

2016

## Controller Synthesis of Multi-Axial Robotic System Used for Wearable Devices

Qian Mou

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

---

### Recommended Citation

Mou, Qian, "Controller Synthesis of Multi-Axial Robotic System Used for Wearable Devices" (2016).  
*Graduate Theses, Dissertations, and Problem Reports*. 6273.  
<https://researchrepository.wvu.edu/etd/6273>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact [researchrepository@mail.wvu.edu](mailto:researchrepository@mail.wvu.edu).

# **Controller Synthesis of Multi-Axial Robotic System Used for Wearable Devices**

**Qian Mou**

**Thesis submitted  
to the Benjamin M. Statler College of Engineering and Mineral Resources  
at West Virginia University**

**in partial fulfillment of the requirements for the degree of**

**Master of Science in  
Mechanical Engineering**

**Marvin H. Cheng, Ph.D., Chair  
Yu Gu, Ph.D.  
Steven Wheeler, Ph.D.**

**Department of Mechanical and Aerospace Engineering**

**Morgantown, West Virginia  
May, 2016**

**Keywords: controller synthesis, multi-axial, robotics, wearable devices  
Copyright 2016 Qian Mou**

# **ABSTRACT**

## **Controller Synthesis of Multi-Axial Robotic System Used for Wearable Devices**

**Qian Mou**

Wearable devices are commonly used in different fields to help improving performance of movements for different groups of users. The long-term goal of this study is to develop a low-cost assistive robotic device that allows patients to perform rehabilitation activities independently and reproduces natural movement to help stroke patients and elderly adults in their daily activities while moving their arms. In the past few decades, various types of wearable robotic devices have been developed to assist different physical movements. Among different types of actuators, the twisted-string actuation system is one of those that has advantages of light-weight, low cost, and great portability. In this study, a dual twisted-string actuator is used to drive the joints of the prototype assistive robotic device. To compensate the asynchronous movement caused by nonlinear factors, a hybrid controller that combines fuzzy logic rules and linear PID control algorithm was adopted to compensate for both tracking and synchronization of the two actuators.

In order to validate the performance of proposed controllers, the robotic device was driven by an xPC Target machine with additional embedded controllers for different data acquisition tasks. The controllers were fine tuned to eliminate the inaccuracy of tracking and synchronization caused by disturbance and asynchronous movements of both actuators. As a result, the synthesized controller can provide a high precision when tracking simple actual human movements.

## **ACKNOWLEDGEMENTS**

I would like to express my deepest gratitude to my advisor, Dr. Marvin Cheng, for his mentoring, encouragement and persistent guidance. His assistance and ideas were essential for the completion of this research.

I would also like to thank my committee member, Dr. Gu and Dr. Wheeler, for their time and advices in reviewing this thesis.

I am indebted to my colleague and friend, Lei Jiang, for his help and time along the master degree.

# TABLE OF CONTENTS

<b>ABSTRACT .....</b>	<b>i</b>
<b>ACKNOWLEDGEMENTS.....</b>	<b>iii</b>
<b>TABLE OF CONTENTS .....</b>	<b>iv</b>
<b>LIST OF FIGURES .....</b>	<b>vi</b>
<b>LIST OF TABLES .....</b>	<b>viii</b>
<b>CHAPTER 1.....</b>	<b>1</b>
<b>INTRODUCTION.....</b>	<b>1</b>
1.1 Research Background .....	1
1.2 Problem Statement.....	3
1.3 Research Objective .....	4
1.4 Document Outline .....	4
<b>CHAPTER 2.....</b>	<b>5</b>
<b>LITERATURE REVIEW .....</b>	<b>5</b>
2.1 Introduction.....	5
2.2 Development of Robot Control .....	7
2.2.1 Control of Robotic Systems .....	7
2.2.2 Multi-Axial Control .....	9
2.2.3 Wearable Devices Control.....	10
2.3 Synthesis Strategies of Multi-Axial Controller .....	12
2.3.1 Master-Slave Synthesis .....	13
2.3.2 Cross-Coupling Synthesis .....	15
2.4 Applications of Wearable Robotic Devices .....	19
2.4.1 Medical Applications – Robotic Devices .....	19
2.4.2 Medical Applications – Other Wearable Devices.....	23
2.4.3 Military Application.....	26
2.4.4 Wearable Virtual Reality Devices and Gaming Applications .....	30
2.5 Development Tendency .....	32
<b>CHAPTER 3.....</b>	<b>33</b>
<b>HARDWARE OF EXPERIMENTAL PLATFORM, MODELING, AND CONTROLLER SYNTHESIS.....</b>	<b>33</b>

3.1 Mechanical Hardware and Electrical Devices .....	33
3.1.1 Twisted-String Actuation .....	33
3.1.2 Mechanism of Assistive Robotic Device .....	37
3.1.3 Data Acquisition and Electrical Driver .....	40
3.2 Software .....	43
3.2.1 MATLAB Simulink .....	43
3.2.2 MATLAB xPC Target Tool .....	44
3.3 Design of Controller .....	46
<b>CHAPTER 4.....</b>	<b>54</b>
<b>EXPERIMENTAL RESULTS.....</b>	<b>54</b>
4.1 Introduction.....	54
4.2 Trajectory Tracking Results of Single Axis Movement .....	56
4.2.1 Results of PID Control .....	56
4.2.2 Results of Fuzzy Control.....	61
4.2.3 Gain Scheduling Based on PID and Fuzzy Control .....	65
4.2.4 Results of Sensor Fusion .....	68
4.3 Trajectory Tracking Results of Two Axial Movement .....	71
<b>CHAPTER 5.....</b>	<b>77</b>
<b>CONCLUSION .....</b>	<b>77</b>
5.1 Summary of Research.....	77
5.2 Future Works.....	78
<b>REFERENCES.....</b>	<b>80</b>
<b>APPENDIX A Arduino Code.....</b>	<b>A1</b>

## LIST OF FIGURES

<b>Figure 2.1</b> Scheme of Watt’s centrifugal governor .....	5
<b>Figure 2.2</b> XOS exoskeleton developed by Raytheon company for military purpose .....	6
<b>Figure 2.3</b> Wearable devices used for rehabilitation purpose .....	6
<b>Figure 2.4</b> Robot “Unimate” in General Motors factory .....	7
<b>Figure 2.5</b> “Shakey” robot invented by Charles Rosen group .....	8
<b>Figure 2.6</b> Block diagram of feedback control system .....	12
<b>Figure 2.7</b> Block diagram of feedback master-slave synthesis .....	14
<b>Figure 2.8</b> Block diagram of nonsymmetrical cross-coupled system.....	16
<b>Figure 2.9</b> Block diagram of the cross-coupled system.....	17
<b>Figure 2.10</b> Therapy is being conducted with a commercial version of MIT-MANUS .....	20
<b>Figure 2.11</b> ARMin mechanical structure with six degree of freedom .....	21
<b>Figure 2.12</b> The da Vinci Surgical System by Intuitive Surgical .....	22
<b>Figure 2.13</b> “Smart Stop” and “Ulcer” used for bad-habits correction .....	24
<b>Figure 2.14</b> Google leans and “QardioCore” used for diseases supervisal.....	25
<b>Figure 2.15</b> “BulletFlight 4.0.0” Mobile ballistics program for I-pod and I-phone .....	27
<b>Figure 2.16</b> Wearable health monitor for the military by odic.....	28
<b>Figure 2.17</b> Wearable Helmet Mounted Radar Program .....	29
<b>Figure 2.18</b> XOS 2 exoskeleton developed by Raytheon company .....	29
<b>Figure 2.19</b> XOS Virtuix Omni Game System .....	30
<b>Figure 2.20</b> Microsoft HoloLens .....	31
<b>Figure 2.21</b> Tesla Suit .....	31
<b>Figure 3.1</b> Single-Axial twisted string actuator .....	34
<b>Figure 3.2</b> Configuration of the twisted string system .....	35
<b>Figure 3.3</b> Operations of two twisted-string actuators used to rotate a pulley .....	36

<b>Figure 3.4</b> Mechanical structure of robot arm.....	37
<b>Figure 3.5</b> Overall structures of twisted string actuators .....	38
<b>Figure 3.6</b> Twisted string actuators experimental platform .....	39
<b>Figure 3.7</b> Schematic diagram of data acquisition of the testing platform .....	40
<b>Figure 3.8</b> Electrical structure of twisted string actuators control system .....	42
<b>Figure 3.9</b> Simulink graphical programmer of synthesis controller model .....	44
<b>Figure 3.10</b> Work scheme of xPC target with simulink .....	45
<b>Figure 3.11</b> Block diagram of twisted string actuator .....	48
<b>Figure 3.12</b> Block diagram of master-slave control architecture .....	53
<b>Figure 4.1</b> Numerical displacement of elbow joint when performing “drinking” motion.....	55
<b>Figure 4.2</b> Block diagram of PID control in a feedback loop .....	56
<b>Figure 4.3</b> Block diagram of simulation process.....	57
<b>Figure 4.4</b> Simulation results of PID controller with calculated parameters .....	58
<b>Figure 4.5</b> Results of PID control from different parameters. ....	59
<b>Figure 4.6</b> Block diagram of fuzzy logic control system.....	62
<b>Figure 4.7</b> Results of fuzzy control from different fuzzy logics .....	63
<b>Figure 4.8</b> Mechanism of gain scheduling method based on PID and fuzzy control .....	66
<b>Figure 4.9</b> Controlled result of combining both PID and fuzzy controllers without tuning .....	67
<b>Figure 4.10</b> Illustration of multi sensor data fusion in the system .....	68
<b>Figure 4.11</b> Curve fitting between string displacement and motor rotation .....	70
<b>Figure 4.12</b> Comparison between sensor fusion result and individual sensors .....	70
<b>Figure 4.13</b> Real elbow motion trajectory from human objective.....	72
<b>Figure 4.14</b> Controlled results of the dual twisted-string actuator . ....	73
<b>Figure 4.15</b> Repeatability test – 10 experiments without any load. ....	74
<b>Figure 4.16</b> Repeatability test – 10 experiments with a 5 N load.....	75
<b>Figure 4.17</b> Synchronization test.....	76



## LIST OF TABLES

<b>Table 3.1</b> Symbols of parameters used to describe motor dynamics .....	46
<b>Table 4.1</b> Ziegler-Nichols Method .....	60
<b>Table 4.2</b> Definition of variables used in fuzzy logic rules .....	64
<b>Table 4.3</b> Rules of fuzzy logic controller.....	65
<b>Table 4.4</b> Comparison of Root-Mean-Square-Deviation.....	71
<b>Table 4.5</b> Performance of the master-slave control scheme.....	74

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Research Background**

Most studies done before the adoption of the Affordable Care Act estimated shortages of at least 124,000 physicians and 500,000 nurses by 2025; and there is general agreement that the additional 32 million population covered by the Affordable Care Act would inflate the medical shortage estimations – by 31,000 physicians according to the Association of American Medical Colleges. The reason for the increasing shortage is the intersection of two trends: 1) growing demand of care, and 2) a supply of certified caregivers that is not keeping up with demand. Thus, between 2012 and 2030, total medical stroke-related costs are projected to be tripled, from \$71.5 billion to \$184.1 billion as well. To overcome the shortage of clinical physicians, nurses and caregivers, self-rehabilitation and self-care will become more and more important and assistive robotic devices will be able to partially fill in some of these identified gaps.

Recently, the advanced development of robotic industrial has improved human's productivity tremendously. It also played an important role in modern automatic control field. There are various kinds of robots adopted in different area according to different demands nowadays. In medical field, robotic devices have been adopted for rehabilitation for years already. Among all different factors, usability and user acceptance are typically the most important factors for the success of any assistive robotic solution, particularly assistive robots that are designed to help the groups of brain-injured patients and elderly people to live independently. The design of assistive robots to help with self-care depends strongly on the specific culture of the user. The ideal robotic solutions must cater for any unique individual user needs and take into consideration the user's socio-demographic profile as well. However, most existing self-care robotic systems aim to replace a specific function that users/patients can originally perform instead of assisting them to gain back their original capabilities. Thus, these “assistive” devices might not fit into the cultures of

the user's daily activities. As a result, in the past few years, several researches investigated portable robotic systems that truly assist users by providing additional force instead of replacing the function of the whole arm.

In the past few decades, many researches are focused on wearable rehabilitation devices and many bio-robots are introduced and adopted in this field to help patients with their body movements. One of them named powered exoskeleton [1], which is a mobile machine consisting primarily of an outer framework power by a system of motors, hydraulics or pneumatics, is a typical wearable device that provides part of the energy for limb movement. With the help of this kind of devices, patients can move limbs as they need whenever they are undergoing inconvenience. Besides, development of such a kind of wearable devices is useful for other applications as well.

Exoskeleton robotic systems that are used to provide or assist arm motions have been developed by various groups. These devices supporting users' daily tasks such as holding, lifting, moving, and free mobility are called as assistive robotics. One important objective of developing these assistive robotic systems is to support disabled people to continue their lives without any help from others. To fabricate an adequate mobile robotic system, there are three important challenges: 1) to generate sufficient range of force with acceptable weight and portable power source; 2) to identify user's current motion and generate corresponding movements of the device; and 3) to control and coordinate angular movements of all joints.

To reduce the size of the actuators and to minimize the weight that needs to be carried by users, twisted string actuators have demonstrated excellent properties and drawn much attention. The basic idea of this type of actuators is to change the linear length of two or more strands by twisting them. The change of length can result in a linear motion of the other end. With two actuators, emulation of contraction/extension of muscles can drive the joint. With its high reduction ratio of length, an actuation system can be small and lightweight. However, it is generally difficult to synthesize a controller to fulfill desired performance and robustness of such a dual-axis mechanism. To apply multi-axial control strategy to an assistive robotic system, it is important to compensate for the disturbance

and nonlinearity of the adopted electromechanical components caused by the arm movement. In particular, one specific characteristic of the assistive robotic system of upper limbs is that the inertia keeps changing while the arm is moving. The goal of this study is to develop an adequate controller that can control an assistive robotic arm with multi-axial movement to follow real human arm trajectories.

## **1.2 Problem Statement**

In wearable device control field, wearable devices are usually actuated by hydraulic or pneumatic power sources, so they are relatively huge and heavy in size. This is desired in manufacture and military fields because it is firm enough and can endure heavy load. However, it might not be very convenient for rehabilitation propose because they are too heavy to wear for a stroke patient. For this reason, a more portable and less-weighted actuator system in this field has been introduced and developed in the past two decades. It's the Twisted String Actuator System [3].

A single twisted string actuator is relatively easy to model and control. But when performing a little bit more complicated movement such as rising hands, at least two degrees of freedom have to be concerned, which means more than two axes of moving are required in the twisted string actuator robotic system. The main issue come up with the addition of another axis is synthesis. It's critical for a multi-axial robot to finish tasks synergistically otherwise some unexpected behavior will happen. So, the control of synthesis is desired and most important when modeling a multi-axial robotic system. In this work, a four axial twisted string actuator system will be built and divide into two groups corresponding to two parts of a robotic arm which are shoulder joint and elbow joint. Then, a corresponding controller will be designed and modeled to control the robotic arm.

### **1.3 Research Objective**

This study aims to design a complete controller that can control the prototype of an existing robotic arm to mimic simple human movements. In order to realize it, three major steps and objectives have to be fulfilled. They are:

1. Setup an experimental platform with necessary hardware, control circuits, and software for an existing mechanical arm actuator system;
2. Develop an effective real-time controller that can control a single joint of the existing robotic arm accurately; and
3. Improve the performance of controllers by integrating different method with gain scheduling approach.

### **1.4 Document Outline**

The organization of this thesis is as following. The second chapter is the literature review of robotic devices and different control theories that compensate for multi-axial devices. The modeling process and basic control methods are introduced in the third chapter. Chapter four describes various approaches of compensator design. In this chapter, the precision and controlled performance are discussed. In the fifth chapter, a summary and future works are concluded.

## LITERATURE REVIEW

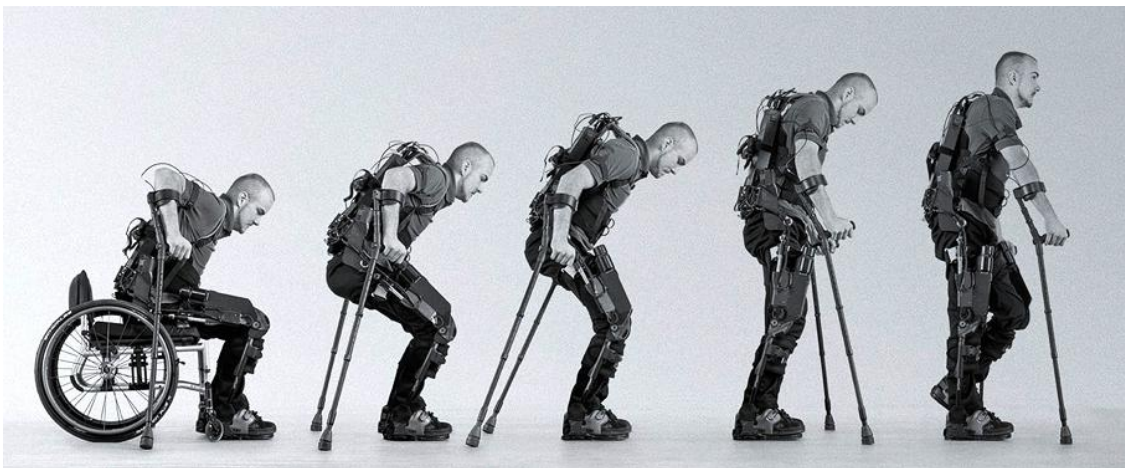
Since the invention of James Watt's centrifugal governor used for speed control of a steam engine in the eighteenth century, many significant works in control theory have been investigated. With the development of frequency-response methods in control field in the 1940s, engineers can start to design linear control systems that satisfy performance requirements. PID controllers were started to be applied to various applications of industrial control systems to compensate for motor speed, chamber pressure, and environment temperature, etc. Since the late 1950s, the focus of controller design has been shifted from single-degree-of-freedom (DOF) systems to multi-degree-of-freedom systems. By introducing the techniques of optimization, modern control theories used nowadays have been gradually developed. As the complexity of industrial plants increases dramatically, calculating capability of modern controllers used for these multi-input/multi-output (MIMO) systems become a major concern while implementing the designed controllers. With the development of digital computers, other modern control theories have been developed to cope with the increased complexity of modern plants and the stringent requirements on accuracy, weight, and cost in military, space, and industrial

applications from the 1960s to the 1980s. These new algorithms have also made controllers more and more intelligent.

The progress of modern control theory, computer science, and electronic devices has made the robotic system, an extremely complex multi-DOF system, a feasible technology to help human beings in various tasks. The applications of robotic systems nowadays include industrial manufacturing devices, assistive devices used for medical rehabilitation, and navigation in different terranes, etc. Among all these different applications, the wearable robotic system has recently gained a lot of attention in the areas of stroke rehabilitation and wearable exoskeleton devices used in the military field.



**Figure 2.2 XOS exoskeleton developed by Raytheon company for military purpose**



**Figure 2.3 Wearable devices used for rehabilitation purpose**

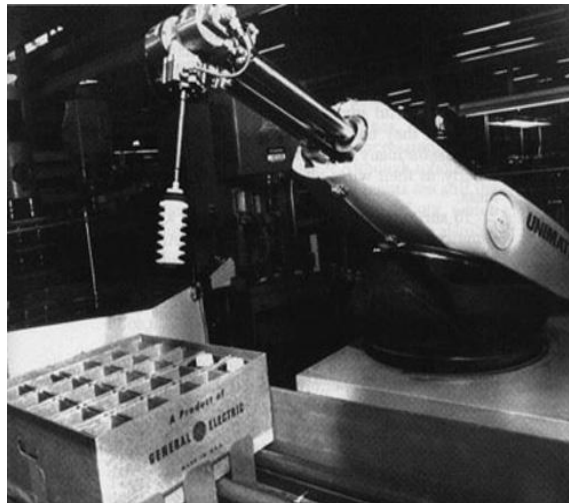
In this chapter, history of the development of multi-axial robotic control system will be discussed. Different synthesis strategies used in multi-axial robotic control will be reviewed as well.

## **2.2 Development of Robot Control**

A robotic system is typically a multi-axial system with high DOF's. Unfortunately, traditional SISO control strategy cannot adequately compensate for the system due to the cross-coupled dynamics [4] among different axes and disturbances caused by unmodeled perturbations in most cases. In order to meet the required performance, the controllers designed for robotic systems are complex MIMO control systems. The development of these control system includes two major trends: integration of separated controllers designed for individual axes and synthesis of MIMO controller with the consideration of movements along all axes.

### **2.2.1 Control of Robotic Systems**

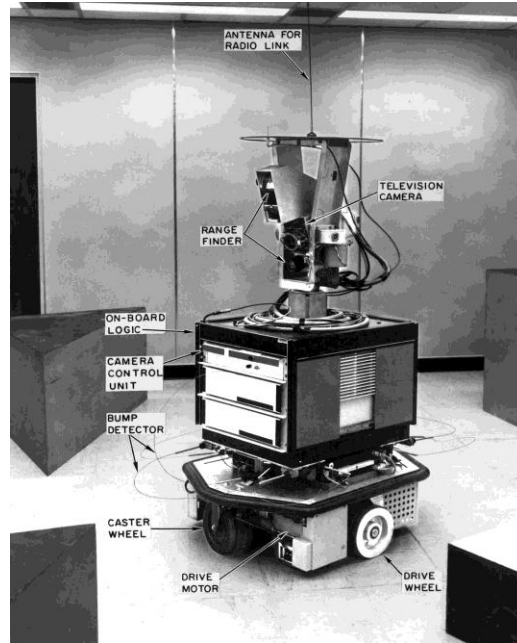
The first modern programmable industrial robot was created in the early 1950s by Devol [5], which was called “Unimate”. This robot was installed in a manufacturing facility of General Motors in 1961, which was used to move workpieces of hot metal.



**Figure 2.4 Robot “Unimate” in General Motors factory [5]**



In addition to the development in industry, academic institutes also made much progress in the creation new robots. In 1958, Rosen et al. [6] at Stanford Research Institute developed a robot named as "Shakey". "Shakey" was a moving robot, which can navigate in a closed space with wheels. It can also navigate across unfamiliar surroundings by detecting its surrounding environment with television "eyes". With these improvements, "Shakey" is a more advanced robot than "Unimate".



**Figure 2.5 “Shakey” robot invented by Charles Rosen group [6]**

These milestones in robotic control history demonstrate that a smart robot can have its own brain to “interact” with the ambient environment. In the 1990s, development of computer industry has produced a significant progress in both hardware and software for various control applications. The rapid development of microprocessors/microcontrollers and embedded systems has made wearable robotic devices feasible. Meanwhile, the development of control theories and software packages also made these “electronic brains” of robotic devices being smart and efficient. In the next decade, wearable robotic devices, interactive home robots, manufacturing robots, and navigation robotic systems are anticipated to dominate the market.

### 2.2.2 Multi-Axial Control

Though the technologies required for the hardware of robotic system has been intensively investigated in the past two decades, the key component of robotic systems, the “brains”, still needs to be improved continuously. In particular, with the growth of human needs and social progress, more and more complex machines and control technology are desired in different area to meet various demands. In other words, mechanisms adopted by robotic devices need to generate flexible motion to accomplish their designated tasks, which means multi-DOF systems are usually necessary. For this reason, highly complex mechanisms [7] have been designed and fabricated to handle various tasks. One fundamental problem in such kinds of complex motion control systems is that the multi-axial motion must be coordinated to fulfill specific performance objectives. Coordinated motion is important in many manufacturing processes where the motions of two or more axes need to move together to follow specific trajectories or achieve a desired performance. Inaccurate synchronization of a multi-axial system can result in poor dimensional accuracy or in unstable systems.

One of the issues of classical and modern control theories is that they only focus on enhancing the performance of tracking along individual axes. These control schemes may not be sufficient to assure superior overall performance unless synchronized motions of all axes are explicitly taken into consideration while designing controllers. Nowadays, it is well known that the coordination performance can be greatly improved by compensating the differences in dynamics of individual actuators. However, early researches did not take the coupling among different axes into consideration and controllers are synthesized independently for individual actuators. Thus, the accuracy of those systems is not satisfactory due to their poor coordination of relevant axes. As a result, different research groups started to propose different coordinating methods and cross-coupled controllers for multi-axial systems to improve synchronization accuracy. In 1984, Koren [8] introduced a cross-coupled controller, which took an "equal-status" viewpoint to address contouring performance, in dual-axis feed drive systems. In addition, the rapid development of the hardware of microprocessors and embedded controllers has also made a real-time control of multi-axial motion systems feasible in various industrial

applications. A generalized synchronization controller was proposed by Xiao *et al.* [9] for multi-axial motion system by integrating cross-coupling technology into optimal control architecture. Jeong and You [10] adopted a cascaded structure to control both acceleration and position. With this structure, the robustness of speed of each axis against disturbances and synchronization errors are both guaranteed. Adaptive controllers are also implemented in various industrial applications. Sun *et al.* adopted adaptive control method to compensate motion control of multi-axial motion systems [11]. Their results reported that both position tracking and synchronization errors can be reduced to zero with the adaptive control scheme. Recently, a framework of adaptive control laws using back-stepping procedure was proposed by Cheng *et al.* [12] to control a multi-axial motion system. With their proposed linear quadratic (LQ) and adaptive controllers, both tracking and synchronization performance can be achieved. However, the drawback of adaptive control method is that the robustness of transient response cannot be guaranteed. To compensate the drawback and provide the corresponding robustness, adaptive robust control scheme is investigated as well [13].

### **2.2.3 Wearable Devices Control**

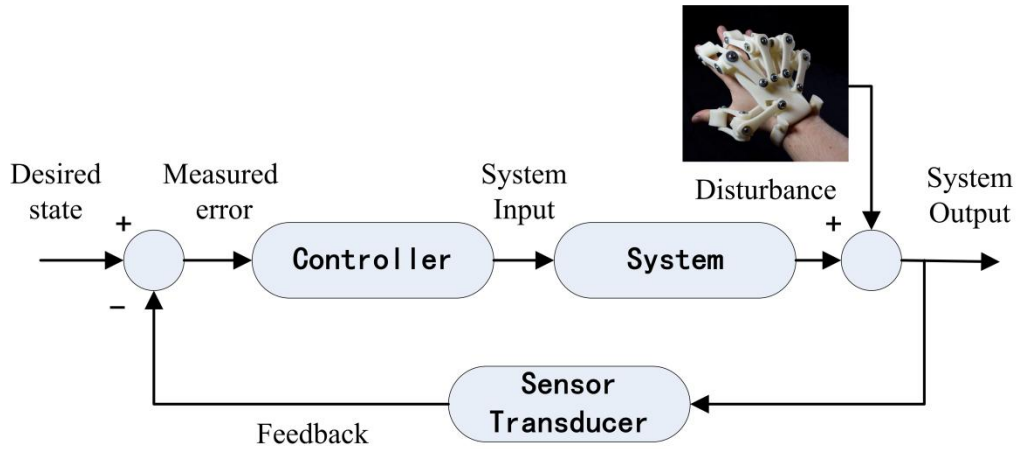
In the past few years, wearable device has become an important application due to the consistent growth of mobile technologies. These devices are usually clothing and accessories incorporating computer and advanced electronic technologies [14]. The available functions of current wearable devices include tracking movement, recording heart rates, and monitoring of clinical and behavioral parameters. Through the history and development of wearable devices, researchers have attempted to enhance the functionality of our daily clothing, or to create other wearable devices with these gadgets. In addition to passively measuring human motion and behavior, many other kinds of wearable equipment, which can actively assist human motion, based on different demands have been developed in the past few decades. One of the most important applications is the wearable exoskeleton used for medical rehabilitation.

In this research, we will focus on the control of a wearable robotic device with multiple string actuators to assist human arm motion. In order to adapt different

trajectories of human movements in our proposed device, it is essential to understand the relationship between habits of human arm movements and characteristics of such a multi-axial mechanism. A lot of research groups have investigated this relationship using biomechanics approach. Biomechanics is built on the basic knowledge of body physics, chemistry, mathematic modeling, physiology, and anatomy, which is an emerging discipline blending aspects of psychology, motor learning, and exercise physiology [15]. This approach adopted by this study will combine mechanical design and motor skill for controller synthesis of the robotic platform developed by West Virginia University.

In the past two decades, design and fabrication of assistive wearable devices used for different purposes have been investigated and studied intensively by different groups. Robot actuation, including electromechanical, pneumatic, hydraulic, and combinatory mechanism [16] plays an important role in performance and functionality of wearable devices. There are many different transmission mechanisms such as cable actuators, gear drives, linked mechanisms and gear drivers, direct drivers, tendon drivers, slip clutches, pneumatic drivers, hydraulic actuators, and their combinations have been adopted in various applications. The selection of different actuators depends on the purpose of the devices. For example, in order to meet high-strength requirement when using a wearable device in a military mission, a pneumatic or hydraulic actuator is required to generate powerful force. On the contrast, a cable actuator can be used in medical rehabilitation process since lighter weight can be much more important than greater provided strength.

When it comes to control, all of these transmission mechanisms can be analyzed by a similar model. As a simple feedback control block diagram shown in Figure 2.6, the measurement from the transducer sensor is the only factor that affects the feedback control route assuming there is no disturbance in the system. In this case, it is relatively easy to compensate the system as a linear one. However, the disturbance of wearable devices caused by user is unavoidable and cannot be ignored in practice while synthesizing controller of a wearable device.



**Figure 2.6 Block diagram of feedback control system**

With the consideration of external disturbance, not only synchronization between individual actuators needs to be carefully compensated, but the coordination between the wearable device and human body also has to be considered. In other words, the cross-coupled relationship between synchronization and individual tracking performance need to be integrated into the controller. Some efficient strategies that handle the synchronization problem will be discussed in the next section.

### 2.3 Synthesis Strategies of Multi-Axial Controller

When dealing with motion tracking of a robotic system, it usually implies the control of a mechanism with at least two degree-of-freedom. The system performance depends on the control of separate motion actuators as well as the control of the relative motion between individual actuators. In order to generate a desired control performance, the cooperation among adopted actuators needs to be carefully planned.

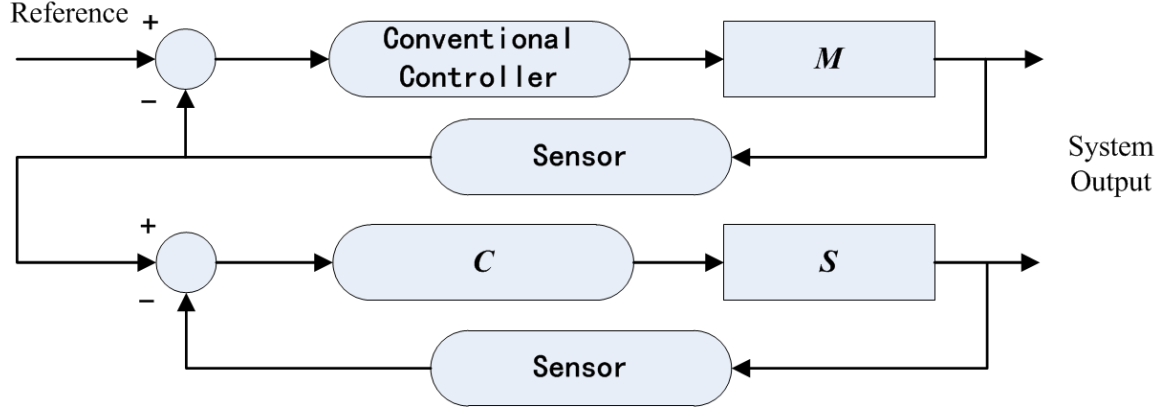
The general meaning of synchronization in a multi-axial system is the concurrence of motions with respect to time. Synchronization, in its most current use in physics, refers to systems of coupled identical systems, where the coupling can be intrinsic or due to a common forcing. The interest in motion synchronization of multi-axial systems has grown rapidly in manufacturing industry since the 1980s. A typical traditional machine design with mechanisms, such as gear trains, has a number of disadvantages: extended

part changeover time, inaccuracies due to wear, rudimentary motion profiles, and the reliability of mechanical transmissions at different machine speeds. Instead of improving mechanical components, many industrial applications of process control have been attracted by possible replacement of the traditional machine with multiple well-organized and controlled actuators. The actuators can be precisely compensated for tracking and synchronization by using some sort of “software” mechanism. With carefully designed controllers, these software mechanisms can effectively reduce errors of both tracking and synchronization to the minimum. The developed algorithms and control methods used for the multi-axial systems can also be adopted by robotic devices. Many research groups have investigated controller synthesis for different kinds of multi-axial systems and robots [17-20]. Among these different control algorithms, a common goal for synchronization is to eliminate matching errors while maintaining excellent tracking performance.

### **2.3.1 Master-Slave Synthesis**

Among all available approaches of controller synthesis, master-slave synchronization is a widely used technique. For a master-slave synchronization scheme for general classes of nonlinear systems, such as Lur’e system [21], it often assumes full state feedback by using the difference between full state vectors as an error signal to synchronize movement between slave and master systems [22]. However, it is not always feasible to obtain all state variables in a practical application. Thus, the use of one single transmission signal is more feasible to synchronize master and slave system than the use of full state feedback [23].

To consider the problem of master-slave synchronization of two robots, it is always assumed that the system can be fully actuated. It is also assumed that both robots have the same scaled dimension with the same type of structure, although the physical parameters like masses, inertia, and lengths of the links can be different. To control such a mechanism, only joint positions of the robots are measured. The objective is that the joint positions of the slave robot need to follow the measured joint positions of the master robot. The master robot is controlled independently from the movement of the slave robot.



**Figure 2.7 Block diagram of feedback master-slave synthesis**

Figure 2.7 illustrates the block diagram of a master-slave control scheme. For the master robot, any conventional controller can be used for tracking the desired trajectory. For the slave robot, a model-based observer is typically included in the synchronization scheme. Therefore, a mathematical model of the slave robot is needed. According to Suykens [24], the systems included in synchronization can be represented as following

$$\begin{aligned}
 M : & \begin{cases} \dot{x} = Ax + B\sigma(Cx) \\ p = Hx \end{cases} \\
 S : & \begin{cases} \dot{z} = Az + B\sigma(Cz) + Du \\ q = Hz \end{cases} \\
 C : & \begin{cases} \dot{\rho} = E\rho + G(p - q) \\ u = M\rho + N(p - q) \end{cases}
 \end{aligned} \tag{2.1}$$

where  $M$  is the master system,  $S$  is the slave system, and  $C$  is the controller. The master system is a Lur'e system with state vector  $x \in R^n$  and matrices  $A \in R^{n \times n}$ ,  $B \in R^{n \times n_h}$ , and  $C \in R^{n_h \times n}$ . A Lur'e system is a linear dynamical system with feedback interconnected to a static nonlinearity  $\sigma(\cdot)$  that satisfies a sector condition. Assuming  $\sigma(\cdot): R^{n_h} \rightarrow R^{n_h}$  is a diagonal nonlinearity with  $\sigma_i(\cdot)$  belonging to sector  $[0, k]$  for  $i = 1, \dots, n_h$ . The output vector of the master system is  $p \in R^l$ , with  $l \leq n$ . The slave consists of an identical Lur's system

with state its vector  $z \in R^n$ , which is controlled by means of the control vector  $u \in R^m$  through the matrix  $D \in R^{n \times m}$ . The signal  $u$  is the output of a linear dynamic output feedback controller. The input of this controller is the difference between the outputs of the master and the slave system  $p$  and  $q$ . The linear dynamic controller has the state vector  $\rho \in R^{n_c}$  and consists of the matrices  $E \in R^{n_c \times n_c}$ ,  $G \in R^{n_c \times l}$ ,  $M \in R^{m \times n_c}$ , and  $N \in R^{m \times l}$ . By defining the error signal  $e = x - z$ , the error dynamics can be expressed as

$$\begin{bmatrix} \dot{e} \\ \dot{\rho} \end{bmatrix} = \begin{bmatrix} A - DNH & -DM \\ GH & E \end{bmatrix} \begin{bmatrix} e \\ \rho \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \eta(Ce; z) \quad (2.2)$$

with the corresponding nonlinearity  $\eta(Ce; z) = \sigma(Ce + Cz) - \sigma(Cz)$ . Then, the dynamic output feedback controller  $C$  can be designed by solving nonlinear optimization problem [25]. This scheme enables one to use fewer measurement signals and fewer control signals to synchronize identical master and slave systems.

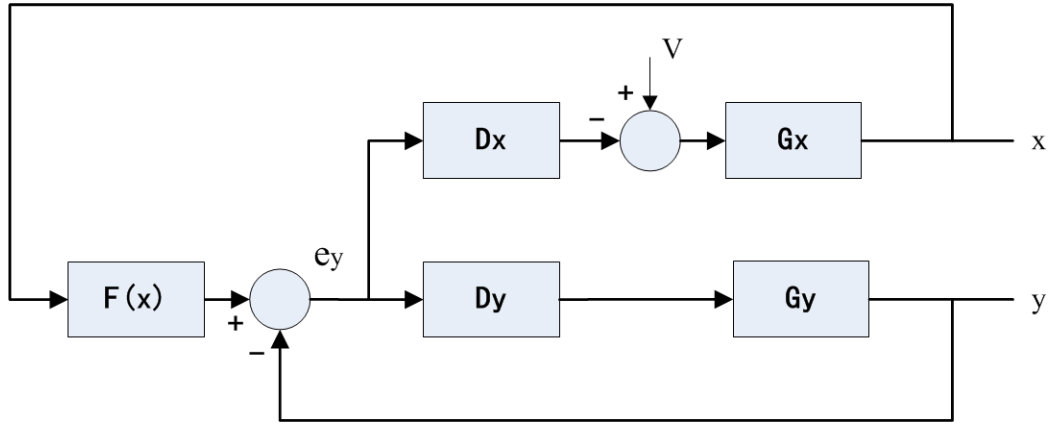
However, in practice, many actuators have to work as a unity for specific purpose and the drawback of this master-slave synthesis strategy is that slave system highly relies on the master system. In this case, the stability of the master system is more critical than the synchronization in between these two systems.

### 2.3.2 Cross-Coupling Synthesis

In addition to the master-slave control strategy, the cross-coupling method also has been widely adopted in various fields. The difference between these two approaches is that the error in either system, while adopting cross-coupling approach, affects the control loops of both systems. The idea of the cross-coupling method is to compensate the errors caused by other systems. For instance, most conventional controllers in a dual-axial system consist of two individual control loops, and the motion coordination is achieved by adjusting the reference value of separate control loops. However, one loop does not receive information of the other loop. The error in each loop is treated as its primary error. With this configuration, the disturbance that causes an error in one control loop is corrected only by the controller in its own loop. The disturbance caused by



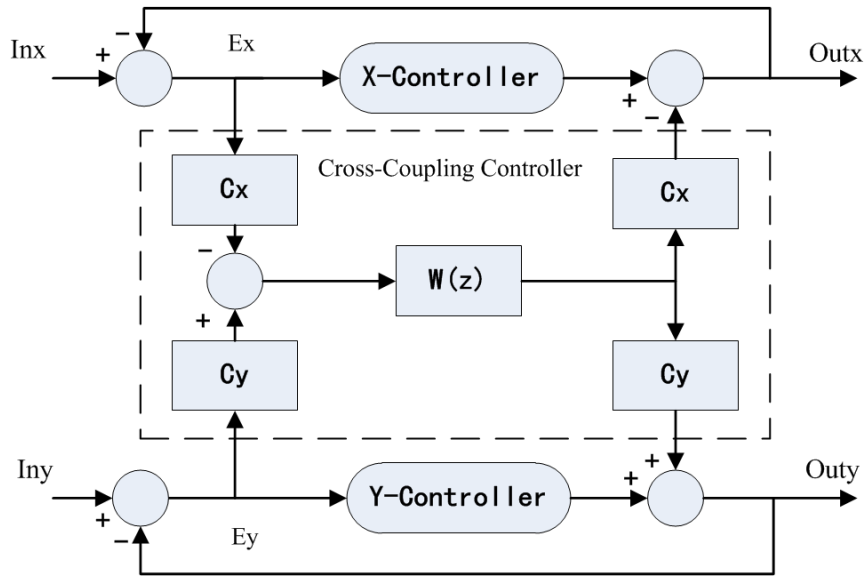
unsynchronized movements of individual axes is not compensated accordingly. This kind of disturbance might still be treated by the other loop independently. Consequently, a cumulative error can result in a large error in the resultant outputs. The major concern of such a phenomenon is that a disturbance caused by improper synchronization needs to be compensated by both loops. Based on past literatures [26], this kind of disturbance is caused by the cross-coupled relationship among individual axes. With the consideration of cross-coupled dynamics, different research groups have investigated various methods of compensating both tracking and synchronization errors in various mechanical systems. The main idea of cross-coupling dynamics is based on calculating the actual contour error, compensating it by a controller, and feeding the result back to individual loops. The first cross-coupling method was proposed by Sarachik and Ragazzini [27]. Their block diagram is shown in Figure 2.8.



**Figure 2.8 Block diagram of nonsymmetrical cross-coupled system**

Such a control system is a master-slave-like structure. The goal of this controller is to ensure that output  $y$  follows the movement of  $x$ . The desired trajectory of  $y$  is generated by a path planner  $F(x)$  with the input of  $x$ . The  $y$ -axis is controlled by a closed loop and the  $x$ -axis is controlled by an open loop whose input is the  $y$ -axis error  $e_y$ . The blocks  $D_x$  and  $D_y$  compensate for desired trajectory and  $e_y$ , respectively. The error in  $y$ -axis affects both  $x$  and  $y$  control loops. However, the error signal of the  $x$ -axis is not compensated. While this unsymmetrical cross-coupled control scheme may be suitable for some open-

loop contour operations, a symmetrical system with equal loop gain in both axes is typically desired in most practical applications. In 1980, a cross-coupled controller for dual-axis feed-drive systems with a symmetrical structure was introduced by Koren and Ben-Uri [28] for motion synchronization. In this study, though it was experimentally shown that a cross-coupled biaxial system can have significantly less contour error than its conventional counterpart, the cost of extra hardware components made it impractical to economically improve the system. Thus, instead of compensating the cross-coupled dynamics using physical components, controllers that compensate the coupled relationship using specific control algorithms have become popular in the past two decades. The controller with the consideration of cross-coupled dynamics serves as an equalizer that compares two types of inputs: a reference input and the feedback signal proportional to the actual output. A block diagram for this cross-coupled feedback control system is shown in Figure 2.9.



**Figure 2.9 Block diagram of the cross-coupled system**

In this system, in order to achieve zero tracking error at the steady-state for a desired reference along individual axes, the controllers must integrate the tracking errors. However, the operation of a cross-coupled system is based upon providing corrections which are proportional not only to the individual axial errors, but to the contour error as

well. This can be accomplished by combining the individual axial errors with a weighted contour error. In addition, the most important factor in the performance of the biaxial control system is the contour error instead of individual axial errors, which is also defined as the “distance-difference” between the desired and actual paths. For the cross-coupled system, the steady-state error and its time response depend on the value from a weighting function  $W$ . According to Koren [28], the steady-state error can be reduced with a carefully selected weighting function  $W$ . The well-designed function can also lead to almost negligible deteriorations in each axis. And the difference between axes is another factor that affects the contour accuracy, which means that the gains and time constants of each feedback control loop are not identical.

With the consideration of coupling dynamics between two different axes, many approaches have been investigated to compensate for both tracking and synchronization errors in different mechanical systems [29], [30]. Kulkarni and Srinivasan have also developed a cross-coupled compensator by introducing an optimal control scheme in 1989 [31]. They have also added an adaptive feedforward scheme to the cross-coupled compensator to improve transient response and disturbance rejection [32]. However, a limitation of this approach is that this scheme only applies to linear systems, which means that the system must have linear relationships among different axes. Srinivasan and Fosdick have proposed a multivariable approach to deal with this nonlinear motion coordination problem. Their method makes an assumption that the desired contour is a combination of many piecewise linear segments. Thus, the resulting controller can be similar to gain scheduling. After that, Koren and Lo have introduced a cross-coupling controller with variable gains. However, the system stability of a cross-coupling controller with time-varying gains has not been examined. In 1992, Chiu and Tomizuka [33] introduced a synchronizing control algorithm for relative degree one systems that took into account of the nonlinear coordination objectives in the synthesis of the control algorithm. They used 3-D curves for a more general coordinated system. The basic idea of this approach is that cross-coupling control scheme uses feedback information from more than one control loop to compensate for a composite error, which integrates individual errors from all control loops, rather than individual loop errors.

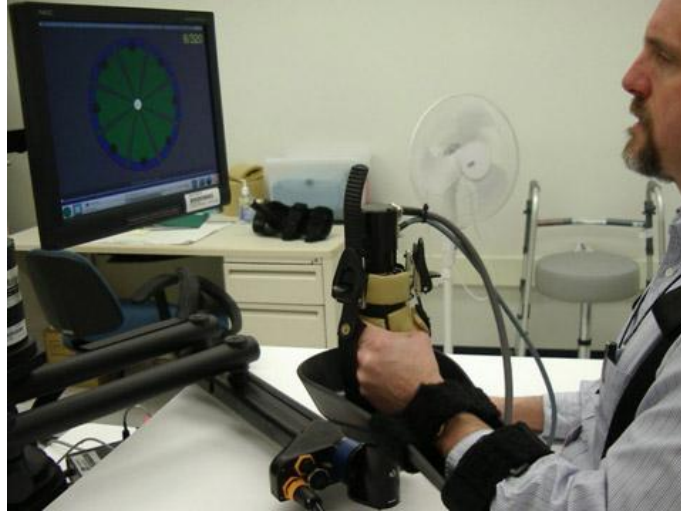
## **2.4 Applications of Wearable Robotic Devices**

Wearable technologies can be found and used in various applications. For example, for a worker working in mines need to have a helmet, protective gears, tools, camera, and communication gadgets. Some of these gadgets have been recently improved by adopting wearable technologies for working efficiency, environment monitoring, and safety. There is no standard definition of wearable technologies. They are usually identified as the combination of electronic and computer technologies incorporated into items of clothing and accessories. The selection of applicable technologies depends on the purposes of specific applications. With the-state-of-the-arts technology, the computational capability of current wearable devices is as powerful as mobile phones or laptop computers. The rapid development of microprocessors and embedded controllers has made real-time control of portable devices feasible in various industrial applications. In this section, the main application of wearable robotic devices will be discussed.

### **2.4.1 Medical Applications – Robotic Devices**

Robotic technology has been wildly adopted in the medical field in the past few decades. There are several different types of wearable devices which have various sizes and structures used for numerous purposes.

First of all, the end-effector wearable robots that can be used for rehabilitation and surgery. In clinical practice, physical disability caused by stroke is a frequent and severe problem. Traditional treatment for this kind of disabilities requires therapists to assist patients to perform certain types of tasks, which can be time consuming and expensive. Thus, researches in robotic rehabilitation system have been motivated by the potential of robots performing repetitive tasks without supervising therapist. The flexibilities of these robotic devices can also be used in different therapy modes. For instance, MIT-MANUS was the first robot designed by Hogan and Krebs [34] dedicated to evaluating the impact of using robot-aided therapy in upper-limb rehabilitation. This robotic device can provide patients therapy by moving their affected arms on a horizontal plane as shown in Figure 2.10.

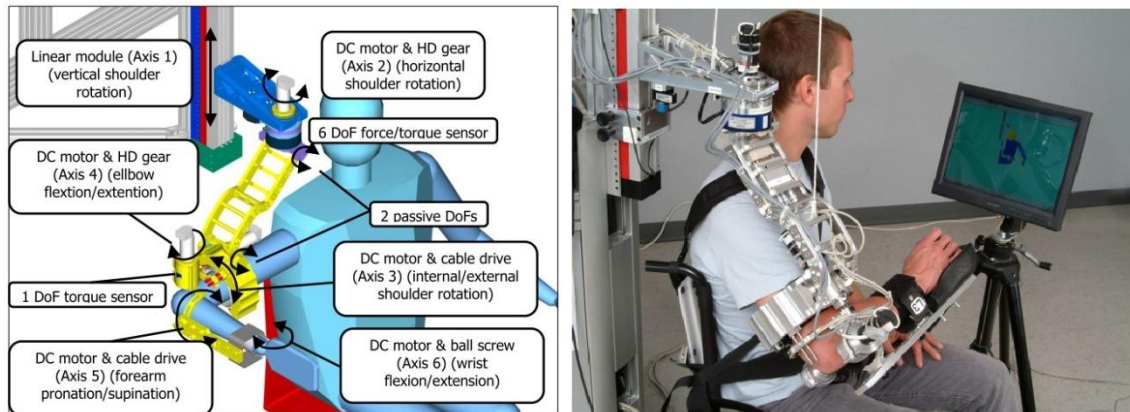


**Figure 2.10 Therapy is being conducted with a commercial version of MIT-MANUS [34]**

It benefits from back-derivable actuators with low friction and inertia which not only allow the robot to provide a smooth top-table movement but also let the robot to use force feedback approach to transfer the force to the end-effector handle grasped by the patient. The required force varies according to the will of the therapist. In order to increase the positive impact of the therapy, some applications such as 2D video games for tracing points or moving targets are integrated to this robot. Significant improvements are reported in both motor recovery gains following stroke and chronic stroke recovery phase months after stroke. One of the bold outcomes of the previously mentioned research discredits the commonly accepted idea that the time for effective rehabilitation therapy is limited. The results show that brain retains the functional gains acquired during robotic therapy both after 3 months and at the end of 3 years following initial treatment [35]. The results not only suggest that the neuro-recovery process continued far beyond the commonly accepted 3–6 months post-stroke interval but also demonstrate that the lesion location plays a crucial role in neuro-recovery. As an end-effector robot, MIT-MANUS is limited for performing various therapies. The latest version of this robot employs new extensions, which increases the capability to perform tasks for forearm and wrist assistance. However, more degrees of freedom are needed to overcome gravity and perform more complicated tasks. The Bi-Manu-Track system [36] is another example of

end-effector robots, which is able to be set on passive and active-resistive modes for both limbs and passive mode of the paretic limb to mirror the movement of the non-paretic limb. The major movements provided by this robot are bilateral rehabilitation of the wrist flexion/extension and forearm pronation/supination. Such therapies are beneficial for reducing spasticity and improving motor control in chronic stroke patients.

End-effector systems have achieved great accomplishments in rehabilitation field during the past few decades. However, their small range of motion has always been a limiting factor for Activities of Daily Activities (ADL) therapies. Therefore, many researchers have investigated different approaches to providing a larger work space for performing ADL's. Among all different solutions, the exoskeleton robot is the best alternative. For instance, ARMin [37] is one of the first active robots designed specifically for arm therapy applications and ADL training. Arms rehabilitation therapy includes shoulder abduction/adduction, elbow flexion/extension, and forearm supination/pronation. ARMin has a semi-exoskeleton structure with six degrees of freedom as shown in Figure 2.11.



**Figure 2.11 ARMin mechanical structure with six degree of freedom [37]**

To increase the efficacy of therapy, audio and visual applications can be integrated to the robotic system by providing game-like exercises. This kind of devices could potentially reduce the cost and inconvenience to patients by allowing rehabilitation

devices being brought home. Exoskeleton robotic devices can collect comprehensive and reproducible data and communicate with therapist remotely.

Other than rehabilitation, robotic devices used for surgical operation have achieved great accomplishments in the past few decades. The goal of using robots in surgeries is to provide: 1) improved diagnostic abilities, 2) a less invasive and more comfortable experience for patients, and 3) the ability to do smaller and more precise interventions. Robots are currently used not just for prostate surgery, but for hysterectomies, the removal of fibroids, joint replacements, open-heart surgeries, and kidney surgeries. Since the physician can see images of the patient and control the robot through a computer, they do not need to be physically in the operation room with the patient. This means that a specialist can remotely operate the surgery on a patient, which can provide a better work environment for the physician by reducing strain and fatigue. Surgeries that last for hours can cause surgeons to experience hand fatigue and tremors, whereas robots can be much steadier and smoother. For example, the da Vinci Surgical System [38] manufactured by Intuitive Surgical is a surgery-assistant system that is intended to assist in the control of several endoscopic instruments, including rigid endoscopes, blunt and sharp dissectors, scissors, scalpels, and forceps. The system has been approved by the FDA to manipulate tissue by grasping, cutting, dissecting and suturing. As shown in Figure 2.12, the da Vinci system consists of three major components: the vision system, the patient-side cart, and the surgeon console.



**Figure 2.12 The da Vinci Surgical System by Intuitive Surgical [38]**

The vision system includes the endoscope, the cameras, and other equipment to produce a 3D image of the operating field. And the patient-side cart has three robotic arms and an optional fourth arm. One arm holds the endoscope while the other arms hold interchangeable surgical instruments. The da Vinci system adopts EndoWrist, a flexible surgical instruments that surgeons can bend and twist like human wrists. This tool can mimic the movements of the human hand and wrist. Finally, the Surgeon Console is a surgeon that sits at a console several feet away from the operating table and manipulates the robot's surgical instruments. The robot has three hands attached to a free-standing cart. One arm holds a camera that has been passed into the patient's body through small openings. The surgeon operates the other two hands by inserting fingers into rings. The surgeon uses hand movements and foot pedals to control the camera, adjust focus, and reposition the robotic arms. The da Vinci has a three-dimensional lens system, which magnifies the surgical field up to 15 times. Another surgeon would need to stay beside the patient to adjust the camera and instruments if needed.

#### **2.4.2 Medical Applications – Other Wearable Devices**

In addition to robotic systems, wearable devices are also widely used in health and medicine monitoring sector. So far, there are many direct or indirect applications that require various software packages and sensors to assist detecting individuals' health conditions. For instance, an intelligent asthma management system called an Automated Device for Asthma Monitoring and Management (ADAMM) that is produced by Health Care allows a patient to acquire his/her asthma status in real time. This device and its corresponding app on mobile devices can alert patients when they are about to trigger a severe asthma situation. The software can also be used to log journaling, reporting treatment plans, and recording information of treatment and symptoms. Such an integrated system can also provide medical staff patients' detailed health information and medic parameters. With the assistance of information technology, this kind of gadgets enables medical centers to immediately react to urgent situations. Recently, many companies, such as Microsoft, Google, and Apple, have started to develop smart wrist bands and watches that can monitor personal activities with medical grade sensors. For



example, an Embrace monitor can be equipped on a kid's wrist while he/she is sleeping. If something unusual happens, like a seizure, an alert will be given to a mobile device which informs parents for urgent care.

The applications of wearable device do not only include physical assistance and healthy monitoring, these intelligent wearable devices can also warn the user for certain undesirable behaviors. For example, Smart Stop produced by Chrono Therapeutics is a smart device that aims to help people to quit smoking. Motion sensors are embedded in the Smart Stop device that can sense changes in user's body and analyze the changes with motion algorithms to determine whether the user is craving for a cigarette and nicotine. Once the algorithms identify a positive craving, the device then delivers medication to the user so that craving can be curtailed. Just like most smart wearable devices, this Smart Stop also has a companion app that gives the user information about quitting and guiding the addicted users to stop their harmful habits. Similarly, ulcer is something that affects everyone just as people get older. Scientists believe that inactivity and constant sitting can also be the major reasons that cause ulcers. Thus, a Leaf Healthcare Sensor is designed to alert a person when it is time to turn and make some body movements. The Leaf Healthcare Sensor includes a tri-axial accelerometer that can monitor the user's body position and assist the user to move in proper ways. This device also works with optimizing tissue pressurization during turning and moving around. Once the monitor is equipped by a patient, caregivers or physicians can receive an alert if the patient is improperly moved or if a proper movement is required.



**Figure 2.13 “Smart Stop” and “Ulcer” used for bad-habits correction**

Many wearable devices have also been investigated to take care of various human diseases, such as diabetes and heart-disease. For instance, Google has started to develop smart contact lenses. The technology is engineered to take the tears in a person's eye and measure the glucose levels that are present, which can be a major advance for diabetes patients. For people who wear glasses, the lens is also engineered to 'restore the eye's natural autofocus'. The wearable ECG monitor, QardioCore, is an easy-to-use monitor that straps around the chest. The sensors in it are able to monitor heart health and health status. This wireless monitoring device does not need any kind of sticky patches or gels. The acquired information can be displayed on an app on mobile devices and uploaded to a cloud-based server, which allows for a physicians or therapists to monitor patients' status remotely.



**Figure 2.14 Google leans and “QardioCore” used for diseases supervisal**

Overall, the applications of wearable devices in medical field are diverse and very purpose-oriented. These applications become more and more popular and available on most recent mobile devices. It is also clear that variable demands of biosensors and bio-modeling used in wearable devices have accelerated the recent development in the related research areas. Some of these technologies include remote sensing, biosensors, communication, computer vision, long life batteries, and modeling integrated into mobile operating systems (iOS, Android, and Windows Phone).

### 2.4.3 Military Application

Wearable technologies are also now widely adopted in various military applications related to security and protection. Military applications of wearable technologies can be traced back many centuries ago. Use of fighting devices such as knuckle style weapons or pointed rings in the fingers to attack the enemy from close vicinity has been recorded in history. For instance, the Indian warrior Maratha king Shivaji (1627-80) is known to have used tiger claws (iron finger-grip with razor claws) to kill Afzal Khan, a commander of the Adil Shahi dynasty [39] in the 17<sup>th</sup> century. In modern times, military troops were always identified with basic hand-held radios carried as a backpack unit supporting secured data and voice communication for many decades. This equipment was popular because of its low power requirements so that troops were not required to carry additional packs of battery. Such equipment, though never called as wearable technologies, were so for practical purposes.

Generally speaking, the objectives of military wearable applications can be classified as:

- 1) Improving aiming capabilities: The technology used in current generation of rifles enables it to track targets, detect wind and weather, and calculate the optimal flight path of the bullet. There is a possibility that soldiers can aim their targets without actually looking at them by hooking these rifles to an augmented reality platform like Google Glass. There are a few products commercially available already today. For instance, the BulletFlight 4.0.0 produced by Runaway Technology adopts a ballistics program that runs on Apple iPod Touch and Apple iPhone platforms. The mobile devices can provide soldiers important parameters such as angle detection from its built-in accelerometer and ballistic plots for multiple firearms, etc. As shown in Figure 2.15, the app of BulletFlight demonstrates a sophisticated program. The operating system and “multi-touch” screen of Apple platform offer an extremely user-friendly interface. A version of BulletFlight has already been in use by the U.S. Military in Iraq and Afghanistan.



**Figure 2.15 “BulletFlight 4.0.0” Mobile ballistics program for I-pod and I-phone**

- 2) Monitoring the physical state of soldiers on the move: The physical condition of a soldier does impact the overall performance in the combat. The physical condition and injuries of a soldier can be identified with tiny biosensors on their body. The information can help the commanders to monitor the condition of soldiers in the combat. It is necessary to include sensors that can monitor various vitals, such as heart rate, breathing, and hydration. Some sensors are required to detect soldiers' injuries. This function can be implemented by a health monitor invented by odic as shown in Figure 2.16, which includes a sensitive ECG front end with a complex algorithm of heart-rate monitoring. It also includes an accelerometer and a respiration sensing circuitry for the derivation of respiration rate, activity level, and body orientation. All of which then generates a life signs indication. A 915 MHz wireless transceiver sends this information to a medic unit or to the Army's local wireless network. The field of monitoring and communication can be expanded to other animals participating in military campaigns like horse, mules and dogs.



**Figure 2.16 Wearable health monitor for the military by odic**

- 3) Providing 360-degree battlefield awareness: Useful inputs about the situation from surrounding environment can always help soldiers to survive on the battlefield. Therefore, It is necessary to keep soldiers to be aware of the changes all the time. This can be done by combining a virtual display of Google Glasses and other real-time inputs received form equipped sensors and other important information in the region. The relevant information includes location, weather condition, topographic details, satellite/radar image, and enemy troop concentrations. With this purpose, the Helmet Mounted Radar Program [40] aims to provide a near-360-degree vision of Moving Target Indicator (MTI) radar sensors. Ideally, the helmet is needed to integrate all sensors and weigh less than two-and-a-half pounds, with less than a pound of sensors mounted on the helmet as shown in Figure 2.17. The integrated system can only emit low power electromagnetic noise in order not to affect the health of soldier wearing the helmet.



**Figure 2.17 Wearable Helmet Mounted Radar Program [40]**

- 4) Providing extra force support: Soldiers on the battlefield usually have to carry necessary weapons and enough life supplies. It can be a burden while moving. In particular, they need free hands to use weapons and other gadgets in combat. To provide additional force and assistance, military exoskeletons started to gain a lot of interest. Currently, military is exploring creating an Iron Man-like specialized suit through the Tactical Assault Light Operator Suit (TALOS) program [41]. The goal of the suit is to provide soldiers with enhanced mobility and protection. With current technology of robotics, existing exoskeleton robotic device can carry weight up to 17 times of what a soldier can typically carry. With an existing suit, XOS 2, as shown in Figure 2.18, a soldier can carry 400 pounds while providing a force to carry a weight of 23.5 pounds.



**Figure 2.18 XOS 2 exoskeleton developed by Raytheon company [41]**

#### **2.4.4 Wearable Virtual Reality Devices and Gaming Applications**

One major application of wearable technology is in the gaming industry. Most Gamers are always looking forward to adopting innovative technology in gaming experience. Wearable gaming products used for virtual reality with big market potential have attracted many hardware producers for different products.

The products range from glasses and gloves to full body platform. For the complete solution, Virtuix Omni [42] is a virtual reality system with a complete body experience approach. It deeply immerses wearers in their favorite games by allowing them to stand up and walk around in the game's world and featuring a VR headset, gun, platform and a special belt.



**Figure 2.19 XOS Virtuix Omni Game System [42]**

Other than a complete solution, different companies are releasing various virtual reality headsets. For example, the cordless holographic lens, HoloLens, produced Microsoft is a smart-glasses headset, which includes a self-contained Windows 10 computer [43]. The holographic lens includes advanced sensors, a high-definition stereoscopic 3D optical head-mounted display, and spatial sound to allow for augmented



reality applications, with a natural user interface that the user interacts with through gaze, voice, and hand gestures. In addition to Microsoft, Oculus, SONY, Zeiss, and HTC all started to release their Virtual Reality glasses in the past few years as well.



**Figure 2.20 Microsoft HoloLens [43]**

A completely new and more immersive approach of wearable virtual reality devices is the Tesla Suit [44]. The full body suit consists of a belt (with a central control unit) and as modular units haptic feedback gloves, and will be fully compatible with existing virtual reality headsets such as the Oculus, Google Glasses and META Space Glasses, as well as gaming consoles (PlayStation, XBOX), PC, and even smartphones. The wireless suit performs a dual role as a motion capture system and a state-of-the-art smart textile, incorporating thousands of nodes to directly stimulate sensation and pleasure in the skin through tiny electric pulses.



**Figure 2.21 Tesla Suit [44]**



## **2.5 Development Tendency**

The role technology plays in our lives is rapidly growing, as we become increasingly dependent on it and less willing to separate ourselves from it. The rise of wearable technology, from fitness trackers and the Apple Watch to Google Glass and Microsoft's HoloLens, has been fueled by technology's tendency to get faster and smaller at the same time.

There are several important characteristics that future wearable devices could be equipped with. First, more smart. Wearable technology tends to be more sophisticated than hand-held technology available in the market today because it can provide sensory and scanning features not typically seen in mobile and laptop devices, such as biofeedback and tracking of physiological function. Secondly, more efficient. Battery life in wearable devices is very scarce, and as a result, alternative forms of energy to power wearable devices are on the rise. Besides, as we can predict, the future wearable devices will become more and more accurate, personalized and functional. If we can be sure of anything, it's that the future means we'll be wearing our technology on the outside, and, perhaps one day, on the inside, too.

## **CHAPTER 3**

### **HARDWARE OF EXPERIMENTAL PLATFORM, MODELING, AND CONTROLLER SYNTHESIS**

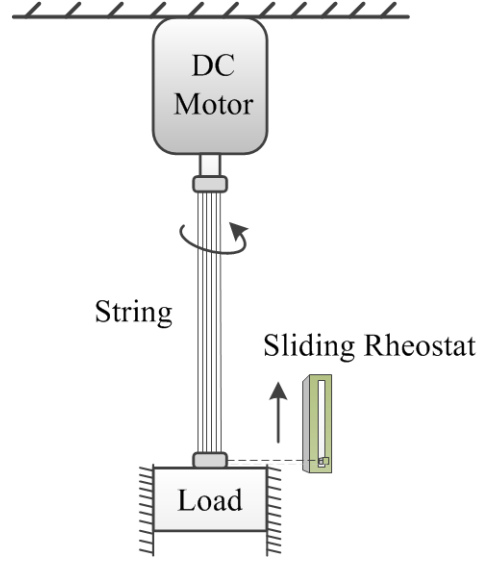
In this study, a mechanism that emulates two joints, or two degrees-of-freedom (DOF), was adopted to demonstrate the control system of an assistive robotic system. The implementation of this experimental setup needs to integrate three parts: mechanical hardware of the experimental system, electrical devices that drive the mechanism, and software that controls the movement of the overall system. This chapter describes the components and individual part and discusses the modeling of the mechanical system.

#### **3.1 Mechanical Hardware and Electrical Devices**

In order to reduce the complexity of the mechanism, the mechanical hardware of the proposed wearable device should be kept as simple as possible. It is also important that the robotic device need to well fit with user's body. Thus, this section starts with the actuation device. Then the biomechanics analyses of human movement will be discussed. The design and setup of experimental platform will be discussed in this section as well.

##### **3.1.1 Twisted-String Actuation**

In this design, the actuation devices adopted are twisted-string actuator. This kind of actuation device has the advantages of compact sizes and predictable linear motions. However, single twisted-string actuator is a one way driver since it can only generate the tensile force along the direction of the DC motor shaft by twisting strings. It cannot generate the dilation force when untwisting. The work principle of single axial twisted string actuator is shown in Figure 3.1.



**Figure 3.1 Single-Axial twisted string actuator**

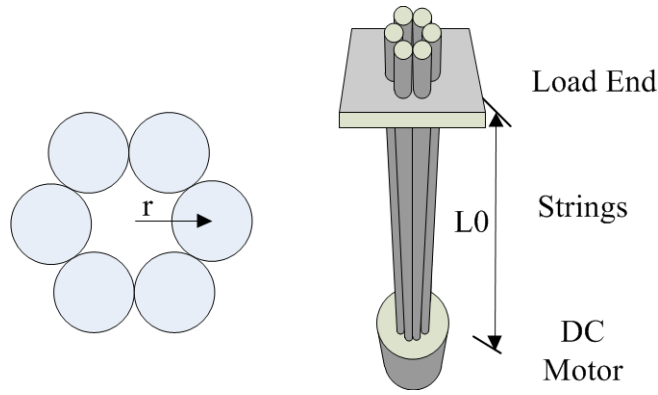
The work principle of twisted-string actuation is straightforward: a multi-strand strings are twisted by the attached motor, which yield the length of the string to be shortened. As shown in the figure, a twisted string actuator consists of a DC motor, a multi-strand string, a load, and a sliding rheostat. One end of the string is attached to the load, and the other end of the string is attached to an adapter connected to the motor. When the driving motor starts to rotate, the angular movement of motor shaft twists the string. As a result, the distance between motor adapter and loads is decreased. If the position of the driving motor is fixed, then the load can only move in a single direction. Therefore, the position control of load can be realized by controlling motor. The sliding rheostat can be used to provide the length information of the twisted string by connecting slide to the load.

Based on Wurt's model [45], the length change of a twisted string can be modeled as:

$$p = \sqrt{L_0^2 \left( 1 + \frac{Fi}{K} \right)^2 - \theta^2 r^2} \quad (3.1)$$

where  $p$  is the position of load end,  $L_0$  is the original length of the string without being twisted,  $Fi$  is the driven force generated by motor,  $K$  is the coefficient of elasticity of

string,  $\theta$  is the angular movement, and  $r$  is the radius of the string. The configuration is demonstrated as Figure 3.2. By analyzing eq. (3.1), the linear displacement of load  $p$  decreases if radius  $r$  increases if other parameters are fixed. If a multi-strand string is adopted in such an actuation device, the configuration of strands can affect linear displacement as well. According to Popov [46], while implementing centered symmetrical configuration, the radius of the synthesized string increases if more strands are included in a string. Therefore, in practical the choice of string configuration has to be balanced between the linear displacement and response time.



**Figure 3.2 Configuration of the twisted string system**

In this research, six-strand string was chosen. The pattern and the centered symmetrical configuration are shown in Figure 3.2. The moving direction force  $Fi$  generated by motor can be derived from eq. (3.1).

$$Fi = \frac{K}{L_0}(L - L_0) = \frac{K}{L_0}(\sqrt{p^2 + \theta^2 r^2} - L_0) \quad (3.2)$$

Every single string in this twisted-string actuation system can be treated as a spring system, which is a function of angular position. And the coefficient of elasticity  $S$  of the spring system consisting of  $n$  strings, according to [46], can be defined as

$$S = \frac{\partial nFi}{\partial p} = nK \left( \frac{1}{L_0} - \frac{1}{\sqrt{p^2 + \theta^2 r^2}} - \frac{p^2}{(p^2 + \theta^2 r^2)^{3/2}} \right) \quad (3.3)$$

According to eqs. (3.1), (3.2) and (3.3), the relationship between linear displacement and number of rotation  $N$  can be derived as following

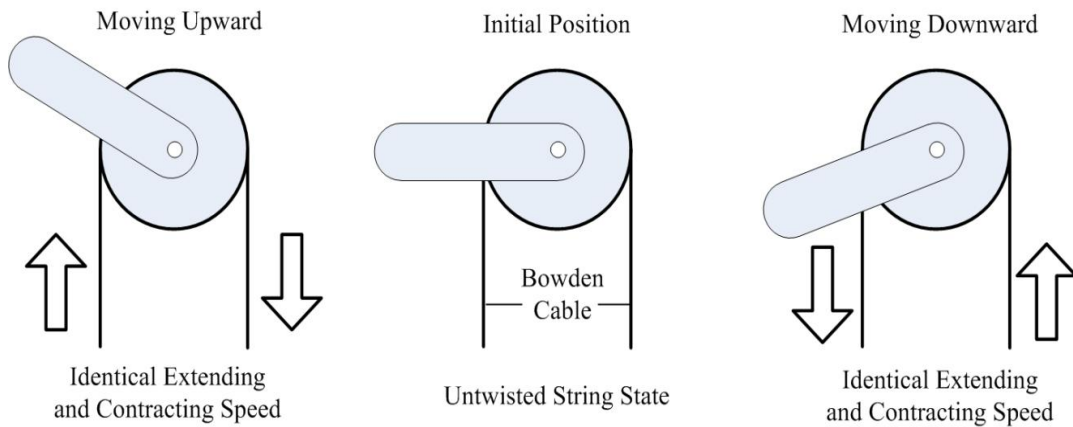
$$\left(\frac{L_0 S^2}{K^2} - 1\right)p^2 + \frac{2SL_0^2}{K}p + L_0^2 = 4\pi^2 r^2 N^2 \quad (3.4)$$

Rearranging and simplifying eq. (3.4), the relationship becomes

$$N = \sqrt{Ap^2 + Bp + C} \quad (3.5)$$

where  $A = (\frac{L_0 S^2}{K^2} - 1)/4\pi^2 r^2$ ,  $B = \frac{2SL_0^2}{K}/4\pi^2 r^2$  and  $C = L_0^2/4\pi^2 r^2$ . Therefore, the rotation number of the driving motor is a parameter of the twisted-string actuation system, which can be used to estimate the linear displacement of the system.

Since a single twisted-string actuator can only provide pulling force, it is necessary to combine two actuators to synthesize a push-pull actuation. With a dual-actuation, the operation of the two twisted-string actuators needs to be realized by changing the lengths of individual actuators simultaneously while one is stretching and the other one contracting. The following figure demonstrates an example of synchronizing two twisted-string actuators to rotate a pulley.

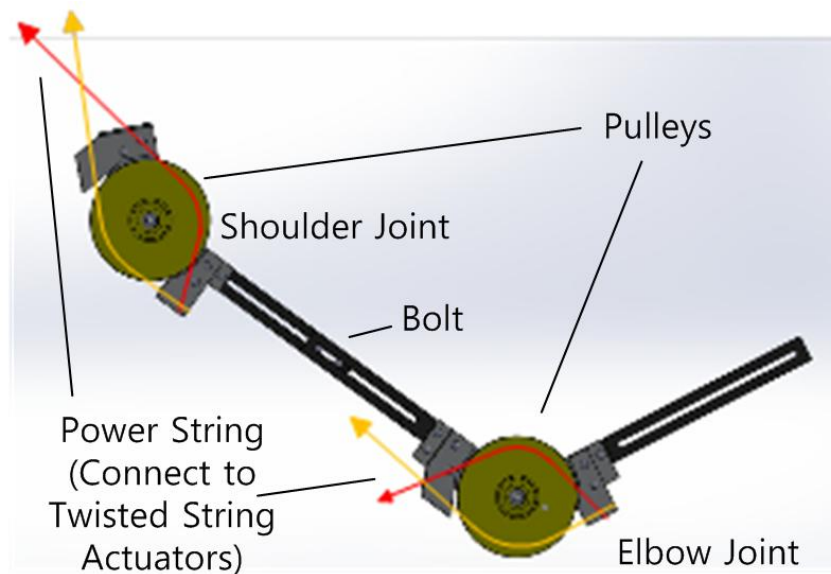


**Figure 3.3 Operations of two twisted-string actuators used to rotate a pulley**

In this configuration, the twisted-strings of both actuators are connected to the pulley using Bowden cables. With the same contracting and extending speed, the pulley can rotate properly without center shift. This configuration is the basic setup of the mechanisms of the joints discussed in the next section.

### 3.1.2 Mechanism of Assistive Robotic Device

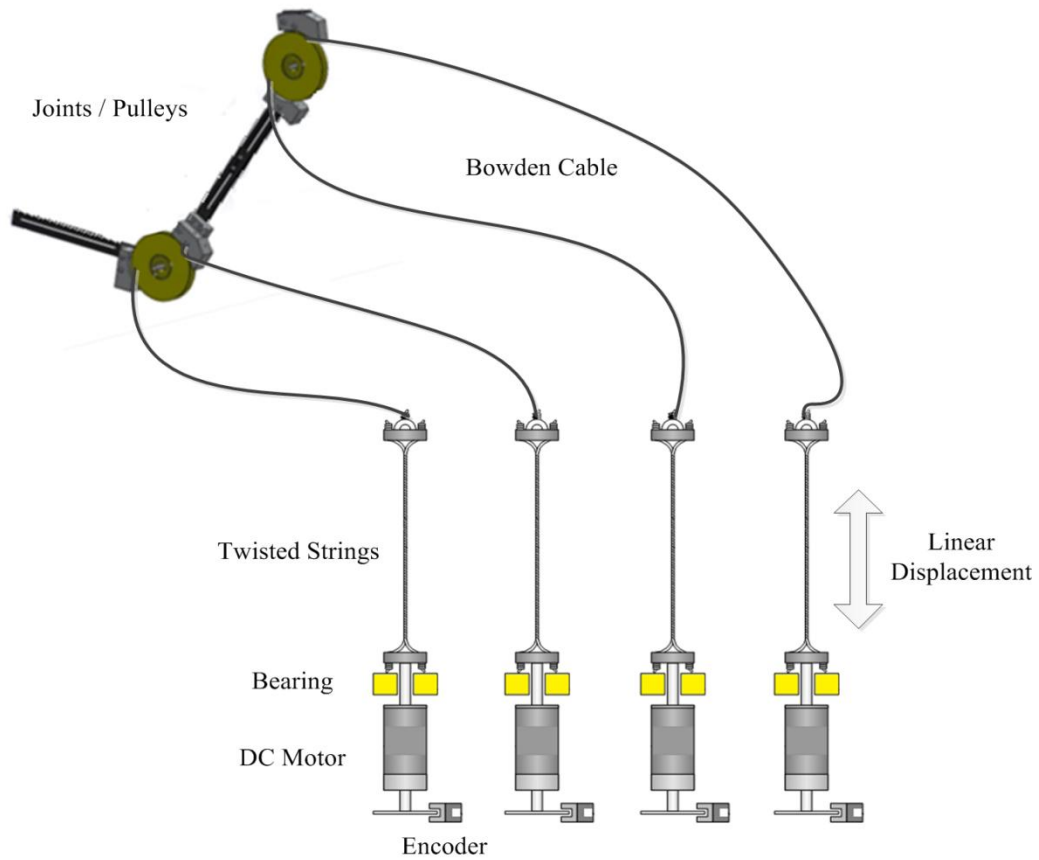
The goal of the designed mechanical structure and force actuators is to provide enough degree of freedom (DOF) for users who wear it to move their upper limbs. In the proposed design, each joint of the robotic mechanism is driven by a pair of twisted-string actuators. It is desired that it needs to have at least 7-DOF to provide a smooth movement. In this preliminary design, the DOF of the prototype robotic arm is 2, which are the two independent robot joints. The configuration is shown as Figure 3.4.



**Figure 3.4 Mechanical structure of robot arm**

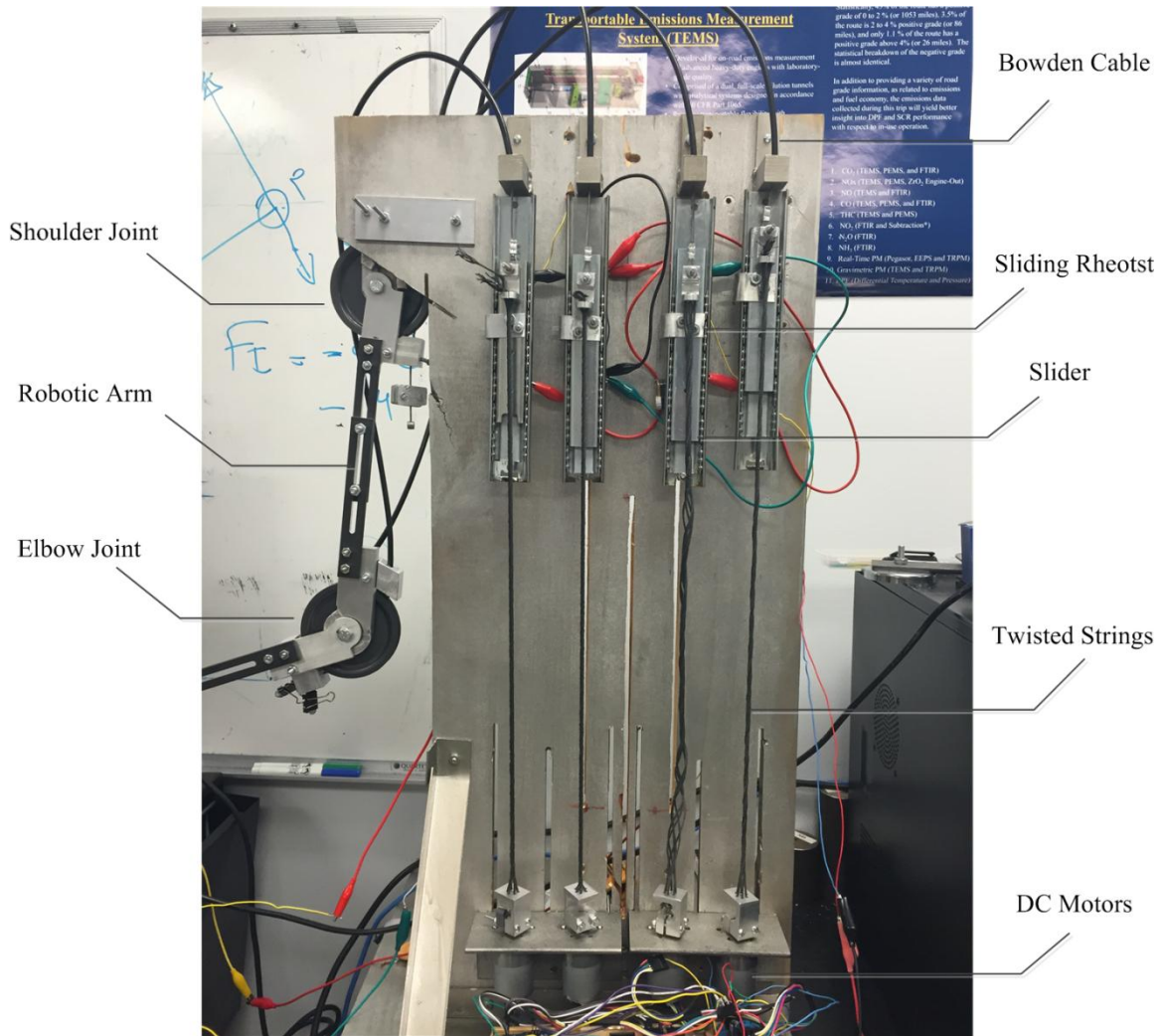
The design aims to be available to all users with different body sizes with minor adjustment. To achieve this, the distance between shoulder joint and elbow joint can be changed by adjusting a bolt between each joint. The limitation is the range of rotation angle. The limitation of the ranges of both joints is required due to a safety reason, which prevents the user to be injured due to the out-of-range angle. The ranges are from 0 to 120 degrees for shoulder joint and from 0 to 135 degrees for elbow joint, respectively. To

demonstrate the moving directions of the robot, the red and yellow lines represent different moving directions driven by individual DC motors.



**Figure 3.5 Overall structures of twisted string actuators**

Figure 3.5 illustrates the overall configuration with both joints connected to driving DC motors. The physical setup of the robotic system is illustrated as Figure 3.6.



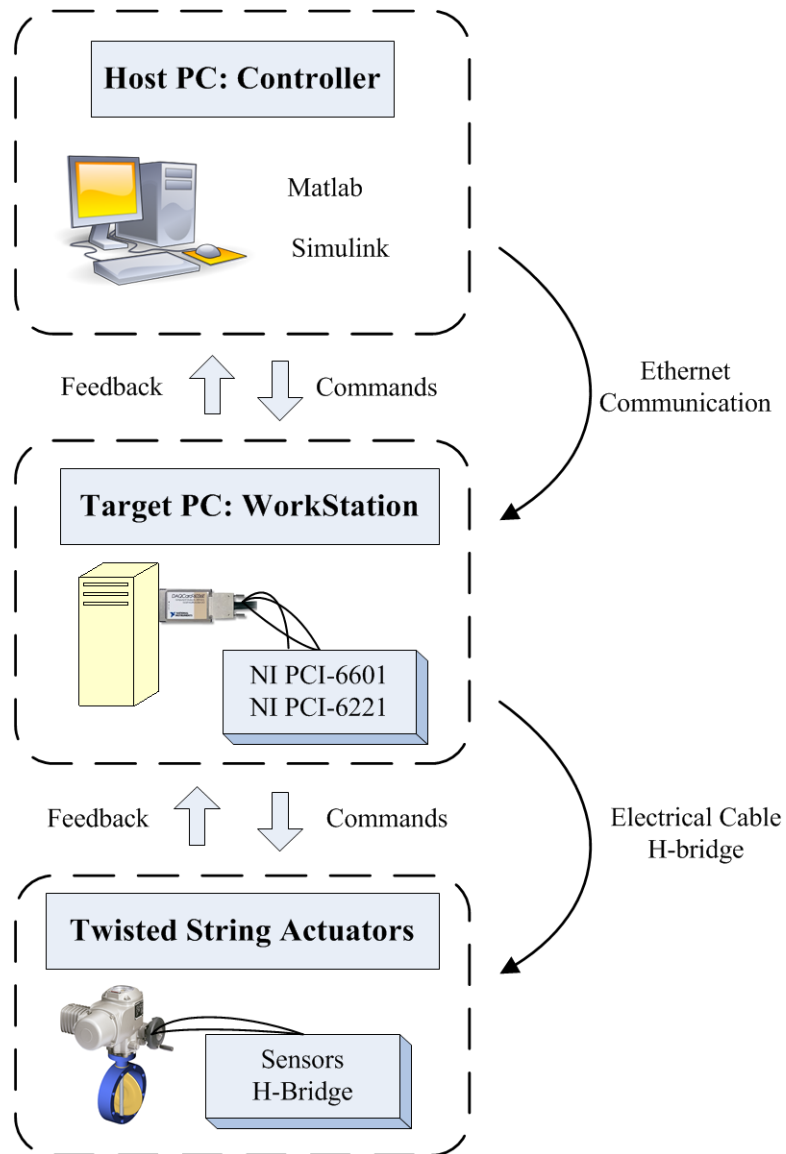
**Figure 3.6 Twisted string actuators experimental platform**

In this study, nylon rope was chosen as the strands in the twisted strings with its high endurance of fraction. As discussed in the previous section, the string is configured as a 6 strands structure. The diameter of each strand is around 1 mm and its initial length is around 500 mm. The two joints are driven by these strings with Shimano Bowden cables. The adopted DC motors are gear motors manufactured by Pololu (12V, 19:1, 64 CPR encoder). The sliding rheostats are ALPS slide potentiometer (10K, 60mm), which were used to measure the linear movements of individual twisted string actuators.



### 3.1.3 Data Acquisition and Electrical Driver

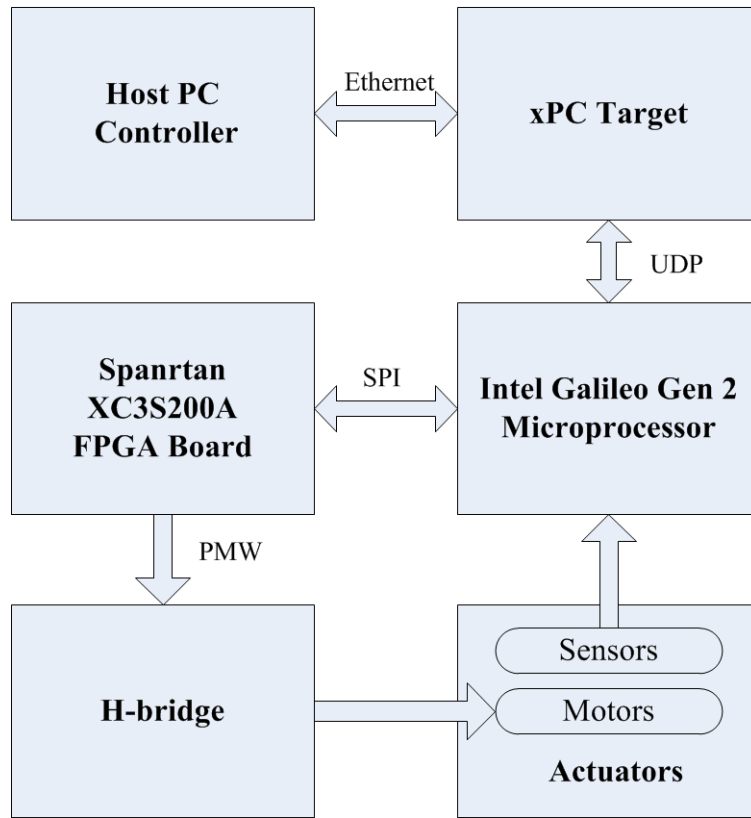
In the data acquisition part, two data acquisition cards were used while setting up the testing platform. Two microprocessor boards and one FPGA boards were used in final product. The schematic diagram of the data acquisition used while setting up the platform is illustrated as the following figure.



**Figure 3.7 Schematic diagram of data acquisition of the testing platform**

In this setup, two PC were used: one is the host PC with MATLAB installed and the other one is a xPC box with the kernel of the corresponding real-time operating system installed. The host PC can be any Windows computer with MATLAB installed. The hardware is physically controlled by the xPC target. The controller is designed and coded using MATLAB Simulink with the toolbox of xPC Target. The Simulink model is compiled using Visual Studio C compiler, and the compiled object file was uploaded to the xPC target machine using Ethernet cable. Then, the real-time workstation sends out the commands to the motor driver. Meanwhile, it collects data from different sensors and sends them back to the host PC. The xPC box include two data acquisition cards: NI PCI-6601 and NI PCI-6621 manufactured by National Instrument. NI PCI-6601 includes 4 up/down counters with 32-bit resolution and 32 digital input/output ports. NI PCI-6621 has 80 analog-to-digital inputs with 16-bits resolution and 2 analog output channels. The readings from the sliding rheostats and motor encoders can be collected by these two DAQ cards. The control commands can be sent to the analog output channels. In this study, two L298 H-bridge devices are used to amplify the current and drive the adopted DC motors.

Though the performance of adopted DAQ cards are excellent, they cannot be used while fabricating a portable device. Most of the data acquisition devices used in the xPC box used in the testing platform need to be replaced by smaller components. Thus, the current control unit includes two microprocessor boards instead of using original xPC box. The schematic diagram is shown as Figure 3.8.



**Figure 3.8 Electrical structure of twisted string actuators control system**

The host PC, xPC Target, H-bridge and actuators are identical to the testing system. The new introduced Board is an Intel Galileo Gen 2, which has a 32bits Intel Pentium processor, 256 Mbytes DRAM, Ethernet port, 512 Kbytes of no-die embedded SRAM, 2-channel UART, 11 Kbytes EEPROM, 100 Mb, 16 Kbytes L1 cache, and 12bits PWM. While Spartan XC3S200A is a low-cost, high-performance FPGA board including 4320 logic cells, 76 input and output pins, 622+ Mb/s data transfer rate per I/O and 18 single-ended signal standards and 8 differential input output standards. By using both boards, enough I/O resource are guaranteed for the system.

In this setup, the controller is still synthesized using MATLAB Simulink and xPC toolbox. The basic layout of programs remains the same. The communication methods between each part are shown in figure. Ethernet cable is used to connect host PC with xPC Target, and the commands and feedback information are transmitted via Ethernet connection. An UDP communication is configured between xPC Target and an Intel

Galileo Gen 2 board. The transmitted UDP packet includes sensor data and commands of motor movement. The communication protocol between the Intel Galileo Gen 2 board and the Spanrta XC3S200A is SPI (Serial Peripheral Interface) communication. SPI communication is relatively easy to programming. One other important concern is high speed data transfer among various components. Both UDP and SPI communication are fast enough for the control requirement of this application.

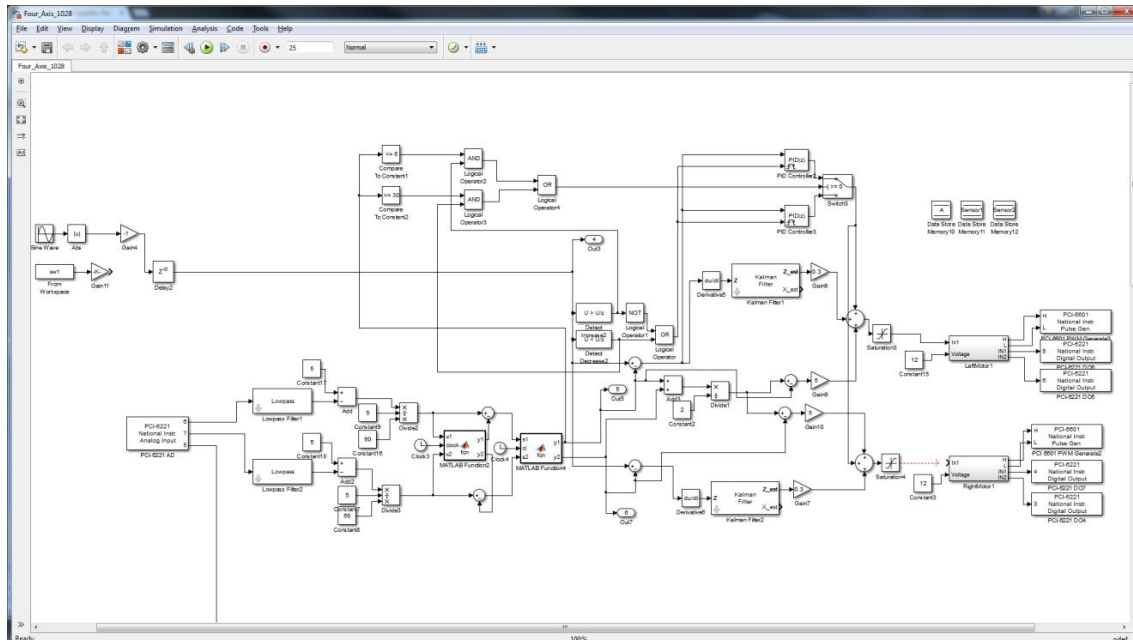
As to the basic components, the PMW signals are sent from Spanrta XC3S200A to H-bridge board to control motor speed. The reading of sensors such as sliding rheostat is transferred between with Intel Galileo Gen 2 microprocessor by electrical cables.

## **3.2 Software**

Adopting adequate software package is one of the most important issues in designing a desired controller because its performance affects controlled results directly. In this research, MATLAB Simulink with xPC Target Toolbox [47] and Arduino developmental environment [48] were used to synthesize the controller. The adopted functions and configuration of the software are discussed in this section.

### **3.2.1 MATLAB Simulink**

MATLAB Simulink is a graphical programming environment for modeling, simulating, and analyzing multi-domain dynamic systems, which is developed by MathWorks. In this study, MATLAB 2013a was used as the development environment and Simulink was used to create the model of the control system. Physical settings and parameters of controllers can be modified to fine-tune system performance. Figure 3.9 demonstrates part of the control Simulink model developed in this study.

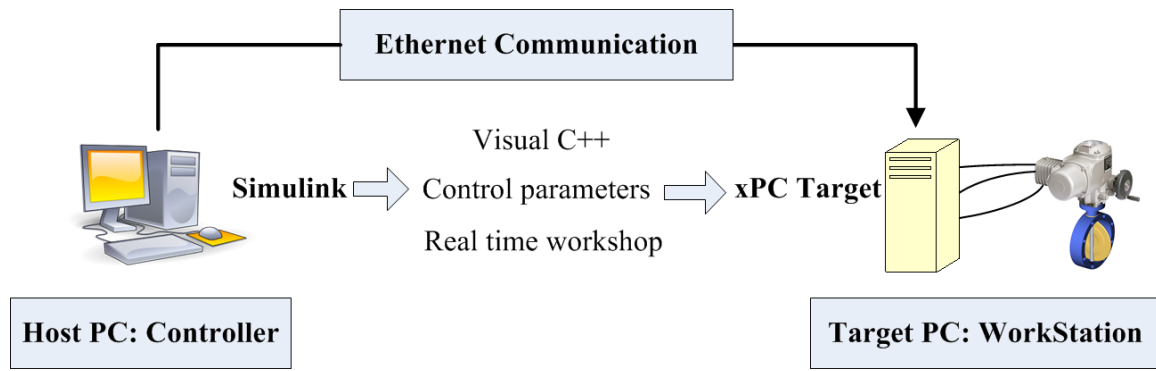


**Figure 3.9 Simulink graphical programmer of synthesis controller model**

One of the most important parameters while simulating the robotic system is the sample rate. According to Sampling Theorem [49], adopted sample rate must be at least twice faster than system frequency, and it is desired to be 10 times faster without severe distortion. In order to assure control performance, the sample rate was configured as 1 kHz. In addition to regular Simulink blocks, there are two important interfaces needed to be connected to xPC Target, which will be included in the next section.

### 3.2.2 MATLAB xPC Target Tool

The toolbox of xPC Target allows a stand-alone workstation to run a real-time operating system provided by MathWorks. All real-time control systems were implemented using xPC Target toolbox to control the robotic device in this study. In real-time control systems, the acts of reading sensors, generating feedback, and changing the corresponding control input are guaranteed within a set response time. The relationship between with Simulink can be illustrated as Figure 3.10.



**Figure 3.10 Work scheme of xPC target with simulink**

The controllers developed using Simulink are compiled by Microsoft Visual C++ compiler and uploaded to the xPC box. The xPC box then runs the program in real-time, which generates control commands and sends acquired data back to the host computer. In this research, two cards, NI PCI-6601 and NI PCI-6221, are used for data acquisition.

The purpose of this study is aimed to design a light weight, portable, and wearable motion-assistant device. Though the adopted data acquisition cards provide excellent performance and responding time, it is bulky and impossible to have a user to carry all the devices. Thus, it is desired to integrate all components within the robotic device with a compact size. With the development of advanced embedded microprocessor, the target PC can be substituted by a powerful microprocessor which has smaller size and less weight. The adopted microprocessor board is an Intel Galileo Gen 2 board and the firmware development environment will be discussed in next section. The protocol used to communicate between the xPC Target and the Intel Galileo Gen 2 is User Datagram Protocol (UDP). The modified development environment of Arduino 1.6.0 [50], which is provided by Intel, was used to develop the controllers. This specific version integrates Arduino IDE and the driver of Intel Galileo 2. The code of controllers can be found in Appendix A. An additional Xilinx Spartan FPGA board is also used for data acquisition.

### 3.3 Design of Controller

In order to synthesize an appropriate controller, the mathematical model of adopted motor and robot mechanism has to be derived. The system dynamic equations of DC motor can be described as following:

$$V = L \frac{di}{dt} + iR + K_E \omega \quad (3.6)$$

$$\tau = K_T i = J \frac{d\omega}{dt} + B\omega + T_D \quad (3.7)$$

where the definition of parameters are listed in Table 3.1.

**Table 3.1 Symbols of parameters used to describe motor dynamics**

Symbols	Description
$R$	Resistance of armature
$L$	Inductance of armature
$K_E$	Back EMF constant
$K_T$	Motor torque constant
$J$	Inertia coefficient
$B$	Viscous coefficient
$T_D$	Torque disturbance
$\omega$	Motor velocity

While deriving the motor model, disturbance caused by external torque,  $T_D$ , is ignored. Rearranging eqs. (3.6) and (3.7), the differential equation of current and motor velocity can be obtained:

$$\frac{L}{R} \frac{d^2 i}{dt^2} + \left(1 + \frac{LB}{RJ}\right) \frac{di}{dt} + \left(\frac{B}{J} + \frac{K_E K_T}{RJ}\right) i = \frac{1}{R} \left(\frac{dv}{dt} + \frac{B}{J} v\right) \quad (3.8)$$

$$\frac{L}{R} \frac{d^2 \omega}{dt^2} + \left(1 + \frac{LB}{RJ}\right) \frac{d\omega}{dt} + \left(\frac{B}{J} + \frac{K_E K_T}{RJ}\right) \omega = \frac{K_T}{RJ} v \quad (3.9)$$

Then, assuming the armature inductance of the DC motor is small enough to be ignored and the analytic solution of current and motor velocity could be obtained by applying a fixed voltage  $V_0$  when motor is static:

$$i(t) = \frac{V_0}{R} \left( \frac{B\tau}{J} + \frac{K_E K_T \tau}{RJ} e^{-\frac{t}{\tau}} \right) \quad (3.10)$$

$$\omega(t) = \frac{V_0 K_T \tau}{RJ} \left( 1 - e^{-\frac{t}{\tau}} \right) \quad (3.11)$$

where  $\tau = 1 / \left( \frac{B}{J} + \frac{K_T K_E}{RJ} \right)$  is the time constant. The initial and steady states of the system are represented as  $i_0$ ,  $\omega_0$ ,  $i_{ss}$  and  $\omega_{ss}$ , respectively. With these two states, important parameters of motor model can be determined by following equations:

$$K_T = K_E = \frac{V_0}{\omega_{ss}} \left( 1 - \frac{i_{ss}}{i_0} \right) \quad (3.12)$$

$$R = \frac{V_0 - K_E \omega_{ss}}{i_{ss}} \quad (3.13)$$

$$B = \frac{\frac{K_E K_T}{R} \frac{i_{ss}}{i_0}}{1 - \frac{i_{ss}}{i_0}} \quad (3.14)$$

$$J = \tau \left( B + \frac{K_E K_T}{R} \right) \quad (3.15)$$

Therefore, the mathematical model of motor can be determined. Now, considering torque disturbance in this system. Taking Laplace transformation of eqs. (3.6) and (3.7), the relationship between input and output can be identified as:



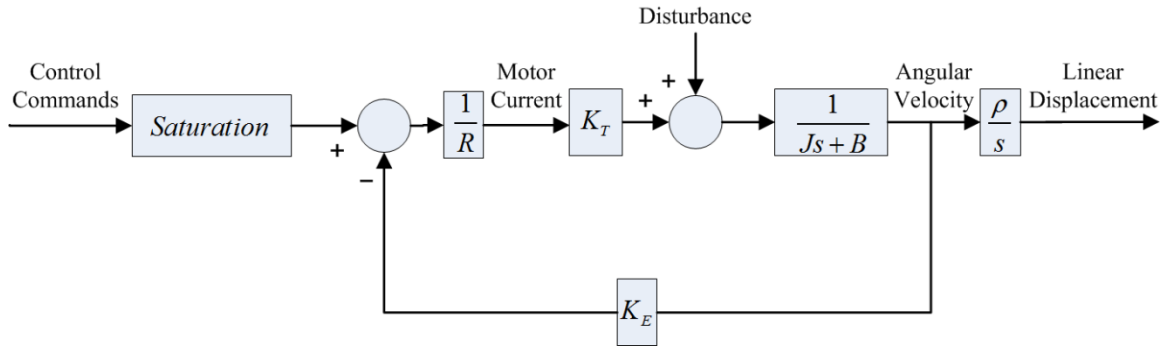
$$\Omega(s) = G(s) \cdot [V(s) - D(s)T_D(s)] \quad (3.16)$$

The inputs of this model are voltage and torque disturbance, and system output is angular velocity of motor. The nominal transfer functions of  $G(s)$  and  $D(s)$  can be derived from eqs. (3.6) and (3.7) as well. They are:

$$G(s) = \frac{K_T}{LJs^2 + (RJ + BL)s + (RB + K_E K_T)} \quad (3.17)$$

$$D(s) = \frac{-R - L(s)}{K_T} \quad (3.18)$$

Those two transfer functions can be identified by system identification techniques such as frequency response measurement technique. Assuming the armature inductance  $L$  of motor is small enough, the transfer function of the DC motor can be approximated as a first order system. Considering with twisted string actuator, the control scheme of twisted string actuator can be illustrated as Figure 3.11.



**Figure 3.11 Block diagram of twisted string actuator**

In Figure 3.11,  $\rho$  is the transmission ratio of the twisted string. If the reference input command is  $r$ , the closed loop response of this system can be modeled as:

$$L(s) = \frac{K_T \rho}{JRs^2 + (BR + K_T K_E)s} r(s) + \frac{\rho}{Js^2 + (B + \frac{K_T K_E}{R})s} d(s) \quad (3.19)$$

where  $L(s)$  is the linear displacement of system output,  $r(s)$  is reference input commands, and  $d(s)$  is system disturbance input.

In practice, nonlinear factors, such as friction, hysteresis, dead-zone effect, and poor repeatability in twisted string system, need to be considered. These nonlinear factors and un-modeled disturbance can possibly cause poor tracking performance. For example, the inconsistent friction among strands is varying due to the changing normal force between contacting surfaces of strands. The dead zone effect can also create nonlinearity while the robotic system is changing its direction. In this work, only the external disturbance is considered. Thus, a linear controller cannot provide an adequate tracking performance. In order to properly track the given trajectories, a nonlinear controller that combines regular PID control scheme and fuzzy logic control was use to compensate for errors.

### **PID Controller**

For PID controller [51], the transfer function of the controller can be formulated as

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (3.20)$$

where  $K_p$ ,  $K_i$ , and  $K_d$  denote the coefficients for the proportional, integral, and derivative controllers. With Laplace transform, the transfer function becomes:

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (3.21)$$

Rearranging eq. (3.21), the controller can be written as

$$G_c(s) = K_p \frac{T_i T_d s^2 + T_i s + 1}{T_i s} \quad (3.22)$$

where  $T_i = \frac{K_p}{K_i}$  is the integral time constant and  $T_d = \frac{K_d}{K_p}$  is the derivative time constant.

### **Fuzzy Controller**

As to fuzzy logic control, it is a rule-based control scheme. According to Tanaka and Sugeno [52], the dynamic model of a fuzzy controller can be represented as following:

$$\begin{aligned} R^i : & \text{If } x_1(k) \text{ is } X_1^i \text{ and } \dots \text{ and } x_G(k) \text{ is } X_G^i, \\ & \text{then } y^i(k+1) = \alpha_0^i(k) + \sum_{n=1}^N \alpha_n^i y(k-n+1) + \sum_{m=1}^M \beta_m^i v(k-m+1), \quad i=1, \dots, I. \end{aligned} \quad (3.23)$$

where  $y^i(k+1)$  is the system output of the  $i^{\text{th}}$  implication,  $y(k), \dots, y(k-N+1)$  are state variables,  $\alpha_n^i$  and  $\beta_m^i$  are consequent parameters,  $v(k), \dots, v(k-M+1)$  are input variables,  $x_1(k), \dots, x_G(k)$  are the premise variables and  $X_1^i, \dots, X_G^i$  are the corresponding fuzzy sets. If the input, premise variables, and state variables are given, the output of the fuzzy logic model can be represented as:

$$y(k+1) = \frac{\sum_{i=1}^I w^i(k) y^i(k+1)}{\sum_{i=1}^I w^i(k)} \quad (3.24)$$

where  $w^i(k)$  is a function of the premise variables. In this study, premise variables denote system states which are the positions of sensors, and the system outputs are control signals generated by fuzzy controller. The fuzzy logic rules used to control the nonlinear system individually are identified based on experiment and are listed in next chapter when discussing the results of fuzzy logic control. Finally, in the synthesized controller, fuzzy controller is used to compensate nonlinear effects while PID controller is used to take care of linear part of control process.

### **Sensor Fusion Technique**

In this study, some extra sensors are available in the system to help improving control performance. For example, the encoder located inside motors can be used as extra position sensor to improve position reading from other sensor by using sensor fusion

technique. The principle of sensor data fusion technique is straightforward. The basic idea of data fusion technique comes from statistics. According to Baye's theorem [53], the relationship between system state  $x$  and observation  $z$  are defined as the joint probability or joint probability distribution  $P(x, z)$  [54]. Chain rule of conditional probabilities can be used to expand a joint probability. That is

$$P(x, z) = P(x | z)P(z) = P(z | x)P(x) \quad (3.25)$$

Rearranging eq. (3.25), the Bayes's rule becomes:

$$P(x | z) = \frac{P(z | x)P(x)}{P(z)} \quad (3.25)$$

By analyzing the system state and observation, their observations are independent that can meet the requirement of conditional independence if more than two sensors are existing in the system. That is

$$P(z_1, z_2, \dots, z_n | x) = P(z_1 | x)P(z_2 | x) \cdots P(z_n | x) = \prod_{i=1}^n P(z_i | x) \quad (3.26)$$

Therefore,

$$P(x | Z^n) = CP(x) \prod_{i=1}^n P(z_i | x) \quad (3.27)$$

Usually, eq. (3.27) is known as the independent likelihood pool [55] where  $C$  is a normalizing constant. The all probabilities of the system state can be determined by this equation if a fixed observation can be given. Rearranging this equation with a recursive form, it becomes:

$$P(x | Z^k) = \frac{P(z_k | x)P(x | Z^{k-1})}{P(z_k | Z^{k-1})} \quad (3.28)$$

The advantage of using this method is that only the prior-state probability density function  $P(x | Z^{k-1})$  is needed to calculate and store. As the next time-staple information  $P(z_k | x)$  is derived, the post-state probability density of the preceding time-staple can be

used as the prior-state probability density function of current time-staple. The post-state probability density function of current time-staple can be plugged into the equation after multiplying by current state information. Thus, the probabilities of sensor measurements can be determined at every time-staple. They can also be used as the weight factors while fusing data weighting.

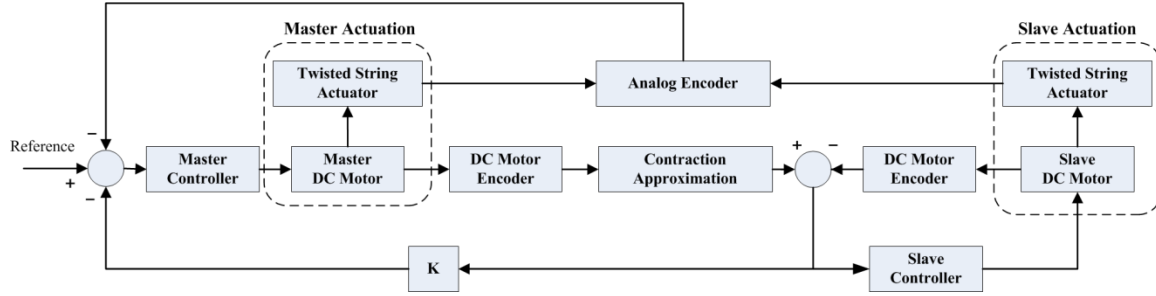
Assuming the measurements from two different sensors are  $X_1$  and  $X_2$ , and the noise variance are  $\sigma_1^2$  and  $\sigma_2^2$ , respectively. Then the result of data sensor fusion is:

$$X_3 = \sigma_3^2 (\sigma_1^{-2} X_1 + \sigma_2^{-2} X_2) \quad (3.29)$$

where  $\sigma_3^2 = (\sigma_1^{-2} + \sigma_2^{-2})^{-1}$  is the noise variance of the fused system. The results are discussed in next chapter. In this function, the measurements of motor encoder and sliding rheostat are used.

### **Master-Slave Controller**

In the past, the design of single twisted string actuator control system has been studied. The individual joint of a robotic arm is controlled by a pair of twisted string actuator as mentioned in previous section. The dual antagonistic twisted string system can be modeled as two independent linear systems which have same control models. However, while operating dual antagonistic twisted string control system, it is necessary to take both single tracking performance and motion synchronization into consideration, which means individual actuation devices need to have corresponding controllers to compensate both tracking and synchronization errors. In this study, a master-slave control method has been used as synthesis strategy due to its simplicity and easy implementation. The scheme of smart-slave used in this research is illustrated in Figure 3.12.



**Figure 3.12 Block diagram of master-slave control architecture**

The synthesis of controller for both master and slave drivers are based on the dynamic model of twisted string actuator, PID controller, fuzzy logic controller and data sensor fusion technique derived early. In other words, as illustrated in Figure 3.12, individual control model of master actuation and slave actuation are same and only difference is the adding of master-slave synchronization mechanism. The output of master actuator affects both mechanisms by a feedback loop. The strings of master and slave were pre-twisted to a certain level so that they are tight enough and ready to generate linear displacement. This is important because the initial positions of both actuators have a huge influence on this system. Due to the special structure of twisted string actuator, strings need certain tiny time period to twist together to generate linear displacement. Such delays coming from twist-loose strings cannot be ignored since they can increase response time that introduces additional uncertainty to the system. With the existence of these nonlinear factors, the amounts of revolutions of two DC motors generating to contracting and releasing are different. The results of revolutions for both contracting and releasing will be different depending on the number of pre-twisted revolutions.

## CHAPTER 4

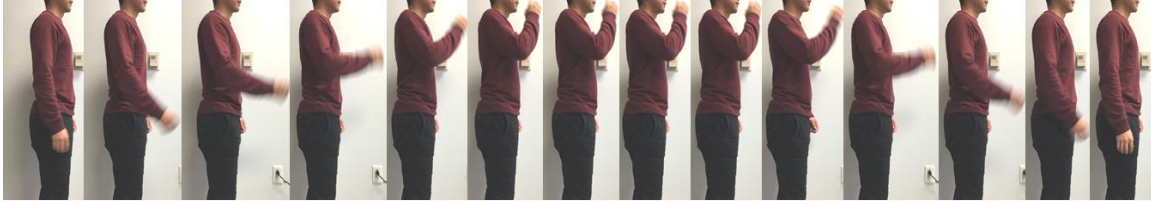
### EXPERIMENTAL RESULTS

#### 4.1 Introduction

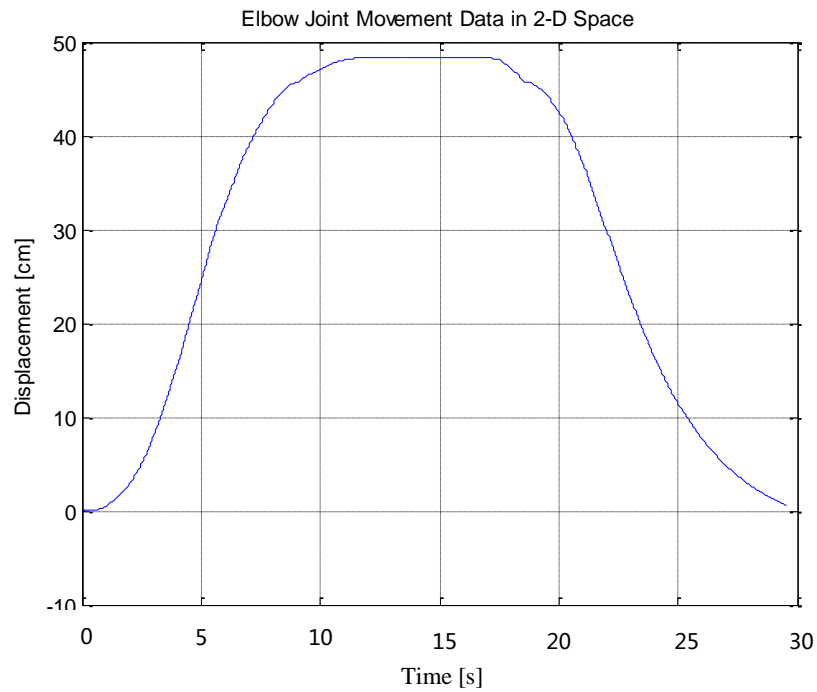
In order to verify the performance and effectiveness of the proposed control algorithms, this chapter demonstrates the control outputs of both tracking and synchronization. Certain types of trajectories were chosen as reference signals to evaluate the performance of the proposed controllers.

The selection of trajectories plays a fundamental role of tracking performance. In particular, the non-parametrical uncertainties due to quantization of digital devices might lead to large tracking errors. Such kinds of non-parametrical uncertainties can make a stable system an unstable one. Therefore, to verify the performance of the proposed controller, carefully selected trajectories are necessary.

The goal of the proposed control algorithms is to compensate both the tracking and synchronization errors simultaneously. Figure 4.1 illustrates the numerical displacement of elbow joint from a healthy male adult subject collected by a depth camera (Microsoft Kinect) [56]. As shown in Figure 4.1(a), the movement of elbow joint was collected by Kinect during the process of mimicking drinking motion performed by a male subject. The sample rate of the adopted video is 10 frames per second, and the acquired trajectory of angles was interpolated between intervals. While performing such a simple motion, Kinect can collect spatial information of specific joint along with time span so that one can restore this movement based on the acquired data. However, due to the limitation of current mechanical structure, the robot arm in this experiment can only perform 2-D movements. Thus, it is necessary to extract 2-D information from the 3-D data collected by Kinect and plot it in terms of time span. Figure 4.1(b) shows the displacement that represents the elbow motion of the subject.



(a) Motion of drinking recorded from a male subject



(b) Quantization of elbow joint movement

**Figure 4.1 Numerical displacement of elbow joint when performing “drinking” motion**

Similarly, other curves of different motions can be obtained by repeating the same procedures. Trajectories of shoulder joint can be obtained with identical procedures as well.



## 4.2 Trajectory Tracking Results of Single Axis Movement

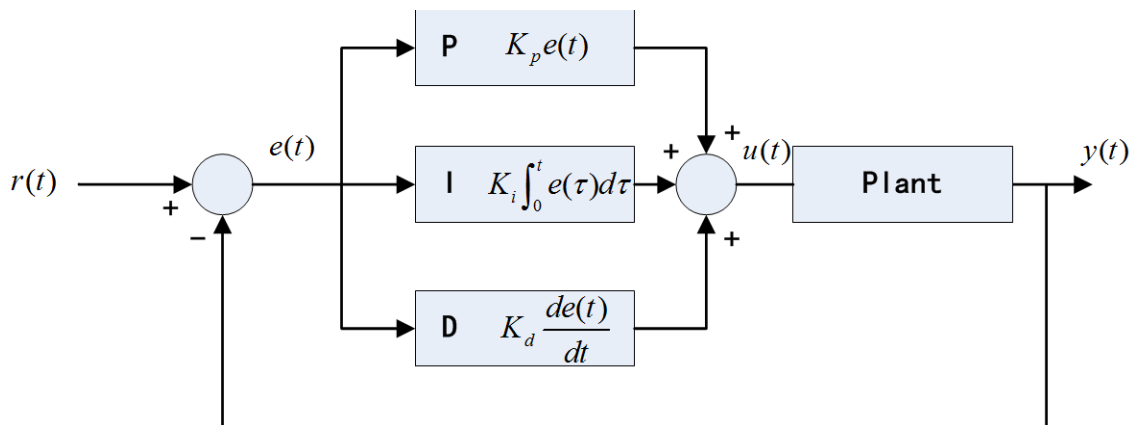
In this control application, the actuation of a single joint is driven by two twisted-string actuation. To control of the dual-axial joint, movements of individual axes have to be controlled precisely to minimize tracking errors. In the past few years, several traditional methods were used to control a single axial system of different twisted-string systems. Two different types of controllers, PID and fuzzy methods, were synthesized to compensate the control results in this chapter.

### 4.2.1 Results of PID Control

Traditional PID (Proportional-Integral-Derivative) method is a classical method, which can continuously calculate tracking error between a measured process variable (system output) and a desired set-point (reference trajectory). It also attempts to minimize the error over time by adjustment of control variable, the position of robot arm in this case, to a new value determined by a weighted sum:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (4.1)$$

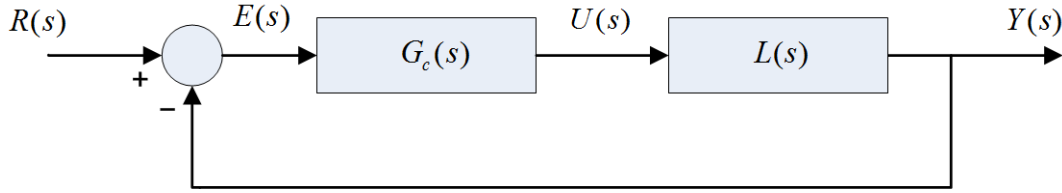
where  $K_p$ ,  $K_i$ , and  $K_d$  denote the coefficients for the proportional, integral, and derivative terms. As shown in Figure 4.2, different coefficients have different meaning and function in this controller.



**Figure 4.2 Block diagram of PID control in a feedback loop**

The three coefficients  $K_p$ ,  $K_i$ , and  $K_d$  are in charge of different responses of error signals. The proportional coefficient  $K_p$  accounts for present values of the error. The integral coefficient  $K_i$  accounts for time impact of past error. It means that the accumulated error can cause a stronger action if the output is not sufficient to reduce the error. The derivative coefficient  $K_d$  accounts for possible future error, which is based on the current change rate of error. Some applications may require using only one or two terms to provide the appropriate system control and the controlled result can decay to zero quickly by adjusting the parameters.

To design an adequate PID controller, the system dynamics of the single axis twisted-string actuator is required. The linear and nonlinear part of this actuator has been discussed in the previous chapter. The PID controller proposed here is based on the linear model of the corresponding transfer function. According to eqs. (3.19) and (3.22), the transfer function of this linear actuator and PID controller are  $L(s)$  and  $G_c(s)$ , respectively. The block diagram of this control process is shown in Figure 4.3.



**Figure 4.3 Block diagram of simulation process**

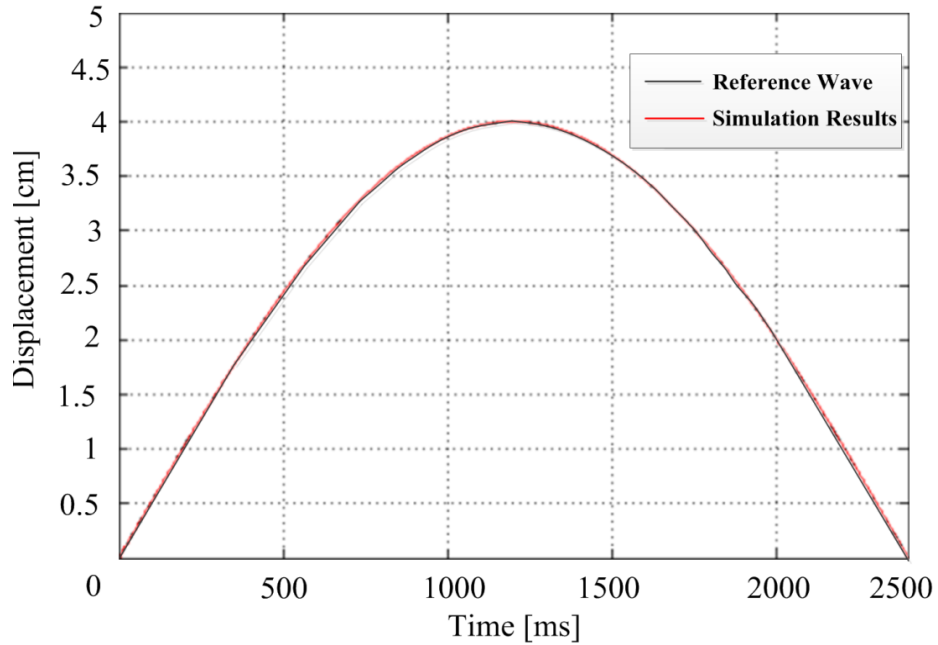
Assuming the system disturbance is ignorable, the transfer function used for simulation is

$$G(s) = \frac{Y(s)}{R(s)} = \frac{G_c(s)L(s)}{1 + G_c(s)L(s)} = \frac{K_p K_T \rho T_i T_d s^2 + K_p K_T \rho T_i s + K_p K_T \rho}{J R T_i s^3 + (T_i B R + T_i K_T K_E) s^2} \quad (4.2)$$

Simplifying eq. (4.2) with  $T_i = \frac{K_p}{K_i}$  and  $T_d = \frac{K_d}{K_p}$ , the transfer function can be written as

$$G(s) = \frac{K_T K_d \rho s^2 + K_T K_p \rho s + K_T K_i \rho}{J R s^3 + (B R + K_T K_E) s^2} \quad (4.3)$$

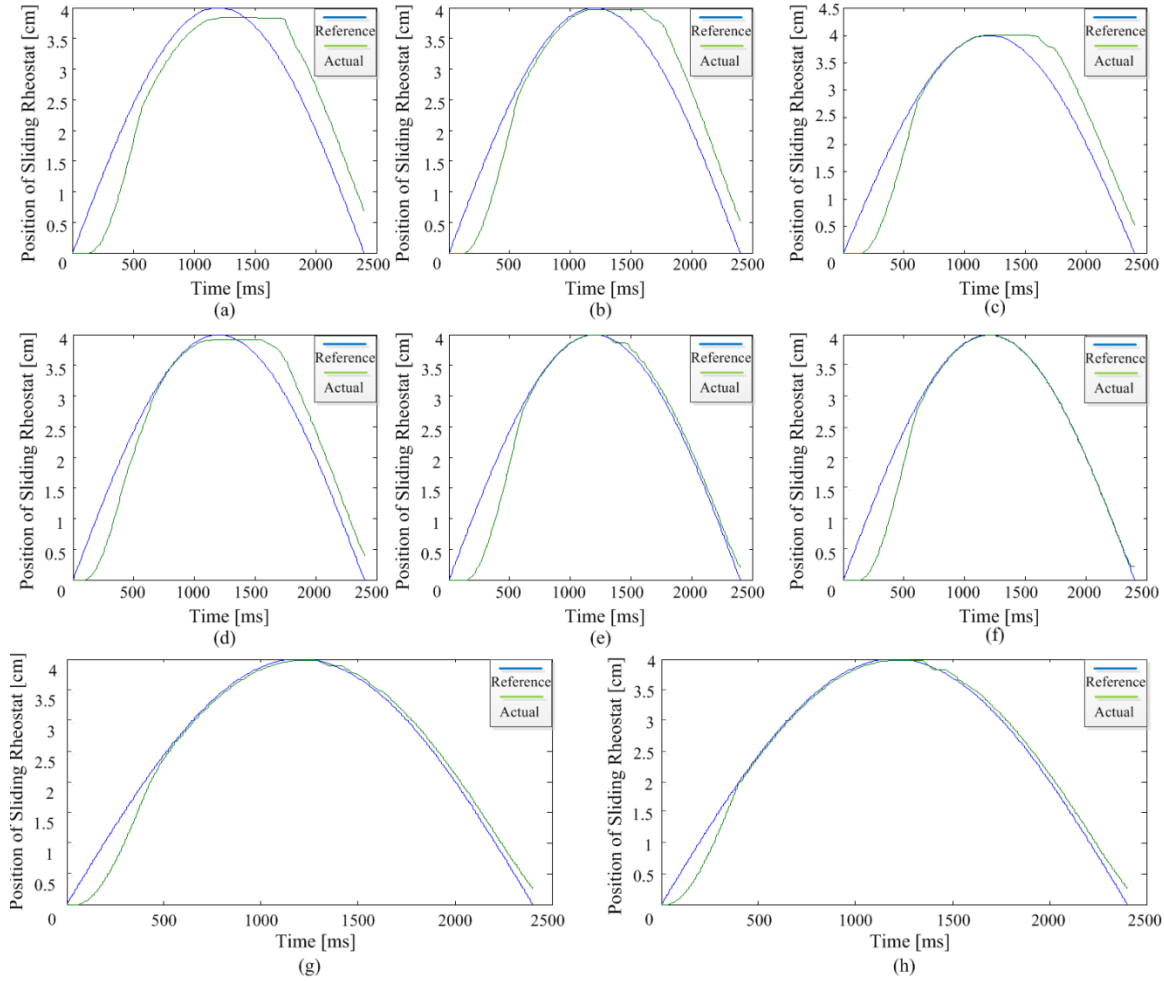
The following parameters of twisted string actuator model are used: the armature resistant  $R$  of the DC motor is 9.5 ohm, the inductance of armature  $L$  is  $2.7 \times 10^{-4}$ , the back EMF constant  $K_E$  is  $6.16 \times 10^{-2}$ , the torque constant  $K_T$  is 0.1182, the inertia coefficient  $J$  is  $2.5 \times 10^{-4}$ , and the viscous coefficient  $B$  is  $10^{-4}$ . In order to achieve desired control performance, the following criteria have to be met: 1) maximum overshoot needs to be less than 5%; 2) settling time is less than 1 sec; 3) damping ratio is close to 1; and 4) time constant is less than 0.3 sec. To guarantee the controlled result, root-locus or pole-placement methods can be used to obtain the three coefficients. With these criteria, the parameters were obtained by using PID tuning tool. In this case,  $K_p$  is 15.011 and  $K_i$  is 1.003. Figure 4.4 demonstrates the simulated results of the calculated parameters.



**Figure 4.4 Simulation results of PID controller with calculated parameters**

Based on the simulated results, it is clear that most of the parameters can satisfy the requirement of tracking and the tracking errors are within an acceptable ranges ( $\pm 0.02$  mm). However, though the parameters based on calculation can have an excellent simulated results, it cannot always be repeated in the experimental outcome due to the nonlinear factors, which were not considered while designing the PID controller. Thus, additional sets of parameters were also used to evaluate the controlled outcome. In the

twisted-string actuator system built in this thesis, the control performances of a single actuator axis corresponding to different control coefficients are shown in Figure 4.5.



**Figure 4.5 Results of PID control from different parameters: (a)  $K_p=15$ ;  $K_i=1$ ;  $K_d=0$ ; (b)  $K_p=16$ ;  $K_i=2$ ;  $K_d=0$ ; (c)  $K_p=16$ ;  $K_i=2$ ;  $K_d=1$ ; (d)  $K_p=16$ ;  $K_i=2$ ;  $K_d=2$ ; (e)  $K_p=16$ ;  $K_i=2$ ;  $K_d=2.5$ ; (f)  $K_p=16$ ;  $K_i=2.5$ ;  $K_d=2.5$ ; (g)  $K_p=16$ ;  $K_i=2.5$ ;  $K_d=2.5$ ; (h)  $K_p=16$ ;  $K_i=2.5$ ;  $K_d=3$ .**

By analyzing transient performance of the system, desired requirements were met for all sets. For example, for set (g) in Figure 4.5, the maximum overshoot should be 1.7%, settling time is 0.96 sec, damping ratio is 1, and the time constant is 0.2951 sec. On the other hand, there are several effective methods to manually tune the parameters of PID controller such as Tyreus Luyben's method, Autotune method, Cohen-Coon's method and Ziegler-Nichols' method [57]. In this experiment, Ziegler-Nichols' method was

adopted to fine tune PID parameters. The standard procedures of getting desired control performance,  $K_i$  and  $K_d$  need to be set to zero. Then only a proportional control parameter was tuned and it is increased until it reaches the ultimate gain  $K_u$  corresponding to the ultimate period  $T_u$ , which causes sustained periodical oscillation. Once these two values are determined, the control parameters can be determined as well. Thus, based on the calculated values of  $K_p$ ,  $K_i$ , and  $K_d$ , the actual parameters can be tuned accordingly. Table 4.1 lists the rules of the tuning rules of different types of controllers.

**Table 4.1 Ziegler-Nichols Method**

Control Type	$K_p$	$K_i$	$K_d$
P	$0.5K_u$	0	0
PI	$0.45K_u$	$1.2K_p / T_u$	0
PID	$0.6K_u$	$2K_p / T_u$	$K_p T_u / 8$

With the experimental setup of robotic device and tuning, the threshold values of  $K_u$  and  $T_u$  are around 33 and 16, respectively. Therefore, for a PI controller, the values of proportional and integral parameter are 15 and 1, respectively. These two parameters are based on the design using PID tuning tool. The controlled result is shown in Figure 4.5(a). The results are relatively poor because system response is not fast enough to compensate for the desired trajectory. By increasing proportional and integral gains to 16 and 2 individually, system response becomes faster, which can be proved by evaluating the characteristic function of the controlled system. It is also clear that the controlled result can be improved as shown in Figure 4.5(b). However, the output is still inaccurate. Therefore, derivative parameter was introduced with an initial value 1. Based on the calculated values derived from the simulated system dynamics, it is easier to approximate the parameters in the tuning process. Figure 4.5(c)-(e) show the results of PID control with different parameters. The increasing  $K_d$  also yields the system to have faster response. Compared with Figure 4.5(c) and (d), the tracking error can be improved dramatically. In particular, the motion that lowers the mechanism is accurately

compensated with the assistance of the weight of the mechanism. Due to dead-zone effect and larger starting force, the part of motion that raises the mechanism is typically a delayed function. However, though the lowering motion has been compensated adequately, steady-state error can still be observed. In order to eliminate steady-state error, the integral gain was increased by 0.5. As shown in Figure 4.5(f), the steady-state error can be decreased effectively while maintaining the same controlled result for the lowering motion. To solve the delay phenomenon while raising the mechanism, the number of pre-twisted revolutions is important in this twisted-string system. To demonstrate the importance of pre-twisting the string, the controlled results were compared. In Figure 4.5(f), the string is pre-twisted for 10 revolutions, and it is 12 revolutions in Figure 4.5(g). It is clear that the pre-twisted string can have a better tracking performance with same control parameters. In particular, the overall control error can be decreased as well. By fine tuning the value of  $K_d$  to 3, the tracking performance can be better. The result is shown as Figure 4.5(h).

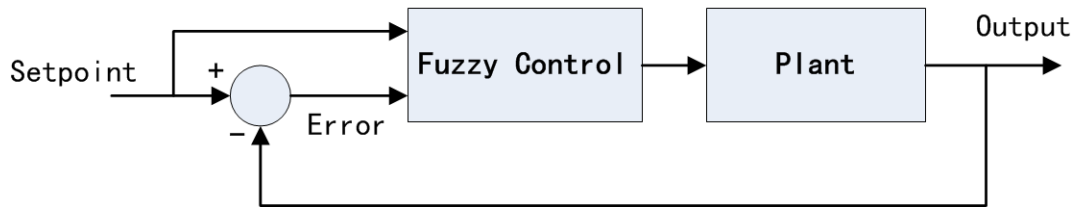
By tuning the three control parameters of PID controller, a series of improvement were obtained as discussed above. However, the use of the PID algorithm does not guarantee optimal controlled outcome of the system. In particular, this method does not take nonlinearity of both input and un-modeled dynamics into consideration. Thus, though it demonstrates excellent controlled outcome, the robustness cannot be guaranteed. In some extreme cases, the system might not be stable due to the excitation of external disturbances. Therefore, nonlinear controller is desired to compensate for these issues.

#### **4.2.2 Results of Fuzzy Control**

In order to have more precise controlled outcome for such a nonlinear system, a fuzzy logic control method was adopted in this study. Once the nonlinear phenomena can be identified, the fuzzy logic can be combined with PID controller for more precise controlled results.

A fuzzy control system [58] is a control system based on fuzzy logic, which is a mathematical system that analyzes analog input values in terms of logical variables.

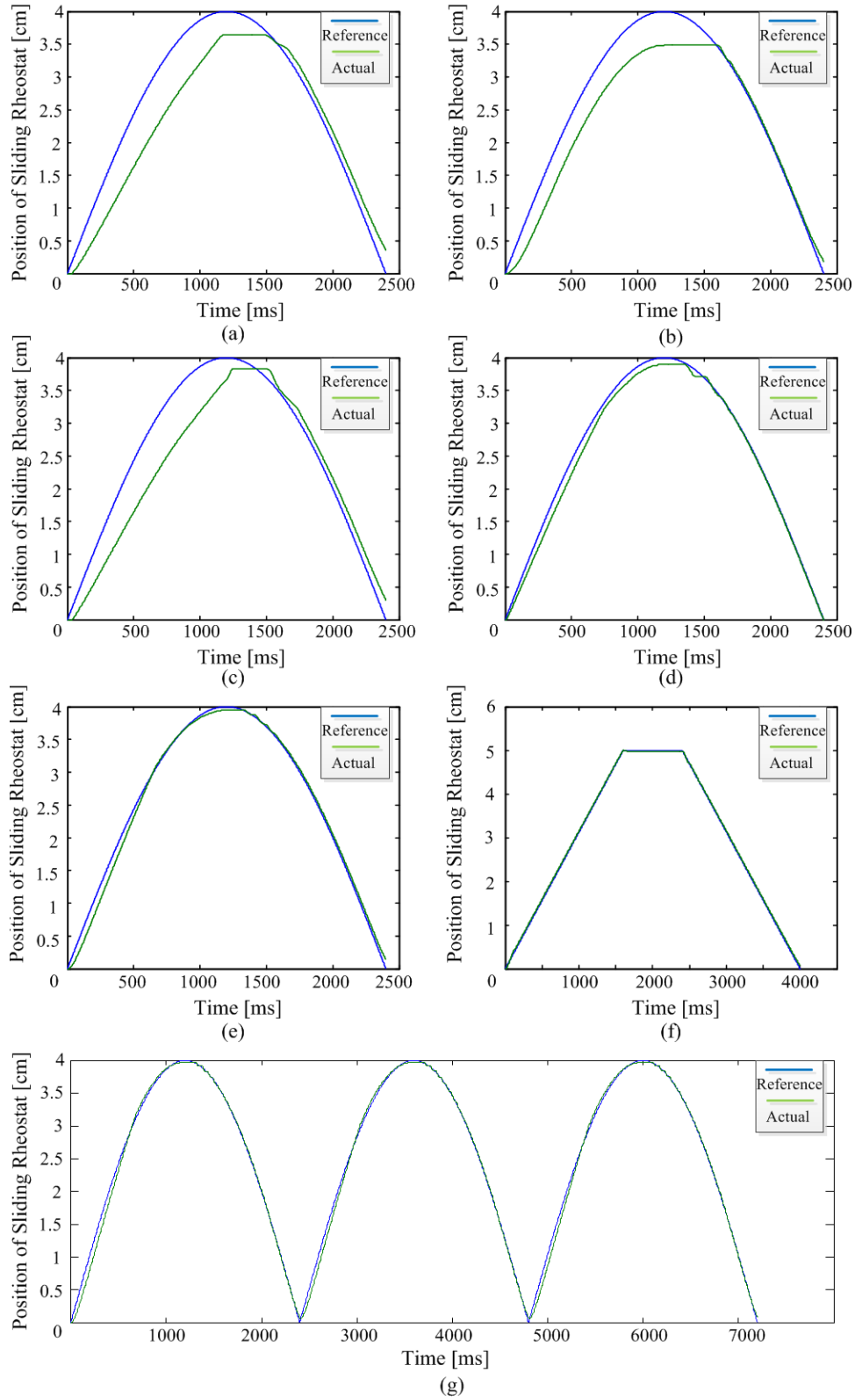
Typically, these variables are continuous values between 0 and 1. The term “fuzzy” refers to the fact the logic involved can deal with concepts that cannot be expressed as only “true” or “false” but rather as “partially true”. Fuzzy logic is widely used in machine control and has the advantages that the solution to the problem can be cast in terms that human operators can understand so that their experience can be used in the design of controllers. A basic block diagram of fuzzy logic control is illustrated as Figure 4.6, which is similar to traditional PID controller.



**Figure 4.6 Block diagram of fuzzy logic control system**

As shown in Figure 4.6, fuzzy logic controller is implemented in the feedback control loop. The control rules are based on experience. In certain system, some repeated control signals will yield similar outputs, and the rules of control to achieve optimal performance can be collected and constructed from these outputs. As long as the rules are configured to handle every possible situation, the fuzzy logic controller can be trusted in controlling specific systems.

Based on the experience of PID control tuning, the rules for fuzzy logic controller can be determined. In this case, in order to improve the performance and reliability of fuzzy logic controller, more than one input factor was chosen to construct fuzzy rules. These factors include feedback error, actual location of position sensor, and the direction of motion. The main goal of this fuzzy logic controller is to compensate for the nonlinear issues that traditional PID controller cannot take care of. Thus, the rules focus on changing direction and mechanism raising up. The controlled results are shown in Figure 4.7.



**Figure 4.7 Results of fuzzy control from different fuzzy logics**



Based on the tuning procedure, it is clear that the tracking error is decreased by adjusting fuzzy logic rules. Figure 4.7(a) illustrates the results of initial guess based on previous tuning experience. However, the tracking performance is poor. The error between reference position and actual position is relatively large. Thus, D-like and I-like rules were added to the fuzzy logic controller according to the controlled outcome. Figure 4.7(b) illustrates the results of tuning fuzzy logic values corresponding to mechanism lowering motion. Figure 4.7(c) and 4.7(d) demonstrate the results of compensating for changing direction and mechanism raising motion, respectively. Minor adjustments were made to improve the accuracy for dead-zone and raising motion. The result is shown as Figure 4.7(e). It is clear that more accurate tracking results can be obtained by tuning fuzzy logic rules. To ensure the repeatability of the proposed controller, a symmetric trapezoid signal and a three-cycle sinusoid signal were used as the desired trajectories with same fuzzy logic used in Figure 4.7(e). Figure 4.7(f) shows the results corresponding to a trajectory of slow movement. It is clear that the tracking performance is excellent if the required response time is relatively longer. Figures 4.7(g) illustrates the results of tracking a triple sinusoid curve. From the experimental result, this controller has a quick response to the desired trajectories and it shows a stable reaction to repeated input signals. To implement the fuzzy rules, Fuzzy Logic Design tool of MATLAB was used to generate the control commands. The ultimate rules used in Figure 4.7(g) are shown in Table 4.2 and 4.3.

**Table 4.2 Definition of variables used in fuzzy logic rules**

I/O Setting Variables		Position Input Range [-4 4]	Error Input Range [-0.8 0.8]	Control Output Range [-13 13]
mf1	Parameters	[-5.799 -5.78 -0.4 -0.3]	[-1.2 -0.8 -0.4]	[-19.56 -12.84 -11.3 -5.7]
	Type	trapmf	trimf	trapmf
mf2	Parameters	[-0.4 -0.2 0]	[-0.8 -0.4 0]	[-9.87 -5.124 0]
	Type	trimf	trimf	trimf
mf3	Parameters	[-0.2 0 0.2]	[-0.4 0 0.4]	[-5 0 5]
	Type	trimf	trimf	trimf
mf4	Parameters	[0 0.2 0.4]	[0 0.4 0.8]	[0 5.276 10.49]
	Type	trimf	trimf	trimf
mf5	Parameters	[0.3 0.4 5.789 5.804]	[0.4 0.8 1.2]	[5.7 11.3 12.89 20.01]
	Type	trapmf	trimf	trapmf

**Table 4.3 Rules of fuzzy logic controller**

Rules Number	Position Input	Error Input	Control Output
1	mf1	mf1	mf1
2	mf1	mf2	mf1
3	mf1	mf3	mf1
4	mf1	mf4	mf1
5	mf1	mf5	mf2
6	mf2	mf1	mf1
7	mf2	mf2	mf1
8	mf2	mf3	mf2
9	mf2	mf4	mf2
10	mf2	mf5	mf2
11	mf3	mf1	mf1
12	mf3	mf2	mf2
13	mf3	mf3	mf3
14	mf3	mf4	mf4
15	mf3	mf5	mf5
16	mf4	mf1	mf4
17	mf4	mf2	mf4
18	mf4	mf3	mf4
19	mf4	mf4	mf5
20	mf4	mf5	mf5
21	mf5	mf1	mf5
22	mf5	mf2	mf5
23	mf5	mf3	mf5
24	mf5	mf4	mf5
25	mf5	mf5	mf5

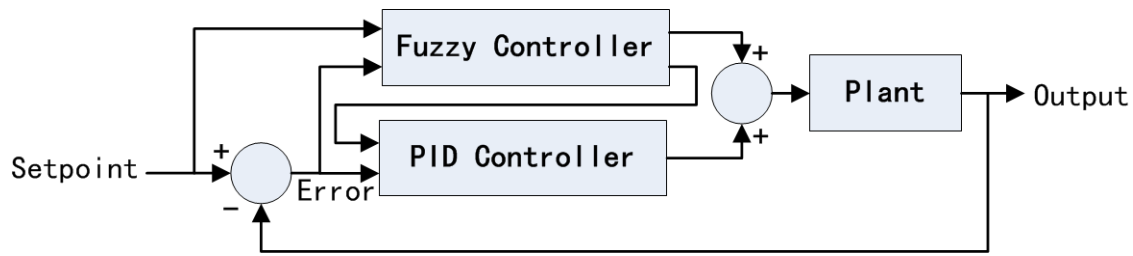
In Table 4.2, the configurations come from MATLAB Fuzzy Logic Design Tool. The parameters of each variable represent distribution along its ranges, while the types of variables determine the shapes of the output curves. In this case, type “trapmf” means trapezoidal membership function and type “trimf” means triangular membership function.

#### **4.2.3 Gain Scheduling Based on PID and Fuzzy Control**

As discussed in the preceding section, a single method is hard to meet comprehensive control requirements and to guarantee a stable performance. To compensate a nonlinear system, gain scheduling is used to combine both PID and fuzzy control methods. This is a common approach that can control a nonlinear system with a family of linear controllers.

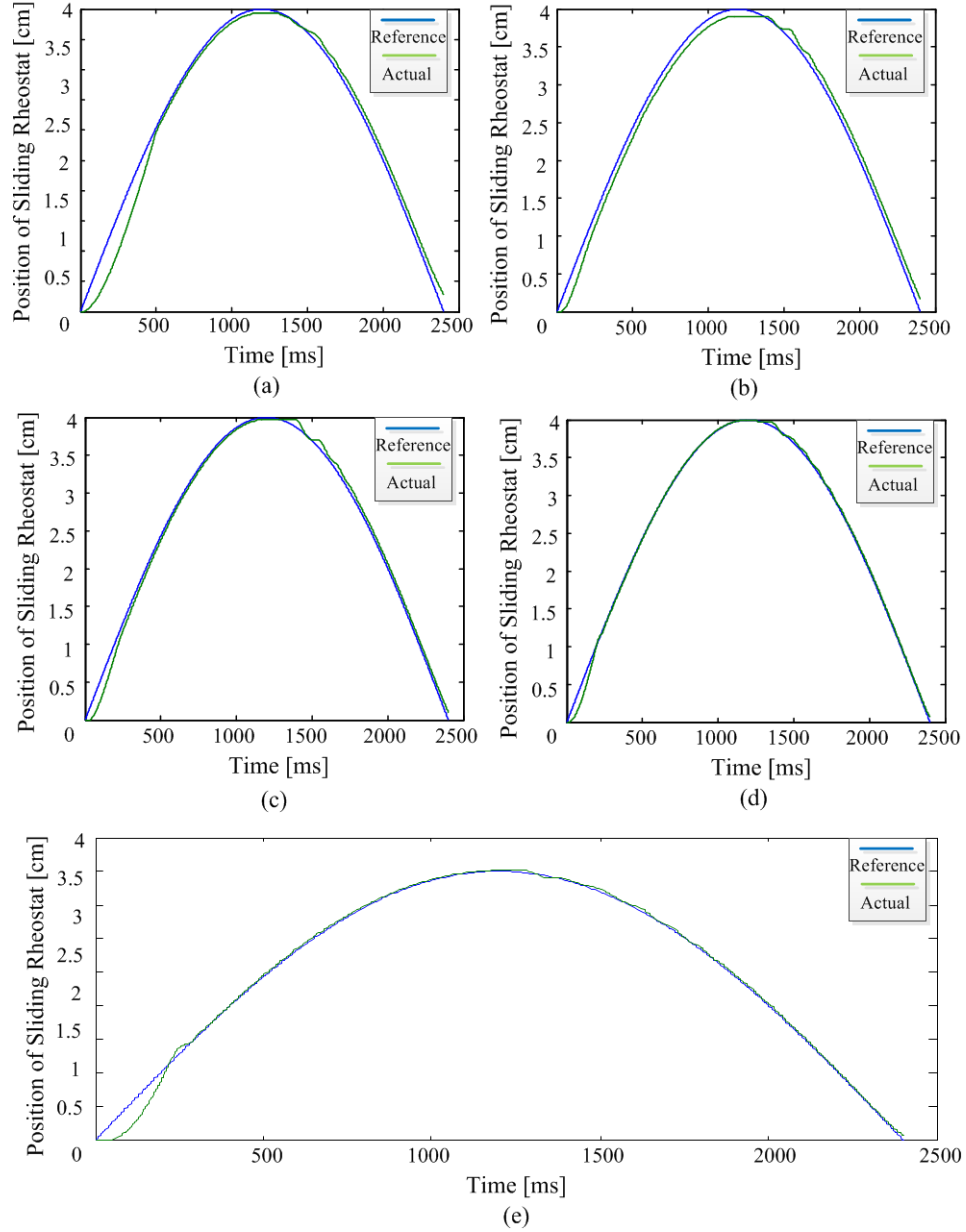
Each subset of the controller provides satisfactory control for a specific operational region of the desired trajectories.

In this study, gain scheduling approach is adopted to improve tracking performance of the twisted-string system. Instead of combining multiple linear controllers, the proposed gain scheduling method switches between the proposed PID and Fuzzy logic controller. The mechanism of gain scheduling used in this system is shown in Figure 4.8.



**Figure 4.8 Mechanism of gain scheduling method based on PID and fuzzy control**

In this control scheme, there are two observable variables, also called as scheduling variables, used to determine the corresponding operational region, which are positions and errors. In a specific operational region, an appropriate controller is adopted with the tracking error and feedback of position sensors. It is clear that the output of this system depends on both PID and fuzzy controllers. For the part of fuzzy controller, both system output and position error are used as input factors to apply fuzzy logic rules. While operating in the linear regions, the gain of output is changed all the time according to the changes of PID and fuzzy logic controllers. Since the output of fuzzy logic controller can affect the change of PID parameters, the scheduled gains of PID controller need to be modified according to the rules of fuzzy logic controller. By carefully testing and adjusting, the rules of fuzzy logic and scheduled gains can be determined. The controlled results of this gain scheduling method are shown below in Figure 4.9.



**Figure 4.9 (a) Controlled result of combining both PID and fuzzy controllers without tuning; (b)-(e) Results of gain scheduling method**

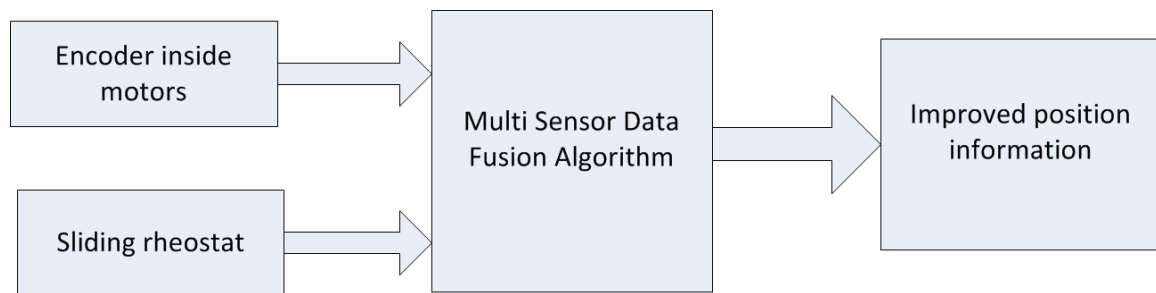
Compare to PID or fuzzy logic controller, the controller combining both methods yields much less tracking error. The results were obtained by adjusting parameters of the new controller using gain scheduling method. The result is illustrated in Figure 4.9. If PID and fuzzy logic controller were combined together without fine tune parameters of individual controllers, the performance shown in Figure 4.9(a) is not satisfactory. It is

therefore necessary to adjust control parameters to fine tune system performance. In particular, the tracking performance is extremely poor in the part of raising movement. The fuzzy logic rules are then adjusted to compensate for the raising movement as shown in Figure 4.9(b). In addition, a constant tracking error can be observed regardless of moving direction. Thus, the  $D$  type gain is increased in the linear PID controller, and the results are shown in Figure 4.9(c). In order to improve tracking performance, several minor adjustments were also modified in the fuzzy logic controller, and the results are shown in Figure 4.9(d) and (e).

From the experimental results, it is clear that PID and fuzzy logic controllers have their own drawbacks either it is difficult to maintain precise tracking or the response of the system is too slow while starting up. By combining the advantages of both controllers with gain scheduling approach, tracking performance can be guaranteed. As the result, high-accuracy tracking and adequate response can both be satisfied.

#### 4.2.4 Results of Sensor Fusion

In this section, a data fusion technique of multiple sensors is integrated into this system. With sensor fusion, data acquired from two sensors were processed together to generate an optimal estimation as the control feedback of this system. In this twisted-string system, the position sensors equipped are an optical encoder and a sliding rheostat. Figure 4.10 illustrates the sensor fusion technique used in this study.



**Figure 4.10 Illustration of multi sensor data fusion in the system**

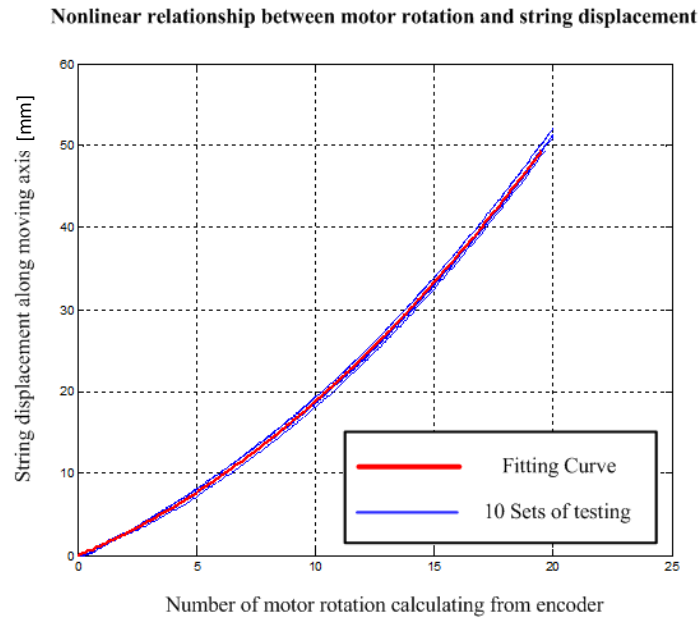
The feedback signal needs to be used is the actual length of the twisted string. A sliding rheostat resistor was used to detect the length change. However, the control output of the motor is related to revolution (or angle) of the shaft and the resolution of the sliding rheostat is also limited. Thus, the encoder inside of the adopted motors can be used as extra position sensor. However, the optical encoder cannot directly measure the actual length of the twisted-string system. The relationship between the numbers of revolutions of the driving motor and the actual displacement of strings needs to be identified. In order to take advantage of encoder, curve fitting based on 10 sets of experimental results has been done and the results of the nonlinear relationship is shown as in Figure 4.11. The function of the fitted curve is:

$$Y_m = 0.023 \times n^2 + n + 0.054 \quad (4.2)$$

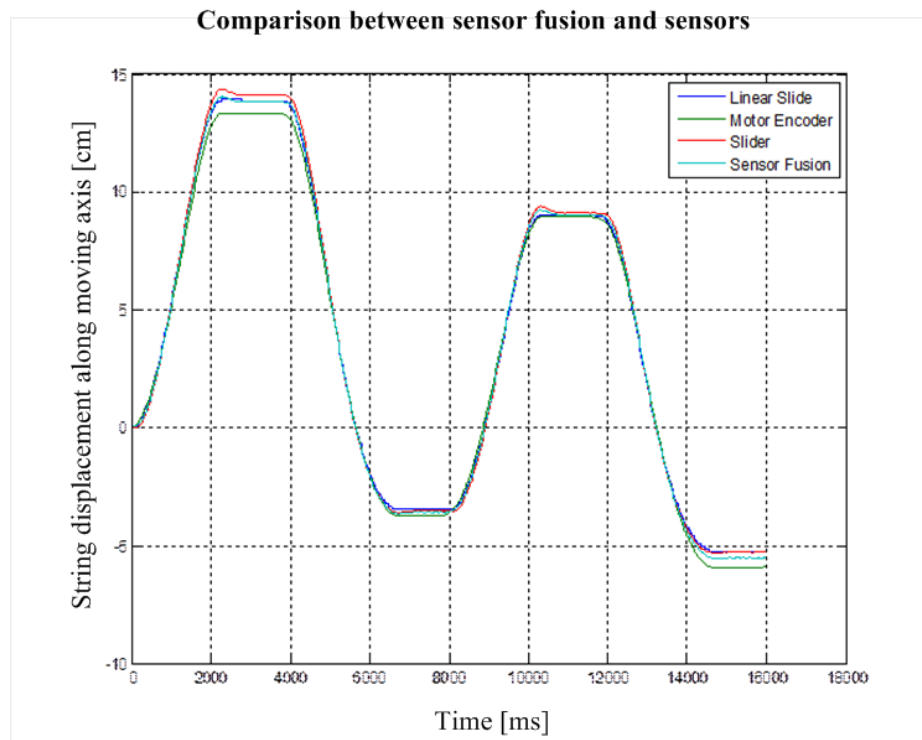
where  $n$  is the number of motor revolution, and  $Y_m$  is the linear displacement of the twisted string. As long as the relationship was determined, the data acquired from encoder can be used to estimate the position of motion. This information will then be fused with the displacement acquired from the sliding rheostat. According to sensor fusion method discussed in chapter 3, the fusion equation can be expressed as

$$Y = \sigma_3^2 (\sigma_1^2 Y_m + \sigma_2^2 Y_s) \quad (4.3)$$

where  $Y_s$  is the data read from sliding potentiometer,  $Y$  is the fused output, and  $\sigma_1, \sigma_2, \sigma_3$  are weight coefficients of the fusion method. In order to verify the performance of sensor fusion method, an accurate linear position sensor was equipped to the system as a reference. The comparison of actual displacement and the estimation of the fused output are shown in Figure 4.12.



**Figure 4.11 Curve fitting between string displacement and motor rotation**



**Figure 4.12 Comparison between sensor fusion result and individual sensors**

To verify the accuracy of the fusion function, the measured outputs of encoder and potentiometer, the fused output, and the reference measurement are plotted in Figure 4.12. As shown in this figure, the calculated results from the fusion equation are closer to reference values. This observation can also be validated by calculating the root mean square deviation (RMSD). The results are shown in following table 4.4.

**Table 4.4 Comparison of Root-Mean-Square-Deviation**

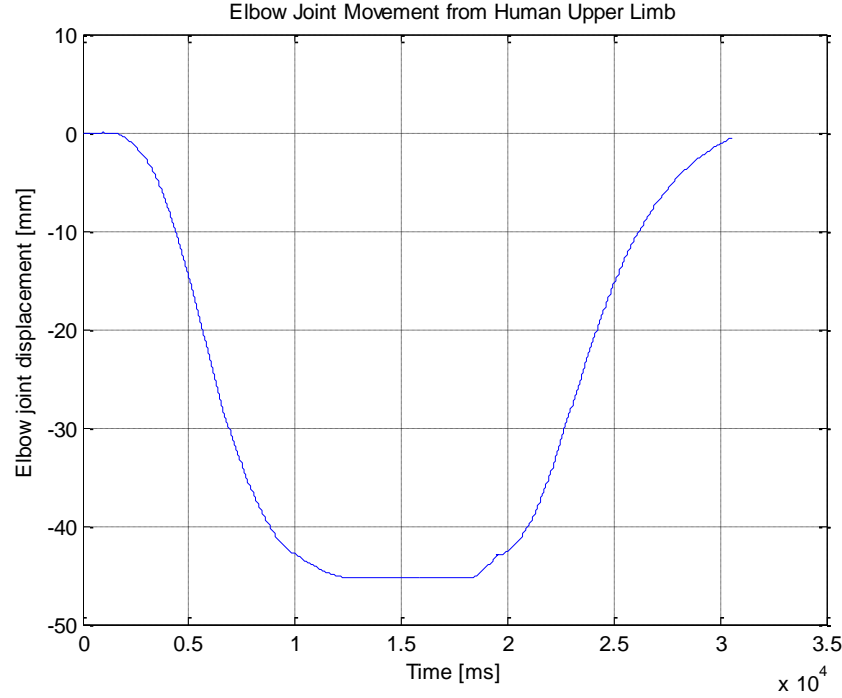
Sensors or method	RMSD
Motor encoder	0.3555
Sliding potentiometer	0.2015
Sensor data fusion	0.1358

The least RMSD value is the fused output, which means the fitting/tracking function can provide an accurate position with the measurements of two adopted sensors. This fusion function will be used in the controller.

### 4.3 Trajectory Tracking Results of Two Axial Movement

From the results in the previous sections, a controller that drives two twisted-string actuators will be synthesized to control the angular movement of the elbow joint in the robotic mechanism. The dual-axial movements are compensated by a master-slave control scheme. In order to verify the applicability of the synthesized controller, a real human motion of upper limb was recorded by Kinect and the unit of the moving trajectories was mapped from angular displacements of joints to linear displacements of twisted strings. In this experiment, only the movement of elbow joint was used to test the master-slave control scheme. The displacements of elbow joint are shown in Figure 4.13.

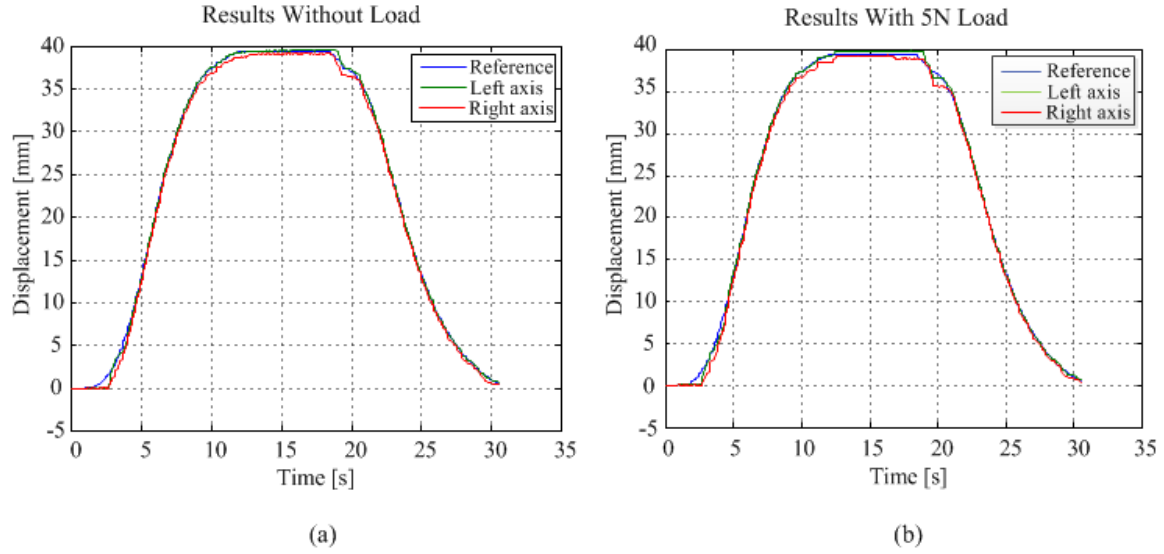




**Figure 4.13 Real elbow motion trajectory from human objective**

The master-slave control scheme is shown as Figure 3.12. In this scheme, the slave actuator needs to follow the trajectory of the master actuator. Gain scheduling method was applied to both master and slave sides. To ensure that the master-slave control scheme with gain scheduling can have adequate tracking and synchronization performance, two experiments, without external load and with a 5 N external load, were conducted. The initial positions of twisted strings on both sides are pre-twisted for 12 revolutions. Each experiment were executed for 10 times to verify the repeatability.

For the free load test, the maximum tracking error among the 10 tests of results is about 1.6 mm and the average error is 0.30 mm, which is relatively small with respect to the total stroke of human elbow movement. Figure 4.14 demonstrates the controlled results of both actuators (left and right of the pulley). If a 5 N load is attached to the pulley, the maximum tracking error among 10 tests is about 1.73 mm and the average error is 0.34 mm. The root-mean-square errors of these two tests are 0.3499 mm and 0.6051 mm, respectively. It is clear that the proposed gain scheduling controller can have great tracking performance individually.



**Figure 4.14 Controlled results of the dual twisted-string actuator (a) free load; (b) 5 N external load.**

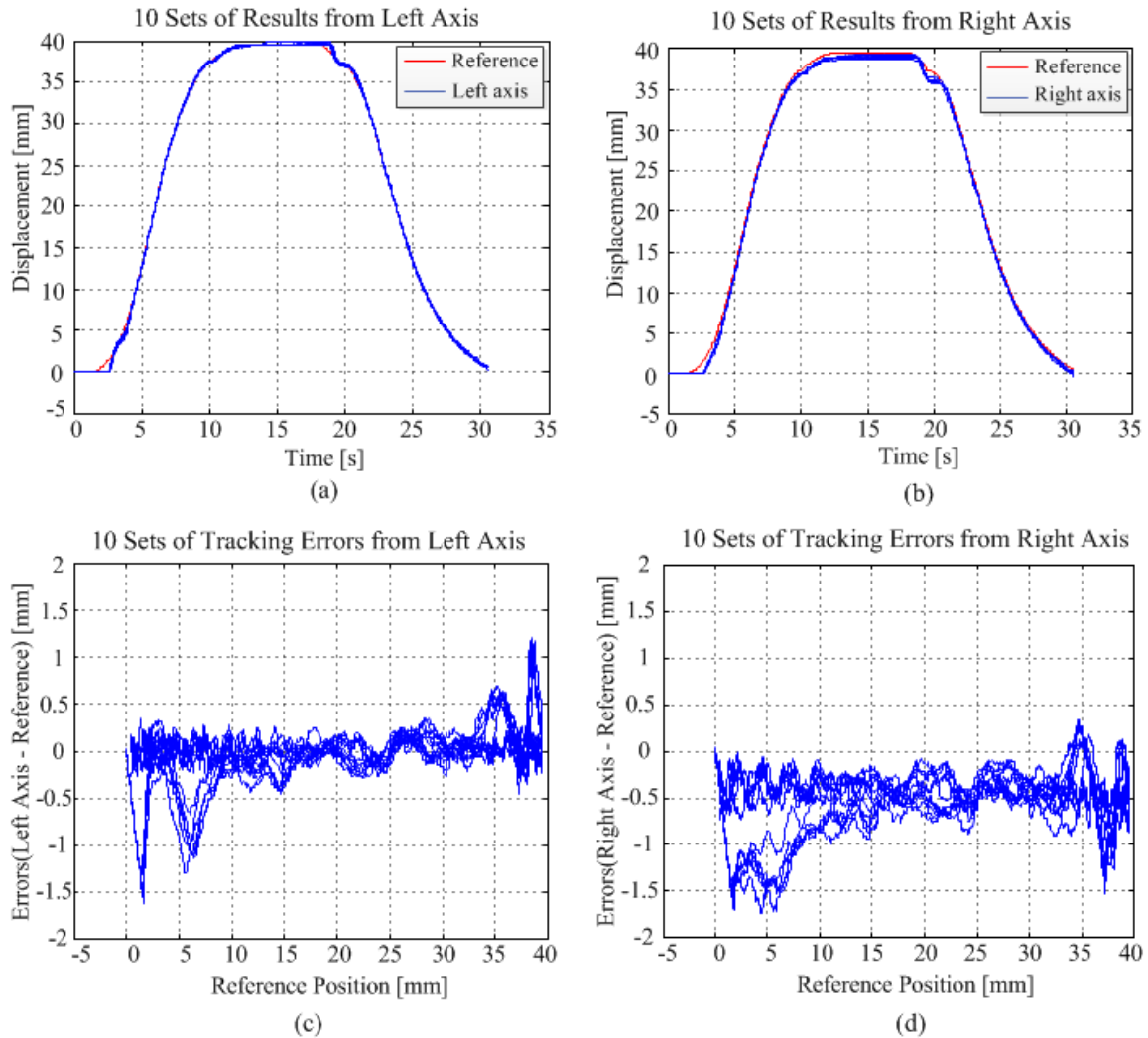
In addition to tracking performance, there are two properties, repeatability and synchronization, are used to evaluate the master-slave control scheme. Figure 4.15 illustrates the tracking of 10 tests without any load. The tracking errors of both actuators can be restricted and the maximum errors of both actuators are less than 2 mm.

With a 5 N external load, the maximum tracking errors are also less than 2 mm though the perturbations are slightly more than the scenario without any load as shown in Figure 4.16. Thus, it can be concluded that the proposed master-slave gain scheduling control scheme demonstrates an excellent repeatability with and without the 5 N external load as well.

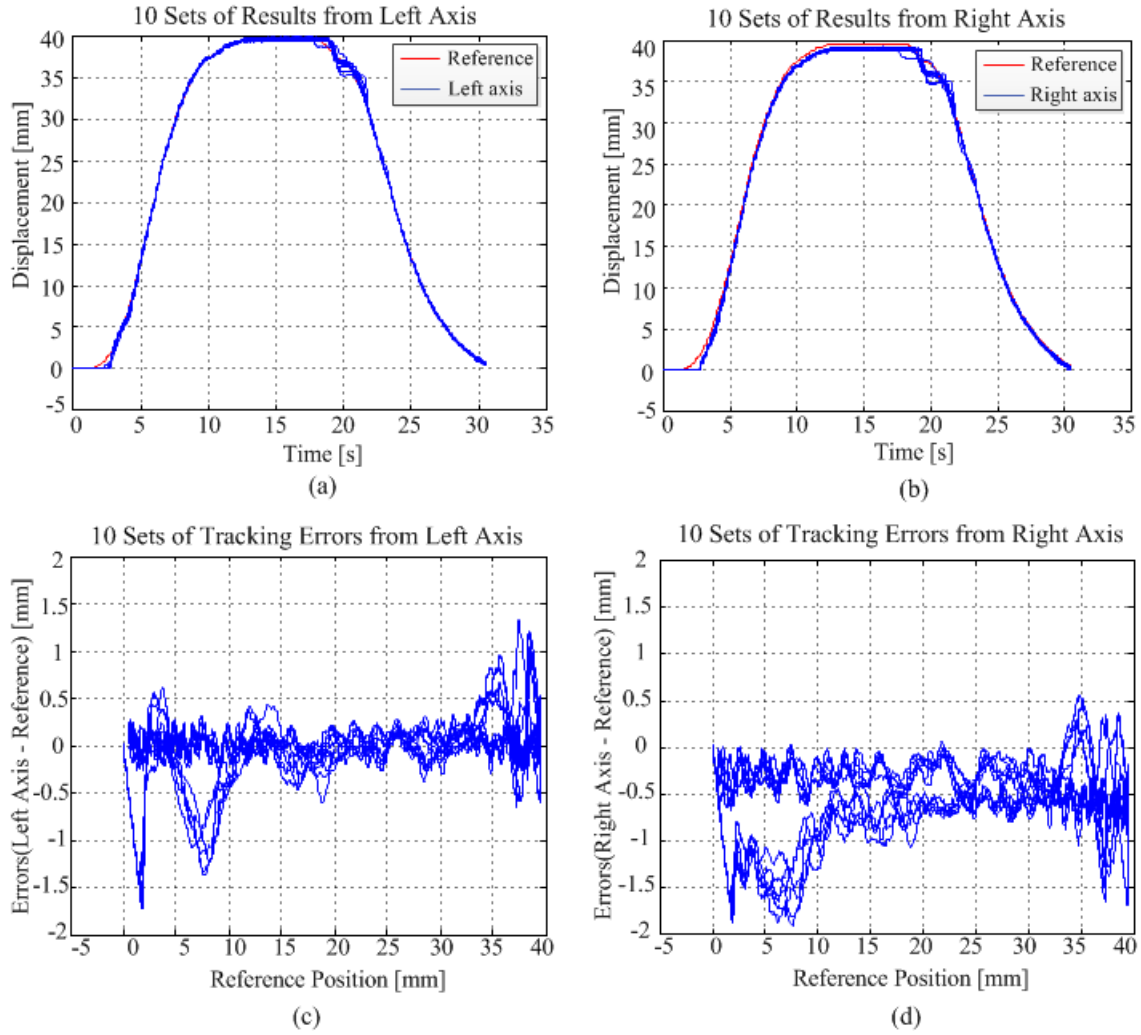
As to synchronization, Figure 4.17 shows the synchronization error between two axes, and Table 4.5 lists the tracking errors and synchronization errors of the two scenarios. The synchronization errors are less than 2 mm for the whole operation with or without a load.

**Table 4.5 Performance of the master-slave control scheme**

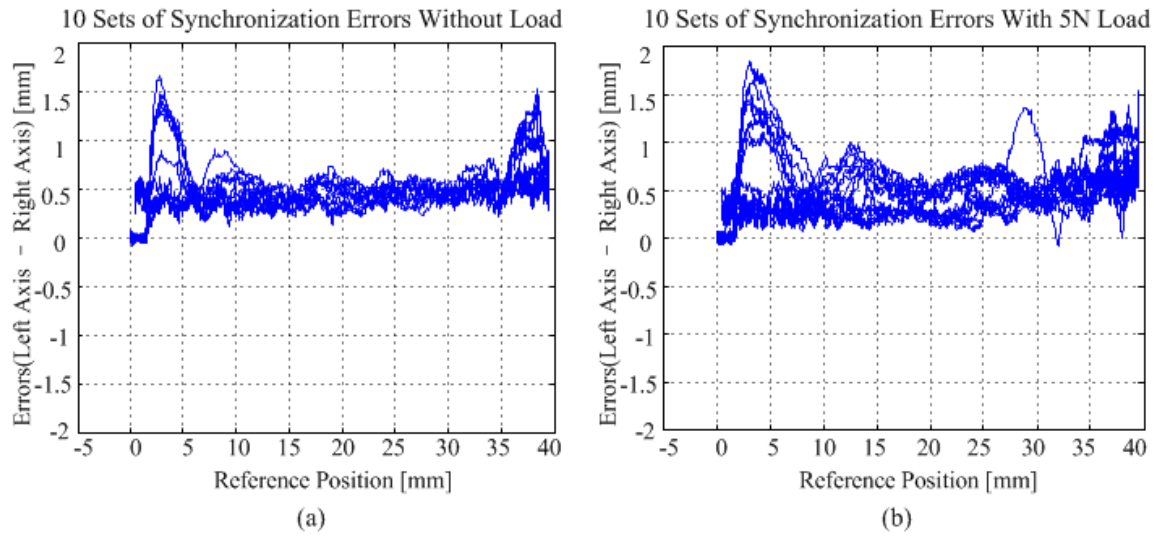
	Left Actuator		Right Actuator		Synchronization Max. error (mm)
	Max. tracking error (mm)	RMSE (mm)	Max. tracking error (mm)	RMSE (mm)	
Load free	0.4102	0.2630	0.6018	0.5536	0.5015
5N load	0.9854	0.3499	1.1168	0.6051	0.5127



**Figure 4.15 Repeatability test – 10 experiments without any load (a) left actuator; and (b) right actuator; (c) tracking errors of left actuator; and (d) tracking errors of right actuator.**



**Figure 4.16 Repeatability test – 10 experiments with a 5 N load (a) left actuator; and (b) right actuator; (c) tracking errors of left actuator; and (d) tracking errors of right actuator.**



**Figure 4.17 Synchronization test: (a) synchronization error between the two actuators without any load, and (b) synchronization error between the two actuators with a 5 N load.**

## CHAPTER 5

### CONCLUSION

#### 5.1 Summary of Research

The objective of this research was to propose an integrated controller that can properly compensate for the movements of an assistive robotic device between adopted actuators. This robotic device aims to compensate for the movement of upper limb used for rehabilitation purpose. A dual twisted-string actuator has been used to drive the elbow joint of this device. To ensure accurate movement of the joint, the controller was designed based on this experimental platform for both tracking and synchronization. Based on traditional control theory, a controller integrates simple linear control techniques has been proposed. In order to mimic certain human motions, several nonlinear factors were considered to fulfill the required performance as well.

The tasks of this research was broken down into two major parts:

1) Design of hardware platform and adequate controller. Based on analysis, twisted-string actuator system was fabricated as the carrier of the movement because of its advantages of simple structure and portability. A 2-DOF prototype of assistive robotic system with elbow and shoulder joints were developed in the lab. Unlike other exoskeleton devices, the actuators were placed on the patient's torso and the force was transmitted through twisted-strings and Bowden cable mechanism to the joints. However, both of the internal and external nonlinear factors of twisted-string actuators make the system difficult to control. In this study, a master-slave controller integrating PID, fuzzy logic and gain scheduling methods has been implemented on a fabricated device. To apply the designed control algorithm to the device, MATLAB xPC Target real-time control toolbox was used to validate the design. In addition to the host controller, few embedded controllers were integrated in the hardware of the controller, including an Intel's Galileo Gen 2 board and a Xilinx Spartan FPGA board.

2) Experiments and analysis of results. Based on the trajectory of a real human motion, the joint movements were collected from a drinking motion. Control performances of different control algorithms, PID controller, fuzzy logic controller, and gain scheduling controller, were compared and analyzed. Gain scheduling controller was adopted to control the elbow joint for individual twisted-string actuators. Sensor data fusion and master-slave control method. The proposed controller demonstrates an excellent performance. Both tracking and synchronization errors can be greatly improved.

The results of this study demonstrate a feasible framework to control a multi-axial mechanism. The errors due to nonlinear factors can be adequately compensated, and both tracking and synchronization can be guaranteed.

## **5.2 Future Works**

Though the proposed frameworks demonstrate great outcome, few additional works have to be developed.

Though gain scheduling method shows great controlled outcome, it still possibly fails while severe nonlinear perturbations affect the system. Thus, more advanced control algorithms that can handle varying scenarios, such as dead zone and varying loading, is required. In addition, only one trajectory was used to test the proposed controller. To validate the effectiveness of developed controllers, it is necessary to test different Activities of Daily Life (or ADLs). More trajectories of ADLs of healthy adults will be acquired for both control and rehabilitation purposes in the future. Other than control algorithms, the hardware of the wearable device needs to be improved greatly. More degree-of-freedom will be added to the existing mechanism. The mechanism of such a robotic device is under development. A prototype setup will be transplanted to a wheelchair, which can have better flexibility for patients performing rehabilitation activities.

A successfully developed device of such a robotic system is anticipated to reduce the cost of homecare. In addition to the development of hardware and software, the adopted trajectories need to satisfy the demands of most users. This requires to collect data from

groups with various body conditions, including gender, age, height, and weight, etc. Mathematic models will be necessary to generate corresponding trajectories based on patient's/user's body condition. Once the integrated robotic device is built, it is also necessary to have feedbacks from users and patients. In particular, patients with different diseases or injuries can have different responses to wearable devices. These responses can be a totally different type of disturbance to the robotic device. Control algorithms compensating for these disturbance can make an assistive robot a more convenient tool in daily lives.



## REFERENCES

- [1] Powered exoskeleton. (2016, April 7). In Wikipedia, The Free Encyclopedia. Retrieved 01:15, April 11, 2016, from  
*[https://en.wikipedia.org/w/index.php?title=Powered\\_exoskeleton&oldid=714135406](https://en.wikipedia.org/w/index.php?title=Powered_exoskeleton&oldid=714135406)*.
- [2] Mavroidis, C., Nikitzuk, J., Weinberg, B., Danaher, G., Jensen, K., Pelletier, P., Prugnarola, J., Stuart, R., Arango, R., Leahey, M. and Pavone, R., 2005. Smart portable rehabilitation devices. *Journal of NeuroEngineering and Rehabilitation*, 2(1), p.1.
- [3] “Twisted String Actuator” [Online]. Retrieved from  
*<http://www.lat.uni-saarland.de/index.php/en/research/twisted-string-actuator>*.
- [4] Chuang, H.Y. and Liu, C.H., 1991. Cross-coupled adaptive feedrate control for multiaxis machine tools. *Journal of dynamic systems, measurement, and control*, 113(3), pp.451-457.
- [5] Unimate. (2015, December 7). In Wikipedia, The Free Encyclopedia. Retrieved 19:23, January 13, 2016, from  
*<https://en.wikipedia.org/w/index.php?title=Unimate&oldid=694185678>*.
- [6] Bin, H.Z., Yamazaki, K., DeVries, M.F. and Bollinger, J.G., 1983. A microprocessor-based control scheme for the improvement of contouring accuracy. *CIRP Annals-Manufacturing Technology*, 32(1), pp.275-279.
- [7] Mechanism (engineering). (2016, March 20). In Wikipedia, The Free Encyclopedia. Retrieved 17:47, April 1, 2016, from  
*[https://en.wikipedia.org/w/index.php?title=Mechanism\\_\(engineering\)&oldid=71092163](https://en.wikipedia.org/w/index.php?title=Mechanism_(engineering)&oldid=71092163)*.
- [8] Koren, Y., 1980. Cross-coupled biaxial computer control for manufacturing systems. *Journal of Dynamic Systems, Measurement, and Control*, 102(4), pp.265-272.
- [9] Xiao, Y., Zhu, K. and Liaw, H.C., 2005. Generalized synchronization control of multi-axis motion systems. *Control Engineering Practice*, 13(7), pp.809-819.
- [10] Jeong, S.K. and You, S.S., 2008. Precise position synchronous control of multi-axis servo system. *Mechatronics*, 18(3), pp.129-140.
- [11] Sun, D., 2003. Position synchronization of multiple motion axes with adaptive coupling control. *Automatica*, 39(6), pp.997-1005.
- [12] Cheng, M.H.M., Mitra, A. and Chen, C.Y., 2009, December. Synchronization controller synthesis of multi-axis motion system. In *Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on* (pp. 918-921). IEEE.

- [13] Cheng, M.H. and Bakhoun, E.G., 2011, June. Adaptive robust control of tracking and synchronization for multi-axis motion system. In *American Control Conference (ACC), 2011* (pp. 1-6). IEEE.
- [14] Wearable technology. (2016, January 10). In Wikipedia, The Free Encyclopedia. Retrieved 21:03, January 13, 2016, from [https://en.wikipedia.org/w/index.php?title=Wearable\\_technology&oldid=699169823](https://en.wikipedia.org/w/index.php?title=Wearable_technology&oldid=699169823).
- [15] Sports biomechanics. (2015, November 15). In Wikipedia, The Free Encyclopedia. Retrieved 17:28, January 14, 2016, from [https://en.wikipedia.org/w/index.php?title=Sports\\_biomechanics&oldid=690730115](https://en.wikipedia.org/w/index.php?title=Sports_biomechanics&oldid=690730115).
- [16] Gopura, R.A.R.C. and Kiguchi, K., 2009, June. Mechanical designs of active upper-limb exoskeleton robots: State-of-the-art and design difficulties. In *Rehabilitation Robotics, 2009. ICORR 2009. IEEE International Conference on* (pp. 178-187). IEEE.
- [17] Srinivasan, K. and Fosdick, R., 1988, June. Multivariable analysis and controller design for coordinated multi-axial motion control. In *American Control Conference, 1988* (pp. 95-101). IEEE.
- [18] Moore, P.R. and Chen, C.M., 1995. Fuzzy logic coupling and synchronised control of multiple independent servo-drives. *Control Engineering Practice*, 3(12), pp.1697-1708.
- [19] Rahman, M., Heikkala, J. and Lappalainen, K., 2000. Modeling, measurement and error compensation of multi-axis machine tools. Part I: theory. *International Journal of Machine Tools and Manufacture*, 40(10), pp.1535-1546.
- [20] Kim, K. and Kim, M.K., 1991. Volumetric accuracy analysis based on generalized geometric error model in multi-axis machine tools. *Mechanism and machine theory*, 26(2), pp.207-219.
- [21] Suykens, J.A., Curran, P.F. and Chua, L.O., 1999. Robust synthesis for master-slave synchronization of Lur'e systems. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 46(7), pp.841-850.
- [22] Curran, P.F. and Chua, L.O., 1997. Absolute stability theory and the synchronization problem. *International Journal of Bifurcation and Chaos*, 7(06), pp.1375-1382.
- [23] Hasler, M., 1994. Synchronization principles and applications. *Circuits and systems tutorials*, pp.314-327.
- [24] Suykens, J.A.K. and Vandewalle, J., 1997. Master-slave synchronization of Lur'e systems. *International journal of bifurcation and chaos*, 7(03), pp.665-669.

- [25] Suykens, J.A.K., Curran, P.F. and Chua, L.O., 1997. Master-slave synchronization using dynamic output feedback. *International Journal of Bifurcation and Chaos*, 7(03), pp.671-679.
- [26] Jiang, L., Shisheie, R., Cheng, M.H., Banta, L.E. and Guo, G., 2013, August. Moving trajectories and controller synthesis for an assistive device for arm rehabilitation. In *Automation Science and Engineering (CASE), 2013 IEEE International Conference on* (pp. 268-273). IEEE.
- [27] Sarachik, P. and Ragazzini, J.R., 1957. A 2-dimensional feedback control system. *American Institute of Electrical Engineers, Part II: Applications and Industry, Transactions of the*, 76(2), pp.55-61.
- [28] Koren, Y., 1980. Cross-coupled biaxial computer control for manufacturing systems. *Journal of Dynamic Systems, Measurement, and Control*, 102(4), pp.265-272.
- [29] Yeh, S.S. and Hsu, P.L., 1999. Theory and applications of the robust cross-coupled control design. *Journal of dynamic systems, measurement, and control*, 121(3), pp.524-530.
- [30] Chin, J.H. and Lin, T.C., 1997. Cross-coupled precompensation method for the contouring accuracy of computer numerically controlled machine tools. *International Journal of Machine Tools and Manufacture*, 37(7), pp.947-967.
- [31] Kulkarni, P.K. and Srinivasan, K., 1989. Optimal contouring control of multi-axial feed drive servomechanisms. *ASME Journal of Engineering for Industry*, 111(2), pp.140-148.
- [32] Srinivasan, K. and Kulkarni, P.K., 1990. Cross-coupled control of biaxial feed drive servomechanisms. *Journal of dynamic systems, measurement, and control*, 112(2), pp.225-232.
- [33] Chiu, G.T.C. and Tomizuka, M., 1998. Coordinated position control of multi-axis mechanical systems. *Journal of dynamic systems, measurement, and control*, 120(3), pp.389-393.
- [34] Krebs, H.I., Volpe, B.T., Williams, D., Celestino, J., Charles, S.K., Lynch, D. and Hogan, N., 2007. Robot-aided neurorehabilitation: a robot for wrist rehabilitation. *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, 15(3), p.327.
- [35] Mihelj, M., Nef, T. and Riener, R., 2006, February. ARMin-toward a six DoF upper limb rehabilitation robot. In *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. the First IEEE/RAS-EMBS International Conference on*(pp. 1154-1159). IEEE.

- [36] Aisen, M.L., Krebs, H.I., Hogan, N., McDowell, F. and Volpe, B.T., 1997. The effect of robot-assisted therapy and rehabilitative training on motor recovery following stroke. *Archives of neurology*, 54(4), p.443.
- [37] Nef, T., Mihelj, M., Colombo, G. and Riener, R., 2006, May. ARMin-robot for rehabilitation of the upper extremities. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on* (pp. 3152-3157). IEEE.
- [38] Da Vinci Surgical System. (2016, January 28). In Wikipedia, The Free Encyclopedia. Retrieved 04:17, February 9, 2016, from [https://en.wikipedia.org/w/index.php?title=Da\\_Vinci\\_Surgical\\_System&oldid=70047519](https://en.wikipedia.org/w/index.php?title=Da_Vinci_Surgical_System&oldid=70047519).
- [39] Brass knuckles. (2016, February 12). In Wikipedia, The Free Encyclopedia. Retrieved 17:34, February 14, 2016, from [https://en.wikipedia.org/w/index.php?title=Brass\\_knuckles&oldid=704629213](https://en.wikipedia.org/w/index.php?title=Brass_knuckles&oldid=704629213).
- [40] Helmet-mounted display. (2016, January 24). In Wikipedia, The Free Encyclopedia. Retrieved 18:48, February 14, 2016, from [https://en.wikipedia.org/w/index.php?title=Helmet-mounted\\_display=702252](https://en.wikipedia.org/w/index.php?title=Helmet-mounted_display=702252).
- [41] TALOS (uniform). (2016, January 26). In Wikipedia, The Free Encyclopedia. Retrieved 19:01, February 14, 2016, from [https://en.wikipedia.org/w/index.php?title=TALOS\\_\(uniform\)&oldid=701708698](https://en.wikipedia.org/w/index.php?title=TALOS_(uniform)&oldid=701708698).
- [42] Virtuix Omni. (2016, February 1). In Wikipedia, The Free Encyclopedia. Retrieved 20:29, February 14, 2016, from [https://en.wikipedia.org/w/index.php?title=Virtuix\\_Omni&oldid=702834835](https://en.wikipedia.org/w/index.php?title=Virtuix_Omni&oldid=702834835).
- [43] Windows Holographic. (2016, January 28). In Wikipedia, The Free Encyclopedia. Retrieved 20:41, February 14, 2016, from [https://en.wikipedia.org/w/index.php?title=Windows\\_Holographic&oldid=7029553](https://en.wikipedia.org/w/index.php?title=Windows_Holographic&oldid=7029553).
- [44] Haptic suit. (2016, January 12). In Wikipedia, The Free Encyclopedia. Retrieved 20:34, February 14, 2016, from [https://en.wikipedia.org/w/index.php?title=Haptic\\_suit&oldid=699503743](https://en.wikipedia.org/w/index.php?title=Haptic_suit&oldid=699503743).
- [45] Wurtz, T., May, C., Holz, B, et al. The twisted string actuation system: Modeling and control[C]. In *Advanced Intelligent Mechatronics(AIM), 2010 IEEE/ASME International Conference*, 2010:1215-1220.
- [46] Popov, D., Gaponov, I. and Ryu, J.H., 2012, October. A study on twisted string actuation systems: Mathematical model and its experimental evaluation. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on* (pp. 1245-1250). IEEE.

- [47] Simulink. (2016, January 13). In Wikipedia, The Free Encyclopedia. Retrieved 19:11, February 27, 2016, from  
*<https://en.wikipedia.org/w/index.php?title=Simulink&oldid=699674725>*.
- [48] Intel Galileo. (2016, January 22). In Wikipedia, The Free Encyclopedia. Retrieved 19:13, February 27, 2016, from  
*[https://en.wikipedia.org/w/index.php?title=Intel\\_Galileo&oldid=701136903](https://en.wikipedia.org/w/index.php?title=Intel_Galileo&oldid=701136903)*.
- [49] Nyquist frequency. (2015, April 6). In Wikipedia, The Free Encyclopedia. Retrieved 16:37, February 29, 2016, from  
*[https://en.wikipedia.org/w/index.php?title=Nyquist\\_frequency&oldid=655236469](https://en.wikipedia.org/w/index.php?title=Nyquist_frequency&oldid=655236469)*.
- [50] Arduino. (2016, February 28). In Wikipedia, The Free Encyclopedia. Retrieved 21:28, February 29, 2016, from  
*<https://en.wikipedia.org/w/index.php?title=Arduino&oldid=707310039>*.
- [51] PID controller. (2016, January 22). In Wikipedia, The Free Encyclopedia. Retrieved 17:13, February 16, 2016, from  
*[https://en.wikipedia.org/w/index.php?title=PID\\_controller&oldid=701134374](https://en.wikipedia.org/w/index.php?title=PID_controller&oldid=701134374)*.
- [52] Tanaka, K. and Sugeno, M., 1992. Stability analysis and design of fuzzy control systems. *Fuzzy sets and systems*, 45(2), pp.135-156.
- [53] Bayes' theorem. (2016, February 29). In Wikipedia, The Free Encyclopedia. Retrieved 22:08, March 2, 2016, from  
*[https://en.wikipedia.org/w/index.php?title=Bayes%27\\_theorem&oldid=707492466](https://en.wikipedia.org/w/index.php?title=Bayes%27_theorem&oldid=707492466)*.
- [54] Hugh, D.W. and Henderson, T.C., 2001. Multi Sensor Data Fusion. *University of Sydney*.
- [55] Huang, H.Z., Zuo, M.J. and Sun, Z.Q., 2006. Bayesian reliability analysis for fuzzy lifetime data. *Fuzzy Sets and Systems*, 157(12), pp.1674-1686.
- [56] Kinect. (2016, January 31). In Wikipedia, The Free Encyclopedia. Retrieved 19:05, February 15, 2016, from  
*<https://en.wikipedia.org/w/index.php?title=Kinect&oldid=702574374>*.
- [57] Shahrokhi, M. and Zomorodi, A., 2013. Comparison of PID controller tuning methods. *Department of Chemical & Petroleum Engineering Sharif University of Technology*.
- [58] Fuzzy control system. (2016, February 10). In Wikipedia, The Free Encyclopedia. Retrieved 19:30, February 16, 2016, from  
*[https://en.wikipedia.org/w/index.php?title=Fuzzy\\_control\\_system&oldid=704326988](https://en.wikipedia.org/w/index.php?title=Fuzzy_control_system&oldid=704326988)*.

## APPENDIX A Arduino Code

```
#include <TimerOne.h>
#include <SPI.h>          // needed for Arduino versions later than 0018
#include <Ethernet.h>
#include <EthernetUdp.h> // UDP library from: bjoern@cs.stanford.edu 12/30/2008
// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168,1,177);
IPAddress host(192, 168,1,43);
int i=0;
unsigned int localPort = 22222;          // local port to listen on
// buffers for receiving and sending data
char packetBuffer[UDP_TX_PACKET_MAX_SIZE]; //buffer to hold incoming packet,
char ReplyBuffer[] = "acknowledged";      // a string to send back
byte encoder[2];
byte Voltage[2];
byte current[2];
byte udp_data[6];
// An EthernetUDP instance to let us send and receive packets over UDP
EthernetUDP Udp;
int j=0;
int time_flag=0;
int motor1=0;
int motor2=1;
int motor3=2;
int motor4=3;
byte state=0;
byte state1=0;
byte state2=0;
unsigned long time1=0;
unsigned long time2;
unsigned long span;
int current_sensor;
void setup() {
  Timer1.initialize(4000); // set a timer of length 100000 microseconds (or 0.1 sec - or 10Hz => the
  led will blink 5 times, 5 cycles of on-and-off, per second)
  Timer1.attachInterrupt( timerIsr ); // attach the service routine here
  // start the Ethernet and UDP:
  Ethernet.begin(mac,ip);
  Udp.begin(localPort);
  Serial.begin(9600);
  pinMode(motor1,OUTPUT);
  pinMode(motor2,OUTPUT);
  pinMode(motor3,OUTPUT);
  pinMode(motor4,OUTPUT);
```

```

pinMode(6,OUTPUT);
SPI.begin();
analogReadResolution(12);
}
void loop() {
  analogWrite(6,128);
  // if there's data available, read a packet
  current_sensor=0;
  for (i=0;i<14;i++)
  {
    current_sensor = current_sensor+analogRead(A0);
  }
  current_sensor=current_sensor/14;
  //Serial.println(current_sensor);
  Serial.println((2052-current_sensor)*75.76/4095);
  current[0] = current_sensor&0x00ff;
  current[1] = current_sensor>>8;
  Voltage[0] = 0;
  Voltage[1] = 0;
  digitalWrite(SS, LOW);
  delayMicroseconds(100);
  int packetSize = Udp.parsePacket();
  encoder[0]=SPI.transfer(255);
  encoder[1]=SPI.transfer(255);
  //Serial.println(encoder[0]);
  //Serial.println(encoder[1]);
  digitalWrite(SS,HIGH);
  udp_data[0]=encoder[0];
  udp_data[1]=encoder[1];
  udp_data[2]=Voltage[0];
  udp_data[3]=Voltage[1];
  udp_data[4]=current[0];
  udp_data[5]=current[1];
  Udp.beginPacket(host,localPort);
  Udp.write(udp_data,6);
  Udp.endPacket();
  delayMicroseconds(100);
  if(packetSize)
  {
    // read the packet into packetBuffer
    Udp.read(packetBuffer,UDP_TX_PACKET_MAX_SIZE);
    // send the packet(pwm signal) to the fpga
    digitalWrite(SS, LOW);
    delayMicroseconds(100);
    SPI.transfer(packetBuffer[0]);
    SPI.transfer(packetBuffer[1]);
    digitalWrite(SS,HIGH);
    state=packetBuffer[2];
    state1=state>>4;
    state2=(state&0x0F);
  }
}

```

```

//Serial.println(state2);
if (state1==1)
{
    digitalWrite(motor1, HIGH);
    digitalWrite(motor2, LOW);
}
if (state1==2)
{
    digitalWrite(motor1, LOW);
    digitalWrite(motor2, HIGH);
}
if (state2==1)
{
    digitalWrite(motor3, HIGH);
    digitalWrite(motor4, LOW);
}
if (state2==2)
{
    digitalWrite(motor3, LOW);
    digitalWrite(motor4, HIGH);
}
time1=millis();
delayMicroseconds(100);
}
}
void timerIsr()
{
    if(time_flag==0)
    {
        time2=time1;
        time_flag=1;
        return;
    }
    if (time_flag==1)
    {
        if(time2==time1)
        {
            digitalWrite(SS, LOW);
            delayMicroseconds(100);
            SPI.transfer(0);
            SPI.transfer(0);
            digitalWrite(SS,HIGH);
        }
        time_flag=0;
    }
}
}

```