

Graduate Theses, Dissertations, and Problem Reports

2016

# Design and Development of a Twisted String Exoskeleton Robot for the Upper Limb

Lei Jiang

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

#### **Recommended Citation**

Jiang, Lei, "Design and Development of a Twisted String Exoskeleton Robot for the Upper Limb" (2016). *Graduate Theses, Dissertations, and Problem Reports.* 5887. https://researchrepository.wvu.edu/etd/5887

This Dissertation 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 Dissertation 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 Dissertation 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.

## Design and Development of a Twisted String Exoskeleton Robot for the Upper Limb

Lei Jiang

Dissertation 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

> Doctor of Philosophy in Mechanical Engineering

Marvin H. Cheng, Ph.D., Chair Larry. Banta, Ph.D. Powsiri. Klinkhachorn, Ph.D. Yu. Gu, Ph.D. Steven. Wheeler, PhD.

# **Department of Mechanical and Aerospace Engineering**

Morgantown, West Virginia 2016

Keywords: twisted string, exoskeleton robotics, multi-axial, adaptive robust control Copyrights 2016 Lei Jiang

## ABSTRACT

# Design and Development of a Twisted String Exoskeleton Robot for the Upper Limb Lei Jiang

High-intensity and task-specific upper-limb treatment of active, highly repetitive movements are the effective approaches for patients with motor disorders. However, with the severe shortage of medical service in the United States and the fact that post-stroke survivors can continue to incur significant financial costs, patients often choose not to return to the hospital or clinic for complete recovery. Therefore, robot-assisted therapy can be considered as an alternative rehabilitation approach because the similar or better results as the patients who receive intensive conventional therapy offered by professional physicians.

The primary objective of this study was to design and fabricate an effective mobile assistive robotic system that can provide stroke patients shoulder and elbow assistance. To reduce the size of actuators and to minimize the weight that needs to be carried by users, two sets of dual twisted-string actuators, each with 7 strands (1 neutral and 6 effective) were used to extend/contract the adopted strings to drive the rotational movements of shoulder and elbow joints through a Bowden cable mechanism. Furthermore, movements of non-disabled people were captured as templates of training trajectories to provide effective rehabilitation.

The specific aims of this study included the development of a two-degree-of-freedom prototype for the elbow and shoulder joints, an adaptive robust control algorithm with cross-coupling dynamics that can compensate for both nonlinear factors of the system and asynchronization between individual actuators as well as an approach for extracting the reference trajectories for the assistive robotic from non-disabled people based on Microsoft Kinect sensor and Dynamic time warping algorithm. Finally, the data acquisition and control system of the robot was implemented by Intel Galileo and XILINX FPGA embedded system.

#### ACKNOWLEDGEMENTS

I would like to thank my mentor Dr. Marvin Cheng gratefully, who gave me an opportunity to study at WVU, for his patience, generosity, and advice during my education. He undertook the tremendous responsibility of supervising me dissertation writing. Whenever I came up with any questions or problems, he was always there offering help and advice. His broad knowledge and great idea enlightened me a lot in my research work.

I extend my thanks to Dr.Banta, Dr.Wheeler, Dr.Klinkhachorn, and Dr.Gu for their suggestions and being a member of my committee. I would also like to thank Reza Shisheie, who gave me the support to build experimental platform, Qian Mou, who offered me support to accomplish this project. I would also like to thank Dr. Yuejuan Li and Wei Qi for their encouragement and help in my life.

I would like to thank my patients for being there with me no matter what. Without their unwavering support and encouragement, I couldn't finish my study.

I would like to thank my wife for her love, taking care of the family, and encouragement during my education. I would like to thank my son who gave me such a happy time.

I am grateful for the opportunity of studying at WVU.

# TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
TABLE OF CONTENTS	iv
NOMENCLATURE	vii
LIST OF FIGURES	viii
LIST OF TABLES	xiii
CHAPTER 1: INTRODUCTION	1
1.1 Background	1
1.2 Objectives	4
1.3 Dissertation Structure	5
CHAPTER 2: LITERATURE REVIEW	7
2.1 History of stroke neuro-rehabilitation	7
2.2 Robotic assistive therapy in stroke rehabilitation	9
2.2.1 End-effector-type robotic devices	10
2.2.2 Exoskeleton-type robotic devices	12
2.3 Twisted string actuator	15
2.3.1 Applications of the twisted string actuation	15
2.3.2 Mathematical model of the twisted string actuation	17
2.4 Synchronization of multi-axial system	19
2.5 Nonlinear control	21
2.5.1 Feedback linearization	22
2.5.2 Adaptive control	23
2.5.3 Robust control	24
2.5 Home stroke rehabilitation	25
CHAPTER 3: TWISTED STRING ACTUATION	28

3.1 Kinetostatic Model of the Twisted String Actuation	28
3.3 Adaptive neuro fuzzy inference system model	
3.2 Experimental Setup of Twisted String Actuation	37
CHAPTER 4: Design of the exoskeleton rehabilitation robot device	47
4.1 Mechanism design of the exoskeleton rehabilitation robot device	47
4.2 Cross-Coupling Dynamics	52
4.3 Experimental setup	55
4.4 Linear Controllers – PID, PID-CC, LQR	56
4.4.1 PID Controller	58
4.4.2 PID controller with feedback of cross-coupling dynamics (PID-FC)	61
4.4.3 LQR controller with feedback of cross-coupling dynamics (LQR-FC)	63
4.5 Summary	73
CHAPTER 5: ARC control synthesis of the twisted string actuation	74
5.1 Nonlinearity of the Twisted String Actuation	74
5.2 Synthesis of Non-linear Controller –ARC (Displacement control)	75
5.2.1 Dead-Zone Compensation	76
5.2.2 Nonlinearity and Uncertainty structure compensation	80
5.3 Synthesis of Non-linear Controller –ARC (Force control)	87
5.3.1 Adaptive Robust Controller (Force) Design	89
5.3.2 Simulation Results	
5.3.3 Experimental Results	
CHAPTER 6: Derivation of motion trajectories	
6.1 Development of motion derivation system	
6.1.1 Introduction of Kinect	
6.1.2 Angle of shoulder joint	
6.1.3 Angle of elbow joint	
6.2 Accuracy of the Kinect	
6.3 Desired Moving Trajectories	

6.3.1 Normalization of trajectories	110
6.3.2 Extracting trajectory	114
CHAPTER 7: Implementation of the robot	
7.1 Prototype of exoskeleton assistive device overview	120
7.1.1 Structure of the prototype	120
7.2 Sensor measurement fusion	122
7.2.1 Kalman filter	122
7.2.2 Sensor fusion	127
7.3 Implementation of the embedded controller	131
7.3.1 Intel Galileo 2	132
7.3.2 XILINX SPARTAN FPGA	133
7.3.3 System hardware architecture	134
7.3.4 Software structure	136
7.4 Tracking motion trajectories of human upper limb	139
CHAPTER 8: Conclusion	145
8.1 Conclusions	145
8.2 Recommendations	146
8.3 Future works	147
REFERENCES	148

# NOMENCLATURE

ADL	Activity of Daily Life
CVA	Cerebrovascular Accident
ANFIS	Adaptive Neuro Fuzzy Inference System Model
ARC	Adaptive Robust Control
DTW	Dynamic Time Warping
DOF	Degree Of Freedom
FES	Functional Electrical Stimulation
ICT	Intensive Comparison Therapy
PM	Pneumatic Muscle
PWM	Pulse Width Modulation
RAC	<b>Robust Adaptive Control</b>
RMSE	<b>Root Mean Square Error</b>
RUPERT	<b>Robotic Upper Extremity Repetitive Trainer</b>
VE	Virtual Environment
WVU	West Virginia University

# LIST OF FIGURES

Figure 2-1 A stroke-patient received Constraint Induced Movement Therapy8
Figure 2-2 A stroke-patient received therapy with a commercial version of MIT-MANUS at the
Burke Rehabilitation Hospital (White Plains, NY)11
Figure 2-3 An EMUL rehabilitation system used fot therapy
Figure 2-4 The current version of RUPERT IV system13
Figure 2-5 A CADEN-7 Exoskeleton robotic system15
Figure 2-6 DEXMART hand16
Figure 2-7 Configuration of a multi-fingered robotic hand proposed by Sonoda et al17
Figure 2-8 Aukes' finger robot17
Figure 2-9 Increase of string's radius after being twisted
Figure 2-10 Tandem control20
Figure 2-11 Master-slave control20
Figure 2-12 Mori's control system to synchronize left and right spindles
Figure 2-13 Feedback Linearization
Figure 2-14 Schematic of the Holden's VE tele-rehabilitation system
Figure 3-1 Concept and configuration of a twisted string actuation system
Figure 3-2 Schematic representation of twisted string and forces
Figure 3-3 A two output ANFIS structure
Figure 3-4 Architecture of a five layer ANFIS
Figure 3-5 Actual structure of ANFIS
Figure 3-6 Comparison between ANFIS model and Experimental results
Figure 3-7 Configuration of experimental setup of twisted string actuation
Figure 3-8 Strand holder mounted on DC motor
Figure 3-9 Relationship between motor revolutions and stroke
Figure 3-10 Relationship between motor revolutions and Force40
Figure 3-11 Relationship between motor revolutions and Force transmission42
Figure 3-12 Comparison between extension and contraction of twisted string

Figure 3-13 Relationship among motor revolutions, Force and Stroke
Figure 3-14 Relationship among force transmission ratio and linear displacements after
pre-twisting44
Figure 3-15 Approximated static characteristic of the twisted string actuator
Figure 3-16 Experimental results and curve fitting of force transmission function45
Figure 4-1 Preliminary design of the assistive robotic device
Figure 4-2 Schematic of upper limb rehabilitation exoskeleton robot49
Figure 4-3 Schematic of adjustable telescopic structure of upper limb rehabilitation exoskeleton
robot49
Figure 4-4 Backpack of exoskeleton robot50
Figure 4-5 Backpack on the stroke patient
Figure 4-6 Exoskeleton rehabilitation robot device on the stroke patient
Figure 4-7 Configuration of a joint of a dual twisted-string actuation system in a robotic system.
Figure 4-8 Three scenarios of synchronization between extending and contracting strings52
Figure 4-9 Scheme of tandem control53
Figure 4-10 Scheme of master-slave control54
Figure 4-11 Scheme of cross-coupling control54
Figure 4-12 Schematic of experimental setup (single joint)
Figure 4-13 Configuration of experimental setup (single joint)56
Figure 4-14 Block diagram of the DC motors used to drive the twisted strings
Figure 4-15 Experimental outcome of PID controller (sinusoidal motion): (a) comparison
between reference trajectory and actual trajectories of left actuator; (b) comparison
between reference trajectory and actual trajectories of right actuator; (c) synchronization
errors between left and right twisted actuators60
Figure 4-16 Experimental outcome of PID controller (trapezoidal motion): (a) comparison
between reference trajectory and actual trajectories of left actuator; (b) comparison
between reference trajectory and actual trajectories of right actuator; (c) synchronization
errors between left and right twisted actuators60

- Figure 5-2 Experimental outcome of ARC controller (sinusoidal motion): (a) comparison between reference trajectory and actual trajectories of left actuator; (b) comparison

between reference trajectory and actual trajectories of right actuator; (c) synchronization
errors between left and right actuators83
Figure 5-3 Experimental outcome of LQR-FC controller (trapezoidal motion): (a) comparison
between reference trajectory and actual trajectories of left actuator; (b) comparison
between reference trajectory and actual trajectories of right actuator; (c) synchronization
errors between left and right actuators84
Figure 5-4 (a) Control effort of twisted string actuator (sine); (b) Control effort of twisted string
actuator (trapezoid)
Figure 5-5 Comparison of tracking and synchronization errors of PID-FC and LQR-FC
controllers
Figure 5-6 Comparison of tracking and synchronization errors of ARC and LQR-FC controllers
Figure 5-7 Free-body diagram for the load88
Figure 5-8 Block Diagram for Twisted String Actuation
Figure 5-9 Desired trajectory
Figure 5-10 Simulation results of ARC (force) controller
Figure 5-11 Configuration of experimental setup100
Figure 5-12 Tracking performance for the sine trajectory101
Figure 5-13 Comparison between actual current and desired current
Figure 6-1 The configuration of Microsoft Kinect sensor104
Figure 6-2 The 20 joints of the skeleton position acquired by a Kinect104
Figure 6-3 The 3-D coordinated system of the Kinect centered space105
Figure 6-4 Schematic of Shoulder and elbow angle107
Figure 6-5 Prototype mechanism of the robotic system with encoder used for motion
measurement
Figure 6-6 Trajectories of the right elbow when performing motor task Hand to Mouth108
Figure 6-7 Comparison of the trajectories of the right elbow between five subjects performing
separately motor task Hand to Mouth three times110
Figure 6-8 Euclidean match and Dynamic time warping match

Figure 6-9 Euclidean distance between Original and shifted trajectories
Figure 6-10 Comparison of the trajectories of the right elbow between two subjects performing
separately motor task Hand to Mouth three times112
Figure 6-11 Euclidean distance matrix and Dynamic time warping distance matrix113
Figure 6-12 Dynamic time warping path114
Figure 6-13 Time-normalized elbow-joint trajectories114
Figure 6-14 Photo frames and joint positions of motion captured by Microsoft Kinect
Figure 6-15 Flow chart of the procedure of extracting averaged trajectory118
Figure 6-16 Averaged trajectory of elbow joint118
Figure 6-17 Averaged trajectory of shoulder joint119
Figure 7-1 Bowden cable on the joints121
Figure 7-2 Slide potentiometer121
Figure 7-3 Prototype of the exoskeleton robotic arm122
Figure 7-4 Schematic of Kalman filter procedure124
Figure 7-5 Comparison between measurement of slide potentiometer and results of Kalman
filter
Figure 7-6 Comparison between measurement of linear encoder and results of Kalman filter. 127
Figure 7-6 Comparison between measurement of linear encoder and results of Kalman filter. 127 Figure 7-7 Results of Sensor fusion
Figure 7-6 Comparison between measurement of linear encoder and results of Kalman filter. 127 Figure 7-7 Results of Sensor fusion
Figure 7-6 Comparison between measurement of linear encoder and results of Kalman filter. 127      Figure 7-7 Results of Sensor fusion.      130      Figure 7-8 System block diagram of the Intel Galileo 2 development board [105].      132      Figure 7-9 Embedded control system block diagram.
Figure 7-6 Comparison between measurement of linear encoder and results of Kalman filter. 127      Figure 7-7 Results of Sensor fusion.      130      Figure 7-8 System block diagram of the Intel Galileo 2 development board [105].      132      Figure 7-9 Embedded control system block diagram.      138      Figure 7-10 Software structure.
Figure 7-6 Comparison between measurement of linear encoder and results of Kalman filter. 127      Figure 7-7 Results of Sensor fusion.    130      Figure 7-8 System block diagram of the Intel Galileo 2 development board [105].    132      Figure 7-9 Embedded control system block diagram.    135      Figure 7-10 Software structure.    138      Figure 7-11 Diagram of state machine.    138
Figure 7-6 Comparison between measurement of linear encoder and results of Kalman filter. 127      Figure 7-7 Results of Sensor fusion.    130      Figure 7-8 System block diagram of the Intel Galileo 2 development board [105].    132      Figure 7-9 Embedded control system block diagram.    135      Figure 7-10 Software structure.    138      Figure 7-11 Diagram of state machine.    139      Figure 7-12 Reference trajectories for elbow and shoulder joints.    139
Figure 7-6 Comparison between measurement of linear encoder and results of Kalman filter. 127      Figure 7-7 Results of Sensor fusion.    130      Figure 7-8 System block diagram of the Intel Galileo 2 development board [105].    132      Figure 7-9 Embedded control system block diagram.    135      Figure 7-10 Software structure.    138      Figure 7-11 Diagram of state machine.    138      Figure 7-12 Reference trajectories for elbow and shoulder joints.    139      Figure 7-13 Repetitive tests of the exoskeleton robot.    140
Figure 7-6 Comparison between measurement of linear encoder and results of Kalman filter. 127      Figure 7-7 Results of Sensor fusion.    130      Figure 7-8 System block diagram of the Intel Galileo 2 development board [105].    132      Figure 7-9 Embedded control system block diagram.    135      Figure 7-10 Software structure.    138      Figure 7-11 Diagram of state machine.    138      Figure 7-12 Reference trajectories for elbow and shoulder joints.    139      Figure 7-13 Repetitive tests of the exoskeleton robot.    140      Figure 7-14 External load mounted on the endpoint of elbow joint.    142
Figure 7-6 Comparison between measurement of linear encoder and results of Kalman filter. 127      Figure 7-7 Results of Sensor fusion.    130      Figure 7-8 System block diagram of the Intel Galileo 2 development board [105].    132      Figure 7-9 Embedded control system block diagram.    135      Figure 7-10 Software structure.    138      Figure 7-11 Diagram of state machine.    138      Figure 7-12 Reference trajectories for elbow and shoulder joints.    139      Figure 7-13 Repetitive tests of the exoskeleton robot.    140      Figure 7-14 External load mounted on the endpoint of elbow joint.    142      Figure 7-15 Stability tests of the exoskeleton robot.    142
Figure 7-6 Comparison between measurement of linear encoder and results of Kalman filter. 127      Figure 7-7 Results of Sensor fusion.    130      Figure 7-8 System block diagram of the Intel Galileo 2 development board [105].    132      Figure 7-9 Embedded control system block diagram.    135      Figure 7-10 Software structure.    138      Figure 7-11 Diagram of state machine.    138      Figure 7-12 Reference trajectories for elbow and shoulder joints.    139      Figure 7-13 Repetitive tests of the exoskeleton robot.    140      Figure 7-14 External load mounted on the endpoint of elbow joint.    142      Figure 7-15 Stability tests of the exoskeleton robot.    142      Figure 7-16 Tracking errors with respect to time without external load.    143

# LIST OF TABLES

Table 3-1 Input and Output of ANFIS model
Table 4-1 Symbols of parameters used to model driving motor
Table 4-2 Comparison of performance of controllers of sinusoidal motion
Table 4-3 Time delay of the three controllers of sinusoidal motion
Table 4-4 Comparison of the disturbance rejection between PID-FC and LQR-FC Controllers
with a 500g loading72
Table 5-1 Comparison of performance of the three controllers of sinusoidal motion
Table 5-2 Comparison of performance of the three controllers of sinusoidal motion100
Table 6-1 Self-comparison of RMSE of recorded motion (Subject 1).
Table 6-2 Self-comparison of RMSE of recorded motion (Subject 2).    116
Table 6-3 RMSEs among the various subjects' trajectory after DTW.      117
Table 7-1 Comparison of different sensor measurements 130
Table 7-2 Main features of the Intel Galileo 2 development
Table 7-3 Comparison of performance of elbow and shoulder joints
Table 7-4 Comparison of performance of elbow and shoulder joints with 5 N external load143

# **CHAPTER 1: INTRODUCTION**

# **1.1 Background**

Since the 1990s, it is well accepted that high-intensity and task-specific upper-limb treatment of active, highly repetitive movements is one of the most effective approaches for patients with motor disorders [1-3]. Previous researches have shown that patients can have better improvements in the proximal movement portion of the Fugl-Meyer test after one to two months intensive treatment activities [1]. This can help patients to return to their regular lives faster. Thus, there is an urgent need for these patients to acquire immediate healthcare resource for rehabilitation and regain capability to return to their occupations for economic reasons. Recent studies also proved that robot-assisted therapy can have similar or better results to intensive conventional therapy offered by professional physicians [4]. However, with the severe shortage of medical service in the United States and the fact that post-stroke survivors can continue to incur significant financial costs associated with their injuries for the first 12 months following discharge from an inpatient rehabilitation setting, patients often choose not to return to the hospital or clinic for complete recovery. Without proper rehabilitation activities, patients' affected arms can develop into spasticity due to muscle weakness. To compensate for the shortage of medical resources and to help patients of these groups to regain their mobility efficiently, it is necessary to develop low-cost and wearable assistive robotic devices that allow patients: 1) to rehabilitate at their own pace and 2) to acquire additional force in their daily activities. Studies of clinical results have also demonstrated that intensive robot-assisted rehabilitation activities can improve the motor outcome of stroke or neuron-injured, patients as effective as intensive therapies offered by therapists [4]. Thus, using robotic systems as therapeutic adjuncts to facilitate clinical practice can be a promising development.

In the past decade, technologies based on robotics and control systems have been adopted in rehabilitation engineering used for upper extremities [5-9] and many different types of assistive robotic devices have been developed. Nevertheless, most of these systems are designed as end-effectors and require patients to come to a medical center for rehabilitation activities [10-12]. In addition to rehabilitation activities, it is also important that patients can regain their mobility in their daily lives and return to their occupations for economic reasons [13, 14]. Some existing assistive robotic systems have been designed as wearable devices. However, due to their sizes and the required power, these devices are typically not mobile enough to be used in daily activities [15-17]. In particular, most of these assistive robotic systems adopt trajectories with the best efficiency and minimal energy cost instead of considering natural human motions. Research in the past two decades has suggested that patients recover better when their muscles are trained with the general trajectories they used to have [18, 19]. Given that assistive robotic systems reproducing natural arm movements can promote more rapid and complete recovery of arms functions than what most of the current rehabilitation resources can achieve, post-stroke/stroke patients can return to their occupations sooner with the help of these systems. In addition to providing rehabilitation activities, such robotic systems can help patients to compensate for the lost force of muscles in their daily activities as well. The manipulator used in therapy systems can apply forces to aid in specific tasks during goal-directed movements. Thus, designing and fabricating a compact assistive robotic system that can be used for both rehabilitation and daily activities is necessary.

To fabricate an effective mobile assistive robotic system, various methods [20-24] of trajectory modeling have been investigated in the area of motor control. However, there is still no clear mathematical parameterization [25, 26], describing the vast repertoire of arm motions. In addition to rehabilitation functions, usability and user acceptance are the most important factors for the success of any assistive robotic solution, particularly assistive robots that are designed to help the groups of stroke patients and elderly people to live independently. An ideal robotic solution must cater for any unique individual user needs and take into consideration the user's socio-demographic profile as well. Instead of deriving energy efficient arm models, this study identifies a model of a basic arm motion based on the

natural movement of different groups of adults, which will be adopted by the wearable exoskeleton device.

These devices supporting users' daily tasks such as holding, lifting, moving, and free mobility are called assistive robotics. So far, most of the wearable assistive robotic systems are bulky and require great energy [16], which reduces their mobility in many activities. 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 both joints (elbow and shoulder).

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 have drawn much attention [27, 28]. The basic idea of this type of actuators is to change the linear length of two or more strands by twisting them (Figure 3-1). 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. Due to its high reduction ratio of length, the actuation system can be small and lightweight. In particular, all actuators can be placed at the same location, which can greatly reduce the weight users need to carry. One specific characteristic of the assistive robotic system of upper limbs is that the inertia keeps changing while the arm is moving. 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. Many control strategies have been proposed to design adequate controllers [29-36] to deal with the coupled dynamics among axes. With the consideration of cross-coupled dynamics, LQR [36] and adaptive robust controller (ARC) [37] both demonstrate excellent performance in tracking and synchronization when the nonlinear variations are bounded. As a result, these two algorithms both guarantee that the estimated parameters stay in known bounded regions all the time. They also ensure that asymptotic tracking will not be lost. These control schemes have been implemented on different three-dimensional moving platforms, and the dynamic tracking errors can be constrained less

than 7% while all three actuators are moving [40]. These control algorithms can be good candidates to compensate for the dynamics of arms and external disturbances proposed in this study.

# **1.2 Objectives**

The objective of this study was to design a light weight, wearable, and affordable assistive exoskeleton robot for the shoulder and elbow assistance. The mechanical design is targeted to the group of the society most vulnerable to stroke, based on the average size and age of the patients, with adjustability to accommodate a variety of individuals. Unlike other exoskeleton robots, two sets of dual twisted-string actuators, each with 7 strands (1 neutral and 6 effective) were used to extend/contract the adopted strings to drive the rotational movements of shoulder and elbow joints through a Bowden cable mechanism.

In order to obtain trajectories of arm movements that can be considered to drive the exoskeleton robot, two issues need to be addressed: 1) collecting movement data of individual motions from enough subjects, and 2) identification and generating common trajectory motions based on the collected data. In particular, trajectories of the same motions are unique from person to person, which can yield a wide range of distance variations even for the same task. To generalize trajectories for specific movements can be a difficult task. Thus, instead of identifying random motions, specific ADLs will be adopted to generate trajectories for the robotic system.

To capture motions from different individuals, a systematic and efficient method is required. In particular, it is necessary to have enough samples from groups with different ages, genders, heights, body sizes and cultures. Thus, a convenient motion capture system for different physical conditions is needed. In this study, a Microsoft Kinect Camera is adopted to capture depth images of field data when a target subject starts to move his/her arms. With the mechanism of the proposed assistive robot, the control references are the rotation angles of shoulder and elbow joints. From the captured depth images, both rotation angles of subjects' shoulder and elbow joints can be obtained by calculating the 3D coordinates of individuals' shoulder, elbow, and wrist joints. This information can be derived by adopting the skeletal tracking technology contained in Microsoft Kinect SDK.

The bi-directional movement of shoulder and elbow joints of the proposed robot driven by the antagonistic twisted-strings system relies on a coupled twisting/untwisting motion of two actuators. The tracking accuracy of separate actuators also affects the synchronization performance of both actuators, which means that the adopted controller needs to respond to both tracking and synchronization errors simultaneously. Though the twisted-string actuator has a lot of advantages, its unique properties of nonlinearity, hysteresis, friction, operational dead-zone, and poor repeatability can deteriorate the desired control performance. Improper control methods can lead to limit cycles or instability. To deal with both nonlinear factors and synchronization between individual actuators, adaptive robust control (ARC) was adopted for the motion system in this study. This study also proposes an approach to compensate for important nonlinear issues: the dead-zone effect near zero-velocity and varying force feedback. To deal with the tracking errors of a stick-slip motion phenomenon caused by the dead-zone, an adaptive robust control algorithm with the consideration of dead-zone effect is studied. Additional adaption mechanisms are added to the ARC structure to provide accurate tracking and synchronization. In addition to position control, one important safety concern is that displacement control of rehabilitation robots cannot guarantee bounded force safety by only considering mechanical dynamics with the inertial load. To guarantee safety while using an assistive robotic device, an adaptive robust controller of twisted-string actuation devices need to integrate both force and displacement feedback to achieve high tracking performance and guarantee the safety demand.

#### **1.3 Dissertation Structure**

The rest of the dissertation is structured in the following way.

- Background and Literature Review (Chapter 2)
- Model Development (Chapter 3)
- Design of the assistive robot and Development of the controller (Chapter 4 and 5)

- Implementation of the assistive robot (Chapter 6 and 7)
- Conclusion (Chapter 8)

The background and literature review covers research already performed and related to the study. Since this dissertation focused primarily on the development of the assistive robot for stroke rehabilitation, the literature review addresses these aspects. However, basic information for understanding the twisted string actuator and nonlinear control was provided. Chapter 3 briefly outlines the main point of the mathematic model of the twisted string actuator but covers the development of an adaptive neuro fuzzy inference system model of twisted string actuator. Chapter 4 presents the mechanical design of the assistive robot. Three different linear control methods were compared. Two different nonlinear control approaches were compared, both of which were tested on the same experimental platform. Chapter 6 and 7 depict the implementation of the complete system, including the derivation of motion trajectories using Kinect and development of the controller using an embedded system. A discussion of the overall study, information resulting from this study, and final concluding remarks are presented in the last Chapter.

# **CHAPTER 2: LITERATURE REVIEW**

#### 2.1 History of stroke neuro-rehabilitation

After Johan Wepfer hypothesized that a stroke was caused by bleeding in the brain or a blockage of one of the main arteries that supply blood to the brain in 1620 [38], people were not recommended for being active after a stroke. Until Thomas Twitchell, one of the pioneers, began to study the types of recovering from a stroke in the early 1950s, this attitude changed and people were encouraged to do therapeutic exercises. Based on observing 121 patients [39], he found that there was 70 percent chance for fully recovering if patients started to recover their hand functions in the first four weeks after stroke. He also reported that most recovery occurs in the first three months, which can be verified by the outcome of recent researches. However, recent studies also indicate that stroke patients can still have a chance of a significant improvement physically a few years after the attack of stroke.

To recover from the disability due to stroke, the constraint-induced therapy was proposed at least 100 years ago [40], [41]. Robert Oden was able to simulate a stroke in a monkey's brain, then set up a constraint with the monkey's good arm and force the monkey to use its bad arm. After two weeks of therapy, the monkey regained the ability to use its bad arm again. With his result, researchers began to adopt such an idea for stroke patients as an alternative treatment method for upper extremity motor deficits. Stroke patients are asked to use their affected hand and arm as much as possible to do repetitive tasks during the constraint-induced therapy. The stroke patients, meanwhile, wear a restraining device on their functioning arm, as shown in Figure 2-1. Evidence from the clinical trials and recent studies[42], [43], [44] also supports this method of improving the upper extremity motor function of stroke patients and reducing motor impairment at varying stages of stoke recovery.



Figure 2-1 A stroke-patient received Constraint Induced Movement Therapy[40].

In the past century, many treatments have been proposed to help stroke patients to recover from their disabilities. Bobath, a neurodevelopmental treatment, was proposed in the 1940s by the Bobath couple [45]. Bobath treatment was mainly used for patients with central nervous system damage, such as stroke patients, cerebral palsy children, and children with Down's syndrome. The main goal of the Bobath is to suppress abnormal reflex and muscle tension through three methods: therapeutic handling, key points of control, and reflex inhibition pattern. Bobath treatment is still the mainstream approach for stroke patients' rehabilitation now although there is a lack of sufficient evidence to demonstrate this trearment is an ideal therapy. The Brunnstrom Approach, including stroke assessment and treatment, was developed by a Swedish physical therapist Signe Brunnstrom [46] in the 1960s. This approach emphasizes the synergic pattern of movement which develops during recovery after a stroke. Based on the research of Brunnstrom, the recovery of stroke patients follows six stages: flaccid period, spasticity, patient gains voluntary control over synergies, decrease in spasticity, synergies decline and disappearance of spasticity. The treatment has been updated by other alternative approaches, but the stroke assessment, six stages of sequential motor recovery after a stroke, has become the foundation and a criterion of the modern stroke rehabilitation.

Inspired by the success of mental practice in enhancing skill development for normal individuals such as balance training for elderly females, researchers combined mental practice and physical practice in hopes of improving recovery after stroke. Mental Practice with motor imagery is a mental process during which stroke patients mentally rehearse and simulate a given action within working memory [47]. However, there has been insufficient evidence until now proving the effectiveness of mental practice with motor imagery in stroke rehabilitation due to the methodological limitation [48].

In addition to physical training, electrical instruments also become important tools in rehabilitation activities. For instance, functional electrical stimulation (FES) applies a pulse of electrical current to stimulate the neuromuscular system in order to generate a muscle contraction [49]. This electrical stimulation imitates brain's signals to control the affected limbs in order to retrain weak muscles and normal movement for stroke patients. Nowadays, different research groups have started to link the connections between muscle motions and neuro-signals with EMG measurements and use the connection to adjust stimulation parameters.

## 2.2 Robotic assistive therapy in stroke rehabilitation

As the number of the stroke patients grows rapidly, the pressure on the medical care resources for stroke patient recovery intensifies. 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. Robot-assisted therapy in stroke rehabilitation as an alternative method that has expanded dramatically in the last two decades.

The intensity of practice plays an important role in motor recovery for stroke patients based on the experience from the conventional therapy in stroke rehabilitation and the motor learning theory [50]. Compared with conventional therapy, robotic rehabilitation therapy can provide safe and high-intensity rehabilitation to patients with mild to severe impairments after stroke [51]. Recent studies have shown the benefits of robotic rehabilitation therapy on the upper-limb movement of patients in the acute phase of recovery after stroke. [10-13]. There are two major types of robots, end-effector-type and exoskeleton type, which can be classified based on the way of the movement is transferred from the robotic device to the patients' limbs and are used by stroke patients at different stages of motor recovery. End-effector type robotic devices normally only attach to the user's limb at its most distal part. Due to the easy setup and limited control of the proximal joints of the limb, end-effector-type robotic devices are usually designed to provide unilateral therapies for the group of mild and moderate stroke patients. In contrast to end-effector-type robotic device which can provide direct control of individual joints. Therefore, the exoskeleton-type robotic device which can support more complex rehabilitation activities especially for stroke patients with a large motor impairment.

#### 2.2.1 End-effector-type robotic devices

End-effector-type robotic devices, commonly having multi-linkage mechanisms, are designed to interact with stroke patients' upper extremity. The exact nature of the devices is to drive the patients' upper extremity to achieve the purpose of doing rehabilitation activities though the direct connection between devices and patients' hand during the rehabilitation. Therefore, these types of the devices have the advantages of simple structure, low cost, reliable and less complicated control algorithms and most early upper limb rehabilitation robots are the end-effector-type robotic devices.

The research on end-effector-type robotic devices has begun since the 1990s. Many universities and institutes of the world have developed various kinds of the end-effector-type devices, such as MIT-MANUS, MIME, ARM Guide, GENTLE/S, NeRobot and MULOS, which have already been used by stroke patients' for rehabilitation or experimental research activities. The MIT-MANUS is the first end-effector-type robotic device and was used in the physical therapy of stroke victims as shown in Figure 2-2. MIT-MANUS, which has a five-link mechanism with two planar degrees of freedom, has proved its worth for shoulder and elbow rehabilitation by many stroke patients [52]. It can assist stroke patients during rehabilitation or act as an impendence, which can be adjusted based on the individual treatment plan. To use this robot, stroke patients need to sit in front of a table with their forearm and wrist attached to a brace of the robot. Depending on the stroke patients' moving capability, therapists who use MIT-MANUS can decide the level of guidance and assistance to facilitate arm movements of the stroke patients. A clinical trial, which involved 56 patients, was hosted by the Department of Mechanical Engineering and the Department of Brain and Cognitive Sciences at MIT. In this trial, 30 patients received standard therapy; the other 29 patients received an additional daily hour of MIT-MANUS therapy are experimental results indicated that patients who received MIT-MANUS therapy earned the scores of tests measuring increased movement as twice as high for patients who only received standard therapy [53].



Figure 2-2 A stroke-patient received therapy with a commercial version of MIT-MANUS at the Burke Rehabilitation Hospital (White Plains, NY) [16].

In 2005, Furusho laboratory of Osaka University developed a 3-DOF-rehabilitation robot for upper limbs "EMUL", as shown in Figure 2-3. With their design, ERF (Electro

Rheological Fluids) actuators were adopted to adjust the force for resistance training. The research group has also developed a 3D rehabilitation system that combines virtual reality technology and video game to improve the dexterity and the moving range of user's limbs [54].



Figure 2-3 An EMUL rehabilitation system used fot therapy [17].

#### 2.2.2 Exoskeleton-type robotic devices

Compared to the end-effector-type robotic devices, the exoskeleton-type robotic devices have the advantages of producing movements for specific joints of the limb through imitating the skeletal structure of human's limb. Therefore, exoskeleton-type robotic devices can be used to guide patients' hand or arm to desired positions, which meets the requirements of independent and concurrent control of the specific movement of patients' upper limb.

He et al. [55] proposed a pneumatic muscle (PM) driven therapeutic device (RUPERT) providing an alternative low cost and take-home method to the clinic treatment in 2005. RUPERT has four pneumatic muscles (shoulder, elbow, pronation and wrist/fingers), which can provide the desired joint ranges of motion to achieve the goal of reaching and self-feeding tasks for users. The latest design of the RUPERT IV has expanded the motion range and the degrees of freedom in aid of performing the assisted tasks [56]. The RUPERT IV now includes hand/wrist extension, forearm supination, elbow extension, humeral external rotation and shoulder elevation as shown in Figure 2-3. For the purpose of offering high performance physical therapy while ensuring safe operation, Perry [57], [58] developed a

7-DOF upper limb rehabilitation robot (CADEN-7) as shown in Figure 2-4, in which the device offers ranges of movement in shoulder flexion and extension, shoulder pronation and supination, upper arm rotation, elbow flexion and extension, forearm rotation, wrist flexion and extension and wrist abduction and adduction. Multi-degree-of-freedom rehabilitation devices have many advantages over conventional devices. For example, multiple actuators can generate larger torques for individual joints, which can make patients move their arms in the normal wide workspace. In particular, the hybrid structure that combines multiple types of actuators in a single system can greatly simplify the mechanical design. For example, in Perry's design, the electric motors are used for the angular movements of both upper arm and forearm and the cable actuators are used for other movements of the device. Such a structure cannot only simplify the complexity of the robotic device; it also reduces its weight and corresponding friction caused by gear transmission.



Figure 2-4 The current version of RUPERT IV system [19].

In order to identify the outcome difference between these two types of therapies (robot-assisted therapy and conventional therapy), Lum et al, [59] randomly divided 27 subjects with chronic hemiparesis into two groups, robot and control group. Subjects in the robot group practiced shoulder and elbow with the help of assistive robots. Subjects in the control group received regular occupational therapy. The improvements in proximal movement of two groups after 2 months and 6 months were then compared by using the

Fugl-Meyer assessment of motor impairment. The robot group had larger improvements in strength and reach extent than the control group after 2 months of treatment. However, after 6 months of treatment there was no significant difference between the two therapies in terms of the Fugl-Meyer test [60].

Lo, et al. [61] also performed a similar analysis to verify the effectiveness of robot-assisted therapy by comparing with intensive therapy and regular care. In their experiment, 127 patients with moderate upper-limb impairment 6 months or more after a stroke were randomly divided into three groups, one group including 49 patients receive robot-assisted therapy, one group including 50 patients received intensive comparison therapy (ICT), and one group including 28 patients received regular care. All groups received 36 one-hour therapy sessions in 12 weeks. At 12 weeks, patients receiving robot-assisted therapy demonstrated higher Fugl-Meyer score than the patients with regular care, but the score is lower than the patients with ICT. However, the differences of the score between the groups of robot-assisted and ICT were not significant. Nevertheless, for stroke patients with long-term upper-limb impairment, the effectiveness of robot-assisted therapy is not significant while comparing with intensive therapy. It was even worse than patients who receive regular care. As the results, this research provides the evidence that robot-assisted therapy can be at least as effective as other therapies within the first 6 months after stroke. With the benefit of less requirement from practical physicians, robot-assisted rehabilitation can possibly compensate for the shortage of medical resources in the future.



Figure 2-5 A CADEN-7 Exoskeleton robotic system [21].

# 2.3 Twisted string actuator

For the application of rehabilitation, weight is always an important consideration while designing assistive devices. The inadequate actuator can make an assistive device bulky and heavy. In this research, twisted string actuation is adopted to implement the linear movements for such a multi-axial system.

#### 2.3.1 Applications of the twisted string actuation

Using the twisted string has come a long way from when ancient Indiana people used the pump drill for starting fires and drilling holes. Another type of twisted string application is the Spanish windlass that can be used to draw heavy loads. In the past five years, twisted string actuators have raised the interest of researchers around the world after T.Wurts applied them to the robotic hand. Twisted-string actuation devices have been adopted by various robotic systems. In 2011, the laboratory of Process automation (LPA) [62] developed a DEXMART hand, which includes 24 twisted string actuation units, to operate the movements

of five fingers and writ. In the DEXMART system, as shown in Figure 2-6, the twisted string systems can be appropriately integrated into the robotic mechanism due to its slender characteristics. The DEXMART hand reveals the potential of the twisted string actuation that can provide high accuracy and stable motion. In addition to LPA, a similar approach called "twist drive actuator" was proposed by Sonoda, et al [63]. A prototype of a robotic hand (Figure 2-7) was developed using such an actuator. In their design, the rotational motions of motors are transmitted to the linear motions and the pulling forces are generated by twisted strings. To solve the issue that twisted strings can only provide pulling forces, a spring is employed in each joint for an extension. With the benefit of the simple structure of twisted drive actuators, his robotic hand can efficiently utilize the space in the finger structure.



Figure 2-6 DEXMART hand [25].



Figure 2-7 Configuration of a multi-fingered robotic hand proposed by Sonoda et al [26].

Aukes and his colleagues [64] developed a new class of multi-fingered robotic hands by adopting twisted string actuator. Providing three independent DOF in the one finger by only one twisted string actuator are the major characteristic of this manipulator. This manipulator is not only able to grasp multiple objects but also conform to their shapes. In order to balance the tendon force between each phalanx and provide return actuation, springs are attached across each joint.



Figure 2-8 Aukes' finger robot [27].

#### 2.3.2 Mathematical model of the twisted string actuation

Gaponov et al. [65] proposed a modified model on the basis of the Wurts model. In

this model, the volume of a string is supposed to remain constant during the process of the twisting since each strand can not be infinitely compressed by its neighbors and that there are no air hollows inside the string. For example,  $V_1=V_0$ , where  $V_1$  and  $V_0$  denote the twisted and untwisted states of the string. Therefore, the actual radius of a string shrinks as it twist or expands as it untwist based on the above assumptions, as shown in Figure 2-9. The volume of a twisted string at different state can be calculated as a cylinder. That is

$$V = \pi r^2 d \tag{2-1}$$

where the r is the effective radius of the string, d is the length of the string after being twisted.



#### Figure 2-9 Increase of string's radius after being twisted.

Thus, the effective radius of a string after being twisted increases nonlinearly, which

is

$$r_{new} = r_0 \sqrt{\frac{L_0}{d}} \tag{2-2}$$

where  $L_0$  is original length of the strand,  $r_0$  is the original radius of the string,  $r_{new}$  is the radius of the string after being twisted. Figure 2-9 illustrates the change of radius of a twisted string. The new conventional kinetostatic model can be obtained

$$d = \sqrt{L_0^2 \left(1 + \frac{F_i}{L_0 K}\right)^2 - \theta^2 r_{new}^2}, \qquad (2-3)$$

,where *K* is the stiffness coefficient of the strands,  $\theta$  is the angle of twisting,  $F_i$  is the tension of the string.

# 2.4 Synchronization of multi-axial system

In some rehabilitation robotic systems, joint movement is controlled by antagonistic pairings of actuators like the pneumatic artificial muscles and twisted string actuators. The accuracy of joint motion depends not only the accuracy of individual actuator but also on their synchronization. Synchronization of antagonistic actuation systems is crucial for robotic joint to achieve the desired tracking requirements. In this study, four twisted string actuators are employed to drive the elbow and shoulder joints of the exoskeleton robot. Therefore, more robust and synchronization control strategies are required to improve the accuracy of the exoskeleton robot.

In order to cope with the synchronization problems in a multi-axial system, a tandem control approach has been applied. The tandem control approach is shown in Figure. 2-10. In this scheme, the synchronization of individual actuators is reached passively thorough tracking the same ideal reference trajectory and dynamic matching of individual actuators. However, this approach suffers from poor synchronization due to mismatched disturbances.

Therefore, various researchers have proposed different synchronization control strategies, such as Master-Slave and cross-coupling control approach, to improve the synchronization performance among multi-axes. The Master-Slave control approach is shown in Figure. 2-11. Individual axes in the multi-axial system are either identical or different. Therefore, in Master-Slave Control approach, the slowest actuators among the system is normally chosen as the master, the motion of the master is the reference command to the slaves. The synchronization can be achieved based on the slaves following the master.



Figure 2-10 Tandem control.



Figure 2-11 Master-slave control.

Mori and his colleagues [66] proposed a master-slave control approach for double-sided milling process machine for thin metal plates. The synchronization of the spindles is achieved by the master-slave control system, which can guarantee the teeth of spindles at both sides to begin cutting at the same time and avoid the thrust force and forced chatter vibration. In their control system, the right spindle was considered as a master and running at the specific speed, at the same time the left spindle followed the right spindle's speed.





In view of the nature of the master-slave control, any disturbance acting on the slave unit cannot be measured and compensated by the controller on the master unit side. There also exists an unavoidable time delay between the master and the salve unit, which limits applications of master-slave synchronization strategy such as in disturbance sensitive applications. To overcome the disadvantages existing in both tandem and master-slave synchronization control strategies, the cross-coupling synchronization strategy was proposed by Koren [67]. The concept of the cross-coupling control is to feedback the multi-axis contour errors to the individual axis, and this error can be compensated by a control law with a cross-coupling gain. However, due to difficulty in deciding the cross-coupling gain and system overall stability, many other researches improved the cross-coupling control by adopting the variable cross-coupling gain [68] or developing a new multi-input and multi-output cross-coupling structure [69].

# 2.5 Nonlinear control

In the real physical world, most systems are nonlinear. Although some nonlinear system can be successfully described and controlled by linear methods, the limits of the classic linear control method were gradually exposed with the increasing complexity of control objects and demand of high accuracy control objectives. To achieve high performance and accuracy, nonlinearities existing in a system, such as dead zones, friction, saturation as well as hysteresis need to be compensated or canceled. However, these nonlinearities are not smooth, and discontinues often lead to the unpredicted behavior of the system, such as limit cycle and oscillation, if not properly handled. Sometimes the parameters of a linear system keep varying when the circumstances change. In this case, the controller would provide the ability to adapt to the changing parameters of the system in real time to maintain stability and performance. Considering the benefit of nonlinear control and powerful computing capability of current microprocessors, the nonlinear control becomes increasingly popularity.

#### 2.5.1 Feedback linearization

Linear control approach is an important means in various engineering applications. The mature linear control theories such as pole-placement, root-locus as well as frequency response methods, can help make sure that a compensated system can be operated efficiently and effectively. Due to the simplicity and highly matured design technique of linear control methods, researchers have also tried to find a way of adopting linear control methods to design linear controllers for the nonlinear system. Therefore, the feedback linearization [70] was proposed as a way of transforming original nonlinear system models into equivalent models of a linear form. The feedback linearization is completely different from Jacobian linearization in which the first order term of Taylor expansion around the point of interest is adopted to approximate the nonlinear function. Feedback linearization is achieved by exact state transformation and feedback rather than by approximation of the nonlinear system dynamics into linear form. For example, if we have a second-order system:

$$\dot{x}_1 = -3x_1 + ax_2 + \cos x_1$$
  
$$\dot{x}_2 = x_2 \sin x_1 + u \sin(2x_1)$$
 (2-4)

The linear control approach can be adopted to stabilize the system by linear approximation in a small region around the equilibrium point. However, the linear control approach cannot guarantee the stability of the system in the global region due to the nonlinearity in the equations.

If we define a new set of state variables
$$z_1 = x_1 z_2 = ax_2 + \cos x_1,$$
(2-5)

then, the dynamics of the system are:

$$\dot{z}_1 = -3z_1 + z_2$$

$$\dot{z}_2 = 3z_1 \sin z_1 - \cos z_1 \sin z_1 + au \sin(2z_1),$$
(2-6)

If we define the control input *u* as

$$u = \frac{1}{a\sin(2z_1)} (v + \cos z_1 \sin z_1 - 3z_1 \sin z_1), \qquad (2-7)$$

where v is a virtual input to be designed. Substituting Eq. (2-7) into Eq. (2-6) gives

$$\dot{z}_1 = -3z_1 + z_2 \dot{z}_2 = v ,$$
 (2-8)

Thus, the dynamics of the system have been changed from nonlinear to linear and linear control approaches can now be applied to stabilize the system. For example, the linear state feedback control law can be used to obtain the value of v.



Figure 2-13 Feedback Linearization.

### 2.5.2 Adaptive control

Like most control techniques, feedback linearization control also has disadvantages. Based on the designing process of feedback linearization, it is obvious that the exact model is necessary to algebraically transform the nonlinear system dynamics into linear form. In the real physical system, it is impossible to correctly obtain the exact value of the parameters of the system. Even if the exact value of the parameters of the system can be achieved, the system has remaining uncertainties and unknown nonlinearity which cannot be easily and correctly described in system dynamics. Therefore, both performance and stability may not be achieved if the feedback linearization is applied to some nonlinear systems.

To deal with nonlinear systems with constant or slow-varying uncertain parameters, adaptive control method was developed as an approach to control such kind of systems. The central idea of the adaptive control method is to estimate the uncertain parameters on-line based on comparing desired and actual signals. The researchers started to study the adaptive control since the early 1950's based on looking for a controller of aircraft that meets high performance requirements within a wide operation range, in which speed and altitudes keep changing.

#### 2.5.3 Robust control

Like the feedback linearization techniques, adaptive control approach also relies on the structure of the model but not parameters of the model. The modeling inaccuracies have strong negative side effects on nonlinear adaptive control. There is no mechanism in adaptive control approach that can ensure the estimated values being bounded in certain regions. Therefore, although the system can be stable eventually based on the Lyapunov theory, the transient performance and stability cannot be guaranteed if the estimated values of parameters are beyond the reasonable boundary. Therefore, researchers were wondering if there is a control approach to keep the system stable in both transient and steady states. With this sense, robust control method was proposed to deal with model inaccuracies and uncertainties. The basic idea of the robust control approach is to adopt known bounding functions to add the restriction on the unknown dynamics of a system. A simple systematic implementation of robust control is sliding mode control[72], which maintaining stability and consistent performance under the situation of modeling imprecisions. However, even though the robustness is guaranteed, the stability and consistent performance provided by robust controllers are achieved at a cost of losing the ability to adapt to changing parameters.

Considering the design process of adaptive control and robust control, it is clear that

neither adaptive control nor robust control can meet the requirements of both transient performance and global stability. To achieve high performance control, some researchers tried to combine adaptive control and robust control together. Rohrs, et al [73] proposed robust adaptive control (RAC) in which the consistent performance and stability can be achieved by modifying the adaptive laws, such as modifying the pure integral action of the adaptive laws via a small feedback gain, constraining parameters estimates to lie within bounded sets in special parameter space, and slowing down adaptation relative to the speed of growth of modeling errors. Yao [74] developed the adaptive robust control (ARC) approach based on the philosophy of that robust performance should not be lost when introducing adaptive mechanism. In adaptive robust control, the adaptive mechanisms are adopted to attenuate model uncertainties as well as the robust feedback are adopted to reduce the side effects generated by non-repeatable uncertainties and external disturbances.

### 2.5 Home stroke rehabilitation

With the growing demand for stroke rehabilitation and lacking of the medical resources, the remote stroke rehabilitation growth is on the rise. In fact, the remote stroke rehabilitation can avoid the cost and burden of travel to the hospitals, as well as increase the exercise frequency. High frequency rehabilitation for stroke patients may increase the chance of improving stroke patients' quality of life and maintain muscle vitality and avoid eventual loss of muscle mass. To make the home stroke rehabilitation easier and effective, the researchers have tried various remote methods, such as telephone, radio, closed-circuit television and satellite, to investigate the actual performance of remote stroke rehabilitation. Hoffmann, et al. [75] developed a remote measurement for stroke patients' upper limb range of motion based on the internet. They compared the measurements performance between the internet-based goniometer and the conventional clinical tool, the universal goniometer. From the experimental results, the internet-based goniometer showed reliability and accuracy which make it possible that therapists can confidently use the internet-based goniometer to measure the stroke patients' upper limb range of motion instead of going to the stroke patient's home.

Holden et al. [76] proposed a tele-rehabilitation system for upper extremity rehabilitation of stroke patients using a virtual environment (VE) as shown in Figure. 2-14. The basic idea of their system is that stroke patients, who sitting in front of VE scene and one monitor for video conference with therapist, imitate the motion of a prerecorded teacher movement being performed in the VE scene, meanwhile, the therapist at the clinic can control the speed of the teacher movement in the VE scene based on the real-time performance of the stroke patients.

Since its invention, the Microsoft Kinect sensor, a depth camera, has gained more interest in home-based rehabilitation movement exercise as an interface and trajectory capturer [77], [78]. One of the greatest advantages of Microsoft Kinect is that it does not require the patient to wear anything while tracking one's movement. In the study done by Chang et al. [79], the accuracy of the Kinect has been examined by comparing the motion tracking performance between the Microsoft Kinect and OptiTrack optical system. In their study, motion trajectory data of six upper limb motor tasks were collected from individual subjects. The experimental results demonstrated that the low-cost Kinect can have good repeatability and adequate accuracy.



Figure 2-14 Schematic of the Holden's VE tele-rehabilitation system [39].

Su [77] also proposed a Kinect-based system for ensuring home-based rehabilitation (KHRD). In his system, the stroke patients try to repeat the same rehabilitation exercise

performed by them under the supervision of therapist in front of Kinect at home. The Kinect captures the stroke patients moving trajectory and sends the trajectory to a computer which evaluates rehabilitation exercise function by comparing it with the trajectory at the clinic. The therapist can provide the suggestion to the patients to decrease the adverse rehabilitation.

# **CHAPTER 3: TWISTED STRING ACTUATION**

The basic concept of a twisted-string actuator is simple: it converts rotational movement of a motor into linear movement by a string as shown in Figure 3-1. With one end fixed on a DC motor shaft and the other end fixed on the load, two or multiple strands can be twisted by the rotational movement of a motor. As the motor starts to rotate, the string connected to the motor shaft starts to reduce the distance between both ends and generate a linear motion. As the motor reverses its direction, the length of the strands gradually returns to its initial value. The twisted string is considered as a one direction linear actuator since the actuator cannot generate the push force to the load during the motor reversing process.



Figure 3-1 Concept and configuration of a twisted string actuation system [28].

# 3.1 Kinetostatic Model of the Twisted String Actuation

A mathematical model for such a twisted-string actuator has been developed by W ürtz et al [28],[27]. The relationship among the angle  $\alpha$ , the revolution of the motor shaft  $\theta$ , and the

length of the string L can be concluded from the geometric configuration of the helix formed by the strands as shown in Figure 3-2. They are

$$L = \sqrt{\theta^2 r^2 + d^2} \tag{3-1}$$

$$\sin \alpha = \frac{\theta r}{L}, \cos \alpha = \frac{d}{L}, \tan \alpha = \frac{\theta r}{d}$$
(3-2)

where  $\alpha$  is the angle between the strand and horizontal level, *L* is the strand length as it is unwrapped,  $\theta$  is the total revolution of the motor shaft, and *d* is the length of the string after being twisted.



Figure 3-2 Schematic representation of twisted string and forces.

The coefficient *r* is the effective radius of the string, which is a combination of multiple strands. The effective *r* which is the distance between the effective strands and center of rotation would be  $r = r_s + r_c$ , where  $r_s$  is the radius of the central neutral strand and  $r_c$  is the radius of the surrounding effective strands. The twisted string is assumed to be a linear spring that only provides the tensile force. As the strands start to be twisted, the length of a single strand is changed from its original, or untwisted, length  $L_0$  to L. Then the relationship between the tension force  $F_i$  and the changed length of the strand can be derived from Hooke's law, which is

$$F_{i} = K_{L}(L - L_{0}) = K_{L}(\sqrt{\theta^{2}r^{2} + d^{2}} - L_{0}).$$
(3-3)

where  $K_L$  is the stiffness coefficient of the strands. Thus, the total length of a twisted-string *L* is a function of the tension  $F_i$ , the strand stiffness  $K_L$ , and its unloaded length  $L_0$ . That is

$$d = \sqrt{L_0^2 \left(1 + \frac{F_i}{L_0 K_L}\right)^2 - \theta^2 r^2} .$$
 (3-4)

If *n* strands are included in a string, the projected forces along vertical and radial directions can be derived, which are

$$F_z = nF_i \cos \alpha \text{ and } F_\tau = nF_i \sin(\alpha).$$
 (3-5)

Thus, the generated torque  $\tau_L$  becomes

$$\tau_L = r F_z \tan(\alpha) = F_z \theta r^2 / d. \qquad (3-6)$$

The relationship between the total axial force generated by twisted string actuation and the torque generated by motor can be derived from (3-6), which is

$$K_{\tau}(\theta) = \frac{\tau_L}{F_Z} = r \tan(\alpha) = \frac{\theta r^2}{d}$$
(3-7)

However, the total stoke of the linear movement of the string is restricted by the number of allowable revolutions of the string. Based on this geometric relationship, there exists a maximum angle (or revolutions)  $\alpha_{max}$  that all strands are wrapped tightly around each other. After this point, strands begin entangling around themselves. Assuming there is no load ( $F_z=0$ ) applied to the transmission and combining (3-4) with the geometry of the strands at maximum  $\theta$  angle,  $\theta_{max}$  and  $d_{min}$  are

$$\theta_{\max} = \frac{L_0}{\sqrt{r^2 + \frac{r_s^2 n^2}{\pi^2}}} \text{ and } d_{\min} = \frac{L_0}{\sqrt{1 + \left(\frac{r\pi}{r_s n}\right)^2}}.$$
 (3-8)

# 3.3 Adaptive neuro fuzzy inference system model

Many attempts have been made to construct simplified models for the twisted string in order to design a model-based controller such as the model proposed by Wurts and Gapomov discussed in the previous section. Although these models can reflect the relationship between force and string length, there are still obvious differences between mathematical modeling and actual performance of a twisted string. The main reason for the differences can be caused by not considering friction among strands, the elasticity of single strands and deformation of individual strands. In this study, an adaptive neuro-fuzzy inference system (ANFIS) is also adopted to model the twisted string actuation system. The fuzzy inference mechanism is adjusted by applying the NN to process the input and output data in ANFIS. The advantage of this model is that the nonlinear phenomena can be included as part of the input to make the estimation more accurate. In the twisted string actuation system, there are three inputs and two outputs. The two inputs are motor revolutions and input current. The output variables are the actuation stroke  $\Delta d$  and the force generated by the twisted string actuator as shown in the Table 3-1.

Input		Output	
Current	Motor Revolutions	Stroke $\Delta d$	Force

Table 3-1 Input and Output of ANFIS model

Due to the fact that there are two outputs in the twisted string and the ANFIS can only have one output, the multi-output model of the twisted string can be designed by putting two single ANFIS model in parallel as shown in Figure 3-2. Based upon the inputs, the two ANFIS models generate two output signals representing the force and linear displacements of the twisted string actuation system.



Figure 3-3 A two output ANFIS structure.

The ANFIS combines the fuzzy inference system with an artificial neural network. In the fuzzy inference system, the decision of the membership function and rule bases of the fuzzy inference system cannot be automatically acquired although the fuzzy inference system can reason with imprecise information. By incorporating the artificial neural network's ability of automatically training the parameters of the system, the ANFIS gives the fuzzy inference system the capability to adjust the membership function parameters based on the input and

output data sets. The Sugeno fuzzy model, which was proposed by Takagi, Sugeno and Kang [80], is used to generate fuzzy rules based on given input-output data sets. Instead of generating the fuzzy rules by summarizing the expert reasoning process, the Sugeno fuzzy model provides a systematic approach to generating fuzzy rules. The twisted string actuation system has two inputs, current (*I*), motor revolutions ( $\theta$ ) and the output are the actuation stoke ( $\Delta d$ ) and the force ( $F_z$ ). Then the if-then rules based on the first -order Sugeno fuzzy model are given in the following equation:

Rule 1: IF I is 
$$A_{m1}$$
 and  $\theta$  is  $B_{m1}$ ; then  $F_{z1} = p_1 \cdot I + q_1 \cdot \theta + c_1$   
Rule 2: IF I is  $A_{m2}$  and  $\theta$  is  $B_{m2}$ ; then  $F_{z2} = p_2 \cdot I + q_2 \cdot \theta + c_2$   
:  
Rule k: IF I is  $A_{m1}$  and  $\theta$  is  $B_{m1}$ ; then  $F_{zH} = p_H \cdot I + q_H \cdot \theta + c_H$ 

$$(3-9)$$

where j=1,2,...,q,  $H=1,2,...,q^2$ ,  $A_m$  and  $B_m$  are the fuzzy membership sets, q is the number of the membership function,  $F_{ZH}$  is the linear functions,  $p_H$ ,  $q_H$  and  $c_H$  are parameters of the linear function  $F_{ZH}$ . The architecture of five layer ANFIS with two inputs and 2q membership functions is shown in Figure 3-4.

Input Layer1 Layer2 Layer3 Layer4 Layer5  $I = \theta$ 



Figure 3-4 Architecture of a five layer ANFIS.

In the Layer 1, the triangular shaped membership function is used to generate the membership grades, which is

$$O_{1,k} = \mu_{A_k}(I) \quad k = 1, 2...q \tag{3-10}$$

or

$$O_{1,k} = \mu_{B_k}(\theta) \quad k = q+1, q+2, \dots 2q$$
(3-11)

where I,  $\theta$  are the inputs to node  $A_k$  or  $B_k$  which are the fuzzy set associated with triangular shaped function.

$$\mu_{A_{k}}(i) = \begin{cases}
0, & I \leq a_{k} \\
\frac{I - a_{k}}{b_{k} - a_{k}}, & a_{k} \leq I \leq b_{k} \\
\frac{c_{k} - I}{c_{k} - b_{k}}, & b_{k} \leq I \leq c_{k} \\
0, & c_{k} \leq I \\
0, & \theta \leq a_{k} \\
\frac{\theta - a_{k}}{b_{k} - a_{k}}, & a_{k} \leq \theta \leq b_{k} \\
\frac{c_{k} - \theta}{c_{k} - b_{k}}, & b_{k} \leq \theta \leq c_{k} \\
0, & c_{k} \leq \theta
\end{cases}$$
(3-12)
(3-13)

where  $a_k, b_k, c_k$  are the parameters of the membership function. In the Layer 2, the output of each node is generated by using T-norm operator for incoming signals and it is given in the following equation.

$$O_{2,H} = W_H = \begin{bmatrix} \mu_{A_1} \times \mu_{B_1} & \cdots & \mu_{A_1} \times \mu_{B_q} \\ \vdots & \ddots & \vdots \\ \mu_{A_q} \times \mu_{B_1} & \cdots & \mu_{A_q} \times \mu_{B_q} \end{bmatrix}$$
(3-14)

Each element of the matrix, which is T-norm algebraic product, represents the firing strength of the associated rule. It is necessary to normalize all the firing strengths in the ANFIS, otherwise the network will be ill-conditioned. In the Layer 3 normalization is done to make the output of each node in this layer have the same range that can guarantee stable convergence of the network. The firing strengths are normalized in the following equation.

$$O_{3,H} = \overline{W}_{H} = \frac{W_{H}}{W_{1} + W_{2} + \dots + W_{q^{2}}}$$
(3-15)

In the Layer 4, the rule outputs based on the Sugeno fuzzy model can be calculated by substituting the value of parameters of the linear function  $f_j$ .

$$O_{4,H} = \overline{W}_H F_{zH} = \overline{W}_H (p_H \cdot I + q_H \cdot \theta + c_H)$$
(3-16)

In the Layer 5, the output of the ANFIS can be acquired by summarizing all incoming signals from the layer 4.

$$O_5 = \sum_H \overline{W}_H F_{zH} \tag{3-17}$$

There exist two kinds of parameters, premise parameters and consequent parameters, based upon the analysis of the architecture of the ANFIS. The parameters in the layer 1, such as  $a_k$ ,  $b_k$ ,  $c_k$ , are referred to as premise parameters. The parameters in the layer 4, such as  $p_H$ ,  $q_H$  and  $c_H$  are referred to consequent parameters. The hybrid learning algorithm is adopted to optimize both premise and consequent parameters in the ANFIS. The hybrid learning algorithm has forward pass and backward pass, in which the premise parameters and consequent parameters and consequent parameters and the premise parameters are obtained individually. The output of the ANFIS can be expressed as a linear combination of the consequent parameters when the values of premise parameters are fixed in the forward pass.

$$F_{z} = \overline{w_{1}}F_{z1} + \overline{w_{2}}F_{z2} + \cdots + \overline{w_{q^{2}}}F_{zq^{2}}$$
  
=  $(\overline{w_{1}}I)p_{1} + (\overline{w_{1}}\theta)q_{1} + (\overline{w_{2}}I)p_{2} + (\overline{w_{2}}\theta)q_{2} + \cdots + (\overline{w_{q^{2}}}I)p_{q^{2}} + (\overline{w_{q^{2}}}\theta)q_{q^{2}}$  (3-18)

Rewrite Eq. (3-18) as:

$$F_z = \mathbf{X}\mathbf{W} \tag{3-19}$$

where

$$\mathbf{X} = \begin{bmatrix} I \quad \theta \quad \cdots \quad I \quad \theta \end{bmatrix}_{1 \times 2q^2}, \mathbf{W} = \begin{bmatrix} p_1 \\ q_1 \\ \vdots \\ p_{q^2} \\ q_{q^2} \end{bmatrix}_{2q^{2*1}}$$
(3-20)

The least-squares estimation method is employed to find the values of the consequent parameters. Since the  $\mathbf{X}^T \mathbf{X}$  is nonsingular matrix, the  $\hat{\mathbf{W}}$ , estimated value of the  $\mathbf{W}$ , can be solved uniquely:

$$\hat{\mathbf{W}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}F_{z}$$
(3-21)

When the consequent parameters are determined based on the output of the ANFIS, we propagate the error signal  $e_p$  between the actual output and output of ANFIS back through the network. Then the value of premise parameters can be updated by gradient descent.

$$a_{k}(t+1) = a_{k}(t) - \eta \frac{\partial E_{p}}{\partial a_{k}}$$

$$b_{k}(t+1) = b_{k}(t) - \eta \frac{\partial E_{p}}{\partial b_{k}}$$

$$c_{k}(t+1) = c_{k}(t) - \eta \frac{\partial E_{p}}{\partial c_{k}}$$
(3-22)

where  $E_p = \frac{1}{2}(F_z - \hat{F}_z)^2$ ,  $\eta$  is the learning rate for  $a_k$ ,  $b_k$ ,  $c_k$ , and t is the time stamp. The

chain rule is employed to calculate the partial derivatives  $(\frac{\partial E_p}{\partial a_k}, \frac{\partial E_p}{\partial b_k} \text{ and } \frac{\partial E_p}{\partial c_k})$ , which can be

used to update the  $a_k, b_k, c_k$ .

$$\frac{\partial E_{p}}{\partial a_{k}} = \frac{\partial E_{p}}{\partial F_{z}} \frac{\partial F_{z}}{\partial F_{zk}} \frac{\partial F_{zk}}{\partial w_{k}} \frac{\partial w_{k}}{\partial \mu_{k}} \frac{\partial \mu_{k}}{\partial a_{k}}$$

$$\frac{\partial E_{p}}{\partial b_{k}} = \frac{\partial E_{p}}{\partial F_{z}} \frac{\partial F_{z}}{\partial F_{zk}} \frac{\partial F_{zk}}{\partial w_{k}} \frac{\partial w_{k}}{\partial \mu_{k}} \frac{\partial \mu_{k}}{\partial b_{k}}$$

$$\frac{\partial E_{p}}{\partial c_{k}} = \frac{\partial E_{p}}{\partial F_{z}} \frac{\partial F_{z}}{\partial F_{zk}} \frac{\partial F_{zk}}{\partial w_{k}} \frac{\partial w_{k}}{\partial \mu_{k}} \frac{\partial \mu_{k}}{\partial c_{k}}$$
(3-23)

Since 
$$E_p = \frac{1}{2} (F_z - \hat{F}_z)^2$$
,  $\frac{\partial E_p}{\partial F_z} = (F_z - \hat{F}_z) = e_p$ , (3-24)

$$F_{z} = F_{z1} + F_{z2} + \dots + F_{zH}, \frac{\partial F_{z}}{\partial F_{zk}} = 1,$$
 (3-25)

$$F_{zk} = \frac{w_k}{w_1 + w_2 + \dots + w_{q^2}} (p_k \cdot i + q_k \cdot r + c_k), \quad \frac{\partial F_{zk}}{\partial w_k} = \frac{(p_k \cdot i + q_k \cdot r + c_k) - F_z}{w_1 + w_2 + \dots + w_{q^2}}, \quad (3-26)$$

and 
$$\frac{\partial w_k}{\partial \mu_k} = \frac{w_k}{\mu_k},$$
 (3-27)

then substituting Eq. (3-24), (3-25), (3-26) and (3-27) into the Eq. (3-23), the gradient are then obtained as

$$\frac{\partial E_p}{\partial a_k} = e_p \frac{(p_k \cdot i + q_k \cdot r + c_k) - F_Z}{w_1 + w_2 + \cdots + w_{q^2}} \frac{\partial \mu_k}{\partial a_k}$$

$$\frac{\partial E_p}{\partial b_k} = e_p \frac{(p_k \cdot i + q_k \cdot r + c_k) - F_Z}{w_1 + w_2 + \cdots + w_{q^2}} \frac{\partial \mu_k}{\partial b_k}$$

$$\frac{\partial E_p}{\partial c_k} = e_p \frac{(p_k \cdot i + q_k \cdot r + c_k) - F_Z}{w_1 + w_2 + \cdots + w_{q^2}} \frac{\partial \mu_k}{\partial c_k}$$
(3-28)

It is necessary to determine the initial value of the premise parameters before performing the hybrid learning process. The Matlab ANFIS tool box is used to train the network in off-line. The experimental results are collected as the training data set. In this study, there are three membership functions for each input. The actual structure of the ANFIS is shown in the Figure 3-5.



Figure 3-5 Actual structure of ANFIS.



Figure 3-6 Comparison between ANFIS model and Experimental results.

The Figure 3-6 demonstrates the comparison between the experimental results and two predicted output of two ANFIS model. As shown in Figure 3-6, the ANFIS model matches well with the experimental results. This model will be used throughout for controller design and optimization in the following chapter.

### 3.2 Experimental Setup of Twisted String Actuation

In this section, an experimental platform was adopted to validate the Kinetostatic model of twisted string actuator. The configuration of the platform is shown in Figure 3-7. One string with 7 strands (1 neutral and 6 effective) is used to drive the external load. The string is connected to two holders that were fixed to the shaft of the DC motor (Pololu motor with a 64 CPR Encoder) and the load, as shown in Figure 3-8. The diameter of each strand of adopted tactical cables is 1.2 mm, and the original length of a single strand is 500 mm. The DC motor used to twist the string is driven by a PWM mode L298 H-Bridge. The AC712 current sensor is used to measure the current of the armature of the DC motor. The WXY linear encoder is used to measure the linear displacements of the twisted-string, the resolution of linear encoder is 0.04 mm. The SHIMPO FGV-XY20 is used to measure the tensile force generated by the twisted-string, the resolution of the force gauge 0.1 N. The ACS712 hall current sensor is used to measure the current through DC motor's armature. To acquire the movement and

force data, a desktop PC with a Pentium IV 2.4 GHz and 1 GB memory is used for data acquisition. Two data acquisition I/O cards (PCI-6221 and PCI-6601) manufactured by National Instruments are used to generate command signals to motor drivers and to acquire signals from the linear encoder, current sensor and force sensor. The data acquisition program is developed using MATLAB Simulink with xPC Toolbox [81]. The sampling rate is 100 Hz. To guarantee enough current and minimize the noise, a regulated DC power supply (HY3000), which can offer continuous output voltage (0 - 30V) and output current (0 - 5A), is adopted as the power supply for the experimental platform.



Figure 3-7 Configuration of experimental setup of twisted string actuation.



Figure 3-8 Strand holder mounted on DC motor.

Figure 3-9 describes the comparison of the theoretical model, Eq. (3-4), and experimental data acquired in the experimental platform. Note the discrepancy between the theoretical model and the experimental data is obvious. There are several factors that affect the correct of the model. First, the theoretical model is a continuous function with respect to motor revolutions, which means that even very small motor revolutions still can generate linear displacements of twisted string. However, it is obvious that the twisted string would keep its length at original length until the motor turns 5 revolutions. The reason is that there are gaps among different strands due to the design of the strand holder as shown in Figure 3-8. Before producing the linear displacement the gap among strands needs to be decreased to zero. Second, the varying diameter of each strand due to the external force also influences the correct of the model.



Figure 3-9 Relationship between motor revolutions and stroke.

Figure 3-10 demonstrates the relationship between motor revolutions and force generated by the twisted string actuator. Since the force transmission of the twisted string actuator plays the key role in controller design, the constant current source is used to drive the DC motor, which can ensure the torque generated by DC motor stays unchanged during the whole process of twisting string. The data, as shown in Figure 3-10, is obtained when the constant current source provides 0.7 Amp current. From the Figure 3-10 (a), it is obvious that the force transmission keeps increasing along with motor revolutions when the torque generated by the DC motor is constant, which means the force transmission efficiency can be increased by pre-twisting string to specify operation range. The two stages of transition are shown in the Figure 3-10 (a), the two phase of the force transmission are: initial phase, which the twisted string actuator does not generate the linear displacement, and normal phase. It is easy to make



Figure 3-10 Relationship between motor revolutions and Force.

a conclusion that the force transmission will increase to infinity based on simply observing the Figure 3-10 (a), however with the increasing of motor revolutions the parts of torque generated by the DC motor used to overcome the friction among strands will also be increased. From the experiment results of the Figure 3-10 (b) and 3-10 (c), we notice that the force and linear displacements generated by the DC motor achieve balance, which means the torque generated by the DC motor is equal to the friction and external load. Once the twisted string actuator achieves balance the torque that can be transferred to external load become less even the force transmission is still increasing.

Figure 3-11 describes the comparison of the theoretical model, Eq. (3-7), and experimental data. By comparing the degree of similarity between shapes of the theoretical model and experimental data, the theoretical model, Eq. (3-7), does not match what we expect for the relationship between motor revolutions and force transmission. The similar phenomenon occurs, the force transmission irregularly changing before the twisted string actuator starting to generate the linear displacement, because the parts of torque generated by the motor is used to make the each strand be tightly together. Except the gaps among each strand in the initial state due to the mechanical, unmolded dynamics and unstructured uncertainties also deteriorate the correct of the model. In such a robotic application, both accuracy and consistency of an actuator are important. The consistency of an actuator is defined by the terms of repeatability and hysteresis. Figure 3-12 illustrates the whole data set. As shown in Figure 3-12, the trajectory of the same set of data for contraction and extension do not match, meaning there is a hysteresis error in the system. With the existing hardware, different loadings were tested for the consistency. The experimental outcome demonstrates that the twisted string actuator has better repeatability for contraction than extension. Meanwhile, the hysteresis effect is less significant if the twisted string actuation is operated more than 15 revolutions of the motor shaft.



Figure 3-11 Relationship between motor revolutions and Force transmission.



Figure 3-12 Comparison between extension and contraction of twisted string.

Based upon above experimental results and above analysis, it is clear that if we pre-twist string before using the twisted string actuator to drive the external load, the hysteresis effect, time delay and the irregular force transmission in the initial phase can be avoided, which is very important for designing a high accuracy controller of motion systems. Five experiments are performed with the goal of verifying this assumption. In this five experiments, the string is pre-twisted five turns before the experiments starting. Figure 3-13 illustrates the relationship among motor revolutions, force and linear displacements generated by the twisted string actuator after pre-twisting string. Both the force and linear displacement curves express better continuity than no pre-twisted curves. Therefore, pre-twisting the string is a method, efficient and easy to achieve, to reduce the complexity of controller design.





Although this Kinetostatic model described in the last section can reflect the relationship between axial forces, the torque generated by motor and string length, there are still obvious differences between mathematical modeling and actual performance of a twisted string. The main reason for the differences can be caused by not considering of friction among strands, the gap among different strands, the elasticity of single strands and deformation of individual strands. Therefore, the twisted string actuation system falls into the class of the real world processes that are not easy to obtain the mathematical representation. In a real-time control system, there are a large variety of plants, similar to twisted string actuator, whose system model are highly nonlinear and cannot be physically derived. For these nonlinear plants, the experiments can be performed to extract approximate model by using sophisticated curve fitting approaches.



Figure 3-14 Relationship among force transmission ratio and linear displacements after pre-twisting.

In this study, the approximation of the static characteristic of the actuator is made using the Matlab toolbox called Curve Fitting Tool. The armature current of DC motor which is proportional to torque is the input of the motion system in this study. Therefore, the function of force transmission and linear displacements are required to design the controller. The approximate function of force transmission and linear displacement are found by curve fitting approach. Then the force generated by the twisted string actuator can be predicted by this approximate function. The approximate function are exponentiation function:

$$K_{\tau}(x) = \alpha + \beta \cdot e^{-\gamma x} \tag{3-29}$$

The values of the coefficients of the function are:

$$\alpha = 0.001894, \beta = 0.005465, \gamma = 0.2139 \tag{3-30}$$

Approximated static characteristic of the twisted string actuator according to the Eq. (3-9) is shown on Figure 3-15.



Figure 3-15 Approximated static characteristic of the twisted string actuator.



Figure 3-16 Experimental results and curve fitting of force transmission function.

Before applying the approximating function obtained by the curve fitting in the controller design, it is necessary to check its accuracy. The first plot of the Figure 3-10 shows the comparison between the curve fitting results of force obtained by using DC motor's armature current to multiply the torque constant and the experimental results directly obtained by the force sensor. From the second plot of Figure 3-16 it is apparent that the error between the curve fitting and experimental results is small, and the discrepancy between them can be compensated in the process of the controller design.

In this study, curve fitting approach and adaptive neuro-fuzzy inference system (ANFIS) [82] are separately adopted to model the transmission ration function of twisted string actuation system. The curve fitting approach is used to provide an approximate model for the adaptive robust controller. The adaptive neuro-fuzzy inference system (ANFIS) is used to optimize the different gain values in the adaptive robust controller by simulation.

# **CHAPTER 4:** Design of the exoskeleton rehabilitation robot device

### 4.1 Mechanism design of the exoskeleton rehabilitation robot device

The aim of the upper limb exoskeleton rehabilitation robot designed in this study is not only to help stroke patients to perform rehabilitation activities but also to provide assistive force for the elderly people and patients with impairments. To meet requirements of the desired function, the mechanism of the robot requires the following criteria and guidelines:

- For the purpose of wearing the robot, the mechanical structure of the robot must be as simple as possible with the premise of meeting the requirement of strength criteria;
- 2. The mechanically adjustable design needs to be adopted in order to make the robot compatible for the patients with different body size; and
- 3. The most important concern of a wearable robot is its safety. To guarantee the safety of a rehabilitation robot while operating a hard mechanical stop and the corresponding safety control strategy, and power-off protection mechanism need to be taken into consideration.

To provide good mobility, the robotic device also needs to be capable of reproducing arm motions with enough DOF's and lightweight. Mechanical mechanism and force actuators need to be carefully designed to provide the largest force with least weight. Based on clinical observation, the full return of arm motion is noted mostly in the elbow, less in the hand, and least in the shoulder [83] for stroke patients and the total number of DOF's of should/elbow/wrist is 10 [84]. Among the three joints, shoulder needs to provide the largest force in many cases. In this design, we focus on the joints of elbow and shoulder. Statistically, 56% of stroke occurs in left hemisphere of the brain, which can affect the right side of body. Stroke that occurs in right hemisphere of brain affects more on language skills rather than motion skills. Thus, the targeted arm is right-hand arm. Figure 4-1 illustrates the preliminary design of the assistive robotic device developed at West Virginia University. The major concern of designing such an assistive robotic system is to minimize the load on the user's arm. The total weight of this design is only ~530 g including both mechanisms for the upper and lower arms. Instead of placing everything on user's arm, batteries, actuators, and supporting circuits are packed in a backpack, in which the users can carry these accessories on their backs.



Figure 4-1 Preliminary design of the assistive robotic device [85].

Based on the above consideration, the upper-limb exoskeleton robot designed in this study is consisted of two independent joints (pulley) driven by two pair of antagonized twisted string actuators respectively. The elbow and shoulder joints are put in motion by the two pulley located at the elbow and shoulder joint separately, as show in Figure 4-2. Figure 4-3 shows that the distances between two pulleys can be increased or reduced by adding an adjustable telescopic structure that can be freely fixed using bolts. The adjustable telescopic structure is located between the shoulder joint and elbow joint. By observing the ranges of shoulder pronation and supination, upper arm rotation and elbow flexion and extension,

mechanical hard stops are used to limit the range of the pulley's motion to protect users. The limited rotation range of the pulley located at shoulder joint is 0 to 120 degrees. The limited rotation range of pulley located at elbow joint is 0 to 135 degrees. Two foam cushions are installed at two adjustable telescopic structures with straps, which are used to fasten the robotic arm and stroke patients' arm together. The foam cushions make it easy for the stroke patients to feel comfortable.



Figure 4-2 Schematic of upper limb rehabilitation exoskeleton robot.



Figure 4-3 Schematic of adjustable telescopic structure of upper limb rehabilitation exoskeleton robot.

Since weight is one of the most important factors affecting the feasibility of the rehabilitation exoskeleton robot, the twisted string actuation, and robot's joints are separated and placed at different locations, which provides a benefit to select different length strands to twist based on the individual stroke patient. In this study, the exoskeleton robot has the detachable design; the twisted string, DC motor, battery and microprocessor are installed on a backpack, which can be worn on the stroke patients' back; the Bowden cables, which is similar to bike brake cable, are used to connected actuators on the backpack to exoskeleton robot arm. Figure 4-4 illustrates the schematics of the backpack from different perspectives.

As shown in Figure 4-4, the four DC motor are fixed at the one side of the belt of the backpack, the battery and microprocessor are fixed at the other side of the belt of backpack. Figure 4-5 and Figure 4-6 illustrate the 3D exterior rendering of stroke patients wearing the exoskeleton robot rehabilitation device.



Figure 4-4 Backpack of exoskeleton robot.



Figure 4-5 Backpack on the stroke patient.



### Figure 4-6 Exoskeleton rehabilitation robot device on the stroke patient.

The 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 the strands and cannot generate the dilation force when untwisting the strands. According to the observation of the rehabilitation process of stroke patients, the stroke patients' arm cannot be back to the initial position once they turn their elbow over a certain angle. Due to the lack of the muscle flexibility, the stroke patients must rely on the gravity or external force to make their arm go back to the initial position. In order to allow the stroke patients who use the exoskeleton robot to exercise their arm in two ways, the robot needs to have the ability to provide both the tensile and dilation force to user's arm. Twisted string based antagonistic actuator designed in this study consists of two twisted string actuators which are acting against themselves and the resultant position of the actuator is given by equilibrium of their tensile forces according to different stroke displacement of twisted string actuators as shown in Figure 4-7. Operation of the present twisted string based antagonistic actuators is realized by length of the string decreasing in the one twisted string actuator and simultaneous length of the string increasing in the other twisted string actuator. The linear displacements of the two twisted strings are cross-coupled with each other. Unsynchronized motion can make the joint to be slack or non-uniform movement of the exoskeleton robot arm as shown in Figure 4-8. To compensate for both tracking and synchronization response cross-coupled control strategy [67], was applied to the twisted-string actuators to compensate for both tracking and synchronization response.

With the current design of our proposed robotic system, four twisted strings are employed to drive two pulleys located at elbow and shoulder joints of the robot in a clockwise or counterclockwise direction.



Figure 4-7 Configuration of a joint of a dual twisted-string actuation system in a robotic

system.





# 4.2 Cross-Coupling Dynamics

The controller is responsible for the coordinating the motion of the different twisted string actuators in the exoskeleton robot device. In orde to improve the synchronization performance among different twisted string actuators, the appropriate synchronization strategy is required to be employed in practice. There are three synchronization strategies, tandem, master-slave, and cross-coupling, have been widely employed in industrial systems. The scheme of the tandem control approach is shown in Figure 4-9. In this scheme, the synchronization errors between two axes are minimized through dynamics matching and reference generator. Reference trajectory generator produces the different trajectory for each axis based on the difference of dynamics of the two axes and the traditional control approached, such as PID, can be implemented in each axis' control loop for accurate synchronization.



### Figure 4-9 Scheme of tandem control.

The scheme of the master-slave control approach is shown in Figure 4-10. In master-slave control, one axis of the multi-axis is considered as the master from which output is fed back as a reference input to the rest of the axes of the system which are considered as the slave. The goal of the synchronization can be achieved through slave following the master. The master-slave synchronization strategy is normally suitable for the system in which each axis has different dynamics and responses. The slowest axis is often be considered as the master; other axes are considered as the slave, which can ensure the slaves have the ability to follow the master. Due to the nature of the tandem control approach, any disturbances acting on one axis in a multi-axis system can not be compensated by the other axes. A similar phenomenon has occurred in the case of master-slave control in which any disturbances

acting on a slave axis in a multi-axis system can not be compensated by the master axis and other slave axes. In order to deal with the disturbances acting on different axes in a multi-axis system, the cross-coupling synchronization strategy was developed to achieve accurate motion synchronization in a multi-axis system. In cross-coupling control strategy, the controller minimizes both the synchronization errors among the different axes and position-tracking error of single axis at the same time to achieve the requirements of synchronization and tracking. The scheme of the cross-coupling control approach is shown in Figure 4-11. In the proposed exoskeleton robot device all twisted string actuators are constituted of the same length, material and number of strands, and will have the same dynamics.



Figure 4-10 Scheme of master-slave control.



Figure 4-11 Scheme of cross-coupling control.

# 4.3 Experimental setup



Figure 4-12 Schematic of experimental setup (single joint).

In this study, a single joint of robotic system was adopted to test the tracking and synchronization of a dual twisted-string actuation system. The configuration of the actuation is shown in Figure 4-13. Two strings with 7 strands (1 neutral and 6 effective) were used to extend and contract the attached strings to drive the rotational movement of the joint. The diameter of each strand of adopted tactical cables is 1.2 mm, and the original length of a

single strand is 500 mm. A single string was connected at both ends with one end driven by a gear motor and one end connected to a belt that rotates the joint. The joint shown in Figure 4-13 is an XL Type Aluminum timing belt pulley with 50 teeth. The pitch of the belt, which is made of rubber, is 5.08mm and the tooth height is 1.5mm. The length of the belt can be elongated due to its elasticity. The extensional or contractile string was connected to two holders which were fixed to the shaft of the DC motor (Pololu motor with a 64 CPR Encoder), as shown in Figure 3-8.

To acquire the movement data, a desktop PC with a Pentium IV 2.4 GHz CPU was used for data acquisition. Two data acquisition I/O cards (PCI-6221 and PCI-6601) manufactured by National Instruments were used to send out command signals to the motor driver and to acquire signals generated by the optical encoders of the motor and the pulley. The data acquisition programs and control algorithm were developed using MATLAB Simulink with xPC Toolbox. The schematic of the a single joint of the robotic system in Figure 4-12.



Figure 4-13 Configuration of experimental setup (single joint).

## 4.4 Linear Controllers – PID, PID-CC, LQR

From Figure 3-9, it is clear that the relationship between motor revolutions and length variation of a twisted string is not linear. A complete extension/contraction stroke of the string actuation is a hysteresis process. To reduce the hysteresis, it is required to pre-twist the strings to tension them slightly instead of operating the strings from a slack situation. Since

the linear actuators have an uncomplicated control design, making them easier to implement in the robotic system. The decreased complexity of control design for linear actuators allows the overall design to be simpler and more stable, which drives us to locate the approximately linear operational range from the whole operational range of the twisted string actuator. The twisted string actuator is intended as a linear actuator in this approximately linear operational range. From the theoretical model and the experimental results, the best approximately linear operational range for a 60 mm stroke is suggested to be between 15 to 35 revolutions for the adopted tactical cable.

To design an adequate controller for the actuator, it is necessary to derive the mathematical models of the adopted DC motors. Figure 4-14 illustrates the block diagram of a DC motor with the consideration of power saturation. The armature inductance of the adopted DC motor is small enough to be ignored. Thus, the transfer function of the DC motor can be approximated by a first order system. The nominal linear model of the motor can be obtained by using system identification techniques in the frequency domain. The physical properties of the plant are listed in Table 4-1.



Figure 4-14 Block diagram of the DC motors used to drive the twisted strings.

Symbol	Description	
J	Effective rotational inertia	
$B_{v}$	Effective viscous friction coefficient	
R	Motor armature resistance	
$K_t$	Motor torque constant	
$K_b$	Motor back-EMF constant	
ρ	Transmission ratio of twisted string	

Table 4-1 Symbols of parameters used to model driving motor.

With this configuration, the transfer function from voltage command to the linear displacement of the twisted string actuator can be represented by the following transfer function, which is

$$G(s) = \frac{K_0 \rho}{s(1+\tau s)},\tag{4-1}$$

where

$$K_0 = \frac{K_t}{B_v R + K_t K_b} \quad \text{and} \quad \tau = \frac{JR}{B_v R + K_t K_b}. \tag{4-2}$$

In this case,  $\rho$  is the transmission ratio of between the revolutions of DC motor and displacement of the attached twisted string by performing a linear curve fitting within the selected operational range.

#### 4.4.1 PID Controller

The controllers used in this section are two PID controllers designed using the pole-placement method that compensates individual actuators independently. The controller synthesis is based on the model of the driving motor and the approximation of the twisted-string actuating mechanism. Since the actuation systems of both sides adopt the same type of driving motors and tactical cable, the transfer functions used to design the controllers are the same. The trajectories of the individual motors are dispatched by a trajectory planning function depending on the desired angular position of the joint. Figure 4-8 illustrates the
block diagram of the two twisted string actuators. With the simulation results, the designed controller can provide adequate tracking performance for a single actuator if the desired poles are carefully selected. With this design, the two actuators are assumed to be two independent SISO systems that do not interfere each other. The two actuators are designed to track the opposite trajectory input, where a trajectory planning function is introduced to transfer a predefined angular movement to a trajectory of linear displacement. Instead of tracking the angular position of the driving motors, two linear encoders that measuring the linear displacement of the two strings were adopted. In the experiments, the joint driven by the twisted string actuating system is designed to track both sinusoidal and trapezoid signals which are converted to the linear displacement of two actuators.

To evaluate the performance of the controlled outcome, two types of motions were used: sinusoidal and trapezoidal movements. For the sinusoidal motion, the amplitude of the periodic displacement is 60 mm and the duration of each cycle is 4 sec. For the trapezoidal motion, each string extends/contracts 40 mm within 2 sec from rest. It then stops for 2 sec and then returns to its starting position within another 2 sec. Figure 4-15 and Figure 4-16 illustrates the experimental results of the sinusoidal and trapezoidal motion tests, including angular positions and synchronization errors. From the experiment results, it is clear that such an approach yields a poor performance in both tracking and synchronization without considering the cross-coupling dynamics between the two actuators. The maximum synchronization error can be as high as 20 mm for sinusoidal motion and 6 mm for trapezoidal motion. The huge error of both tracking and synchronization can be caused by disturbances created by the string actuator in the opposite direction. Such disturbances can also generate unevenly distributed loads on the twisted-string actuators at both sides. Therefore, the required amounts of contraction/extension for the two sides are often not the same, which can change the required revolution of the driving DC motors.



Figure 4-15 Experimental outcome of PID controller (sinusoidal motion): (a) comparison between reference trajectory and actual trajectories of left actuator; (b) comparison between reference trajectory and actual trajectories of right actuator; (c)

synchronization errors between left and right twisted actuators.



Figure 4-16 Experimental outcome of PID controller (trapezoidal motion): (a) comparison between reference trajectory and actual trajectories of left actuator; (b) comparison between reference trajectory and actual trajectories of right actuator; (c) synchronization errors between left and right twisted actuators.

## 4.4.2 PID controller with feedback of cross-coupling dynamics (PID-FC)

As the motor on one side starts to twist one string, the string on the other side must untwist. Unsynchronized motion can make the hysteresis phenomenon of linear displacement of the string more severe so that both tracking and synchronization performance are degraded. In this case, regular PID controller cannot compensate for trajectory tracking precisely due to the disturbance and unsynchronized motion. In order to improve both tracking and synchronization performance, cross-coupling dynamics in the multi-axial system needs to be added to individual feedback loops. With the consideration of cross-coupling dynamics, the control structure can be modified as shown in Figure 4-11. The cross-coupled dynamics adopted in this study is the deviation of the linear positions of both actuators. The control goal of such a control strategy is to make both tracking and synchronization errors converging to zero. In this case, synchronization error between two actuators is introduced as input disturbance in the feedback loop with a cross-coupling gain. Then the cross-coupling gain can be adjusted according to its importance. High gain compensation of synchronization error due to the cross-coupled relationship can compensate for the synchronization performance. However, the trade-off is that the performance of tracking controller of individual axes can be less effective since more control effort is focused on synchronization.

Figure 4-17 and 4-18 illustrate the controlled results of PID-FC. It is clear that the maximal synchronization error can be dramatically reduced to only less than 3 mm for the desired sinusoidal motion. It also improves tracking error to less than 1 mm for trapezoidal trajectory as well. However, the gain of cross-coupling control system design is generally a trial-and-error process in which the design parameters are determined by comparing the performance among different parameters. However, the gain of cross-coupling controller is determined based on a trial-and-error process instead of a systematic approach. Thus, it can be difficult to fine tune controlled performance.



Figure 4-17 Experimental outcome of PID-FC controller (sinusoidal motion): (a) comparison between reference trajectory and actual trajectories of left actuator; (b) comparison between reference trajectory and actual trajectories of right actuator; (c)

synchronization errors between left and right twisted actuators.



Figure 4-18 Experimental outcome of PID-FC controller (trapezoidal motion): (a) comparison between reference trajectory and actual trajectories of left actuator; (b) comparison between reference trajectory and actual trajectories of right actuator; (c) synchronization errors between left and right twisted actuators.

## 4.4.3 LQR controller with feedback of cross-coupling dynamics (LQR-FC)

To compensate for trajectory tracking of multi-axial systems, linear quadratic optimal control law has been found to be an effective mathematic algorithm that minimizes a cost function with specified design goals. Cheng et al. [36] proposed a generalized method in which multi-axis motion systems can be represented as multiple sub-systems with dynamics coupling among them. To apply LQR method to such a cross-coupled dual-axis system, the state-space presentation of two actuators is given by

$$\Delta \mathbf{d}(t) = \mathbf{A} \Delta \mathbf{d}(t) + \mathbf{B} \mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C} \Delta \mathbf{d}(t)$$
(4-3)

where *t* is time,  $\Delta d(t)$  and u(t) are the vectors of state variables and control inputs, **A** and **B** are system matrices in the following form

$$\begin{bmatrix} \dot{\Delta} d_{1}(t) \\ \vdots \\ \Delta d_{1}(t) \\ \vdots \\ \Delta d_{2}(t) \\ \vdots \\ \Delta d_{2}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{\tau_{1}} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{\tau_{2}} \end{bmatrix} \begin{bmatrix} \Delta d_{1}(t) \\ \vdots \\ \Delta d_{1}(t) \\ \Delta d_{2}(t) \\ \vdots \\ \Delta d_{2}(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{K_{01}}{\tau_{1}} & 0 \\ 0 & 0 \\ 0 & \frac{K_{02}}{\tau_{2}} \end{bmatrix} \begin{bmatrix} u_{1}(t) \\ u_{2}(t) \end{bmatrix}, \quad (4-4)$$

and

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$
 (4-5)

In this study, the actuators are two independent motors. Thus, the coefficients of the matrices  $A_{ij}$  and  $b_{ij}$  ( $i\neq j$ ) are all null matrices.  $\tau_1$ ,  $\tau_2$ ,  $K_{01}$ , and  $K_{02}$  are the physical parameters of individual twisted-string actuators. Since the goal is to regulate the linear displacement,  $\Delta d_1$  and  $\Delta d_2$ , as well as the difference between these two displacements, a linear quadratic (LQ) optimal control method is applied to reduce the synchronization error. In order to achieve accurate tracking and synchronization performance, the performance criteria, in which both tracking and synchronization errors need to be considered, can be expressed as the following equation,

$$J_{p} = \int_{t_{0}}^{t_{f}} \left[\sum_{i=1}^{2} \varepsilon_{i} \cdot \Delta d_{i} \left(t\right)^{2} + \sum_{j=1}^{2} \delta_{j} u_{j}^{2} + \eta (\Delta d_{1} - \Delta d_{2})^{2}\right] dt$$
(4-6)

where  $\varepsilon_i$ ,  $\delta_j$ , and  $\eta$  are the parameters that need to be determined. To consider the feedback cross-coupling dynamics, the deviation of the linear displacements of both actuators are used. The synchronization error can then be transformed to the representation of state variables by selecting the specified **Q** matrix. The **Q** matrix can be defined as

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 0 & 0 & 0 & 0 \\ 0 & 1 & -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \varepsilon_1 & 0 & 0 & 0 \\ 0 & \varepsilon_2 & 0 & 0 \\ 0 & 0 & \eta & 0 \\ 0 & 0 & \eta & 0 \\ 0 & 0 & 0 & \eta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 \end{bmatrix},$$
(4-7)

and the **R** matrix is

$$\mathbf{R} = \begin{bmatrix} \delta_1 & 0\\ 0 & \delta_2 \end{bmatrix}. \tag{4-8}$$

Based on the performance index Eq. (4-6) the Hamiltonian equations can be obtained:

$$\mathbf{H} = \frac{1}{2}d(t_f)^T \mathbf{H}d(t_f) + \frac{1}{2}u(t)^T \mathbf{R}u(t) + P^T(\mathbf{A}x + \mathbf{B}u)$$
(4-9)

Based on the necessary conditions the control input can be derived:

$$\frac{\partial \mathbf{H}}{\partial u} = 0 \Longrightarrow \mathbf{R}u(t) + \mathbf{B}^T P(t) = 0 \Longrightarrow u(t) = -\mathbf{R}^{-1}\mathbf{B}^T P(t)$$
(4-10)

$$\frac{\partial \mathbf{H}}{\partial \Delta d} = -\dot{P} \Longrightarrow \dot{P} = -(\mathbf{Q}(\Delta d(t) - ref(t)) + \mathbf{A}^T P(t))$$
(4-11)

where ref(t) is the reference input. In order to acquire the co-state variable P(t) we assume,

$$P(t) = P(t)d(t) \tag{4-12}$$

Then we differentiate the Eq. (4-12)

$$\dot{P}(t) = \dot{P}(t)d(t) + P(t)\dot{d}(t)$$
(4-13)

Combining the Eq. (4-10) and Eq. (4-11) the control input can be obtained:

$$\dot{P} = \mathbf{A}^T P + P \mathbf{A} - P \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T P + \mathbf{Q}$$
(4-14)

$$u(t) = -(\mathbf{R}^{-1}\mathbf{B}^T P(t))\Delta d(t)$$
(4-15)

The selection of weight matrices Q and R in linear quadratic optimal control is very

important and they have a direct effect on controller's performance. In general,  $\mathbf{Q}$  and  $\mathbf{R}$  matrices are determined based on experience. Therefore, the controller might not be truly optimal or possibly be unstable. To ensure the best performance, genetic algorithm (GA) based on the Darwin's theories was used to search for the coefficients of  $\mathbf{Q}$  and  $\mathbf{R}$  matrices for global optimal. In this study, the  $\mathbf{Q}$  and  $\mathbf{R}$  matrices are

$$\mathbf{Q} = \begin{bmatrix} 4975 & 0 & -2189 & 0 \\ 0 & 0 & 0 & 0 \\ -2189 & 0 & 4975 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 0.4978 & 0 \\ 0 & 0.4978 \end{bmatrix}.$$
 (4-16)

Once the Q and R matrices have been determined, the optimal control law is given by

$$\mathbf{K} = \begin{bmatrix} 97.38 & 4.23 & -22.57 & -0.53 \\ -22.57 & -0.53 & 97.38 & 4.23 \end{bmatrix}.$$
 (4-17)

The control law is derived to drive the output, or the state, to 0. However, in this case, the objective of the controller is to drive the joint to track a reference input. For  $r_{ref}(t)$  is the reference input, the goal is to make  $y(t) \rightarrow r_{ref}(t)$  as  $t \rightarrow \infty$ . To achieve this objective, a feedforward controller is required. We first design a steady state vector  $d_{ss} = N_x r_{ref}$  for constant reference input  $r_{ref}$ , in which  $N_x$  converts the reference inputs  $r_{ref}$  into the desired values of the states. The new control law can be defined as:

$$\mathbf{u} = -\mathbf{K}(\mathbf{d} - \mathbf{d}_{ss}) + \mathbf{u}_{ss}, \qquad (4-18)$$

where  $u_{ss} = N_u r_{ref}$  is the steady state control input to maintain  $\Delta d$  at  $\Delta d_{ss}$ . To find  $N_x$  and  $N_u$ , we define the desired steady state relationships:

$$\Delta \dot{\mathbf{d}}_{ss}(t) = \mathbf{A} \cdot \Delta \mathbf{d}_{ss}(t) + \mathbf{B} \cdot \mathbf{u}_{ss}(t)$$
  
=  $(\mathbf{A}\mathbf{N}_{\mathbf{x}} + \mathbf{B}\mathbf{N}_{\mathbf{u}})\mathbf{r}_{ref} = 0$ , (4-19)  
 $\mathbf{y}(t) = \mathbf{C} \cdot \Delta \mathbf{d}_{ss}(t) = \mathbf{C}\mathbf{N}_{\mathbf{x}}\mathbf{r}_{ref}$ 

and need to solve:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{N}_{\mathbf{x}} \\ \mathbf{N}_{u} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}.$$
 (4-20)

Then, we can rewrite the control law to be:

$$\mathbf{u} = \mathbf{g} \cdot \mathbf{r} - \mathbf{K} \cdot \mathbf{d} \tag{4-21}$$

where  $\mathbf{g}$  is the feedforward gain. That is

$$\mathbf{g} = \mathbf{N}_{\mathbf{u}} + \mathbf{K}\mathbf{N}_{\mathbf{x}} = \begin{bmatrix} \mathbf{K} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix}.$$
 (4-22)

However, this feedforward controller only account for the constant reference input because the  $N_x$  and  $N_u$  rely on modeling of the plant which might not be accurate and LQR is a kind of optimal controllers which is not focusing on disturbance rejection. To track the time varying reference input and reject disturbances, integral control can be added into the system. This can be achieved by augmenting the system by the integrator state  $\Delta d_{I_c}$ 

$$\begin{bmatrix} \dot{\mathbf{d}}_{\mathbf{I}}(\mathbf{t}) \\ \dot{\mathbf{d}}(\mathbf{t}) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -\mathbf{C} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{d}_{\mathbf{I}}(\mathbf{t}) \\ \Delta \mathbf{d}(\mathbf{t}) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B} \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \mathbf{r}_{ref}(\mathbf{t}), \qquad (4-23)$$

Since the system in this study is the Type-1 system, the  $N_u$  is equal to zero. Therefore the final control law is:

$$\mathbf{u} = -[\mathbf{K}_{\mathrm{I}} \quad \mathbf{K}] \begin{bmatrix} \mathbf{d}_{\mathrm{I}} \\ \mathbf{d} \end{bmatrix} + \mathbf{K} \mathbf{N}_{\mathrm{x}} \mathbf{r}(\mathbf{t}), \qquad (4-24)$$

where  $K_I$  is integral gain. With this technique, the disturbance can be rejected effectively as well [15]. Figure 4-19 and 4-20 demonstrate the experimental results of tracking performance for both sinusoidal and trapezoidal motions. It is clear that the LQR-FC controller can improve both the transient and steady-state tracking performance over that of the PID controllers. With the LQR-FC approach, the synchronization performance can be improved as well. It can be observed that the maximum synchronization error can be restricted to within 2 mm, which is slightly better than PID-FC controller. It also demonstrates better tracking performance than PID-FC controller.

Based on different types of controllers, the tracking performances of the dual twisted-string actuating systems are different as well. Among the three controllers, LQR-FC has the smallest synchronization and tracking errors. Table I lists the comparison of both tracking and synchronization errors (root-mean-square and maximum errors) among all three controllers. It is clear that the PID controller is not able to provide satisfactory tracking performance

while the other two types of controllers, PID-FC and LQR-FC, have significantly smaller errors. It is also noticed that all three controlled systems have a slightly delayed response. Table 4-2 lists the time delays of three controllers. Longer time delay can yield greater tracking error. Among all the controllers, the LQR-FC has a significantly smaller time delay than the other two controllers. Similarly, the PID-FC uses the cross-coupling feedback to obtain a much smaller synchronization error than that of the plain PID controller. This also implies that PID-FC can have better disturbance rejection if the disturbance is caused by the connected actuators.



Figure 4-19 Experimental outcome of LQR-FC controller (sinusoidal motion): (a) comparison between reference trajectory and actual trajectories of left actuator; (b) comparison between reference trajectory and actual trajectories of right actuator; (c)

synchronization errors between left and right actuators.



Figure 4-20 Experimental outcome of LQR-FC controller (trapezoidal motion): (a) comparison between reference trajectory and actual trajectories of left actuator; (b) comparison between reference trajectory and actual trajectories of right actuator; (c) synchronization errors between left and right actuators.

 $\frac{RMSE (mm)}{Sync error} \frac{RMSE (mm)}{Tracking error}$   $PID without cross coupling 3.88 \frac{4.03 (L)}{6.92 (R)}$   $PID-FC \frac{0.62}{5.32 (R)}$   $LQR-FC \frac{0.63}{1.66 (L)}$ 

Table 4-2 Comparison of performance of controllers of sinusoidal motion.



Figure 4-21 (a) Control effort of left twisted string actuator (sine); (b) Control effort of right twisted string actuator (sine); (c) Control effort of left twisted string actuator (trapezoid);(d) Control effort of right twisted string actuator (trapezoid).

To compare the performance between PID-FC and LQR-FC controllers, Figure 4-22 illustrates tracking and synchronization errors of three full-cycle movements of these two controllers. The arrows indicate the directions of displacements of string actuation. Both controllers demonstrate good repeatability of tracking outcomes. Although the synchronization errors are not significantly different, the LQR-FC demonstrates a better tracking performance with its smaller root mean-square error (RMSE). This can be due to the greater delay of PID-FC caused by the structure of different controllers.

From the previous sections, it is clear that the PID controller alone does not provide satisfactory tracking performance. To test disturbance rejection, only PID-FC and LQR-FC controllers were considered. In particular, disturbance rejection has been considered and integrated while synthesizing the LQR-FC controller. In this section, two types of disturbance were applied to the dual twisted-string actuation joint. They are: 1) a fixed 500 g loading, and 2) a 500 g loading attached to the mechanism when the linear displacement of the string actuation is at a specific location. In this case, the displacement was at 40 mm.



Figure 4-22 Comparison of tracking and synchronization errors of PID-FC and



Figure 4-23 Tracking and Synchronization performance of LQR controller with 500g fixed loading and periodic loading.



Figure 4-24 Comparison of tracking and synchronization errors of PID-FC and LQR-FC controlled results with 500g fixed loading and periodic loading.

Figure 4-23 illustrates the experimental results of the controllers with a fixed 500 g, or 4.91 N, loading attached to the joint on the left actuator while moving, and a 500 g loading attached to the left actuator when the displacement of the string actuation is 40 mm. From the experimental results, the tracking errors of the individual axes were affected slightly. There is no significant difference of tracking and synchronization performance between LQR-FC and PID-FC controllers. However, the PID-FC does have a longer time delay which yields greater tracking error. This error can be as large as 12 mm on one side. And the disturbance caused by the attached mass at a specific location does not generate a clear impact on the movement. Figure 4-24 demonstrates the position based errors of both controllers for a complete cycle. In this figure, it is clear that LQR-FC can have a better tracking performance which also results in a slightly better synchronization performance due to its smaller time delay when a fixed loading or periodic loading are attached. Synchronization errors started to increase as a 500g loading was attached at a specific displacement for both LQR-FC and PID-FC controllers. Such errors became smaller as the 500 g was removed from this specific location.

From the experimental results, the LQR-FC demonstrates a smaller deviation from the desired trajectories for both types of disturbances. With the current setup, both tracking and

synchronization performance can be guaranteed regardless of the load disturbances as tested here. Table 4-3 lists the performance of both controllers under two different scenarios.

	Delay (sec)
PID without cross	0.37 (L)
coupling	0.37 (R)
PID-FC	0.14 (L)
	0.28 (R)
LQR-FC	0.08 (L)
	0.02 (R)

Table 4-3 Time delay of the three controllers of sinusoidal motion.

Table 4-4 Con	nparison of the	disturbance	rejection bety	ween PID-FC	and LOR-	FC

	Maximum	Maximum	
	tracking error	synchronization error	
	(mm)	(mm)	
PID-FC with a fixed 500g	11.55(L)	1.02	
loading	12.56(R)	1.25	
PID-FC with a 500g	12.07(L)	2.70	
loading attached at 40 mm	13.01(R)	2.19	
LQR-FC with a fixed	4.32(L)	2.97	
500g loading	5.01(R)	5.87	
LQR-FC with a 500g	4.02(1.)	6.12	
loading attached at 40 mm	4.02(L)	0.12	

# Controllers with a 500g loading.

## 4.5 Summary

To precisely compensate for the tracking performance of a single axial system, both PID and LQR algorithms can generate satisfactory results. However, the controlled results can be fairly inaccurate if the cross-coupled relationships between individual actuators are not carefully compensated for a multi-axial system. This paper demonstrates two effective approaches to eliminate the deviation caused by such a relationship. With the consideration of cross-coupling dynamics, the proposed PID-FC and LQR-FC controllers can greatly reduce the synchronization errors between the two axes. If the delay is especially pernicious in the system, the PID-FC controller cannot satisfy the requirements for tracking performance of a twisted-string actuation system. Thus, the LQR-FC controller can provide a more adequate performance for both tracking and synchronization.

# **CHAPTER 5:** ARC control synthesis of the twisted string actuation

## **5.1 Nonlinearity of the Twisted String Actuation**

Twisted string actuation has the advantages of compact size, flexible mechanism, and light weight at locations where actuations are required. However, though twisted-string actuation has a lot of advantages, its unique properties of nonlinearity including hysteresis, friction, operational dead-zone, and poor repeatability can deteriorate the desired control performance. Improper control methods without consideration of nonlinearities can lead to limit cycles or instability.

Nonlinearity in the field of engineering technology and the natural world is a very common phenomenon. With the development of modern control technology and theories, researchers focus on the understanding dynamic behaviors and designing different compensator for specific nonlinear systems. For the proposed robotic system, the major nonlinearities is the dead-zones of adopted DC motors, friction, and other nonlinear factors. For instance, pulse width modulation (PWM) actuators are adopted to drive DC voltage across their armatures. The relation between motor speed and duty cycle of PWM is not linear, which can add more nonlinearity to the system. The friction between strands is generated by twisting and untwisting the cable. Such kind of friction between strands also adds additional nonlinearity to the system. With the above analysis, the twisted string actuation cannot be assumed to be linear system. Linear controller cannot properly deal with the uncertainty in parameter and nonlinear factors of the system. Without compensating for these nonlinear factors, the controlled precision of the robotic system cannot always be guaranteed. In order to control this nonlinear system, an adaptive robust controller was applied to twisted string actuator. To ensure the transient and tracking performance of the system, the projection type parameters adaption mechanism and robust control term are employed here to both improve the performance and stability of the system.

## 5.2 Synthesis of Non-linear Controller –ARC (Displacement control)

The bi-directional movement of the antagonistic twisted-strings system proposed in this study relies on the coupled twisting/untwisting motion of individual actuators. The tracking accuracy of separate actuators also affect the synchronization performance of the both actuators, which means that the adopted controller needs to respond to both tracking and synchronization errors simultaneously. To deal with both nonlinear factors and synchronization between individual actuators, adaptive robust control (ARC) was adopted for such a motion system. The structure of ARC controller used for multi-axial systems has been proposed by Cheng et al [36, 37, 86]. With this framework, a multi-axial motion system can achieve excellent tracking and synchronization performance in normal working conditions. However, the previous work only demonstrates the adaption of varying parameters, some nonlinear phenomena are not fully discussed. This study proposes an approach to compensate for an important nonlinear issue: the dead-zone effect near zero-velocity. To deal with the tracking errors of stick-slip motion phenomenon caused by dead-zone, an adaptive robust control algorithm with the consideration of dead-zone effect is studied. Additional adaption mechanisms are added to the ARC structure to provide accurate tracking and synchronization. With all these considerations, the dynamics of a twisted-string actuated joint can be described by:

$$\begin{aligned} x_{i,1} &= \Delta d \\ \dot{x}_{i,1} &= x_2 \\ \dot{x}_{i,2} &= -\frac{B_v R + K_t}{JR} x_{i,2} + \frac{K_t \rho}{JR} (w_i(t) - F_{i,f}(t)) \\ w_i(t) &= D(u_i(t)) \end{aligned}$$
(5-1)

where *i* denotes the *i*<sup>th</sup> axis,  $x_{i,1}$  represents the linear displacements of a twisted-string actuator,  $u_i(t)$  is the voltage control command with the compensation of dead-zone phenomenon,  $w_i(t)$ denotes the consideration of both dead-zone behavior and the control command to motor,  $F_{i,f}(t)$  is bounded input disturbance due to the change in the angle of the joint as a function of time and static friction of the twisted-strings. The dead zone  $D(u_i(t))$ , as shown in Figure 5-1(a), can be represented as follows:

$$w_{i}(t) = D(u_{i}(t)) = \begin{cases} \vartheta(u_{i}(t) - b_{r}), \text{ if } u_{i}(t) \ge b_{r} \\ 0, \quad \text{ if } b_{l} \le u_{i}(t) \le b_{r} \\ \vartheta(u_{i}(t) - b_{l}), \text{ if } u_{i}(t) \le b_{l} \end{cases}$$
(5-2)

#### 5.2.1 Dead-Zone Compensation

To compensate the dead-zone phenomenon, two assumptions were made.

**Assumption 1:** The dead-zone has the same slopes in both positive and negative directions of the horizontal axis.

Assumption 2: Though the function of dead-zone  $w_i(t)$  can be obtained, the dead-zone parameters  $\vartheta$ ,  $b_r$ , and  $b_l$  are bounded within known regions, which are  $\vartheta \in [\vartheta_{min}, \vartheta_{max}]$ ,  $b_r \in [b_{rmin}, b_{rmax}]$ ,  $b_l \in [b_{lmin}, b_{lmax}]$  with  $\vartheta > 0$ ,  $b_r > 0$ ,  $b_l < 0$ .

Thus, Eq. (5-2) can be re-defined as:

$$w_i(t) = D(u_i(t)) = \vartheta u_i(t) + \delta(u_i(t))$$



Figure 5-1 (a) Model of a dead-zone required to initialize a motor and (b) the inverse function of the dead-zone.

where  $\delta(u_i(t))$  is a function further defined as:

$$\delta(u_i(t)) = \begin{cases} -\vartheta b_r, & \text{if } u_i(t) \ge b_r \\ -\vartheta u_i(t), & \text{if } b_l \le u_i(t) \le b_r \\ -\vartheta b_l, & \text{if } u_i(t) \le b_l \end{cases}$$
(5-3)

To compensate for the dead-zones of individual motors, one simple rule of non-linearity needs to be integrated into the controller. If the voltage applied to a motor is smaller than a threshold, the motor cannot be started. To implement an anti-dead-zone controller, an inverse function of dead-zone is used. The output of the dead-zone inverse function, as shown in Figure 5-1(*b*),  $v(t) = D_I(w_i(t))$  can be defined as

$$D_{I}(w_{i}(t)) = \psi_{+}\left(u_{i}(t)\right)\frac{u_{i}(t) + \mathcal{P}b_{r}}{\mathcal{P}} + \psi_{-}\left(u_{i}(t)\right)\frac{u_{i}(t) + \mathcal{P}b_{l}}{\mathcal{P}},$$
(5-4)

where the functions  $\psi_+$  and  $\psi_-$  are defined as

$$\psi_{+}(u_{i}(t)) = \begin{cases} 1, \text{ if } u_{i}(t) > 0\\ 0, \text{ if } u_{i}(t) \le 0 \end{cases}, \text{ and}$$
(5-5)

$$\psi_{-}(u_{i}(t)) = \begin{cases} 1, \text{ if } u_{i}(t) < 0\\ 0, \text{ if } u_{i}(t) \ge 0 \end{cases}.$$
(5-6)

Thus, the inverse function of dead-zone can be added back to Eq. (5-1) if the range of dead-zone is identified. The equation can be modified as

$$\dot{x}_{i,1} = x_{i,2}$$

$$\dot{x}_{i,2} = -\frac{B_v R + K_i}{JR} x_{i,2} + \frac{K_i \rho}{JR} (D_i(w_i(t)) - F_{i,f}(t)) \cdot w_i(t) = D(u_i(t))$$
(5-7)

Thus, the dead-zone phenomenon can be compensated with the inverse-functions. However, though the behavior of a dead-zone due to friction can be anticipated, the range of a dead-zone can only be estimated based on corresponding parameters, such as static friction and ratio between the revolutions of driving motor and the change of extension/contraction. The unstructured uncertainties of the twisted-string can still be the main obstacle of synthesizing an accurate tracking controller. Thus, an adaptive robust control algorithm (ARC) with considerations of dead-zone effect, bounded nonlinearity, and cross-coupling dynamics is proposed. Based on Eq. (5-2) and (5-4), the state-space representation of such a system becomes

$$\dot{x}_{i,1} = x_{i,2}$$

$$\dot{x}_{i,2} = \frac{-\xi_{i,2}}{\xi_{i,1}} x_{i,2} + \frac{1}{\xi_{i,1}} \left( \psi_+(u_i(t))u_i(t) + \psi_+(u_i(t))\xi_{i,3} + \psi_-(u_i(t))u_i(t) + \psi_-(u_i(t))\xi_{i,4} - \xi_{i,5} \right)$$
(5-8)

where the coefficients  $\xi_{i,1} = \vartheta JR/(K_t\rho)$ ,  $\xi_{i,2} = \vartheta(B_vR+K_t)/(K_t\rho)$ ,  $\xi_{i,3} = \vartheta b_r$ ,  $\xi_{i,4} = \vartheta b_l$ , and  $\xi_{i,5} = \vartheta F_{i,f}(t)$  are all unknown but can be bounded for individual twisted-string systems. The bounded regions of these five coefficients can be defined by  $\xi_{i,j} \in [\xi_{i,j\min}, \xi_{i,j\max}]$ . Since the two actuators of a single joint are physically connected with each other, linear displacement of one motor is affected by the actuated motion of the other cable, or cross-coupled.

In order to precisely control the rotational movement of a joint and to synchronize the linear motions of cables on both sides, cross-coupled dynamics becomes an important issue while designing controllers. In particular, the relationship between the revolution of motor shafts and the variation of linear displacement  $\Delta d$  of the two cables cannot be the same while one is extending and the other one is contracting. Thus, as one motor starts to twist a cable, the other one must untwist at a different rate. Poor synchronization can yield vibration of the joint. With this consideration, it is necessary to carefully plan the rotational movements of both motors according to desired angular trajectories of the joint. Controller with inadequate synchronizing scheme between the two twisted-string actuators can yield the whole system being unstable. It has been proved that both tracking and synchronization errors can be decay to the error of a multi-axial system can be decayed to zero if the cross-coupled relationship is feedback to the controllers of individual axes. With adaptive robust control algorithm and backstepping procedures, Cheng et al. [6] have demonstrated the feasibility and excellent performance of an ARC controller for compensating a multi-axis system. The control law of such a controller is

$$Z_{i} = \dot{e}_{i} + K_{i,p} e_{i} + K_{i,s} e_{s}$$
(5-9)

where  $e_i$  is the tracking error,  $K_{i,p}$  is the feedback gain,  $e_s$  is the synchronization error,  $K_{i,s}$  is the synchronization gain. Both  $K_{i,p}$  and  $K_{i,s}$  are greater than 0. This implies that making  $Z_i$ converging to 0 is equivalent to making both  $e_i$  and  $e_s$  converging to 0. Thus, the objective of the controller design is to make  $Z_i$  as small as possible. Since the  $\xi_{i,3}$  and  $\xi_{i,4}$  exchange with each other when the control signal  $u_i(t)$  change from negative to positive, two controllers then can be employed, each of which provides satisfactory control with  $\xi_{i,3}$  and  $\xi_{i,4}$ , respectively. By differentiating  $Z_i$  with respect to time, we have

$$\dot{Z}_{i} = \ddot{x}_{i,1d} - \ddot{x}_{i,1} + K_{i,p}\dot{e}_{i} + K_{i,s}\dot{e}_{s} 
= \ddot{x}_{i,1d} + \frac{\xi_{i,2}}{\xi_{i,1}}x_{i,2} - \frac{1}{\xi_{i,1}}(u_{i}(t) + \xi_{i,3} - \xi_{i,5}) + K_{i,p}\dot{e}_{i} + K_{i,s}\dot{e}_{s}$$
(5-10)

The control input  $u_i(t)$  can then be defined

$$u_{i}(t) = K_{i}Z_{i} + \xi_{i,1}\ddot{x}_{i,1d} + \xi_{i,2}x_{i,2} - \xi_{i,3} + \xi_{i,5} + \xi_{i,1}K_{i,p}\dot{e}_{i} + \xi_{i,1}K_{i,s}\dot{e}_{s}$$
(5-11)

where  $K_i$  is always greater than 0. This control input guarantees that  $e_i$  and  $e_s$  approach to zero asymptotically by canceling the parameters and uncertain nonlinearity and anticipating the nature of the desired signal. However, it should be noted that the controller requires explicit knowledge of the parameters of the system according to Eq. (5-11). Therefore, the proposed desired ARC laws have the same form as Eq. (5-11) but with the estimation of parameters. That is:

$$u_{i,a}(t) = K_i Z_i + \hat{\xi}_{i,1} \ddot{x}_{i,1d} + \hat{\xi}_{i,2} x_{i,2} - \hat{\xi}_{i,3} + \hat{\xi}_{i,5} + \hat{\xi}_{i,1} K_{i,p} \dot{e}_i + \hat{\xi}_{i,1} K_{i,s} \dot{e}_s$$
(5-12)

where  $\hat{\xi}_{i,j}$  are the five time-varying parameters on the *i*<sup>th</sup> axis that are the estimations of the real parameters  $\xi_{i,j}$ . Substituting  $u_{i,a}(t)$  in Eq. (5-12) into (5-13), it becomes

$$\dot{Z}_{i} = \frac{1}{\xi_{i,1}} \left[ -K_{i}Z_{i} + (\xi_{i,2} - \hat{\xi}_{i,2})x_{i,2} - (\xi_{i,3} - \hat{\xi}_{i,3}) + (\xi_{i,5} - \hat{\xi}_{i,5}) + (\xi_{i,5} - \hat{\xi}_{i,5}) + (\xi_{i,1} - \hat{\xi}_{i,1})(K_{i,p}\dot{e}_{i} + K_{i,s}\dot{e}_{s} + \ddot{x}_{i,ld}) \right]$$
(5-13)

To verify the stability of the proposed controller, a Lyapunov function  $V_{i,a}$  can be defined as

$$V_{i,a} = \frac{1}{2} \left( \xi_{i,1} Z_i^2 + \frac{1}{\lambda_{i,1}} \tilde{\xi}_{i,1}^2 + \frac{1}{\lambda_{i,2}} \tilde{\xi}_{i,2}^2 + \frac{1}{\lambda_{i,3}} \tilde{\xi}_{i,3}^2 + \frac{1}{\lambda_{i,3}} \tilde{\xi}_{i,5}^2 \right)$$
(5-14)

where  $\xi_{i,j} = \xi_{i,j} - \hat{\xi}_{i,j}$  represent the estimation errors of the corresponding parameters.  $\lambda_{i,1}$ ,  $\lambda_{i,2}$ ,  $\lambda_{i,3}$  and  $\lambda_{i,5}$  are positive constants used to evaluate the selected Lyapunov function. Since the  $\xi_{i,1}$  is positive in the twisted-string system,  $V_{i,a}$  is always positive definite. The derivative of this Lyapunov function with respect to time becomes

The adaptation learning mechanisms for adjusting the parameters through the adaptation gains $\lambda_{i,1}$ ,  $\lambda_{i,2}$ ,  $\lambda_{i,3}$  and  $\lambda_{i,5}$  and the past information , which need to be designed to guarantee the derivation of the Lyapunov function defined above. That is

$$\tilde{\xi}_{i,1} = -\lambda_{i,1} Z_i (\dot{x}_{i,1d} + K_{i,p} \dot{e}_i + K_{i,s} \dot{e}_s), 
\dot{\xi}_{i,2} = -\lambda_{i,2} Z_i x_{i,2}, \dot{\xi}_{i,3} = \lambda_{i,3} Z_i 
\dot{\xi}_{i,5} = -\lambda_{i,5} Z_i$$
(5-16)

Substituting Equ. (4-16) into (4-15), the time derivative of  $V_{i,a}$  becomes

$$\dot{V}_{i,a} = -K_i Z_i^2 \tag{5-17}$$

Since  $K_i$  is positive, the time derivative of  $V_{i,a}$  is negative definite. With Barbalat's lemma,  $e_i$ , and  $e_s$  can always decay to zero asymptotically as time approaches infinite.

## 5.2.2 Nonlinearity and Uncertainty structure compensation

The adaptive control mechanism compensates the unknown parameters and structured uncertainties. However, the controller proposed does not take consideration of the unstructured uncertainties but just processes of the adjustable parameters. As the result, even a small disturbance may lead the adaptive process to instability and large transient tracking error. The deterministic robust control (DRC) with the projection of parameter adaptation has also been investigated by some researchers to attenuate the effect of unstructured uncertainties and improve the transient performance [14], [15]. The ARC controller can be given by

$$u_{i}(t) = u_{i,a}(t) + u_{i,s}(t)$$
(5-18)

In Eq. (5-18),  $u_{i,a}(t)$  is generated by the adaptive control law governed by Eq. (5-12), and  $u_{i,s}(t)$  is compensated by the robust control law to stabilize the system. Substituting Eq. (5-18) into (5-13), it becomes

For the deterministic robust control, the robust term  $u_{i,s}(t)$  should be any continuous function which satisfies the following conditions:

$$(1) Z_{i} \Big[ -u_{i,s} + \tilde{\xi}_{1,i} (\ddot{x}_{i,1d} + K_{i,p} \dot{e}_{i} + K_{i,s} \dot{e}_{avg}) + \tilde{\xi}_{i,2} x_{i,2} - \tilde{\xi}_{i,3} + \tilde{\xi}_{i,5} \Big] \leq \varepsilon_{i}$$

$$(5-20)$$

$$(2) \quad Z_{i} \cdot u_{i,s} \geq 0$$

where  $\varepsilon_i$  is a positive design parameter that can be set arbitrarily small. Furthermore, the Lyapunov function can be defined as following

$$V_{i,s} = \frac{1}{2}\xi_{i,1}Z_i^2.$$
 (5-21)

The derivative of the  $V_{i,s}$  becomes

$$\mathbf{\hat{V}}_{i,s} = -K_i Z_i^2 + Z_i \cdot \left[ -u_{i,s} + \tilde{\xi}_{1,i} (\ddot{x}_{i,1d} + K_{i,p} \dot{e}_i + K_{i,s} \dot{e}_s) + \tilde{\xi}_{i,2} x_{i,2} - \tilde{\xi}_{i,3} + \tilde{\xi}_{i,5} \right].$$
(5-22)

From the conditions in Eq. (5-20), the derivative of the  $V_{i,s}$  satisfies

The following inequality then can be derived by integrating Eq. (5-23). That is

$$V_{i,s}(t) \le \exp\left(\frac{-2K_i t}{\xi_{i,1\max}}\right) \cdot V_{i,s}(0) + \int_0^t \exp\left(\frac{-2K_i (t-v)}{\xi_{i,1\max}}\right) \varepsilon_i dv$$
(5-24)

where *t* is time variant,  $\xi_{i,1max}$  is the upper bound of  $\xi_{i,1}$ . According to Bellman-Gronwall, the positive definite  $V_{i,s}$  is bounded by

$$V_{i,s}(t) \le \exp\left(\frac{-2K_i t}{\xi_{i,1\max}}\right) V_{i,s}(0) + \frac{\varepsilon_i \cdot \xi_{i,1\max}}{2K_i} \left[1 - \exp\left(\frac{-2K_i t}{\xi_{i,1\max}}\right)\right].$$
(5-25)

Based the relationship between  $Z_i$  and  $e_i$ ,  $e_i$  is bounded by

$$e_i(t) \le \exp\left(-K_{i,p}t\right) \cdot e_i(0) + \frac{Z_i}{K_{i,p}} \left[1 - \exp\left(-K_{i,p}t\right)\right]$$
 (5-26)

From the above inequality, the tracking error of twisted-string system exponentially approaches to a ball with converging rate  $K_{i,p}$ . Within a finite duration, the asymptotic output tracking is achieved. The controller  $u_{i,s}(t)$  satisfying Equ. (5-20) can be designed by using the method of completion of square, which is

$$u_{i,s} = hZ_i \tag{5-27}$$

Substituting Eq. (5-27) into (5-20), the inequality becomes

$$\begin{bmatrix}
-\frac{1}{4}hZ_{i}^{2} + \tilde{\xi}_{i,1}(\ddot{x}_{i,1d} + K_{i,p}\dot{e}_{i} + K_{i,s}\dot{e}_{s})Z_{i} - \frac{\left(\tilde{\xi}_{i,1} \cdot (\ddot{x}_{i,1d} + K_{i,p} \cdot \dot{e}_{i} + K_{i,s} \cdot \dot{e}_{s})\right)^{2}}{h} - \frac{1}{4}h \cdot Z_{i}^{2} + \tilde{\xi}_{i,2} \cdot x_{i,2} \cdot Z_{i} - \frac{\left(\tilde{\xi}_{i,2} \cdot x_{i,2}\right)^{2}}{h} - \frac{1}{4}h \cdot Z_{i}^{2} + \tilde{\theta}_{i,5} \cdot Z_{i} - \frac{\tilde{\xi}_{i,5}^{2}}{h} - \frac{1}{4}h \cdot Z_{i}^{2} - \tilde{\xi}_{i,3} \cdot Z_{i} - \frac{\tilde{\xi}_{i,3}^{2}}{h}\right] + \frac{\left(\tilde{\theta}_{i,1} \cdot (\ddot{x}_{i,1d} + K_{i,p} \cdot \dot{e}_{i} + K_{i,s} \cdot \dot{e}_{avg})\right)^{2} + \left(\tilde{\theta}_{i,2} \cdot x_{i,2}\right)^{2} + \left(\tilde{\theta}_{i,3}\right)^{2} + \left(\tilde{\theta}_{i,5}\right)^{2}}{h} \leq \varepsilon_{i}$$
(5-28)

One example of h satisfying the Equ. (4-28) can be found in the following way, which is

$$h \ge \frac{\xi_{i,1m}^{2} \cdot \left(\ddot{x}_{i,1d} + K_{i,p} \cdot \dot{e}_{i} + K_{i,s} \cdot \dot{e}_{avg}\right)^{2}}{\varepsilon_{1}} + \frac{\left(\xi_{i,2m} \cdot x_{i}\right)^{2}}{\varepsilon_{2}} + \frac{\left(\xi_{i,3m}\right)^{2}}{\varepsilon_{3}} + \frac{\left(\xi_{i,5m}\right)^{2}}{\varepsilon_{4}},$$
(5-29)

where  $\varepsilon_i = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4$  is a design parameters of the *i*<sup>th</sup> twisted-string actuator,  $\xi_{i,1m} = \xi_{i,1max} - \xi_{i,1min}$ ,  $\xi_{i,2m} = \xi_{i,2max} - \xi_{i,2min}$ ,  $\xi_{i,3m} = \xi_{i,3max} - \xi_{i,3min}$  and  $\xi_{i,5m} = \xi_{i,5max} - \xi_{i,5min}$  are the known bounded ranges for the unknown parameters. To ensure all parameters are always bounded in the adjusting process, a simple discontinuous projection type adaption mechanism is employed here and thus does not affect the closed-loop system stability, which is

$$proj(\bullet) = \begin{cases} 0 & if \\ 0 & if$$

Eventually, the ARC control can be obtained as follows

$$u_{i}(t) = -hZ_{i} + KZ_{i} + \hat{\xi}_{i,1}(\ddot{x}_{i,1d} + K_{i,p} e_{i} + K_{i,s} e_{s}) + \hat{\xi}_{i,2}x_{i,2} - \hat{\xi}_{i,3} + \hat{\xi}_{i,5}.$$
(5-31)

Similarly, the same design process can be adopted when the  $u_i(t)$  is positive. This ARC controller combing two controllers for different  $u_i(t)$ , both positive and negative. By taking the advantage of both adaptive and robust control commands, system stability and asymptotic tracking performance can be ensured.

Figure 5-2 and 5-3 demonstrate the experimental results of tracking performance for both sinusoidal and trapezoidal motions. It is clear that the ARC controller can improve both the transient and steady-state tracking performance over that of the LQR-FC and PID-FC controllers. From the Figure 5-5 and 5-6, it is clear that the ARC approach can significantly reduce the tracking error when comparing to LQR-FC and ARC. With the ARC approach, the synchronization performance can be improved as well. It can be observed that the maximum synchronization error can be restricted to within 1 mm, which is better than LQR-FC and PID-FC controller.



Figure 5-2 Experimental outcome of ARC controller (sinusoidal motion): (a) comparison between reference trajectory and actual trajectories of left actuator; (b) comparison between reference trajectory and actual trajectories of right actuator; (c) synchronization errors between left and right actuators.



Figure 5-3 Experimental outcome of LQR-FC controller (trapezoidal motion): (a) comparison between reference trajectory and actual trajectories of left actuator; (b) comparison between reference trajectory and actual trajectories of right actuator; (c)



Figure 5-4 (a) Control effort of twisted string actuator (sine); (b) Control effort of twisted string actuator (trapezoid).



Figure 5-5 Comparison of tracking and synchronization errors of PID-FC and LQR-FC controllers.

Table 5-1 Comparison of performance of the three controllers of sinusoidal motion.

	Maximum	Maximum	
	tracking error	synchronization	
	(mm)	error (mm)	
PID-FC	17.95(L)	3.18	
	20.05(R)		
LQR-FC	3.40 (L)	2.85	
	3.35 (L)		
ARC	1.02(L)	0.6	
	1.419(R)	0.0	



Figure 5-6 Comparison of tracking and synchronization errors of ARC and LQR-FC controllers.

In this study, three different types of the controller have been studied. The experimental validation has been used to investigate the performance of each controller. The performance of LQR-FC and PID-FC was first compared. An ARC controller was then compared with LQR-FC. With the experimental results, the following conclusions can be drawn from the above Figures and Table:

- 1. All three controllers, PID-FC, LQR-FC, and ARC can reduce synchronization errors in the process of bi-direction movements, while conventional PID controller is possibly resulting in joint shifting.
- 2. The tracking and synchronization errors of both PID-FC and LQR-FC controller reach their maximum value when the joint begins to change motion direction. The

largest overshoot produced by PID-FC is due to the fact that low gain is adopted and integrator makes the control effort arrive at the maximum when the joint changes its direction of motion. This indicates that the control effort needs to be small enough at this time.

- 3. The LQR-FC has better disturbance rejection ability than PID-FC with the incorporation of a feedforward controller.
- 4. The ARC shows the best tracking and synchronization performance among three controllers. The projection mapping can be used to keep the parameters within a known bounded convex set, which ensures the robustness of this controller.

## 5.3 Synthesis of Non-linear Controller –ARC (Force control)

The safety consideration is of great importance for rehabilitation robot, which must interact with the patients' body. In an attempt to ensure the safety of users, several critical aspects need to be considered, such as active force and corresponding displacement generated by the robot. The aims of the rehabilitation robot discussed in this study are to provide active force and produce movement of the upper-extremity. Therefore, the displacement control is not enough to ensure the safety of the rehabilitation robot. Along with the strands twist, the inner friction of the twisted string keeps increasing, which means producing the same stroke required twisted string actuator generating more force than beginning. When the stroke of the actuator reaches the upper limit, the force to reduce the error between the desired and actual trajectories could be very larger even if the error is minor. The minor error between the desired and actual trajectories is not so important when patients do rehabilitation with the help of the robot, however at the same time the large force would hurt the patients. Therefore, it is necessary to incorporate the force control strategy into the displacement control strategy. The force control strategy not only limits the force output of the actuator but also increases or decreases the force directly once the external load is increasing or decreasing. In this section we first develop and validate the performance of the ARC controller (force) based the test platform described in last chapter, Figure 3-3. The free-body diagram for the load is shown in Figure 5-7. The equation of motion of the mass *m* is:

$$m\ddot{x} = F - mg - F_L - f + \Delta \tag{5-32}$$

where *m* is the mass of the load, *F* is the tensile force generated by the twisted-string actuator, *x* is the displacement of the load,  $F_L$  is the external force generated by the spring, which can be measured by the force sensor, *f* is the friction among strands,  $\Delta$  is unstructured uncertainties of the model. Based on the equation (3-29), we can rewrite the equation of motion (5-32) as follows:

$$m\ddot{x} = \frac{\tau_{l}}{K_{\tau}(x_{1})} - mg - F_{L} - f + \Delta$$

$$mg + F_{L}$$

$$F$$
(5-33)

Figure 5-7 Free-body diagram for the load.

where  $\tau_L$  is the torque generated by the DC motor, which is equal:

$$\tau_I = K_t I \tag{5-34}$$

where  $K_t$  is the torque constant of the DC motor, *I* is the armature current of the DC motor. Based on (5-33) and (5-34), we can build the dynamic model of the twisted string actuation system in a state-space form as follows:

$$\dot{x}_{1} = x_{2}$$

$$\dot{x}_{2} = \frac{1}{m} \left( \frac{\tau_{l}}{K_{\tau}(x_{1})} - mg - F_{L} - f + \Delta \right)$$
(5-35)

where  $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T = \begin{bmatrix} x & \dot{x} \end{bmatrix}$  is the system state. In the traditional velocity and position control design considering only the mechanical dynamics with inertial load but ignoring the electrical dynamics. With the aim of high tracking performance, it is required to consider the electrical

system and the connection between the electrical system and mechanical systems. The connection between the electrical system and mechanical systems is shown in the Figure 5-8.



Figure 5-8 Block Diagram for Twisted String Actuation.

From the Figure 5-8, it is obvious that the control input voltage doesn't directly influence the displacement of the load. To cope with this, the concept of backstepping [87] can be employed to deal with the first equation in Eq. (5-35) not including the control input. We can rewrite the state-space model by integrating the electrical subsystem as follows:

$$\dot{x}_{1} = x_{2}$$

$$\dot{x}_{2} = \frac{1}{m} \left( \frac{K_{t}I}{K_{\tau}(x_{1})} - mg - F_{L} - f + \Delta \right)$$

$$L\dot{I} = u - RI - K_{b}\omega$$
(5-36)

where *L* is the constant rotor inductance, *R* is the constant rotor resistance,  $K_b$  is the constant back-emf coefficient,  $\omega$  is the speed of DC motor and *u* is the input control voltage.

#### 5.3.1 Adaptive Robust Controller (Force) Design

Define a switch function, which is original from sliding mode control, as:

$$Z = \dot{e} + k_1 e \,, e = x_{1d} - x_1 \tag{5-37}$$

where *e* is the error between desired trajectory and the actual trajectory of external load's displacement,  $k_1$  positive feedback gain. The transfer function from *Z* to *e* is:

$$E(s) = \frac{1}{s + k_1} Z(s)$$
(5-38)

Since the Z(s) is linear system and  $k_I$  is positive, the e(t) will be converging zero if the Z(t) and e(t) are converges to zero. Then the  $\dot{e}(t)$  will be converging to zero if the Z(t) and e(t) are converging to zero. View the state variable I as an independent virtual "input". The first step of the controller design is to ensure this virtual "input" can both stabilize the second equation in Eq. (5-36) and let the Z(t) eventually go to zero, which guarantees that the tacking error e(t)converges to zero. Differentiating Z with respect to time leads to

$$\dot{Z} = \ddot{x}_{1d} - \ddot{x}_1 + k_1 \dot{e}$$
 (5-39)

By combing the second equation in Eq. (5-36), we have

$$\dot{Z} = \ddot{x}_{1d} - \frac{1}{m} \left( \frac{K_t I}{K_\tau(x_1)} - mg - F_L - f + \Delta \right) + k_1 \dot{e}$$
(5-40)

Rewrite Eq. (5-40) as:

$$\frac{K_{\tau}(x_{1})m}{K_{t}}\dot{Z} = \frac{K_{\tau}(x_{1})m}{K_{t}}\ddot{x}_{1d} - (I - \frac{K_{\tau}(x_{1})(mg + F_{L} + f - \Delta)}{K_{t}}) + \frac{K_{\tau}(x_{1})m}{K_{t}}k_{1}\dot{e}$$
(5-41)

Substitute Eq. (3-29) into Eq. (5-41):

$$\frac{(\alpha + \beta e^{-\gamma x_{1}})m}{K_{t}}\dot{Z} = \frac{(\alpha + \beta e^{-\gamma x_{1}})m}{K_{t}}\ddot{x}_{1d} - (I - \frac{(\alpha + \beta e^{-\gamma x_{1}})(mg + F_{L} + f - \Delta)}{K_{t}}) + \frac{(\alpha + \beta e^{-\gamma x_{1}})m}{K_{t}}k_{1}\dot{e}$$
(5-42)

Rewrite Eq. (5-42) as:

$$p(x_1)\dot{Z} = \mathbf{W}\,\mathbf{\theta} - I \tag{5-43}$$

where 
$$\mathbf{W} = [\frac{m}{K_t}(\ddot{x}_{1d} + k_1\dot{e} + g) \frac{m}{K_t}e^{-\gamma x_1}(\ddot{x}_{1d} + k_1\dot{e} + g) \frac{1}{K_t} \frac{e^{-\gamma x_1}}{K_t}],$$
  
 $\boldsymbol{\theta} = [\alpha \ \beta \ \alpha (F_L + f - \Delta) \ \beta (F_L + f - \Delta)]^T, \ p(x_1) = \frac{(a + be^{-\gamma x_1})m}{K_t}.$ 

The adaptive robust control mechanism compensates the unknown parameters and structured uncertainties by adjusting parameters online. However, even a small disturbance may lead the parameters adaptive process to instability and large transient tracking error. By integrating the essence of the deterministic robust control (DRC) the adaptive robust controller can attenuate the effect of unstructured uncertainties and improve the transient performance. Therefore, the desired virtual input *I* can be given by:

$$I = I_a + I_s$$

$$I_s = I_{s1} + I_{s2}$$
(5-44)

where  $I_a$  is adjustable feedforward controller used to compensate the model by parameter adaption,  $I_s$  is the robust controller in which  $I_{s1}$  is linear feedback to stabilize the nominal model of the twisted string actuation and  $I_{s2}$  is used to attenuate the parameter estimation error and nonlinear terms of the model. In the Eq. (5-43) the vector  $\boldsymbol{\Theta}$  are all unknown parameters that need to be estimated. Therefore, the vector  $\boldsymbol{\Theta}$  will be replaced by the  $\hat{\boldsymbol{\Theta}}$ , the estimated value of the vector  $\boldsymbol{\Theta}$ , in the controller. Since the  $K_{\tau}(x_1)$  and  $K_t$  are positive function and constants, the Lyapunov candidate functions can be defined as following:

$$V = \frac{1}{2} p(x_1) Z^2 + \frac{1}{2} \tilde{\boldsymbol{\Theta}}^{\mathrm{T}} \tilde{\boldsymbol{\Theta}}$$
(5-45)

where  $\tilde{\Theta}$  is the difference between the real value of the vector  $\Theta$  and  $\hat{\Theta}$ , the estimated value of the vector. Taking the derivative of Eq. (5-45) gives:

$$\dot{V} = \frac{1}{2}\dot{p}(x_1)Z^2 + p(x_1)Z\dot{Z} + \tilde{\boldsymbol{\theta}}^{\mathrm{T}}\dot{\tilde{\boldsymbol{\theta}}}$$
(5-46)

Since the  $\dot{\tilde{\theta}} = \dot{\theta} \cdot \dot{\hat{\theta}}$  and  $\Theta$  is constant, we have

$$\dot{V} = \frac{1}{2}\dot{p}(x_1)Z^2 + p(x_1)Z\dot{Z} - \tilde{\boldsymbol{\theta}}^{\mathrm{T}}\dot{\hat{\boldsymbol{\theta}}}$$
(5-47)

Substituting Eq. (5-43) into Eq. (5-47) gives:

$$\dot{V} = \frac{1}{2}\dot{p}(x_1)Z^2 + Z(\mathbf{W}\,\mathbf{\theta} - I) - \tilde{\mathbf{\theta}}^{\mathrm{T}}\dot{\hat{\mathbf{\theta}}}$$
(5-48)

Since the  $I_a$  is adjustable feedforward controller used to compensate the model by parameter adaption, the  $I_a$  is:

$$I_a = \mathbf{W}\,\hat{\mathbf{\theta}} + \frac{1}{2}\,\dot{p}(x_1)Z\tag{5-49}$$

Letting  $I_{s1} = KZ$ , we get

$$\dot{V} = -KZ^2 + Z\mathbf{W}\tilde{\boldsymbol{\theta}} - \tilde{\boldsymbol{\theta}}^{\mathrm{T}}\hat{\boldsymbol{\theta}}$$
(5-50)

where K is a positive feedback gain. In order to ensure the stability of the system, the derivative of V should be less or equal to zero by letting

$$Z\mathbf{W}\tilde{\boldsymbol{\theta}} - \tilde{\boldsymbol{\theta}}^{\mathrm{T}}\hat{\boldsymbol{\theta}} = 0 \tag{5-51}$$

Solving the Eq. (5-51) gives:

$$\hat{\boldsymbol{\theta}} = \mathbf{W}^{\mathrm{T}} \boldsymbol{Z} \tag{5-52}$$

where  $\dot{\hat{\boldsymbol{\theta}}} = \begin{bmatrix} \dot{\hat{\theta}}_1 & \dot{\hat{\theta}}_2 & \dot{\hat{\theta}}_3 & \dot{\hat{\theta}}_4 \end{bmatrix}$ . Therefore the adaptation mechanisms for adjusting parameters are:

$$\dot{\hat{\theta}}_{1} = \frac{m}{K_{t}} (\ddot{x}_{1d} + k_{1}\dot{e} + g)Z, \quad \dot{\hat{\theta}}_{2} = \frac{m}{K_{t}} e^{-cx_{1}} (\ddot{x}_{1d} + k_{1}\dot{e} + g)Z, \quad \dot{\hat{\theta}}_{3} = \frac{1}{K_{t}}Z,$$

$$\dot{\hat{\theta}}_{4} = \frac{e^{-cx_{1}}}{K_{t}}Z$$
(5-53)

Assuming the error between the actual virtual input and the desired virtual input is

$$e_2 = I - I_d \tag{5-54}$$

where the I is the actual virtual input,  $I_d$  is the desired virtual input.

Substituting Eq. (5-54) into Eq. (5-43) gives

$$p(x_1)\dot{Z} = \mathbf{W}\,\mathbf{\theta} - (e_2 + I_d) \tag{5-55}$$

Substituting Eq. (5-49) into Eq. (5-55) gives

$$p(x_1)\dot{Z} = \mathbf{W}\,\mathbf{\theta} - (e_2 + \mathbf{W}\,\hat{\mathbf{\theta}} + \frac{1}{2}\,\dot{p}(x_1)Z + KZ + I_{s2})$$
(5-56)

Rewriting Eq. (5-56) as:

$$p(x_1)\dot{Z} = -e_2 - \frac{1}{2}\dot{p}(x_1)Z - KZ + \mathbf{W}\,\tilde{\mathbf{\theta}} - I_{s_2}$$
(5-57)

For the deterministic robust control, the robust term  $I_{s2}$  should be any continuous function which satisfies the following conditions:

(1) 
$$Z(\mathbf{W} \boldsymbol{\theta} - I_{s_2}) \leq \varepsilon$$
  
(2)  $ZI_{s_2} \geq 0$  (5-58)

where  $\varepsilon$  is a positive design parameter that can be set arbitrarily small. Furthermore, the

Lyapunov function can be defined as following:

$$V_s = \frac{1}{2} p(x_1) Z^2$$
 (5-59)

The derivative of the  $V_s$  becomes

$$\dot{V}_{s} = \frac{1}{2}\dot{p}(x_{1})Z^{2} + p(x_{1})Z\dot{Z}$$
(5-60)

One example of  $I_{s2}$  satisfying the (5-58) is

$$I_{s2}(t) = Z \frac{(\mathbf{W}\boldsymbol{\theta}_m)^2}{\varepsilon}$$
(5-61)

where  $\theta_m = \theta_{max} - \theta_{min}$ ,  $\theta_{max}$  is the upper limit of the  $\theta$ , and  $\theta_{min}$  is the lower limit of the  $\theta$ .

Substituting Eq. (4-26) and Eq.(4-30) into Eq. (4-29) gives

$$\dot{V}_{s} = -Ze_{2} - KZ^{2} + Z(\mathbf{W}\,\tilde{\boldsymbol{\theta}} - Z\,\frac{(\mathbf{W}\boldsymbol{\theta}_{m})^{2}}{\varepsilon})$$
(5-62)

From the definition of the  $\theta_m$ , the derivative of the  $V_s$  satisfies

$$\dot{V}_{s} \leq -Ze_{2} - KZ^{2} + |Z| |\mathbf{W}\boldsymbol{\theta}_{m}| (1 - |Z| \frac{|\mathbf{W}\boldsymbol{\theta}_{m}|}{\varepsilon})$$
(5-63)

Case 1: if  $|Z|| \mathbf{W} \mathbf{\Theta}_m | > \varepsilon$  then

$$|Z|\frac{|\mathbf{W}\boldsymbol{\theta}_{m}|}{\varepsilon} > 1 \rightarrow (1 - |Z|\frac{|\mathbf{W}\boldsymbol{\theta}_{m}|}{\varepsilon}) < 0 \rightarrow |Z||\mathbf{W}\boldsymbol{\theta}_{m}|(1 - |Z|\frac{|\mathbf{W}\boldsymbol{\theta}_{m}|}{\varepsilon}) < 0$$
  
$$\Rightarrow \dot{V}_{s} \leq -Ze_{2} - KZ^{2}$$
(5-64)

Case 2: if  $|Z|| \mathbf{W} \mathbf{\Theta}_m | < \varepsilon$  then

$$|Z| \frac{|\mathbf{W}\mathbf{\theta}_{m}|}{\varepsilon} < 1 \rightarrow 0 \le (1 - |Z| \frac{|\mathbf{W}\mathbf{\theta}_{m}|}{\varepsilon}) < 1 \rightarrow |Z| |\mathbf{W}\mathbf{\theta}_{m}| (1 - |Z| \frac{|\mathbf{W}\mathbf{\theta}_{m}|}{\varepsilon}) \le |Z| |\mathbf{W}\mathbf{\theta}_{m}|$$

$$\Rightarrow \dot{V}_{s} \le -Ze_{2} - KZ^{2} + \varepsilon$$
(5-65)

In the first case, assuming  $e_2=0$  the inequality (5-64) then becomes

$$\dot{V}_{s} \leq -KZ^{2} \leq -2\frac{K}{p(x_{1})}V_{s}$$
(5-66)

The following inequality then can be derived by integrating (5-66)

$$V_{s}(t) \le \exp(\frac{-2Kt}{\|p(x_{1})\|_{\max}})V_{s}(0)$$
(5-67)

where t is time variant,  $\|p(x_1)\|_{\max}$  is the upper bound of  $p(x_1)$ . According to Bellman-Gronwall [88], the positive definite  $V_s$  is bounded by

$$V_s(t) \le \exp(\frac{-2Kt}{\|p(x_1)\|_{\max}})V_s(0)$$
 (5-68)

Based the relationship between Z and e, e is bounded by

$$e(t) \le \exp(-kt) \cdot e(0) \tag{5-69}$$

In the second case, assuming the  $e_2=0$  the inequality (5-65) then becomes

$$\dot{V_s} \le -KZ^2 + \varepsilon \le -2\frac{K}{p(x_1)}V_s + \varepsilon$$
(5-70)

The following inequality then can be derived by integrating (5-70). That is

$$V_{s}(t) \le \exp(\frac{-2Kt}{\|p(x_{1})\|_{\max}})V_{s}(0) + \int_{0}^{t} \exp\left(\frac{-2K(t-v)}{\|p(x_{1})\|_{\max}}\right) \varepsilon dv$$
(5-71)

where t is time variant,  $||p(x_1)||_{\max}$  is the upper bound of  $p(x_1)$ . According to Bellman-Gronwall, the positive definite  $V_s$  is bounded by

$$V_{s}(t) \le \exp(\frac{-2Kt}{\|p(x_{1})\|_{\max}})V_{s}(0) + \frac{\varepsilon \|p(x_{1})\|_{\max}}{2K} [1 - \exp(\frac{-2Kt}{\|p(x_{1})\|_{\max}})]$$
(5-72)

Based the relationship between Z and e, e is bounded by

$$e(t) \le \exp(-k_1 t) \cdot e(0) + \frac{Z}{k_1} [1 - \exp(-k_1 t)]$$
(5-73)

From the above inequalities, the tracking error of twisted-string system exponentially approaches to a ball with converging rate  $k_1$ . Within a finite duration, the asymptotic output tracking is achieved. If we know the error between the actual virtual input and desired virtual input goes to zero, we can stabilize the *e*. Therefore, the next objective of the controller design is to ensure the  $e_2$  approaches to zero. Taking the derivative of the  $e_2$  gives

$$\dot{e}_2 = \dot{I} - \dot{I}_d \tag{5-74}$$

Substituting the third equation in Eq. (5-36) into Eq. (5-74) gives
$$\dot{e}_2 = \frac{u - RI - K_b \omega}{L} - \dot{I}_d \tag{5-75}$$

Since  $I_d$  is designed as  $I_d = I_a + I_{s1+} I_{s2}$ , then

$$\dot{I}_{d} = \dot{I}_{a} + \dot{I}_{s1} + \dot{I}_{s2}$$

$$= \frac{d}{dt} (\mathbf{W} \,\hat{\mathbf{\theta}} + \frac{1}{2} \,\dot{p}(x_{1})Z) + K\dot{Z} + \dot{I}_{s2}$$

$$= \mathbf{W} \,\dot{\hat{\mathbf{\theta}}} + \frac{1}{2} \,\ddot{p}(x_{1})Z + \frac{1}{2} \,\dot{p}(x_{1})\dot{Z} + K\dot{Z} + \dot{I}_{s2}$$
(5-76)

Rewrite Eq. (5-75) as:

$$L\dot{\boldsymbol{e}}_2 = \boldsymbol{u} - \mathbf{W}_1 \boldsymbol{\theta}_1 \tag{5-77}$$

where  $\mathbf{W}_{1} = [I \ \omega \ i_{d}], \mathbf{\theta}_{1}^{\mathrm{T}} = [R \ k_{b} \ L].$ 

Using the same design procedure as in the above, the input u also can be given by:

$$u = u_a + u_s u_s = u_{s1} + u_{s2}$$
(5-78)

where  $u_a$  is adjustable feedforward controller used to compensate the model by parameter adaption,  $u_s$  is the robust controller in which  $u_{s1}$  is linear feedback to stabilize the system and  $u_{s2}$  is used to attenuate the parameter estimation error and nonlinear term of the model. In the Eq. (4-39) the vector  $\boldsymbol{\theta}_1$  are all unknown parameters that need to be estimated. Therefore, the vector  $\boldsymbol{\theta}_1$  will be replaced by the  $\hat{\boldsymbol{\theta}}_1$ , the estimated value of the vector  $\boldsymbol{\theta}_1$ , in the controller. Design u as  $V_2 = \frac{L}{2}e_2^2 + \frac{1}{2}\tilde{\boldsymbol{\theta}}_1^T\tilde{\boldsymbol{\theta}}_1$ , then

$$\dot{V}_{2} = Le_{2}\dot{e}_{2} - \tilde{\boldsymbol{\theta}}_{1}^{T}\hat{\boldsymbol{\theta}}_{1}$$

$$= e_{2}(u - \mathbf{W}_{1}\boldsymbol{\theta}_{1}) - \tilde{\boldsymbol{\theta}}_{1}^{T}\dot{\hat{\boldsymbol{\theta}}}_{1}$$
(5-79)

Since the  $u_a$  is adjustable feedforward controller used to compensate the model by parameter adaption, the  $u_a$  is:

$$u_a = \mathbf{W}_1 \hat{\mathbf{\theta}}_1 \tag{5-80}$$

Letting  $u_{s1} = -K_1 e_2$ , we get

$$\dot{V}_2 = e_2(-K_1e_2 + \mathbf{W}_1\hat{\boldsymbol{\theta}}_1 - \mathbf{W}_1\boldsymbol{\theta}_1) \cdot \tilde{\boldsymbol{\theta}}_1^{\mathbf{T}}\dot{\hat{\boldsymbol{\theta}}}_1$$
(5-81)

where  $K_1$  is a positive feedback gain. In order to ensure the stability of the system, the derivative of  $V_2$  should be less or equal to zero by letting

$$-e_{2}\mathbf{W}_{1}\tilde{\boldsymbol{\theta}}-\tilde{\boldsymbol{\theta}}_{1}^{\mathbf{T}}\hat{\boldsymbol{\theta}}_{1}=0$$
(5-82)

Solving the Eq. (5-82) gives:

$$\hat{\boldsymbol{\theta}}_1 = -\mathbf{W}_1^{\mathrm{T}} \boldsymbol{e}_2 \tag{5-83}$$

Therefore the adaptation mechanisms for adjusting parameters are:

$$\dot{\hat{\theta}}_{11} = -Ie_2, \ \dot{\hat{\theta}}_{12} = -\omega e_2, \ \dot{\hat{\theta}}_{13} = -\dot{I}_d Z$$
 (5-84)

Substituting Eq. (5-80) into Eq. (5-77) and adding the  $u_{s2}$  gives

$$L\dot{e}_2 = -K_1 e_2 + \mathbf{W}_1 \tilde{\mathbf{\theta}}_1 + u_{s2} \tag{5-85}$$

For the deterministic robust control, the robust term  $u_{s2}$  should be any continuous function which satisfies the following conditions:

(1) 
$$e_2(\mathbf{W}_1 \tilde{\mathbf{\theta}}_1 + u_{s2}) \le \varepsilon_1$$
  
(2)  $e_2 u_{s2} \ge 0$  (5-86)

where  $\varepsilon_1$  is a positive design parameter that can be set arbitrarily small. Furthermore, the Lyapunov function can be defined as following:

$$V_{s1} = \frac{L}{2}e_2^2 \tag{5-87}$$

The derivative of the  $V_{s1}$  becomes

$$\dot{V}_{s1} = Le_2 \dot{e}_2 \tag{5-88}$$

Substituting Eq. (5-85) into Eq. (5-88) gives

$$\dot{V}_{s1} = -K_1 e_2^2 + e_2 (\mathbf{W}_1 \tilde{\boldsymbol{\theta}}_1 + u_{s2})$$
(5-89)

One example of  $u_{s2}$  satisfying the (5-86) is

$$u_{s2}(t) = -e_2 \frac{(\mathbf{W}_1 \boldsymbol{\theta}_{1m})^2}{\varepsilon_1}$$
(5-90)

where  $\theta_{1m}=\theta_{1max}-\theta_{1min}$ ,  $\theta_{1max}$  is the upper limit of the  $\theta_1$ , and  $\theta_{1min}$  is the lower limit of the  $\theta_1$ . Substituting Eq. (5-90) into Eq. (5-89) gives

$$\dot{V}_{s1} = -K_1 e_2^2 + e_2 (\mathbf{W}_1 \tilde{\mathbf{\theta}}_1 - e_2 \frac{(\mathbf{W}_1 \mathbf{\theta}_{1m})^2}{\varepsilon_1})$$
(5-91)

From the definition of the  $\theta_{1m}$ , the derivative of the  $V_{s1}$  satisfies

$$\dot{V}_{s1} \le -K_1 e_2^2 + |e_2| |\mathbf{W}_1 \mathbf{\theta}_{1m}| (1 - |e_2| \frac{|\mathbf{W}_1 \mathbf{\theta}_{1m}|}{\varepsilon_1})$$
(5-92)

Case 1: if  $|e_2|| \mathbf{W}_1 \mathbf{\Theta}_{1m} | > \varepsilon_1$  then

$$|e_{2}|\frac{|\mathbf{W}_{1}\mathbf{\theta}_{1m}|}{\varepsilon_{1}} > 1 \rightarrow (1 - |e_{2}|\frac{|\mathbf{W}_{1}\mathbf{\theta}_{1m}|}{\varepsilon_{1}}) < 0 \rightarrow |e_{2}||\mathbf{W}\mathbf{\theta}_{m}|(1 - |e_{2}|\frac{|\mathbf{W}_{1}\mathbf{\theta}_{1m}|}{\varepsilon_{1}}) < 0$$
  
$$\Rightarrow \dot{V}_{s1} \le -K_{1}e_{2}^{2}$$
(5-93)

Case 2: if  $|e_2||\mathbf{W}_1\mathbf{\Theta}_{1m}| < \varepsilon_1$  then

$$|e_{2}|\frac{|\mathbf{W}_{1}\mathbf{\theta}_{1m}|}{\varepsilon_{1}} < 1 \rightarrow 0 \leq (1 - |e_{2}|\frac{|\mathbf{W}_{1}\mathbf{\theta}_{1m}|}{\varepsilon_{1}}) < 1 \rightarrow |e_{2}||\mathbf{W}_{1}\mathbf{\theta}_{1m}| (1 - |e_{2}|\frac{|\mathbf{W}_{1}\mathbf{\theta}_{1m}|}{\varepsilon_{1}}) \leq |e_{2}||\mathbf{W}_{1}\mathbf{\theta}_{1m}|$$

$$\Rightarrow \dot{V}_{s} \leq -K_{1}e_{2} + \varepsilon_{1}$$
(5-94)

In the first case, the derivative of the  $V_{sI}$  satisfies

$$\dot{V}_{s1} \le -K_1 e_2^2 \le -2\frac{K_1}{L} V_{s1}$$
(5-95)

The following inequality then can be derived by integrating (5-95). That is

$$V_{s1}(t) \le \exp(\frac{-2K_1 t}{\|L\|_{\max}}) V_{s1}(0)$$
(5-96)

where *t* is time variant,  $||L||_{max}$  is the upper bound of *L*. According to Bellman-Gronwall, the positive definite  $V_{s1}$  is bounded by

$$V_{s1}(t) \le \exp(\frac{-2K_1 t}{\|L\|_{\max}}) V_s(0)$$
(5-97)

Therefore,  $e_2$  is bounded by

$$e_2(t) \le \exp(-K_1 t) \cdot e_2(0)$$
 (5-98)

In the second case, the derivative of the  $V_{s1}$  satisfies

$$\dot{V}_{s1} \le -K_1 e_2^2 + \varepsilon_1 \le -2\frac{K_1}{L} V_{s1} + \varepsilon_1$$
(5-99)

The following inequality then can be derived by integrating (5-99). That is

$$V_{s1}(t) \le \exp(\frac{-2K_{1}t}{\|L\|_{\max}})V_{s1}(0) + \int_{0}^{t} \exp(\frac{-2K_{1}(t-v)}{\|L\|_{\max}})\mathcal{E}_{1}dv$$
(5-100)

where *t* is time variant,  $\|L\|_{\max}$  is the upper bound of *L*. According to Bellman-Gronwall, the positive definite  $V_{s1}$  is bounded by

$$V_{s1}(t) \le \exp(\frac{-2K_1t}{\|L\|_{\max}})V_s(0) + \frac{\varepsilon \|L\|_{\max}}{2K_1} [1 - \exp(\frac{-2K_1t}{\|L\|_{\max}})]$$
(5-101)

Therefore  $e_2$  is bounded by

$$e_2(t) \le \exp(-K_1 t) \cdot e_2(0) + \frac{1}{K_1} [1 - \exp(-K_1 t)]$$
 (5-102)

From the above inequalities, the tracking error of twisted-string system exponentially approaches to a ball with converging rate  $K_1$ . Within a finite duration, the asymptotic output tracking is achieved. Furthermore, the control input *u* can be derived as following:

$$u = -K_1 e_2 + W_1 \hat{\theta}_1 + u_{s2} \tag{5-103}$$

#### **5.3.2 Simulation Results**

In this section, we present simulation studies to illustrate the performance of the adaptive robust controller based on the force. The following parameters are used: the mass *m* of the load is 0.5 kg, the stiffness of the spring generating the external force is 4000 N/m, the armature resistant of *R* of the DC motor is 9.7 ohm, the inductance of the DC motor is 0.0003H, the torque constant  $K_t$  of DC motor is 0.1186, and the Back EMF Constant  $K_b$  of DC motor is 0.0611. The desired trajectory is shown in Figure. 5-9. Figure. 5-10 shows the control results. It can be seen that the maximum tracking error is 0.12mm. As the controlling process continues the error between the desired and actual current is further reduced to about 0.004 after t = 2 s. The reason for this is that the adaptation mechanism can make the control

input reducing the error between the desired and actual current by estimating the unknown parameters.



Figure 5-10 Simulation results of ARC (force) controller.

# **5.3.3 Experimental Results**

In this study, we use the experimental platform, shown in Figure 5-11, to test the tracking performance of the adaptive robust controller based on the force. The adaptive robust controller and data acquisition system are all implemented using Simulink and xPC target

toolbox. The sampling rate is chosen as 1 kHz. The controller parameters are designed by selecting  $k_1$ =10, K=3,  $K_1$ =7. According to the system properties, the mass m of the load is 0.5 kg, the stiffness of the spring generating the external force is 4000 N/m, and the bounded unknown parameters are specified in Table 5-2.



Figure 5-11 Configuration of experimental setup.

Table 5-2 Comparison of performance of the three controllers of sinusoidal	motion
--	--------

Parameter	Lower bound	Higher bound
$ heta_1$	0.001	0.0015
$ heta_2$	0.004465	0.006465
$ heta_3$	0	3.788
$ heta_4$	0	10.93
R	8	12
$k_b$	0	0.5
L	0	0.0003

To compare the performance, two types of control structures, ARC controller with displacement feedback and ARC controller with force feedback, are used to compensate for both tracking and synchronization with a sinusoidal trajectory. For such a sinusoidal motion,

the amplitude of the periodic displacement is 60 mm and the duration of each cycle is 12 seconds. The experimental results of the ARC controller with displacement feedback and the ARC controller with force feedback proposed in this paper are demonstrated in Figure. 5-12. Fig 5-13 illustrates the comparison between actual current and desired current. From both Figures 5-12 and Fig 5-13, the maximal tracking error of the proposed controller is ~ 2 mm. It is clear that either tracking errors of desired trajectory or the difference between the desired virtual input and actual virtual input (armature current of the DC motor) asymptotically approaches to zero, which proves the feasibility of the proposed controller.



Figure 5-13 Comparison between actual current and desired current.

In this Chapter, an ARC control scheme with force feedback has been developed for a high accuracy twisted-string actuation system. Compared with the output controlled by ARC

controller with displacement feedback, the proposed approach demonstrates excellent results for tracking performance.

# **CHAPTER 6:** Derivation of motion trajectories

Conventional stroke rehabilitation activities generally include intensive and repetitive range-of-motion exercises and require professional therapists to help patients complete the exercises sometimes and to assess patients' progress. Therefore, for the purpose of self-rehabilitation, trajectories used in intensive and repetitive training activities play an important role in the rehabilitation process. It is, therefore, important to acquire patients' arm motions and verify their outcome automatically and efficiently without the attendance of medical staff. To make such home-based rehabilitation activities possible, new technical developments of mobile technologies and video game industries have recently gained more interest. With these technologies, it is possible to monitor, or supervise patients to perform their programmed activities independently [89]. This is also one of the most important tasks of home rehabilitation. To provide effective rehabilitation, it is required to adopt movements of non-disabled people as templates of training trajectories. This chapter discusses how patients' motion can be acquired.

# 6.1 Development of motion derivation system

#### **6.1.1 Introduction of Kinect**

In this study, motion trajectories were acquired by a Microsoft Kinect sensor. The configuration of the Kinect is as shown in Figure. 6-1. The Skeletal Viewer application included in the Developer Toolkit (1.8.0) was modified to capture motion data from Kinect and save the raw data to digital storage [90]. The computer used for this motion capturing systems is a 64-bit Windows 8 system with a 3.50 GHz Intel core i5 CPU.

A Kinect is composed of an RGB camera that captures a color image, an infrared (IR) emitter and an IR depth sensor. The IR emitter emits infrared light beams to objects. The IR depth sensor can then read the infrared signal back from the objects. Once received by the IR receiver, the reflected infrared signals are converted into the information of distance between object surfaces and the sensor. The tilt motor in the Kinect is used to automatically adjust the angle between Kinect and objects. A Kinect can simultaneously track two objects without any mark. It can also provide three-dimensional location information of two objects up to 20 joints. Compared with other tracking methods such as optical tracking which need additional marking on the object, a Kinect can greatly facilitate tracking objects. As to the sample rate a Kinect provides 30 frames of images per second, in which tracked skeleton, as shown in Figure 6-2, can provides detailed position information of the subject's joints. All position information of the subject's 20 joints is described in a three-dimensional space, where the Kinect is the center of the 3-D coordinate system, as shown in the Figure 6-3.



Figure 6-1 The configuration of Microsoft Kinect sensor [91].



Figure 6-2 The 20 joints of the skeleton position acquired by a Kinect [92].



Figure 6-3 The 3-D coordinated system of the Kinect centered space.

#### 6.1.2 Angle of shoulder joint

The exoskeleton assistive robot in this study has two independent degrees of freedom used for elbow and shoulder joints. In order to adopt movements of non-disabled people as template trajectories for training or rehabilitation purposes, the angular positions of both shoulder and elbow joints need to be acquired in real time. The acquired positions (in 3-D coordinates) of the shoulder, elbow, and wrist are used to calculate the angular positions of individual joints.

In this study, the shoulder and elbow joints are assumed to be aligned in the same plane when they rotate simultaneously based on the limit of two degrees of freedom design. Therefore, the equation of line connecting shoulder and elbow joints can be derived without Z-axis coordinates. That is

$$y_c = k_{c1} x_c + \zeta_1, (6-1)$$

where

$$k_{c1} = \tan \varphi_1 = \frac{|y_{c2} - y_{c1}|}{x_{c2} - x_{c1}} (x_{c2} \neq x_{c1}).$$
(6-2)

The coordinates  $(x_{c1}, y_{c1})$  and  $(x_{c2}, y_{c2})$  are the positions of shoulder and elbow joints, respectively. The value of  $\varsigma_1$  can be ignored because  $\varsigma_1$  does not affect the calculation of the angle  $\varphi_1$  between the shoulder joint and vertical plane. Thus, the angular position can be obtained by

$$\varphi_{1} = \arctan(\frac{|y_{c2} - y_{c1}|}{x_{c2} - x_{c1}}), x_{c2} \neq x_{c1}$$

$$\varphi_{1} = 0, \quad x_{c1} = x_{c2}$$
(6-3)

#### 6.1.3 Angle of elbow joint

Comparing to the calculating process of shoulder angle, calculation of the elbow angle needs two equations of line, the line connecting the shoulder and elbow as well as the line connecting the elbow and wrist, as shown in the Figure 6-4. The equation of line connecting the elbow and wrist can be derived by adopting the same method for the shoulder.

$$y_c = k_{c2} x_c + \zeta_2 \tag{6-4}$$

where

$$k_{c2} = \tan \varphi_3 = \frac{|y_{c3} - y_{c2}|}{x_{c3} - x_{c2}} (x_{c2} \neq x_{c3})$$
(6-5)

where  $(x_{c3}, y_{c3})$  is the coordinates of wrist joints. The value of  $\varsigma_2$  can be ignored because  $\varsigma_2$  does not affect the calculation of the angle  $\varphi_3$  between the wrist joint and horizontal plane.

$$\varphi_{3} = \arctan(\frac{y_{c3} - y_{c2}}{x_{c3} - x_{c2}}), x_{c2} \neq x_{c3}$$

$$\varphi_{3} = 0, \quad x_{c2} = x_{c3}$$
(6-6)

According to the geometric relationship in the Figure 6-4, the rotation angle  $\varphi_2$  of the elbow is

$$\varphi_{2} = 90^{\circ} - \varphi_{1} + \arctan(\frac{y_{c3} - y_{c2}}{x_{c3} - x_{c2}}), x_{c2} \neq x_{c3}$$

$$\varphi_{2} = 90^{\circ} - \varphi_{1}, \quad x_{c3} = x_{c2}$$
(6-7)

# 6.2 Accuracy of the Kinect

In addition to the non-contact measurement using Kinect, an incremental optical encoder (Avago HEDS-5500-A06) with 500 lines was attached to the shaft of the elbow joint of the prototype robotic mechanism as shown in the Figure 6-5. With a single channel

measurement, the resolution of the encoder is 0.72 Degree. The resolution can quadrupled if signals measured from both channels are used.



Figure 6-4 Schematic of Shoulder and elbow angle.



# Figure 6-5 Prototype mechanism of the robotic system with encoder used for motion measurement.

In this section, only the measurements of angular position at elbow were used for comparison due to the limitation of the current mechanical design. With the software of Kinect library, the 3D coordinates of the shoulder joint, elbow joint, and wrist joint can be obtained directly. A program that can derive the angle of the shoulder joint and the angle between upper arm and forearm from raw sensing data was developed.

Figure 6-6 illustrates three cycles of the same movements acquired by these two acquiring methods. The specific motion adopted is that a subject performs a motor task of raising his/her hand to mouth and returning to the initial position three times. In Figure 6-6, the red curve is the data recorded by the optical encoder and the blue dot line is measured by the adopted Microsoft Kinect. It is assumed that an optical encoder can have a more accurate measurement since it directly measure the angular position of the subject's elbow joint. With the experimental results of five subjects, the measured data of Kinect demonstrates great consistency with the measurement of the attached optical encoder, which means that acquired locomotion patterns between the optical encoder and Kinect are similar.



Figure 6-6 Trajectories of the right elbow when performing motor task Hand to Mouth.

# **6.3 Desired Moving Trajectories**

Task-specific training for stroke rehabilitation in general is unconstrained with regard to accurate position and specific details of task execution. Therefore, the most important factors when performing task-specific training are the pace and pattern. Postural control or developing movement patterns are the initial steps for the stroke patients before they have the ability to accomplish some simple ADL through task-specific training. Today's therapists have been trying to develop optimal movement patterns as the reference trajectories of rehabilitation robots, which falls outside the scope of this research. The goal of this research is discovering movement patterns as reference trajectory of the prototyped robot from participants' trajectory data of upper extremity.

To identify trajectories of human movements, Microsoft Kinect was adopted to acquire arm motion data in this research due to its convenience and the similarity of measured outcomes of these two sensors. In this study, a total of five non-disabled participants were recruited from the population of faculty members and students at West Virginia University. The age range of the subjects is between 25 to 45 years old. (four males and one female). Subjects were requested to perform five basic daily movements or tasks in front of the Kinect, which were derived from the Wolf Motion Function Test [93] commonly used for evaluation of motor deficit level in stroke patients: (1) forearm to table; (2) forearm to box on table; (3) extend elbow on table; (4) hand to mouth; and (5) hand on top of head. While performing the experiment, each participant was standing in front of the Kinect sensor at a distance of 1.8m.

Figure 6-7 shows the recorded angular trajectories of elbow joints captured from these five subjects. Each subject needs to perform an identical movement for 3 cycles. It is obvious the trajectories of individual subjects have different durations and magnitudes. The required duration varies from 12 to 28 seconds for three cycles. The variation of actual trajectories of joints among people is caused by physical parameters, such as height and arm span. In particular, the duration of a complete cycle of the specified motion cannot be the same for

different people while even performing the same task.



Figure 6-7 Comparison of the trajectories of the right elbow between five subjects performing separately motor task Hand to Mouth three times.

#### 6.3.1 Normalization of trajectories

When people move their arms to finish specific tasks they do not move randomly. Instead, people always follow specific trajectories or identical motion patterns [94]. Thus, it is reasonable to extract a common trajectory or motion pattern from trajectories of temporal variations. A measure called Dynamic Time Warping (DTW) [95, 96] can be employed to normalize the trajectories in terms of time in this study. DTW, which is a dynamic programming based technique, is a method to calculate the similarity and alignment between two data sequences with different time durations. This method adopts the algorithm of the Euclidean match, which aligns two sequences to the same time scale with a one-on-one match, as shown in the Figure 6-8. In addition to align elements of two sequences to the same scale, this method can also match multiple elements in one sequence to the same element in the other sequence. This technique has been widely adopted in the field of voice recognition. In particular, DTW can be applied to cope with different speaking speeds that can be recognized or classified into a linguistic category of words [97].



Figure 6-8 Euclidean match and Dynamic time warping match.

Euclidean distance, as well as other *L* norms, are usually used to compare the similarity of two data sets. For instance, if set  $A_e$  (Subject 1) has *n* components and set  $B_e$  (Subject 2) has *m* components, the distance between two elements  $A_e(i)$  and  $B_e(j)$  can be defined as

$$d_e(i,j) = \left(A_e(i) - B_e(j)\right)^2 \tag{6-8}$$

and the Euclidean distance can be defined as

$$D_E = \sum d_e(i,j) . \tag{6-9}$$

However, the results are not reasonable or accurate when the Euclidean distance is applied to measure the similarity of two data sets with different time durations. In this study, individual recorded trajectories are analyzed to identify start points. In Figure 6-9, the original trajectory (blue line) along time scale is shifted for 100 samples (red line) to obtain a new trajectory. The major reason of such a shift is to compensate for the inconsistent durations of individual subjects. For instance, the durations of the same movement performed by different subjects can have different time duration due to various factor (age, gender, height, etc) as shown in Figure 6-10.

To synchronize the acquired data of two trajectories with different time duration, the two sets of data are placed along x- and y- directions of a two-dimensional map. Once the two sets of data are aligned correctly, an  $n \times m$  matrix is constructed to find the best alignment. The best alignment is called the warping path  $P_{dtw}$ . In other words, the warping path  $P_{dtw}$  is a contiguous set of matrix elements that defines a mapping between  $A_e$  and  $B_e$ . Another distance needs to be defined is called the DTW *distance*,  $(D_{dtw}(A,B))$ , which is a time-normalized distance between  $A_e$  and  $B_e$ . The DTW distance can be derived by using dynamic programing with the following procedures:

- 1. The initial condition is defined as  $D_{dtw}(A_e(1), B_e(1)) = d_e(1,1)$ .
- 2. The individual components of the DTW distance can be found by

$$D_{dtw}(A_{e}(i), B_{e}(j)) = \min \begin{cases} D_{dtw}(A_{e}(i), B_{e}(j-1))^{2} \\ D_{dtw}(A_{e}(i-1), B_{e}(j-1))^{2} \\ D_{dtw}(A_{e}(i-1), B_{e}(j))^{2} \end{cases} + d_{e}(i, j), i \in [2, n],$$

$$j \in [2, m].$$

3.  $D_{dtw}(A_e, B_e) = D_{dtw}(A_e(n), B_e(m)).$ 



Figure 6-9 Euclidean distance between Original and shifted trajectories.



Figure 6-10 Comparison of the trajectories of the right elbow between two subjects performing separately motor task Hand to Mouth three times.

			/	d <sub>e</sub> (i,	<i>j</i> )	
7	25	16	25	36	16	9
3	1	0	1	4	0	1
4	4	1	4	9	1	0
5	9	4	9	16	4	1
2	0	1	0	1	1	4
1	1	4	1	0	4	9
	2	3	2	1	3	4
					$D_{dhe}(A_{e}$	$(i), B_{\epsilon}(j))$
7	40	22	31	43 🖌	D <sub>dbs</sub> (A <sub>e</sub>	$(i), B_{\varepsilon}(j))$
7 3	40	22 6	31 7	43 🖌 11	D <sub>dhe</sub> (A <sub>e</sub>	$(i), B_{\varepsilon}(j))$
7 3 4	40 15 14	22 6 6	31 7 9	43 🖌 11 18	D <sub>dhe</sub> (A <sub>e</sub> 24 <del>√</del> 8 ⊭ 8	$(i), B_{\epsilon}(j))$
7 3 4 5	40 15 14 10	22 6 6 5	31 7 9 11	43 × 11 18 18	D <sub>dbe</sub> (A <sub>e</sub> 24 8 ⊭ 8	$(i), B_e(j))$
7 3 4 5 2	40 15 14 10 1	22 6 5 2	31 7 9 11 2	43 ⊭ 11 18 18 3	D <sub>dbs</sub> (A <sub>s</sub> 24 8 ⊮ 8 7 4	$(i), B_{e}(j))$ $15$ $6$ $5$ $5$ $8$
7 3 4 5 2 1	40 15 14 10 1 1	22 6 5 2 5	31 7 9 11 2 6	43 ✔ 11 18 18 3 6	D <sub>dh</sub> (A <sub>e</sub> ) 24 8 ⊭ 8 7 4 10	$(i), B_e(j))$ 15 6 5 5 8 19



For example, if have two time sequence, one is (1, 2, 5, 4, 3, 7), the other one is (2, 3, 2, 1, 3, 4), need to be aligned, a  $6 \times 6$  matrix will be constructed, in which each element is equal to  $d_e(i,j)$ . Once the matrix is constructed, the above algorithm is applied to find the DTW distance of each element in the matrix.

According to the DTW distance matrix, as shown in Figure 6-11, we start from the element located at the right top corner of the matrix and end at the element located at the left bottom corner of the matrix. In this process, we just move one step at a time along one direction in which the next element has the minimum DTW distance component in three directions. In the Figure 6-12, the Dynamic time warping path can be found by linking all

7	40	22	31	43	24	15
3	15	6	7	11	8	6
4	14	6	9	18	8	5
5	10	5	11	18	7	5
2	1	2	2	3	4	8
1	1	5	6	6	10	19
	2	3	2	1	3	4

selected elements one by one.

Dynamic time warping path

#### Figure 6-12 Dynamic time warping path.

Typically, the path with the lowest DTW distances implies the best match between two trajectories. Figure 6-13 illustrates the results of time-normalized trajectories from two subjects that perform the same motion. From this figure, it is clear that different subjects can demonstrate good consistencies in terms of time for the same movement of elbow joint trajectories after being time-normalized.



Figure 6-13 Time-normalized elbow-joint trajectories.

# 6.3.2 Extracting trajectory

To ensure the consistency of motions performed by the individual subject, a

self-comparison test was performed. Root-mean-square-error (RMSE) is used to assess the similarity among acquired trajectories. Lower RMSE implies a better similarity between two acquired data sets. In this test, each subject was requested to perform the same motion for 8 times, as shown in Figure 6-14. The recorded data of joint angles was then processed by DTW for an identical duration. Then the 8 sets data are compared among each other by calculating the RMSE of the deviation of each data. The 8 sets of data were averaged to synthesize an estimated trajectory. Table 6-1 and 6-2 lists the comparison of elbow angles among real trajectories and the estimated motion of two subjects. The heights of these two subjects are 168 cm and 177 cm, individually. With the calculated RMSE values, it is clear that individual subjects can generate consistent trajectories for specific motions. However, RMSEs are increased while comparing the data with other subjects. Table 6-3 lists the RMSEs among various subjects for the same motion. Comparing row 4 with other rows in Table 6-3, it is obvious that subject 4 has relatively larger differences of elbow joint movement. This is due to the fact that subject 4 is the only female participant in the experiments, whose body size is relatively smaller than other the male participants. With this experimental result, it can be concluded that people with similar body sizes can possibly have more similar movement. In other words, a movement can be highly dependent on the body size. Therefore, different reference trajectories of the same motion need to be extracted separately for people with different body conditions, such as gender, age, and body size.



Figure 6-14 Photo frames and joint positions of motion captured by Microsoft Kinect.

	1	2	3	4	5	6	7	8	Avg
1	0	3.63	3.83	3.78	4.52	2.52	1.62	3.7	3.16
2	3.63	0	1.18	0.37	1.56	1.34	2.47	0.96	0.61
3	3.83	1.18	0	0.84	1.37	2.37	3.26	0.47	0.78
4	3.78	0.37	0.84	0	1.64	1.82	2.89	0.97	0.74
5	4.52	1.56	1.37	1.64	0	2.44	3.2	0.92	1.37
6	2.52	1.34	2.37	1.82	2.44	0	1.09	1.69	0.69
7	1.62	2.47	3.26	2.89	3.2	1.09	0	2.57	1.42
8	3.7	0.96	0.47	0.97	0.92	1.69	2.57	0	0.54
Avg	3.16	0.61	0.78	0.74	1.37	0.69	1.42	0.54	0

 Table 6-1 Self-comparison of RMSE of recorded motion (Subject 1).

 Table 6-2 Self-comparison of RMSE of recorded motion (Subject 2).

	1	2	3	4	5	6	7	8	Avg
1	0	2.04	0.70	0.77	1.87	2.10	2.11	0.66	0.94
2	2.04	0	1.61	1.25	0.74	1.52	1.48	1.42	0.84
3	0.70	1.61	0	0.46	1.76	1.53	1.50	0.44	0.26
4	0.77	1.25	0.46	0	1.19	2.66	2.48	0.27	0.44
5	1.87	0.74	1.76	1.19	0	1.52	1.66	1.24	1.02
6	2.10	1.52	1.53	2.66	1.52	0	0.36	1.66	1.03
7	2.11	1.48	1.50	2.48	1.66	0.36	0	1.68	0.88
8	0.66	1.42	0.44	0.27	1.24	1.66	1.68	0	0.41
Avg	0.94	0.84	0.26	0.44	1.02	1.03	0.88	0.41	0

Subject Subject	1	2	3	4	5
1	0	7.03	8.12	11.00	6.13
2	7.03	0	7.78	10.76	6.63
3	8.12	7.78	0	10.04	5.74
4	11.00	10.76	10.04	0	10.90
5	6.13	6.63	5.74	10.90	0

Table 6-3 RMSEs among the various subjects' trajectory after DTW.

To further observe the impact of different body conditions, the subjects are divided into two groups with different heights. The range of Groups 1 is between 163 to 168 cm, and the range of Group 2 is between 177 to 182 cm. The RMSEs among the subjects in the same groups, either Group 1 or Group 2, is increased to ~5 degrees. This value can be as high as 11 degrees while comparing subjects in different groups.

The result discussed above suggests that DTW can be a valid tool to extract common moving profiles of healthy people for specific ADLs. The trajectories extracted from selected ADLs can then be used by the assistive robotic systems. Figure 6-15 demonstrates the process of obtaining the averaged trajectory from 5 subjects' trajectories. Figure 6-16 and 6-17 illustrate the averaged trajectory, original trajectory before normalization as well as the normalization trajectory. These averaged profiles can also be used as standards to evaluate the effectiveness of rehabilitation activities. By comparing the standard profiles with the motion trajectories, the similarity scores can be assessed. Since only 5 subjects were recruited in this study, the size of the database is considered small. Larger groups of subjects of different body conditions need to be recruited for the extraction of motion trajectories.



Figure 6-15 Flow chart of the procedure of extracting averaged trajectory.



Figure 6-16 Averaged trajectory of elbow joint.



Figure 6-17 Averaged trajectory of shoulder joint.

# **CHAPTER 7:** Implementation of the robot

# 7.1 Prototype of exoskeleton assistive device overview

The goal of the design exoskeleton assistive device is to verify feasibility of the project. To achieve this goal, all the designs of individual parts remain simple. A microprocessor, a FPGA, DC motors, encoders, and potentiometer are used to meet the requirements to drive a 2-joint exoskeleton robotic arm via four twisted-string actuators.

#### 7.1.1 Structure of the prototype

The prototype of a 2-joint exoskeleton robotic device has been setup on a 16 in × 32 in wood stand. The mechanism of this 2-DOF robotic arm is installed at the left-top corner of the stand. Two Shimano Bowden cables [98], as shown in Figure 7-1, are used for flexion and extension at each joint of the robotic arm. These four Bowden cables are responsible for driving shoulder and elbow joints to rotate by transferring force generated by the twisted-string actuators. To avoid self-rotation of string and ensure linear motion of the twisted-string actuators, four independent sliders are installed on the top of the stand to guide the movements of the strings. The configuration of string holders is shown in Figure 3-4, which are mounted on the sliders and the shaft of DC motors. To measure the linear movements of the twisted-string actuators, four analog slide potentiometers, as shown in Figure 7-2, are fixed on the back of individual sliders and locked on the corresponding holders. Four DC motors are also set up at the bottom of the stand to drive individual actuators. The lengths of twisted-strings can also be adjusted by changing the distances between DC motors and sliders with the vertical slots below the sliders. The experimental setup of the exoskeleton assistive robot is shown in Figure 7-3.



Figure 7-1 Bowden cable on the joints.



Figure 7-2 Slide potentiometer.



Figure 7-3 Prototype of the exoskeleton robotic arm.

# 7.2 Sensor measurement fusion

In the experimental setup demonstrated in Chapter 3, 4 and 5, an optical linear encoder (WXY-XY20) is used to measure the length variation of the twisted-string actuator. However, the linear encoder is not only expensive but also bulky, which can be a major drawback for an affordable wearable device. In order to reduce the cost and simplify the structure of the robotic mechanism, a slide potentiometer and a rotary encoder mounted on the shaft of driving DC motor are adopted to replace the optical linear encoder. In this section, we will illustrate the method of combing the measurement of two sensors.

# 7.2.1 Kalman filter

Unlike the linear optical encoder, the slide potentiometer is an analog sensor. Therefore, the existence of noise in the measurement is inevitable. In some monitoring system, such as

room temperature monitoring system, the noise caused by sensors may not affect the performance of the system. However, if the noise is introduced to the control system as a feedback signal, not only the precision but also the stability of the controlled system cannot be guaranteed. Therefore, it is necessary to filter the noise from the measured data before it is fed into the control loop.

With the advantages of simplicity and better performance, Kalman filter has been widely used in various control systems since its introduction in the early 1960s. The filter was first proposed by Rudolf E. Kálmán [99], one of the primary developers of the popular theory. Kalman filter, also known as linear quadratic estimation, is an optimal recursive data processing algorithm that takes all obtained information into consideration, such as measurements over time and statistical noises [100]. In this study, four Kalman filters are in charge of processing all information to estimate the current output values of slide potentiometers.

In order to ensure that the Kalman filter can successfully be implemented to each slide potentiometer, a linear model is necessary as well. If the variable of slider position is denoted as *s* and  $\Delta t$  represents the sampling period, then position equation can be estimated using Euler's method. That is

$$s(t) = s(t - \Delta t) + \dot{s}(t - \Delta t)\Delta t \tag{7-1}$$

If the continuous function is represented as a discrete system, an index k is used to replace the time stamp t. Thus, Eq. (7-1) can be rewritten as:

$$s(k) = s(k-1) + \dot{s}(k-1)$$
(7-2)

If the state variable can be defined as:

$$x_{k}(k) = \begin{bmatrix} s(k) \\ \dot{s}(k) \end{bmatrix}$$
(7-3)

then the linear process model can be defined as:

$$x_k(k) = \mathbf{A}_k \cdot x_k(k-1) \tag{7-4}$$

where

$$\mathbf{A}_{k} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}. \tag{7-5}$$

This linear process model assumes that the derivative does not change over time. Although this is not true in the practical environment, the motion of the slider can still be approximated by this proposed linear model. The Kalman filter estimates the state in a process that can be described as the linear stochastic difference equation:

$$x_{k}(k) = \mathbf{A}_{k} \cdot x_{k}(k-1) + \mathbf{B}_{k} \cdot u_{k-1} + w_{k-1}.$$
(7-6)

with one single measurement  $z_k(k)$  that is

$$z_k(k) = \mathbf{H}_k \cdot x_k(k) + v(k) \tag{7-7}$$

where  $\mathbf{H}_{k}=[1 \ 0]$ , v(k) and w(k) are the measurement noise and process noise respectively. The schematics of the discrete Kalman filter algorithm, which is used to estimate the state variables, is illustrated in the following figure.



Figure 7-4 Schematic of Kalman filter procedure.

In this study, the movement of the slider is part of the process noise since there is no control effort included in the linear stochastic difference equation above. Thus, the  $\mathbf{B}_k$  and  $u_k$  are always zero in Figure 7-4.  $\mathbf{Q}_k$  describes the process covariance of linear model as well as  $\mathbf{P}$  describes the estimate error covariance.  $\mathbf{Q}_k$  and  $\mathbf{P}_k$  are the 2 × 2 dimension matrices. Since the value of estimate error covariance  $\mathbf{P}_k$  is updated recursively by deriving the equations in Kalman filter, its initial value defined in the first step will not affect the performance of the Kalman filter. We assume that the measurement noise of slide potentiometer and the speed of slider movement are independent. Thus, the non-diagonal matrix items in  $\mathbf{Q}_k$  and  $\mathbf{P}_k$  matrices are equal to zero, the diagonal matrix items are the variances of process noises and estimate errors. In the implementation of the Kalman filter,  $\mathbf{R}_k$  is measurement noises variances that can be measured by processing some off-line sample measurements in which the output of the slide potentiometer is constant. Then the variance of the slide potentiometer can be determined by computing the mean value. The matrix  $\mathbf{Q}_k$  and  $\mathbf{R}_k$  determine which set of data can be more reliable between the linear model and measured outcome of slider potentiometer.

Figure 7-5 illustrates the comparison between the measurements of slide potentiometer and results of Kalman filter when the slider moving up and down. From this figure, it can be observed that the slide potentiometer cannot provide a reliable displacements data which can possibly deteriorate the control performance of the system. Therefore, it is necessary to implement a Kalman filter on the slide potentiometer to remove the noise from the measurements of the sensor. In the experiments, the process covariance of the linear model is

$$\mathbf{Q}_{k} = \begin{bmatrix} 0.0000001 & 0\\ 0 & 0.0000001 \end{bmatrix}.$$
(7-8)

and  $\mathbf{R}_{k}$  is 32. In order to ensure the convergence before feeding Kalman filter data to the controller, the slider keeps still in the first second when the experiments start. A linear encoder (assumed to have more accurate measurement) is applied to verify the accuracy of the slide potentiometer. Figure 7-6 compares the measured outcome of a linear encoder with calculated result of the Kalman filter when the slider moves up and down. In this test, the

slider also keeps still in the first 1 second when the experiments starts. From the experimental results, there is no significant difference between the measurement of linear encoder and result of the Kalman filter, especially from 1 sec to 6 sec. The largest difference happens at the first second, which the error covariance of the Kalman filter starts to get convergence.



Figure 7-5 Comparison between measurement of slide potentiometer and results of Kalman filter.



Figure 7-6 Comparison between measurement of linear encoder and results of Kalman filter.

#### 7.2.2 Sensor fusion

From the previous section, we know that it is possible to improve the accuracy of the slide potentiometer simply by adding a Kalman filter. Each DC motor adopted in this study already has an encoder mounted on the shaft, which is an independent sensor. In order to enhance the accuracy of the displacement measurement, sensor fusion theory is applied in this study. The basic idea of the sensor fusion is to combine measurements from independent sensor sources into a single measurement in which way the noise can be efficiently reduced [102]. There are two major methods for senor fusion, one is based on the Central Limit Theorem, the other one is to use optimal Kalman filter [103]. Before combining the measurements of slide potentiometer as well as the encoder mounted on the shaft of the DC motor, it is necessary to convert the angular movement of the DC motor into the linear displacement of the slider. The ANFIS model of the twisted-string actuator, which has been derived in Chapter 3, describes the relationship between motor revolutions and linear

displacement of the slider as shown in Figure 3-14. In spite of high accuracy and zero noise, the measurement of the encoder mounted on the shaft of the DC motor still possibly exists error to some extent due to its indirect measurement. To obtain a fused measurement from different sensors, all the methods mentioned above need to take the noise of senor measurements into consideration. Compared to the noise of the encoder, processed errors caused by the ANFIS model of twisted-string actuators are the major reason of measurement errors. The fusion of the measurement data in this study can be developed similarly by applying the Central Limit Theorem to calculate the weighted arithmetic mean of the measurements of slide potentiometer and the output of the ANFIS model. The weighted arithmetic mean is:

$$\hat{Z}_{k} = \frac{\sigma_1 \cdot Z_{k1} + \sigma_2 \cdot Z_{k2}}{\sigma_1 + \sigma_2} \tag{7-9}$$

where  $Z_{k1}$  is the measurement of the slide potentiometer,  $\sigma_1$  is the noise variance of the slide potentiometer,  $Z_{k2}$  is the output of the ANFIS model of the twisted-string actuator,  $\sigma_2$  is the process error variance of ANFIS model and  $\hat{Z}_k$  is the fusion of the measurements.

Rewrite Eq. (7-9) as:

$$\hat{Z}_{k} = (1 - \mu) \cdot Z_{k1} + \mu \cdot Z_{k2}$$
(7-10)

where

$$\mu = \frac{\sigma_2}{\sigma_1 + \sigma_2} \tag{7-11}$$

The objective of sensor fusion is to minimize the estimate error variance of fusion of measurements. This estimate error variance can be defined as:

$$\mathbf{P}_{f} = E[(\hat{Z}_{k} - E[\hat{Z}_{k}])^{2}].$$
(7-12)

Substitute Eq. (7-10) into Eq. (7-12)

$$\mathbf{P}_{f} = E[\{Z_{k} - ((1-\mu) \cdot E[Z_{k1}] + \mu \cdot E[Z_{k1}])\}^{2}] \\ = E[\{(1-\mu) \cdot (Z_{k1} - x_{actual}) + \mu \cdot (Z_{k2} - x_{actual})\}^{2}] \\ = (1-\mu)^{2} \cdot E[(Z_{k1} - x_{actual})^{2}] + \mu^{2} \cdot E[(Z_{k2} - x_{actual})^{2}] + 2 \cdot \mu E[(Z_{k1} - x_{actual}) \cdot (Z_{k2} - x_{actual})]$$
(7-13)

where  $x_{actual}$  is the actual displacement of the slider. Since the measurement of slide potentiometer and the process noise of ANFIS model are assumed to be uncorrelated, the last term in Eq. (7-13) is equal to zero.

Rewrite Eq. (7-13) as:

$$\mathbf{P}_{f} = (1-\mu)^{2} \cdot \boldsymbol{\sigma}_{1} + \mu^{2} \cdot \boldsymbol{\sigma}_{2}$$
(7-14)

The estimate error variance  $\mathbf{P}_{\mathbf{f}}$  can be minimized by:

$$\frac{d}{d\mu}\mathbf{P}_{f}(\mu) = 0 \tag{7-15}$$

Substituting Eq. (7-14) into Eq. (7-15) gives

$$(1-\mu) \cdot \sigma_1 - \mu \cdot \sigma_2 = 0 \tag{7-16}$$

Solving Eq. (7-16) gives

$$\mu = \frac{\sigma_1}{\sigma_1 + \sigma_2} \tag{7-17}$$



#### Figure 7-7 Results of Sensor fusion.

In order to check the performance of sensor fusion, the two cycles of sine signal are selected as the input source of the twisted-string actuator, while the ANFIS model, slide potentiometer and linear encoder, whose measurement is considered as the reference, measure the displacement of the slider at the same time. In the experiments  $\sigma_1$  and  $\sigma_2$  are 0.4 and 0.6, separately. Figure 7-7 illustrates the measurements of different sensors. Table 7-1 illustrates the root mean square error of different sensors with respect to the reference. From the results of both figure and table, it is obvious the sensor fusion has the best performance and less uncertainty than ANFIS model and slide potentiometer when they are used individually. Therefore, it is feasible to combine the output of ANFIS model and slide potentiometer sensor, instead of using linear encoder, to obtain the high accuracy measurement in this study.

Sensor	Root Mean square error (mm)
ANFIS	0.36
Slide	0.20
potentiometer	
Sensor	0.12
fusion	0.15

Table 7-1 Comparison of different sensor measurements.
#### 7.3 Implementation of the embedded controller

This section describes the embedded system that is used to control the exoskeleton robot device performing specific rehabilitation tasks. The system is composed of four basic modules: the microprocessor module, the I/O (FPGA) module, the sensor module and the actuator.

Development for a controller is made of two phases: designing and implementation. In previous chapters, the adaptive robust controller for the twisted-string actuator has been successfully designed and tested through Matlab Simulink and xPC Target toolboxes. However, in practical application, the xPC target machine is cumbersome to carry out and not suitable for the stroke patients wearing the exoskeleton robot. Therefore, it is necessary to implement the controller using the embedded processor. Some prerequisites, such as the establishment of the cross-platform debugging environment, the design of PCB board as well as the peripherals and modular design, need to be accomplished before using the specific type of microprocessor to implement the controller. Fulfilling these prerequisites is always time-consuming and needs advanced electronic background. Therefore, to allow for fast prototyping and easy way to create devices and a programming environment to use sensors and actuators, the Arduino Platform, which is an open-source computer hardware, was introduced by Arduino Company in 2005 [104]. It is able to control the speed of DC motor and be able to display the speed of DC motor in LCD it in a matter of minutes, instead of hours. Meanwhile, the Arduino platform also provide various of free code libraries that save time on writing tons of low-level code.

In this study, we use an Intel Galileo 2, which is an Arduino-certified development boards based on Intel x86 architecture, to implement the embedded adaptive robust controller for the exoskeleton robot. The Intel Galileo 2 provides the highest operating speeds up to 400 MHz compared with other Arduino platform while keeping the compatibility with the Arduino shield ecosystem. The following section describes the architecture and key specifications of Intel Galileo 2.

### 7.3.1 Intel Galileo 2

Intel's Galileo Gen 2 Board has various features including an Intel Quark<sup>™</sup> SoC X1000 which is a 32-bit Intel Pentium<sup>®</sup> processor, 16 Kbytes L1 cache, 512 Kbytes of on-die embedded SRAM, 256 Mbytes DRAM, 11 Kbytes EEPROM, a full-sized mini-PCI Express slot, 100 Mb Ethernet port, 2-channel UART, and 12-bit pulse-width-modulation (PWM) [105].



Figure 7-8 System block diagram of the Intel Galileo 2 development board [105].

Features	Descriptions	
Physical Characteristics	• 10 cm long and 7 cm wide with the USB connectors, UART	
	jack, Ethernet connector, and power jack extending beyond	
	the former dimension	
	• Four screw holes allow the board to be attached to a surface	
	or case	
	• Reset button to reset the sketch and any attached shields	
Processor Features	• Instruction set architecture (ISA)-compatible 32-bit Intel®	
	Pentium <sup>®</sup> processor	
	• 16 Kbytes L1 cache	
	• 512 Kbytes of on-die embedded SRAM	
	• Simple to program: single thread, single core, constant	
	speed	
	ACPI-compatible CPU sleep states supported	
	• Integrated real-time clock (RTC) with optional 3V "coin	
	cell" battery for operation between turn on cycles	
	• 400 MHz clock speed	
Storage Options	• 8 Mbyte Legacy SPI Flash	
	• 512 Kbytes embedded SRAM	
	• 256 Mbytes DRAM	
	• USB storage works with any USB 2.0 compatible drive	
	• 11 Kbytes EEPROM programmed via the EEPROM library	

# Table 7-2 Main features of the Intel Galileo 2 development [105].

## 7.3.2 XILINX SPARTAN FPGA

In order to keep the compatibility with the Arduino shield ecosystem, the Intel Galileo 2 does provide the same numbers of I/O with the Arduino UNO platform. In this study 4 DC motors, 4 encoders, 4 slide potentiometers as well as 4 current sensors are required to drive or

acquire data. Since each DC motor is driven by H-bridge, it needs 3 digital output to control the speed and direction. Each encoder also needs 2 digital input to acquire the revolutions of DC motor, while each current sensor needs 1 analog input to obtain the displacements of the slide. Therefore, 20 digital I/O and 4 analog inputs are required to build the whole control system. However, it is impossible to only use 1 Intel Galileo 2 development board to implement the control system based on the I/O resource provided by the Intel Galileo 2. Indeed, another two Intel Galileo 2 can be employed to provide extra I/O resource to satisfy the requirements of the control system. The synchronization among the different processors in the control process will be the new issue affecting the stability of the system. Adopting a FPGA is a good option to avoid synchronization issue and provide enough I/O resource at the same time.

FPGA is an integrated circuit that can be configured by a designer after manufacturing. FPGAs provide an array of programmable logic blocks as well as reconfigurable interconnects allowing the blocks to be wired together which saves time in designing a circuit on a desktop computer and testing the circuit in minutes. In the designing process of the circuit, a schematic or a text file can be used to describe the function of the circuit or the desired logic function. In this study, XILINX Spartan XC3S200A is selected to expand the I/O resources.

The Spartan XC3S200A is designed for low-cost, high-performance applications and offers 4320 logic cells, 76 I/O pins, 622+ Mb/s data transfer rate per I/O, 18 single-ended signal standards and 8 differential I/O standards [106].

#### 7.3.3 System hardware architecture

The system hardware architecture block diagram is given in Figure 7-9. There is no doubt that performance of the embedded control systems depends not only on the correctness of the computation but also on the maximum sampling frequency that can be provided. Therefore, to reduce the calculation burden of microprocessor module and enhance the efficiency of the control algorithm, the Spartan XC3S200A is responsible for acquiring the

measurements from different sensors, such as the encoder, slide potentiometer as well as the current sensor, instead of the Intel Galileo 2.



Figure 7-9 Embedded control system block diagram.

Due to the existence of more than one chips in the embedded system, it is necessary to adopt a communication method to connect the Intel Galileo 2 with Spartan XCS200A. Both SPI (Serial Peripheral Interface) and I2C (Inter-Integrated Circuit) protocols are considered as good communication method among different devices.

Originally, the I2C bus was developed to connect a CPU to peripherals chips in a TV set using 2 wires. The maximum bus speed of I2C can arrive 3.4 Mbps that is even faster than peripherals chips [107]. Physically, I2C bus consists of SDA (serial data) and SCL (serial clock) lines. The devices that initiate a data transfer on the bus is regarded as the Master, the other devices on the bus are considered as Slave based on the I2C protocol specification. All devices' SDA lines are connected together, meanwhile, all devices' SCL line are connected together. Each device on the I2C bus has a unique address to avoid collisions in the process of communication and ensure the correct transfer of data among master and slaves. Comparing to I2C, SPI protocol is quite straightforward but need more signal wires to fulfill the communication among master and slaves. The SPI protocol requires two lines for data transfer. One is from the master to slaves, the other one is from slaves to master. Unlike the I2C protocol in which the Master uses the unique address to identify different slaves, the Master in SPI protocol uses the select signal for each slave. Although the extra I/O resources are required for SPI protocol, the implementation of code becomes easier by reason of the existence of the select line.

From the bus topology and resources point of view, the I2C is clearly better than SPI when there are more than two chips in the system. There are only two chips in our system, so the amount of required pins is not the major consideration. From the control theory point of view, the high-speed data transfer between the sensors and microprocessor plays the key roles in implementing the control system. Consequently, SPI is a better choice because it is full-duplex, however, the I2C is not. Theoretically, there is no speed limit on SPI. The speed of SPI often goes over 10 Mbps, meanwhile, the 3.4 Mbps is the maximum speed of I2C. Both SPI and I2C offer good support for communication with different devices, but SPI is better suited to our application in which devices need high-speed data transmission. In this study, the Intel Galileo 2 is used as an SPI master, while the Spartan XCS200A is used as an SPI slave.

The decoder for encoder mounted on DC motor as well as two ADCs for both slide potentiometer and current sensor are implemented by FPGA using VHDL programming language. The Spartan XCS200A also generate the PWM signal to the L298 H-bridge used to drive the DC motors.

#### 7.3.4 Software structure

The software flow diagram is shown in Figure. 7-10. Main responsibilities of the

software are controlling the twisted string actuators when the exoskeleton robotic arm is moving, acquiring and processing sensor data as well as doing initialized work. After the system passes all physical safety checks and is given the start signal, the initialization will start including setting up the SPI between the Intel Galileo 2 and Spartan XCS200A, pre-twisting the string to the operational range and turning on interrupts. Once the initialization is finished and the target training trajectory is selected, the MCU will generate the control signal to raise and drop the robotic arm following the target trajectory while keeping track of the rotations of elbow and shoulder joints. To guarantee the safety of a rehabilitation robot while operating, we need to implement safety halt function that deals with different hazardous conditions such as pressing the stop button and trigging the hard mechanical stop switch. This function is achieved through a state machine, as shown in Figure 7-11, that determines the moment of halting the robotic arm. There are five states: safety halt, stop button pressed, mechanical stop switch toggled, one for current limit toggles and normal operation. Either one of the mechanical stop switches is toggled, the safety button is pressed or the current limit is exceeded at any time during operation, the operation will halt and the system will enter into the safety halt state in which the power supply of the system will be cut until the twisted strings are untwisted to the original status, at which point the system will wait for the initialization to be finished to activate again.



Figure 7-10 Software structure.



Figure 7-11 Diagram of state machine.

## 7.4 Tracking motion trajectories of human upper limb

In Chapter 4 and 5, sinusoidal and trapezoidal trajectories are used as reference trajectories to verify tracking and synchronization performances of proposed controllers. Compared with actual motion trajectories of human movements, sinusoidal and trapezoidal trajectories are much more predictable. Therefore, it is not sure whether high tracking and synchronization performances can be obtained when the proposed ARC controller is applied to compensate for actual human motion trajectories. In particular, the proposed controllers were only verified on a single joint mechanism. Since the primary objective of proposed exoskeleton robot is to provide stroke patients shoulder and elbow assistance, it is necessary to verify that the proposed controller can be applied to both elbow and shoulder joints simultaneously. In this case, four twisted-string actuators need to be compensated at the same time.





In this section, the moving trajectories of elbow and shoulder joints obtained from Chapter 6 are adopted to verify the overall performance of the exoskeleton robot. The trajectories are shown in Figure 7-12. The physical setup of the exoskeleton robot does not include any rotary encoder at either elbow and shoulder joints due to the limitation of the mechanical design. For this reason, the unit of the moving trajectories need to be changed from the angular displacements of joints to the linear displacements of the twisted-string actuators. The rotational range of elbow joint (from 0 degree to 73 degrees) is mapped to the stroke of the dual twisted-string actuator, which ranges from 0 mm to 45 mm. The rotation range of shoulder joint (from 0 degree to 42 degrees) is mapped to the stroke of the other dual twisted-string actuators, which ranges from 0 mm to 26 mm.

To ensure the consistency of the exoskeleton robot, the robot must be tested using a repetitive motion. For the repetitive test, the elbow and shoulder joints are fed by two individual trajectories, as shown in Figure 7-12, at the same time. This ensures the exoskeleton robot can imitate simple movements of human upper limbs. Figure 7-13 demonstrates the results of the repetitive test. The results were acquired by performing 10 tests with the same trajectories. By comparing with experimental results shown in the previous chapter which only elbow joint is controlled, it is clear that there is no obvious discrepancy between tracking and repetitive performances. For the shoulder joints, to a certain extent both tracking and repetitive performances are worse than the performance that only shoulder is controlled. This discrepancy between experimental results and desired trajectory is the disturbance generated by the moving of elbow joint, which means that location changing of elbow at different time will lead to constantly changing the elbow inertia that can be considered as an external constantly changing load with respect to the shoulder joint.



Figure 7-13 Repetitive tests of the exoskeleton robot.

To ensure the stability of the exoskeleton robot, the robot also must be tested using external loads. For the test in this study, a 5 N external load was mounted at the endpoint of

the forearm, as shown in Figure 7-14. Figure 7-15 demonstrates the results of the stability tests. The results were acquired by performing 10 tests with the same trajectories. Based on the comparison between Table 7-3 and 7-4 (one without external load, the other one with an external load), it is clear that either maximum errors or RMSE of both elbow and shoulder joints are almost identical when the external load was mounted at the endpoint of the forearm. This ensures that the ARC controller can guarantee the tracking performance and stability even when the external load is applied. As to the shoulder joint, both maximum error and RMSE are worse than those of the elbow joint. This can be caused by the distance between the load and elbow joint is closer than the distance between the load and shoulder joint. Such a factor of external load can create a greater impact to the shoulder than to the elbow joint. In the experimental platform, identical DC motors are used to drive elbow and shoulder joints. However, the maximum torque for driving shoulder joint is at least 3 times greater than elbow joint. Therefore, it is clear that adopting the appropriate DC motor to drive the shoulder joint can improve the tracking performance of the exoskeleton robot when the external load is applied.

Figure 7-16 and 7-17 demonstrate the tracking errors with and without external load. With the increasing revolutions of driving motors, the friction among the strands increases as well based on both experimental results and Wurt's model. This friction can affect the performances of the controllers. Especially, the side effects of the friction become worse when joints need to change the direction of rotation. This is because that both type and direction of the friction among the strands are changed. The friction type is changed from dynamic friction to static friction when joints change the direction of rotation. Therefore, the controller must put more effort on adjusting the parameters immediately to compensate for the uncertainty in this situation. Before the parameters of the controller can be adjusted to appropriate values, the tracking errors may become greater. In the beginning of the experiment and when joints change their rotational directions, the parameters of the controller need to be adjusted and to be convergent again. Therefore, the tracking performance in these two stages is worse compared to other stages, as shown in Figure 7-16 and 7-17.

	Maximum tracking error	Root Mean square error
	(mm/degree)	(mm/degree)
Elbow joint	1.63 mm/2.66°	0.30 mm/0.50°
Shoulder joint	3.56 mm/5.82°	0.80 mm/1.31°

Table 7-3 Comparison of performance of elbow and shoulder joints.



Figure 7-14 External load mounted on the endpoint of elbow joint.



Figure 7-15 Stability tests of the exoskeleton robot.



Table 7-4 Comparison of performance of elbow and shoulder joints with 5 N external



In this chapter, the implementation of the robot including sensor measurement fusion, implementation of the embedded controller, and the experimental test has been introduced. From the experimental results, the designed exoskeleton upper limb system has the ability to provide stroke patients elbow and shoulder assistance in two degrees of freedom. The ARC controller also demonstrates the ability to maintain tracking performance and stability while an external load being applied.

# **CHAPTER 8:** Conclusion

## **8.1 Conclusions**

In this study, the development of a novel exoskeleton robot has been proposed with a specific aim to help stroke patients do upper limb rehabilitation. The size and the weight of the proposed exoskeleton robot were reduced by adopting twisted-string actuators without any intermediate devices, such as a gear box, to actuate. Unlike other existing exoskeleton robots, the actuators were placed on the patient's torso and the force was transmitted through twisted strings and Bowden cable mechanisms. The proposed robotic device is both agile and powerful.

Nonlinearity was a big issue of twisted-string actuation, which can make the controller design much more difficult. In this study, two adaptive robust control methods were investigated: adaptive robust control with cross-coupling dynamics based on displacement feedback and adaptive robust control based on force feedback, both of which can compensate for nonlinearities of the twisted-string actuator, such as hysteresis, friction, operational dead-zone, and poor repeatability. In addition, three different control schemes were synthesized for comparison: a simple PID controller, a PID controller with feedback of cross-coupling dynamics, and an LQR controller with feedback of cross-coupling dynamics. All algorithms are implemented and compared using a single joint of the robotic system. The comparison of experimental results has shown the importance of parameter adaption, robust consideration and using feedback of cross-coupling dynamics in designing high performance controllers for multi-axial and nonlinear systems. It is observed that in these experiments, the proposed adaptive robust controller achieves the best tracking and synchronization performance.

In addition to the control algorithms, a 2-DOF prototype of an assistive robotic system with elbow and shoulder joints is fabricated. The prototype device is light weight and is able to assist stroke patients to perform combined motion in the range of motion of the elbow and shoulder joints. A Microsoft Kinect depth camera was used to acquire 2-D motions of upper limbs of healthy adults. The acquired trajectories were decomposed into the angular displacements for corresponding joints of the exoskeleton robot. To derive a common trajectory of a specific motion from various people, Dynamic Time Warping (DTW) was adopted to normalize different trajectories and find the model.

To apply the derived motion to the fabricated assistive robotic device, Matlab xPC Target real-time platform was used to verify the design of the controller at the early stage of this study. To reduce the size of the overall system, the supplemental circuits and control units were implemented using two embedded devices. These two embedded devices are Intel's Galileo Gen 2 Board and an XILINX SPARTAN FPGA board, which were used to implement the data acquisition system and controllers.

## **8.2 Recommendations**

Though the proposed controllers have demonstrated excellent performance in tracking both angular position and force, additional research needs to be carried out to ensure the stability of the controlled results of twisted-string actuators with higher disturbances. In particular, the twisted-string actuator needs to provide high torque for driving the shoulder joint. While changing the angular position of the shoulder joint, the inertia also will vary in a wide range. This can be more critical if the robotic device is actually used by patients. Thus, a controller with higher accuracy and stability is required to track the desired trajectory and decrease safety issues caused by greater disturbances. Based on the experimental results, this actuation is not the best choice for providing high axial force for full assistance, due to the load sensitivity and stability. Thus, it is suggested to use the twisted string for light applications such as finger exoskeleton or elbow actuation. A high torque actuation such as a pneumatic or DC motor actuator will be a good choice for the shoulder joint. Also, an appropriate mechanical design should be developed to ensure safety if the twisted-string is accidentally broken. A more sophisticated design for the shoulder joint, which includes the scapular motion and does not undermine the affordability and simplicity of design should be developed for providing comprehensive rehabilitation.

Human–Robot Interaction is an important issue in designing a rehabilitation robot system that works with patients in everyday environments. The robot system should adjust the treatment plan and assistive force provided by perceiving the health condition and improvement of different stroke patients.

In addition, the synchronization of the dual twisted-string actuators could be mitigated by the feedback of cross-coupling dynamics. In this study, a time-delay is observed due to the slow start-up of one of the twisted-string actuators. This phenomenon can cause some vibration at joints while a motion is initiated.

### 8.3 Future work

With the proposed methods, the robotic system can demonstrate effective controlled outcome for both elbow and shoulder joints. However, some work and technologies still need to be developed. They are:

- A more sophisticated robotic device needs to be designed to reproduce arm motions with enough DOF's and lightweight;
- More experiments for extracting various trajectories from subjects with different arm motions will be carried out to derive more arm motion trajectories for the purpose of arm rehabilitation;
- An evaluation algorithm will be developed to determine the metric of individual patient's rehabilitation performance; and
- A distributed control framework that actively combines both ARC controllers for both position and force feedback while synchronizing all movements will be developed.

# REFERENCES

- Lum, P.S., et al., Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke. Archives of Physical Medicine and Rehabilitation, 2002. 83(7): p. 952-959.
- L.E. Kahn, P.S.L., W.Z. Rymer, and D.J. Reinkensmeyer, *Robot-assisted movement training for the stroke-impaired arm: Does it matter what the robot does?* J. of rehabilitation research and development, 2006. 43(5): p. 619.
- 3. Prange, G.B., et al., *Systematic review of the effect of robot-aided therapy on recovery of the hemiparetic arm after stroke*. Journal of Rehabilitation Research and Development, 2006. **43**(2): p. 171-183.
- Lo, A.C., et al., *Robot-Assisted Therapy for Long-Term Upper-Limb Impairment after Stroke*. New England Journal of Medicine, 2010. 362(19): p. 1772-1783.
- Gupta, A. and M.K. O'Malley, *Design of a haptic arm exoskeleton for training and rehabilitation*. Ieee-Asme Transactions on Mechatronics, 2006. 11(3): p. 280-289.
- 6. Riener, R., T. Nef, and G. Colombo, *Robot-aided neurorehabilitation of the upper extremities*. Medical & Biological Engineering & Computing, 2005. 43(1): p. 2-10.
- Kong, K. and M. Tomizuka, *Design of a Rehabilitation Device Based on a Mechanical Link System*.
   Journal of Mechanisms and Robotics-Transactions of the Asme, 2012. 4(3): p. 035001.
- 8. L. Jiang, R.S., M.H. Cheng, L.E. Banta, and G. Guo, *Moving trajectories and controller synthesis for an assistive device for arm rehabilitation*, in 2013 IEEE International Conf. on Automation Science *and Engineering* 2013. p. 268-273.
- 9. R. Shisheie, L.J., L.E. Banta, and M. Cheng, *Design and fabrication of an assistive device for arm rehabilitation using twisted string system*, in 2013 IEEE International Conf. on Automation Science and Engineering. 2013. p. 255-260.
- al, H.I.K.e., *Rehabilitation robotics: pilot trial of a spatial extension for MIT-Manus*. Journal of NeuroEngineering and Rehabilitation, 2004. 1(1): p. 5.
- J. Daly Janis J., e.a., *Response to upper-limb robotics and functional neuromuscular stimulation following stroke*. Journal of rehabilitation research and development, 2005. 42(6): p. 723.

- Masiero, S., et al., *Robotic-assisted rehabilitation of the upper limb after acute stroke*. Arch Phys Med Rehabil, 2007. 88(2): p. 142-9.
- 13. Saeki, S., et al., *Return to Work after Stroke a Follow-up-Study*. Stroke, 1995. 26(3): p. 399-401.
- Carod-Artal, J., et al., *Quality of life among stroke survivors evaluated 1 year after stroke Experience of a stroke unit.* Stroke, 2000. **31**(12): p. 2995-3000.
- 15. Riener, T.N.a.R., *ARMin Design of a novel arm rehabilitation robot*, in 2005 *IEEE International Conf* on *Rehabilitation Robotics*. 2005: Chicago, IL. p. 57-60.
- R. J. Sanchez, J., E. Wolbrecht, R. Smith, J. Liu, S. Rao, S. Cramer, T. Rahman, J. E. Bobrow, D. J. Reinkensmeyer, A Pneumatic Robot for Re-Training Arm Movement after Stroke: Rationale and Mechanical Design, in 2005 International Conf. on Rehabilitation Robotics. 2005: Chicago, IL, USA. p. 500-504.
- Y. Ren, H.-S.P., and L.-Q. Zhang, Developing a whole-arm exoskeleton robot with hand opening and closing mechanism for upper limb stroke rehabilitation, in 2009 IEEE International Conf on Rehabilitation Robotics. 2009: Kyoto, Japan. p. 761-765.
- 18. Mulder, T., *A process-oriented model of human motor behavior: toward a theory-based rehabilitation approach.* Phys Ther, 1991. **71**(2): p. 157-64.
- Zhou, H.Y. and H.S. Hu, *Human motion tracking for rehabilitation-A survey*. Biomedical Signal Processing and Control, 2008. 3(1): p. 1-18.
- 20. Winter, D.A., *Biomechanics and motor control of human movement 2nd Ed.* 1990, New York, NY: Wiley.
- 21. Admiraal, M.A., J.M.A.M. Kusters, and S.C.A.M. Gielen, *Modelling kinematics and dynamics of human arm movements*. Motor Control, 2004. **8**(3): p. 312-338.
- 22. J. Rosen, J.C.P., N. Manning, S. Burns, and B. Hannaford, *The human arm kinematics and dynamics during daily activities toward a 7 DOF upper limb powered exoskeleton*, in 12th International Conf on Advanced Robotics. 2005, 532-539: Seattle, WA.
- H.J. Nagarsheth, P.V.S., and M.A. Patel, *Modeling and dynamics of human arm*, in 4th IEEE Conf on Automation Science and Engineering. 2008: Washington DC. p. 924-928.
- 24. P.T. Katsiaris, P.K.A., and K.J. Kyriakopoulos, Modeling anthropomorphism in dynamic human arm movements, in 2010 IEEE/RSJ International Conf on Intelligent Robots and Systems. 2010: Taipei, 149

Taiwan. p. 3507-3512.

- 25. Morasso, P., Spatial control of arm movements. Exp Brain Res, 1981. 42(2): p. 223-7.
- 26. Flash, T. and N. Hogan, The Coordination of Arm Movements - an Experimentally Confirmed Mathematical-Model. Journal of Neuroscience, 1985. 5(7): p. 1688-1703.
- 27. Palli G, N.C., May C, Melchiorri C, and Wurtz T., Modeling and control of the twisted string actuation system. IEEE/ASME transMechatronics, 2013. 18: p. 664-673.
- 28. Würtz T, M.C., Holz B, Natale and C, Palli G., The twisted string actuation system: Modeling and control., in IEEE/ASME International Conference onAdvanced Intelligent Mechatronics. 2010. p. 1215-1220.
- 29. M. Tomizuka, J.H., T. Chiu, and T. Kamano, Synchronization of two motion control axes under adapative feedforward control. ASME J of

Dynamic Systems, Measurement and Control, 1992. 114: p. 196-203.

- 30. Leon, H.C.L.a.G.J., A neuro-controller for synchronization of two motion axes. International J of Intelligent System, 1998. 13: p. 571-586.
- 31. Chiu, H.S.a.G.T.-C., Motion Synchronization for Dual-Cylinder Electro-Hydraulic Lift Systems. IEEE/ASME Trans on Mechatronics, 2002. 7.
- 32. Tomizuka, G.T.-C.C.a.M., Coordinate position control of multi-axis mechanical systems. ASME J of Dynamic Systems, Measurement, and Control, 1998. 120: p. 383-393.
- 33. Xiao, Y., K.Y. Zhu, and H.C. Liaw, Generalized synchronization control of multi-axis motion systems. Control Engineering Practice, 2005. 13(7): p. 809-819.
- 34. You, S.-K.J.a.S.S., Precise position synchronous control of multi-axis servo system. Mechatronics, 2008. 18: p. 129-140.
- 35. Sun, D., Position synchronization of multiple motion axes with adaptive coupling control. Automatica, 2003. **39**(6): p. 997-1005.
- 36. Cheng, M.M., Mitra, A., & Chen, C. Y., Synchronization controller synthesis of multi-axis motion system., in Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on. IEEE. 2009.
- 37. Cheng, M.H., Y.J. Li, and E.G. Bakhoum, Controller Synthesis of Tracking and Synchronization for Multiaxis Motion System. Ieee Transactions on Control Systems Technology, 2014. 22(1): p. 378-386.

- 38. Oden, R., Systematic therapeutic exercises in the management of the paralyses in hemiplegia. Journal of the American Medical Association, 1918. 70(12): p. 828-833.
- 39. Twitchell, T.E., The restoration of motor function following hemiplegia in man. Brain, 1951. 74(4): p. 443-80.
- 40. Pulvermüller, F., Neininger, B., Elbert, T., Mohr, B., Rockstroh, B., Koebbel, P., & Taub, E, Constraint-induced therapy of chronic aphasia after stroke. Stroke, 2001. 32(7): p. 1621-1626.
- 41. Wittenberg, G.F., et al., Constraint-induced therapy in stroke: magnetic-stimulation motor maps and cerebral activation. Neurorehabil Neural Repair, 2003. 17(1): p. 48-57.
- 42. Miltner WL, B.H., Sommer M, Psych D, Dettmers C, Taub and E, Effects of constraint induced movement therapy on patients with chronic motor deficits after stroke. Stroke, 1990. 30(5): p. 86-92.
- 43. Taub E, M.N., Novack TA, Techniques to improve chronic motor deficit after stroke. Arch Phys Med Rehabil, 1993. 74(3): p. 47-54.
- 44. van der Lee JH, W.R., Lankhorst GJ, Vogelaar TW, Deville WL, Bouter LM., Forced use of the upper extremity in chronic stroke patients. Stroke, 1999. 30(23): p. 69-75.
- 45. Paci, M., Physiotherapy based on the Bobath concept for adults with post-stroke hemiplegia: a review of effectiveness studies. J Rehabil Med, 2003. 35(1): p. 2-7.
- 46. Brunnström, S., Movement therapy in hemiplegia: a neurophysiological approach. 1970: Facts and Comparisons.
- 47. Page, S.J., P. Levine, and A.C. Leonard, Effects of mental practice on affected limb use and function in chronic stroke. Archives of Physical Medicine and Rehabilitation, 2005. 86(3): p. 399-402.
- 48. Jackson, P.L., et al., Potential role of mental practice using motor imagery in neurologic rehabilitation. Archives of Physical Medicine and Rehabilitation, 2001. 82(8): p. 1133-1141.
- 49. Rushton, D.N., Functional electrical stimulation. Physiol Meas, 1997. 18(4): p. 241-75.
- 50. Winstein, C.J., et al., A randomized controlled comparison of upper-extremity rehabilitation strategies in acute stroke: A pilot study of immediate and long-term outcomes. Arch Phys Med Rehabil, 2004. **85**(4): p. 620-8.
- 51. Kassler, M., Robotics for health care: a review of the literature. Robotica, 1993. 11(6): p. 495-516.
- 52. Hogan, N., Krebs, H. I., Charnnarong, J., Srikrishna, P., & Sharon, A. MIT-MANUS: a workstation for manual therapy and training. in Robot and Human Proceedings. 1992. IEEE International Workshop.

- Cooper, R.A., Ohnabe, H., & Hobson, D. A. (Eds.), An introduction to rehabilitation engineering. 2006: CRC Press.
- 54. Furusho, J., et al., A 3-D rehabilitation system for upper limbs developed in a 5-year NEDO project and its clinical testing., in ICORR 2005. 9th International Conference on. IEEE. 2005.
- 55. He, J., Koeneman, E. J., Schultz, R. S., et al., *Design of a robotic upper extremity repetitive therapy device.*, in *ICORR 2005. 9th International Conference on. IEEE*. 2005.
- 56. Wei, R., Balasubramanian, S., Xu, L., & He, J., *Adaptive iterative learning control design for RUPERT IV.*, in 2nd IEEE RAS & EMBS International Conference on. IEEE. 2008.
- 57. Rosen, J., & Perry, J. C., *Upper limb powered exoskeleton*. International Journal of Humanoid Robotics, 2007. **4**(3): p. 529-548.
- Perry, J.C., J. Rosen, and S. Bums, *Upper-limb powered exoskeleton design*. Ieee-Asme Transactions on Mechatronics, 2007. 12(4): p. 408-417.
- 59. Lum, P.S., et al., Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke. Arch Phys Med Rehabil, 2002. 83(7): p. 952-9.
- 60. Rehab Measures: Fugl-Meyer Assessment of Motor Recovery after Stroke. Available from: http://www.rehabmeasures.org/lists/rehabmeasures/dispform.aspx?ID=908.
- 61. Lo, A.C., et al., *Robot-assisted therapy for long-term upper-limb impairment after stroke*. N Engl J Med, 2010. **362**(19): p. 1772-83.
- 62. *Twisted* String Actuator. Available from: <u>http://www.lat.uni-saarland.de/index.php/en/research/twisted-string-actuator</u>.
- 63. Sonoda, T., & Godler, I., Multi-fingered robotic hand employing strings transmission named "Twist Drive, in 2010 IEEE/RSJ International Conference on IEEE. 2010. p. 2733-2738.
- 64. Aukes, D., et al. . Selectively compliant underactuated hand for mobile manipulation. in Robotics and Automation (ICRA), 2012 IEEE International Conference. 2012.
- 65. Popov, D., Gaponov, I., & Ryu, J. H., 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. IEEE. 2012.
- 66. Mori, T., Tomoki Hiramatsu, and Eiji Shamoto., *Simultaneous double-sided milling of flexible plates* 152

with high accuracy and high efficiency—Suppression of forced chatter vibration with synchronized single-tooth cutters. Precision Engineering, 2011. **35**(3): p. 416-423.

- 67. Koren, Y., *Cross-Coupled Biaxial Computer Control for Manufacturing Systems*. Journal of Dynamic Systems Measurement and Control-Transactions of the Asme, 1980. **102**(4): p. 265-272.
- 68. Koren, Y., and Ch-Ch Lo, *Variable-gain cross-coupling controller for contouring*. CIRP Annals-Manufacturing Technology, 1991. **40**(1): p. 371-374.
- Ishizaki, K., Burak Sencer, and Eiji Shamoto, Cross Coupling Controller for Accurate Motion Synchronization of Dual Servo Systems. Journal ref: International Journal of Automation Technology 2013. 7(5): p. 514-522.
- 70. Slotine, J.-J.E., and Weiping Li, *Applied nonlinear control*. 1991, Prentice-hall.
- Petros Ioannou, B.F., Adaptive Control Tutorial (Advances in Design and Control). 2006: SIAM, Society for Industrial and Applied Mathematics
- 72. Perruquetti, W., and Jean-Pierre Barbot., *Sliding mode control in engineering*. 2002: CRC Press.
- 73. Ioannou, P.A., and Jing Sun, *Robust adaptive control*. 2012: Courier Corporation.
- 74. Yao, B. and M. Tomizuka, *Adaptive robust control of SISO nonlinear systems in a semi-strict feedback form.* Automatica, 1997. **33**(5): p. 893-900.
- 75. Hoffmann, T., Russell, T., & Cooke, H., *Remote measurement via the Internet ofupper limb range of motion in people who have had a stroke*. Journal of Telemedicineand Telecare, 2007. **13**(8): p. 401-405.
- 76. Holden, M.K., T.A. Dyar, and L. Dayan-Cimadoro, *Telerehabilitation using a virtual environment improves upper extremity function in patients with stroke*. Ieee Transactions on Neural Systems and Rehabilitation Engineering, 2007. 15(1): p. 36-42.
- Su, C.-J., Personal rehabilitation exercise assistant with kinect and dynamic time warping.
   International Journal of Information and Education Technology, 2013. 3(4): p. 448-454.
- 78. Chang, Y.J., S.F. Chen, and J.D. Huang, *A Kinect-based system for physical rehabilitation: a pilot study for young adults with motor disabilities.* Res Dev Disabil, 2011. **32**(6): p. 2566-70.
- 79. Chang, C.Y., Lange, B., Zhang, M., et al., *Towards pervasive physical rehabilitation using Microsoft Kinect.*, in *Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, 2012 6th International Conference on. IEEE. 2012.
- 80. Sugeno, M. and G.T. Kang, *Structure Identification of Fuzzy Model*. Fuzzy Sets and Systems, 1988. 153

**28**(1): p. 15-33.

- 81. *xPC Target*. Available from: <u>http://www.mathworks.com/products/datasheets/pdf/xpc-target.pdf</u>.
- 82. Jang, J.S.R., & Sun, C. T., *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*. 1996: Prentice-Hall, Inc.
- 83. Cailliet, R., *The Shoulder in Hemiplegia*. 1980: F.A. Davis Co.
- 84. S. K. Manna and S. Bhaumik, A bioinspired 10 DOF wearable powered arm exoskeleton for rehabilitation. J. of Robotics, 2013.
- 85. Marvin H. Cheng, L.J., Steve Wheeler, Reza Shisheie, and Ezzat Bakhoum, *Design, Fabrication, and Control of Twisted Actuated Robotic Device*, in *American Control Conference*. 2016.
- 86. Cheng, M.H., & Bakhoum, E. G., Adaptive robust control of tracking and synchronization for multi-axis motion system, in American Control Conference (ACC). 2011.
- 87. Zhou, J., C.Y. Wen, and Y. Zhang, *Adaptive backstepping control of a class of uncertain nonlinear systems with unknown backlash-like hysteresis*. Ieee Transactions on Automatic Control, 2004. **49**(10): p. 1751-1757.
- Debeljković, D.L., Lazarević, M. P., Milinković, S. A., Finite time stability analysis of linear time delay systems: bellman-gronwall approach., in 1st IFAC Workshop on Linear Time Delay Systems. 1998.
- 89. Chang, E. and M. Lee. *Stroke Recovery with Kinect*. Available from: http://research.microsoft.com/en-us/projects/stroke-recovery-with-kinect/.
- 90. Jana, A., *Kinect for Windows SDK Programming Guide*. 2012: Packt Publishing Ltd.
- 91. *Kinect for Windows Sensor Components and Specifications*. Available from: https://msdn.microsoft.com/en-us/library/jj131033.aspx?f=255&MSPPError=-2147217396.
- 92. *Tracking Users with Kinect Skeletal Tracking*. Available from: https://msdn.microsoft.com/en-us/library/jj131025.aspx?f=255&MSPPError=-2147217396.
- 93. Wolf Motor Function Test (WMFT) Manual. Available from: http://www.uab.edu/citherapy/images/pdf\_files/CIT\_Training\_WMFT\_Manual.pdf.
- Bennewitz, M., et al., *Learning motion patterns of people for compliant robot motion*. International Journal of Robotics Research, 2005. 24(1): p. 31-48.
- 95. Keogh, E. and C.A. Ratanamahatana, *Exact indexing of dynamic time warping*. Knowledge and 154

Information Systems, 2005. 7(3): p. 358-386.

- 96. Salvadora, S. and P. Chan, *Toward accurate dynamic time warping in linear time and space*. Intelligent Data Analysis, 2007. 11(5): p. 561-580.
- 97. Abdulla, W.H., Chow, D., & Sin, G, Cross-words reference template for DTW-based speech recognition systems, in TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region. 2003.
- 98. *indespension*. Available from: <u>http://www.indespension.co.uk/Trailer-Parts/Alko-Cables/BR062.html</u>.
- 99. *Kalman filter*. Available from: https://en.wikipedia.org/wiki/Kalman\_filter.
- 100. STENGEL, R.F., Optimal control and estimation. 1994, New York: Dover publications.
- 101.
   Kalman
   Filter
   For
   Dummies.
   Available
   from:

   http://bilgin.esme.org/BitsBytes/KalmanFilterforDummies.aspx.
   Available
   from:
- 102. Haghighat, M.B.A., Ali Aghagolzadeh, and Hadi Seyedarabi, *Multi-focus image fusion for visual sensor networks in DCT domain*. Computers & Electrical Engineering, 2011. **37**(5): p. 789-797.
- 103. Einicke, G., Smoothing, Filtering and Prediction Estimating The Past, Present and Future. 2012: InTech.
- 104. Arduino. Available from: https://en.wikipedia.org/wiki/Arduino.
- 105. center, I.E.d., Intel® Galileo Gen 2 Development Board.
- 106. Spartan-3 FPGA Family:Introduction and Ordering Information. 2013; Available from: http://www.xilinx.com/support/documentation/data\_sheets/ds099.pdf.
- 107.
   Introduction
   to
   I C
   and
   SPI
   protocols.
   Available
   from:

   http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/.
   http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/.
   http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/.