MODEL-BASED EVALUATION OF THE CONTROL STRATEGIES OF A HAND REHABILITATION ROBOT BASED ON MOTOR LEARNING PRINCIPLES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ONUR CAN YAĞMUR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

JANUARY 2022

Approval of the thesis:

MODEL-BASED EVALUATION OF THE CONTROL STRATEGIES OF A HAND REHABILITATION ROBOT BASED ON MOTOR LEARNING PRINCIPLES

submitted by **ONUR CAN YAĞMUR** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Mehmet Ali Sahir Arıkan
Head of the Department, **Mechanical Engineering** ________________

Assist. Prof. Dr. Ali Emre Turgut
Supervisor, **Mechanical Engineering, METU** ________________

Assist. Prof. Dr. Kutluk Bilge Arıkan
Co-Supervisor, **Mechanical Engineering, TED Uni.** ________________

Examining Committee Members:

Assoc. Prof. Dr. M. Bülent Özer

Mechanical Eng, METU ________________

Assist. Prof. Dr. Ali Emre Turgut
Mechanical Eng, METU ________________

Assoc. Prof. Dr. Ender Yıldırım
Mechanical Eng, METU ________________

Assist. Prof. Dr. Kutluk Bilge Arıkan
Mechanical Eng, TED Uni. ________________

Assist. Prof. Dr. Selçuk Himmetoğlu
Mechanical Eng, Hacettepe Uni. ________________

Date: 28.01.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Last Name : Onur Can Yağmur

Signature :

ABSTRACT

MODEL-BASED EVALUATION OF THE CONTROL STRATEGIES OF A HAND REHABILITATION ROBOT BASED ON MOTOR LEARNING PRINCIPLES

Yağmur, Onur Can
Master of Science, Mechanical Engineering
Supervisor: Assist. Prof. Dr. Ali Emre Turgut
Co-Supervisor: Assist. Prof. Dr. Kutluk Bilge Arıkan

January 2022, 62 pages

Stroke is an important health problem that occurs after blockage or bleeding in the vessels feeding the brain. It is one of the leading causes of death in the world. Patients with surviving hemiplegia often have a loss or decrease in voluntary movement of the right or left side of the body. The disease reduces the quality of life of patients with its spasticity and limits their independence; This situation poses an important problem for patients, their relatives, and the whole society. This thesis aims to develop control strategies for a robotic hand rehabilitation exoskeleton, which focuses on recovery based on motor learning principles, for patients with hemiplegia. The targeted system is designed to support/encourage motor learning. In this way, it is aimed to increase the effectiveness of therapy. Simulink/ MATLAB is used for both the simulations of the mechanism and the patient. To maximize motor learning in the therapy process, it has been suggested that the control system should be developed under assist as needed approach. Detailed mathematical models that describe the resistance torques in the joints include the mechanics of contact and interaction with the object squeezed during the pinching action. Reinforcement learning is used to model the cortical reorganization and to simulate the patient in the therapy process. Kinematic and admittance control strategies are used for rehabilitation simulations and slacking hypothesis is investigated. Admittance controller results in less slacking compared to kinematic controller. Moreover, the effect of spring stiffness in the admittance controller over learning is investigated.

Keywords: Rehabilitation Robotics, Kinematic Controller, Admittance Controller, Reinforcement Learning

ÖZ

EL REHABİLİTASYON ROBOTU DENETÇİLERİNİN MOTOR ÖĞRENME İLKELERİNE DAYALI MODEL ÜZERİNDE KIYASLANMASI

Yağmur, Onur Can
Yüksek Lisans, Makina Mühendisliği
Tez Yöneticisi: Assist. Prof. Dr. Ali Emre Turgut
Ortak Tez Yöneticisi: Assist. Prof. Dr. Kutluk Bilge Arıkan

Ocak 2022, 62 sayfa

İnme, beyni besleyen damarlarda tıkanma veya kanama sonrası oluşan önemli bir sağlık sorunudur. Dünyada önde gelen ölüm nedenlerinden biridir. Sağ kalan hemiplejili hastalarda genellikle vücudun sağ veya sol tarafının istemli hareketinde bir kayıp veya azalma olur. Bu durum hastalar, hasta yakınları ve tüm toplum için önemli bir sorun teşkil etmektedir. Bu tez, hemiplejili hastalarda motor öğrenme ilkelerine dayalı iyileşmeye odaklanan bir el rehabilitasyon robotu için kontrol stratejileri geliştirmeyi amaçlamaktadır. Hedeflenen sistem, motor öğrenmeyi desteklemek/teşvik etmek için tasarlanmıştır. Bu şekilde terapinin etkinliğinin arttırılması hedeflenmiştir. Mekanizma, denetçi yapısı ve hasta modeli için Simulink/ MATLAB programı kullanılmıştır. Terapi sürecinde motor öğrenmeyi en üst düzeye çıkarmak için, kontrol sisteminin gereği kadar destek prensibi altında geliştirilmesi önerilmiştir. Pekiştirmeli öğrenme, kortikal yeniden yapılanmayı modellemek ve terapi sürecinde hastayı simüle etmek için kullanılmıştır. Rehabilitasyon simülasyonları için kinematik ve admitans denetçiler kullanılmış ve tembellik hipotezi araştırılmıştır. Admitans denetçinin, kinematik denetçiye kıyasla daha az tembelliğe sebep olduğu görülmüştür. Ayrıca, admitans denetçideki yay sertliğinin öğrenme üzerine olan etkisi araştırılmıştır.

Anahtar Kelimeler: Rehabilitasyon Robotiği, Kinematik Denetçi, Admitans Denetçi, Pekiştirmeli Öğrenme

To my family and loved ones

ACKNOWLEDGMENTS

I would like to express my heartfelt appreciation to my supervisor and co-supervisor, Assist. Prof. Dr. Ali Emre Turgut and Assist. Prof. Dr. Kutluk Bilge Arkan, for their guidance, advice, criticism, encouragement, and, most importantly, their friendship and understanding throughout my three-year journey toward earning my master's degree. This study would not be possible without their encouragement and support.

I would like to thank my friends, research assistants Nursel Güler Kasal and Kaan Gürel for their support and encouragement.

I would like to show my gratitude to my friends Adnan Bora Baykal, Bilal Aydoğdu, Çağatay Altuntaş, Güney Çimen, and Ghazal Khodkar for their support and encouragement.

I am deeply grateful to my dear friend Umut Bekçi for his support and guidance, always being with me throughout this journey.

I would like to express my heartfelt gratitude to my family Aliye Yağmur, Yetkin Yağmur, and Eylül Yağmur for their unwavering support during my educational career.

Finally, I would like to express my deepest gratitude to my dear wife, Özge Nur Yağmur, for always being there for me whenever I need moral support and guidance.

This work is funded by the Scientific and Technological Research Council of Turkey under grant number TUBİTAK 121E107.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Hemiparesis and hemiplegia are the most common results of a stroke. An increasing number of people suffer from these conditions each year. Recovery methods have been developed over the past decades and among these robot-assisted rehabilitation therapy stands out from the conventional therapy methods. Early studies were mostly focused on the mechanical design and designing a robot that can help the patient to achieve the desired therapy such as following a trajectory of a reaching arm. Then, it is found out that if the robot fully assists the patient it results in laziness and harms the recovery process (slacking). Therefore, control methods of therapy robots have a great effect on rehabilitation robotics. Early studies mostly focus on impedance, stiffness, and force control of the various robots designed for reaching, grasping, walking, etc. Assistive robotic devices help patients to complete required rehabilitation task. However, it is found out that repeated movements with small errors result in no motor recovery, which harms the rehabilitation process by encouraging laziness of the impaired body part named as slacking.

There are various types of rehabilitation robots focused on the therapy of different limbs such as upper limb therapy robots, walking therapy robots, and robots for hand rehabilitation, etc. Hand rehabilitation is the stimulus of this study. A previously designed hand rehabilitation robot is used as a model and the aim is to improve and evaluate the control strategies for hand rehabilitation [1].

In this thesis, two control strategies namely, Proportional – Integral – Derivative (PID) and admittance control performances on motor learning during the therapy process are compared. A robotic hand exoskeleton rehabilitation robot designed by [2] is used for the model. The exoskeleton for the hand is designed for index and thumb fingers for the left hand and pinching of a spring action is modeled. Reinforcement learning is applied to model the motor learning of the patient during rehabilitation. It is aimed to simulate the slacking phenomena due to the kinematic control system, i.e., the PID based position control. In addition, the advantage of interaction type control system, i.e., the admittance control, upon the kinematic control is discussed. The performances are assessed for typical circular pinching motion including the thumb and index fingers equipped with the exoskeleton mechanism. The physical implementation on patients is out of the scope of the thesis.

CHAPTER 2

LITERATURE REVIEW

Recovery after stroke and robot-assisted rehabilitation is a complex process that requires the study of stroke, motor learning, rehabilitation robots, and control methods of the robots. In the following parts, these aspects of recovery are presented.

2.1 Stroke and Motor Learning

Stroke is a serious health problem that affects millions of lives each year. It is mainly caused by the death of brain cells which is a result of hemorrhage into the brain or blockage of oxygen-rich blood flow through arteries into the brain[3]. Stroke may result in disabilities in movement, talking, etc. as well as death [4]. It is stated by the World Health Organization (WHO) that over 15 million people suffer from stroke each year and the number of people that suffer from stroke is expected to be higher in the next decades [4], [5]. Due to the permanent damage in the brain, hemiparesis and hemiplegia are the most common long-term results. Most of the severely impaired patients cannot perform activities of daily living (ADL) and human or robotic assistance is required [6]. It is found out that upper limbs (i.e., hand) have more impact than others to perform ADL [5] – [7]. An impairment in hand function would have a significant influence on the patient's quality of life, implying a greater emphasis on hand motor rehabilitation. However, whereas most patients achieve good motor recovery in the proximal upper extremity, recovery in the distal upper extremity has been restricted owing to poor effectivity [10]. Fine motor skills are difficult to recover.

There are two primary explanations for the difficulties encountered during hand rehabilitation. To begin, the hand has more than 20 degrees of freedom (DOF) during movement, making it challenging for therapists or training equipment to match satiety and diverse movement requirements [11]. Second, the region of the cortex associated with the hand is much greater than that associated with the other motor cortex, implying a high degree of flexibility in creating a range of hand postures and controlling the specific joints of the hand. However, the majority of research to date has concentrated on the opposite, on the absence of individuation in finger motions [12], [13]. Improved rehabilitative treatments are critical. Robot-assisted treatment for post-stroke rehabilitation is a novel kind of physical therapy in which patients train their paretic limbs by using or resisting the robots' force [14]. For instance, the MIT-Manus robot uses an amassed training strategy to train the upper limbs by performing reaching actions [15]; the Mirror Image Movement Enabler (MIME) employs a bilateral training technique to educate the paretic limb [16].

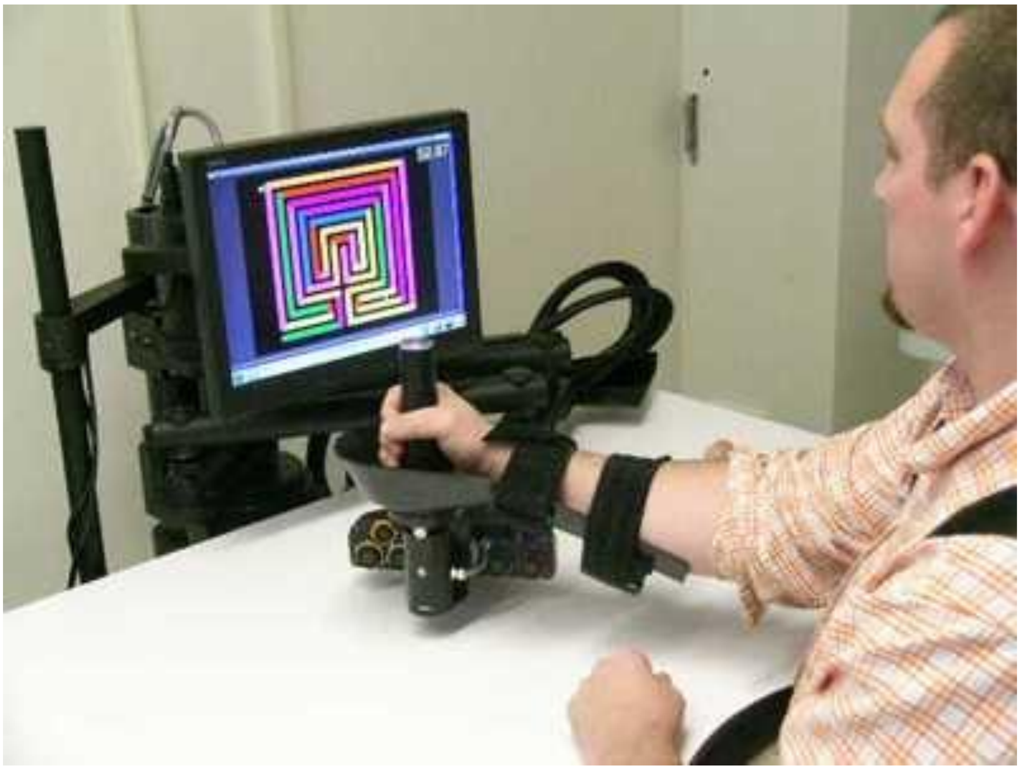

Figure 2.1. MIT Manus Robot [15].

Robot-assisted treatment has advanced significantly over the last three decades as a result of advancements in robotic technology and bioengineering and has established itself as a significant addition to conventional physical therapy [18], [19]. For example, in contrast to the therapist who is weary from teaching patients via physical work, exoskeleton given in [20] allows patients to move their fingers dexterously and repeatedly [20], [21]. Additionally, certain robots may be controlled by bio signals such as electromyography (EMG) and electroencephalography (EEG) signals derived from a patient's desire [22], [23]. These enable the creation of a closed-loop rehabilitation system using robotic technology, which is not achievable with any other kind of rehabilitation treatment [24].

One important method used in post-stroke rehabilitation is mirror therapy. In this method, a mirror is placed on the patient's weak extremity. The patient is asked to move his healthy extremity. The patient monitors the movement of the intact extremity as the movement of the stroked extremity. The method used here is based on the knowledge that thinking about movement and performing movement develop as a result of the activity of the same anatomical structures. In imaging studies, it has been shown that performing a movement and only thinking about that movement, thinking as if it is being done (imagination, motor imagery) or just watching the movement activate the common nervous system structures. In mirror therapy, it is aimed to activate the damaged motor cortex, which is not sufficiently activated by voluntary movement due to paralysis, by monitoring the movement and adapting the motor plasticity [25]. Robotic mirror therapy, on the other hand, aims to apply the same movement to the limb that has weakness or loss of function, using the robot, while monitoring the movement.

It has been observed that mirror neurons play a role in different areas such as imitation, social learning, motor learning, and interpretation of the observed movement [26]. It is emphasized that mirror neurons have an important role in cortical restructuring during the post-stroke therapy process [27]. The cortical processes and mechanisms used during motor skill acquisition in healthy people are also seen in the process of complete recovery with therapy. The model utilized in this thesis was built for a robotic mirror therapy system for hand rehabilitation [2], [28]. The simulation framework will be further developed to include the role of mirror neuron system on motor learning.

Mirror Neurons are a group of visual-motor neurons that are activated when observing an action and performing the same action. Neurons with this feature have been observed in the F5 and PF regions of the premotor cortex in monkeys [29]. Since it is not possible to observe a single mirror neuron electrophysiologically in humans, research is steered to the "mirror neuron system (MNS). There are studies showing that the neuron system, which is active in the ventral premotor cortex and the rostral inferior parietal lobule regions, mediates the observation and understanding of the action [30]. It is stated that mirror neurons take an active role in the mechanism behind mirror therapy and the related neuroplasticity which is the core mechanism behind the motor learning.

Motor learning can be defined as the improvement of fluency, accuracy, and precision of movement performance depending on experience [30]. Speaking, riding a bike, and playing an instrument are examples of complex motor control practices which are the outcomes of advanced motor learning process. In addition, calibrating reflexes according to the current body condition and environmental conditions can be given as an example of motor learning. In the learning process, depending on the feedback, useful actions are reinforced while the probability of performing unhelpful actions decreases. The motor skill attained by motor learning improves with practice and repetition. Motor outputs (movements) are realized through hierarchical neural structures. Inputs from spinal motor neurons trigger muscle movements. Motor neurons emerging from the spinal cord receive information about the environment and body position from sensory neurons reaching the spinal cord level; this type of information is generally useful for the realization of reflexive behaviors. These motor neurons also receive inputs about body position and balance from centers located in the brainstem and mesencephalon (midbrain). However, the main input of these neurons comes from neurons in the primary motor cortex (M1) via the corticospinal tract. In the brain, direct cortical information to M1 comes from the secondary motor cortex (M2), posterior parietal cortex (PPC), and prefrontal cortex (PFC) regions. PPK acts as a bridge between sensory cortical regions (primary visual cortex, primary auditory cortex, somatosensory cortex) and M1. In addition, the PFC also performs functions related to the cognitive aspect of the movement. M1 is modulated by two main feedback networks. The first of these modulatory links is the basal ganglia. Brain regions connected to this system perform important functions in selecting and triggering movements. The second modulatory system is managed by the cerebellum. The cerebellum plays a role in motor coordination and motor learning based on cortical feedback, as well as playing a role in the processing of motion predictions and feedback [31]. Recovery after stroke is governed by the same motor learning process as the one in healthy people. Therefore, recent rehabilitation protocols aim to understand the motor learning in healthy subjects and promote the motor re-learning sequences in stroke patients.

Existing assessments of the effects of hand rehabilitation robotics on post-stroke motor recovery are insufficient, with the majority of studies focusing on the use of robot-assisted treatment on other limbs rather than the hand [24]. Additionally, current studies emphasize either the hardware design of the robots or the use of specific training paradigms [24], [32], even though both are necessary components of an efficient hand rehabilitation robot. The hardware system acts as the basis for the robot's functionality. Together with the control system topology and performance, the overall robotic system has the potential to be the main source for motor recovery. In the following 2 sections hand rehabilitation robots in the literature and control methods for rehabilitation robots are going to be presented.

2.2 Hand Rehabilitation Robots

In kinematic terms, the human hand may have twenty degrees of freedom (DOFs). All fingers have four degrees of freedom. Each of the index, middle, ring, and little fingers has three joints. The joints are classified as follows: distal interphalangeal (DIP), proximal interphalangeal (PIP), and metacarpophalangeal (MCP). The DIP and PIP joints have flexion/extension degrees of freedom, while the MCP joints have flexion/extension and abduction/adduction degrees of freedom [33].

The thumb has three joints and four degrees of freedom. The joints are numbered from distal to proximal: interphalangeal (IP), metacarpophalangeal (MCP), and carpometacarpal (CMC) [33]. While the IP and MCP joints have flexion/extension degrees of freedom, the CMC joint has both flexion/extension and abduction/ adduction degrees of freedom.

Exoskeletons for the hands or active hand orthoses are a kind of robotic hand. In comparison to other types of robotic hands, a hand exoskeleton is a mechanical system that is actively controlled and linked to a human hand, allowing the two systems to move together by interaction. Rehabilitation exoskeletons are used to reduce the spasticity in hand by performing rehabilitation protocols and aim to reach the motor recovery.

There are several categorization schemes for hand rehabilitation robots, some of which follow mechanical design conventions (which concentrate on the hardware system) and others that follow rehabilitation conventions (which focus on the training paradigms) [34]. Indeed, each of these categorization schemes has inherent value and is dependent on the others. For instance, the hardware system is dependent on the rehabilitation robots' fundamental capabilities (e.g., movement and feedback information), while the training paradigms are the primary functional components of recovery (e.g., application of certain rehabilitation theories) [33]. In the following, hand rehabilitation robots are classified under two main categories with their mechanical design aspects.

2.2.1 End Effector Hand Rehabilitation Robots

The end effector is located exterior to the patient's body and supplies the necessary force to the end of the user's extremities to assist or resist motion. For instance, a robotic system AMADEO which is also a commercial product given in Figure 2.2 is a specialized robot that performs task-related training by automating the extension and flexing of fingers sequentially or simultaneously. The linear (2 DOF) forward and backward motion simulates the grabbing action in a continuous and ergonomic manner, after attaching, the fingertip and thumb are supported by the finger and bending and stretching actions may be done in conjunction with the slider. Each patient's strength and tempo were specifically adjusted. The total number of gripping motions, their speed, strength, and range of motion were recorded and used as recovery data [35].

Figure 2.2. AMADEO Hand rehabilitation robot [36]

HandCARE given in Figure 2.3 is another end effector aims to provide an interface for training distal regions of the upper limbs that combines the benefits of both non-actuated and active robotic devices [37]. Each finger is connected to an instrumented cable loop, allowing for force control and linear movement [33]. The end effector generates force without regard to the particular joint movements of the patients' limbs, resulting in limitations in range of motion and dead point difficulties. It is intended to allow poststroke patients to train at home or in rehabilitation clinics via the use of virtual reality (VR) game-like activities. Additionally, the interface is adaptable to the biomechanics of the patient, easy to use, and be cost-effective [37].

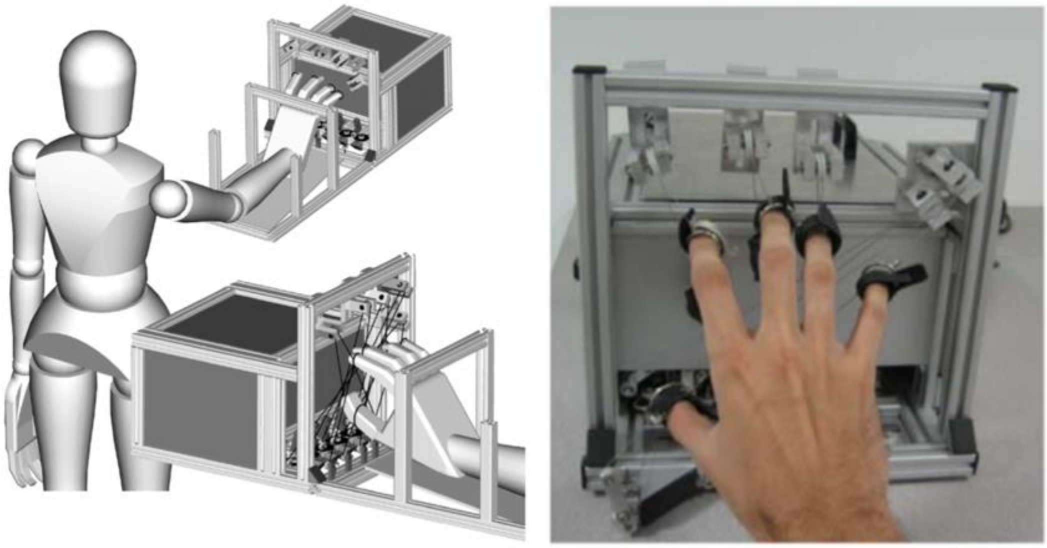

Figure 2.3. HandCARE Hand rehabilitation robot [37]

2.2.2 Exoskeleton Hand Rehabilitation Robots

The exoskeleton rehabilitation robots are to be worn by patients while the robot's joints and linkages match directly with the joints and limbs of the patient [38]. For instance, an exoskeleton given in Figure 2.4 is a robotic device detects the intention of the patient with impaired hand by measuring the EMG signals and assists the hand for open/ close actions [39].

Each hand module is made up of five finger assemblies and a platform for the palm. Each finger assembly is operated by a single linear actuator, and the mechanical linkage system offers two degrees of freedom for each finger at the MCP and PIP. The finger assembly offers a range of motion of 55 degrees and 65 degrees for the MCP and PIP joints, respectively, from fully extended to completely flexed position [39].

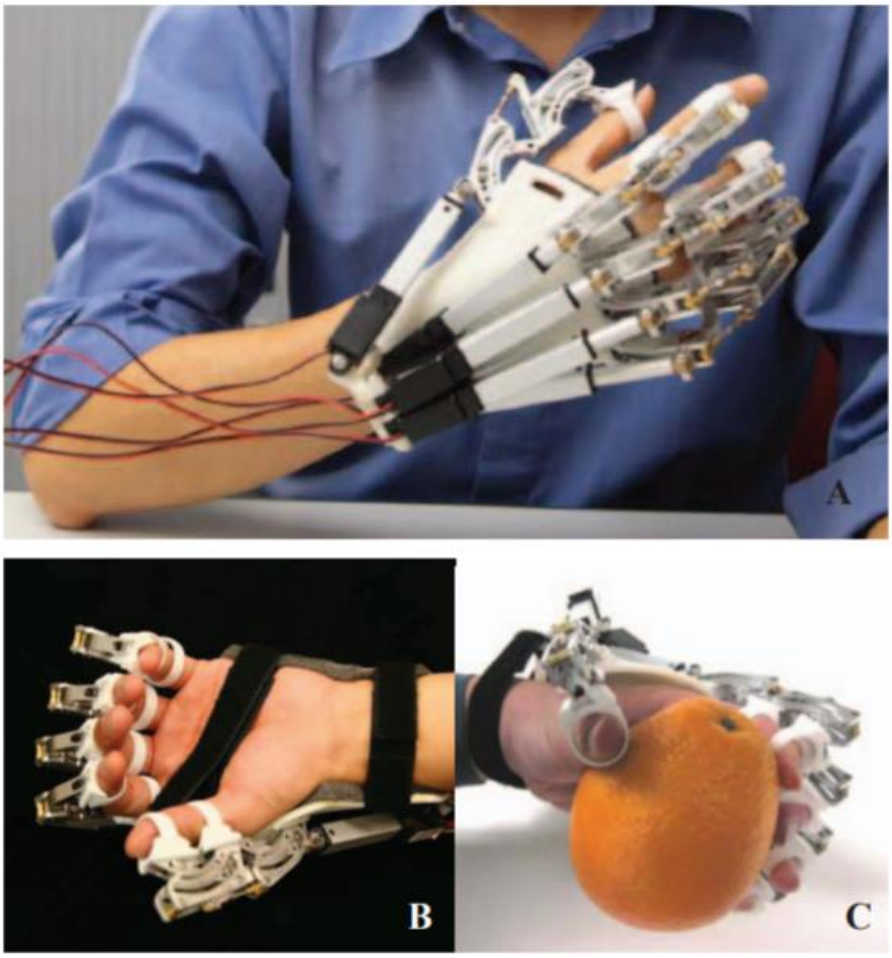

Figure 2.4. A hand exoskeleton robot [39]

Another exoskeleton device for hand rehabilitation, HANDEXOS, which is compact and lightweight given in Figure 2.5. The mobility of the exoskeleton makes it an attractive option for stroke rehabilitation, particularly for patients in the latter stages of the disease who can train at home [40].

The primary concern of HANDEXOS is to provide complete hand mobility with natural rotational movement and to do this, the number of degrees of freedom is equivalent to that of the natural hand skeleton. Additionally, the design criteria were tried to be kept as inclusive as possible. Average values of 51mm, 26mm, and 25mm for the index finger from the proximal to the distal phalanx were chosen, but the exoskeleton was designed to partially fit over hands of varying sizes via a passive and adjustable mechanism on the intermediate phalanx [40].

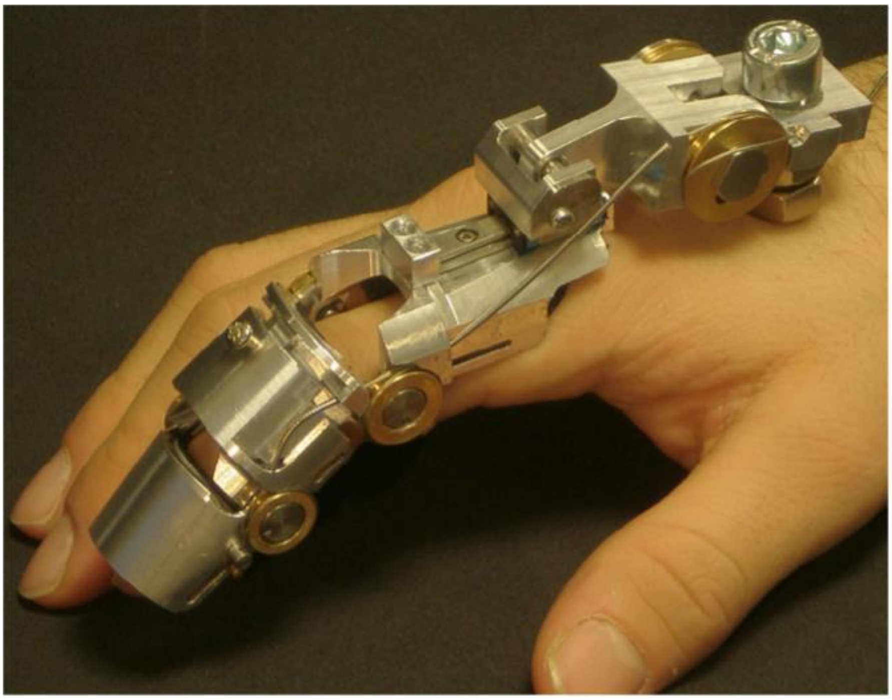

Figure 2.5. HANDEXOS rehabilitation robot attached to index finger [40]

Although there are challenges, such as aligning the robot's axes with the anatomical axes of the hand, exoskeleton robots are commonly utilized in rehabilitation robotics and have advanced significantly in recent years. The use of functional degrees of freedom to simplify complicated multi-DOF motions and the development of soft-bodied robots both enhance the use of exoskeleton robotics [41]. Nowadays, exoskeleton robots becoming more popular in post-stroke rehabilitation.

2.3 Control Methods for Rehabilitation Robots

The aim of the post-stroke rehabilitation process is to reach motor recovery by the application of selected/designed therapy protocols. The purpose of the control algorithms is to promote motor learning network to gain motor recovery. Control algorithms can be grouped under the following categories:

- Assistive
- Challenge-based
- Haptic simulation (ADL)

Among these 3 categories, the most important and developed algorithm used for rehabilitation is the assistive control approach. Assistive control algorithms basically help patients to complete the required rehabilitation task such as pinching, grasping, walking, reaching, etc. Challenge-based control algorithms apply resistance to the rehabilitation movement such as counterforce field or extra stiffness. In haptic simulation-based approaches, activities of daily living (ADL) are simulated in a virtual environment [42]. In the following sections, assistive controllers are going to be presented.

2.3.1 Assistive Controllers

As mentioned before, the aim of the assistive controllers is to help patient to make the movement however, it is found out that too much assistance results in counter effect on the recovery and causes slacking. For example, an upper limb, adaptively controlled, compliant rehabilitation robot to provide reaching tasks with "taking over" a reaching job from the patient, lowered patient’s own force production, resulted in the robot to perform more of the effort than patient of raising their arm [43]. According to the motor learning perspective, motor error and variability are required for learning. Kinematically stiff control systems don’t let the motor system make error and consequently neuroplasticity mechanisms are being provoked. Similar findings suggest that providing too much assistance harms the rehabilitation process. As a result, “assist as needed” approach is developed so that assistive controllers to aid as little as possible to patient to accomplish the rehabilitation task. In Figure 2.6 an assist as needed controller given is applied to the Pneu-WREX [44] upper extremity rehabilitation robot.

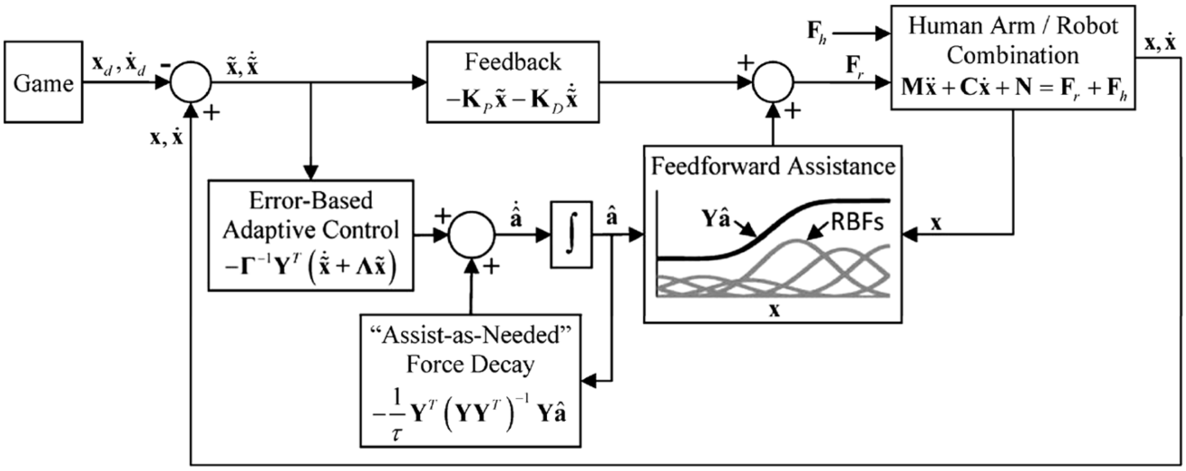

Figure 2.6. Assist as needed controller diagram [45]

Assistive controllers can be grouped under five categories as follows:

- Impedance-based
- Admittance-based
- Counterbalance-based
- EMG-based
- Performance-based

2.3.1.1 Impedance-Based Assistance

Impedance control is used to create a desirable dynamical connection between the location of the end-effector and the applied force. Mechanical impedance Z is the relationship between velocity $\dot{X}$ and applied force F. The control loop has the effect of altering the manipulator's damping constant as it comes into touch with the surroundings.

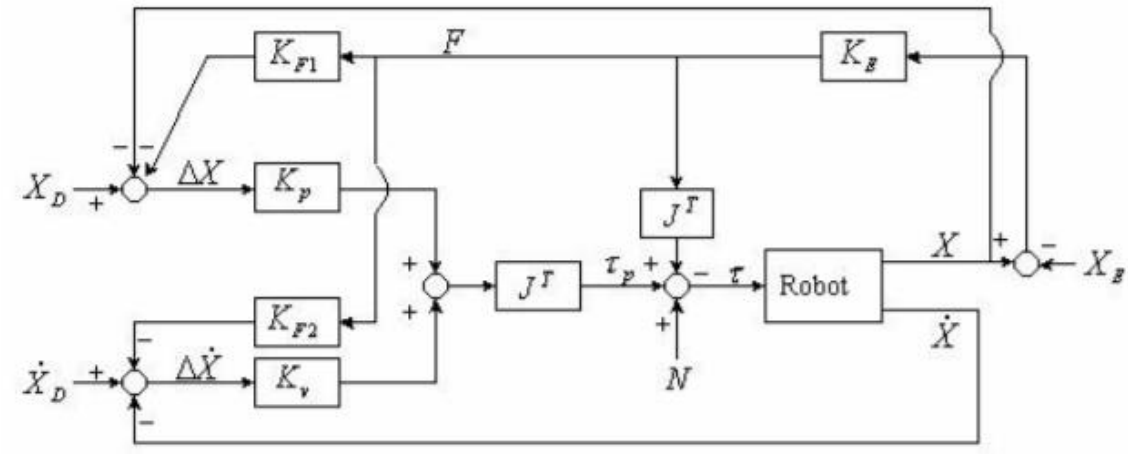

Figure 2.7. Impedance control scheme of a robot manipulandum [46]

Assistive controllers should not provide input to the robot as the patient moves on the desired trajectory. When the patient deviates from the desired trajectory, it is expected the robot to intervene and keep the patient on the desired trajectory (reduce error) by creating a restoring input due to a mechanical impedance. For example, a Proportional – Derivative (PD) controller output would increase as the patient deviates from the desired trajectory since the controller acts as a damped spring. A type of impedance-based assistance is triggered assistance. In triggered assistance, the participant can try a movement without the help of the robot but automatically begins some sort of impedance-based aid when a performance variable hits a threshold value. This kind of triggered assistance promotes self-initiated movement, which is regarded to be crucial for motor learning [47]. A variation of the triggered assistance is used to control the hand rehabilitation robot HWARD [48]. If the force applied by the patient is below a threshold for a fixed time and/or the patient cannot complete the task, the robot assists the patient to finish the task. This approach is called time-triggered assistance.

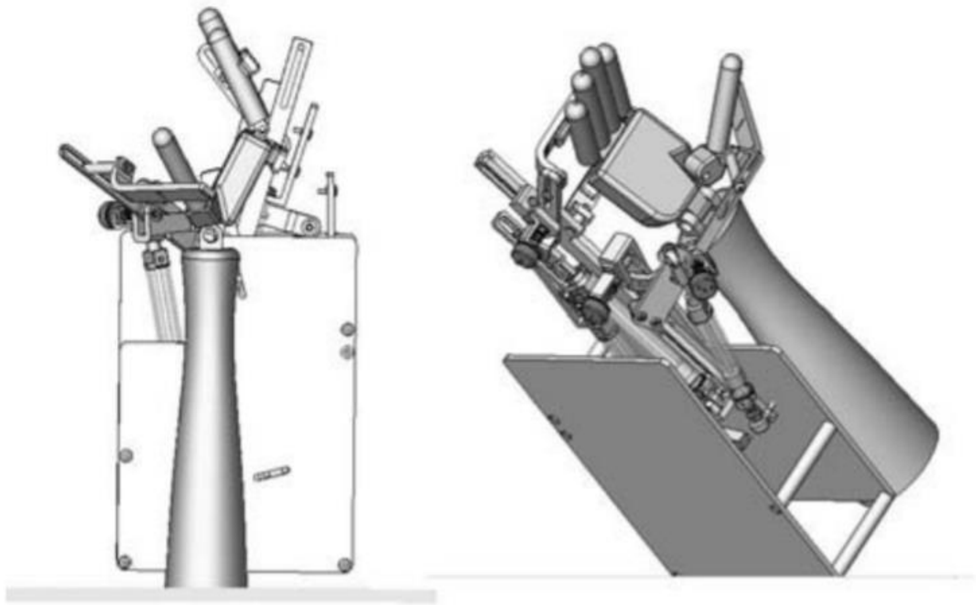

Figure 2.8. HWARD hand rehabilitation robot.

2.3.1.2 Admittance-Based Assistance

The admittance control specifies a force setpoint, which is monitored by a force compensator. In contrast to control strategies that rejects disturbance forces in order to maintain a set reference motion trajectory, the force compensator aims to conform to environmental interactions and respond fast to contact forces by rapidly altering the reference motion trajectory [46]. The admittance matrix A in Figure 2.9 is used to link the force error vector E to the end-effector velocity perturbation. Effective and accurate admittance control may be done by selecting an A matrix that is appropriate for the environment's known stiffness. When the environment changes dramatically, A matrix should be updated to adapt to the new situation. Adaptive algorithms enable the admittance value to be changed in response to changes in the environment [46]. Admittance control structure is suitable for robotic applications that contain human factor (i.e., rehabilitation) since human interaction causes an unpredictable change in the environment.

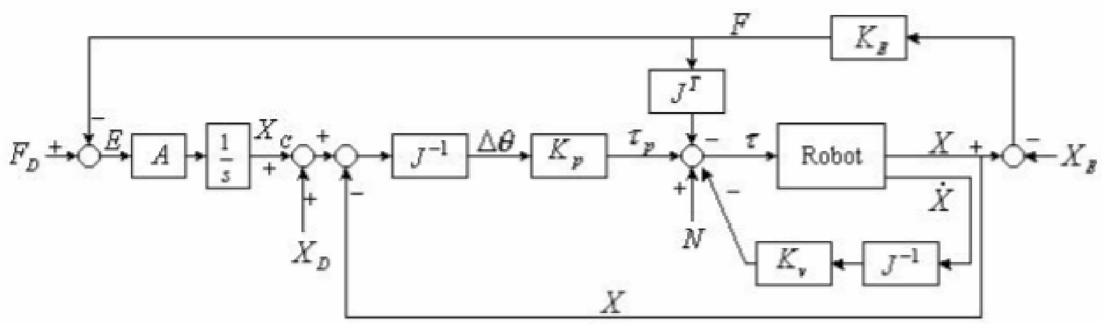

Figure 2.9. Admittance control scheme of a robot [46]

2.3.1.3 Counterbalance-Based Assistance

Another assistive method is to provide a limb with a counterweight of its own weight. Patients in rehabilitation facilities have long relied on equipment such as arm skateboards or towels that slide on tables and harnesses to offset the weight of their bodies when they walk to help them regain their mobility. When it comes to rehabilitation, the use of swimming pools may also be seen as a variation of this approach: buoyancy aids active assistance.

Newer devices include passive counterbalancing systems that allow for a broader range of motion than previously used clinical devices [49]. There are several options for counterbalancing the weight of an arm, such as Therapy-WREX, which is based on WREX's movable arm support and incorporates two four-bar linkages and elastic bands [49]. A therapist may adjust the amount of support provided by adding or removing elastic bands based on the degree of impairment shown by the participant.

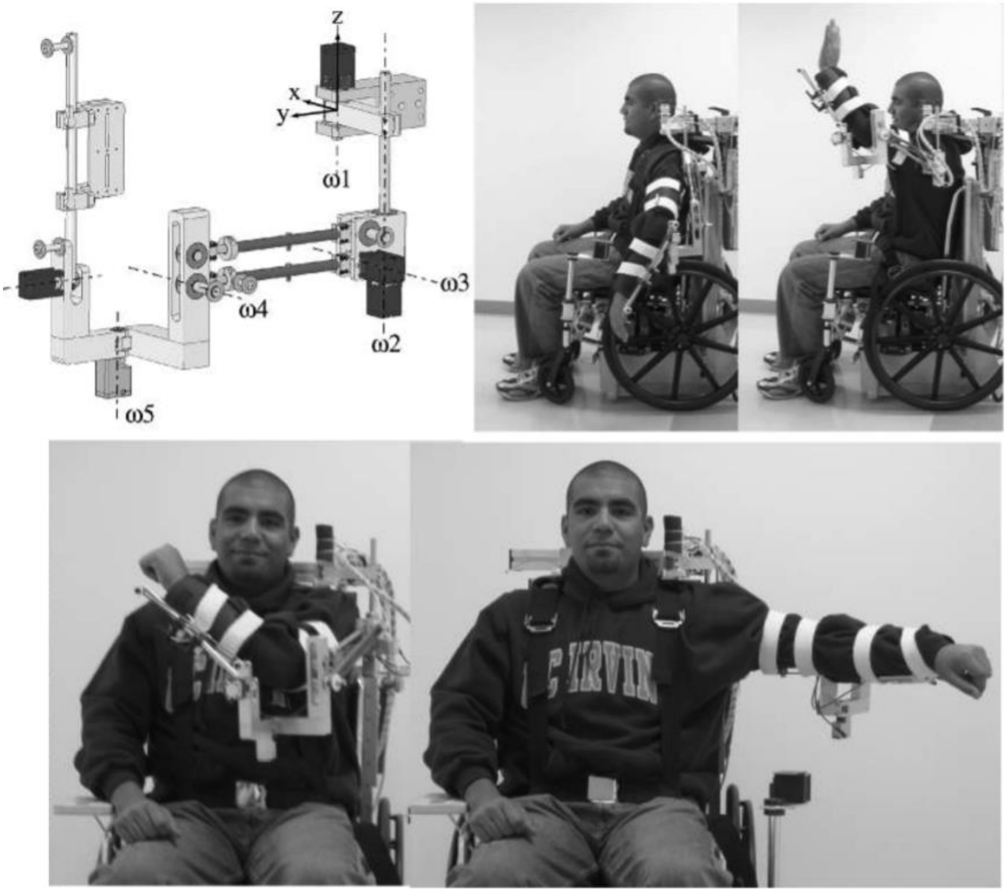

Figure 2.10. T-WREX upper limb rehabilitation robot [49]

2.3.1.4 EMG-Based Assistance

Using EMG signals captured from specified muscles, it is possible to activate assistance when a person exerts effort. The MIT-MANUS robot provides an example of an EMG-triggered assistance in which signals from various muscles in the shoulder and elbow are gathered and the assistance is activated when the processed EMG signals rise beyond a threshold [50].

2.3.1.5 Performance-Based Assistance

Control strategies reviewed above are based on fixed control parameters. It is advantageous to adapt control parameters during the therapy process based on individual patients' needs [50]. Patient-cooperative training procedures, designed originally for the Lokomat, make extensive use of "adaptive control parameters," which allow the robot to take into consideration the patient's intentions rather than imposing a rigid control approach [51]. For MIT-MANUS, it's a critical component of the "performance-based, progressive robot-assisted treatment" control method [50]. There have been several adaptive strategies of this type:

$$P_{i+1} = fP_i - ge_i \tag{1}$$

where,

P_i: control parameter (ex. gain of the robot assistance)

e_i: performance, error measure (ex. ability to reach a target)

f: forgetting factor

g: gain factor

For MIT-MANUS, a performance-based, position-feedback assistive controller was developed that gave participants the option of moving faster than the planned trajectory. The length of the targeted trajectory and the stiffness of the robot controller were adjusted to make the reaching task easier for participants with greater impairments.

The introduction of a forgetting term in this kind of error-based adaptive controller addresses the possibility of participant laziness in response to help. Without forgetting ($f = 1$), if the performance error is zero, the method maintains the control parameter constant and does not push the participant further. If, on the other hand, the forgetting factor is set to 0, the error-based learning algorithm decreases the control parameter when performance error is modest, effectively always challenging the participant. Recently, adaptive controllers with forgetting factors have been developed [52] to gradually lower the feedforward assistive force for reaching when tracking mistakes are modest. It's worth noting that the human motor system, when adapting to new dynamic situations, seems to include such a forgetting element into an error-based learning rule in order to limit its own effort.

Within the scope of patient cooperation, an impedance-based assistive controller was developed for Lokomat. When the patient's effort decreases below a certain threshold, mechanical impedance is increased. A similar approach was developed for controlling an ankle-foot orthosis used for the recovery of the foot gait [53].

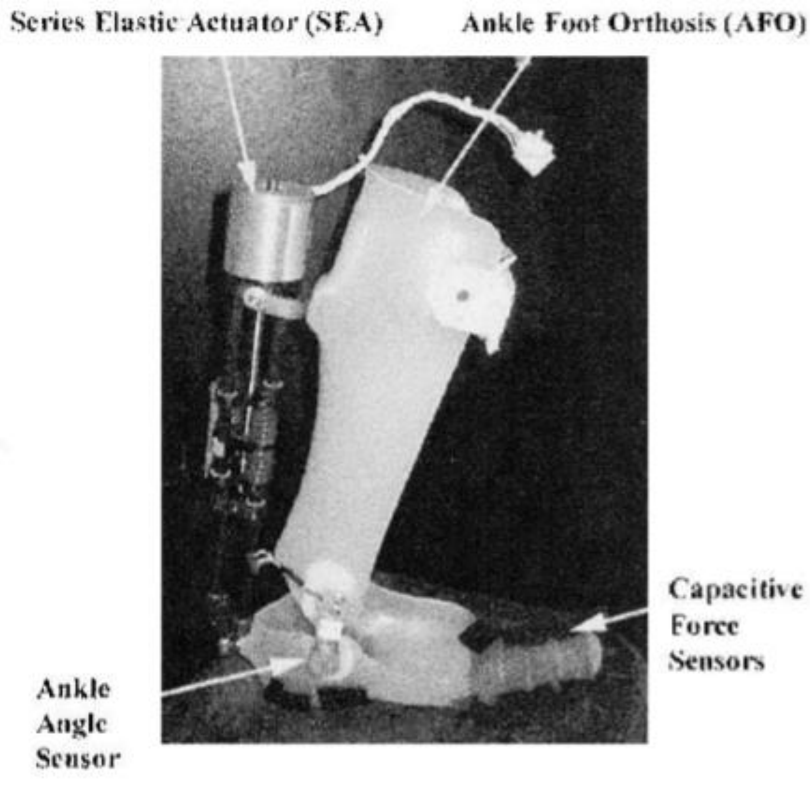

Figure 2.11. Adaptive controlled ankle foot orthosis [53]

Rehabilitation is a complex process that contains many different aspects. The overall aim is to help a patient to recover motor skills lost due to stroke. Up to now, it is known that robotic rehabilitation is superior to conventional rehabilitation methods.

Developments in robotic technologies create an opportunity for the treatment of various stroke patients of different types to accomplish different tasks. With the advancements in the hardware design of robotic devices, the controllers are gaining great importance since the control strategies have great influence over therapy. Mostly, assist as needed approach is modeled with an assistance function changing the impedance controller output [44].

In this thesis an internal reinforcement learning model that represents a patient is presented and two control strategies' performances are compared over the therapy process. One of the main contributions is the investigation of slacking in rehabilitation with a stiff kinematic controller and an admittance controller.

CHAPTER 3

METHODOLOGY

In the following sections human index and thumb finger joint dynamics and a hand rehabilitation robot will be presented so that they can be simulated and combined in a simulation environment to model a human hand with spasticity to promote motor recovery.

3.1 Index and Thumb Finger Passive Torque Dynamics

To design a hand rehabilitation robot, it is crucial to model the finger joints. In Figure 3.1 the simplified model of an index finger is given. It is modeled that three joints and four segments in the extension–flexion plane provides the necessary motion of the index finger [54].

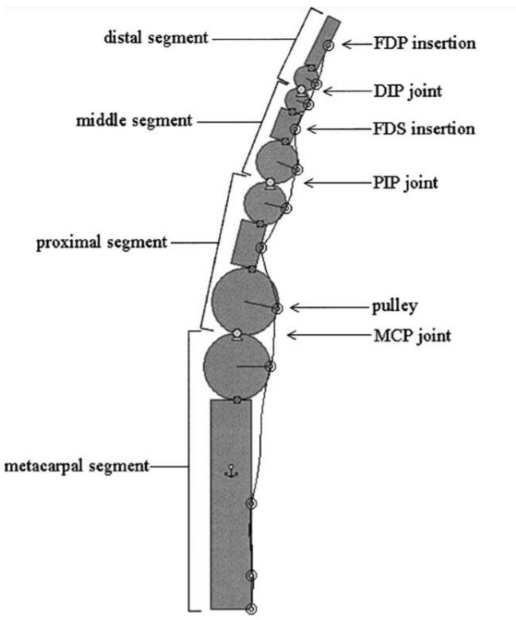

Figure 3.1. Human index finger model [54]

The passive joint torques in the human finger act as resistance to motion [31]. The following equation (2) is a typical model for a healthy human finger. Static and passive torque dynamics are introduced for each joint of the index finger and thumb finger. For joint j, the resultant passive joint torque, τ_j is determined as follows:

$$\tau_j = \tau_j^s - B_j\dot{\theta}_j - K_j(\theta_j)\Delta\theta_j \tag{2}$$

where,

τ_j^s: static passive torque (a function of $(\theta_j - \theta_{j+1})$)

B: damping stiffness coefficient (function of θ_j)

K: dynamic stiffness coefficient (function of θ_j)

Table 3.1 illustrates the physiological static passive joint torque and Table 3.2 illustrates the stiffness and damping at each joint, as described in [54].

Table 3.1 Static passive torque

Joint	τ^s
DIP	$-0.103\theta^3 + 0.102\theta^2 - 0.052\theta - 0.019$
PIP	$0.056\theta^3 + 0.016\theta^2 - 0.132\theta + 0.015$
MCP	$-0.071\theta^3 + 0.14\theta^2 - 0.154\theta + 0.0129$

Table 3.2 Stiffness and damping coefficients

Joint	*K (N.m/rad)*	*B (N.m.s/rad)*
DIP	$0.38\theta^2 - 0.09\theta + 0.13$	81
PIP	$1.06\theta^2 - 0.76\theta + 0.4$	105
MCP	$1.02\theta^2 - 0.54\theta + 0.45$	142

The passive joint torques are introduced in the mathematical model of the thumb and index fingers with exoskeleton mechanisms. Multibody dynamical model of the integrated finger-exoskeleton system in built in SimMultibody environment and the control systems are implemented in Simulink.

3.2 Exopinch Exoskeleton Mechanism

The model of the Exopinch exoskeleton system [2] as shown in Figure 3.2 – Figure 3. 5 is used to study the performances of control systems in terms of motor learning. For the index finger, a fully actuated 2 DOF mechanism is used. The mechanism consists of four 4-bar mechanisms. Note that, the index finger has 3 DOF in the extension–flexion plane but the DOF of the mechanism is 2. For simplicity, mechanism is designed in a way that 2 actuators (inputs) drive MCP and PIP joints directly and DIP joint is driven indirectly. A similar 1 DOF mechanism is used for the thumb finger.

A PIP joint-driven system is used to control thumb finger. The underactuated 1 DOF mechanism is constructed from two 4-bar loops in such a manner that the actuator directly drives the PIP joint and drive the DIP joint indirectly through the MCP and PIP joints. The thumb finger has 2 DOF in the extension-flexion plane. Under-actuation is avoided by modeling the MCP joint's motion as it was coming from the patient so that one of the torque outputs of the reinforcement learning block drives the MCP joint of the thumb finger. The reduced degrees of freedom gained simplify both the mechanism and the controller. Controlling this under actuation will need more research.

Two actuators of the index finger are placed at the M1 and M2 shown in Figure 3.3 at A_0 and H_0. For the thumb finger, one actuator is placed at the M3 as shown in Figure 3.5. at A_{10}. Since the thumb finger is driven by 1 actuator, it is an under actuated mechanism.

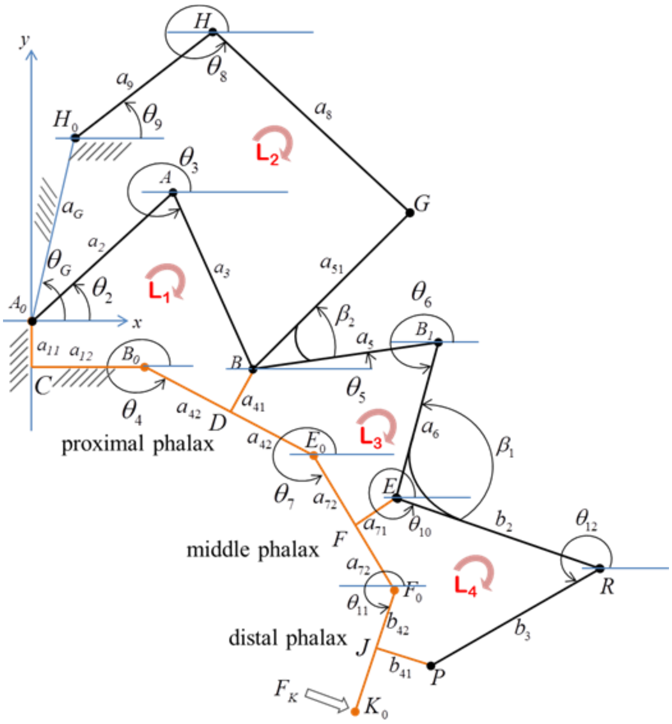

Figure 3.2. Topology of the index finger mechanism

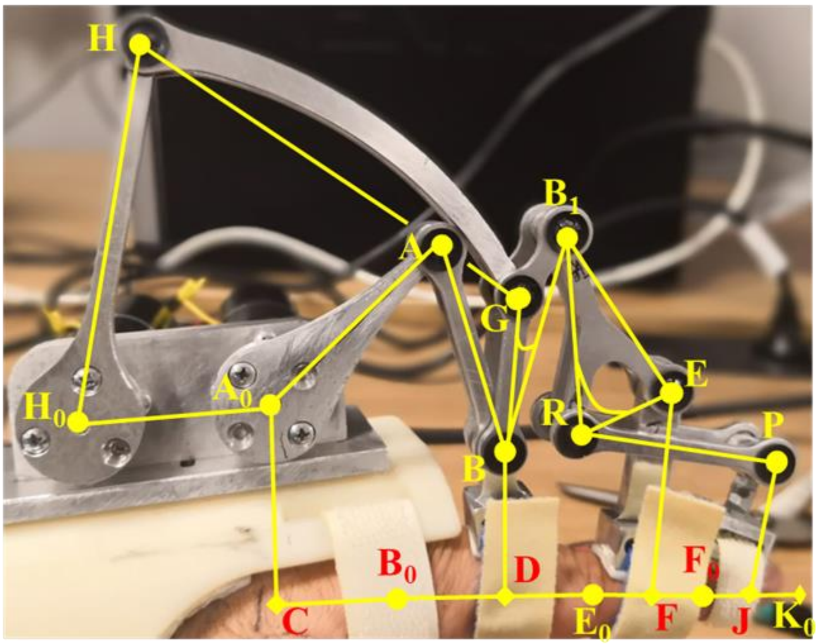

Figure 3.3. Index finger mechanism manufactured and assembled

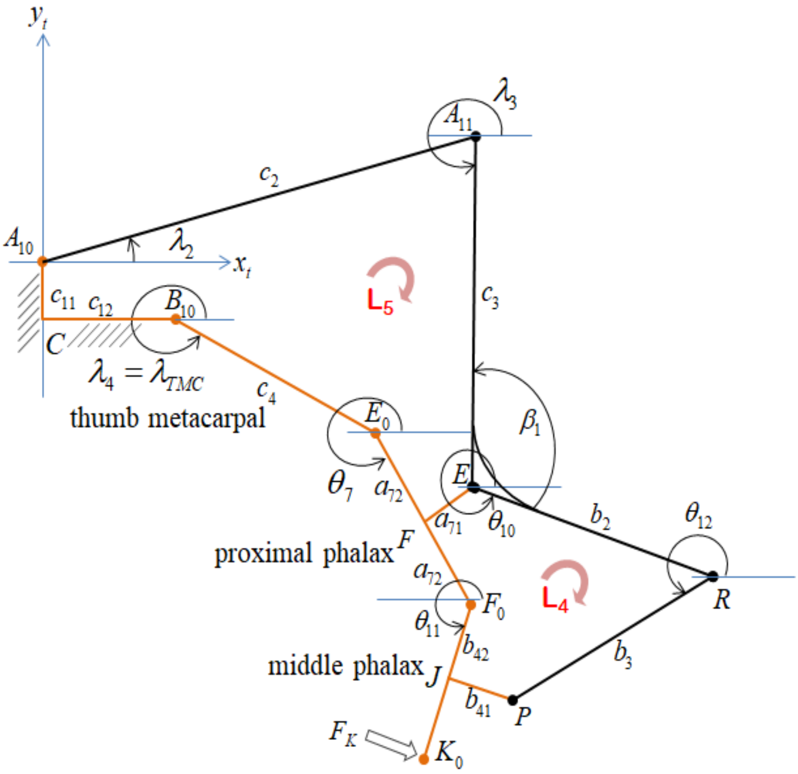

Figure 3.4. Topology of the thumb finger mechanism

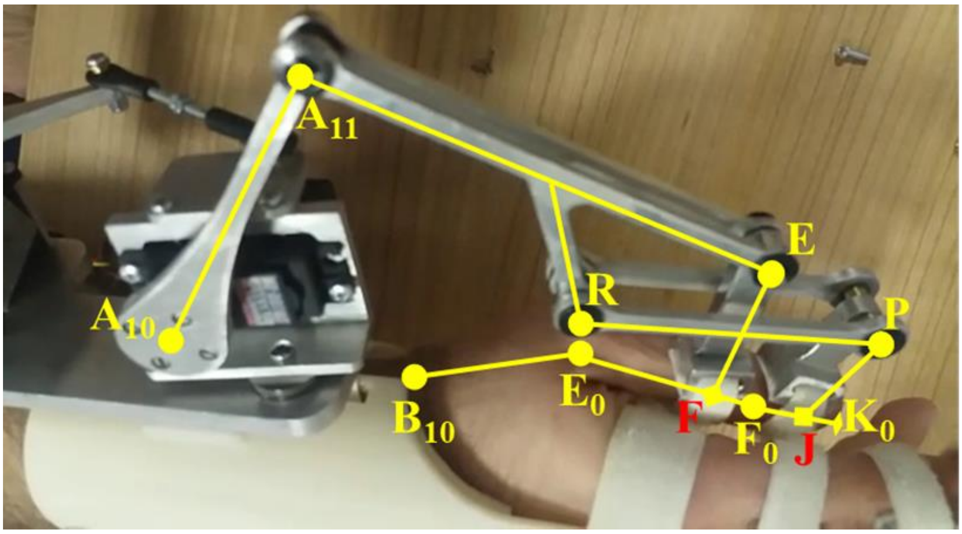

Figure 3.5. Thumb finger mechanism manufactured and assembled

3.3 Simulation Modeling

Exopinch exoskeleton mechanism is designed in Solidworks and exported using Simscape Multibody Link. System is modelled in Simulink/MATLAB and Simscape Multibody Toolbox is used. In Figure 3.6 Simscape animation of the model is shown. Passive torques in the joints of index and thumb fingers are applied in the simulations. Each finger joint and revolute joints in the mechanism is modeled as revolute joint in the Simscape model.

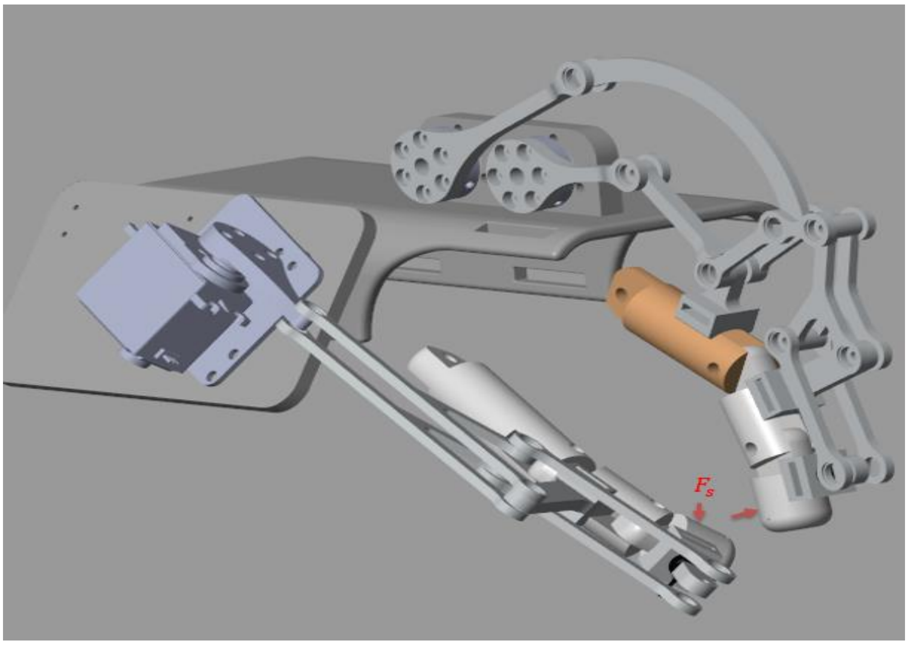

Figure 3.6. Simulink animation of the Exopinch mechanism

Spasticity is modeled by increasing the spring and damping coefficients of the passive torques. System is modelled as if fingers were pinching a spring-like elastic object during rehabilitation. It is a typical task in a therapy session. In Figure 3.7 the tip force applied to index and thumb finger due to the pinching of the elastic object is given. By altering the stiffness coefficient of the spring, tip forces are changed.

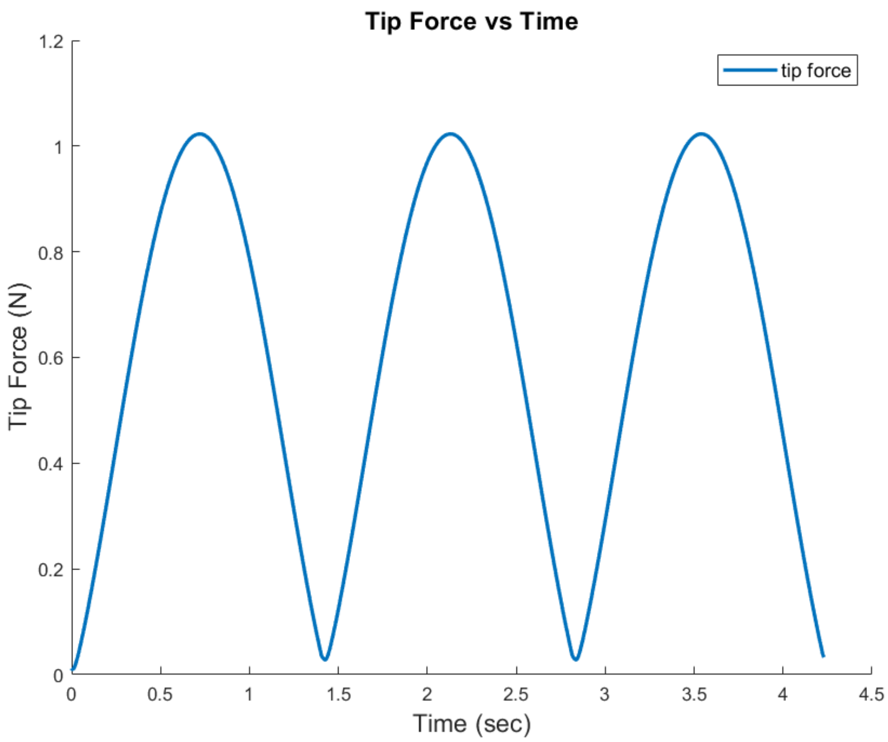

Figure 3.7. Pinching force applied to the finger tip

3.4 Controller Design

To control the mechanism in the extension flexion plane for both index and thumb fingers, two control strategies are used. First the kinematic control is implemented as a PID control structure given in Figure 3.8 and Figure 3.9. Then, admittance control structure given in Figure 3.10 and Figure 3.11 is used. Control structures are implemented separately for each finger.

R_{θ_2}, R_{θ_9} and R_{λ_2} are the reference angles for motor 1 (M1), motor 2 (M2) and motor 3 (M3) for the kinematic controller. T_1, T_2 and T_3 are applied to the exoskeleton mechanism by M1, M2 and M3 respectively. The references are calculated by using the inverse kinematic model to simulate the pinching action of various types.

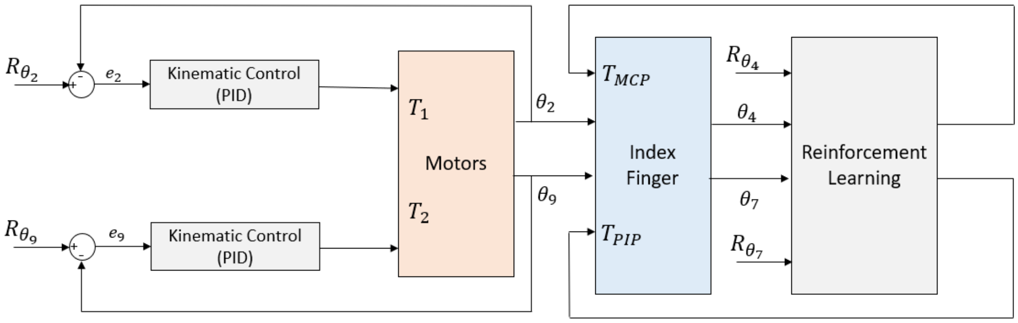

Figure 3.8. Architecture of PID controller for index finger

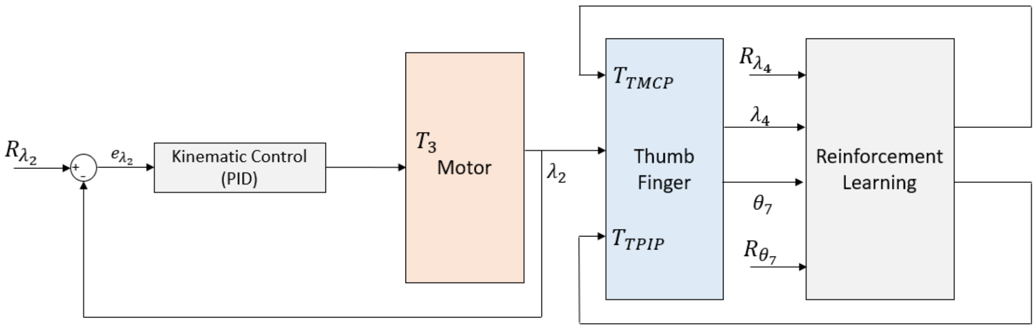

Figure 3.9. Architecture of PID controller for thumb finger

R_{θ_4}, and R_{θ_7} are the reference inputs calculated by inverse kinematics for index finger MCP and PIP joints respectively. Similarly, R_{λ_4}, and R_{θ_7} are the reference inputs for the thumb MCP (TMCP) and PIP (TPIP) joints respectively as shown in Figure 3.9. Motors are assumed to be ideal in the simulations. Reference inputs are calculated by trial and error so that a pinching action is modeled. In addition, references from [2] are used to verify the reference inputs.

Neural networks are trained by implementing the Reinforcement Learning Toolbox provided by Simulink/MATLAB [55] to model the motor learning and motor control processes. Parameter settings for the reinforcement learning is case dependent and determined by trial and error. Reward function for the reinforcement learning has significant effect over the system performance and learning. The following reward function, *R* is used for the system to model positive and negative reinforcement (i.e., the penalty) to the neural network thus to the patient. r_1, r_2 and r_3 are determined experimentally by trial and error and modeled such that they can be changed throughout the recovery process; however, in this study they are kept constant. In human motor re-learning, kinematic and kinetic features of functional recovery at the tip point and joints do not evolve simultaneously. The time varying $r_i's$ will be used to make more realistic simulations for motor learning in future research.

$$R = -r_1\theta_{je} - r_2X_{tip} + r_3F_s - u_a + t \tag{3}$$

where,

θ_{je}: joint kinematic errors for MCP and PIP joints

X_{tip}: tip point error for index and thumb fingers

F_s: tip force

u_a: reinforcement learning torque output

t: time reward

Reward function R will provide negative reward for kinematic errors and positive reward for tip force. As spring stiffness increases (tip force increases) higher reward will be received.

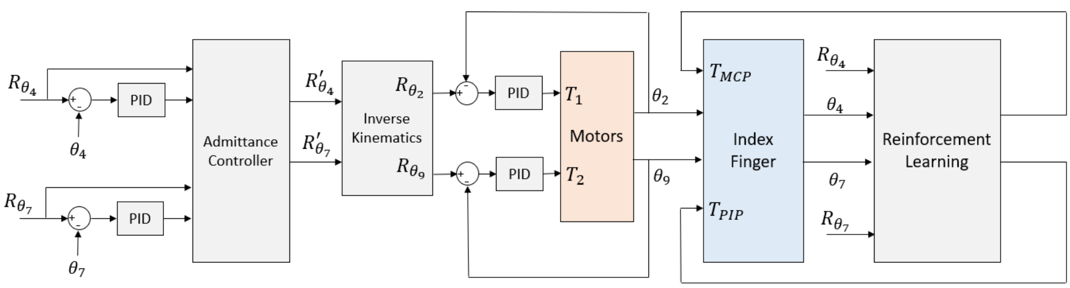

Figure 3.10. Architecture of admittance controller for index finger

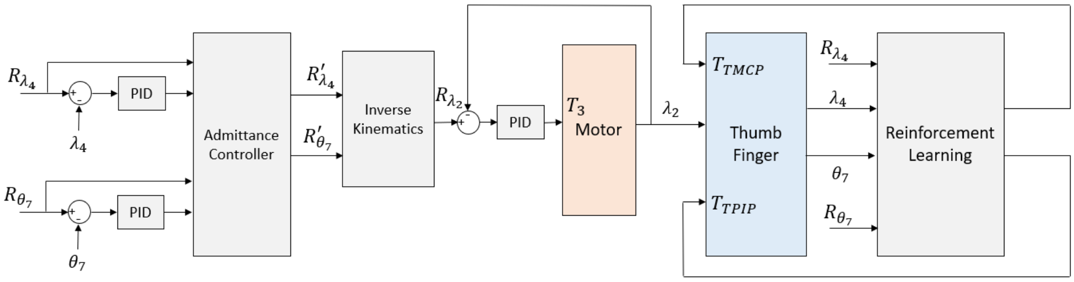

Figure 3.11. Architecture of admittance controller for thumb finger

In admittance control structure, the virtual mass, spring and damping parameters are selected by trial and error. However, spring constant, k_v effects the system performance the most. The control gets more stiff as k_v increases thus allowing less kinematic error.

3.5 Modeling of the Rehabilitation Protocol

The patient is assumed to have a complete functional loss. Therefore, the neural network representing the motor control network to be trained is not able to perform any motor output to achieve the reference pinching motion.

The rehabilitation model is divided into two parts. First, the PID and the admittance controllers are compared, then, the admittance controller's performance is investigated by changing the spring constant, k_v. A pinching task namely, periodic pinching is selected as reference action for the patient to learn. The task is determined from the finger joint angels given in Figure 3.12 and defined as motor reference inputs given in Figure 3.13, as well.

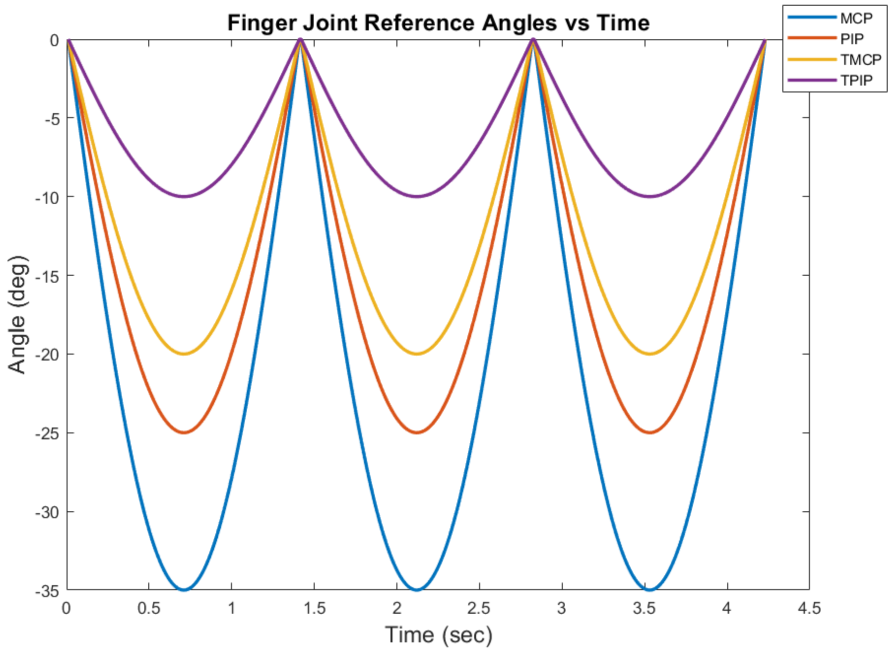

Figure 3.12. Reference finger motion for periodic pinching

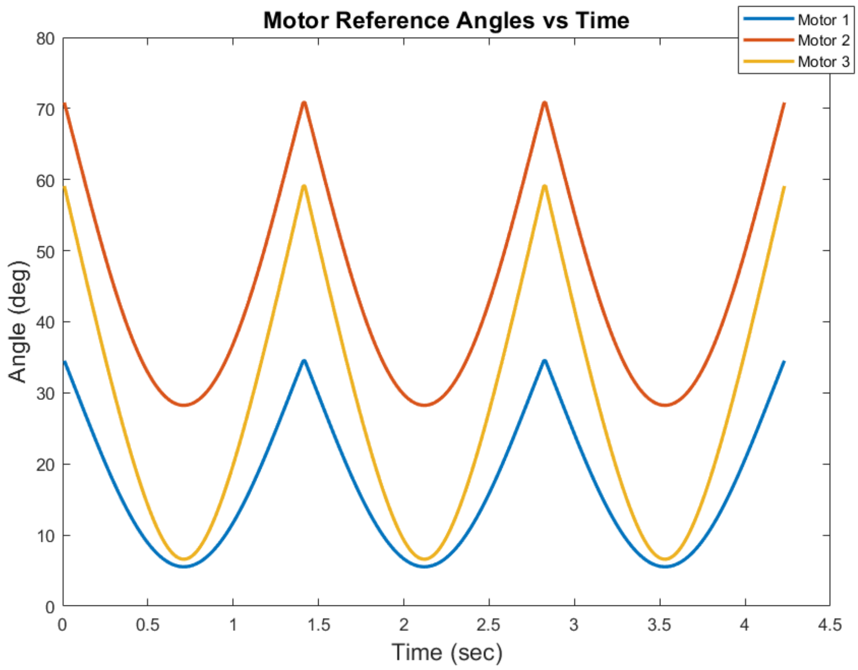

Figure 3.13. Reference motor inputs for periodic pinching

To compare the effect of controllers on recovery and learning performance, assist as needed metric, J [56] is used as follows:

$$J = \sum \lambda_E (\theta_i - \theta_d)^2 + \lambda_R (T_i)^2 \tag{4}$$

where, θ_d is the desired position of the finger and θ_i is the position of the finger at ith step. T_i is the applied motor torque to the system by the motor. λ_E and λ_R are the weight constants to determine the relative effect of torque and error. Note that minimizing the assist as needed metric requires minimizing the kinematic error and applied motor torque. In the simulations, $\lambda_E = 0.1$ and $\lambda_R = 0.5$ was taken.

In addition, to compare the learning performance, reward function is used. Note that, reward function is modeled in a way that the highest reward is obtained with the least kinematic error. Therefore, higher reward results in better learning.

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter, results of the training sessions and rehabilitation modeling is presented. 2 types of simulations are carried out. In Table 4.1 simulation details for 2 control strategies are presented. In the next section, results of the simulation details provided in Table 4.2 are discussed.

Table 4.1 Simulation details of 2 types of control strategies

Simulation Number	*Controller*	k_v
1	PID	-
2	Admittance	1.2

4.1 Performances of PID and Admittance Controllers on Rehabilitation

The task, i.e., periodic circular pinching motion, is defined for both kinematic control system and interaction control system. The patient is rehabilitated in simulations 1 and 2 with the help of PID and Admittance controllers, respectively. The reinforcement learning is simulating the motor learning process of the patient's brain due to the applied robotic therapy. In a real implementation, designed video stimuli are observed by the patient and he/she is asked to perform the same pinching motion observed in the video stream. Kinematic and kinetic references of the observed action are known and set as reference inputs for the robotic system. In Figure 4.1. the reference angle values of the index and thumb finger joints are presented for periodic pinching.

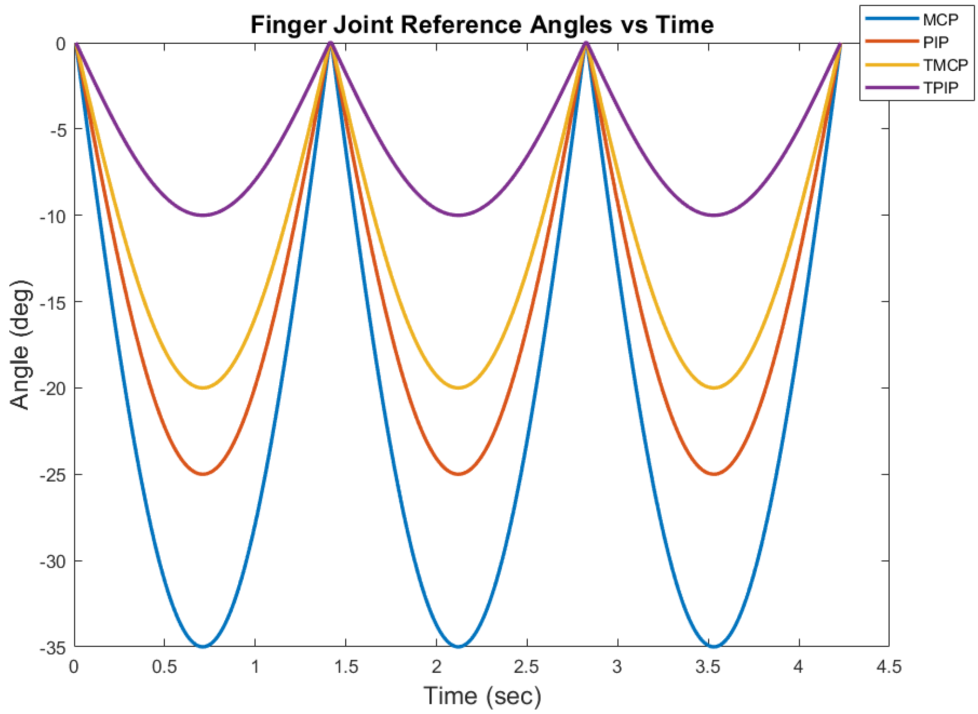

Figure 4.1. Reference finger joint angles for periodic pinching

The overall system in Figure 4.2 is designed and implemented in [2], [28]. The cognitive architecture together with the interaction type of control system will be evaluated and implemented in the project with the grant number TÜBİTAK 121E107.

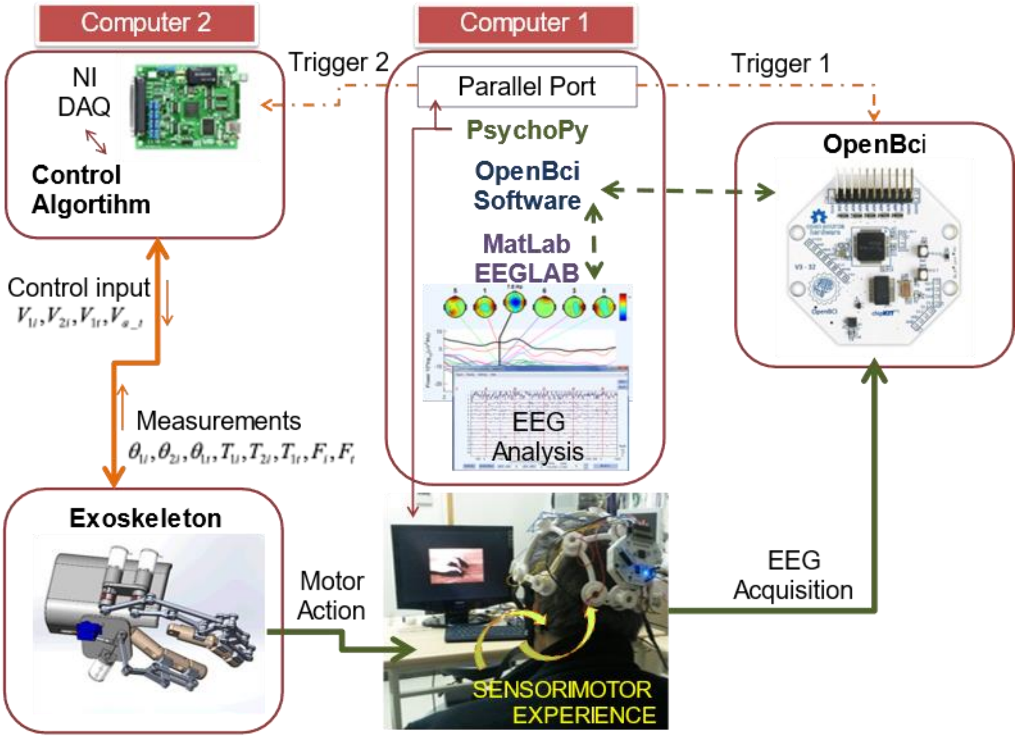

Figure 4.2. Real implementation scheme of Exopinch

As the training process goes on, reinforcement learning agents are compared by obtaining the reward values from training results. The training process starts with maximum exploration and optimal neural structure for the designed rehabilitation protocol is tried to be found. After 500 to 1000 training episodes, depending on the rehabilitation protocol, reward value converges to a maximum point indicating that the patient is moved forward in rehabilitation. The reinforcement learning agents that reach the average of the converged reward value are selected. These agents of reinforcement learning are modeled as treated patients. After the training procedure, trained agents are saved and used as patient models for controller performance comparison and further training. Retraining of an agent as continuation of the rehabilitation protocol is beyond the scope of this thesis; therefore, will not be investigated.

Note that, k_v is selected as 1.2 to make the controller relatively stiff compared to lower values. As k_v increases, controller becomes stiffer by allowing less error.

In simulations 1 and 2, although the reward is increased, meaning the pinching action is done with more accuracy, slacking must be checked to see whether the controller or the patient performs the action. In addition to the evaluation of the cost function, *J*, simulation results are also compared to the references in Figure 4.1 to check slacking.

In simulations 1 and 2, the rehabilitation protocol is performed, and the following results are obtained. Motor learning guided by the PID, and the admittance controlled robotic system continues until the learning reaches to a maximum. In Figure 4.3, the movement of the index and thumb finger as joints angle output trained with PID controller is given. Only a slight pinching action is observed. Compared to the reference input, it can be concluded that the slacking has occurred. The main reason is that PID controller is dictating the movement to the patient. In other words, the kinematic controller defines the motion generated by the voluntary torque as a disturbance and acts to reject the disturbance which attenuates the error formation, motor variability, and the search for motor learning.

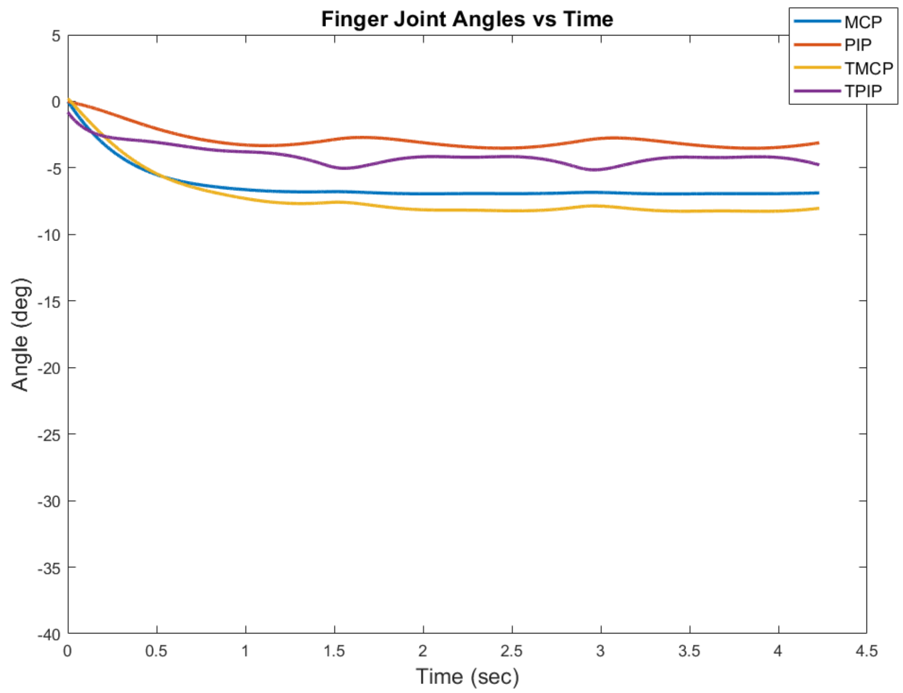

Figure 4.3. Rehabilitation results of the patient trained with PID controller

In Figure 4.4 the movement of the index and thumb finger as joints angle output trained with admittance controller is given. In contrast to the PID controller, both flexion and extension actions are attempted. Slacking is occurred with admittance controller as well; however, compared to the PID trained patient slacking is relatively less. Moreover, comparing performances of PID and admittance trained patients from the reward values given in Figure 4.5, the admittance results in better training than PID since it has obtained higher reward value.

The main reason of the admittance controller's superiority is that the control system of the robot is interacting with the human control system. As a result, kinematic errors appear, and motor learning finds space to explore and exploit as a skill. PID controller is quite accurate while resulting in very little error which results in no learning (slacking).

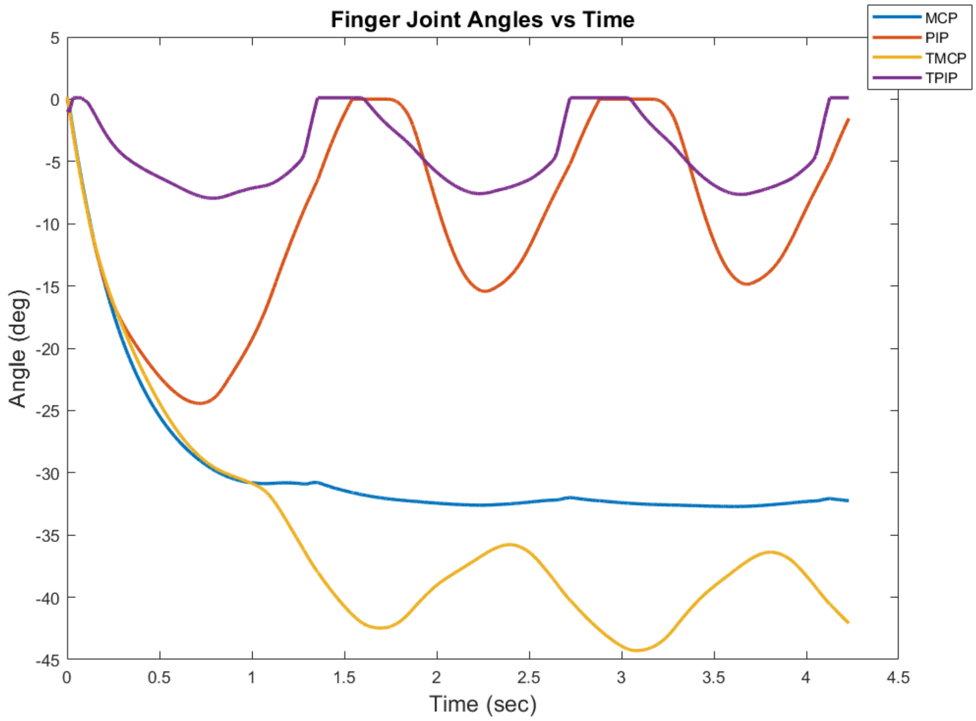

Figure 4.4. Rehabilitation results of the patient trained with admittance controller

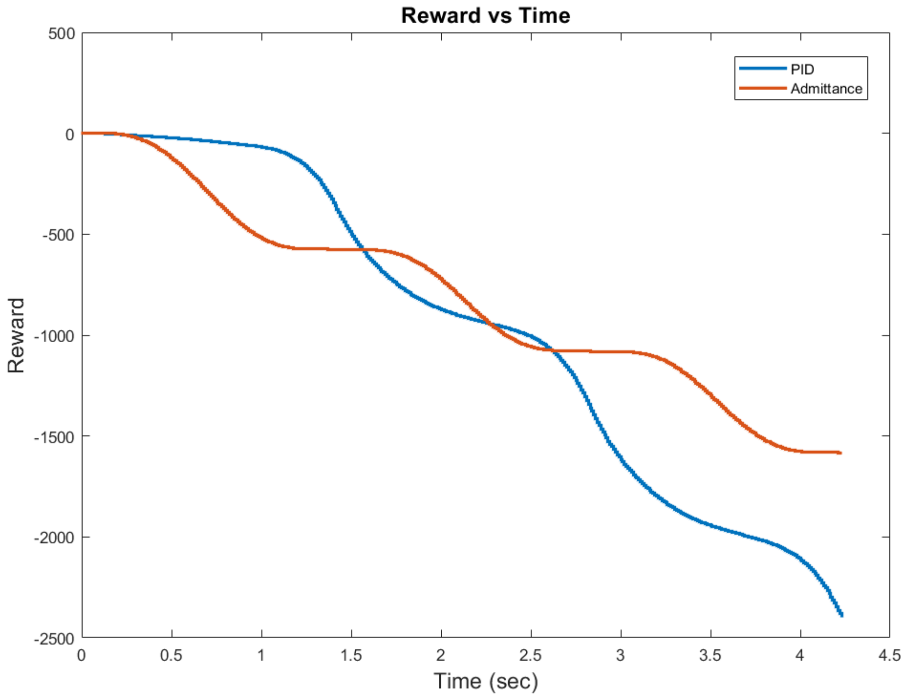

Figure 4.5. Reward values obtained from PID and admittance trained patients

It is observed that learning of the patient with the admittance controller highly depends on the virtual stiffness value k_v and the PID controller without an additional "assist as needed" component would result in slacking. In the following section, effect of k_v on learning is investigated.

4.2 Effect of Virtual Stiffness (k_v) on Rehabilitation

In the previous section, performances of the admittance and the PID controllers were investigated, and it is found out that one of the main factors effecting the admittance controller performance on learning is the virtual spring stiffness, k_v. In this section, the results of the simulations given in Table 4.2 are presented.

Table 4.2 Simulation details of different k_v values

Simulation Number	*Controller*	*Spasticity*	*Tip Force*	k_v
3	Admittance	Medium	Medium	0.1
4	Admittance	Medium	Medium	0.5
5	Admittance	Medium	Medium	0.8
6	Admittance	Medium	Medium	1

For each k_v value ranging from 0.1 to 1, patient is trained with admittance controller. Results of the training is compared with assist as needed metric J, and reward value. In Figure 4.6, reward values obtained from the reinforcement learning is given. As the stiffness decreases, admittance controller allows for more kinematic error so that motor learning is promoted. In other words, motor learning finds space to explore and exploit the rewarded actions to take the motor control. From the simulation results, it is found out that k_v and learning are inversely correlated. In Table 4.3 the results of simulations done with trained agent (patient) are given. As reward increases, learning enhances thus assistance decreases.

Table 4.3 Results of the simulations depending on different k_v values

Simulation Number	*Controller*	k_v	*Reward*	J
3	Admittance	0.1	-481.48	11997
4	Admittance	0.5	-815.44	15345
5	Admittance	0.8	-911.96	18271
6	Admittance	1	-1517.68	20674

In Table 4.4, RMS values of the three motors are given. As learning increases, applied motor torque decreases meaning that patient is learning the kinematics of the movement and can apply more force, thus motor torques decrease.

Table 4.4 RMS values of motor torque inputs depending on different k_v values

Simulation Number	k_v	*Motor 1*	*Motor 2*	*Motor 3*
3	0.1	0.8513	0.9864	0.6656
4	0.5	1.2542	1.1743	0.7373
5	0.8	1.4459	1.4428	0.8368
6	1	1.4942	1.4926	0.8598

In Figure 4.6, the results of assistance applied to the trained patient is given. It is found out that k_v and J are directly proportional. Note that, assist as needed metric is expected to be lower as learning increase. Therefore, results are coherent with the therapy process as well.

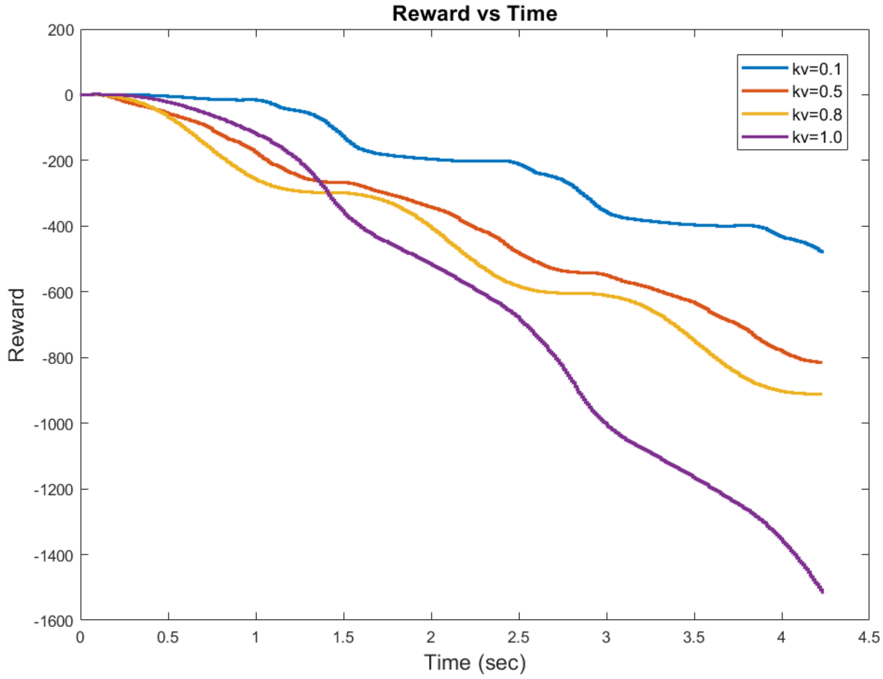

Figure 4.6. Reward values obtained from patients trained with different k_v values

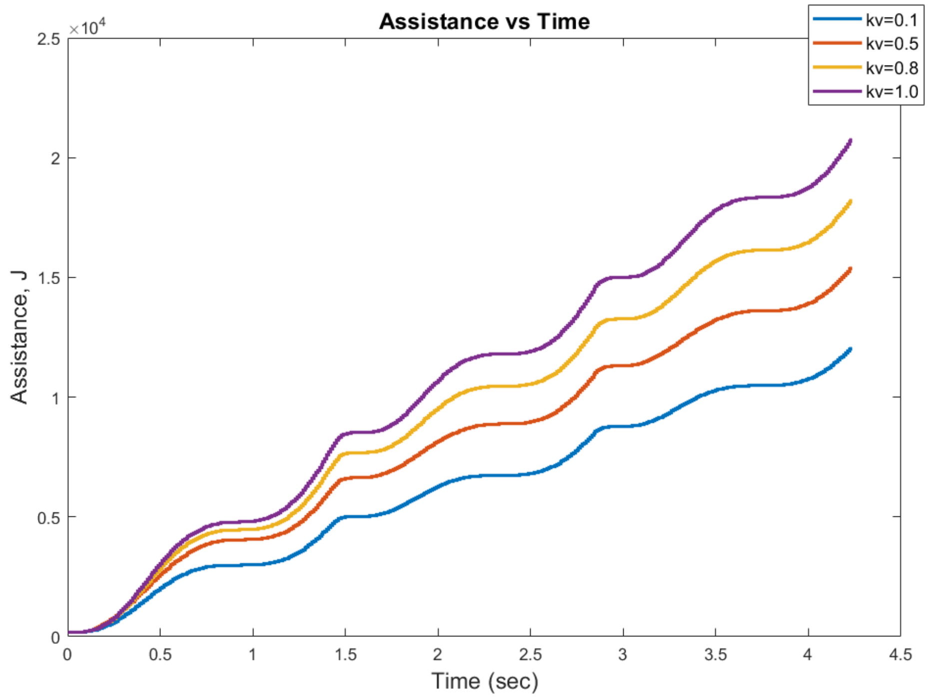

Figure 4.7. Assistance applied the system for different k_v values

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Conclusion

This study focuses on the effect of control strategies over therapy process with a patient model. The performance of the Exopinch robotic mirror therapy system is simulated with kinematic and interaction control systems. One of the key aspects is that the patient is modeled with reinforcement learning algorithm. During the simulations, the following results have been obtained:

- PID controllers without an assist as needed component are not suitable for the complete rehabilitation process since controller dictates the movement to the patient. According to the patient's state, kinematic controllers should be used for a certain period in order not to suppress the motor learning process.
- Robotic control strategy must interact with the human control system and let the human make errors during motion generation. Full assistance resulted in slacking.
- Admittance based control strategies are superior compared to the kinematic control strategies in terms of learning.
- Stiffness of the admittance controller effects the learning performance severely. Stiffness and learning are inversely correlated.
- Under actuation of the thumb finger and full actuation of the index finger should have different approaches. If the mechanism is fully actuated, kinematic controllers result in slacking; however, under actuation may result differently. Investigation of the effect of under actuation is beyond the scope of this thesis.
- As the learning increases, assist as needed metric, *J* decreases as expected. In the simulations $k_v = 0.1$, is the best case for learning and *J* value is obtained

less compared to the other cases. During recovery process, J is expected to decrease and ultimately go to zero while the patient gaining motor functions.

- Neural networks are used to represent the patient. Reinforcement learning is selected as neural network structure since it is a good representative of human motor learning. In the simulations, learning parameters are determined by trial and error since the learning is model based.

5.2 Future Works

As further studies, the internal model for motor learning must be advanced. More realistic internal models can be developed as presented in [57]. Effect of exploration and exploitation should be more distinct for the new models. Developed models should also be tested on real patients. Rehabilitation data taken from patients can be used with system identification tools to determine new model parameters.

In addition, different control strategies and assist as needed approaches will be simulated. Assist as needed added kinematic controllers' performance should be investigated since it is observed that slacking decreases as the controller allows for error.

Effect of reward function's parameters (r_i) will be investigated. How will the learning process have affected while the parameters changed dynamically throughout the rehabilitation process? In the simulations, only the pinching action is simulated.

During the simulation processes, it is observed that under actuation in the thumb finger may result in different learning dynamics than fully actuated index finger. Although control strategies applied to both fingers are the same, controller performances were different. The effect of under actuation of the index finger can be investigated and learning performances will be compared. Learning may occur with full kinematic control strategies if the under actuation is added.

REFERENCES

[1] D. J. Reinkensmeyer, O. M. Akoner, D. P. Ferris, and K. E. Gordon, “Slacking by the human motor system: Computational models and implications for robotic orthoses,” in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Sep. 2009, pp. 2129–2132. doi: 10.1109/IEMBS.2009.5333978.

[2] Mohammad Hassan GOL MOHAMMADZADEH, “OPTIMAL DESIGN OF THE ROBOTIC EXOSKELETON FOR HAND REHABILITATION,” Ankara, 2019.

[3] S. A. Randolph, “Ischemic Stroke,” *Workplace Health & Safety*, vol. 64, no. 9, pp. 444–444, Sep. 2016, doi: 10.1177/2165079916665400.

[4] D. Mozaffarian *et al.*, “Heart Disease and Stroke Statistics—2015 Update,” *Circulation*, vol. 131, no. 4, Jan. 2015, doi: 10.1161/CIR.0000000000000152.

[5] M. O. Owolabi and B. Ovbiagele, “Global clinical neuroscience: Journey of a thousand miles begins with a single step,” *Journal of the Neurological Sciences*, vol. 344, no. 1–2, p. 223, Sep. 2014, doi: 10.1016/j.jns.2014.06.011.

[6] E. L. Miller *et al.*, “Comprehensive Overview of Nursing and Interdisciplinary Rehabilitation Care of the Stroke Patient,” *Stroke,* vol. 41, no. 10, pp. 2402–2448, Oct. 2010, doi: 10.1161/STR.0b013e3181e7512b.

[7] P. M. Pedersen, A. Wandel, H. S. Jørgensen, H. Nakayama, H. O. Raaschou, and T. S. Olsen, “Ipsilateral pushing in stroke: Incidence, relation to neuropsychological symptoms, and impact on rehabilitation. The Copenhagen stroke study,” *Archives of Physical Medicine and*

Rehabilitation, vol. 77, no. 1, pp. 25–28, Jan. 1996, doi: 10.1016/S0003-9993(96)90215-4.

[8] A. W. Dromerick, C. E. Lang, R. Birkenmeier, M. G. Hahn, S. A. Sahrmann, and D. F. Edwards, “Relationships between upper-limb functional limitation and self-reported disability 3 months after stroke,” *The Journal of Rehabilitation Research and Development*, vol. 43, no. 3, p. 401, 2006, doi: 10.1682/JRRD.2005.04.0075.

[9] D. S. Nichols-Larsen, P. C. Clark, A. Zeringue, A. Greenspan, and S. Blanton, “Factors Influencing Stroke Survivors’ Quality of Life During Subacute Recovery,” *Stroke*, vol. 36, no. 7, pp. 1480–1484, Jul. 2005, doi: 10.1161/01.STR.0000170706.13595.4f.

[10] N. S. K. Ho *et al.*, “An EMG-driven exoskeleton hand robotic training device on chronic stroke subjects: Task training system for stroke rehabilitation,” in *2011 IEEE International Conference on Rehabilitation Robotics*, Jun. 2011, pp. 1–5. doi: 10.1109/ICORR.2011.5975340.

[11] A. Gustus, G. Stillfried, J. Visser, H. Jörntell, and P. van der Smagt, “Human hand modelling: kinematics, dynamics, applications,” *Biological Cybernetics*, vol. 106, no. 11–12, pp. 741–755, Dec. 2012, doi: 10.1007/s00422-012-0532-4.

[12] M. Santello, M. Flanders, and J. F. Soechting, “Postural Hand Synergies for Tool Use,” *The Journal of Neuroscience*, vol. 18, no. 23, pp. 10105–10115, Dec. 1998, doi: 10.1523/JNEUROSCI.18-23-10105.1998.

[13] M. Schieber, “Muscular production of individuated finger movements: the roles of extrinsic finger muscles,” *The Journal of Neuroscience*, vol. 15, no. 1, pp. 284–297, Jan. 1995, doi: 10.1523/JNEUROSCI.15-01-00284.1995.

[14] A. Basteris, S. M. Nijenhuis, A. H. Stienen, J. H. Buurke, G. B. Prange, and F. Amirabdollahian, “Training modalities in robot-mediated upper limb rehabilitation in stroke: a framework for classification based on a systematic

review," *Journal of NeuroEngineering and Rehabilitation*, vol. 11, no. 1, p. 111, 2014, doi: 10.1186/1743-0003-11-111.

[15] H. Krebs *et al.*, "Rehabilitation robotics: Pilot trial of a spatial extension for MIT-Manus," *Journal of NeuroEngineering and Rehabilitation*, vol. 1, no. 1, p. 5, 2004, doi: 10.1186/1743-0003-1-5.

[16] P. S. Lum, C. G. Burgar, and P. C. Shor, "Evidence for improved muscle activation patterns after retraining of reaching movements with the MIME robotic system in subjects with post-stroke hemiparesis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 12, no. 2, pp. 186–194, Jun. 2004, doi: 10.1109/TNSRE.2004.827225.

[17] A. 'Trafton, "Robotic therapy helps stroke patients regain function," *MIT News Office*, Apr. 19, 2010.

[18] J. Iqbal and K. Baizid, "Stroke rehabilitation using exoskeleton-based robotic exercisers: Mini Review Towards autonomous cleaning of photovoltaic modules: Design and realization of a robotic cleaner View project Industrial Robotics Simulation Design Planning and Optimization Platform View project Stroke rehabilitation using exoskeleton-based robotic exercisers: Mini Review," 2015. [Online]. Available: www.biomedres.info

[19] G. Narang, A. Narang, S. Singh, and J. Lempiainen, "Use of unobtrusive human-machine interface for rehabilitation of stroke victims through robot assisted mirror therapy," in *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, Apr. 2013, pp. 1–6. doi: 10.1109/TePRA.2013.6556363.

[20] A. Wege and A. Zimmermann, "Electromyography sensor based control for a hand exoskeleton," in *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2007, pp. 1470–1475. doi: 10.1109/ROBIO.2007.4522381.

[21] C. Fleischer, A. Wege, K. Kondak, and G. Hommel, “Application of EMG signals for controlling exoskeleton robots,” *Biomedizinische Technik/Biomedical Engineering*, vol. 51, no. 5_6, pp. 314–319, Dec. 2006, doi: 10.1515/BMT.2006.063.

[22] M. Barsotti *et al.*, “A full upper limb robotic exoskeleton for reaching and grasping rehabilitation triggered by MI-BCI,” in *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, Aug. 2015, pp. 49–54. doi: 10.1109/ICORR.2015.7281174.

[23] A. Frisoli *et al.*, “A New Gaze-BCI-Driven Control of an Upper Limb Exoskeleton for Rehabilitation in Real-World Tasks,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1169–1179, Nov. 2012, doi: 10.1109/TSMCC.2012.2226444.

[24] B. R. Brewer, S. K. McDowell, and L. C. Worthen-Chaudhari, “Poststroke Upper Extremity Rehabilitation: A Review of Robotic Systems and Clinical Results,” *Topics in Stroke Rehabilitation*, vol. 14, no. 6, pp. 22–44, Dec. 2007, doi: 10.1310/tsr1406-22.

[25] M. Lotze and U. Halsband, “Motor imagery,” *Journal of Physiology-Paris*, vol. 99, no. 4–6, pp. 386–395, Jun. 2006, doi: 10.1016/j.jphysparis.2006.03.012.

[26] G. Rizzolatti and L. Craighero, “THE MIRROR-NEURON SYSTEM,” *Annual Review of Neuroscience*, vol. 27, no. 1, pp. 169–192, Jul. 2004, doi: 10.1146/annurev.neuro.27.070203.144230.

[27] M. E. Michielsen *et al.*, “Motor Recovery and Cortical Reorganization After Mirror Therapy in Chronic Stroke Patients,” *Neurorehabilitation and Neural Repair*, vol. 25, no. 3, pp. 223–233, Mar. 2011, doi: 10.1177/1545968310385127.

[28] G. ’Bayer, “INVESTIGATION AND ASSESSMENT OF MIRROR NEURON ACTIVITY FOR REHABILITATION PURPOSES,” 2017.

[29] G. di Pellegrino, L. Fadiga, L. Fogassi, V. Gallese, and G. Rizzolatti, “Understanding motor events: a neurophysiological study,” *Experimental Brain Research*, vol. 91, no. 1, pp. 176–180, Oct. 1992, doi: 10.1007/BF00230027.

[30] G. Rizzolatti, L. Fogassi, and V. Gallese, “Neurophysiological mechanisms underlying the understanding and imitation of action,” *Nature Reviews Neuroscience*, vol. 2, no. 9, pp. 661–670, Sep. 2001, doi: 10.1038/35090060.

[31] E. R. ’ ’Kandel, J. H. ’ ’Schwartz, and T. M. ’ ’Jessel, *Principles of Neural Science*, 3rd ed. 1991.

[32] R. A. Bos *et al.*, “A structured overview of trends and technologies used in dynamic hand orthoses,” *Journal of NeuroEngineering and Rehabilitation*, vol. 13, no. 1, p. 62, Dec. 2016, doi: 10.1186/s12984-016-0168-z.

[33] Z. Yue, X. Zhang, and J. Wang, “Hand Rehabilitation Robotics on Poststroke Motor Recovery,” *Behavioural Neurology*, vol. 2017, pp. 1–20, 2017, doi: 10.1155/2017/3908135.

[34] B. R. Brewer, S. K. McDowell, and L. C. Worthen-Chaudhari, “Poststroke Upper Extremity Rehabilitation: A Review of Robotic Systems and Clinical Results,” *Topics in Stroke Rehabilitation*, vol. 14, no. 6, pp. 22–44, Dec. 2007, doi: 10.1310/tsr1406-22.

[35] D. Pinter *et al.*, “Exploratory Study on the Effects of a Robotic Hand Rehabilitation Device on Changes in Grip Strength and Brain Activity after Stroke,” *Topics in Stroke Rehabilitation*, vol. 20, no. 4, pp. 308–316, Jul. 2013, doi: 10.1310/tsr2004-308.

[36] “https://tyromotion.com/en/products/amadeo/.”

[37] L. Dovat *et al.*, “HandCARE: A Cable-Actuated Rehabilitation System to Train Hand Function After Stroke,” *IEEE Transactions on Neural Systems*

and Rehabilitation Engineering, vol. 16, no. 6, pp. 582–591, Dec. 2008, doi: 10.1109/TNSRE.2008.2010347.

[38] R. A. R. C. Gopura, D. S. V. Bandara, K. Kiguchi, and G. K. I. Mann, "Developments in hardware systems of active upper-limb exoskeleton robots: A review," *Robotics and Autonomous Systems*, vol. 75, pp. 203–220, Jan. 2016, doi: 10.1016/j.robot.2015.10.001.

[39] N. S. K. Ho *et al.*, "An EMG-driven exoskeleton hand robotic training device on chronic stroke subjects: Task training system for stroke rehabilitation," in *2011 IEEE International Conference on Rehabilitation Robotics*, Jun. 2011, pp. 1–5. doi: 10.1109/ICORR.2011.5975340.

[40] A. Chiri *et al.*, "HANDEXOS: Towards an exoskeleton device for the rehabilitation of the hand," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2009, pp. 1106–1111. doi: 10.1109/IROS.2009.5354376.

[41] J. Tang, X. Zhang, and Z.-M. Li, "Operational and maximal workspace of the thumb," *Ergonomics*, vol. 51, no. 7, pp. 1109–1118, Jul. 2008, doi: 10.1080/00140130801958667.

[42] L. Marchal-Crespo and D. J. Reinkensmeyer, "Review of control strategies for robotic movement training after neurologic injury," *Journal of NeuroEngineering and Rehabilitation*, vol. 6, no. 1, p. 20, Dec. 2009, doi: 10.1186/1743-0003-6-20.

[43] E. T. Wolbrecht, V. Chan, V. Le, S. C. Cramer, D. J. Reinkensmeyer, and J. E. Bobrow, "Real-time computer modeling of weakness following stroke optimizes robotic assistance for movement therapy," in *2007 3rd International IEEE/EMBS Conference on Neural Engineering*, May 2007, pp. 152–158. doi: 10.1109/CNE.2007.369635.

[44] R. J. Sanchez *et al.*, "A Pneumatic Robot for Re-Training Arm Movement after Stroke: Rationale and Mechanical Design," in *9th International*

Conference on Rehabilitation Robotics, 2005. ICORR 2005., pp. 500–504. doi: 10.1109/ICORR.2005.1501151.

[45] E. T. Wolbrecht, V. Chan, D. J. Reinkensmeyer, and J. E. Bobrow, "Optimizing Compliant, Model-Based Robotic Assistance to Promote Neurorehabilitation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, no. 3, pp. 286–297, Jun. 2008, doi: 10.1109/TNSRE.2008.918389.

[46] M. İ. C. 'Dede, "POSITION/FORCE CONTROL OF ROBOT MANIPULATORS ," 2003.

[47] M. Lotze, "Motor learning elicited by voluntary drive," *Brain*, vol. 126, no. 4, pp. 866–872, Apr. 2003, doi: 10.1093/brain/awg079.

[48] C. D. Takahashi, L. Der-Yeghiaian, V. H. Le, and S. C. Cramer, "A Robotic Device for Hand Motor Therapy after Stroke," in *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, pp. 17–20. doi: 10.1109/ICORR.2005.1501041.

[49] R. J. Sanchez *et al.*, "Automating Arm Movement Training Following Severe Stroke: Functional Exercises With Quantitative Feedback in a Gravity-Reduced Environment," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 3, pp. 378–389, Sep. 2006, doi: 10.1109/TNSRE.2006.881553.

[50] H. I. Krebs *et al.*, "Rehabilitation Robotics: Performance-Based Progressive Robot-Assisted Therapy," 2003.

[51] R. Riener, L. Lunenburger, S. Jezernik, M. Anderschitz, G. Colombo, and V. Dietz, "Patient-cooperative strategies for robot-aided treadmill training: first experimental results," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 3, pp. 380–394, Sep. 2005, doi: 10.1109/TNSRE.2005.848628.

[52] E. T. Wolbrecht, V. Chan, V. Le, S. C. Cramer, D. J. Reinkensmeyer, and J. E. Bobrow, “Real-time computer modeling of weakness following stroke optimizes robotic assistance for movement therapy,” in *2007 3rd International IEEE/EMBS Conference on Neural Engineering*, May 2007, pp. 152–158. doi: 10.1109/CNE.2007.369635.

[53] J. A. Blaya and H. Herr, “Adaptive control of a variable-impedance ankle-foot orthosis to assist drop-foot gait,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 12, no. 1, pp. 24–31, Mar. 2004, doi: 10.1109/TNSRE.2003.823266.

[54] D. G. Kamper, T. George Hornby, and W. Z. Rymer, “Extrinsic flexor muscles generate concurrent flexion of all three finger joints,” *Journal of Biomechanics*, vol. 35, no. 12, pp. 1581–1589, Dec. 2002, doi: 10.1016/S0021-9290(02)00229-4.

[55] Mathworks, “Reinforcement Learning Toolbox: User’s Guide (R2021b).” Mathworks, 2012.

[56] J. L. Emken, J. E. Bobrow, and D. J. Reinkensmeyer, “Robotic Movement Training As an Optimization Problem: Designing a Controller That Assists Only As Needed,” in *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, pp. 307–312. doi: 10.1109/ICORR.2005.1501108.

[57] E. Oztop, M. Kawato, and M. Arbib, “Mirror neurons and imitation: A computationally guided review,” *Neural Networks*, vol. 19, no. 3, pp. 254–271, Apr. 2006, doi: 10.1016/j.neunet.2006.02.002.

APPENDICES

A. Matlab Code for Inverse Kinematics and Reference Calculations

```
order=1;
fel=0;
Ts=0.01; %sample time
Te = 1.4;
t=0:Ts:Te;

N=length(t);


load('th_index_ref_task1.mat')
load('th_thumb_ref_task1.mat')

load('th_index_ref_task2.mat')
load('th_thumb_ref_task2.mat')

load('th_index_ref_task0.mat')
load('th_thumb_ref_task0.mat')

sat=0.05;
fs=0.3; %spring force at the tip
k_obj=0.25;
mv=0.1;
kv=0.3;
cv=0.5;

C=[20 0.5 200]; %angle error, tip force, tip point kin

task=1

if task==1

%Index Motors Input Set

a11=3.825e-2;
a12=4.57e-2;
a2=5e-2;
a3=4.7e-2;
a41=3e-2;
a42=2.5e-2;
a5=5e-2;
a51=3.5e-2;
a5T=a5+a51;
a6=4.5e-2;
a71=5e-2;
a72=1.5e-2;
a8=9.6e-2;
```

```
a9=8e-2;

aG=4e-2;
thG=pi;
b2=2.5e-2;
b3=4.82e-2;
b41=3e-2;
b42=1.5e-2;
beta1=110*pi/180;
beta2=10*pi/180;


task1_mcp=-(35*pi/180)*sin(pi*(t/Te));
task1_pip=-(60*pi/180)*sin(pi*(t/Te));
hold1=zeros(1,50);
th4=[task1_mcp task1_mcp task1_mcp];
th7=[task1_pip task1_pip task1_pip];

Tf=Ts*length(th4);

for i=1:length(th4)
front_A0_in(i,1)=Ts*i;
back_H0_in(i,1)=Ts*i;
th_MCP_in(i,1)=Ts*i;
th_PIP_in(i,1)=Ts*i;

tip_pos(i,1)=Ts*i;
xK0_ref(i,1)=Ts*i;
yK0_ref(i,1)=Ts*i;
end

th_MCP_in(:,2)=th4';
th_PIP_in(:,2)=th7'-th4';


[th2,th3]=V2_L1_inv_kin(th4,a11,a12,a2,a3,a41,a42);
[th5,th6]=V2_L3_inv_kin(th4,th7,a41,a42,a5,a6,a71,a72);
[th8,th9]=V2_L2mod_inv_kin(th2,th3,thG,th5,a2,a3,a51,a8,aG,a9,beta2);

th10=th6+pi-beta1;
[th11,th12]=V2_L4_direct_kin(th7,th10,b2,b3,b41,b42,a71,a72);
th11_d=(5*th7/3)-(2*th4/3);

front_A0_in(:,2)=th2'*180/pi;
back_H0_in(:,2)=th9'*180/pi;

ta11=th3-th2-pi;
ta12=(pi/2)+(th4-th3)-atan(a42/a41);
ta21=th8-th9-pi;
ta22=2*pi+th5+beta2-th8;
ta31=th6-th5-pi;
ta32=(pi/2)+(th7-th6)-atan(a72/a71);
ta41=th12-th10+pi;
```

```
ta42=(pi/2)+(th11-th12)-atan(b42/b41);

F1=[(ta11'-(pi/2));(ta21'-(pi/2));(ta31'-(pi/2));(ta41'-(pi/2))];
F2=[(ta12'-(pi/2));(ta22'-(pi/2));(ta32'-(pi/2));(ta42'-(pi/2))];
F3=[th11'-th11_d'];
F=[F1;F2;F3];

xA0=0*th2; yA0=0*th2;
xB0=a12*ones(1,length(th2)); yB0=-a11*ones(1,length(th2));
xA=a2*cos(th2); yA=a2*sin(th2);
xB=xB0+a42*cos(th4)+a41*cos(th4+(pi/2));
yB=yB0+a42*sin(th4)+a41*sin(th4+(pi/2));
% xB=xA+a3*cos(theta3_1); yB=yA+a3*sin(theta3_1);
xC=0*th2; yC=-a11*ones(1,length(th2));
xD=xB0+a42*cos(th4); yD=yB0+a42*sin(th4);
xE0=xB0+2*a42*cos(th4); yE0=yB0+2*a42*sin(th4);
xB1=xB+a5*cos(th5); yB1=yB+a5*sin(th5);
xE=xB1+a6*cos(th6); yE=yB1+a6*sin(th6);
xF=xE0+a72*cos(th7); yF=yE0+a72*sin(th7);
xF0=xE0+2*a72*cos(th7); yF0=yE0+2*a72*sin(th7);
xJ=xF0+b42*cos(th11); yJ=yF0+b42*sin(th11);
xK0=xF0+2*b42*cos(th11);
yK0=yF0+2*b42*sin(th11);

xK0_ref(:,2)=xK0;
yK0_ref(:,2)=yK0;

tip_pos(:,2)=xK0.^2+yK0.^2;

xR=xE+b2*cos(th10); yR=yE+b2*sin(th10);
xP=xR+b3*cos(th12); yP=yR+b3*sin(th12);

xH0=aG*cos(thG); yH0=aG*sin(thG);
xH=xH0+a9*cos(th9);yH=yH0+a9*sin(th9);
xG=xB+a51*cos(th5+beta2);yG=yB+a51*sin(th5+beta2);

%%%---------------Index END-------------------%%%%



%%%---------------Thumb Finger-----------------%%%%

c4=6e-2;
a72=1.5e-2;
b42=1.25e-2;
c11=5.03e-2;
c12=4.5e-2;
c2=5e-2;
a71=6e-2;
c3=10e-2;
b2=4e-2;
b3=6.8e-2;
b41=3.5e-2;
```

```
beta1=-44*pi/180;


task1_th7=-(30*pi/180)*sin(pi*(t/Te));
task1_lm4=-(20*pi/180)*sin(pi*(t/Te));
th7=[task1_th7 task1_th7 task1_th7];
lm4=[task1_lm4 task1_lm4 task1_lm4];


th11_d=(23*lm4/8)-(15*pi/16);

[lm2,lm3]=V2_L5_inv_kin(lm4,th7,c11,c12,c2,c3,c4,a71,a72);

th10=lm3+pi-beta1;
[th11,th12]=V2_L4_direct_kin(th7,th10,b2,b3,b41,b42,a71,a72);
% th11_d=(5*th7/3)-(2*th4/3);
ta51=lm3-lm2-pi;
 ta52=(pi/2)+(th7-lm3)-atan(a72/a71);
 ta41=th12-th10+pi;
 ta42=(pi/2)+(th11-th12)-atan(b42/b41);


% ---------------------------

F1=[(ta51-(pi/2)).^2;1*((2*pi-ta41)-(pi/2)).^2];
F2=[(ta52-(pi/2)).^2;1*(ta42-(pi/2)).^2];
F3=[(th11(1)-th11_d(1))^2; (th11(end)-th11_d(end))^2]; %(th11(6)-
th11_d(6))^2;
F=0.1*(sum(F1)/(length(F1)))+0.45*(sum(F2)/(length(F2)))+0.45*(sum((F3)
)/(length(F3))); %

% F=[F1;F2;F3];

xA10=0*th7; yA10=0*th7;
xC=xA10; yC=0*th7-c11;
xB10=c12*ones(1,length(th7)); yB10=-c11*ones(1,length(th7));
xA11=c2*cos(lm2); yA11=c2*sin(lm2);
xE0=xB10+c4*cos(lm4); yE0=yB10+c4*sin(lm4);
xF=xE0+a72*cos(th7); yF=yE0+a72*sin(th7);
xE=xA11+c3*cos(lm3); yE=yA11+c3*sin(lm3);
xF0=xE0+2*a72*cos(th7); yF0=yE0+2*a72*sin(th7);
xJ=xF0+b42*cos(th11); yJ=yF0+b42*sin(th11);
xR=xE+b2*cos(th10); yR=yE+b2*sin(th10);
xP=xR+b3*cos(th12); yP=yR+b3*sin(th12);
xK0=xF0+2*b42*cos(th11);yK0=yF0+2*b42*sin(th11);

for i=1:length(th7)
l2(i,1)=Ts*i;
l2(i,2)=lm2(1,i)*180/pi-59.1;

l4(i,1)=Ts*i;
l4(i,2)=lm4(1,i)*180/pi-52.5;
```

```
th_TMCP_in(i,1)=Ts*i;
th_TPIP_in(i,1)=Ts*i;
end

th_TMCP_in(:,2)=lm4';
th_TPIP_in(:,2)=th7'-lm4';


end
```

B. Matlab Code for Reinforcement Learning

```
mdl = 'El_Mekanizmasi_Montaj3';
open_system(mdl)
useFastRestart = true;
useGPU = true;

numAct = 4;
actionInfo = rlNumericSpec([numAct 1],'LowerLimit',-
0.6,'UpperLimit',1);
actionInfo.Name = 'joint_torque';
actionInfo.Description = 'MCP_T, PIP_T,TMCP_T, TPIP_T';
numActions = actionInfo.Dimension(1);

open_system([mdl '/PLANT/Index_Finger_2/generate observations'])

numObs = 37;
observationInfo = rlNumericSpec([numObs 1],...
    'LowerLimit',[-inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -
inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -
inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -inf -inf]',...
    'UpperLimit',[ inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf]');
observationInfo.Name = 'observations';
observationInfo.Description = 'MCP_angle, PIP_angle, MCP_vel, PIP_vel,
xK0, yK0, MCP_in, PIP_in, th_DIP, DIP_vel, xe11, xe22, xe33, xe44,
xe55, ye11, ye22, ye33, ye44, ye55, Fs, TMCP_in, TPIP_in, th_TMCP,
TMCP_vel,th_TPIP, TPIP_vel, txe11, txe22, txe33, txe44, txe55, tye11,
tye22, tye33, tye44, tye55';
numObservations = observationInfo.Dimension(1);

open_system([mdl '/PLANT/Index_Finger_2/calculate reward'])

open_system([mdl '/PLANT/Index_Finger_2/stop simulation'])

env = rlSimulinkEnv(mdl,[mdl '/PLANT/Index_Finger_2/RL
Agent'],observationInfo,actionInfo);


rng(0)

criticLayerSizes = [400 300];
statePath = [
    imageInputLayer([numObs 1 1],'Normalization','none','Name','State')
    fullyConnectedLayer(criticLayerSizes(1), 'Name', 'CriticStateFC1',
...
            'Weights',2/sqrt(numObs)*(rand(criticLayerSizes(1),numObs)-
0.5), ...
            'Bias',2/sqrt(numObs)*(rand(criticLayerSizes(1),1)-0.5))
    reluLayer('Name','CriticStateRelu1')
```

```
    fullyConnectedLayer(criticLayerSizes(2), 'Name', 'CriticStateFC2', ...
'Weights',2/sqrt(criticLayerSizes(1))*(rand(criticLayerSizes(2),criticLayerSizes(1))-0.5), ...
'Bias',2/sqrt(criticLayerSizes(1))*(rand(criticLayerSizes(2),1)-0.5))
    ];

actionPath = [
    imageInputLayer([numAct 1 1],'Normalization','none', 'Name','Action')
    fullyConnectedLayer(criticLayerSizes(2), 'Name', 'CriticActionFC1', ...
            'Weights',2/sqrt(numAct)*(rand(criticLayerSizes(2),numAct)-0.5), ...
            'Bias',2/sqrt(numAct)*(rand(criticLayerSizes(2),1)-0.5))
    ];
commonPath = [
    additionLayer(2,'Name','add')
    reluLayer('Name','CriticCommonRelu1')
    fullyConnectedLayer(1, 'Name', 'CriticOutput',...
            'Weights',2*5e-3*(rand(1,criticLayerSizes(2))-0.5), ...
            'Bias',2*5e-3*(rand(1,1)-0.5))
    ];


criticNetwork = layerGraph();
criticNetwork = addLayers(criticNetwork,statePath);
criticNetwork = addLayers(criticNetwork,actionPath);
criticNetwork = addLayers(criticNetwork,commonPath);
criticNetwork = connectLayers(criticNetwork,'CriticStateFC2','add/in1');
criticNetwork = connectLayers(criticNetwork,'CriticActionFC1','add/in2');


criticOpts = rlRepresentationOptions('LearnRate',1e-3,'GradientThreshold',1);

critic = rlQValueRepresentation(criticNetwork,observationInfo,actionInfo,'Observation',{'State'},'Action',{'Action'},criticOpts);



actorLayerSizes = [400 300];
actorNetwork = [
    imageInputLayer([numObs 1 1],'Normalization','none','Name','State')

    fullyConnectedLayer(actorLayerSizes(1), 'Name', 'ActorFC1', ...
            'Weights',2/sqrt(numObs)*(rand(actorLayerSizes(1),numObs)-0.5), ...
```

```
            'Bias',2/sqrt(numObs)*(rand(actorLayerSizes(1),1)-0.5))
    reluLayer('Name', 'ActorRelu1')
    fullyConnectedLayer(actorLayerSizes(2), 'Name', 'ActorFC2', ...

'Weights',2/sqrt(actorLayerSizes(1))*(rand(actorLayerSizes(2),actorLaye
rSizes(1))-0.5), ...

'Bias',2/sqrt(actorLayerSizes(1))*(rand(actorLayerSizes(2),1)-0.5))
    reluLayer('Name', 'ActorRelu2')
    fullyConnectedLayer(numAct, 'Name', 'ActorFC3', ...
            'Weights',2*5e-3*(rand(numAct,actorLayerSizes(2))-0.5), ...
            'Bias',2*5e-3*(rand(numAct,1)-0.5))
    tanhLayer('Name','ActorTanh1')
    ];


actorOptions = rlRepresentationOptions('LearnRate',1e-
02,'GradientThreshold',1);

actor =
rlDeterministicActorRepresentation(actorNetwork,observationInfo,actionI
nfo,'Observation',{'State'},'Action',{'ActorTanh1'},actorOptions);

agentOpts = rlDDPGAgentOptions(...
    'SampleTime',Ts,...
    'TargetSmoothFactor',1e-3,...
    'DiscountFactor',0.99, ...
    'MiniBatchSize',128, ...
    'ExperienceBufferLength',1e6, ...
    'SaveExperienceBufferWithAgent',true);
agentOpts.NoiseOptions.Variance = 0.5;
agentOpts.NoiseOptions.VarianceDecayRate = 0;
agentOpts.NoiseOptions.MeanAttractionConstant = 2;

agent = rlDDPGAgent(actor,critic,agentOpts);

maxepisodes = 5000;
maxsteps = ceil(Tf/Ts);
trainOpts = rlTrainingOptions(...
    'MaxEpisodes',maxepisodes, ...
    'MaxStepsPerEpisode',maxsteps, ...
    'ScoreAveragingWindowLength',20, ...
    'Verbose',false, ...
    'Plots','training-progress',...
    'StopTrainingCriteria','EpisodeCount',...
    'StopTrainingValue',5000);

doTraining = false;
if doTraining
    % Train the agent.
    trainingStats = train(agent,env,trainOpts);
else
end
```