2020

A Series-Elastic Robot for Back-Pain Rehabilitation

Elhussein Shata
South Dakota State University

A SERIES-ELASTIC ROBOT FOR BACK-PAIN REHABILITATION

BY

ELHUSSEIN SHATA

A thesis submitted in partial fulfillment of the requirements for the

Master of Science

Major in Mechanical Engineering

South Dakota State University

2020

THESIS ACCEPTANCE PAGE

ELHUSSEIN SHATA

This thesis is approved as a creditable and independent investigation by a candidate for the master's degree and is acceptable for meeting the thesis requirements for this degree. Acceptance of this does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department.

Kim-Doang Nguyen
Advisor Date

Kurt Bassett
Department Head Date

Dean, Graduate School Date

“I have no special talent. I am only passionately curious.”

– Albert Einstein

ACKNOWLEDGEMENTS

I am deeply grateful for my thesis supervisor Dr. Kim-Doang Nguyen for guiding me through this remarkable opportunity. Dr. Nguyen, you convincingly directed and encouraged me to be professional and do the right thing even when the road got tough. Without your persistent help, the goal of this project would not have been realized. It is whole-heartedly appreciated that your patience and great advice for my endeavors proved monumental towards the success of this study, and I will forever be effusive.

To Dr. Jeffry Doom, your passion for mathematics was my first true introduction to modeling and simulation and has helped me form a strong theoretical foundation in my work. Thank you for all the knowledge you taught us in class.

To Praneel Achray and Travis Deegan, my lab partners. Thank you for being part of the team and for all the insightful conversations we have had. Your friendship means a lot to me and I will always remember it.

To the department of Mechanical Engineering at South Dakota State University, thank you for providing an astonishing educational environment with all the available research resources. Thank you to each professor that has helped and supported me.

To Garret and Tayler, the workshop managers. Thank you for your physical and technical contributions, it is truly appreciated. Without your support, this project could not have reached its goal

To my parents and big brother, thank you for your moral support and always believing in me to make it this far. I hope I was able to make you proud someday.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

A SERIES-ELASTIC ROBOT FOR BACK-PAIN REHABILITATION

ELHUSSEIN SHATA

2020

Robotics research has been broadly expanding into various fields during the past decades. It is widely spread and best known for solving many technical necessities in different fields. With the rise of the industrial revolution, it upgraded many factories to use industrial robots to prevent the human operator from dangerous and hazardous tasks. The rapid development of application fields and their complexity have inspired researchers in the robotics community to find innovative solutions to meet the new desired requirements of the field. Currently, the creation of new needs outside the traditional industrial robots are demanding robots to attend to the new market and to assist humans in meeting their daily social needs (i.e., agriculture, construction, cleaning.). The future integration of robots into other types of production processes, added new requirements that require more safety, flexibility, and intelligence in robots.

Areas of robotics has evolved into various fields. This dissertation addresses robotics research in four different areas: rehabilitation robots, biologically inspired robots, optimization techniques, and neural network implementation. Although these four areas may seem different from each other, they share some research topics and applications.

CHAPTER 1: OVERVIEW

1.1 INTRODUCTION

In recent years, robotic systems have been integrating in many applications due to the necessity of assistance. Rehabilitation robots, for example, are gradually emerging into the medical field. They aim to aid a person achieve a physical level which will eventually improve their quality of life. Although the medical field has been occupied by robots, they are not meant to replace qualified personnel such as doctors and nurses, but to assist them in daily routine work and certain tasks. Rehabilitation robotics is a promising field that has slowly evolved through the years and is taking off in the recent years. A wide variety of medical applications have adopted the concept of rehabilitation robot. This field may include a wide array of mechatronic devices ranging from robots for supporting rehabilitation therapy or for providing personal assistance in hospital and residential sites or artificial limbs. The field of rehabilitation robotics is less developed than that of industrial robotics. Many assistive robotic systems have featured an industrial robot arm for reasons of economy and availability. However, the specifications for robots in these two application areas are very different. The differences arise from the involvement of the user in rehabilitation applications. Industrial robots are typically powerful and rigid to provide speed and accuracy. They operate autonomously and, for reasons of safety, no human interaction is permitted. Rehabilitation robots must operate more slowly and be more compliant to facilitate safe user interaction. Thus, rehabilitation robotics is more akin to service robotics, which integrates humans and robots in the same task.

Apart from the rehabilitative robots, there is an extensive activity in introducing inspiration from biology to produce novel types of robots with adaptive locomotion

systems. Probably the most widely used biologically inspired locomotion system is the leg. However, there are some research groups focusing on other types of locomotion, such as the systems used by snakes and fishes. Brachiating robots, inspired from gibbons, is an evolving area that not many scientists have worked on.

Another field in robotics is optimization. Such a problem arises in many different disciplines. It is the selection of the best element from a set of alternative solutions with respect to a given constraints. There is a growing interest in optimization in robotics as it offers an interesting way to generate optimal behaviors based on basic principles (cost functions, constraints).

In general, in creating a new robotic system the process can be carried out in four stages. The first stage is the mathematical modeling, where different methods are used to describe the behavior of the system. The gap between the theoretical model and the real model is found in the second stage by direct measurements through a set of sensors. Thus, the true position of the robot's end effector or torque behavior is determined. In the third stage, the means of optimization techniques play an important role in identifying the parameters that vary from their nominal values and choosing the optimum points. And finally, in the fourth stage, different artificial intelligence (AI) techniques are used to provide the robot with intelligence and flexibility so it can operate in dynamic environments and in the presence of uncertainty.

A popular branch of artificial intelligence is called Neural Network (NN), or more precisely artificial neural network. NN is inspired by the biological neural network of animal brains. Multilayer perceptron (MLP) is one type of neural networks that learns how

to perform a task by using different supervised techniques. The use of neural networks, and in particular the multilayer perceptron (MLP), have been shown to be an effective alternative to more other traditional techniques.

In this thesis, we will explore rehabilitation robots, brachiating robots, optimization techniques, and MLP neural networks. The following section describes the main four topics investigated and the overall organization of each chapter.

1.2 ORGANIZATION OF THE THESIS

1.2.1 CHAPTER 2: REHABILITATION ROBOT FOR BACK-PAIN

This chapter is written based on my paper in [1]. This chapter addresses the robot-assisted rehabilitation for back pain, also known as lumbago. Lumbago is one of the most serious health problems affecting a large portion of the population. In this chapter, we design a robotic system to assist in the rehabilitation process a patient must go through while recovering. The design is composed of two springs in series connected to an end-effector via a pair of antagonistic cables. The spring and cable arrangement form an elastic coupling from the actuator to the output shaft. An input-output torque model of the series-elastic mechanism is established and studied numerically. The study also illustrates the variation of the mechanism's effective stiffness by changing the springs' position. In addition, we built a prototype of the robotic mechanism and design experiments with a robotic manipulator to experimentally investigate its dynamic characteristics. The experimental results confirm the predicted elasticity between the input motion and the output torque at the end-effector. We also observe an agreement between the data generated by the torque model and data collected from the experiments. An experiment with a full-scale robot and a human subject is carried out to investigate the human-robot interaction and the mechanism behavior.

1.2.2 CHAPTER 3: BRACHIATION ROBOT

This chapter is written based on my paper in [2]. Chapter 2 explores a different type of locomotion robots, known as brachiation robots. A robot with the ability to brachiate could prove to be very useful in reducing the number of work-related accidents due to falling from high altitudes or maintenance of high voltage towers. It could also help a

company's bottom line by reducing the number of workers required to accomplish a task. In addition, it can be used to scale lattice structures for inspection. Inspired by nature, gibbons prove to have the best pendulum like motion and power to weight ratio.

In this Chapter, we investigate the mechanics of gibbons and try to simplify its dynamics by a mechanical system. A simple way to characterize the motion of a gibbon is a two-bar linkage controlled by one motor. It has proven that this mechanism can achieve a good power to weight ratio and can approach the desired motion. By exploiting this motion, the device can create its own momentum to swing, and utilizing a system of grippers to grab and release members, will have the ability to navigate any structure. Furthermore, outfitting these robots with inspection tools such as cameras could allow the inspection of transmission towers and bridges safer, easier, and quicker. The hands-on approach of a mechanism like this could also accomplish tasks that may prove to be too dangerous or difficult for drones to accomplish.

1.2.3 CHAPTER 4: OPTIMIZATION OF A NONLINEAR SYSTEM

This chapter is written based on my paper in [3]. In this Chapter, we revisit the Modified Quasi-linearization Algorithm as a method to solve Nonlinear Programming Problems. The algorithm structure is first derived from the necessary conditions for a candidate solution to be an extremal. Secondly, we apply the numerical method to determine the optimal solution for a chemical reaction problem. The motivation for this application is the challenge of deciding the chemical mixture composition of any complex combination held under chemical equilibrium conditions. The problem mostly arises in the implementation and analysis of the performance of fuels and propellants that is used to

aerospace propulsion. Moreover, the algorithm can also be virtually applied to the study of complex organic compounds.

1.2.4 CHAPTER 5: A MULTI-LAYER PERCEPTRON NEURAL NETWORK WITH BACKPROPAGATION ALGORITHM

In this chapter, a neural network multi-layer perceptron is presented that contains one input layer, one hidden layer and one output layer, backed with back propagation algorithm. Neural networks (NN) are a set of algorithms, inspired from the human brain, it is mainly used to recognize patterns. They interpret data through a kind of machine awareness, labeling or clustering raw inputs. The data they recognize are values, contained in vectors, into which all real-world data must be converted.

In this chapter, the layers are constructed with eight-bit inputs, three bits hidden and eight-bit outputs where the outputs exactly follow the inputs. The back-propagation algorithm is implemented to understand how changing the weights and biases in a network changes the cost function. It is deployed to update the weights between input and hidden layer as well as the weights between hidden and output layer. The results shown present the behavior of the designed NN including convergence of error, hidden unit encoding and weight convergence.

1.2.5 CHAPTER 6: DESIGN PORTFOLIO

Presented in this chapter is a variety of different design projects that I had the honor of working on. From different types of variable stiffness actuators to behave-like-gibbons' robots. This chapter illustrates diverse projects that was designed during my master's program.

CHAPTER 2: REHABILITATION ROBOT FOR BACK-PAIN

2.1 INTRODUCTION

2.1.1 BACKGROUND ON BACK-PAIN

Back pain is one of the most epidemical health problems [4]. Up to 80% of all people suffer from this spinal musculoskeletal disorder (SMD) at some point in life [5]. Back pain affects different parts of the body including shoulders, neck and especially upper and lower back. Among these, low back pain, also known as lumbago [6], is one of the most common syndromes that occur at various ages [7]. It is a major cause for disability, also the second leading cause of activity limitation and sick leave throughout the world [8]. One common cause leading to lower back pain is due to activities involving carrying heavy loads. In such cases, the connective fibers of ligaments and tendons can begin to adhere to each other and lose resilience and may also tear down when a sudden overload occurs [9]. Since muscles are in constant communication with the central nervous system [10], ongoing tension prevents normal muscle functions [11] and lead to muscle spasms and further stability problems, which in turn can lead to chronic lower back pain and disability [12].

Recovery from back pain is slow and uncertain [13]. A recent study in [4] reported that about 60-70% of patients recover within 6 weeks, and 80-90% recuperate within 12 weeks with some help of rehabilitative therapy. However, after 12 weeks, back pain becomes chronic and leads to periods of intense pain, significant physical limitations, and activity impairment. Those who do not recover by 12 weeks account for up to 90% of total expenses related to this health-care problem. For example, the expenses exceed $90 billion

per year in the U.S. [14], $8.1 billion per year in Canada [15], and $9.17 billion per year in Australia [16].

The correlation between SMDs and motor control has been widely reported, for example, recent discussions in [17]. Rehabilitative therapy for the spine is essential for back-pain patients to regain their spinal mobility. Structured and repetitive exercises that result in bodily movement and energy expenditure by activation of skeletal muscles has proved to be effective for the recovery of functional spinal motor skills [18]. For instance, sit-up exercises, knee extension, cat-stretch, and aerobics would assist in back-pain recovery [19]. However, patients who are subjected to back pain disorders have challenges and difficulties in performing these tasks.

Current rehabilitative techniques for back pain require intensive, subjective assessment of motor function, and therapeutic procedures supervised by a team of physiotherapists [20]. In addition, the current practice lacks quantification in instructing patients as well as in the monitoring of rehabilitative progress. This hinders the design of the therapeutic procedures that fit patient characteristics and the prediction of therapy success. Finally, rehabilitation programs for back pain patients are excessively costly and are restricted to hospital environments.

Robot-assisted rehabilitation therapy is an emerging approach for musculoskeletal disorder treatment after neurologic injuries such as stroke and spinal cord injury. Robotic devices can help patients achieve the intensive, repetitive practice needed to stimulate neural recovery, reduce the need for supervision, and improve cost-benefit profiles [21]. Most recent research focuses on robotic assistive devices for the recovery of functional upper and lower limbs [22]. The few existing robotic systems for back-pain rehabilitation

(see [23], for example) tend to override the user's motion. They are designed with rigid links and joints and do not account for the neuromotor delays in the musculoskeletal systems [24]. In this chapter, we bridge these gaps in rehabilitation robotics by:

- Designing a bio-assistive device that assists in the rehabilitation of back pain patients.
- Ensuring the elasticity of the system to accommodate unintentional movements from the patient.
- Varying the stiffness of the mechanism when desired to achieve the required resistance.
- Validating the proposed model with a series elastic actuator prototype.

The compliant actuator that drives the device works by restoring a desired position and creating an adjusted resistance accordingly. This movement, with the assistance of the mechanism, will help increase the strength in the muscles that function to stabilize or mobilize the spinal column. The device will offer the amount of motor practice needed to relearn spinal motor skills with less therapist assistance. In addition, the compliant design of the robotic rehabilitator will guarantee safe interactions between a patient and the robot during a rehabilitative session due to the elastic component that is integrated in the design. This device will be one of the very few robotic systems that assist in the recovery of patients with severe back-pain.

2.1.2 BACKGROUND ON ELASTIC ACTUATORS

Robotic rehabilitators are essentially mechanical structure-based devices built for physical rehabilitation. Since these robots are designed to share a common space and proximity to participants, the users' safety is a major concern. In addition, the rehabilitative

mechanism used must accommodate for any uncontrollable or involuntary movements produced by patients due to musculoskeletal disorders. To address these concerns, adding compliant elements to traditional rigid link-joint mechanisms has proved to be an efficient way to inherently reduce the risk of accidents in human-robot interactions and minimize the effect of high-frequency involuntary movements [25].

In particular, early seminal work in [26] develops a series-elastic actuator (SEA), which is composed of a motor connected in series with a spring and a load. The torque transmitted to the end load by the actuator is proportional to the deformation of the spring. This mechanism is effective for shock absorbance, which is critical to legged robots. However, since the stiffness of the spring is not variable, it is limited to one natural frequency. In general, SEA can be categorized into three types depending on the springs' configurations:

- Reaction Force Sensing Series Elastic Actuator (RFSEA) — where the elastic element (springs) is positioned before or after the motor.
- Transmitted Force Sensing Series Elastic Actuator (TFSEA) — where the elastic element is positioned inside the force transmission's phase or gearbox.
- Force-sensing Series Elastic Actuator (FSEA) — where the elastic element is positioned after the force transmission and before the applied load.

Figure 1 graphically illustrates each configuration.

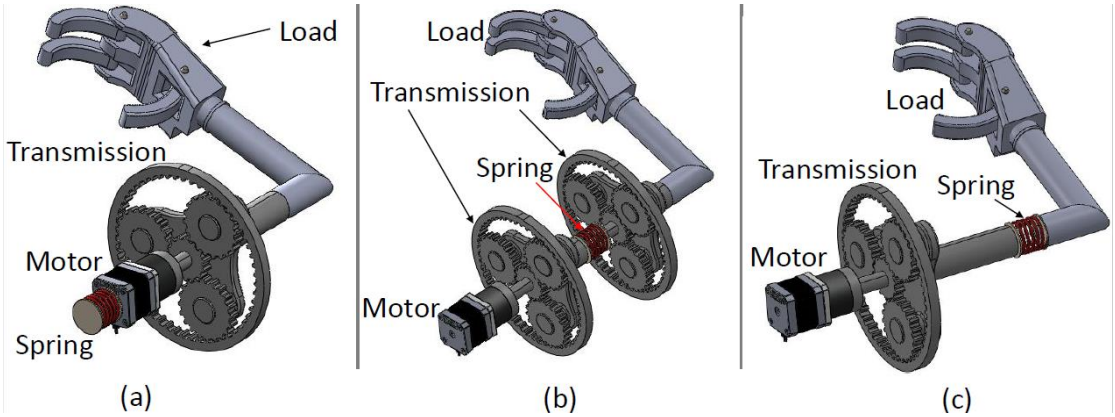

Figure 1: Configurations of a series-elastic actuator: (a) RFSEA, (b) TFSEA, and (c) FSEA

An example of the RFSEAs is the one developed in [27], which consists of a motor housed with a spring to absorb all the horizontal shocks. The torque of the motor is transferred to an oscillating end-effector. Another example is the SAE presented in [28]. This mechanism is designed with two springs in series attached with a rope and connected to a motor on one side, and a load on the other side. The relationship between the torque exerted from the motor and the load applied is dependent on the spring's deformation and stiffness.

An example of the TFSEAs is the cPEA (Compact Planetary-geared Elastic Actuator) introduced in [29], which is composed of with a torsional spring mounted between a planetary gearbox and the load. In this mechanism, the careful arrangement of the components reduces redundant space and optimizes its compactness. In addition, work in [30] develops the BIC-PEA (Bi-directional Clutched Parallel Elastic Actuator). The mechanism is designed with a differential gear, a spring that coupled to the joint of the robot, and brakes to stop the motion at any time. The concept of this mechanism is to store the kinetic energy in the spring and then use that energy to accelerate the joint in any anticipated direction. Since the spring takes over part of the torque that is required to

perform the task, the energy consumption of the motor is reduced; however, a prediction for the future final position must be made in a safe manner to avoid collision.

An example of the FSEA is the Compact Rotary SEA (cRSEA) reported in [31]. In this mechanism, a worm gear is utilized as the power transmission mechanism with a rotary spring as the compliant element. This mechanism was the actuating component of a robotic orthosis that assists the motor function of a knee joint. In the same category, there falls the differential elastic actuator in [32]. This intriguing mechanism employs a differential transmission system in the form of a harmonic drive. The torque produced by a brushless DC motor is transmitted to the wave generator of the harmonic. A torsion spring is installed between the motor and the load to provide the differential coupling. In this mechanism, the torque is captured as the transmitting torque by the spring connected between the flex spline and the ground.

There are many variable stiffness actuators that are tangential to the above three categories of elastic actuators. For instance, via a special arrangement of springs, a belt, and two motors, work in [25–33] shows that it is possible to vary the effective stiffness and impedance properties of the elastic actuators. Another example is the hybrid variable stiffness actuator in [34], which is based on the adjustable moment-arm principle. The mechanism can switch between a rigid mode, in which it works like a traditional rigid joint, and an elastic mode with a wide range of stiffness depending on the length of the moment arm. Similarly, in [35] that is a Rotary Flexible Joint (RFJ) that can be used in many applications. And in [36], a robot for lower limbs rehabilitation that can produce smooth and precise motions for ankle, knee, and hip. The Mechanically Adjustable Compliance and Controllable Equilibrium Position Actuator (MACCEPA) [37] was originally designed

for a biped robot. It is then adopted for many other legged robots. This work develops a clever procedure for establishing the torque model, which is leveraged in our chapter to establish our own force model.

Through the brief review above, there have been many phenomenal design innovations that facilitate elastic couplings in actuation systems. However, these design principles have been mostly applied to enable the locomotion of legged robots and rehabilitation robots for limbs. They have not been used to design a rehabilitation robot for the recovery of back pain. Motivated by this, this paper develops a series-elastic actuating mechanism for back-pain physical rehabilitation. Hence, this chapter is arranged as follows:

- Section 2.2 describes the bio-inspired design of the mechanism including the requirements and the working principle, as well as presenting a CAD model for the proposed mechanism.
- Section 2.3, we construct a torque model of the mechanism and show that the mechanism is capable of varying its effective stiffness by adjusting the positions of the springs. We also discuss the effects of each parameter in the elastic actuation system.
- Section 2.4, a small-scale prototype of the mechanism is fabricated and presented. In this section, we also discuss the experimental set-up and analyze the experimental results that validate the theoretical torque model.
- Section 2.5 experiments a full-scale mechanism with a human subject.
- Finally, Section 2.6. gives several concluding remarks.

2.2 DESIGN

2.2.1 DESIGN REQUIREMENTS SPECIFICATION

In designing a robotic rehabilitation device, clinical acceptance relies on the added value in offering features or functions difficult to achieve with conventional therapy [38]. Features such as exact repetitive movements, adjustable resistance, and movement sensing capabilities would theoretically increase clinical acceptance [39]. The systems on the market are large, expensive, not portable and intended only for clinic use with therapy staff present.

Additionally, cost includes system maintenance and training [40]. Essential factors, including how easy a rehabilitation robot is to use can have a great impact on the acceptance and integration of the rehabilitative robots into clinical practice. Hence, in order to design mechanical systems, it is vital to obtain a better understanding of why the currently available devices are not commonly used. Designing an acceptable robotic rehabilitation system must address the needs of the end users. User-centered design is an approach that can help in the development of a usable system by focusing on the users and their needs and by including them at every stage of the design process. The development of effective, safe and appropriate rehabilitation technology is crucial in our design process to meet the needs of stroke survivors.

During a rehabilitation session, a patient is an integral part of the robotic mechanism where the robot senses the command motion of the patient and provides assistance in terms of effort. This helps the patient complete an assignment to promote his/her active movement and improve the progress of motor function recovery. The main requirement in the development of such a robotic rehabilitator is that safe interactions between a human user and the bio-assistive device must be absolutely ensured [41].

This challenge is the fundamental motivation for the robot design in this paper. In gym equipment, elastic coupling belts are commonly used [42], which partially absorb unintentional or instinctive actions produced by users as well as reduce the risk to them while being used. Inspired by this, our design is based on a cable-driven elastic coupling between an actuator and an end-effector that interact with a user. An elastic coupling actuator with adaptable compliance can be modeled as spring driven by a cable of which both the force in the cable and the position of the springs can be controlled separately.

Furthermore, one of the requirements for the proposed design is that the torque applied on the springs should be zero when there is no spring displacement, and the torque should be symmetrical around the equilibrium position. In the design, the springs can gradually manipulate their position to change the formed stiffness as needed to meet the necessity of the patient, therefore the control of the equilibrium position and the compliance is completely independent. The mechanism should be simple, easy to use and control. Simplicity results mostly in sturdiness and low-cost designs.

2.2.2 DESIGN PRINCIPLES

Exercise therapy is a management strategy that is widely used in low-back pain. It encompasses a diverse group of involvements ranging from general physical fitness or aerobic exercise, to muscle strengthening. Repetitive exercises have proven to be an effective method for back pain treatment [43]. Partial crunches or Pilates for instance, can help strength the muscles around the spine therefore reinforce the back [44]. However, patients with chronic back pain would have challenges performing these exercises without assistance. Also, Sit-Ups exercises, although they create pressure on spinal disks, they require assistance for back pain patients to accomplish. Various types of flexibility and

stretching exercises can treat lumbago patients, but they require a helping hand and lack the assistance element. Current assistive systems used in physiotherapy clinics such as Isostation B200 [45] or other exercise devices like [46] may lack the ability to variate stiffness. Therefore, the proposed robot can provide a useful aid for the therapist in performing repetitive tasks of the treatment program, and in assisting the patient and the therapist to achieve progressive results.

Most of the rehabilitation devices that are currently used for back pain rehabilitation doesn't accommodate therapeutic resistance [47]. On top of that, prefabricated machines would not be suitable for this application. In particular, available industrial robotic systems have a number of capabilities or characteristics that do not suit therapeutic applications in-terms of manipulator speed and payload, size of workspace, range of motions, types of movements, safety and man–machine interface. Most importantly, industrial robots are not designed with the safety considerations required for systems that work as intimately with patients as a rehabilitation robot must.

We design the robotic mechanism following the principle of series-elastic actuation, in which the compliant element's stiffness is variable. Traditional actuation systems are designed such that the interface between the motor shaft and the payload is as stiff as possible, and is usually a rigid coupling, like most industrial manipulators. In contrast to rigid coupling actuators, a variable-stiffness actuator is installed with an elastic interface between the motor output and the payload.

Variable-stiffness actuators offer many advantages especially to a rehabilitation robot that physically interacts with human users. Firstly, since the patient and the device perform a therapeutic exercise for musculoskeletal recovery in collaboration, low-stiffness

joints and low reflected inertia reduce the risk of the robot hurting the participant. In addition, variable impedance actuators offer better accessibility to haptic feedback, and therefore, may achieve more accurate and stable force control. Moreover, the elastic couplings partially absorb uncontrollable or involuntary movements often produced by patients with spinal musculoskeletal disorders. The residual vibrations will be actively isolated by the control algorithm of the rehabilitator. This is important because we want the robotic system to provide force and movement assistance that is as stable as possible during operation.

2.2.3 CONCEPTUAL DESIGN

Figure 2 shows the conceptual design of the backpain rehabilitation robot prototype. We use a crank-slider mechanism to convert the rotation of a motor to linear motion of a slider, which is connected to a carriage via a combination of springs and dampers (spring box). Both slider and carriage are constrained to only move horizontally. The motion of the carriage then pulls an antagonistic pair of cables to rotate the joint as shown in the kinematic diagram.

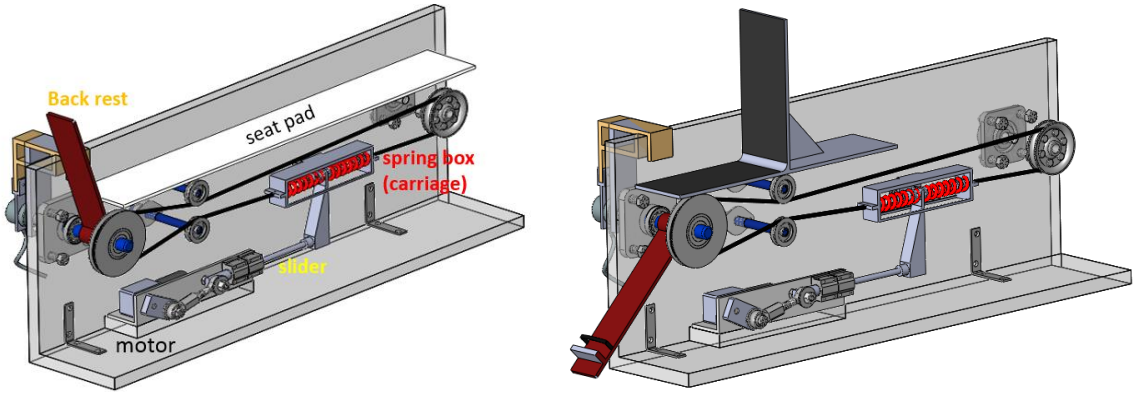

Figure 2: CAD model of the proposed design with sit-up(left) and knee-extension(right) configuration

Initially, we design this prototype to test the validity of the mechanism. The frame used to serve as the main plane is made from wood and have dimensions of 81 cm width and 31.75 cm height. The motor used has a speed of 35 RPM and drives a slider that is 23 cm long. The carriage (spring box) is 17.78 long and 3 cm wide. The springs used are 1.8 cm diameter and 8 cm long. The vertical link that is driving the springs is 12 cm height and 2 cm wide. Moreover, the idle pulley on the right is 7.32 cm diameter, and the driving pulley on the left is 10 cm diameter. The total length of the cable is 132 cm and the driving shaft is 2 cm diameter.

This design is inspired by the antagonistic pairs of muscles found in many animals. The mechanism set-up forms a compliant coupling between the motor shaft and the joint, which guarantees safe human-robot interactions when the patient and the device perform a therapeutic exercise for musculoskeletal recovery in collaboration. The design offers accessibility to haptic feedback by measuring the spring displacements, and therefore, may achieve more accurate and stable force control. Furthermore, the mechanism is so versatile that it is compatible with different therapeutic exercises. In particular, Fig. 3 and Fig. 4 show two configurations with slightly different end-effector designs: one for sit-up exercises and the other for knee extension exercises, which are some of the most common exercises for back-pain rehabilitation.

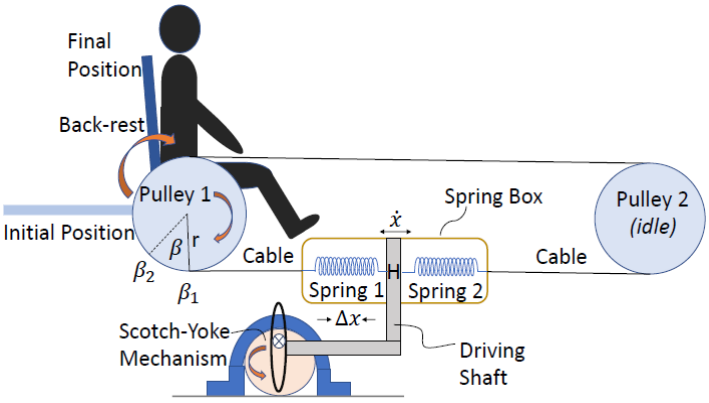

Figure 3: Sit-up configuration

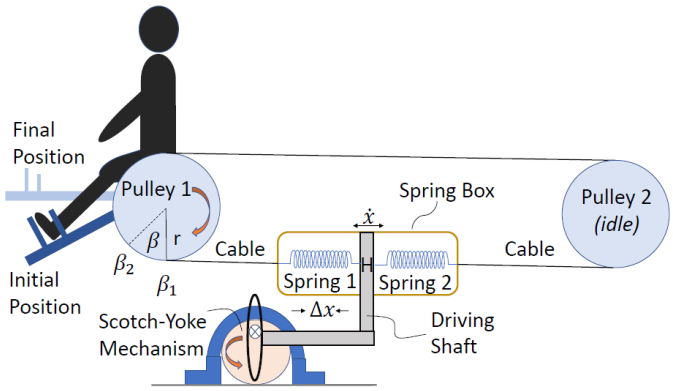

Figure 4: Knee-extension configuration

As seen in both configurations, two springs are installed in a box and are connected in series to a shaft. The vertical position of the other two ends of the springs can be adjusted to change the effective stiffness of the elastic coupling. To ensure force transition, a cable is deployed and is connected to each side of the spring box and can rotate around two pulleys, one of which is idle and the other drive the end-effector that interact with a user. The length of the cable is always constant. Once the Driving Shaft H moves horizontally with a velocity $\dot{x}$, it will result in an instantaneous displacement Δx for Spring 1 and $-\Delta x$ for Spring 2. As the spring box is designed to respond to the drive shaft motion, it will only move horizontally, i.e. to the left or to the right depending on the elastic force direction. As a result, tension will be generated in one side of the cable, generating an angular

displacement of Pulley 1, and hence of the end-effector that provides the assisting force for the user. For instance, when the driving shaft moves to the right, Spring 1 is extended and Spring 2 is compressed, the bottom cable is under tension, and the end-effector is displaced in the counterclockwise direction, and vice versa.

This design system accommodates the variation of the effective stiffness of the elastic coupling from the torque produced by the motor to the torque at the end-effector that interacts with a user. This stiffness variability enables a change in the level of physical assistance or resistance to match with individual users and their stage in backpain recovery. There is a torque model that governs the relationship between the actuator's motion and the output torque acting on the user. In the next section, we will have a deeper look into this torque model as well as the mechanism for varying the effective stiffness of the proposed design.

2.3 TORQUE MODEL

As discussed earlier, the designed mechanism facilitates the capability of changing the geometry of the spring arrangement, resulting in a variable stiffness actuation mechanism. A zoomed in illustrative diagram for the Spring Box is shown below in Figure 5 that conceptually depicts this idea. The fundamental hypothesis is that when the positions of the ends of the two spring are changed, the dependence of the output torque at the end-effector and the input motor's motion will change accordingly. Therefore, the resultant vertical motion of the springs will create two new variables to control, theta (θ) and alpha (α). Which will eventually result in an adjustable compliance output. To validate this hypothesis, we will first establish a torque model with the position of the springs being variables.

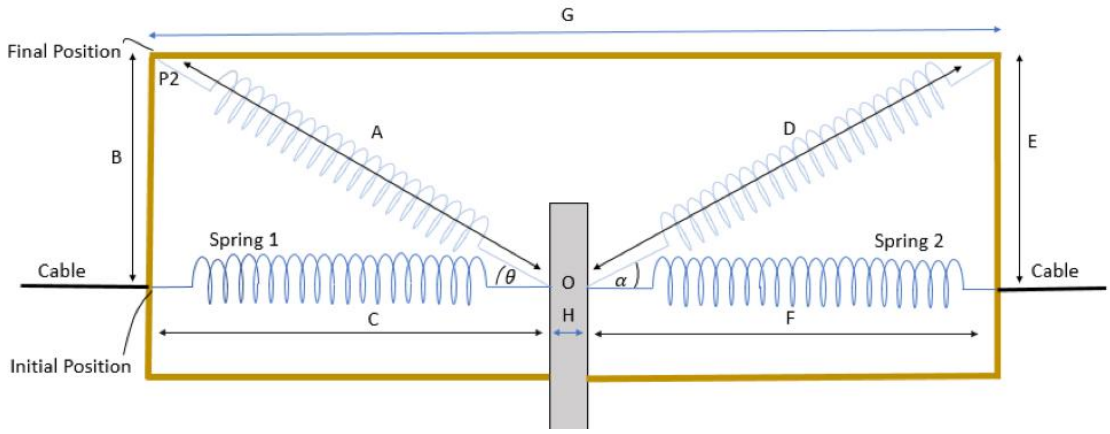

Figure 5: The spring box that facilitate variable stiffness by changing the position of the springs.

2.3.1 NOMENCLATURE

All variables and parameters are defined as follows (depicted in Figures 3, 4, and 5):

β = Angular displacement of Pulley 1
β_1 = Initial position of Pulley 1
β_2 = Final position of Pulley 1
A = Hypotenuse displacement of Spring 1
B = Opposite displacement of Spring 1
C = Adjacent diplacement of Spring 1
θ = Angular displacement of Spring 1
θ_{max} = Maximum angular displacement of Spring 1
D = Hypotenuse displacement of Spring 2
E = Opposite displacement of Spring 2
F = Adjacent displacement of Spring 2
G = Total length of Spring Box
α = Angular displacement of Spring 2
α_{max} = Maximum angular displacement of Spring 2
H = Driving shaft's thickness
F_s = Spring force acting horizontally to the spring box
F_{in} = Initial spring force of Spring n
F_{fn} = Final spring force of Spring n
F_f = Friction force
L_{10} = Natural length of Spring 1
L_{20} = Natural length of Sprin 2
k = Spring constant
Δx = Linear displacement of the spring
r = Pulley radius
T_n = Torque generated due to Spring n
T = Total torque generated
T_f = Totoal frictional torque

2.3.2 DERIVATION

We will first analyze the force model of Spring 1 and that of Spring 2 will follow accordingly due to the symmetry of the spring box which is shown in Figure 5. The torque produced at the shaft of Pulley 1 (the end-effector) is the total torque of the system due to the net displacement of both springs. We consider the input of the mechanism to be the spring displacement Δx, and the output to be the torque T generated on the rotating joint (the driving shaft). At initial position of Spring 1, $\theta = 0°$ and$\Delta x = C - L_{10}$. Applying Hooke's law, the elastic force of Spring 1 at initial configuration can be obtained as follows

$$\begin{gathered} At\ initial\ position: \quad \theta = 0°, \qquad A = L_{10} \\ F_s = k\Delta x \\ F_{i1} = k(C - L_{10}) \end{gathered} \tag{1}$$

$$At\ final\ position: \quad \theta \neq 0°, \qquad A \neq L_{10}$$

$$F_1 = k(A - L_{10})$$

$$F_1 = k\left(\frac{C}{cos\theta} - L_{10}\right)$$

$$F_1 = k\left(\frac{C}{\frac{C}{\sqrt{C^2 + B^2}}} - L_{10}\right)$$

$$F_1 = k\left(\sqrt{C^2 + B^2} - L_{10}\right)$$

$$F_s = F_1 cos\theta$$

$$F_s = k\left(\sqrt{C^2 + B^2} - L_{10}\right) cos\theta$$

$$T = F_s\, r$$

$$T_1 = k\left(\sqrt{C^2 + B^2} - L_{10}\right) cos\theta\, r$$

$$T_1 = k\left(\sqrt{C^2 + B^2} - L_{10}\right)\left(\frac{C}{\sqrt{C^2 + B^2}}\right) r$$

$$T_1 = r\, k\, C\left(1 - \frac{L_{10}}{\sqrt{C^2 + B^2}}\right)$$

As mentioned earlier, the effect of spring 2 can be obtained following a similar approach for Spring 1. Thus, accounting for the effect of both springs, and assuming T_f is the total frictional torque along the components' movements, we obtain the total torque by combining both springs and considering the expansion of Spring 1 and the compression of Spring 2 at an instantaneous moment, will therefore result in,

$$T = r\,k\,C\left(1 - \frac{L_{10}}{\sqrt{C^2 + B^2}}\right) - r\,k\,F\left(1 - \frac{L_{20}}{\sqrt{F^2 + E^2}}\right)$$

$$T = r\,k\left[C\left(1 - \frac{L_{10}}{\sqrt{C^2 + B^2}}\right) - F\left(1 - \frac{L_{20}}{\sqrt{F^2 + E^2}}\right)\right] - T_f \qquad (2)$$

where, $F = G - H - C$. Equation 2 is the torque model that governs the torque produced by the mechanism at the end-effector in response to a displacement C and F (but these are dependent) of the springs produced by the motor's input motion. The effective stuffiness of the elastic coupling can be varied by adjusting B and E as shown in the equation. While this model is nonlinear in C and F, it recovers the linear form in (1) when $B = E = 0$. Indeed, this is the equation governing a linear force relationship of a traditional series elastic actuation mechanism, in which the stiffness is exactly k.

2.3.3 INFLUENCE OF DESIGN VARIABLES

In this section, the influence of design variables will be investigated and simulated. To better study the mechanism, the variables k, G, L_{10} and L_{20} are chosen during the design and are fixed during normal operation. We set $B = E$ and change their values after each experiment. The natural length of both springs (L_{10} and L_{20}) is 8 cm. In each experiment, we vary the spring displacement C and employ the torque model in (2) to calculate the torque. The results from these numerical experiments are shown in Fig. 6. We can see from

these figures that the effect of the spring position is evident. When $B = E = 0$, the model produces a straight line corresponding to a linear torque relationship in traditional series-elastic actuators as discussed earlier. When B and E are increased, the model displays clear nonlinearity proportionally. This variation confirms our hypothesis stated earlier.

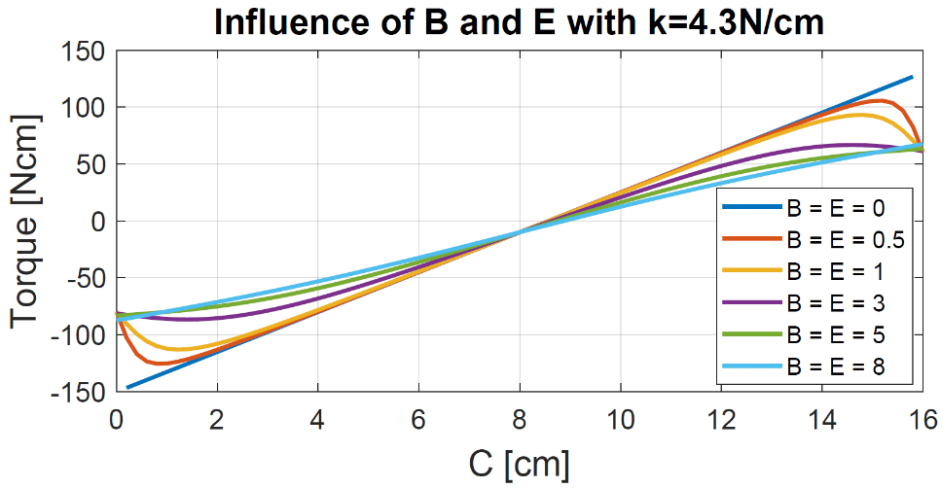

Figure 6: Dependence of the output torque on the spring displacement for different positions of the springs.

The dependence of the torque on the spring displacement is depicted in Fig. 6 as well. For every curve, since when C = 8 cm, the springs are at their natural length, the mechanism produces no torque. When $C \in (2,8]$, Spring 1 is compressed while Spring 2 is extended. The torque is acting in the clockwise direction, and hence is negative. Pulley 1 along with the end-effector are moving in the clockwise direction. The opposite trend is observed when $C \in (8,16]$. To observe the torque symmetry and the zero torque at equilibrium requirements, explained in the design section, B and E are set to be equal and adjusted identically in the vertical direction. However, B and E can be controlled separately to obtain the optimal stiffness as desired

It is worth noting that the torque model derived in (2) not only depends on the spring displacements C and F as shown explicitly, it is also a function of the angular displacement

of the springs, α and θ, implicitly. In order to visualize their influence in the torque model, we calculate the torque using (2) for a wide range of values of α and θ from -60° to 60° The results are shown in Fig. 7, which again illustrates that the torque model is highly nonlinear. Hence the effective stiffness of the elastic coupling is dependent on the spring positions a and q as well as the spring displacement.

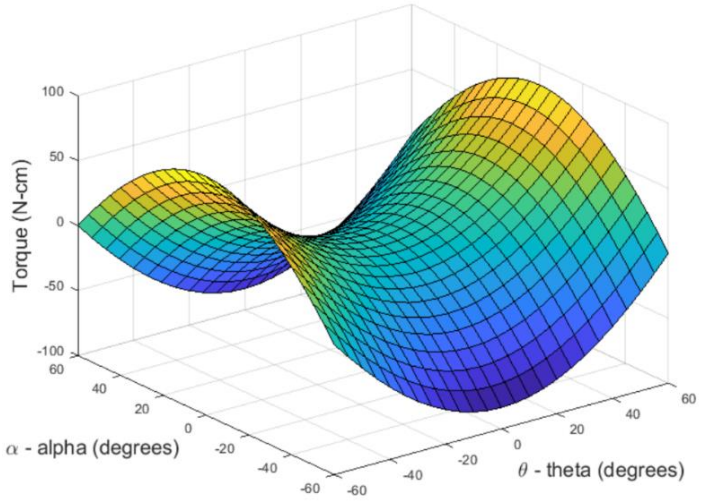

Figure 7: A three-dimensional visualization of the torque model

Furthermore, we discussed in section 2.1 that one of our design requirements is that the torque should be symmetrical around the equilibrium position and zero when there is no spring displacement or force added. Fig. 7 shows the symmetry of the system while changing α and θ. For instance, when α is zero, the exerted torque T is 50 N-cm. And when q is also zero, the exerted torque T is -50 N-cm. Which will cancel each other due to the symmetry. In return, the torque will be zero at this equilibrium position. Moreover, If the displacement of the springs is not balanced, the net displacement of the springs will not be zero, which will result in a produced torque on the end effector (Pulley 1).

In order to validate some of the properties discussed in Section 3., we build a prototype of this series-elastic actuation mechanism in a back-pain rehabilitation robot and analyze the experimental results with this prototype in the following section.

2.4. EXPERIMENT

Figure 8 shows the prototype of the series-elastic actuation mechanism for a back-pain rehabilitation robot.

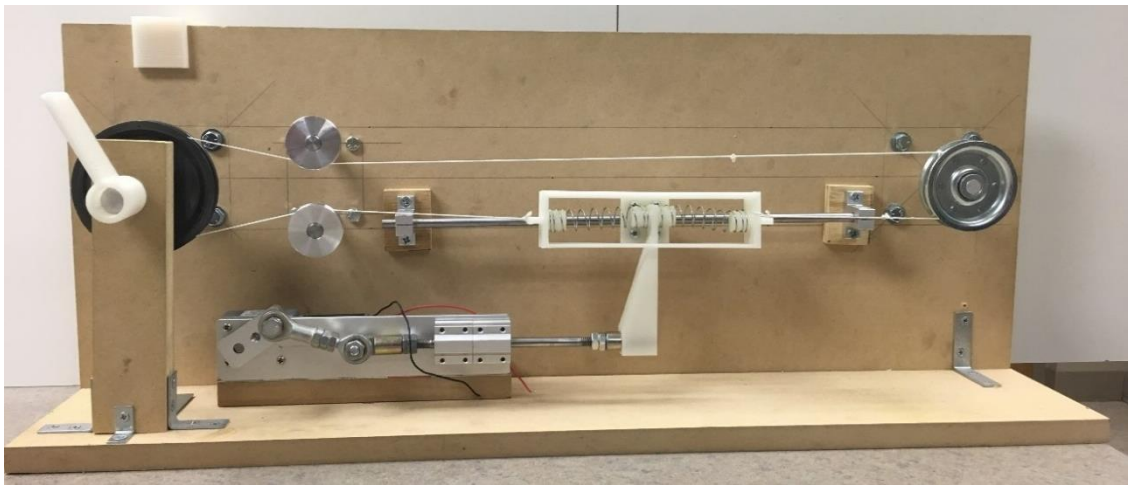

Figure 8: Fabricated prototype of the series-elastic mechanism proposed

A linear actuator is designed with a crank driven by a programmable DC motor. The crank's rotary motion is transmitted to a slider via a coupler link and two revolute joints. The slider's linear motion is supported by linear bearings and drive the motion of a vertical shaft. The other end of the shaft is mounted to two springs each of which is on one side as shown. The other ends of the two springs are connected to a 3D printed box. The entire spring box is supported by another set of linear bearings, that constrains the box to move along the horizontal direction. The outsides of the spring box's walls are attached to two polymer cables, whose ends are mounted to the pulley. The cable drives the end-

effector that interact with a user. The end-effector is the back-rest pad for sit-up exercises, or it can be the leg-rest pad for knee-extension exercises. The cables' tension is maintained by three idle pulleys, each of which is supported by a shaft inserted into a bearing flange mounted to the wooden support frame. Each cable pulls the output pulley in an opposite direction, for example, if the top cable is pulling, the end-effector rotate in the clockwise direction, while if the bottom cable is pulling, the end-effector rotates in the counterclockwise direction. The range of motion of the slider in the linear actuator is $2r_{crank}$. We want the range of motion of the end-effector to be 90°, which is equivalent to the cable range (i.e. circumference) of (1/4)$2\pi r_{pulley}$. Therefore, the size of the output pulley is designed as

$$r_{pulley} = \left(\frac{4r_{crank}}{\pi}\right) \tag{3}$$

The springs used in this design are compression springs with a coil thickness of 1.05 mm, the total diameter of 18.26 mm, and the total length of 80 mm. For this initial prototype, the compliance is fixed without changing the angular position of the springs. To capture the motion of the end-effector, an encoder is attached to the rotational axis of the shaft that is driving the output pulley and the end-effector. The encoder is a Signswise's incremental rotary encoder with 600 pulses per revolution. A Microchip ATMEGA 2560 microcontroller is used to program the motor and record the analog angle information produced from the rotary encoder. Through a serial communication interface, the data are exchanged between the controller and a computer for further analysis. In the first experiment, the equilibrium position is set to 0°. The output pulley was pulled manually out of the equilibrium position and released at an arbitrary angle (approximately 36° in this

case). As a result, the spring box and the pulley oscillate around the equilibrium position with a specific frequency. After the joint stops oscillating (less than 1 sec), the data is then gathered and plotted in Figure 9. The damping effect as observed is caused by friction at the pulleys. The response shows that the input-output relationship of the design behaves like a second order system, which represents an elastic coupling as opposed to a traditional rigid joint.

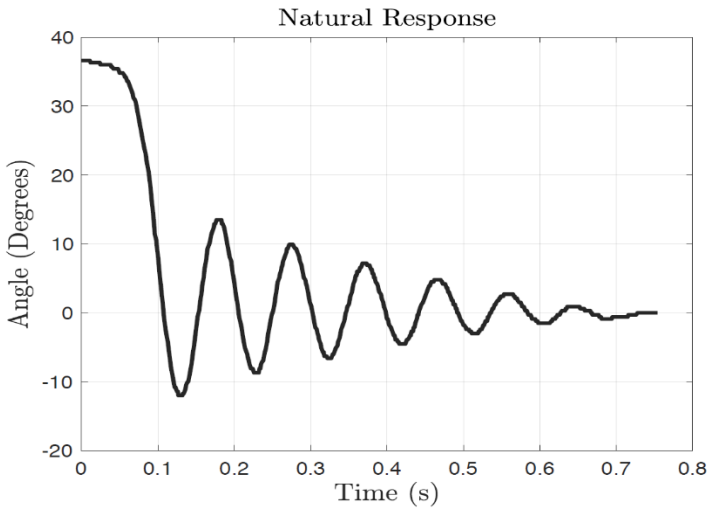

Figure 9: Natural response of the output pulley and the end-effector

In the second set of experiments, we validate the torque model established in Equation (2). This experiment focuses on examining the linear elastic behavior of the mechanism and the output torque due to various spring displacement. A seven axial robotic manipulator is used to provide and measure the precise displacements of the spring as shown in Figure 10. Moreover, the robot arm is embedded with force-torque sensors in all its joints, so it facilitates the measurements of the force applied onto its end-effector. The manipulator's end-effector is attached to the cable to pull and release the output pulley and

therefore move the spring box accordingly. The entire series-elastic actuation mechanism is clamped on a stationary table that is the base of the manipulator.

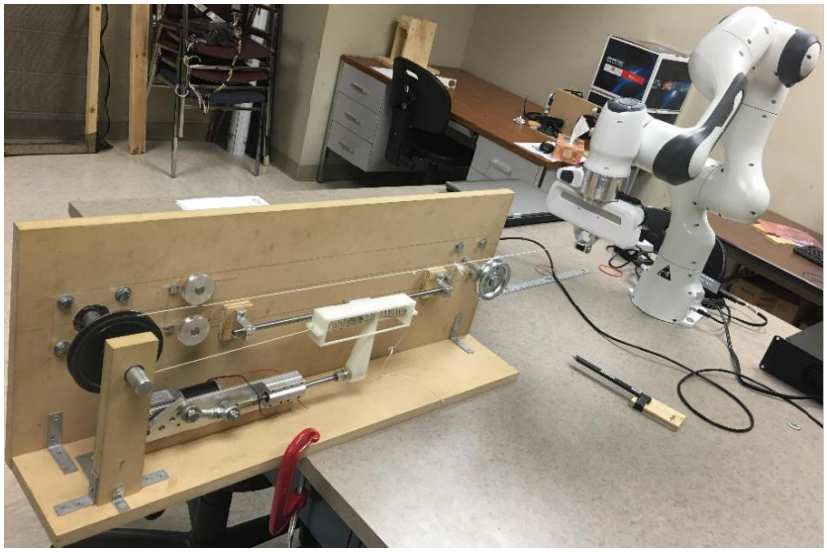

Figure 10: Experimental setup

Each experiment was divided into two segments. The first segment was to tension Spring 1 while compressing Spring 2. The manipulator pulled the output pulley in the clockwise direction, and the spring box was moving horizontally to the left. Spring 1 was tensioned up to 10 cm, while Spring 2 was compressed to 6 cm. The second segment was to compress Spring 1 and tension Spring 2. The manipulator was pulling the output pulley in the counterclockwise direction, and the spring box was moving horizontally to the right. Spring 1 was compressed down to 6 cm, while Spring 2 was tensioned to 10 cm.

The experiment was done ten times in each direction. The displacement for each spring was measured every 2 mm and the output force required to move the mechanism this displacement was obtained from the robot's sensors and software. The data was then filtered to include all the data points collected that are between 6 cm to 10 cm. The spring

stiffness was also calibrated based on the same experiment. Figure 11 shows all the experimental data plotted for the ten runs, along with the model data points computed using the torque model derived in equation (2). In this figure, the back dashed line indicates the data computed by the torque model in (2), while other solid lines are data collected from the ten experiments.

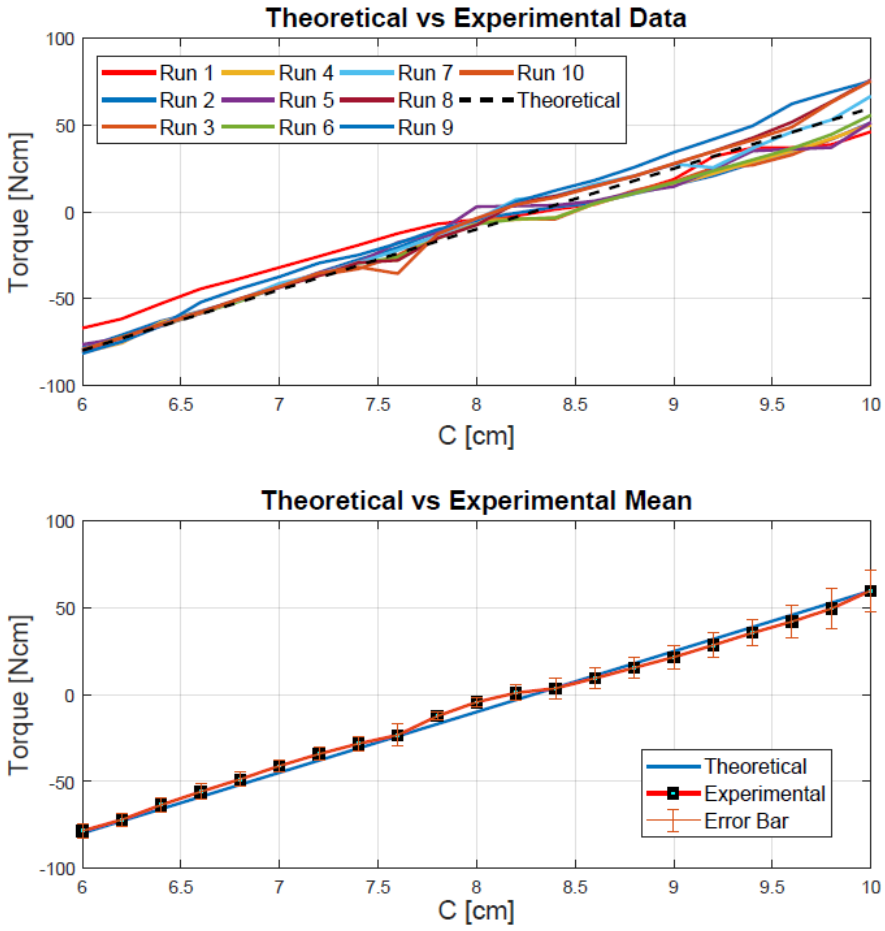

Figure 11: Experimental data for the measured torque versus the spring displacement

The collected data are summarized in Figure 11 (bottom) and shown with the error bars that indicates the standard deviation of each data point. From these data figures, we can see that the torque produced by the theoretical torque model is close to those measured from the experiments both in terms of the trend and the mean values. The small gap

between the two sets of data stems from the fact that the theoretical model does not take into account the effect of friction, while the friction forces influence the experimental data. In addition, the imperfect calibration of the spring stiffness is another source for the mismatch observed in these data figures.

2.5. EXPERIMENT WITH HUMAN SUBJECT

To conduct an experiment of the proposed robot interacting with a human subject, a full-size robot was built. A CAD model is shown below in Fig. 12. A stepper motor is used instead of the scotch-yoke mechanism to drive the system and to ensure the rigidity of the driving force by reducing the number of variable components (e.g. crank, gears). The stepper motor is connected to a screw shaft that drives the spring box. Linear bearings are added below the screw shaft to support and maintain the screw shaft alignment. A chain is used to transfer the motion of the spring box to a set of sprocket gears. A large sprocket gear is used to transmit the motion of the chain to end-effector that is then used to move the back-pad.

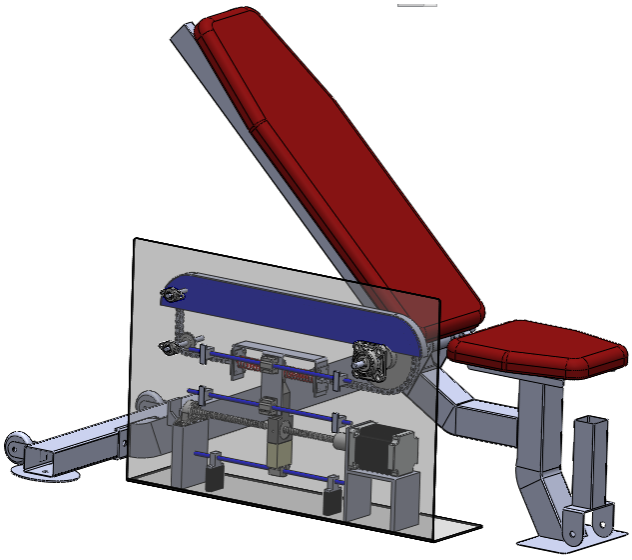

Figure 12: Full-size CAD model

The experimental apparatus, shown in Fig. 13, consists of A matrix of 14 force sensors, including 7 rows and 2 columns, was mounted on the back-pad. A Signswise's incremental rotary encoder is attached to the output shaft that drives the back-pad shown in Fig. 8 to measure the end-effector's angular motion. The force sensors and the encoder's data are both collected with a Microchip ATMEGA 2560 microcontroller. A Nema 34-step motor was used to actuate the mechanism. The stepper motor can produce a torque of up to 13 Nm (1841 oz-in) and is commanded by a digital stepping driver DM860I. The stepper motor and the driver are both controlled by a Microchip ATMEGA 328P microcontroller. The full-size rehabilitation robot has dimensions of 84×56×23 cm (width×height×depth). A screwshaft is used to convert the rotation of the motor to the spring box's linear motion. A system of sprocket gears and chains is then used to transmit that motion into the end-effector, resulting in the angular motion of the back-pad which interacts with the human.

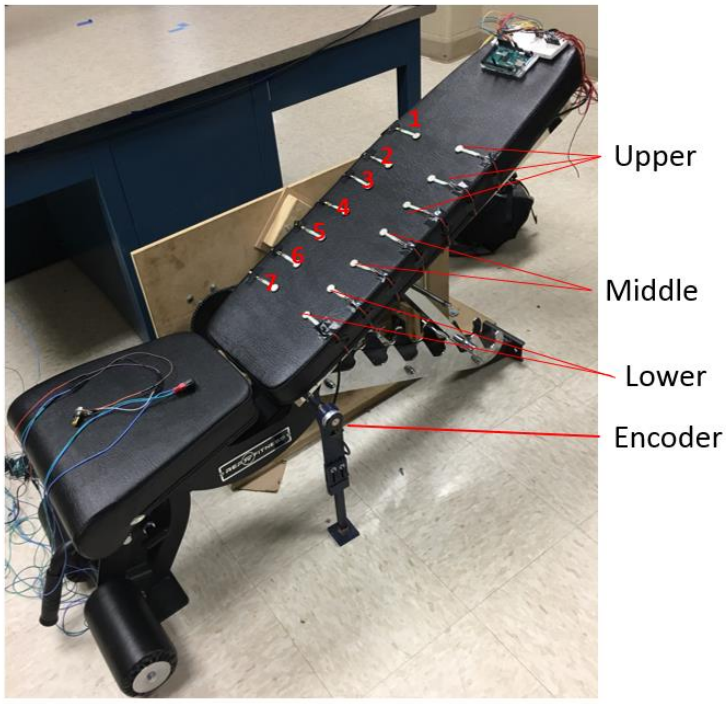

Figure 13: Human subject experimental apparatus

During the experiment, a human subject lays down on the bench shown in Fig. 14 and performs a sit-up exercise. The robot moves consequently with the subject providing additional support to complete the task. The robot is programmed to aid the patient while exercising and not to override the patient's motion to ensure safety.

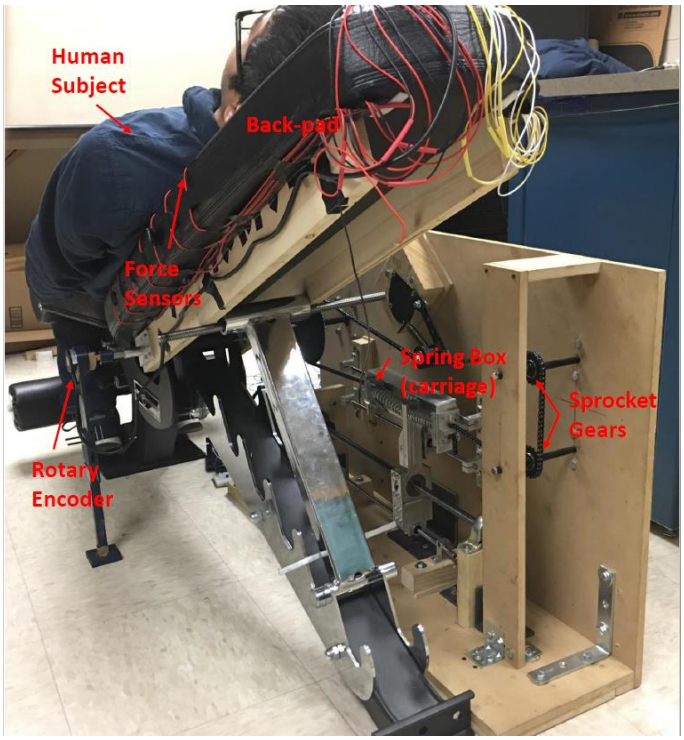

Figure 14: Experiment with human.

The human subject is 152 cm tall, and 54 kg. The experiment was repeated ten times for statistical data and the sensory signals were collected accordingly. Each row of the force sensors on the back pad was added together to obtain the total amount of force occurred in each row. During the experimental runs, row 1, 2, and 3 (top three rows on the Back-pad) didn't generate significant force signals since the human subject's upper back does not touching the force sensors. Sensor rows 4, 5, 6, and 7 recorded significant values which were captured and analyzed in Fig. 15.

Due to the sensitivity of the sensors and the noise produced by electrical components, they sometimes overshoot the sensed force when applied, this happened once and can be observed in the uppermost curve in the 5th Row Sensors shown in Figure 15.

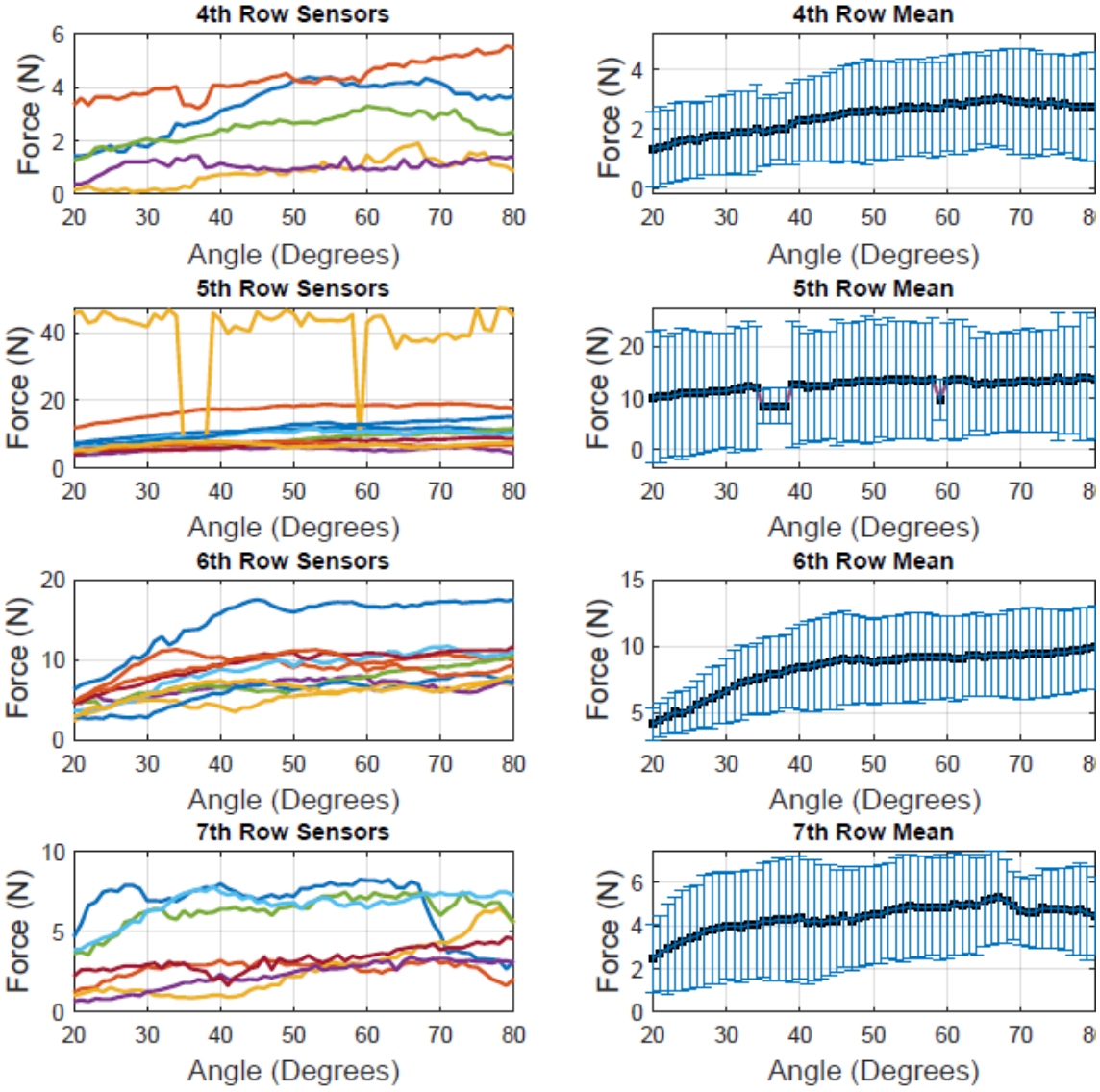

Figure 15: Force vs Angle recorded by the encoder and force sensors

Rows [4,7] (topmost and lowest sensors) recorded a peak of [3,5] (N). Rows [5,6] (middle sensors) recorded a peak of [14,10] (N) which is higher than the values in the topmost and lowest sensors. That is due to the posterior pelvic rotation of the spine

undergoing a sit-up exercise. That is to say, the nature of the sit-up exercises leads to the bending of the upper back [48] leaving the spine semi-curved, which can be observed in Fig. 15. Rows 4 and 7 recorded the lowest force value while rows 5 and 6 recorded the highest. Muscles are not mechanically independent from each other; they are synergistic, and some muscles have an antagonistic pair. This means that the mechanical threshold of one muscle cannot be evaluated independently since it varies with the force exertion abilities of other muscles [49]. A study in [50] shows the amount of force/effort exerted by a human while performing a sit-up exercise. It also shows that an increase in hip flexion results in posterior pelvic rotation.

That is to say, as the human subject come closer to their knees, the central nervous system activates more muscles to exert higher force for accomplishing the motion. Therefore, as the angular position increases, the force required to achieve that position also increases. The force measurements illustrated in our experiment demonstrate the amounts of effort contributed by the robot to assist the human subject with the physical exercise.

2.6. CONCLUSION

We have discussed the design of a bio-assistive robotic system that targets the physical rehabilitation of backpain, a musculoskeletal problem that imposes a huge burden on societies and health care systems. The main challenge in the development of a robotic rehabilitator is the safe interactions between a human user and the device since the human is an integrated part of the cyber-physical loop. To address this challenge, we design the robot with the series-elastic actuation principle that facilitates an elastic coupling, instead of a rigid joint, between an actuator and an end-effector that interacts with a user. This allows the unwanted forces to be absorbed along the compliant elements, and hence, is

easier to regulate the impedance of the mechanism. Furthermore, low-stiffness joints and low reflected inertia reduce the risk of the robot hurting the user, while stiffness can also be adjusted to adapt to each patient's recovery profile.

The fundamental hypothesis for the design is that the end-effector produces an elastic force in response to the motion of the input actuator. In addition, the effective stiffness of the elastic coupling can be varied by adjusting the position of the springs. In order to validate the hypothesis, we established a torque model, built a prototype of the design, and perform numerical studies and experiments using the prototype and a robotic manipulator. The results confirm the hypothesis and display agreement between the data produced by the torque model and the data collected from the experiments.

Moreover, we built a full-scale model to study the human behavioral interaction with the mechanism. The mechanism aim is to assist the patient in performing the required task while they are exerting 70-80 percentage of their effort to accomplish the motion. It is observed that the proposed mechanism assists the human subject to achieve a desired position, which in return will strength the spinal tendon.

Future work will focus on the control aspects of the robotic rehabilitator. In particular, we will consider the entire robot as a control system with a human user in the loop interacting with a variable-stiffness rehabilitation robot. Such a control system is challenging to handle. Due to the physical interaction the robot undergoes with a human patient in a rehabilitation session, safety and accuracy of response is crucial. In designing a controller, the control framework should integrate a compensatory adaptive controller to accommodate any fortuitous spasm occurs by the user. And to enhance the accuracy of the

output complaint and end-effector position with avoiding the supervision of a physiotherapist. To address these challenges, we will look into advanced control techniques, such as human-robot collaborative controller in [51] and the safety-enhanced collaborative framework in [52].

CHAPTER 3: BRACHIATION ROBOT

3.1 INTRODUCTION

Robots have been emerging rapidly in daily lives in the past decades. As technology advances, robots have enabled humans to live more convenient lives [53]. More importantly, robotic technology allows us to decrease the loss of human lives due to dangerous working conditions [54].

According to the U.S. Department of Labor, falls are one of the leading causes of death for workers while on the job [55]. In 2015, falls accounted for nearly 17% of the total workplace deaths. Secondly, the federal government is notoriously understaffed when it comes to infrastructure inspection [56]. A device that can increase the efficiency of an inspection or reduce the number of workers needed to complete an inspection will save tax dollars and possibly human lives.

Currently drones are being utilized and outfitted with cameras and other hardware to make inspecting quicker and easier [57]. However, the use of these drones has its own drawbacks. One issue is that drones could have problems navigating tight lattice structures resulting in a size concern. Another issue is the fact that system failures and damage will almost certainly end in complete destruction of the device due to the structural requirements for flying and the heights the drone will be working at. To remedy these concerns, a robot that can swing from a member to another in a lattice structure can be more durable than a drone and prove to be effective at aiding in infrastructure inspection and scaling lattice structures.

3.2 CONCEPT

Inspired from nature, gibbons are the finest animals that swings and generate pendulum like motion [58]. With the capability to produce the best power-to-weight ratio and perfect centralized mass distribution, gibbons brachiate with a very low mechanical cost. The closer the design comes to achieving this power to weight ratio the more efficient the design will be able to accomplish its task. The pendulum-like motion gibbons use to build up moment for a swing by kicking its legs can be replicated in a device that can use a similar concept to build up its own momentum and achieve alike motion [59].

Upon studying gibbons, their upper body is found to be a five-bar linkage [60] with the ability to grab and release surfaces using their hands, as illustrated in Figure 16. Although a five-bar linkage would be ideal to represent a gibbon's biomechanical body, it requires unnecessarily large number of degrees of freedom, which in turn implies a large number of actuators required to drive the robot. This dramatically complicate the controls of the robot [61,62].

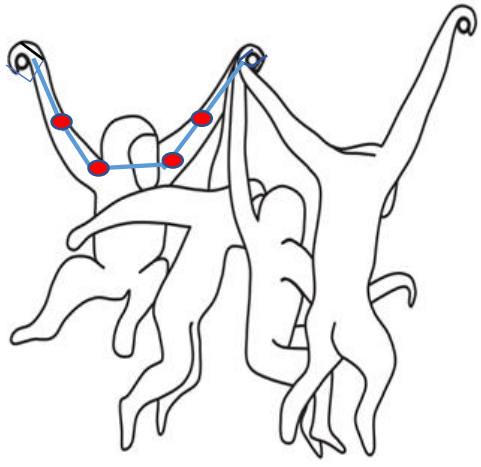

Figure 16: Five-bar linkage illustration

To address this problem, we employ a two-bar linkage mechanism that will reduce the complexity and weight of the mechanism, making the system easier to control and more applicable to use. It is similar to the mechanism in [63], which use a variable stiffness actuator with two motors. In contrast, here, we only design one actuating input at the elbow joint for swinging. Once successful swinging motions are achieved, the robot can brachiate among a rope or a lattice structure like bridges or transmission towers.

In addition, in order for the robot to reach more places and maneuver around corners, a motor is added at the end of each arm to create a wrist joint and to ensure the dexterity of turning in any structure. As shown in Figure 17, the brachiator mechanism is introduced with the wrist motion that allows each linkage to turn 180°. A set of two gears with identical number of teeth and pitch diameter are added at the joint to increase the stroke angle by allowing us to move the driving motor away from the joint. In addition, it ensures the force transmission with minimal energy loss.

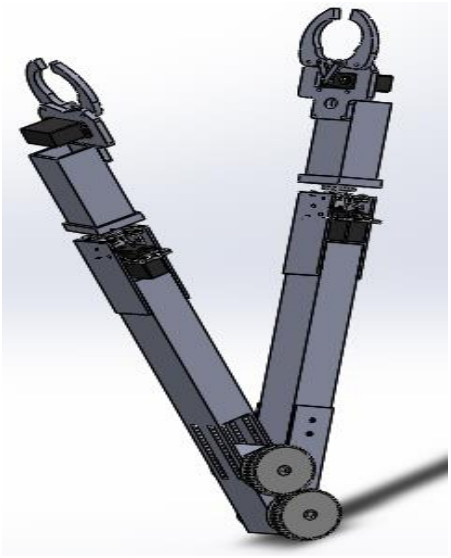

Figure 17: Proposed design

To reduce the weight and avoid adding any counterbalance deadweight, the motor is inserted in one of the links. By mounting the motor within the frame, the brachiator would be more stable since the motor's center of mass will be close to that of the frame's center of mass.

A gripper is added to the end of each link in order to ensure grasping and releasing surfaces. Each gripper is composed of a pair claw-like pieces driven by a servo motor. The motor used in each of the gripper's mechanism is HS-422 servomotor with torque of 57 oz-in which is tested to be enough to hold the system.

The wrist mechanism added to the design consists of a servomotor in a case that is mounted to a bearing. As a result, the body of the servomotor and the output rotates in different axis. The mechanism increases the servo's load-bearing capabilities by helping to isolate the lateral load from the servo spline and case. A three-dimensional model for the wrist mechanism in shown in Figure 18 below.

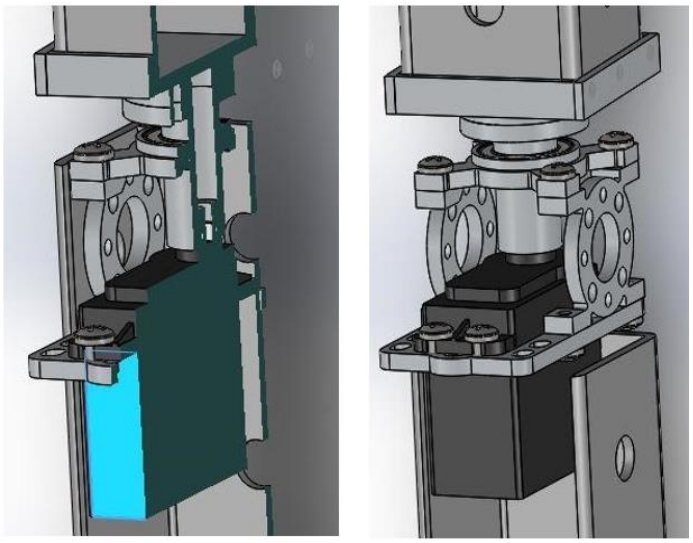

Figure 18: Wrist mechanism CAD

3.3 DYNAMIC MODEL

To express the dynamics of the model, we assume that the center of mass of each link is at its end, we then have

$$\begin{aligned} x_1 &= l_1 sin\theta_1 \\ y_1 &= -l_1 cos\theta_1 \\ x_2 &= l_1 sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \\ y_2 &= -l_1 cos\theta_1 - l_2 cos(\theta_1 + \theta_2) \end{aligned} \quad (4)$$

where (x_1, y_1) and (x_2, y_2) represent the position of linkage 1 in the x and y coordinate. Similarly, the position of linkage 2 is noted as Using the set of equations in (4), we are able to arrive at the dynamical equation of motion for the brachiator as follows:

$$M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \quad (5)$$

where

$$\begin{aligned} M_{11} &= m_1 l_1^2 + m_2(l_1^2 + 2l_1 l_2 \cos(\theta_1 - \theta_2) + l_2^2) \\ M_{12} &= m_2(l_1 l_2 cos(\theta_1 - \theta_2) + l_2^2) \\ M_{21} &= m_2(l_1 l_2 cos(\theta_1 - \theta_2) + l_2^2) \\ M_{22} &= m_2 l_2^2 \end{aligned}$$

$$c(\dot{q}, q) = \begin{bmatrix} m_2 l_1 l_2 sin(\theta_1 - \theta_2)(-\dot{\theta}_1^2 + 4\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) \\ -2m_2 l_1 l_2 (\dot{\theta}_1 + \dot{\theta}_2) sin(\theta_1 - \theta_2) \end{bmatrix} \quad (6)$$

$$g(q) = \begin{bmatrix} (m_1 + m_2) g l_1 sin\theta_1 + m_2 g l_2 \sin(\theta_1 + \theta_2) \\ m_2 g l_2 \sin(\theta_1 + \theta_2) \end{bmatrix} \quad (7)$$

Equations (5), (6), and (7) can now be summarized to,

$$M(q)\ddot{q} + c(\dot{q}, q) + g(q) = \begin{bmatrix} 0 \\ \tau \end{bmatrix} \quad (8)$$

where $q = [\theta_1, \theta_2]^{\mathsf{T}}, M(q)$ is the inertial matrix, and $c(\dot{q}, q)$ represents the nonlinearity, and $g(q)$ represents the gravity, friction, and the effect of disturbance. τ is the torque input.

3.4 MOTOR TORQUE CALCULATION

To anticipate the right value of torque needed per swing, a free fall calculation is performed on one linkage of mass of 0.6732 kg. We first calculate the expected velocity of one arm linkage as illustrated in Figure 19:

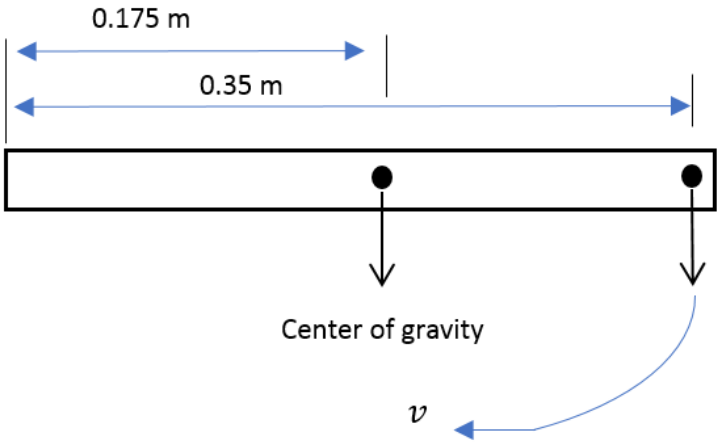

Figure 19: Velocity analysis

$$K.E. = P.E.$$
$$\frac{1}{2}mv^2 = mgh$$
$$\frac{1}{2}v^2 = gh$$
$$v = \sqrt{2gh}$$
$$v = \sqrt{2\left(9.81\frac{m}{s^2}\right)(0.175\ m)}$$
$$v = 1.8529\frac{m}{s}.$$

Next, we calculate the angular velocity of the link:

$$\omega = \frac{v}{R} = \frac{1.8529}{0.175}$$
$$\omega = 10.588\, rad/\sec = 101.12\, rpm. \quad (9)$$

Therefore, the torque required is:

$$\tau = F\, d$$
$$\tau = mgd$$
$$\tau = (0.6732\, kg)\left(9.81 \frac{m}{s^2}\right)(0.175\, m)$$
$$\tau = 1.155\, N - m$$
$$\tau = 163.56\, oz - in. \quad (10)$$

Multiplying the torque value by a safety factor gives $\tau = 163.56\, oz - in * 4 = 654.24\, oz - in$. This means we need a motor that can produce a minimum of $654.24\, oz - in$ of stall torque.

3.5 MOTOR EXPERIMENT

To validate the calculated values of the motor, this section discusses an experimental process to systematically select the motor. Another purpose for testing different motors was to study the motor's torque behavior versus current, and to validate that any motor that has a torque below the calculated torque in Equation (10) will not be a valid selection.

An initial experiment took place by testing a DC motor with torque of 120 oz-in, and a speed of 130 rpm. The experimental set-up is shown in Figure 20, a mounting hub is connected to the DC motor's shaft. A rubber grommet is then mounted on the mounting hub. A polypropylene string is tied around the rubber grommet. A weight holder is tied at

the bottom of the string with different weights. A tachometer is used to record the angular velocity of the motor while a load is being added.

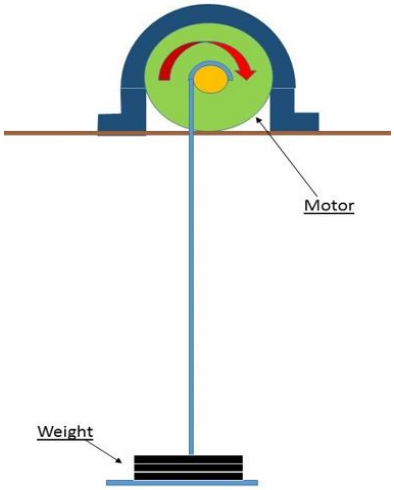

Figure 20: Experimental setup

For each weight, different voltages were supplied to the motor. The current and angular velocity were recorded during the experiment. Torque was calculated using the equation $\tau = F * d$. Where F is the weight added, and d is the radius of the spool plus the rubber grommet. Table 1 shows the data collected during the experiment, the weight and voltage were increased gradually, while the torque and speed of the motor were recorded.

Table 1: Torque, current, and speed data

		8 Volt		10 Volt		12 Volt	
Weight (lb.)	Torque (lb.in)	Current (A)	Speed (rpm)	Current (A)	Speed (rpm)	Current (A)	Speed (rpm)
0.806892	0.476510877	0.3	83.3	0.27	104.6	0.37	126.2
1.356892	0.801313927	0.31	69.7	0.32	102.5	0.42	122.6
1.806892	1.067061877	0.39	76.2	0.38	99.9	0.39	122
2.356892	1.391864927	0.46	74.5	0.44	98.2	0.46	120
2.806892	1.657612877	0.52	71.4	0.54	93.3	0.57	117
3.806892	2.248163877	0.65	67	0.67	90.4	0.72	108.4
4.806892	2.838714877	1.4	52	0.8	80.6	0.94	105.4

Before coming down to the right value of the torque needed, three different motors with different torque values were experimented. The first one electrically failed due to the high load on the motor shaft which drawn high current from the power source. The second one mechanically failed due to the shear stress created on the gears inside the gearbox which broke the gears teeth. The last one was the most suitable motor found to produce the amount of torque and speed needed in the next section. It is a planetary gear motor with stall torque of 680 oz-in and a maximum shaft speed of 165 rpm.

Testing the DC motor proves the theory behind the torque's relationship with the current to be true. Calibrating the torque produced by the motor gives more insight into how much torque will be needed to operate a successful brachiating motion. Through analyzing the collected data, a better understanding on the maximum velocity that the motor can achieve under different loading conditions was obtained. In addition, we verified the right value of the motor's torque selection.

3.6 FABRICATION

Based on our experiment and calculations, the selected motor has a stall torque of 680 oz-in and a shaft speed of 165 rpm. By adding gears at the joint as shown in Figure 8, the speed of the motor is ensured to transmit from one arm to the other. The gears used are made from aluminum and have characteristics of 84 teeth, an outer diameter of 2.688 inches, and a bore of 0.5 inches. The robot is made from aluminum 6063 rather than 6061 because it is lighter and softer in fabrication. With yield strength of 16,000 psi and hardness of 55 Brinel, which give a good combination of high strength and machinable aluminum [64].

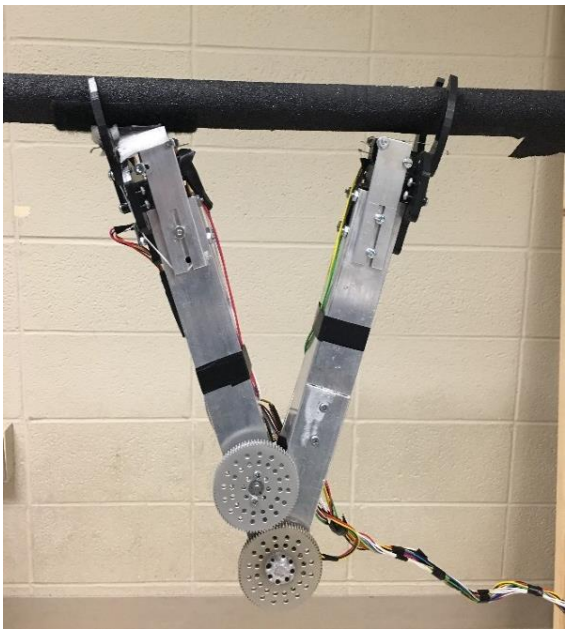

Figure 21: Fabricated mechanism

Shown in Figure 21, the fabricated brachiator in a stationary position holding onto a bar that is wrapped with rubber. The rubber is used to insure a friction surface between the gripper and the bar. Figure 22 shows a full span of the brachiator with one gripper holding onto the bar as part of testing the motor and initial control algorithm.

Figure 22: Full span illustration

Shown in Figure 23 is the fabricated wrist mechanism for changing axes. The mechanism was initially fabricated with polylactide (PLA) by using a three-dimensional printer. However, the material proved to be not strong enough to hold the entire arm and failed after few swings.

Another version of the wrist case can be made of aluminum 6061 with yield strength of 35,000 psi, and tensile strength of 42,000 psi [65]. That will allow the frame to withstand more load while avoiding material fracture or failure. The frame will act as the servo exoskeleton and it will greatly enhance the mechanical loads that the servo and the brachiator can withstand. The designed frame will have repeated pattern throughout the case to allow endless attachment options and ensure arm's security and stiffness while brachiating.

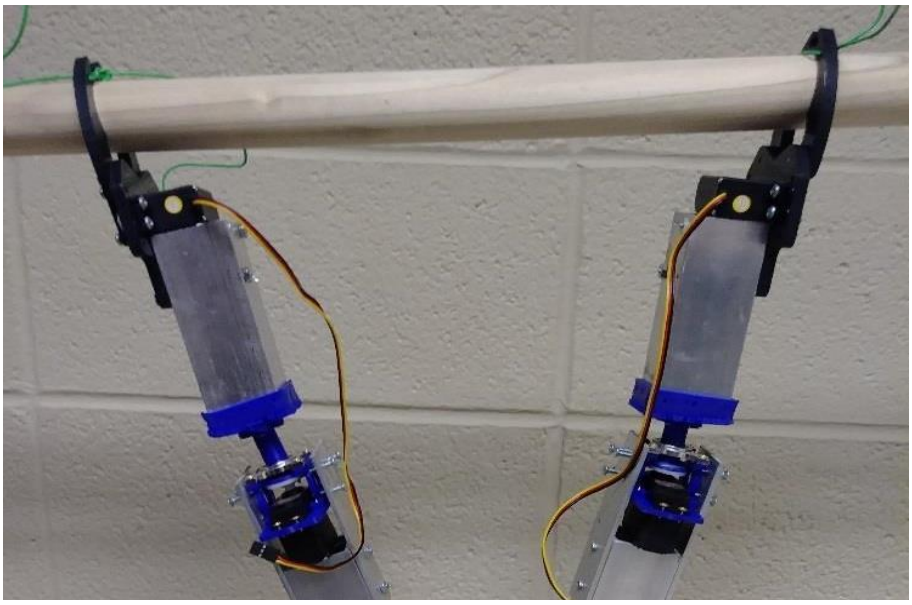

Figure 23: Wrist mechanism

3.7 CONCLUSION

In summary, this paper analyzes the behavior and design of a bio-inspired mechanism that can be mimicked with a two-bar linkage, as a simple way to characterize gibbons swinging motion [66]. With the added wrist mechanism, it allows the brachiator to turn and change axis. We then experiment different actuators until one met the requirements.

Moreover, this research has potential influence in various industries. Especially, it can be used in the maintenance process of power transmission towers. If successful, the usage of the robotic brachiators may help avoid sending humans to elevated levels risking their lives. Another application it can impact is investigating and maintaining bridges. The proposed design has the ability to swing and climb lattice structures where bridges would need inspections regularly.

3.8 LIMITATIONS

The performed experiment was necessary to validate our torque calculations. The motor selection was depending on that experiment; however, a better yet more qualified set-up for the experiment would be recommended to collect more accurate results. For instance, a larger motor spool and rubber gourmet to insure the maintenance of the string's position. A rigid setup is also recommended to protect the motor from moving.

3.9 FUTURE WORK

A brachiator with the ability to swing among a slope line and change axis will highly add value to this research topic. The proposed controller will be implemented and applied to the current model. In addition to that, using a lighter material such as carbon

fiber will greatly impact the mechanism. Moreover, flexible linkages could also replace the aluminum bars, which will revolute the design to a more durable yet more reliable robot to brachiate.

CHAPTER 4: OPTIMIZATION OF A NONLINEAR SYSTEM

4.1 INTRODUCTION

In the past decades, the optimization of a nonlinear process with a set of constraints has been a challenging problem due to the guaranteed convergence difficulty [67]. Many ad-hoc techniques have been combined with traditional optimal controllers to address the challenges with nonlinearity [68]. For example, a neural network is employed to estimate the cost function to establish a region of attraction for nonlinear systems with constrained control inputs in [69]. Particle swarm optimization is usually applied in nonlinear systems with stochastic nature, such as power systems, chemical processes, and financial applications [70]. Sequential quadratic programming techniques were shown to be effective in avoiding singularity in constrained nonlinear control allocation problems [71].

In the context of process control, optimization techniques have proved to be very effective when an optimal solution is desired to minimize a cost function of a process. In particular, optimization of machining operation has shown the economic benefits often via the use of advanced computer-aided, fuzzy logic and genetic algorithms [72]. Topology optimization, a method for enhancing design efficiency with minimal a priori decisions, has been employed extensively to minimize the amount of materials used in additive manufacturing processes [73]. In addition, optimal control is a systematic method commonly used in chemical processes, for example, to find an optimal mixing solution in integrated water-chemical systems [74], to determine the most promising compounds among a large database of potential compounds [75], or to find an optimal evaporation ramp for Bose-Einstein condensates production [76].

In this work, we first revisit the Modified Quasi-linearization Algorithm (MQA) as a method to solve Nonlinear Programming Problems (NLP) and apply it to a chemical reaction optimization problem. The MQA has been initially used to solve nonlinear two-point boundary value problems [77]. Subsequent modifications by Miele and Levy in the early 70's allowed its applications to solve NLP problems [78]. The main property of the MQA method algorithm discussed in this paper is its descent property for the case of linear cost function and quadratic constraint equations. This property guarantees the progressive improvement of the solution as the algorithm searches for the optimum.

We apply the MQA algorithm to a chemical reaction optimization problem. In particular, we want to minimize the Gibbs free energy in a chemical process while satisfying the reagents chemical mass constraints to ensure the feasibility of the reaction. The remaining part of the paper is organized as follow. Section 4.2 describes the theory and structure of the MQA method. Section 4.3 introduces the model of the chemical reaction with identification of cost function, problem parameters, and constraints. Section IV describe the application of MQA and discuss the results. Section V presents some conclusive remarks.

4.2 ALGORITHM FORMULATION

A NLP can be stated as the problem of finding the minimum of a cost function

$$f = f(x), \tag{11}$$

subjected to the constrain

$$\phi(x) = 0, \tag{12}$$

where $x \in \mathbb{R}^n$ is the vector of variables to be optimized, $f: \mathbb{R}^n \to \mathbb{R}$ is the nonlinear function to be minimized, $\phi: \mathbb{R}^n \to \mathbb{R}^q$ is a nonlinear vector function representative of the set of constraints to be satisfied. For the previous problem to be a bonafide optimization problem the Number of Degrees of Freedom NDF $= n - q$ must be greater than zero. In fact, if $n = q$ the problem admits only the solution of the nonlinear equation (11). It is important to note that the constraint Eq. (12) being an equality constraint does not reduce the generality of the method that we will discuss in this paper, in fact a scalar inequality constraint equation of the form

$$\phi_i(x) \geq 0, \tag{13}$$

can be converted into an equality constraint by introducing a slack variable s_i

$$\phi_i(x) - s_i^2 = 0 \tag{14}$$

With this in mind, and without loss of generality, we will consider NLP with equality constraints.

The first order conditions for an extremal of a problem identified by Eqs. (11) and (12) are [78]:

$$F_x(x, \lambda) = 0 \tag{15}$$

$$\phi(x) = 0 \tag{16}$$

with

$$F(x, \lambda) = f(x) + \lambda^T \phi(x) \tag{17}$$

the augmented function and $\lambda \in \mathbb{R}^q$ the Lagrange multiplier vector.

The MQA algorithm is based on the following performance indexes:

$$P = \phi^T \phi \tag{18}$$

$$Q = F_x^T F_x. \tag{19}$$

Notably, P is the constraint error and Q is optimality error. In the numerical search for the solution of the NLP convergence can be declared when the following conditions are achieved:

$$P \leq \varepsilon_1 \tag{20}$$

$$Q \leq \varepsilon_2 \tag{21}$$

with ε_1 and ε_2 prescribed small tolerances. Whenever condition (20) is satisfied the candidate solution is a feasible solution. If both (20) and (21) are satisfied, the feasible solution is also optimal. The total error is defined as the sum of the constraint error and the optimality error

$$R = P + Q = \phi^T \phi + F_x^T F_x \tag{22}$$

In turn the convergence condition can be represented as

$$R \leq \varepsilon_3 \tag{23}$$

with ε_3 a prescribed total error tolerance. For our application, we have determined, through a series of convergence analysis, that satisfactory results are obtained by choosing $\varepsilon_1 = 10^{-9}$, $\varepsilon_2 = 10^{-4}$, $\varepsilon_3 = 10^{-4}$.

In order to track the change in each variable introduced above, we define nominal points for each one such that, any change occurs from the nominal function to the varied function will cause the performance index P to change. Varied point $\tilde{x}$ can be obtained by using first order Taylor approximations of nominal points:

$$\tilde{x} = x + \Delta x$$

$$\tilde{\lambda} = \lambda + \Delta\,\lambda$$

$$\tilde{P} = P + \Delta P$$

$$\tilde{Q} = Q + \Delta Q$$

$$\tilde{R} = R + \Delta R$$

where Δx denotes the perturbation from x. For optimization, the performance indexes $P, Q,$ and R need to be maintained as small as desired. Therefore, the convergence conditions are,

$$P \leq \epsilon_1, Q \leq \epsilon_2, and\ R \leq \epsilon_3 \tag{24}$$

where $\epsilon_1, \epsilon_2, \epsilon_3$ are the tolerances.

The ultimate goal now is that at every iteration, the total error R, also can be referred to as δR, should meet the following conditions:

$$\delta R < 0$$

$$\Delta R < 0 \tag{25}$$

Therefore,

$$\tilde{R} < R \tag{26}$$

To achieve the conditions in Eqs. (25) and (26), we apply Taylor expansion on ΔR such that,

$$\Delta R = \delta R + \frac{1}{2}\delta^2 R + H.O.T$$

$$\Delta R \cong \delta R < 0 \tag{27}$$

Substituting R from Equation (15), we get:

$$\delta R = \delta P + \delta Q$$

$$\delta R = \delta(\phi^T \phi) + \delta(F_x^T F_x)$$

$$\delta R = 2\phi^T \delta\phi + 2F_x^T \delta F_x \tag{28}$$

The selection for Δx and $\Delta\lambda$ should be considered such that:

$$\delta\phi = -\alpha\phi \tag{29}$$

$$\delta F_x = -\alpha F_x \tag{30}$$

Now, substituting $\delta\phi$ and δF_x from Eqs. (29) and (30) into δR from Eq. (28) will result in:

$$\delta R = 2\phi^T(-\alpha\phi) + 2F_x^T(-\alpha F_x)$$

$$\delta R = -2\alpha(\phi^T \phi + F_x^T F_x)$$

$$\delta R = -2\alpha R \tag{31}$$

Therefore, if alpha is sufficiently small such that $\alpha \ll 1$, then our convergence condition $\delta R < 0$ will occur. Therefore, the conditions of Eq. (27), that is $\Delta R \approx \delta R < 0$, $\tilde{R} < R$, will be achieved, where again, $R = P + Q \geq 0$.

We now introduce a set of linear equations to simplify the change in the augmented function and the constraint, such that,

Since, $$\delta F_x = -\alpha F_x \tag{32}$$

$$\delta\phi = -\alpha\phi \tag{33}$$

Therefore, solving for δF_x and $\delta\phi$ results in,

$$\delta F_x = F_{xx}\Delta x + F_{x\lambda}\Delta\lambda = -\alpha F_x \tag{34}$$

$$\delta\phi = \phi_x^T \Delta x + \phi_\lambda \Delta\lambda = -\alpha\phi \tag{35}$$

where

$$F_{x_\lambda} = \frac{\partial F_x}{\partial \lambda} = \frac{\partial}{\partial \lambda}(f_x + \phi_x \lambda)$$

$$F_{x_\lambda} = \phi_x \tag{36}$$

$$\phi_\lambda = \frac{\partial \phi}{\partial \lambda} = 0. \tag{37}$$

The set of Eqs. (34) and (35) then becomes,

$$F_{xx}\Delta x + \phi_x \Delta\lambda = -\alpha F_x \tag{38}$$

$$\phi_x^T \Delta x + 0 = -\alpha\phi \tag{39}$$

where $\phi_x \in \mathbb{R}^{n*q}, \phi_x^T \in \mathbb{R}^{q*n}$, and $\alpha \in \mathbb{R}^{q*1}$. We next introduce auxiliary variables A and B such that,

$$A = \frac{\Delta x}{\alpha} \tag{40}$$

$$B = \frac{\Delta\lambda}{\alpha} \tag{41}$$

After substituting A and B, Eqs. (38) and (39) then becomes

$$\begin{bmatrix} F_{xx} & \phi_x \\ \phi_x^T & 0 \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = -\begin{bmatrix} F_x \\ \phi \end{bmatrix}. \tag{42}$$

The differential system in (42) is linear and nonhomogeneous and can be solved without assigning value to the step size α. Furthermore, expression in (42) resembles the bordered hessian matrix.

4.3 ALGORITHM APPLICATION

The discussed algorithm can be used in various applications where optimal control is required. In this section, we consider a mixture of chemical species held at a constant temperature and pressure. The mixture reaches its chemical equilibrium state simultaneously with reduction of the free energy of the mixture to a minimum [79]. This is a consequence of the second law of thermodynamics [80]

$$\Delta G(p,T) = \Delta H - T * S \tag{43}$$

where ΔG is the Gibbs free energy, ΔH is the internal energy, T is the temperature, and S is the entropy. To achieve a chemical equilibrium, the objective function to be minimized is the expression of the free energy of the chemical mixture ΔG [81].

A chemical equilibrium problem can be defined as a mathematical programming problem as in [82], with linear mass balance constraints that represents the possible chemical combinations of the chemical components of the mixture, and a nonlinear objective function representing the free energy of the mixture.

4.4 CHEMICAL MODEL

According to the mathematical model formulated in [82], a mixture consists of m chemical elements. It has been prearranged that the m different types of atoms can combine chemically to produce n compounds, where the monotonic atom is regarded for our purpose as a possible compound. Therefore,

x_j = the number of moles of compound j present in the mixture at equilibrium,

$\bar{x}$ = the total number of moles in the mixture, where $\bar{x} = \sum_{j=1}^{n} x_j$

a_{ij} = the number of atoms of element i in a molecule of compound

b_i = the number of atomic weights of element in the mixture.

The mass balance relationships that must be considered for the in elements are

$$\sum_{j=1}^{n} a_{ij}\, x_j = b_i, \qquad i = 1, \dots, m \tag{44}$$

And

$$x_j \geq 0, \quad j = 1, \dots, n \tag{45}$$

Equations (44) and (45) will be used later in this section to develop the linear constraints. However, to determine the composition of the mixture at equilibrium state, values of $x_j (j = 1, \dots, m)$ that satisfy (44) and (45) must be obtained. The total free energy of the mixture is given by

$$\sum_{j=1}^{n} x_j \left[c_j + \ln\left(\frac{x_j}{\bar{x}}\right) \right] \tag{46}$$

Where

$$c_j = \left(\frac{F^\circ}{RT}\right)_j + \ln(P) \tag{47}$$

where $\left(\frac{F^\circ}{RT}\right)$ is the modal standard (Gibbs) free energy function for the j^{th} compound, and P is the total pressure in the atmosphere.

Therefore, the nonlinear programming problem then becomes: determine the values of a given component x_j $(j = 1, \dots, n)$ to minimize the nonlinear objective function (46) that's is subjected to the linear constraints (44) and non-negativity restrictions in (45).

The problem considered in this section is the determination of the equilibrium composition resulting from subjecting the compound

$$\frac{1}{2}N_2H_4 + \frac{1}{2}O_2 \tag{48}$$

to a temperature of $3500°K$ and a pressure of $750\ psi$.

Table 2 shows the 10 possible compounds, the Gibbs free energy function $\left(\frac{F°}{RT}\right)_j$, as well as the computed values of c_j. The number of atomic weights of H, N, and O in the chemical mixture in (48) are assumed to be $b_1 = 2, b_2 = 1, \text{and } b_3 = 1$.

Table 2: Mixture data at 3500 K and 750 psi

j	$Compund$	$\left(\frac{F°}{RT}\right)_j$	c_j
1	H	-10.021	-6.089
2	H_2	-21.096	-17.164
3	H_2O	-37.986	-34.054
4	N	-9.846	-5.914
5	N_2	-28.653	-24.721
6	NH	-18.918	-14.986
7	NO	-28.032	-24.100
8	O	-14.640	-10.708
9	O_2	-30.594	-26.662
10	OH	-26.111	-22.179

To formulate the nonlinear programming model, based on (46), the nonlinear objective function to be minimized is

$$f(x_1,..,x_{10}) = x_1\left[-6.089 + \ln\left(\frac{x_1}{\sum_{j=1}^{n} x_j}\right)\right] + x_2\left[-17.164 + \ln\left(\frac{x_2}{\sum_{j=1}^{n} x_j}\right)\right] + \cdots$$

$$+ x_{10}\left[-22.179 + \ln\left(\frac{x_{10}}{\sum_{j=1}^{n} x_j}\right)\right] \tag{49}$$

And the linear constraints ϕ_q of the nonlinear programming problem are:

$$\left.\begin{aligned}\phi_1 &= x_1 + 2x_2 + 2x_3 + x_6 + x_{10} - 2 = 0\\ \phi_2 &= x_4 + 2x_5 + x_6 + x_7 - 1 = 0\\ \phi_3 &= x_3 + x_7 + x_8 + 2x_9 + x_{10} - 1 = 0\end{aligned}\right\} \quad (50)$$

where: $x_1 \geq 0, \dots, x_{10} \geq 0$

4.5 QUASILINEARIZATION ALGORITHM IMPLEMENTATION

As discussed in the previous section, in order to optimize the proposed problem using the quasi-linearization algorithm, an augmented function that is equivalent to equation (49) is introduced. $F = f(x_1, \dots, x_{10}) + \lambda_1^T \phi_1 + \lambda_2^T \phi_2 + \lambda_3^T \phi_3$ where $\lambda_1, \lambda_2, \lambda_3$ are Lagrange multipliers satisfying the number of constraints. To develop the algorithm, a performance index is created as following:

$$P = \phi^T \phi$$

$$Q = F_x^T F_x$$

$$R = P + R \quad (51)$$

Where P is the cumulative error, Q is the optimality condition error, and R is the overall error.

In our design, we desire that at every iteration $\tilde{R} < R \ and \ \Delta R < 0$. To achieve this statement, first order Taylor series is performed on ΔR as following

$$\Delta R = \delta R + \frac{1}{2}\delta^2 R + H.O.T. \cong \delta R \quad (52)$$

In addition, the tolerances $\epsilon_1, \epsilon_2, \epsilon_3$ are set to be $10^{-8}, 10^{-4}, 10^{-4}$, respectively.

4.6 PSEUDO CODE

In Summary, we can summarize the MQA in few steps. The following pseudo code describes all the steps that were used in the approach and executed in MATLAB R2018a environment.

After the problem and constraints are defined, the algorithm can follow:

Step 0.

a) Start by defining the initial nominal points x, λ

b) Evaluate ϕ, F_x at the nominal points to get $P, Q, and\ R$

Step 1.

a) Calculate $F_{xx}\ and\ \phi_{xx}$

b) Solve System of Equations (26) to get A and B

$$\begin{bmatrix} F_{xx} & \phi_x \\ \phi_x^T & 0 \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = - \begin{bmatrix} F_x \\ \phi \end{bmatrix}$$

Step 2.

a) Propose $\tilde{x} = x + \alpha A$ and $\tilde{\lambda} = \lambda + \alpha B$

b) Select α by bisecting from $\alpha = 1$ such that $\tilde{R}(\alpha) < \tilde{R}(0) = R$

4.7 RESULTS

Depending on the initial guess for the values of x (the molecular mass) the algorithm will keep iterating until an optimal point for each value is reached. After trial and error, and to minimize the iterations, the following nominal points were considered,

$$x_0 = [0.07\ 0.3\ 0.9\ 0.002\ 0.5\ 0.0009\ 0.03\ 0.02\ 0.05\ 0.10] \qquad (53)$$

After three iterations, the following results in Table 3 and Table 4 were obtained:

Table 3: At First Iteration

j	Compound	x_j
1	H	0.0338
2	H_2	0.2116
3	H_2O	0.7177
4	N	0.0014
5	N_2	0.4815
6	NH	0.0006
7	NO	0.0351
8	O	0.0233
9	O_2	0.0585
10	OH	0.1070

Table 4: At Last Iteration

j	Compound	x_j
1	H	0.0407
2	H_2	0.1477
3	H_2O	0.7833
4	N	0.0014
5	N_2	0.4852
6	NH	0.0007
7	NO	0.0274
8	O	0.0180
9	O_2	0.0373
10	OH	0.0966

Due to the small number of iterations, which in this case three interactions, the minimum values of each compound of x_j are found to be close to each other. A two-dimensional plot is shown below in Figure 24 to visualize the results.

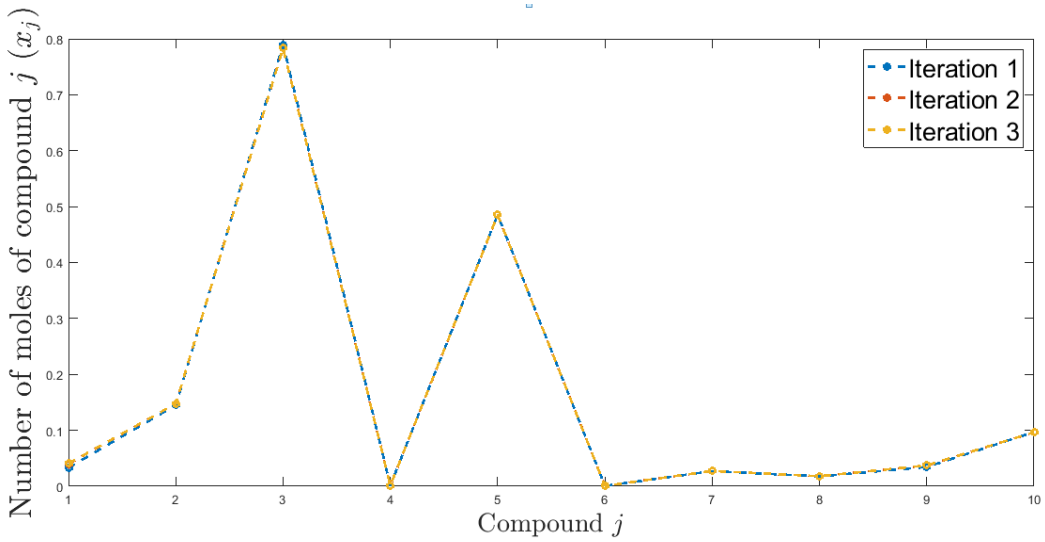

Figure 24: Explanation of the convergence of each element to its minimum values

As can be seen in Figure 24, solution converges within three iterations. For every element in the chemical mixture, the number of moles required for the reaction have been reduced after optimizing the objective function (49) with respect to the set of constraints in (50). To achieve chemical equilibrium and minimum Gibbs free energy, the following relationship must be accomplished

$$\Delta G = G_{products} - G_{reactants} < 0$$

And that is to say, $G_{product}$ must be negative. Therefore, based on the approach taken in this chapter, and after substituting the optimized values of x_j obtained from Table 4 into Eq. (49), $f(x_1,..,x_{10})$ will result in a $G_{product}$ of -47.7580. The values achieved are very similar and agrees to the values obtained by Dantzig in [82].

4.8 CONCLUSION

The modified quasi-linearization algorithm can be used in many optimization problems. In this paper, our objective was to minimize the Gibbs free energy coming out

of the mixture to reduce the amount of energy lost and achieve equilibrium concentration. The problem mostly arises in the implementation and analysis of the performance of fuels and propellants that is used in aerospace propulsion. The used algorithm has proven to be reliable and compatible to many other optimization methods currently used. And that validates the efficiency and accuracy of the MQA.

CHAPTER 5: A MULTI-LAYER PERCEPTRON NEURAL NETWORK WITH BACKPROPAGATION ALGORITHM

5.1. INTRODUCTION

In the past decade, computation has adopted the neural approach to tackle problems for which most conventional computational approaches have proved unsuccessful [83]. Such problems arise when a computer is asked to interface with the real world, for instance, the difficulty of visual pattern recognition becomes apparent when a computer program is asked to recognize digits which is difficult because the real world cannot be modeled with concise mathematical expressions. Problems of this type are pattern recognition, image processing, and machine vision [84]. Not coincidentally perhaps, these are precisely the types of problems that humans execute seemingly without effort. Therefore, people interested in building computers to tackle these problems have tried to adopt the existing understanding on how the brain computes. Neural networks (NNs) can learn and adapt themselves to inputs from the actual processes allowing representation of complex engineering systems, which are difficult to model either with traditional model-based engineering or knowledge based expert systems [85].

NN model relies on several very simple neuron-like processing elements locally interacting through a set of unidirectional weighted connections. Knowledge is internally represented by the values of the weights and topology of the connections. Learning involves modifying the connection weights [86]. There are two types of learning methods used in neural network: supervised and unsupervised learning [87]. In this chapter, the supervised learning method is investigated to design the NN where the output goal is to achieve the target by using back propagation learning algorithm. The backpropagation algorithm (BA) is used to learn the weights of a multilayer neural network with a fixed

architecture. It is therefore usually considered to be a supervised learning method, although it is also used in some unsupervised networks such as autoencoders [88]. The BA performs gradient descent to try to minimize the sum squared error between the networks output values and the given target values.

The rest of this chapter is organized as follows. The mathematical problem and neural network model formulation to solve the given problem are presented in Section 5.2. Section 5.3 explains the backpropagation algorithm used. The results and simulations are shown in Section 5.4. Finally, the conclusion of the work done is presented in Section 5.5.

5.2 MATHEMATICAL PROBLEM AND NEURAL NETWORK MODEL

The problem needs to be learned is shown below in Table 5, where the network must recognize the input and produce the exact values in the output. The input is a set of eight binary numbers that goes from one to eight consequently. Each input is assigned in a vector x_i as follows:

Table 5: Problem to be solved where x_i is the binary input

x_i	*INPUT*	*DESIRED OUTPUT*
x_1	10000000	10000000
x_2	01000000	01000000
x_3	00100000	00100000
x_4	00010000	00010000
x_5	00001000	00001000
x_6	00000100	00000100
x_7	00000010	00000010
x_8	00000001	00000001

The network used is eight-three-eight layers, where the hidden layer consists of three perceptron, and the input and output consist of eight perceptron where $x_i \in [x_1, x_8]$. The architecture of the neural network is shown in the following figure 25

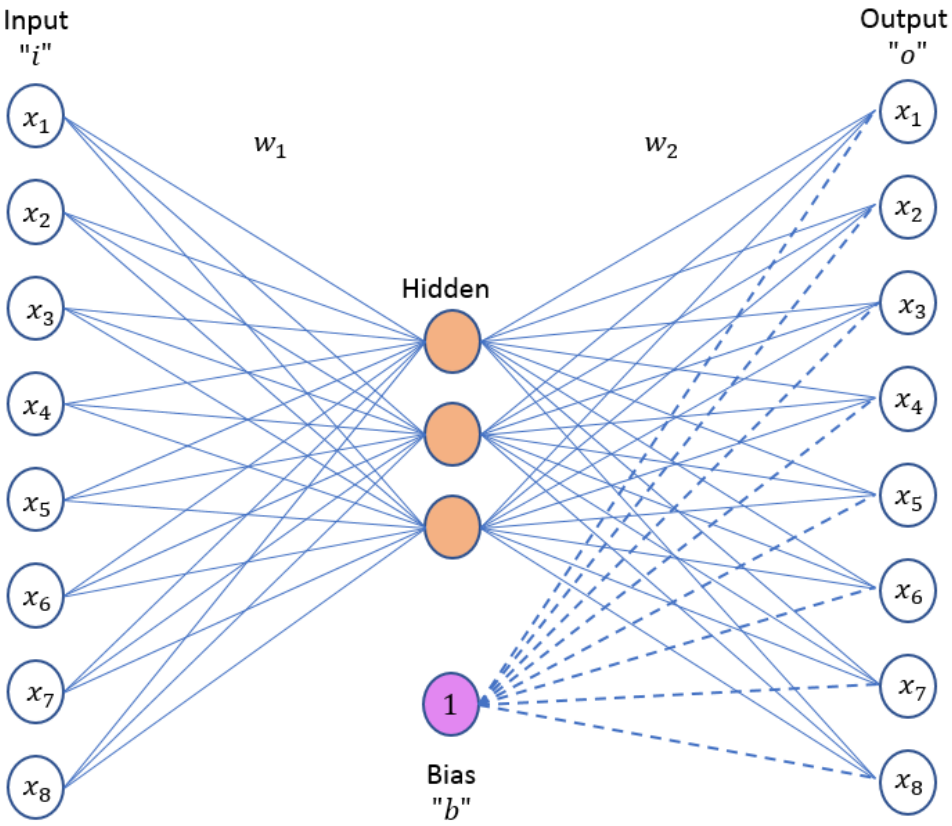

Figure 25: The used Neural Network model with eight inputs, three hidden, and eight output layers

The input neurons are all multiplied by a weight w_1 depending on the size of the input and are connected to the hidden layer. The hidden layer introduces the incoming values to a sigmoid function. The output of the hidden layer is then multiplied by a weight w_2 and stores the values in the output neurons along with a constant bias of 1 whose value remains constant throughout the training and is added in the hidden layer. The bias is also connected to all the output neurons.

This is called feedforward propagation, where an input pattern w_i is applied to the input layer and its effect propagates, layer by layer, throughout the network until an output is produced. In the feedforward propagation, the value of the hidden and output neurons is updated using sigmoid function as shown in Figure 26 below,

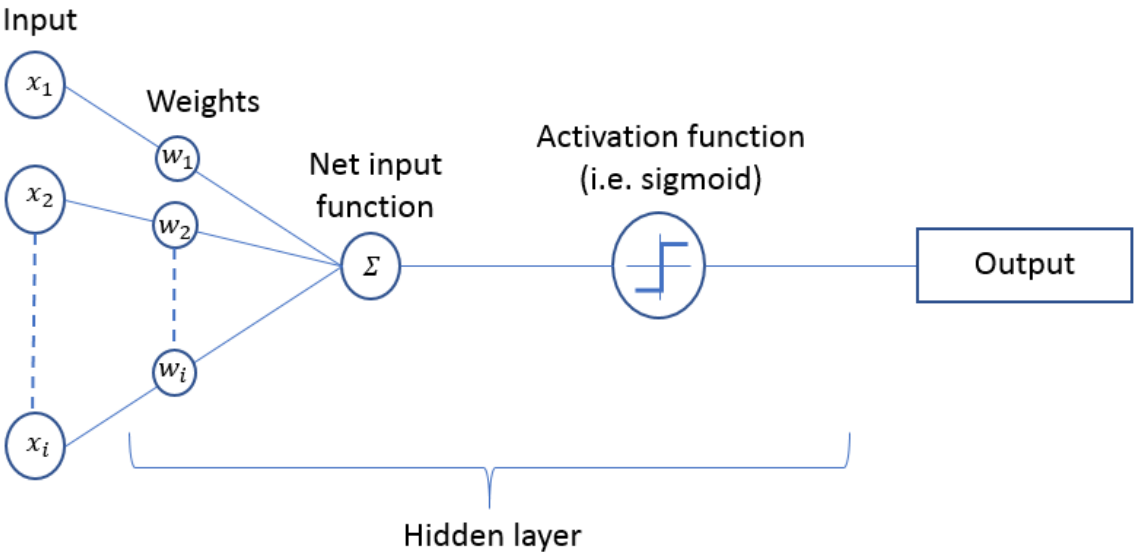

Figure 26: Feedforward propagation illustration

In general, for each of the incoming connections, a weight value w_i will be assigned for each input x_i. When the network is active, the node receives the value x_i and multiplies it by the associated weight. It then adds the resulting products together, yielding a single number, known as the net input function. It can be expressed as

$$\sum_{n=1}^{i} w_n x_n \tag{54}$$

Where i is the number of inputs to the perceptron, w is a vector of real-valued weights, and x is a vector of the input values. The neuron's output, 0 or 1, is determined by whether the weighted sum in equation (54) is less than or greater than a threshold value. The threshold is a real number which is a parameter of the neuron and can be expressed by,

$$output\ of\ the\ net\ input\ function\ f(x) = \begin{cases} 1, & if \sum_{n=1}^{i} w_n x_n > threshold \\ 0, & if \sum_{n=1}^{i} w_n x_n \leq threshold \end{cases} \tag{55}$$

By moving the threshold to the other side of the inequality, it can be replaced by what's known as the perceptron's bias, $b \equiv -$ threshold. Using the bias instead of the threshold, the perceptron rule can be rewritten as,

$$\Sigma f(x) = \begin{cases} 1, & if\ w * x + b > 0 \\ 0, & otherwise \end{cases} \tag{56}$$

bias is a measure of how easy it is to get the perceptron to output a value of 1. A perceptron with a positive bias, can easily output a value of 1. However, if the bias is negative, then it's challenging for the perceptron to output a value of 1.

The net input function then introduces its value to an activation function (i.e. sigmoid function). If the outcome number is below a threshold value, the node passes no data to the next layer. If the number exceeds the threshold value, the node sends the number (the sum of the weighted inputs) to all its output connections.

$$\sigma(net) = \frac{1}{1 + e^{-net}} \tag{57}$$

To put the output of the sigmoid neuron in one explicit equation the with inputs $x_1, x_2, \dots, x_i$ and weights $w_1, w_2, \dots, w_i$ and bias b. The equation then becomes

$$\sigma(net) = \frac{1}{1 + \exp(-\sum_{n=1}^{i} w_n x_n - b)} \tag{58}$$

The sigmoid function is one of the most popular functions in neural network because it satisfies a property between the derivative and itself such that it is computationally easy to perform. The sigmoid function takes the weighted sum net as

the input of the values coming from the units connected to it and produce $\sigma(net)$. The same process is carried between the hidden layer and the final output layer.

5.3 BACKPROPAGATION ALGORITHM

In backpropagation, the first step is to calculate the error produced by the activation function to eventually update the weight as shown in figure 27.

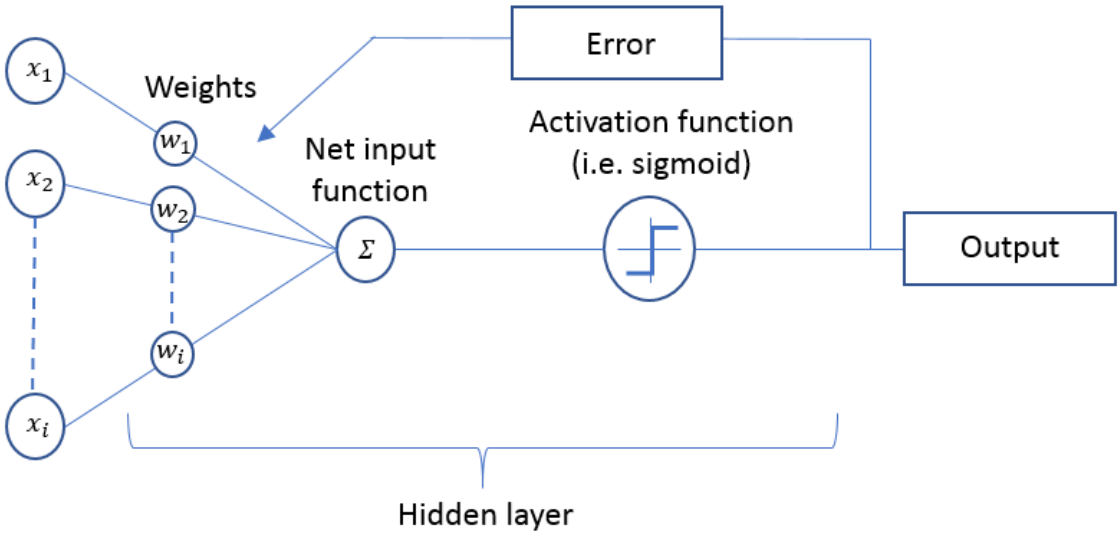

Figure 27: Feedforward propagation after obtaining the error

An expected output needs to be defined into the system and then the error is determined by comparing the output of the network with the expected output via the squared error formula as

$$E[\vec{w}] = \frac{1}{2}\sum(t_k - o_k)^2 \tag{59}$$

Where t_k is the desired target value, o_k is the perceptron output from the hidden layer at any output instance k. $o_k = w_0 + w_1x_1 + \cdots + w_ix_i$. The error terms of the output units are sometimes written as,

$$\delta_k = o_k(1 - o_k)(t_k - o_k)$$

The next step now is to update the weights using gradient descent so that they can cause the predicted output to be close to actual output. Therefore, minimizing the error for each output neuron and getting closer to the desired target. The gradient descent can be expressed as

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial o_k}\frac{\partial o_k}{\partial net_k}\frac{\partial net_k}{\partial w_i} \tag{60}$$

$$\nabla E[\vec{w}] = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_i}\right]$$

$$\frac{\partial E}{\partial w_i} = \frac{\partial}{\partial w_i}\left(\frac{1}{2}\sum(t_k - o_k)^2\right)$$

$$= \frac{1}{2}\sum\frac{\partial}{\partial w_i}(t_k - o_k)^2$$

$$= \frac{1}{2}\sum 2(t_k - o_k)\frac{\partial}{\partial w_i}(t_k - o_k)$$

$$= \sum(t_k - o_k)\left(-\frac{\partial o_k}{\partial w_i}\right)$$

$$= \sum(t_k - o_k)\frac{\partial o_k}{\partial net_k}\frac{\partial net_k}{\partial w_i} \tag{61}$$

The derivative of the sigmoid function $\sigma(z)$ is equal to

$$\frac{\partial o_k}{\partial net_k} = \frac{\partial \sigma(net_k)}{\partial net_k} \tag{62}$$

From Equation (57). Since $\sigma(net) = \frac{1}{1+e^{-net}} = \frac{e^{net}}{1+e^{net}}$

Therefore,

$$\frac{\partial \sigma(net)}{\partial net} = \frac{e^{net}(1+e^{net}) - e^{net}e^{net}}{(1+e^{net})^2}$$

$$\frac{\partial \sigma(net)}{\partial net} = \frac{e^{net}}{(1+e^{net})^2} = \sigma(net)(1-\sigma(net)) \tag{63}$$

$$\frac{\partial net_k}{\partial w_i} = \frac{\partial(\vec{w}\cdot\vec{x})}{\partial w_i} = x_{i,k} \tag{64}$$

Substituting equations (61), (63), (64) in (60) yields to

$$\frac{\partial E}{\partial w_i} = \sigma(net)(1-\sigma(net))\sum(t_k - o_k)x_{i,k} \tag{65}$$

Therefore, equation (65) can be rewritten to calculate the error terms from the output units that are used to calculate error terms for the hidden units as,

$$\delta_z = o_z(1-o_z)\sum\delta_k w_2^T \tag{66}$$

where, δ_k represents the output error term for any output instance k and o_z represents the value for any hidden instance unit z that is found in feedforward step. After calculating all the errors values associated with each unit (hidden and output), we can now transfer this information into the weight changes Δ_{ij} between units i and j. For weights w_1 between input unit I_i and hidden unit h_j, the alteration Δ_{ij} can be written as,

$$\Delta_{ij} = \eta\delta_{h_j}x_i \tag{67}$$

where, η is a small constant (learning rate e.g. 1), x_i is the input to the i-th input node and δ_{h_j} is the error term to the j-th hidden node. For weights w_2 between hidden unit h_i and output unit o_j, the alteration can be written as,

$$\Delta_{ij} = \eta \delta_{oj} h_i \quad (68)$$

where, h_i is the hidden value to the i-th hidden node and δ_{oj} is the error term to the j-th output node. Each alteration Δ is added to the weights to update the weight vectors. Finally, the weight that minimizes error function is then considered to be solution. The backpropagation algorithm pseudo code is shown in the next section.

5.4 PSEUDO CODE

With the given desired output and input in Table 5, and backpropagation algorithm was designed and run in MATLAB2018a environment as shown in the following table. The error tolerance was kept as 0.01, a learning rate was set as 0.5.

THE BACKPROPAGATION ALGORITHM

Step 0.

a. Initialize weights w_1 and w_2, and learning rate η = 0.5.
b. Set input vector (x) and target vector (t).
c. Set maximum number of iterations (e.g. 3000), and tolerance = 0.01.

While iteration<maximum number of iteration or error>tolerance

Step 1.

a. Calculate hidden activations using sigmoid function.
b. Calculate output activations using sigmoid function.

Step 2.

a. Compute error for each network output unit k:

$$\delta_k = o_k(1 - o_k)(t_k - o_k)$$

b. Compute error for each network hidden unit h:

$$\delta_h = o_h(1 - o_h)\sum \delta_k w_2^T$$

c. Update each network weights:

$$w_1 = w_1 + \eta * \delta_h * x$$
$$w_2 = w_2 + \eta * \delta_o * h$$

end

A constant bias value of 1 was added in hidden layer and maximum number of iterations was set to 3000.

5.6 RESULTS

The output results after training the neural network are shown in Table 6. Notice that the diagonal values are close to value of one.

Table 6: The obtained results with x_i, o is the diagonal values

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
0.9107	0.0027	0.0602	0.0000	0.0002	0.0800	0.0027	0.0000
0.0613	**0.9095**	0.0109	0.0624	0.0541	0.0011	0.0000	0.0003
0.0492	0.0007	**0.9298**	0.0373	0.0000	0.0001	0.0418	0.0000
0.0000	0.0053	0.0615	**0.9285**	0.0001	0.0000	0.0033	0.0564
0.0009	0.0457	0.0000	0.0002	**0.9202**	0.0539	0.0001	0.0527
0.0456	0.0000	0.0000	0.0000	0.0615	**0.9047**	0.0538	0.0010
0.0021	0.0000	0.0455	0.0004	0.0000	0.0400	**0.9235**	0.0350
0.0000	0.0001	0.0000	0.0349	0.0351	0.0020	0.0366	**0.9400**

It is found that the error tolerance was less than defined threshold and after 2221 iterations, that means the simulation does not need to run all way to defined 3000 iterations. Figure 28 shows the convergence of error for all eight inputs. The sum of squared errors for each output unit is found by

$$E = \frac{1}{2}\sum(t_k - o_k)^2$$

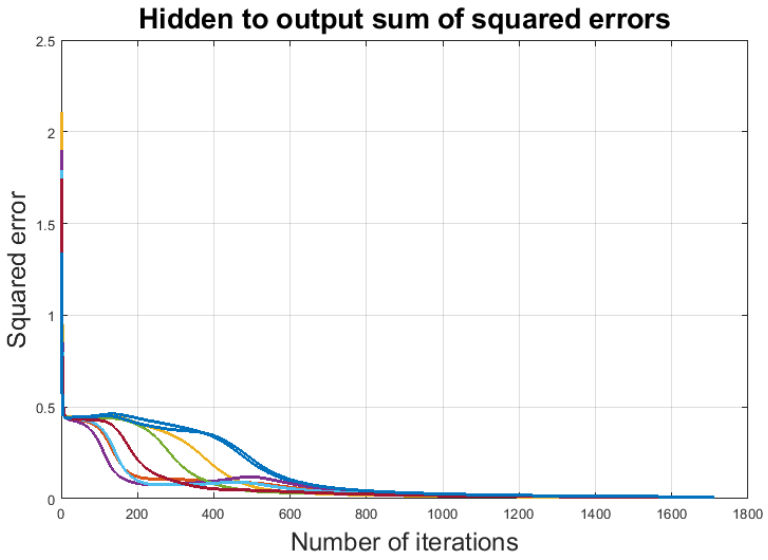

Figure 28: Sum of squared errors

Figure 29 shows the hidden unit encoding for third input. The three hidden unit values for that input [00100000] after 2221 iterations.

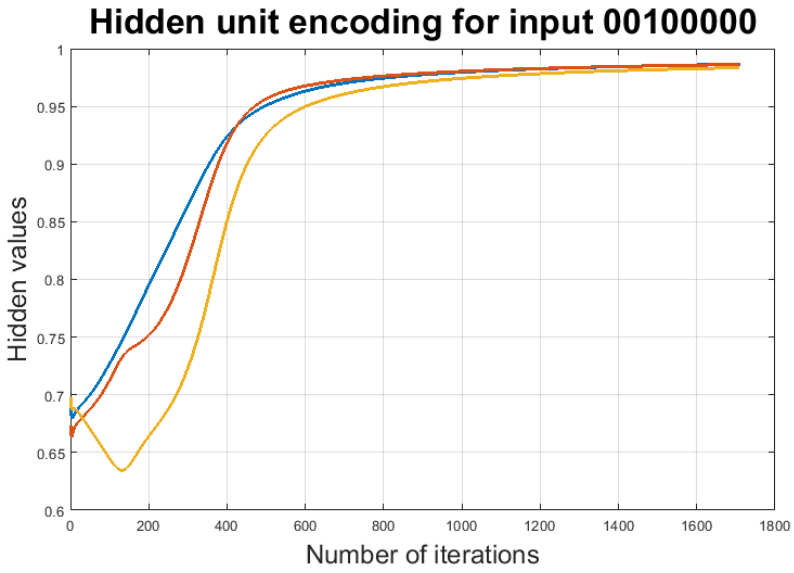

Figure 29: Three hidden unit values for third input

Figure 30 shows the weights from input layer to second hidden unit to observe how the randomly initialized weights were converged to final value.

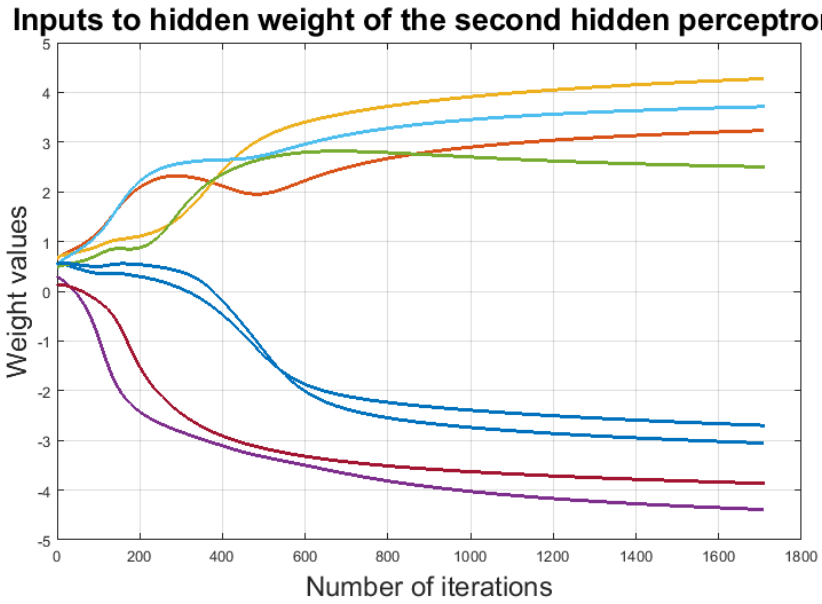

Figure 30: Weights from input to second hidden unit

5.5 CONCLUSION

In this chapter a neural network multi-layers perceptron is implemented and explained to solve a binary problem. The back-propagation algorithm is investigated to train and update the weight from input-to-hidden and from hidden-to-output. Gradient descent of the error is derived and used in the back-propagation algorithm. During the simulation, it was observed that all the errors are not always going to be converged although the iterations were enough. Therefore, a constant bias is added to the hidden layer of the system to ensure convergence.

CHAPTER 6: DESIGN PORTFOLIO

6.1 MATERIAL-HANDLING MANIPULATOR

A ceiling material-handling manipulator with a variable stiffness joint is shown below in Figure 31. This flexible joint can be used in many fields, for instance, agriculture, production lines, and warehouses.

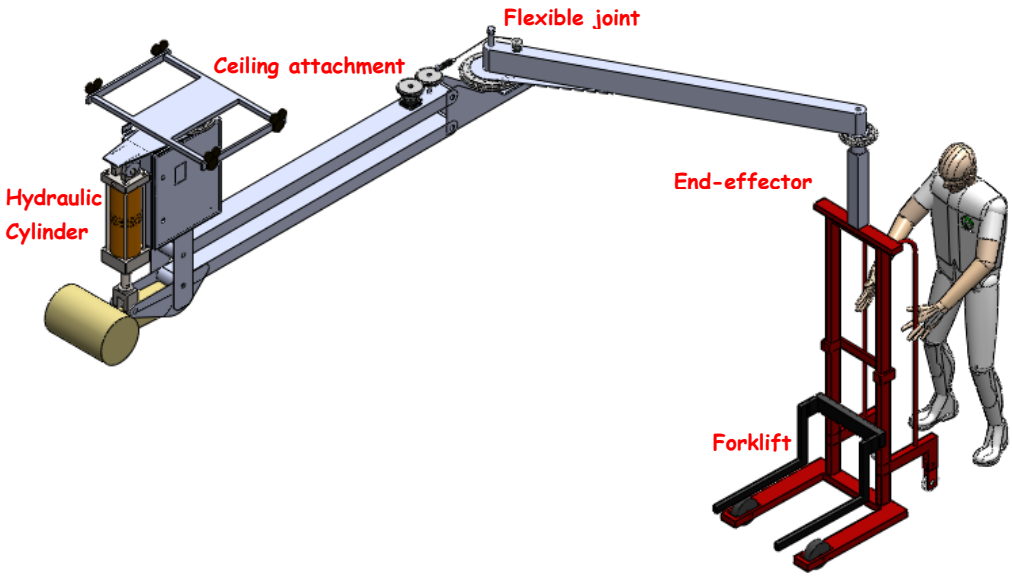

Figure 31: material-handling manipulator

Shown in Figure 32 is a closer look at the flexible joint, i.e. variable stiffness joint.

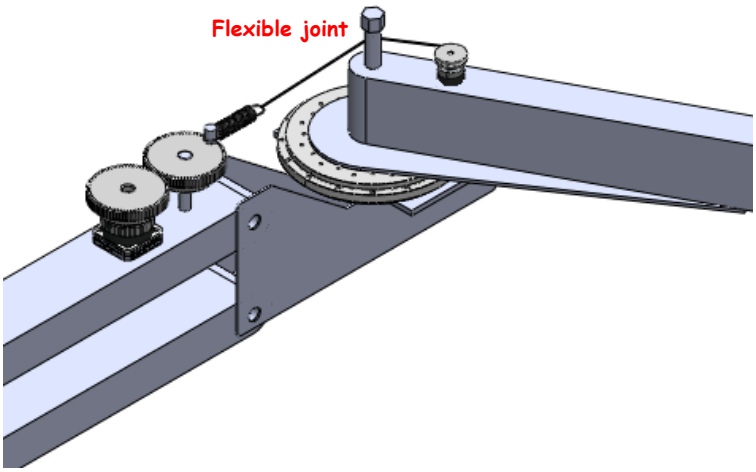

Figure 32: Material-handling flexible joint illustration

The material-handling manipulator is not only limited to ceiling attachments but can also be reconfigured for mobility and access to fields or plants by attaching it to a mobile robot or a pick-up truck, for example, as shown in Figure 33.

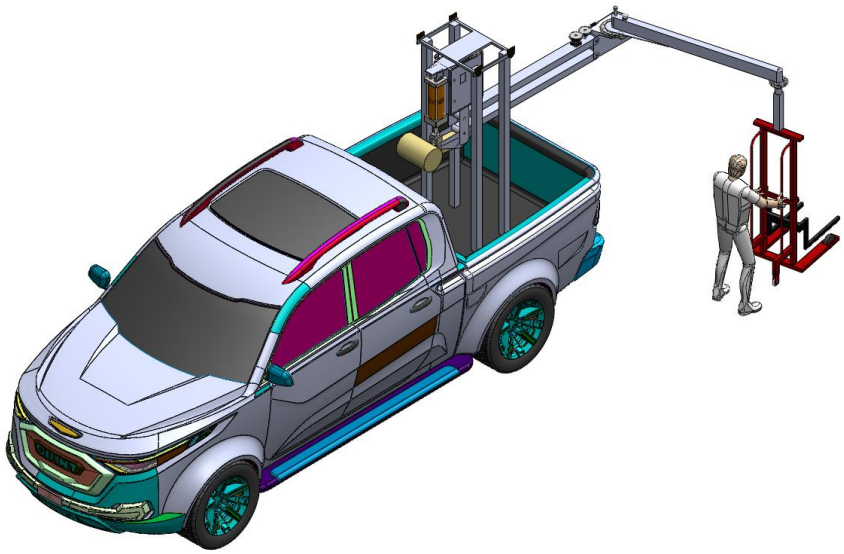

Figure 33: Material-handling manipulator on a pick-up truck

The flexibility of the joint and wide range of motion can allow the end-effector and the operator move around the field as desired. Shown in Figure 34 and Figure 35 is the material-handling manipulator with a different working angle.

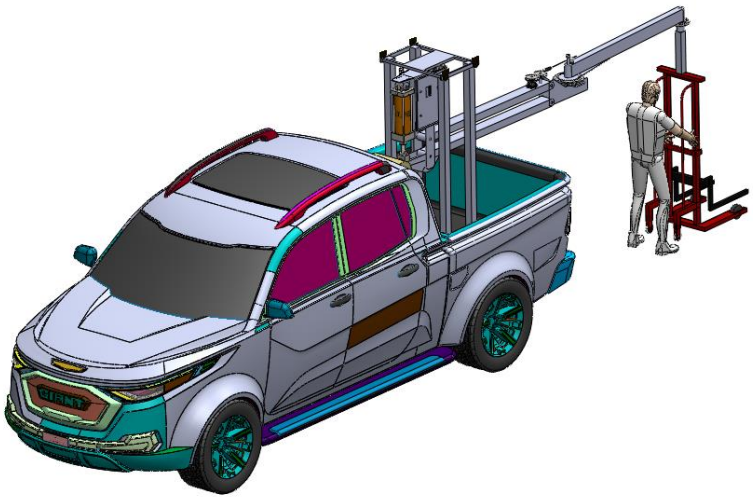

Figure 34: Illustration of the material-handling manipulator working space

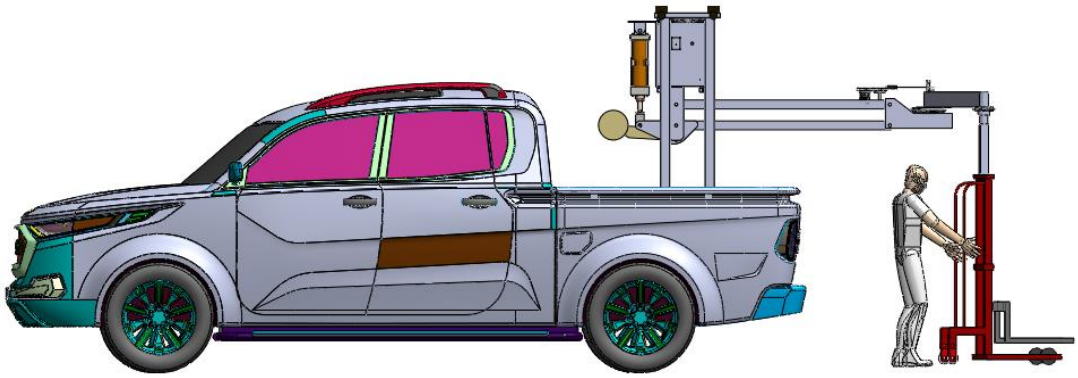

Figure 35: Side view of the mobile robot

6.2 BACK-PAIN REHABILITATION ROBOT

An illustration for the scaled-down prototype CAD model of the Back-Pain Rehabilitation Robot presented in Chapter 2 is shown below in Figure 36.

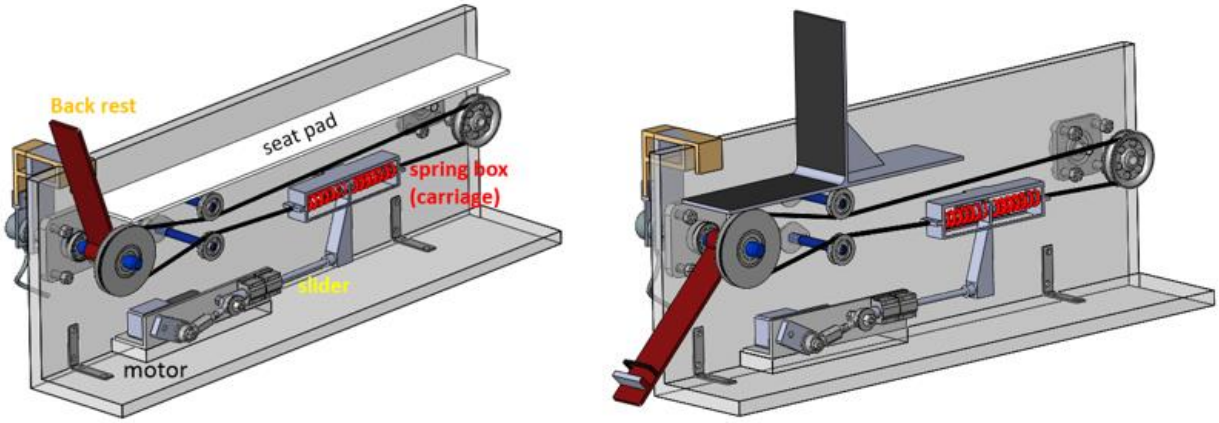

Figure 36: Two configurations: (left) for back-pain. (right) for knee-extension robot

A full-scale model following the concept of series elastic actuation is then designed and built. Figure 37 shows the CAD model of the driving mechanism of the back-pain rehabilitation robot.

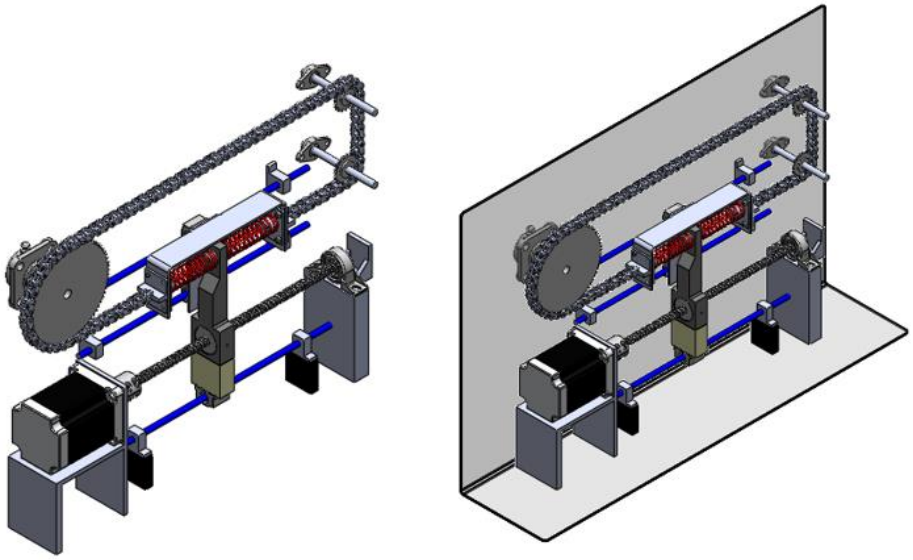

Figure 37: Driving mechanism design of the back-pain rehabilitation robot

Figure 38 shows the full-scale model CAD with the back bench used in the rehabilitation process. The mechanism was then fabricated.

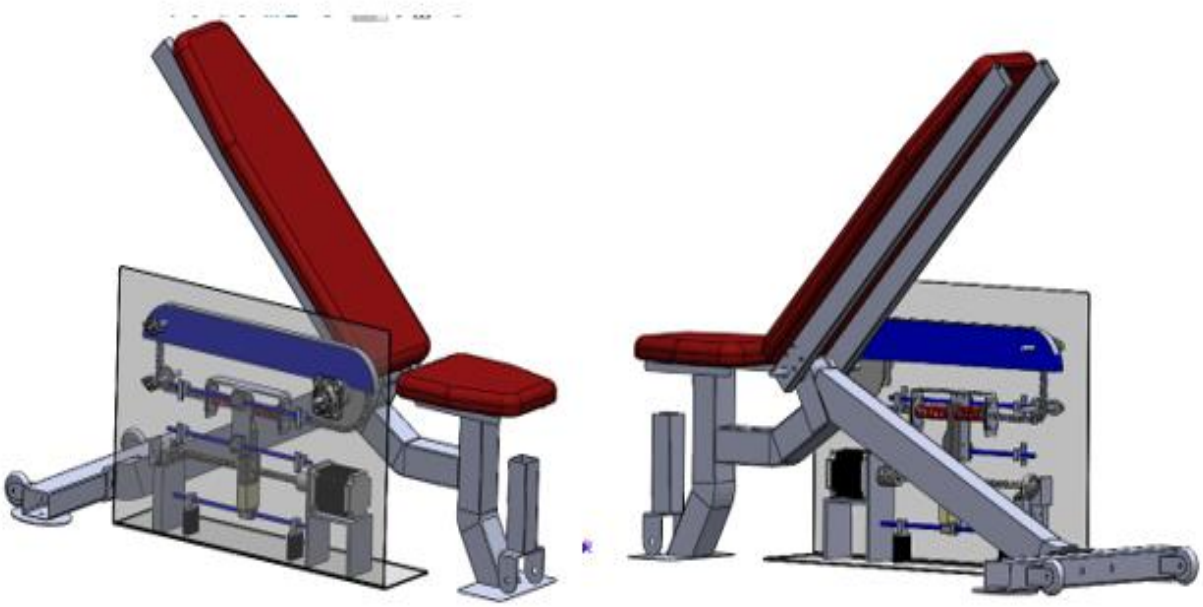

Figure 38: Different views of the back-pain rehabilitation robot

6.3 FLYING BRACHIATOR

Derived from the Brachiating Robot presented in Chapter 3, the flying brachiator is equipped with four propellers in each linkage. This allows the robot to fly and navigate different structures by switching the mobility mode as desired, swing or fly. A camera is attached to the robot to be used in inspecting the structures navigated. Shown in Figure 39 is the flying brachiator in the stationary mode, activating the swing-like-pendulum motion.

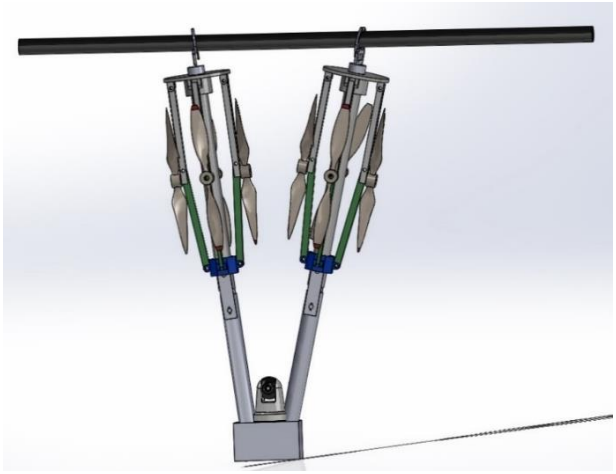

Figure 39: Flying robot in a stationary mode

Shown in Figure 40, the flying brachiator with all its eight propellers activated for flying mode.

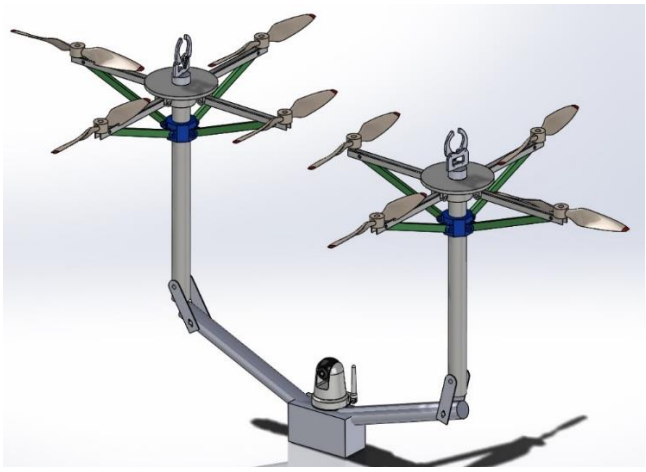

Figure 40: Brachiator robot flying mode illustration

REFRENCES

1. **ElHussein Shata**, Kim-Doang Nguyen, Praneel Acharya, Jeffrey Doom "Series Elastic Robot for Back Pain Rehabilitation," International Journal of Control, Automation, and Systems, 2020.
2. **ElHussein Shata**, Praneel Acharya, Kim-Doang Nguyen, "Brachiating Robot Analysis and Design," IEEE International Conference on Electro/Information Technology, May 2019.
3. **ElHussein Shata**, Praneel Acharya, Marco Ciarcia, Kim-Doang Nguyen, "Optimization of a Chemical Reaction Using the Modified Quasilinearization Algorithm," IEEE International Conference on Electro/Information Technology, May 2019.
4. J. Friedly, C. Standaert, and L. Chan, “Epidemiology of spine care: the back pain dilemma,” Physical Medicine and Rehabilitation Clinics, vol. 21, no. 4, pp. 659–677, 2010.
5. R. van den Berg, M. de Hooge, F. van Gaalen, M. Reijnierse, T. Huizinga, and D. van der Heijde, “Percentage of patients with spondyloarthritis in patients referred because of chronic back pain and performance of classification criteria: experience from the spondyloarthritis caught early (space) cohort,” Rheumatology, vol. 52, no. 8, pp. 1492–1499, 2013.
6. F. Hofmann, U. Stossel, M. Michaelis, M. Nubling, and A. Siegel, “Low back pain and lumbago–sciatica in nurses and a reference group of clerks: results of a comparative prevalence study in germany,” International archives of occupational and environmental health, vol. 75, no. 7, pp. 484–490, 2002.

7. Hult, Lennart. "Cervical, dorsal and lumbar spinal syndromes: a field investigation of a non-selected material of 1200 workers in different occupations with special reference to disc degeneration and so-called muscular rheumatism." Acta Orthopaedica Scandinavica 25.sup17 (1954): 1-102.

8. T. Vos, A. D. Flaxman, M. Naghavi, R. Lozano, C. Michaud, M. Ezzati, K. Shibuya, J. A. Salomon, S. Abdalla, V. Aboyans et al., "Years lived with disability (ylds) for 1160 sequelae of 289 diseases and injuries 1990–2010: a systematic analysis for the global burden of disease study 2010," The lancet, vol. 380, no. 9859, pp. 2163–2196, 2012.

9. G. Wiberg, "Back pain in relation to the nerve supply of the intervertebral disc." Acta Orthopaedica Scandinavica 19.2 (1949): 211-221.

10. T. M. Manini, S. L. Hong, and B. C. Clark, "Aging and muscle: a neuron's perspective," Current opinion in clinical nutrition and metabolic care, vol. 16, no. 1, 2013.

11. Inman, Verne T., et al. "Relation of human electromyogram to muscular tension." Electroencephalography and clinical neurophysiology 4.2 (1952): 187-194.

12. W. E. Garrett Jr, "Muscle strain injuries," The American journal of sports medicine, vol. 24, no. 6 suppl, pp. S2–S8, 1996.

13. M. Gore, A. Sadosky, B. R. Stacey, K.-S. Tai, and D. Leslie, "The burden of chronic low back pain: clinical comorbidities, treatment patterns, and health care costs in usual care settings," Spine, vol. 37(11), pp. 668–677, 2012.

14. B. I. Martin, et al., “Expenditures and health status among adults with back and neck problems,” Jama, vol. 299, no. 6, pp. 656–664, 2008.

15. P. C. Coyte, C. V. Asche, R. Croxford, and B. Chan, “The economic cost of musculoskeletal disorders in canada,” Arthritis and Rheumatism, vol. 11(5), pp. 315–325, 1998.

16. B.Walker, R. Muller, andW. Grant, “Low back pain in australian adults: the economic burden,” Asia Pacific Journal of Public Health, vol 15, no. 2, pp. 79–87, 2003.

17. Y. Henchoz, C. T´etreau, J. Abboud, M. Pich´e, and M. Descarreaux, “Effects of noxious stimulation and pain expectations on neuromuscular control of the spine in patients with chronic low back pain,” The Spine Journal, vol. 13, no. 10, pp. 1263–1272, 2013.

18. N. S. Ward and L. G. Cohen, “Mechanisms underlying recovery of motor function after stroke,” Archives of neurology, vol. 61, no. 12, pp. 1844–1848, 2004.

19. S. M. McGill, “Low back exercises: evidence for improving exercise regimens,” Physical therapy, vol. 78, no. 7, pp. 754–765, 1998.

20. C. Liebenson, Rehabilitation of the spine: a practitioner’s manual. Lippincott Williams & Wilkins, 2007.

21. N. Schweighofer, Y. Choi, C. Winstein, and J. Gordon, “Task-oriented rehabilitation robotics,” American Journal of Physical Medicine & Rehabilitation, vol. 91, no. 11, pp. S270–S279, 2012.

22. I. D´ıaz, J. J. Gil, and E. S´anchez, “Lower-limb robotic rehabilitation: literature review and challenges,” Journal of Robotics, vol. 2011, 2011.

23. W. Choi, J. Won, H. Cho, and J. Park, “A rehabilitation exercise robot for treating low back pain,” in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 482–489.

24. J. Stein, “Robotics in rehabilitation: technology as destiny,” 2012.

25. B. Vanderborght, et al., “Variable impedance actuators: A review,” Robotics and Autonomous Systems, vol. 61, no. 12, pp. 1601–1614, 2013.

26. G. A. Pratt and M. M. Williamson, “Series elastic actuators,” IEEE/RSJ Int’l Conference on Intelligent Robots and Systems, pp. 399–406, 1995.

27. N. Paine, S. Oh, and L. Sentis, “Design and control considerations for high-performance series elastic actuators,” IEEE/ASME Transactions on Mechatronics, vol. 19, no. 3, pp. 1080–1091, 2013.

28. J. S. Sulzer, M. A. Peshkin, and J. L. Patton, “Marionet: An exotendon-driven rotary series elastic actuator for exerting joint torque,” in Int’l Conference on Rehabilitation Robotics, pp. 103–108, 2005.

29. C. Lee and S. Oh, “Configuration and performance analysis of a compact planetary geared elastic actuator,” in IEEE Industrial Electronics Society Conf., pp. 6391–6396, 2016.

30. M. Plooij, M. Van Nunspeet, M. Wisse, and H. Vallery, “Design and evaluation of the bi directional clutched parallel elastic actuator (bic-pea),” 2015 IEEE Int’l Conference on Robotics and Automation, pp. 1002–1009, 2015.

31. K. Kong, J. Bae, and M. Tomizuka, “A compact rotary series elastic actuator for human assistive systems,” IEEE/ASME transactions on mechatronics, vol. 17, no. 2, pp. 288–297, 2011.

32. M. Lauria, M. Legault, M.A. Lavoie, and F. Michaud, “Differential elastic actuator for robotic interaction tasks,” Int’l Conf. on Robotics and Automation, pp. 3606–3611, 2008.

33. G. Tonietti, R. Schiavi, and A. Bicchi, “Design and control of a variable stiffness actuator for safe and fast physical human robot interaction,” IEEE Int’l Conference on Robotics and Automation, pp. 526–531, 2005.

34. B.-S. Kim and J.-B. Song, “Design and control of a variable stiffness actuator based on adjustable moment arm,” IEEE Transactions on Robotics, 28(5), pp. 1145–1151, 2012.

35. Y. Xin, Z.C. Qin, J.Q. Sun , “Robust Experimental Study of Data-driven Optimal Control for an Underactuated Rotary Flexible Joint,” International Journal of Control, Automation and Systems, vol 18, no. 1, pp. 1–13, 2019.

36. H. Azcaray, et al., “Robust GPI control of a new parallel rehabilitation robot of lower extremities,” International Journal of Control, Automation and Systems, vol 16, no. 5, pp. 2384–2392, 2018.

37. R. Van Ham, M. Van Damme, B. Vanderborght, B. Verrelst, and D. Lefeber. "Maccepa: The mechanically adjustable compliance and controllable equilibrium position actuator," Proceedings of CLAWAR, pp. 196-203. 2006.

38. S. Fasoli, H. Krebs, N. Hogan, “Robotic technology and stroke rehabilitation: translating research into practice,” Topics in Stroke Rehabilitation, vol 11(4), pp. 11–19, 2004.

39. P Lum, D Reinkensmeyer, R Mahoney, “Robotic devices for movement therapy after stroke: current status and challenges to clinical acceptance,” Topics in stroke rehabilitation, vol 8, no. 4, pp. 40–53, 2002.

40. CA Stanger, C Anglin, WS Harwin, “Devices for assisting manipulation: a summary of user task priorities,” IEEE Transactions on rehabilitation Engineering, vol 2, no. 4, pp. 256–265, 1994.

41. M Lee, M Rittenhouse, HA Abdullah “Design issues for therapeutic robot systems: results from a survey of physiotherapists,” Journal of Intelligent and robotic systems, vol 42, no. 3, pp. 239–252, 2005.

42. B. Kim, “Analysis on effective rehabilitation of human arms using compliant strap,” Int’l Journal of Control, Automation and Systems, vol. 16(6), pp. 2958–2965, 2018.

43. J. Hart, et al., “Quadriceps inhibition after repetitive lumbar extension exercise in persons with a history of low back pain,” J. Athletic Training, vol. 41(3), pp. 264–269, 2006.

44. N. Kofotolis, E. Kellis, S. P. Vlachopoulos, I. Gouitas, and Y. Theodorakis “Effects of Pilates and trunk strengthening exercises on health-related quality of life in women with chronic low back pain,” Journal of Back and Musculoskeletal Rehabilitation, vol. 29, no. 4, pp. 649–659, 2016.

45. M. M. R. Hutten, H. J. Hermens “Reliability of lumbar dynamometry measurements in patients with chronic low back pain with test-retest measurements on different days," European spine journal, vol 6, no. 1, pp. 54–62, 1997.

46. B. Hwang and D. Jeon “Estimation of the user’s muscular torque for an over-ground gait rehabilitation robot using torque and insole pressure sensors,” Journal of Control, Automation and Systems, vol 16(1), pp. 275–283, 2018.

47. C. Larivière, et al., “Specificity of a back muscle exercise machine in healthy and low back pain subjects,” Medicine and Science in Sports and Exercise, 42(3), 592–599, 2010.

48. P. Cordo, et al., “The sit-up: complex kinematics and muscle activity in voluntary axial movement,” J. Electromyography and Kinesiology, vol 13, no. 3, pp. 239–252, 2003.

49. R. Escamilla, C. Lewis, D. Bell, G. Bramblet, J. Daffron, S. Lambert, A. Pecson, R. Imamura, L. Paulos, and J. Andrews, et al., “Core muscle activation during Swiss ball and traditional abdominal exercises,” journal of orthopaedic and sports physical therapy, vol 40, no. 5, pp. 265–276, 2010.

50. C. Johnson, and JG. Reid, et al., “Lumbar compressive and shear forces during various trunk curl-up exercises,” Clinical Biomechanics, vol 6, no. 2, pp. 97–104, 1991.

51. H. Su, C. Yang, G. Ferrigno, and E. De Momi, 2019, “Improved human-robot collaborative control of redundant robot for teleoperated minimally invasive surgery,” IEEE Robotics and Automation Letters, vol. 4(2), pp.1447-1453.

52. H. Su, J. Sandoval, P. Vieyres, G. Poisson, G. Ferrigno, and E. De Momi, “Safety-enhanced collaborative framework for tele-operated minimally invasive surgery using a 7-DoF torque-controlled robot”, Int’l Journal of Control, Automation and Systems, vol. 16(6), pp. 2915-2923, 2018.

53. Thüring, Manfred, and S. Mahlke. "Usability, aesthetics and emotions in human–technology interaction." International journal of psychology 42.4 (2007): 253-264.

54. Takayama, Leila, Wendy Ju, and C. Nass. "Beyond dirty, dangerous and dull: what everyday people think robots should do." 2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI). IEEE, 2008.

55. United States Department of Labor, Commonly Used Statistics.

56. “Bridges.” ASCE's 2017 Infrastructure Report Card.

57. Rakha, Tarek, and A. Gorodetsky. "Review of Unmanned Aerial System (UAS) applications in the built environment: Towards automated building inspection procedures using drones." Automation in Construction 93 (2018): 252-264.

58. J. R. Usherwood and J. E. Bertram, “Understanding brachiation: insight from a collisional perspective,” Journal of Experimental Biology, vol. 206, no. 10, pp. 1631–1642, 2003.

59. T. Libby, T. Y. Moore, E. Chang-Siu, D. Li, D. J. Cohen, A. Jusufi, and R. J. Full, “Tail-assisted pitch control in lizards, robots and dinosaurs,” Nature, vol. 481, no. 7380, pp. 181–184, 2012.

60. Lo, Alex, et al. "Model-Based Design and Evaluation of a Brachiating Monkey Robot with an Active Waist." Applied Sciences 7.9 (2017): 947.

61. T. Fukuda and Y. Hasegawa, “Mechanism and control of mechatronic system with higher degrees of freedom,” Annual reviews in control, vol. 28, no. 2, pp. 137–155, 2004.

62. Huissoon, Jan Paul, and D. Wang. "On the design of a direct drive 5-bar-linkage manipulator." Robotica 9.4 (1991): 441-446.

63. Nakanishi J, Radulescu A, Braun DJ, Vijayakumar S. “Spatio-temporal stiffness optimization with switching dynamics.” Autonomous Robots. 2017 Feb 1;41(2):273-91.

64. Alaneme, K. K., and M. O. Bodunrin. "Corrosion behavior of alumina reinforced aluminium (6063) metal matrix composites." Journal of Minerals and Materials Characterization and Engineering 10.12 (2011): 1153.

65. Siddiqui, Rafiq A., Hussein A. Abdullah, and Khamis R. Al-Belushi. "Influence of aging parameters on the mechanical properties of 6063 aluminium alloy." Journal of Materials Processing Technology 102.1-3 (2000): 234-240.

66. F. Michilsens, K. D’Aoˆut, and P. Aerts, “How pendulum-like are siamangs? energy exchange during brachiation,” American journal of physical anthropology, vol. 145, no. 4, pp. 581–591, 2011.

67. S. Kameswaran, and L. T. Biegler. “Simultaneous dynamic optimization strategies: Recent advances and challenges,” Computers & Chemical Engineering, vol. 30, pp. 1560-1575, 2006.

68. W.H. Chen, D.J. Ballance, and P.J. Gawthrop, “Optimal control of nonlinear systems: a predictive control approach”, Automatica, vol. 39, no. 4, pp. 633-641, 2003.

69. M. Abu-Khalaf and F. L. Lewis, “Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach,” Automatica, vol. 41, no. 5, pp.779-791, 2005.

70. M.R. AlRashidi and M.E. El-Hawary, “A survey of particle swarm optimization applications in electric power systems,” IEEE transactions on evolutionary computation, vol. 13, no. 4, pp. 913-918, 2009.

71. T.A. Johansen, T. I. Fossen, and S. P. Berge, “Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming,” IEEE Transactions on Control Systems Technology, vol. 12, no. 1, pp.211-216, 2004.

72. A. D. Jayal, F. Badurdeen, O. W. Dillon Jr, and I.S. Jawahir, “Sustainable manufacturing: Modeling and optimization challenges at the product, process and system levels,” CIRP Journal of Manufacturing Science and Technology, vol. 2, no. 3, pp. 144-152, 2010.

73. T. Zegard and G. H. Paulino, “Bridging topology optimization and additive manufacturing,” Structural and Multidisciplinary Optimization, vol. 53, no. 1, pp.175-192, 2016.

74. R. Karuppiah and I. E. Grossmann, “Global optimization for the synthesis of integrated water systems in chemical processes,” Computers & Chemical Engineering, vol. 30, no. 4, pp. 650-673, 2006.

75. H. Ma, M. Fei, and Z. Yang, “Biogeography-based optimization for identifying promising compounds in chemical process,” Neurocomputing, vol. 174, pp. 494-499, 2016.

76. P. B. Wigley, P. J. Everitt, A. van den Hengel, J. W. Bastian, M. A. Sooriyabandara, G. D. McDonald, K. S. Hardman, C. D. Quinlivan, P. Manju, C. C. Kuhn, and I. R. Petersen, “Fast machine-learning online optimization of ultra-cold-atom experiments,” Scientific reports, vol. 6, p.25890, 2016.

77. R. E. Bellman, and R. E. Kalaba. "Quasilinearization and nonlinear boundary-value problems." 1965.

78. A. Miele, and A. V. Levy, Modified Quasilinearization and Optimal Initial Choice of the Multipliers, Part 1, Mathematical Programming Problems, Journal of Optimization Theory and Applications, Vol. 6, No. 5, 1970.

79. S. Werner, and James J. Morgan. “Aquatic chemistry: chemical equilibria and rates.” natural waters. Vol. 126. John Wiley & Sons, 2012.

80. S. Jarungthammachote and A. Dutta. "Equilibrium modeling of gasification: Gibbs free energy minimization approach and its application to spouted bed and spout-fluid bed gasifiers." Energy Conversion and Management Vol 49.6, pp. 1345-1356, 2008.

81. G. Eriksson and K. Hack. "ChemSage—a computer program for the calculation of complex chemical equilibria." Metallurgical transactions vol 21.6, pp 1013-1023, 1990.

82. W. B. White, S. H. Johnson, and G. B. Dantzig, "Chemical Equilibrium in Complex Mixtures," J. Chem. Phys., vol 28, pp 751-755, 1958.

83. H. M. Gomes, and A. M. Awruch. "Comparison of response surface and neural network with other methods for structural reliability analysis." Structural safety 26.1 (2004): 49-67.

84. I. Koutis, G. L. Miller, and D. Tolliver. "Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing." Computer Vision and Image Understanding 115.12 (2011): 1638-1646.

85. S. I. Gallant. Neural network learning and expert systems. MIT press, 1993.

86. J. Zupan, "Introduction to artificial neural network (ANN) methods: what they are and how to use them." Acta Chimica Slovenica 41 (1994): 327-327.

87. N. B. Karayiannis, and G. W. Mi. "Growing radial basis neural networks: Merging supervised and unsupervised learning with network growth techniques." IEEE Transactions on Neural networks 8.6 (1997): 1492-1506.

88. P. Baldi, "Autoencoders, unsupervised learning, and deep architectures." Proceedings of ICML workshop on unsupervised and transfer learning. 2012.