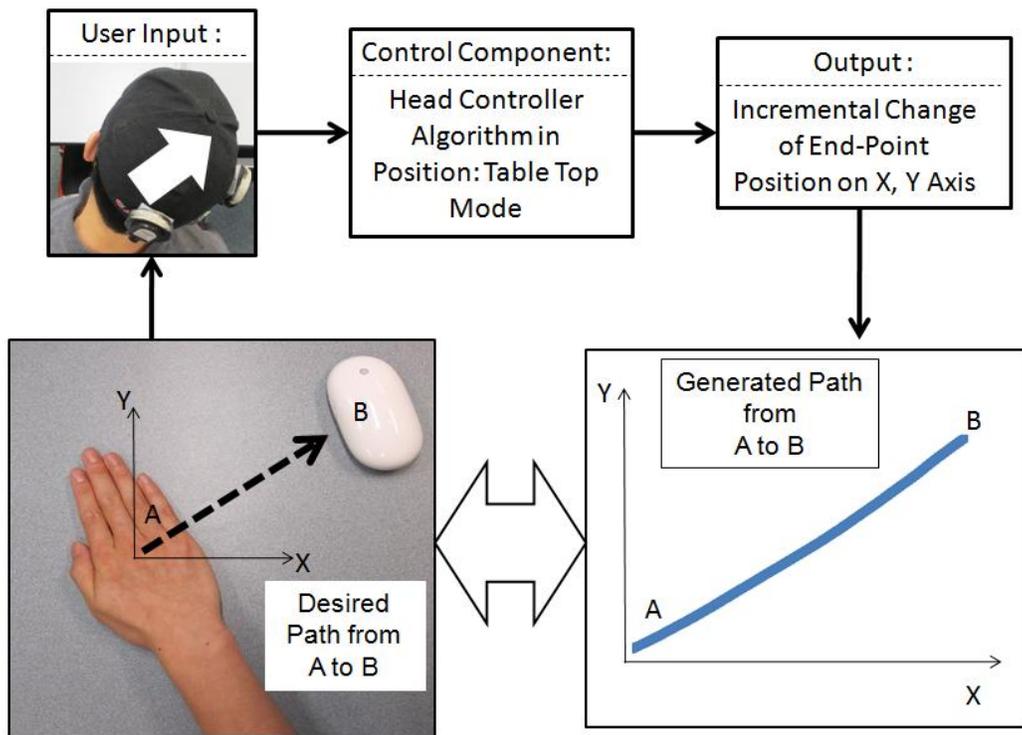


Figure 5.6: Schematic of the Head Controller Optimization

The user's hand and the target object (in this case, the mouse) were placed on the table surface, and "Position: Table Top" mode was selected. The user was asked to move end-point (Point A) directly toward the target object (Point B) via the control of head controller. Therefore, the user's intention of moving the end-point on the desired path from point A to point B was known and referenced as control inputs to the head controller algorithm. As the output of this

control algorithm, the incremental change of the end-point position on X and Y axis was produced in "Position: Table Top" mode. The data of the end-point changing position was recorded and used to generate the end-point path corresponding to the user's input head motion. The end-point path interpreted by the head controller was then compared to the desired end-point path as shown in Figure 5.7, and the error was computed as shown in Table 5.7. Thus, the user's intension of moving the end-point to the target object in "Position: Table Top" mode was interpreted via the use of head controller.

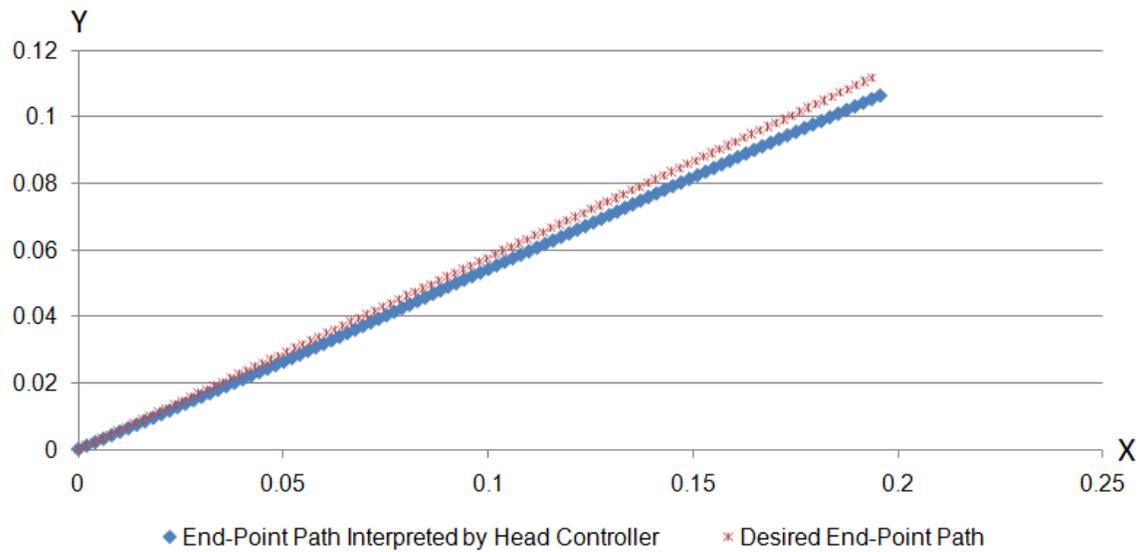


Figure 5.7: Results of the Head Controller Algorithm Optimization

	Error (mm)	Error Percentage
Maximum	5.3	6.7%
Average	1.8	3.5%

Table 5.7: Error of the Head Controller Algorithm

5.3 TESTING OF THE INVERSE KINEMATICS ALGORITHM

The next step in optimizing the control system was to test the performance of the inverse kinematics algorithm. As reviewed in Chapter 2, based on the desired incremental change of the end-point position, the inverse kinematics algorithm computes the necessary joint angles that move the end-point from the current location to the desired location. The calculation using this algorithm is guided by the inverse Jacobian matrix, and the model of the inverse Jacobian matrix for the UBC PULO was derived and documented in Chapter 3. In order to test the accuracy of this control algorithm, the end-point path generated from the head controller algorithm was utilized as the referenced inputs for the inverse kinematics algorithm to compute the necessary joint angles as demonstrated in Figure 5.8

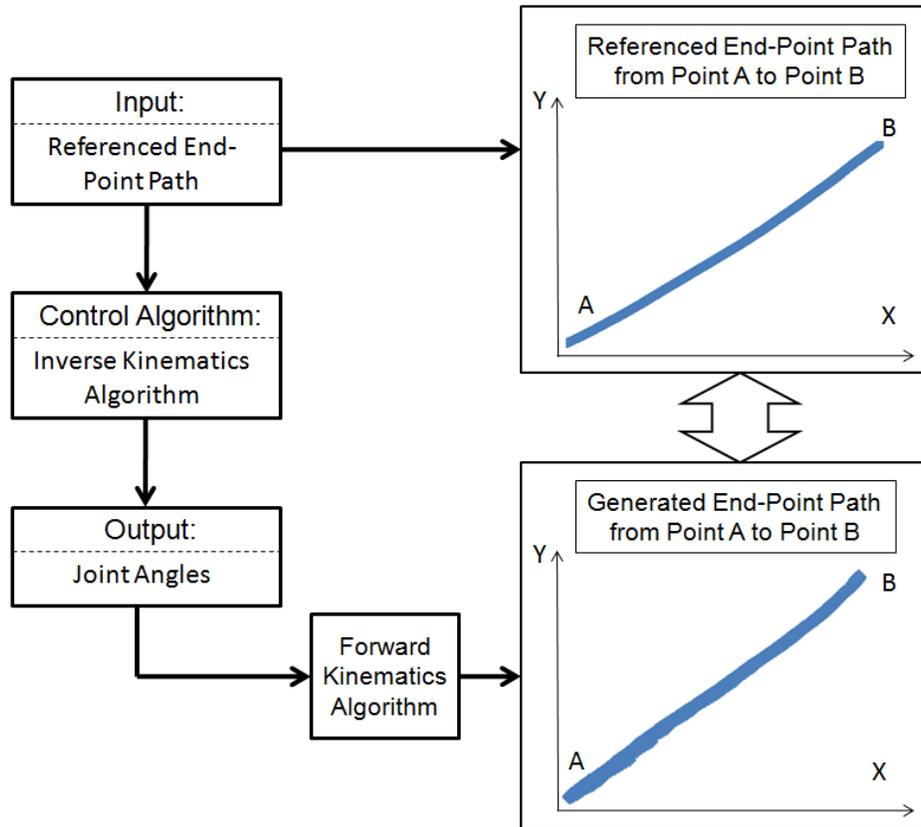


Figure 5.8: Schematic of the Inverse Kinematics Optimization

The necessary joint angles computed from the inverse kinematics algorithm were recorded and utilized for the forward kinematics computation. As discussed in Chapter 3, the forward kinematics algorithm developed for the UBC PULO calculates the current end-point position based on the joint angles. Therefore, the end-point path corresponding to the joint angles was generated. The generated end-point path from the inverse kinematics algorithm was compared to the referenced end-point path acquired from the testing of the head controller algorithm (shown in Figure 5.9), and the error was computed (shown in Table 5.8).

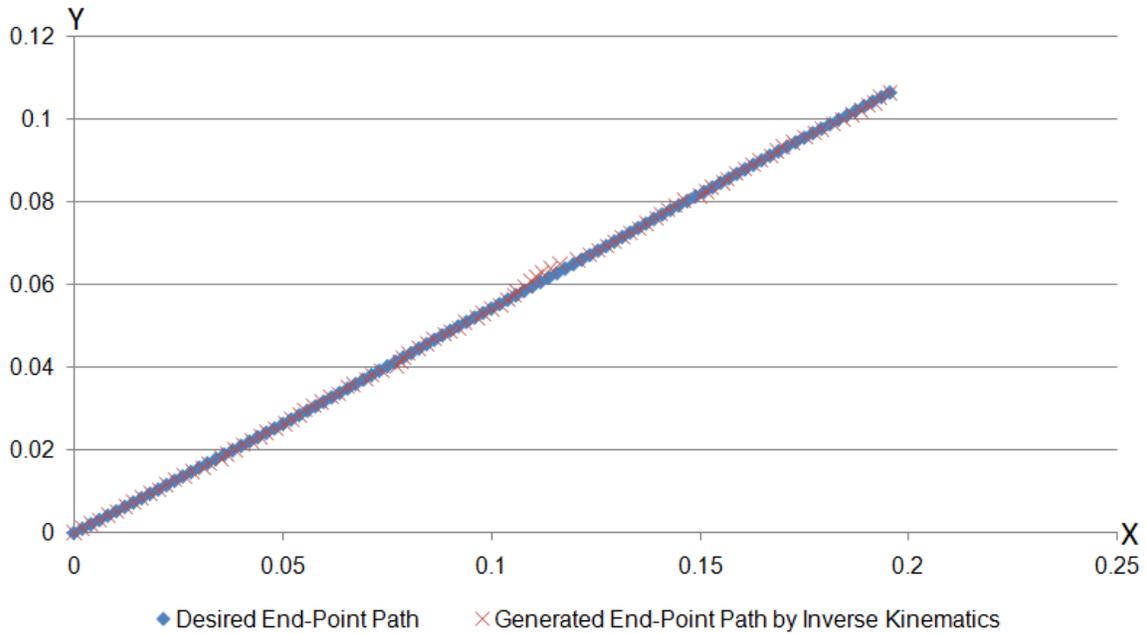


Figure 5.9: Results of the Inverse Kinematics Optimization

	Error (mm)	Error Percentage
Maximum	3.6	5.05%
Average	0.003	0.52%

Table 5.8: Error of the Inverse Kinematics

Figure 5.9 demonstrates that the model of the inverse kinematics developed for the UBC PULO is capable of calculating the necessary joints angles that moves the end-point on the path that follows the referenced inputs. Once the accuracy of the joints angles computed from inverse kinematics algorithm were tested, then, the joint angles were sent to the drive system as the referenced inputs.

5.4 TUNING OF THE PID CONTROLLER

It is critical that the PID controller minimize the error between the actually produced and the referenced joint angles. As demonstrated in Figure 5.10, the referenced joint angles are the inputs to the drive system which outputs the joint angles for creating arm motions. Therefore, a tuning process of the P, I, D gains are required.

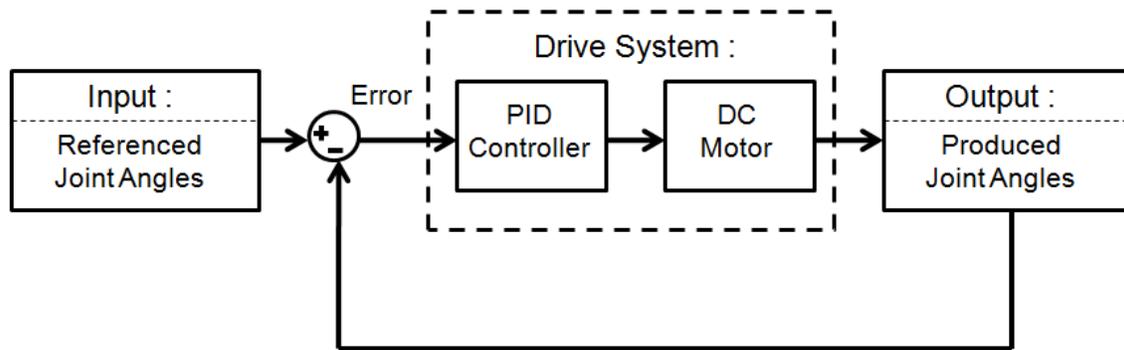


Figure 5.10: Schematic of the Drive System Optimization

In this work, the tuning process for the PID controller was conducted using the automatic PID tuner from Simulink with the guidance [26] discussed below.

The general rules for tuning the P gain for the PID controller is shown in Figure 5.11. With the referenced input at 1, the effect of changing the P gain on the response is demonstrated. Higher P gain produces smaller steady-state error, meanwhile, the stability of the response decreases. Lowering the P gain produces a slower response, however, it decreases the oscillation and increases stability.

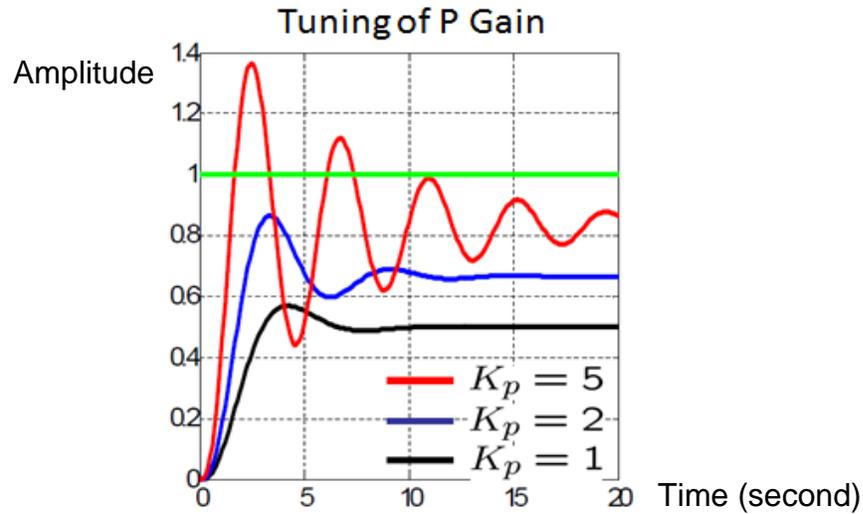


Figure 5.11: Guideline for Tuning P Gain [26]

The Figure 5.12 demonstrates the effect of tuning the I gain toward the system output with fixed P gain. Introducing I gain to the control algorithm provides zero steady-state error. Similar to the performance of P gain, higher value of I gain increases the response time to reach the reference input, at the same time, with increased I gain, more oscillatory response is expected.

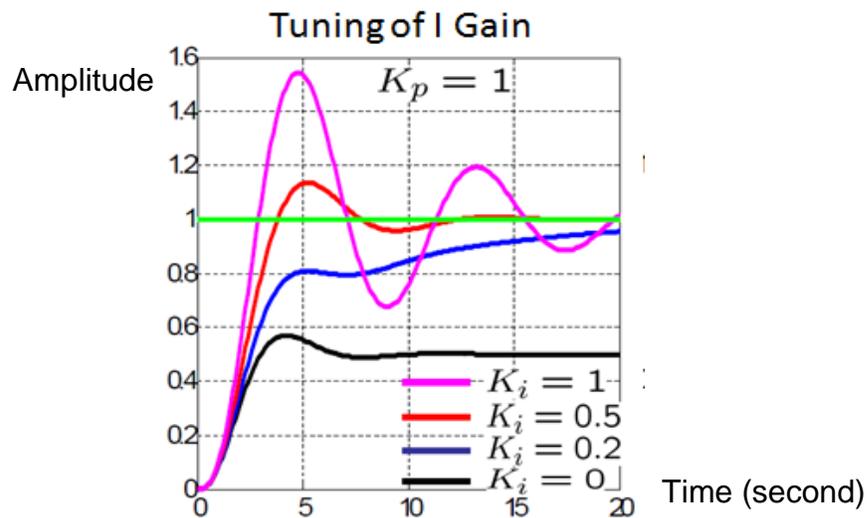


Figure 5.12: Guideline for Tuning I Gain [26]

With constant P and I gains, the system response corresponding to the changing of D gain in the PID controller is illustrated in Figure 5.10. Due to the existence of the I gain, the steady-state error is eliminated. Higher D gain dampens the response and reduces overshoot of the output. However, the overall performance could be worsened because of a high setting of the D gain.

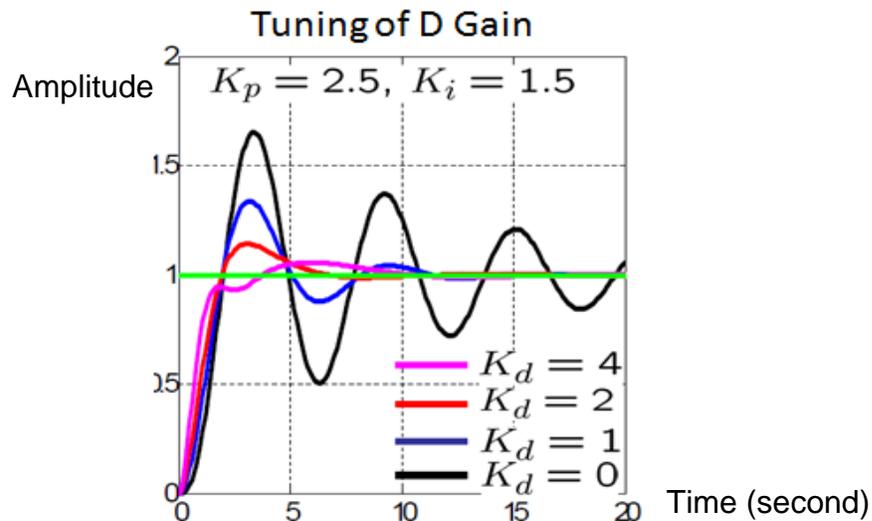


Figure 5.13: Guideline for Tuning D Gain [26]

With the general tuning rules discussed above, the model of the PID controller implemented for the UBC PULO was tuned. The first requirement of the tuning is that the oscillatory response needs to be eliminated. While multiple joints move concurrently during the operation of the UBC PULO, any oscillatory responses from the joints can create unpredicted and uncontrollable end-point motion which poses potential danger to the user. Another requirement is that the overshoot from the response needs to be reduced. Having overshoot in the response of the UBC PULO control system means that before the joints finally reach the referenced position, the joints are likely to pass the desired angles and slowly decrease back to the referenced angles. As a result of having the

overshoot in the response, the accuracy in controlling the end-point to the target position is greatly reduced. Another consideration in tuning the gains is the rise time. With higher rise time, the joint takes less time to reach the referenced joint angles, however, it may reduce stability in the response.

Therefore, the tuning process performed in this work aimed to eliminate the oscillation, reduce overshoot and balance the rise time and stability in the response. The Simulink PID tuner was used as the software tuning tool with a step input as the referenced input signal shape for the drive system to produce output responses. The PID tuner provides a Graphical User Interface (GUI) which allows the interactive tuning of the PID controller (see Figure 5.14) [21].

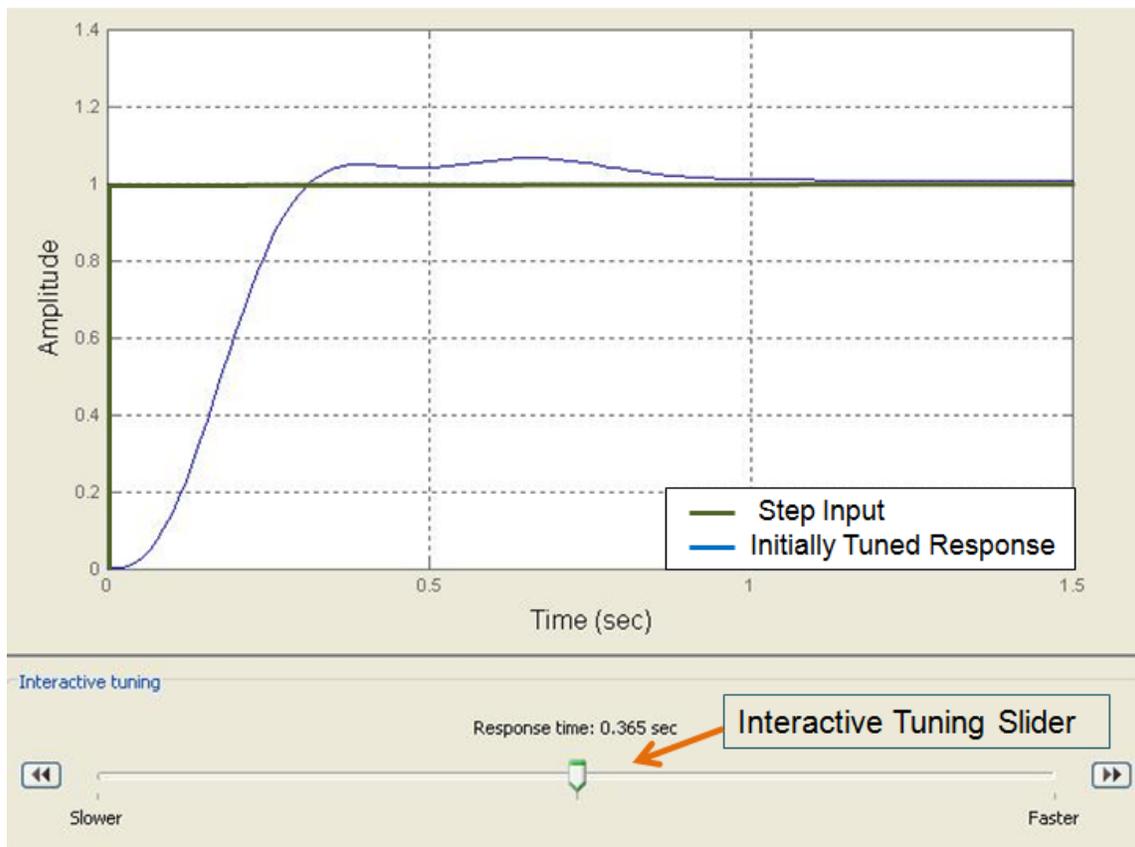


Figure 5.14: PID Tuner GUI

In this work, the initial values of the gains were automatically generated by the PID tuner based on the reference input signal and the system performance (i.e. the model of DC motors). The performance of the PID controller was further tuned by changing the response time until a desired system response was observed in the PID tuner GUI. The tuned P, I, D gains and the characteristics of the PID controller performance were listed in the following table.

P Gain	14.18
I Gain	0.34
D Gain	-1.41
Rise Time (sec)	0.45
Response Time (sec)	0.73
Settling Time (sec)	0.78
Peak Response	1.01
Steady-State Final Value	1

Table 5.9: P, I, D Gains and Response Characteristics

The PID controller response to a step input signal is shown in Figure 5.14. As a result of tuning the gains for the PID controller, the drive system was able to generate an output that well satisfies the stated requirements, since the oscillation in the response was eliminated, the overshoot was reduced, and the stability of the response was increased thus falling within an acceptable response time of 0.78 seconds.

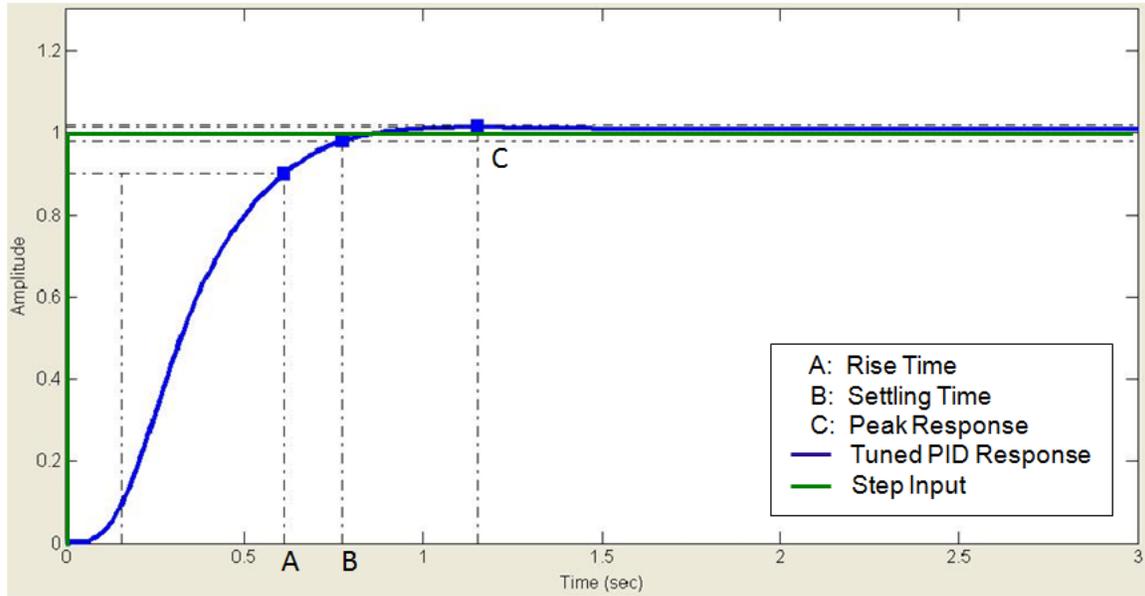


Figure 5.15: Tuned PID Controller Response

5.5 SUMMARY

Based on the results from the real-time simulation, the virtual simulator was utilized as a tool to optimize the performance of the individual control components. For the head controller algorithm, the frequency of processing the signals from the head controller was set to balance the accuracy of inclinometers measurement against the ease of use for the user to provide control inputs. By adding an additional algorithm that scales the operation zone related to the end-point speed, the head controller algorithm under "Position: Table Top" mode was optimized to interpret the user's intention accordingly to the user's input head motion.

Then, the generated end-point path from the head controller algorithm was used as the reference input for testing the inverse kinematics algorithm. The inverse kinematics computed the necessary joint angles which were then used to generate the actual end-point path. A comparison of the actually produced and

the referenced end-point path reveals that the model of the inverse kinematics algorithm developed for the UBC PULO is capable of converting the referenced end-point movement to the necessary joint angles to move the end-point on the desired path.

The performance of the PID controller was tuned for the position control of the DC motors by using the Simulink PID tuner. A step input signal was utilized as the referenced input. The output of the drive system was tuned in terms of eliminating the response oscillation, reducing the response overshoot and balancing the system stability and response time. As a result, the performance of the drive system meets the UBC PULO design requirements.

By completing the optimization process, it has demonstrated the performance characteristics and verified the feasibility of utilizing the developed simulator for testing and optimizing a range of individual control components to ensure the overall performance of the UBC PULO control system.

CHAPTER 6: DISCUSSION

6.1 INTRODUCTION

In order to test the current control system for UBC PULO prototype, a virtual simulator was developed through the modeling of the control system and the essential mechanical structures, tested a real-time simulation mode with designed virtual tasks, and optimized by testing individual control components. As the electrical and mechanical systems of the UBC PULO prototype are currently being optimized by other researchers and students, it is necessary that the virtual simulator developed in this research be capable of integrating future hardware designs. However, the limitations of the current simulator need to first be addressed.

This chapter discusses the limitations of the virtual simulator as they relate to the assumptions made in this research, and further elaborates on methods of incorporating potential new hardware designs.

6.2 ASSUMPTIONS

The predominant assumptions made in this work were required due to the hardware issues that currently exist in the UBC PULO prototype. Therefore, for the purpose of simulating how the UBC PULO was originally designed to perform, it was necessary to provide some basic assumptions during the process of modeling the drive system and other mechanical components, which are outlined in the following paragraphs as they relate to potential limitations.

The drive system implemented on the current orthosis prototype includes a series of DC motors, flexible drive cables and gearboxes at each joint of the orthosis arm. The DC motors are mounted on the torso body-cast to reduce the weight of the cantilevered orthosis structure, and the motor power is transmitted through individual flexible drive cables to each joint gearbox thus providing joint rotation and torque [35]. Another graduate student has been testing the response behaviour of both the flexible drive cables and the gearbox for specific joints of concern. Therefore, the specific model of the PULO flexible drive cables and the gearboxes were not included in the virtual simulator with plans to be incorporated at a later date. Thus as no other information was available, it was assumed that there is no significant twist (i.e. wind-up) on the drive cable and thus no time delay for the motion transmission. Additionally, the potential problems in the gearbox, such as backlash in the gear teeth, were also not modeled in the simulator at this point. Thus, the model of drive system in the simulator developed in this work output the joint angles which were directly sent to the model of the revolute joint in the physical model of the orthosis for creating the virtual motion.

The physical model of the UBC PULO prototype in this work was designed to model the mechanical structure of the orthosis arm in order to create virtual motions. Some assumptions were made while the orthosis arm was being modeled to expedite this process. As discussed in Chapter 3, due to the design issues in the orthosis arm, a new DOF, i.e. shoulder elevation, was introduced to replace the upper arm roll in the physical model. As a result, the geometry of the

shoulder mechanism in the simulator was different from the one implemented in the current prototype. The geometric properties of the orthosis arm, such as the dimension of the orthosis upper arm segment, the forearm segment, and the supporting structure for the human arm, were modeled based on the current prototype. Additionally, the material selected for the physical model of the orthosis arm was assumed to be aluminum as currently used in the PULO prototype.

6.3 LIMITATIONS

Due to the above assumptions made in the development of the virtual simulator, the following limitations in the performance of the simulator can be expected. Without modeling the drive cable in the drive system, the simulator is unable to simulate the effect of using flexible drive cables for transmitting motion from the DC motors to the joints. With the assumptions of having fully optimized backlash-free gearboxes, the possible backlash phenomenon in the joints was not modeled for simulation. With the assumptions made in the geometry of the orthosis shoulder mechanism and the material selection for the structure, the physical model of the orthosis is not fully representative to the current UBC PULO prototype in the aspects of the appearance and the weight.

Moreover, the selected software tools also limit the simulator to fully simulate the realistic UBC PULO operation in the virtual environment. To virtually display the arm motion directed by the orthosis, a virtual environment was created by using the Simulink 3D Animation toolbox. This particular toolbox lacks the feature of simulating the physical property of virtual objects, which means

that the objects displayed in the virtual environment are massless and thus only graphical representations. For instance, to simulate a collision of objects in the virtual scene, the motion of the objects needs to be pre-determined, otherwise, the objects pass through each other without stopping. The effect of this is that the interactivity between the user controlled virtual arm and the rest of the objects in the virtual scene is currently limited. Additionally, the weight of the human arm was approximated and added to the corresponding orthosis arm segments in the physical model in order to simulate the counter torque applied to the joints.

The limitations imposed in the current work due to the hardware problems can be eliminated upon the completion of the hardware optimization for the UBC PULO. Therefore, it requires the virtual simulator to be capable of adapting new hardware designs.

6.4 CAPABILITY OF INTEGRATING FUTURE HARDWARE DESIGNS

One of the design requirements for the virtual simulator was to provide the capability of integrating future hardware designs. The hardware in the UBC PULO prototype associated with the virtual simulator are the head controllers, the DC motors in the drive system and the mechanical structure of the orthosis arm.

For the head controller, the inclinometers implemented in the current UBC PULO prototype are expected to be updated with newer rotary sensors that are more power efficient, more accurate in measuring angular displacement and less bulky in size. The head controller algorithm was modeled to require minimum

changes to integrate the new head controller hardware. Based on the sensitivity of the new inclinometers, the counter frequency needs to be adjusted. Although the range of the neutral zone and operation zone need to be reset to each specific user, the same experimental method for defining neutral and operation zones developed in this work (discussed in Chapter 3) can be used in the future. The data acquisition (DAQ) set-up is also compatible with more advanced inclinometers for providing the interface between the user input control devices and the Simulink-based model of the simulator.

The DC motors implemented in the current UBC PULO prototype were modeled in terms of the transfer functions available in Simulink. To incorporate the property of the new DC motors, the parameters of each of the DC motor were selected and coded in a pop-up window where the value of each parameter in the DC motor models can be easily adjusted as shown in Figure 6.1.

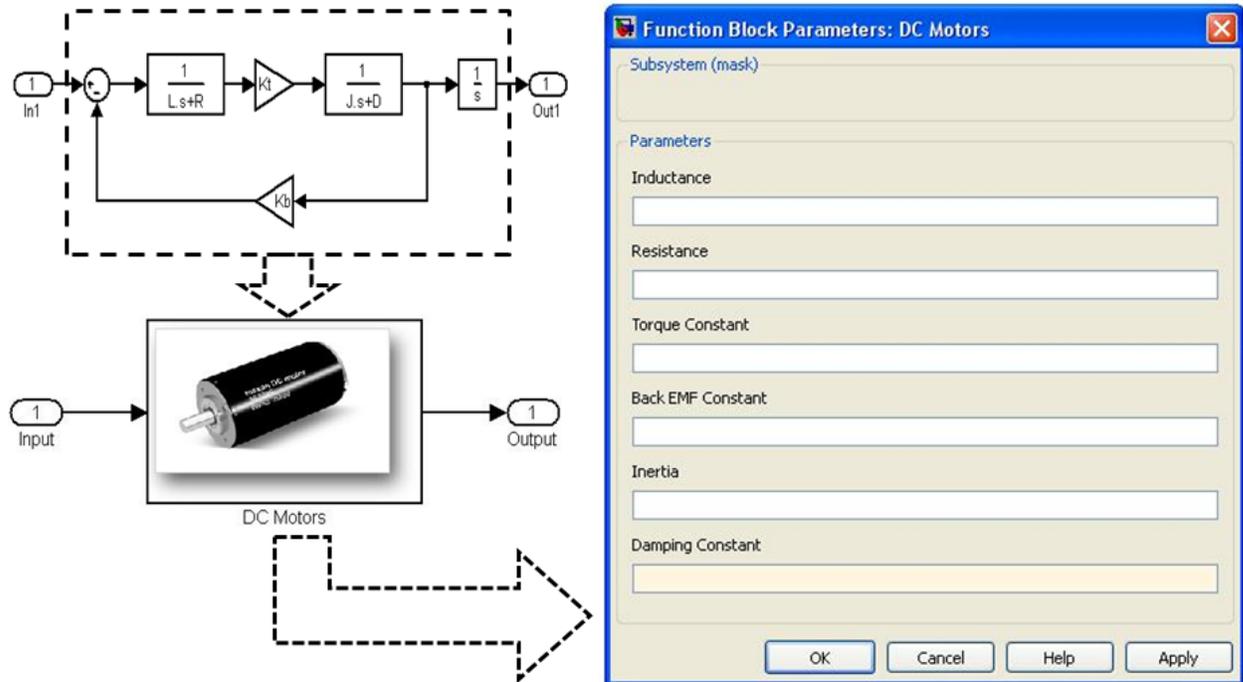


Figure 6.1: DC Motor Model

As the hardware optimization for the current prototype is progressively completed, a new physical model can be easily create to properly model the optimized orthosis arm structure. The method of developing the physical model for the current UBC PULO prototype was demonstrated in Chapter 3. The individual orthosis arm segments were graphically modeled in SolidWorks and assembled together according to the kinematics relationship between each segments. Then the assembly file of the orthosis arm in SolidWorks was converted to the physical model in SimMechanics. This method of developing a physical model allows the optimized orthosis prototype modeled in SolidWorks to directly update the current physical model for the virtual simulator, thus requiring minimal modifications for developing new physical models for the future orthosis arm structure.

6.5 SUMMARY

In order to simulate the design of the UBC PULO, a series of assumptions were made during the development of the virtual simulator. Consequently, the simulator developed was limited to be fully representative of the current PULO prototype. The assumptions made in this work and the corresponding limitations are summarized as following:

Assumption: The flexible drive cables implemented for the current PULO prototype do not twist, therefore, there is no torque loss or time delay between the ends of the cable.

Limitation: The model of the flexible drive cables was not included in the simulator drive system.

Assumption: The gearbox for each joint of the orthosis arm was assumed to function properly.

Limitation: The possible gearbox behaviours (i.e. backlash) was excluded in the simulator.

Assumption: The geometry and the mechanical property of certain orthosis arm segments (i.e. shoulder mechanism and material of the orthosis arm) were assumed.

Limitation: The simulator was limited to fully represent the appearance and the weight of the current PULO prototype.

Software limitation: The limitations from the Simulink 3D Animation toolbox reduced the interactivity between the user and the virtual

scene by not having the feature of simulating the collision phenomenon.

The stated limitations due to the hardware issues in the current prototype can be eliminated upon the completion of the hardware optimizations. Therefore, it requires the simulator developed in this research to have the feature of integrating the new hardware designs. The components in the simulator include the DAQ setup, the head controller algorithm, the drive system model and the physical model of the orthosis arm were developed to require minimum changes for implementing the new PULO hardware in the future.

In the next chapter, conclusion and future work are outlined based on all of the tasks conducted in this research.

CHAPTER 7: CONCLUSIONS AND RECOMMENDATIONS

7.1 INTRODUCTION

The overall goal of the UBC PULO program is to develop a highly functional powered upper-limb orthosis for people with flail arms to perform daily living tasks. A PULO prototype was previously designed and built, however, a design review concluded that it requires optimization in the areas of both the hardware design and the control system. Thus, the specific objectives of this research were to develop a virtual simulator which provides:

- 1) a design tool for researchers to model, test and optimize the current control system,
- 2) a screening and training tool for potential users to practice the control operation,
- 3) a platform with the capability to integrate future hardware designs.

The defined objectives were achieved upon the completion of the required tasks. All the specific work and development conducted in this research are summarized in this chapter, as well as specific research conclusions and recommendations for future work.

7.2 REVIEW OF POWERED UPPER LIMB ORTHOSIS

The state of the art of the powered upper limb orthoses, as well as the UBC PULO design, was first reviewed to gain background information needed for planning the current research tasks (Chapter 2). The previously developed powered upper limb orthoses (prior to the UBC PULO) resulted in limited

success because the device was either too complicated to control or suffered from a lack of functionality to sufficiently assist a user in the wide variety of daily living activities. Although, more recently developed devices improved the functionality and controllability in the orthosis design, the portability of the devices was greatly compromised. The UBC PULO was specifically designed to address the needs of people with both flail arms, and aimed to develop a device with functionality, portability and operability to assist the user to perform the daily living tasks.

Previously, a significant quantity of research and development work was conducted leading to the development of the UBC PULO by researchers and students under the leadership and supervision of Dr. Douglas Romilly. In order to carry on the current research work to optimize the current UBC PULO control system, the previously designed control concepts, strategy and developed user control interfaces were reviewed.

7.3 DEVELOPMENT OF THE VIRTUAL SIMULATOR

In order to optimize the current control system, a virtual simulator for the UBC PULO has been developed as documented as discussed Chapter 3. The tasks completed to develop the UBC PULO virtual simulator are itemized below with conclusions discussed as follow:

- A full review was performed of the available software tools and it was concluded that Matlab and Simulink, as well as their associated toolboxes, would provide the best solution for achieving the objectives set for this

research work. The model was then created using these selected tools and has resulted in a suitable model for this application

- To verify the installation, the user's inputs were successfully transmitted into the Matlab & Simulink in real-time via the implementation of the data acquisition (DAQ).
- The microcontroller implemented for the current PULO prototype was successfully modeled in Simulink. As a result, the control features designed for the UBC PULO control system are modeled and included in the simulator to mimic the performance of the actual microcontroller in the virtual environment. The modeled control system for the simulator includes: the head and shoulder control interfaces for interpreting the user's intention, the control strategy for the user to switch between operation modes, the inverse kinematics for computing the necessary joint angles, the PID controller for minimizing the error and the DC motors for creating joint motion.
- A virtual physical model of the UBC orthosis was created to represent the kinematic relationships of the orthosis arm segments. The method of converting the SolidWorks assembly model of the orthosis arm to the physical model in SimMechanics was investigated and utilized in this research. During the task of modeling the UBC PULO in SolidWorks, a design configuration problem in the current prototype design (i.e. non-concentric rotational centres in the upper arm) was identified and has been discussed to potentially solve this problem. This work as been

valuable as this early revelation has led other researchers to the task of redesigning the orthosis to solve this discovered problem.

- A virtual environment was developed to display the virtual arm motion, which currently provides the interface between the user and the simulator.

The completion of the above tasks concludes the development process of the virtual simulator in this work. The developed simulator consists of a Simulink-based model of the PULO control system, a physical model of the orthosis arm structure and a virtual environment, which provides a design tool for testing and optimizing the control system previously designed for the UBC PULO prototype.

7.4 SIMULATED TASKS

To test the developed virtual simulator, a specific virtual task was selected (i.e. reaching/picking up a cup with various stages designed for the user) to perform in real-time via the simulator (Chapter 4). The virtual task was split into three stages for the purpose of testing different control features modeled in the simulator. Throughout this simulation, the performance of the control system previously designed for the orthosis prototype was first time tested via the simulator. The specific goal designed into each stage of the task was successfully completed by following the necessary user input procedures. It concludes that the end-point control concept, the user control interfaces (i.e. head and shoulder controllers) and the control strategy previously designed and developed for the UBC orthosis are capable of providing a highly functional and operable control system for completing the predefined high priority tasks. The results of the simulation (in terms of the end-point path and joint angles during

the operation) also indicates a high level of potential suitability of using the virtual simulator for screening and training of potential users for the UBC PULO based on its ability to quantify and visualize the users inputs to the PULO system.

7.5 OPTIMIZATION FOR THE UBC PULO CONTROL SYSTEM

To ensure the overall performance of the UBC PULO control system, the individual control components in the simulator were optimized (Chapter 5) as their associated benefits to the orthosis development:

1. Head controller algorithm: Required for interpreting the user's intention in "Position: Table Top" mode.
2. Inverse kinematics algorithm: Required for calculating the necessary joint angles to ensure the desired end-point motion.
3. PID controller: Required for minimizing the error between the referenced joint angles and the actually produced joint angles.

For the head controller algorithm, the frequency of processing the signals from the head controller was optimized to balance the accuracy of inclinometer measurements against the ease of use for the user to provide control inputs. Based on the experimental results for defining the neutral and operation zones, a supplemental algorithm was added to the head controller algorithm under "Position: Table Top" mode to scale the operation zone related to the end-point speed. This optimization allowed the user's intention to be more accurately interpreted based on the user's input head motion. The comparison (provided in Chapter 5) of the desired end-point path generated by the user's input and the actual end-point path computed by the inverse kinematics algorithm verifies the

capability of the inverse kinematics algorithm model developed for the UBC PULO to convert the referenced end-point movement to the necessary joint angles to move the end-point on a desired path with an error of 0.52%. Lastly, the performance of the PID controller modeled for the UBC PULO simulator drive system was tuned to produce a rise time of 0.45 seconds, a response time of 0.73 second and a maximum over-shoot of 1%, which allowed the generated output to well satisfy the design requirements of the drive system.

7.6 CAPABILITY OF INTEGRATING THE FUTURE HARDWARE DESIGNS

The assumptions made during the development of the virtual simulator, as well as the corresponding limitations, were discussed (Chapter 6). Based on these assumptions, the developed simulator created in this work is limited in its ability to fully and properly mimic the performance of current UBC PULO prototype. However, these limitations can be eliminated upon the completion of the hardware optimization for the current prototype. The virtual simulator was developed in a manner such to require minimum changes to integrate the future hardware designs, including the data acquisition set-up for updated inclinometers for the head controller, the model of the drive system for optimized joints and new DC motors (all currently being developed within another research project), and finally the method of creating the physical model for the orthosis arm structure.

7.7 SUMMARY

Within this research, a virtual simulator was developed for the current UBC PULO prototype. The performance of the simulator was tested via real-time simulation of a virtual task. The results of the simulation indicate that the simulator provides a design tool with features capable of testing and optimizing the control system previously designed for the UBC PULO, screening and training potential users, and integrating the future hardware designs.

7.8 RECOMMENDED FUTURE WORK

The following recommendations are regarding potential future work as part of the UBC PULO development project:

- The mechanical components of the current UBC PULO prototype should be redesigned to solve the issue of the mis-aligned rotational axis between the human upper arm and the orthosis arm. The design of a new shoulder mechanism is recommended to solve the potential singularity issue.
- As the hardware in the current UBC PULO prototype is optimized, the features of the updated mechanical and control components should be modeled and implemented in the simulator to eliminate the limitations from the current work.
- To test the performance of the optimized UBC PULO prototype, a user should perform the prioritized tasks (i.e. reaching/picking up an object) both via the physical prototype and virtual simulator. The virtual arm

motion simulated in the virtual environment should be validated against the arm motion produced from the orthosis prototype in reality.

- The validated simulator should be used as a tool for screening candidates for the use of the UBC PULO. The user's capability of providing the required user inputs for operating the device should be tested via the simulator.
- The potential users selected for the UBC PULO should be first trained with the simulator before operating the physical prototype. During the training, all the control features designed for the UBC PULO control system should be tested via virtual task operation for the purpose of tuning the device prior to construction and commissioning.
- The developed simulator should also be used as a tool for collecting the user preferences (i.e. range of neck motion for defining neutral and operation zones, preferred end-point speed). Based on the collected data, the UBC PULO control system should be customized to provide solutions for each potential user's need, as is planned.

REFERENCES

1. Anglin, C. "A Functional Task Analysis and Motion Simulation for the Development of A Powered Upper-Limb Orthosis," *Masters Thesis*, Department of Mechanical Engineering, University of British Columbia, 1993.
2. Brown, D. "Control System Development for a Powered Upper-Limb Orthosis," *Masters Thesis*, Department of Mechanical Engineering, University of British Columbia, 1996.
3. Buckley, M. A., & Johnson, G. R. (1997). "Computer Simulation of the Dynamics of a Human Arm," *Journal of Engineering in Medicine* , Vol. 211 (5), 349-357.
4. Buckley, M. A., Yardley, A., Platts, R. G., & Marchese, S. S. (1995). "Development of the Specification for a Motorized Upper Limb Orthotic System," *IEE Computing and Control Division Colloquium on Mechatronic Aids for the Disabled*, (pp. 9/1-9/2). London.
5. Burdea, G. C., & Coiffet, P. (2003). *Virtual Reality Technology* (2nd ed.). Hoboken, New Jersey: Wiley-Interscience.
6. Cavallaro, E. E., Rosen, J., Perry, J. C., & Burns, S. (2006). "Real-Time Myoprocessors for a Neural Controlled," *Biomedical Engineering* , Vol. 53 (11), 2387-2396.
7. Craig, J. J. (2005). *Introduction to Robotics: Mechanics and Control* (3rd ed.). Upper Saddle River, New Jersey: Pearson Education, Inc.
8. Croft, E. (2009). "Robotics: Kinematics, Dynamics and Controls" [Class Notes] Retrieved from <https://www.vista.ubc.ca>
9. de Silva, C. W. (2007). *Sensors and Actuators: Control System Instrumentation*. Boca Raton, Florida: CRC Press.
10. Du, D. "Evaluation and Design of the Control System of a Powered Upper Limb Orthosis," *Masters Thesis*, Department of Mechanical Engineering, University of British Columbia, 1996.
11. Engen, T. J. (1985). "Development of Externally Powered Upper Extremity Orthotic Systems," *Journal of Bone and Joint Surgery* , 48B (3), 465-468.

12. Galway, R., Naumann, S., Sauter, B., Somerville, J., & Kurtz, I. (1991). *Evaluation of a Powered Orthotic Device for the Enhancement of Upper Limb Movement (PODEUM)*. Department of Rehabilitation Engineering, Hugh MacMillan Rehabilitation Centre, Toronto.
13. Johnson, G. R., Carus, D. A., Parrini, G., Scattareggia Marchese, S., & Vallengi, R. (2001). "The Design of a Five-Degree-of-Freedom Powered Orthosis for the Upper Limb," *Proceedings of the Institution of Mechanical Engineers*, Vol. 215, pp. 275-284.
14. Klamroth-Marganska, V., Nef, T., Guidali, M., & Reiner, R. (2009). "ARMin-Exoskeleton Robot for Stroke Rehabilitation," *World Congress on Medical Physics and Biomedical Engineering*. Vol. 25/9, pp. 127-130. Munich: IFMBE Proceedings.
15. Ligos, "V-Real Builder User's Guide and Reference," Retrieved from http://metalab.uniten.edu.my/~farrukh/vrml/user_guide.pdf. 2010
16. Maplesoft, "MapleSim User's Manual," Retrieved from <http://www.maplesoft.com>. 2009
17. Marco, G., Nef, T., & Reiner, R. (2009). "ARMin III - Arm Therapy Exoskeleton with an Ergonomic Shoulder Actuation," *Applied Bionics and Biomechanics*, Vol.6, 127-142.
18. MathWorks, "Real-Time Windows Target User's Guide," Retrieved from <http://www.mathworks.com/help/toolbox/rtwin/index.html>. 2010
19. MathWorks, "SimMechanics User's Guide," Retrieved from <http://www.mathworks.com/help/toolbox/phymod/mech/>. 2010
20. MathWorks, "Simulink 3D Animation User's Guide," Retrieved from <http://www.mathworks.com/help/toolbox/sl3d/>. 2010
21. MathWorks, "Simulink User's Guide," Retrieved from <http://www.mathworks.com/help/techdoc/>. 2009
22. Merlet, J. P. (2006). "Jacobian, Manipulability, Condition Number and Accuracy of Parallel Robots," *Journal of Mechanical Design*, Vol. 128 (1), 199-206.
23. Microsoft, "MS Robotics Developer Studio User Guide," Retrieved from <http://msdn.microsoft.com/en-us/robotics> Dec 2009

24. Mihelj, M., Nef, T., & Reiner, R. (2007). "A Novel Paradigm for Patient-Cooperative Control of Upper-Limb Rehabilitation Robots," *Advanced Robotics* , Vol. 21 (8), 843-867
25. Mihelj, M., Nef, T., & Riener, R. (2007). "ARMin II – 7 DoF Rehabilitation Robot: Mechanics and Kinematics," *2007 IEEE International Conference on Robotics and Automation, ICRA'07*, (pp. 4120-4125). Rome.
26. Nagamune, R. "Automatic Control,"[Power Point Slides]. from Dr. Ryozo Nagamune by personal communication. 2009
27. National Instruments, "DAQ M Series User Manual," Retrieved from <http://www.ni.com/pdf/manuals/371022k.pdf>. 2009
28. Nef, T., & Reiner, R. (2005). "ARMin-Design of a Novel Arm Rehabilitation Robot," *9th International Conference on Rehabilitation Robotics*, (pp. 57-60). Chicago.
29. Nef, T., Mihelj, M., & Reiner, R. (2007, July). "ARMin: A Robot fo Patient-Cooperative Arm Therapy," *Medical and Biological Engineering* , Vol.45, pp. 887-900.
30. Nef, T., Mihelj, M., Colombo, G., & Riener, R. (2006). "ARMin-Robot for Rehabilitation of the Upper Extremities," *2006 IEEE International Conference on Robotics and Automation, ICRA 2006*. Vol. 2006, pp. 3152-3157. Orlando: Institute of Electrical and Electronics Engineers Inc.
31. Nise, N. S. (2000). *Control Systems Engineering* (3rd ed.). New York, U.S.A: John Willey & Sons, INC.
32. Perry, J. C., Rosen, J., & Burns, S. (2007). "Upper-Limb Powered Exoskeleton Design," *Mechatronics* , Vol. 12 (4), 408-417.
33. Raschke, S. U., Romilly, D. P., Anglin, C., Gosine, R. G., & Hershler, C. (1993). "A Modified Powered Upper-Limb Orthosis", *The Canadian Yearbook of Prosthetics and Orthotics*.
34. Riener, R., Mihelj, M., Nef, T., Kiefer, G., Pemdl, C., & Muller, R. (2007). "ARMin-Exoskeleton for Arm Therapy in Stroke Patients," *2007 IEEE 10th International Conference on Rehabilitation Robotics, ICORR'07*, (pp. 68-74). Noordwijk.
35. Romilly, D. P. *Development of a Powered Upper-Limb Orthosis Prototype*. Department of Mechanical Engineering, University of British Columbia.

36. Romilly, D. P., & Du, D. *Control System and Shoulder Control Interface Development for the UBC Powered Upper-Limb Orthosis*. Department of Mechanical Engineering, University of British Columbia.
37. Romilly, D. P., Anglin, C., Gosine, R. G., Hershler, C., & Raschke, S. U. (1994). "A Functional Task Analysis and Motion Simulation for the Development of a Powered Upper-Limb Orthosis," *Rehabilitation Engineering* , Vol. 2 (3), 119-129.
38. Rosen, J., & Perry, J. C. (2007). "Upper Limb Powered Exoskeleton," *International Journal of Humanoid Robotics* , Vol. 4 (No. 3), 529-548.
39. Sciavicco, L., & Siciliano, B. (2000). *Modelling and Control of Robot Manipulator* (2nd ed.). Great Britain: Springer.
40. Sherman, W. R., & Craig, A. B. (2003). *Understanding Virtual Reality*. San Francisco, California: Morgan Kaufmann.
41. Stanger, C. A., Anglin, C., Harwin, W. S., & Romilly, D. P. (1994). "Devices for Assisting Manipulation: A Summary of User Task Priorities," *Rehabilitation Engineering* , Vol. 2 (4), 256-265.
42. Stuart, R. (1996). *The Design of Virtual Environments*. McGraw-Hill.
43. WorldViz, Vizard Manual, Retrieved from <http://docs.worldviz.com/vizard/> 2009

APPENDIX: UBC PULO FORWARD KINEMATICS

The D-H parameters used for computing the forward kinematics homogenous transformation matrix (T_{05}) in this research were define in the following table.

Name	Joint	θ_i	α_i (degree)	a_i (mm)	d_i (mm)
A_{01}	1	θ_1	$\alpha_1 = 90$	$a_1 = 0$	$d_1 = 0$
A_{12}	2	θ_2	$\alpha_2 = -90$	$a_2 = 270$	$d_2 = 10$
A_{23}	3	θ_3	$\alpha_3 = -90$	$a_3 = 0$	$d_3 = 0$
A_{34}	4	θ_4	$\alpha_4 = 90$	$a_4 = -50$	$d_4 = 330$
A_{45}	5	θ_5	$\alpha_5 = -90$	$a_5 = 150$	$d_5 = 0$

Table Appendix: D-H Parameters

Based on the homogeneous matrix, $A_{i-1,i}$, (discussed in Chapter 3), the transformation matrix for each joint was defined as following:

A_{01} : Shoulder Azimuth-to-Shoulder Elevation:

$$A_{01} = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A_{12} : Shoulder Elevation-to-Elbow Flexion/Extension

$$A_{12} = \begin{bmatrix} \cos(\theta_2) & 0 & -\sin(\theta_2) & 0.27 \cos(\theta_2) \\ \sin(\theta_2) & 0 & \cos(\theta_2) & 0.27 \sin(\theta_2) \\ 0 & -1 & 0 & 0.01 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A_{23} : Elbow Flexion/Extension-to-Forearm Rotation

$$A_{23} = \begin{bmatrix} \cos(\theta_3) & 0 & \sin(\theta_3) & 0 \\ \sin(\theta_3) & 0 & \cos(\theta_3) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A_{34} : Forearm Rotation-to-Wrist Flexion/Extension

$$A_{34} = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & -0.05\cos(\theta_4) \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & -0.05\sin(\theta_4) \\ 0 & 1 & 0 & 0.33 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A_{45} : Wrist Flexion/Extension-to-Grasp

$$A_{45} = \begin{bmatrix} \cos(\theta_5) & 0 & -\sin(\theta_5) & 0.15\cos(\theta_5) \\ \sin(\theta_5) & 0 & \cos(\theta_5) & 0.15\sin(\theta_5) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The over-all transformation matrix, T_{05} , was generated by multiplying the transformation defined for each joint.

$$T_{05} = A_{01} \cdot A_{12} \cdot A_{23} \cdot A_{34} \cdot A_{45}$$

$$T_{05} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where:

$$r_{11} = -\cos(\theta_1)(\cos(\theta_2)\sigma_2 - \cos(\theta_5)\sin(\theta_2)\sin(\theta_4)) - \sin(\theta_1)\sigma_5;$$

$$r_{12} = \cos(\theta_1)\sigma_7 + \sin(\theta_1)\sin(\theta_3)\sin(\theta_4);$$

$$r_{13} = -\cos(\theta_1)((\cos(\theta_2)\sigma_3 + \sin(\theta_2)\sin(\theta_4)\sin(\theta_5)) - \sin(\theta_1)\sigma_6);$$

$$r_{21} = \cos(\theta_1)\sigma_5 - \sin(\theta_1)(\cos(\theta_2)\sigma_2 - \cos(\theta_5)\sin(\theta_2)\sin(\theta_4));$$

$$r_{22} = \sin(\theta_1)\sigma_7 - \cos(\theta_1)\sin(\theta_3)\sigma_2 \sin(\theta_4);$$

$$r_{23} = \cos(\theta_1)\sigma_6 - \sin(\theta_1)(\cos(\theta_2)\sigma_3 + \sin(\theta_2)\sin(\theta_4)\sin(\theta_5));$$

$$r_{31} = -\sin(\theta_2)\sigma_2 - \cos(\theta_2)\cos(\theta_5)\sin(\theta_4);$$

$$r_{32} = -\cos(\theta_2)\cos(\theta_4) - \cos(\theta_3)\sin(\theta_2)\sin(\theta_4);$$

$$r_{33} = \cos(\theta_2)\sin(\theta_4)\sin(\theta_5) - \sin(\theta_2)\sigma_3;$$

$$P_x = \sin(\theta_1)(\sin(\theta_3)\sigma_4 - \cos(\theta_3)\sigma_8 + 0.01) - \cos(\theta_1)\sigma_1;$$

$$P_y = -\cos(\theta_1)(\sin(\theta_3)\sigma_4 - \cos(\theta_3)\sigma_8 + 0.01) - \sin(\theta_1)\sigma_1;$$

$$P_z = 0.27\sin(\theta_2) + \cos(\theta_2)(0.05\sin(\theta_4) - 0.15\cos(\theta_5)\sin(\theta_4)) - \sin(\theta_2)(\sin(\theta_3)\sigma_8 + \cos(\theta_3)\sigma_4$$

$$\sigma_1 = \cos(\theta_2)(\sin(\theta_3)\sigma_8 + \cos(\theta_3)\sigma_4) - 0.27\cos(\theta_2) + \sin(\theta_2)(0.05\sin(\theta_4) - 0.15\cos(\theta_5)\sin(\theta_4));$$

$$\sigma_2 = \sin(\theta_3)\sin(\theta_5) - \cos(\theta_3)\cos(\theta_4)\cos(\theta_5);$$

$$\sigma_3 = \cos(\theta_5)\sin(\theta_3) + \cos(\theta_3)\cos(\theta_4)\sin(\theta_5);$$

$$\sigma_4 = 0.05\cos(\theta_4) - 0.15\cos(\theta_4)\cos(\theta_5);$$

$$\sigma_5 = \cos(\theta_3)\sin(\theta_5) + \cos(\theta_4)\cos(\theta_5)\sin(\theta_3);$$

$$\sigma_6 = \cos(\theta_3)\cos(\theta_5) - \cos(\theta_4)\sin(\theta_3)\sin(\theta_5);$$

$$\sigma_7 = \cos(\theta_4)\sin(\theta_2) - \cos(\theta_2)\cos(\theta_3)\sin(\theta_4);$$

$$\sigma_8 = 0.15\sin(\theta_5) + 0.33;$$

This computed transformation matrix, T_{05} , describes end-point orientation and position based on the current joint angles $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$.