DESIGN, SIMULATION AND CONTROL OF A TENDON-DRIVEN ROBOT ARM

By

ASHMITA LAHIRI

A thesis submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Mechanical and Aerospace Engineering

Written under the direction of

Professor Aaron D. Mazzeo

And approved by

______________________________
______________________________
______________________________
______________________________

New Brunswick, New Jersey

May 2022
ABSTRACT OF THE THESIS

DESIGN, SIMULATION AND CONTROL OF A TENDON-DRIVEN ROBOT ARM

By ASHMITA LAHIRI

Thesis Director:

Professor Aaron D. Mazzeo

This thesis presents a design for cable-driven serial manipulators with two degrees of freedom which is lightweight and safe to use in a non-industrial setting. Conventional industrial robot arms have several applications such as welding, palletization and on the assembly line, but are heavy to use, have large sizes and are not always safe for human operators in proximity. This thesis aims to explore the idea of a cable-driven serial manipulator, where a simple system of cables and pulleys transmits motion between the actuators and links. The model proposed here is a cable-driven robot arm which attempts to overcome the issues of conventional robots by placing all the actuators in a motor bank at the base of the robot body as opposed to positioning them at every joint along the arm, and by reducing the total number of actuators in the model. An experimental model constructed using 3D printed parts forms the basis of digital models built using the Simscape Multibody toolbox. The design explored here is based on a microgravity environment.

Initially, physical models of the DC motors are used in the Simscape models, which are later represented using transfer functions. PID controllers control these motors, and the required parameters are obtained using the root locus method through calculations and MATLAB codes. The next step involves testing the Simscape model by recording its response to a step input. This
is first done on two different single degree-of-freedom arms, followed by a 2 degree-of-freedom model. In order to limit fluctuations in voltage of the motors, a low-pass filter has been used in these models. Comparison with output plots from MATLAB calculations validate these results.
Acknowledgment

I am thankful to my advisor Prof. Aaron D. Mazzeo for giving me the opportunity to be a part of his research group and to work on this project. His continued guidance and support throughout the duration of this project made this research possible. I have gained immensely from his in-depth knowledge of the subject and feel better prepared for my future career as an engineer.

I would also like to thank Prof. Haim Baruh and Prof. Onur Bilgen for agreeing to be on my committee and for taking the time to review my work.

I am grateful to Chinmay Sridhar, Paul Wang and Cyril Nwako for sharing their research data with me, and to my fellow graduate students Noah Harmatz and Xiyue Zou for providing their assistance and support whenever I needed it.
Dedication

I dedicate this thesis to my wonderful parents, Avik and Sarbani, and to Arpan, for always believing in me.
# Table of Contents

Abstract of the Thesis ............................................................................................................. ii

Acknowledgment .................................................................................................................. iv

Dedication .............................................................................................................................. v

List of Tables ......................................................................................................................... viii

List of Figures ....................................................................................................................... ix

Chapter 1 - Introduction ........................................................................................................ 1

1.1 Background ....................................................................................................................... 1

1.1.1 Serial Manipulators ..................................................................................................... 1

1.1.2 Cable-Driven Robots .................................................................................................. 2

1.1.3 Robotics in Microgravity ............................................................................................ 4

1.1.4 Control Strategy ......................................................................................................... 5

1.2 Motivation and Objective ............................................................................................... 7

1.3 Overview .......................................................................................................................... 7

Chapter 2 - Modeling ............................................................................................................. 9

2.1 Experimental Setup ....................................................................................................... 9

2.2 Simulink Model .............................................................................................................. 11

Chapter 3 - Methodology and Calculations ........................................................................ 16

3.1 Methodology .................................................................................................................. 16

3.2 Derivation of Motor Transfer Function ......................................................................... 17

3.3 Open-Loop Analytical Model ......................................................................................... 19

3.4 Closed-Loop Analytical Model ...................................................................................... 21
Table 1. Comparison of properties of Dyneema, Vectran and Kevlar materials for cables [20]. . 11

Table 2. Original and Reduced Dimensions of Different Parts of the Model. ............................ 17

Table 3. List of Motor Parameters [22]. ...................................................................................... 24

Table 4. Dimensional Parameters of Single Degree-of-Freedom Robot Model. ........................ 25

Table 5. Comparison of weights of various serial robots [4, 23-26]. ........................................ 47
List of Figures

**Figure 1.** The KUKA KR CYBERTECH nano KR 6 R1840-2 articulated robot arm. It is a serial manipulator that has many industrial applications. Image from [3]. ............................................. 1

**Figure 2.** Schematic of a cable-driven parallel robot (CDPR). The model represented here has an end-effector suspended from a rigid frame with an eight-cable configuration. Image from [9]...... 3

**Figure 3.** A tendon robot built using Myorobotics that uses two separate cables and actuators to mimic muscle motion. Image from [12]. .................................................................................. 4

**Figure 4.** The Eurobot mobile system developed by the ESA. It has two seven joint arms attached to a rover torso and will assist astronauts in space. Image from [15]. ............................................. 5

**Figure 5.** Block diagram of a closed-loop control system ........................................................................... 6

**Figure 6.** Experimental setup of robot arm. (a) Intended assembly of base of robot and three-link arm. Temporary fixtures and cables represented using twine ropes. (b), (c), (d) Intended locations of DC motor mounts inside the housing. .................................................................................. 10

**Figure 7.** Simscape models of 1 DoF and 3 DoF robot arm. (a) Single degree-of-freedom model designed on Simscape. Model consists of a driver pulley that is directly driven by the motor. The driver pulley is connected to the driven pulley with cables. The driven pulley is attached to the link so that the link moves along with the driven pulley. (b) Simscape model with three degrees-of-freedom. Pulleys dr1, dr2 and dr3 are the three driver pulleys that are independently actuated by one DC motor each. The three driven pulleys dn1, dn2 and dn3 are directly attached to links 1, 2 and 3 respectively and they move together as a single solid. Driver pulleys dr2 and dr3 require idle pulleys (i1, i2 and i3) to transmit motion to links 2 and 3. The idle pulleys transmit motion from the driver pulleys to the links with the help of pulleys directly attached to them (a1, a2 and a3). (c) A view of the 3 DoF model when provided with a step Voltage input................................. 13
Figure 8. Simscape models of DC motor and 1 DoF arm. (a) Image of the DC motor model developed using Simscape Electrical. (b) Open-loop 1 DoF model. (c) Closed-loop 1 DoF model.

Figure 9. Circuit diagram of an armature-controlled DC motor. Image from [21].

Figure 10. Open-loop Simulink model developed based on derived analytical model.

Figure 11. Closed-loop Simulink model developed based on derived analytical model. The angular position output of the link feeds back to the controller to establish a control loop.

Figure 12. Block diagram representing transfer functions of components in a closed-loop system with unity feedback for single degree-of-freedom model.

Figure 13. Figure representing the problem of finding PID parameters for a single degree-of-freedom system using root locus. The red crosses on the real axis indicate the poles. The green circle represents the zero to be added. Pole and zero angles are represented with θ and pole and zero lengths are marked by L. The figure is not drawn to scale.

Figure 14. Root locus plots from MATLAB. (a) Root locus plot for open-loop system. (b) Root locus plot with PD controller (where a is found). (c) Root locus plot of system with PID controller. Similarity to (b) satisfies pole-zero cancellation.

Figure 15. Open-loop response plots of 1 DoF model with original dimensions. (a) Angular position vs time. (b) Angular velocity vs time. (c) Effective voltage input vs time (ΔV). (d) Current vs time.

Figure 16. Closed-loop response plots of 1 DoF model with original dimensions. (a) Angular position vs time. (b) Angular velocity vs time. (c) Effective voltage input vs time. (ΔV). (d) Current vs time.

Figure 17. Open-loop response plots of 1 DoF model with reduced dimensions. (a) Angular position vs time. (b) Angular velocity vs time. (c) Effective voltage input (ΔV) vs time. (d) Current vs time.
Figure 18. Closed-loop response plots of 1 DoF model with reduced dimensions. (a) Angular position vs time. (b) Angular velocity vs time. (c) Effective voltage input vs time. (Δ𝑉). (d) Current vs time.

Figure 19. Plots of changes in step input and resulting responses. (a) Effect of addition of low-pass filter on step input. (b) Δ𝑉 vs time. (c) Current vs time.

Figure 20. Responses of both links of the 2 DoF open-loop model to step input voltage. Angular displacement (𝜃𝐿), angular velocity (𝜃𝐿), Δ𝑉 and current are plotted with respect to time.

Figure 21. Responses of both links of the 2 DoF closed-loop model to step input voltage. Angular displacement (𝜃𝐿), angular velocity (𝜃𝐿), Δ𝑉 and current are plotted with respect to time.

Figure 22. Voltage and Current response of 1 DoF model with time constant τ = 10 ms.
Chapter 1 - Introduction

1.1 Background

1.1.1 Serial Manipulators

Serial manipulators are the most common type of robot arms used in industry today [1]. These are anthropomorphic robot arms, that is, they mimic the human arm in form and motion. The links that make up the arm connect one after the other, with the first link connected to a base and the final link connected to an end effector. Such robots are very versatile and have a wide range of applications in industrial settings, for example, on the assembly line, for welding, for painting, for palletization and in cleanroom operations [2]. Figure 1 shows an articulated robot arm developed by KUKA which can perform all these operations [3].

![Articulated Robot Arm](image)

**Figure 1.** The KUKA KR CYBERTECH nano KR 6 R1840-2 articulated robot arm. It is a serial manipulator that has many industrial applications. Image from [3].

Traditionally, industrial robots have actuators located at every joint along the arm where each actuator facilitates the motion of the link immediately following it. Based on the required payload, each joint on these robots can accommodate multiple actuators and geared transmissions. This enables the robots to have a wide range of payload capabilities from 0.5 kg [4] to 2300 kg [5].
With the increasing popularity of serial manipulators in industrial applications, there is a rising demand for such robots in non-industrial settings as well. This brings into consideration, the safety of human operators when using robots which are primarily meant to serve as industrial solutions. Industrial robots perform tasks that are often hazardous for human workers, which in turn requires them to be heavy and have higher strength and inertia. This makes them unsafe for usage in proximity to human beings without safety features such as fences, barricades and automatic shut-off. By conventional design, since serial manipulators have actuators at every joint along their lengths, this also means that the arms carry the weight of all the actuating components whenever they move, which is a major factor that contributes to their increased weights.

1.1.2 Cable-Driven Robots

Robot arms commonly have pneumatic, hydraulic or electric drives or their combination [6]. Cable-driven robots are manipulators where the transmission of motion is achieved by controlling the winding and unwinding of wires or cables rather than rigid links [7]. Based on their kinematic structures, robots are classified as serial, parallel or hybrid manipulators. While serial manipulators have links connected successively to each other, a parallel manipulator is one where a moving platform is connected to a fixed based by at least two arms. Cable-based actuation has been explored primarily for parallel manipulators, thus called cable-driven parallel robots (CDPR). Figure 2 shows an example of a CDPR. Currently, cable-driven robots are very popular in robotics rehabilitation [8].
**Figure 2.** Schematic of a cable-driven parallel robot (CDPR). The model represented here has an end-effector suspended from a rigid frame with an eight-cable configuration. Image from [9].

Cable-driven robots have several advantages over conventionally actuated manipulators. In these manipulators, the motors actuating the robot are placed at the base of the robot body, thus reducing the bulk and inertia of the robot. They have a potentially large workspace, are easy to reconfigure and implement, can achieve high speed motion, and have high payload to weight ratio [10]. Cable-driven robots find applications as robotic rehabilitation exoskeletons, camera robots and pick and place robots [7].

A major disadvantage encountered with cable-driven robots is that the cables have to be in tension for motion to be possible [11]. Therefore, these robots can only operate by pulling and cannot push. Hence, in the usual designs, just one actuator is not enough to mimic the motion of both agonist and antagonist muscles. Figure 3 is an example of a tendon robot arm which uses two different cables, each actuated by a brushless DC motor [12].
**Figure 3.** A tendon robot built using Myorobotics that uses two separate cables and actuators to mimic muscle motion. Image from [12].

### 1.1.3 Robotics in Microgravity

Robots in space assist astronauts in performing dangerous, dull or repetitive tasks, similar to their usage on earth. Some of their uses include collecting samples, moving equipment around, moving astronauts and taking measurements. There are additional considerations when it comes to designing and developing these robots. For example, there are extreme thermal conditions and radiations in space that these robots have to withstand [13]. Since launching any mass into space is expensive, these robots have to be lightweight as well. They must also be extremely reliable, autonomous and consume less power. Microgravity robots have higher payload capabilities than those in a gravitational field. An example of a microgravity robot is the Nanokhod developed by the European Space Agency (ESA), which is a micro-rover that weighs less than 2 kg but can carry a mass of 1 kg [14]. The ESA Eurobot ground prototype, shown in figure 4, will assist astronauts during space exploration [15].
1.1.4 Control Strategy

Control systems are important in regulating the working of mechanical and electrical systems. They help correlate the input and output signals of the system by controlling the interaction of all its components. Control algorithms can be open-loop or closed-loop depending on the feedback mechanism. In an open-loop control system, the controller does not receive any feedback from the system. While this produces a simple design, an open-loop system cannot detect any unforeseen changes due to external disturbances. In contrast, a closed-loop control system is one where the output signal of the system loops back to the controller so that it can recognize any deviations from the desired output. The difference between the reference input signal and the looped output signal generates an error signal, which the controller uses to account for any changes that may occur due to external disturbances. Figure 5 represents a block diagram of a closed-loop system.

Figure 4. The Eurobot mobile system developed by the ESA. It has two seven joint arms attached to a rover torso and will assist astronauts in space. Image from [15].
There are several control algorithms that can be used in a control system. PID control is the most commonly used algorithm [16]. It uses three control parameters, namely, Proportional (P), Integral (I) and Derivative (D), which can be used in any combination. Each of these parameters interacts differently with the signals in the system. The P-controller provides an output which is proportional to the error of the system. The output of the controller is the error of the system multiplied with a proportional constant, called the proportional gain or $K_p$. Since the control process is based only on the actual value of the error, and not the rate of error, the gains of such a system can become too high causing a lot of overshoot and oscillations in the output response. High gains can cause a system to become unstable. The I-controller calculates its output by multiplying the gain ($K_i$) with the integral of the error. This type of controller accumulates or sums up the error over time, which is useful in avoiding steady-state errors. I-controllers are slow in their operation and should be paired with other controllers for a faster response. The D-controller calculates its output by multiplying the gain ($K_d$) with the derivative of the error. The derivative controller, therefore, takes the time rate of change of the error into consideration. The advantage of this controller is that it tries to prevent the oscillations that can occur in a system. A D-controller does not achieve steady-state on its own and works on minimizing the rate of change of error, rather than reaching the desired output. It must be used with another type of controller. A PID controller combines all three types of these parameters. The gains are calculated based on the plant, input and

Figure 5. Block diagram of a closed-loop control system.
desired output. Some methods to find the controller gains are using plots like root locus, Bode plots or Nyquist plots, or using methods like Ziegler-Nichols.

1.2 Motivation and Objective

As described earlier, the traditional design of a serial manipulator arm generally has actuator motors at every joint. This makes the arm quite heavy which reduces its overall payload capacity. Moreover, such a heavy arm can pose stability risks in dynamic environments involving rapid movements. For delicate applications, such as using a robot arm for aiding medical patients, a heavier design poses further safety risks. The objective of this thesis is to provide a new design for serial manipulators that is lighter in weight and safer to use as compared to robot arms commercially available today. Weight considerations are also crucial in microgravity robots, which should preferably be light in weight. Actuating with cables provides an option to position all the actuators at the base of the arm, rather than along its length. Hence, a robot arm with two degrees-of-freedom, driven by a system of cables and pulleys, has been designed with a microgravity environment. Additionally, the model presented in this thesis uses only one DC motor for each joint to replicate the antagonistic muscle motion. Using fewer and lighter actuators instead of the heavier ones used in the industry significantly contributes to the reduction in weight and inertia of the arm. For these motors to work, the material then used for the robot arm also needs to be one that is lightweight but strong, and Polylactic Acid (PLA) is chosen for this purpose.

1.3 Overview

The thesis begins with a literature study of existing serial manipulators and cable driven robots. An experimental model is constructed and then replicated in Simulink for further testing and simulations. A single degree-of-freedom model is first designed using the Simscape Multibody toolbox in order to accurately simulate a controlled motion as well as to tune the PID parameters. Cables and pulleys are added to the Simscape model using the Belts and Cables toolbox. After this,
the model is extended to a two degree-of-freedom system with one motor for each joint. PID controllers are implemented to accurately control the three motors. The gains of the controllers are found using the root locus method. The Simscape models are then subjected to step inputs and the responses are recorded. For validation, the results of the simulations are compared with the results of manual calculations and MATLAB codes. On studying these results, anomalies in the models so created come to light, due to which the DC motors are remodeled. Simulations are then performed, and these results are compared again with results from analytical models on MATLAB. The entire study is done with a microgravity environment in consideration.
Chapter 2 - Modeling

2.1 Experimental Setup

The experimental model consists of a robot arm attached to a robot base. The base is made from Plexiglass sheets which are held together using L-clamps and houses the actuators and the controller. The arm consists of three rigid links, each connected to a pulley at the joint. There is a shaft at each of the joints to hold the corresponding pulley and end of the link at a fixed axis about which they can rotate. The pulleys and links have grooves for the cables. The actuators are fixed to the housing using 3D-printed motor mounts. Spools of cables for each link are attached to the shafts of the actuators. Figure 6(a) shows an image of a rough assembly of the experimental model, and figures 6(b), 6(c) and 6(d) depict the positions of placement of motors.

The arms, pulleys, spools, and motor mounts were made from Polylactic Acid (PLA) using Fused Deposition Modeling (FDM). First, CAD models of the parts were designed using the Autodesk Inventor software. These were then converted to STL files to be sent to the 3D printer. PLA and ABS (Acrylonitrile butadiene styrene) are the two most commonly used materials while manufacturing parts using FDM. Among these, PLA was chosen due to the following advantages. PLA (7250 psi) is stronger than ABS (4100 psi). PLA has better printability and surface finish when compared to ABS. Another advantage is that post-processing is easy with PLA. The base of the robot was constructed using plexiglass for its optical clarity and aesthetics.
Figure 6. Experimental setup of robot arm. (a) Intended assembly of base of robot and three-link arm. Temporary fixtures and cables represented using twine ropes. (b), (c), (d) Intended locations of DC motor mounts inside the housing.

Choice of material of cables depends on properties such as strength, flexibility, friction coefficient, elasticity and cost. Different materials such as Dyneema, Kevlar and Vectran were compared based on these criteria and Vectran cables of 2mm diameter were chosen for this model. Vectran is made from a Liquid Crystal Polymer (LCP) fiber and shows high tensile strength and excellent resistance to abrasion and creep [17]. In comparison to Kevlar, Vectran is between 8 to
14 times more flexible [18]. Compared to Dyneema, it has better creep resistance as well as a higher coefficient of friction, which is essential to reduce slippage between the cable and the pulleys. It also displays very good heat and chemical resistance and has minimal moisture retention. Due to these properties of Vectran, it is used by NASA to manufacture spacesuits and airbags [19]. Table 1 compares the characteristics of Vectran, Dyneema and Kevlar. Vectran is more expensive than either Dyneema or Kevlar, but this does not make a significant difference for the quantity required in this project.

**Table 1.** Comparison of properties of Dyneema, Vectran and Kevlar materials for cables [20].

<table>
<thead>
<tr>
<th>Property</th>
<th>Dyneema</th>
<th>Vectran</th>
<th>Kevlar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tenacity (g/den)</td>
<td>34.2</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>Elongation at Break (%)</td>
<td>3.4</td>
<td>3.3</td>
<td>4</td>
</tr>
<tr>
<td>Friction Coefficient</td>
<td>0.05 – 0.07</td>
<td>0.12 – 0.15</td>
<td>0.12 – 0.15</td>
</tr>
</tbody>
</table>

**2.2 Simulink Model**

Due to the Covid-19 pandemic, further assembly and testing of the experimental model could not be pursued. Alternatively, a model was built on the Simulink interface on MATLAB, and this was used to obtain results and make observations.

The arm was constructed using the Simscape Multibody toolbox on Simulink. The Simscape Multibody toolbox helps to build 3D models in the Simulink environment for simulations and testing. In addition, the cables and pulleys were added to the models using the Belts and Cables toolbox on Simulink.

A step-by-step approach was taken, beginning with the modeling of a virtual single degree-of-freedom arm. Figure 7(a) is an image of the Simscape model of the 1 DoF arm. This included a driver pulley connected to a driven pulley with a single cable wound around them, to form a simple
conveyor type arrangement. The driver pulley was actuated directly by the motor and motion is transmitted by the cable. The driven pulley and the link rotate in unison.

The model with a single degree-of-freedom was then developed further to include another link to form a 2 DoF robot arm. The design of this model had to be altered significantly from the experimental model due to constraints in modeling using Simscape. Similar to the 1 DoF model, the design first starts with a pulley attached directly to the shaft of the DC motor. This driver pulley is then connected in a conveyor arrangement with an idle pulley. The idle pulley is on the same axis as the driven pulley of the previously attached link. By this arrangement, the rotation of the idle pulley about its own axis is only affected by the driver pulley it is directly attached to, and any translational motion is influenced by link 1. The idle pulley then facilitates the motion of link 2. The same design concept is then applied to build the model with three degrees-of-freedom as shown in figures 7(b) and 7(c).
Figure 7. Simscape models of 1 DoF and 3 DoF robot arm. (a) Single degree-of-freedom model designed on Simscape. Model consists of a driver pulley that is directly driven by the motor. The driver pulley is connected to the driven pulley with cables. The driven pulley is attached to the link so that the link moves along with the driven pulley. (b) Simscape model with three degrees-of-freedom. Pulleys dr1, dr2 and dr3 are the three driver pulleys that are independently actuated by one DC motor each. The three driven pulleys dn1, dn2 and dn3 are directly attached to links 1, 2 and 3 respectively and they move together as a single solid. Driver pulleys dr2 and dr3 require idle pulleys (i1, i2 and i3) to transmit motion to links 2 and 3. The idle pulleys transmit motion from
the driver pulleys to the links with the help of pulleys directly attached to them (a1, a2 and a3). (c)
A view of the 3 DoF model when provided with a step Voltage input.

The DC motors used to actuate the three driver pulleys are also modelled in the Simulink interface, as shown in figure 8(a). The Simscape Electrical toolbox library has a predefined block called DC motor which can be attached to any other required blocks from the Simulink library in order to obtain the desired function. For a model that takes a voltage input and gives angular displacement as output, the DC motor block is connected to a Controlled Voltage Source and an Ideal Rotational Motion Sensor. A current sensor is also added to observe the behavior of the DC motor when it receives the input. The actuators are controlled using PID controllers, whose parameters are evaluated by manual calculations and by running codes on MATLAB. A unity feedback loop is established by measuring the error between the input and the angular displacement output from the DC motor subsystem. Figures 8(b) and 8(c) show the open-loop and closed-loop models developed using this DC motor model.

The results obtained after running simulations using the above DC motors and feedback loop imply that this setup does not account for any payload torque that act on the motors. The feedback loop only takes the angle of the motor into consideration, and the addition of a torque sensor to this model helps to measure the torque coming out of the motor but does not affect the payload, in this case the links and pulleys attached. Therefore, the DC motors were then modeled using transfer functions on Simulink. In these models, the motor outputs a torque, which then acts as input to the revolute joint of the driver pulley after multiplication with a suitable gear ratio. The open-loop model uses a voltage as reference input and the closed-loop model has a reference angular position. The next chapter describes the derivation of these equations.
Figure 8. Simscape models of DC motor and 1 DoF arm. (a) Image of the DC motor model developed using Simscape Electrical. (b) Open-loop 1 DoF model. (c) Closed-loop 1 DoF model.
Chapter 3 - Methodology and Calculations

3.1 Methodology

Based on the experimental model, we constructed a physical model using Simscape Multibody. This initial model had the dimensions of the experimental model. The Simscape models were built starting from a 1 DoF tendon-driven robot arm model, and then developed into 2 DoF models. Simscape Electrical toolbox provided the blocks necessary to build the DC motors used for this model. The angular position output from the DC motor served as an input to the revolute joints of the corresponding driver pulleys. These models, however, did not take the torque of the payload into account and so could not achieve accurate motor control.

As a modification, we derived the transfer function of the motor used to control the driver pulley and built Simulink model based on these equations. After obtaining appropriate motor torque control from the modified 1 DoF models, we then added another link to form a 2 DoF model connected to another transfer function-based DC motor model. While these models solved the problem of improper torque, the voltage and current spiked up to very high values at the instant of application of the step input. Although these signals immediately fell down to more manageable values after the spike, the significantly high values of voltage and current at the spikes could prove harmful to the equipment used. As a way of rectifying this issue, we reduced the dimensions of the links and pulleys. Table 2 gives a list of the original and modified dimensions used. This helped ease the load on the motor and reduced the voltage and current spikes to a certain level, but this was still significantly high and potentially damaging. A low-pass filter was then attached to the input signal and the gear ratio raised. Through these steps, we successfully brought down the signal spikes to more tolerable values.
Table 2. Original and Reduced Dimensions of Different Parts of the Model.

<table>
<thead>
<tr>
<th>Part/Distance</th>
<th>Original Dimensions</th>
<th>Modified Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radius of Pulley</td>
<td>2 cm</td>
<td>0.5 cm</td>
</tr>
<tr>
<td>Height of Pulley</td>
<td>2 cm</td>
<td>0.5 cm</td>
</tr>
<tr>
<td>Length of Link</td>
<td>30 cm</td>
<td>10 cm</td>
</tr>
<tr>
<td>Breadth of Link</td>
<td>4 m</td>
<td>1 cm</td>
</tr>
<tr>
<td>Thickness of Link</td>
<td>2 cm</td>
<td>1 cm</td>
</tr>
<tr>
<td>Distance between the centers of Driver Pulley and</td>
<td>10 cm</td>
<td>10 cm</td>
</tr>
<tr>
<td>Driven Pulley</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.2 Derivation of Motor Transfer Function

Figure 9 shows a circuit diagram of an armature-controlled DC motor, as used in this thesis.

Figure 9. Circuit diagram of an armature-controlled DC motor. Image from [21].
There are four governing equations that determine the response of the motors. The motors used to actuate the joints are armature-controlled DC motors. The torque of the motor is only proportional to the armature current while the magnetic field remains constant. Therefore,

\[ T = K_t i , \]  

where \( T \) – motor torque,
\( i \) – armature current and
\( K_t \) – proportionality constant, also called the motor torque constant.

When the armature of a DC motor rotates through the magnetic field, emf is induced in the coils as in a generator, which is opposite to the direction of the applied voltage. This is the back emf \( (V_b) \) and it is proportional to the angular velocity \( \dot{\theta} \) of the shaft.

\[ V_b = K_b \dot{\theta} , \]  

where \( K_b \) – back emf constant. By the law of conservation of energy, in SI units, \( K_t = K_b \).

By applying Newton’s second law to the diagram, we get,

\[ J\ddot{\theta} + B \dot{\theta} = T = K_t i , \]  

where \( B \) – motor viscous friction constant.

From Kirchoff’s law,

\[ L\frac{di}{dt} + Ri = V - V_b = V - K_b \dot{\theta} . \]  

Applying Laplace transform to equations (3) and (4) results in the following equations:

\[ s(Js + B)\theta(s) = K_t I(s) , \]  

\[ (Ls + R)I(s) = V(s) - K_b s\theta(s) , \]
The transfer function of a system is the ratio of its output and input parameters. For a motor controlled by input voltage where the required output is its angular position, the transfer function is given by \( \theta(s)/V(s) \), which from equations (5) and (6) can be obtained as,

\[
G(s) = \frac{\theta(s)}{V(s)} = \frac{K_t}{s(Ls+R)(Js+B)+K_tK_b}.
\]

(7)

Based on these equations, we derived the open-loop and closed-loop analytical models for the 1 DoF arm system.

3.3 Open-Loop Analytical Model

Starting with the governing equations of the DC motor, we get,

\[
T_m = K_t i,
\]

(8)

\[
T_m = J_m \ddot{\theta}_m + B_m \dot{\theta}_m,
\]

(9)

\[
\Delta V = V_{app} - V_{emf} = L_m \frac{di}{dt} + i R_m,
\]

(10)

\[
V_{emf} = K_b \dot{\theta}_m.
\]

(11)

The dynamics of the link can be described using,

\[
T_L = J_L \ddot{\theta}_L.
\]

(12)

The angular positions of the motor and the link are related as,

\[
\theta_m = n\theta_L,
\]

(13)

where \( n \) – gear ratio.

Thus, the transmission between the motor and link is given as,

\[
T_L \dot{\theta}_L = T_m \dot{\theta}_m,
\]

(14)

\[
T_m s \theta_m = T_L s \theta_L,
\]

(15)

\[
T_m \theta_m = T_L \theta_L.
\]

(16)
\[ T_L = T_m \theta_m / \theta_L = n T_m. \quad (17) \]

When \( V_{app} \) is the voltage input provided to the motor,

\[ V_{app} - V_{emf} = (L_s + R)i = \frac{T_m}{K_t} (L_s + R) = \frac{T_L}{nK_t} (L_s + R), \quad (18) \]

\[ T_L = \frac{nK_t (V_{app} - V_{emf})}{L_s + R} = \frac{nK_t (V_{app} - K_b s \theta_m)}{L_s + R} = \frac{nK_t (V_{app} - K_b s n \theta_L)}{L_s + R}. \quad (19) \]

To calculate relationship between \( T_L \) and inertial damping,

\[ J_m s^2 \theta_m + B_m s \theta_m = T_m, \quad (20) \]

which implies,

\[ J_m s^2 n \theta_L + B_m s n \theta_L = \frac{T_L}{n}, \quad (21) \]

\[ J_m s^2 n^2 \theta_L + B_m n^2 \theta_L = T_L. \quad (22) \]

Also,

\[ J_L s^2 \theta_L = T_L. \quad (23) \]

Combining effects of \( J_m, B_m, J_L \).

\[ [(J_L + J_m n^2)s^2 + B_m n^2 s] \theta_L = T_L. \quad (24) \]

Equating the expressions for \( T_L \) obtained from (19) and (24),

\[ \frac{nK_t (V_{app} - K_b s n \theta_L)}{L_s + R} = [(J_L + J_m n^2)s^2 + B_m n^2 s] \theta_L, \quad (25) \]

\[ V_{app} - K_b s n \theta_L = \frac{L_s + R}{nK_t} [(J_L + J_m n^2)s^2 + B_m n^2 s] \theta_L, \quad (26) \]

\[ V_{app} = \frac{L_s + R}{nK_t} [(J_L + J_m n^2)s^2 + B_m n^2 s] \theta_L + K_b s n \theta_L \]
\[
\theta_L = \frac{1}{nK_e} \left[ \frac{(L_s + R)(J_L + J_m n^2)s^2 + B_m n^2 s}{(L_s + R)(J_L + J_m n^2)s^2 + B_m n^2 s + K_b s n} \right] \theta_L.
\]  

Equation (28) represents the transfer function of this model. Figure 10 is an image of the open-loop 1 DoF model built on Simulink based on the above derivation.

**Figure 10.** Open-loop Simulink model developed based on derived analytical model.

### 3.4 Closed-Loop Analytical Model

The transfer function for a closed-loop model is similar to the one derived for the open-loop model. In addition, the closed-loop model contains a PID controller for the DC motor.

Error signal \( e \) is given by

\[
e = \theta_D - \theta_L,
\]  

where \( \theta_D \) is the reference input angle. The input voltage then becomes,
\[ V_{app} = K_p e + K_i \int e \, dt + K_d \dot{e}. \] (30)

From equation (28),
\[
\theta_L = \frac{1}{V_{app}} \cdot \frac{(Ls + R)}{nK_t} \cdot \frac{1}{((J_L + J_m n^2) s^2 + B_m n^2 s) + K_b s n}.
\]

Figure 11 is an image of the closed-loop 1 DoF model built on Simulink based on equations (28) and (30).

**Figure 11.** Closed-loop Simulink model developed based on derived analytical model. The angular position output of the link feeds back to the controller to establish a control loop.

### 3.5 PID Parameters using Root Locus Method

#### 3.5.1 For 1 DoF System

In this model, PID controllers control the actuators, whose parameters are determined with the root locus method. Root locus plot is a graphical representation of how the roots of a closed-loop system change with different gains. It is an effective way of determining and tuning the stability of a system. The characteristic equation of the system provides the roots of a system, called poles and zeros. Poles and zeros are the frequencies at which the denominator and numerator of the
characteristic equation of the system respectively become zero. Figure 12 is a block diagram representing the control system.

**Figure 12.** Block diagram representing transfer functions of components in a closed-loop system with unity feedback for single degree-of-freedom model.

The aim is to design a system that has a settling time ($T_s$) of 0.15 s, an overshoot ($\%OS$) of 5% and zero steady state error (SSE) with a step input. With the given amount of $\%OS$, the damping ratio $\zeta$ can be calculated using the following relation:

$$
\zeta = \sqrt{\frac{\ln^2 \left(\frac{\%OS}{100}\right)}{\pi^2 + \ln^2 \left(\frac{\%OS}{100}\right)}} = 0.69.
$$

The natural frequency ($\omega_n$) of the system is found using the settling time and damping factor.

$$
T_s = \frac{4}{\omega_n \zeta},
$$

$$
0.15 = \frac{4}{0.69 \omega_n},
$$

From which we get,

$$
\omega_n = 57.97 \text{ rad/s}.
$$

Desired pole locations are then determined to be

$$
-\omega_n \zeta \pm i \omega_n \sqrt{1 - \zeta^2} = -40 \pm 42i.
$$
The first step to finding PID parameters in a system using the root locus method is to find the transfer function representing the system. In this model the controller regulates the working of the actuator. In this section, the calculations involved in finding the PID parameters for a single degree-of-freedom system have been demonstrated. The controller parameters for models with additional number of links are also found in a similar manner.

To begin with this method, the transfer function of the motor is first found. From equation (28), the transfer function is obtained as,

\[
\frac{\theta_L}{V_{app}} = \frac{1}{(L_s + R)(nK_t)((J_L + J_m n^2)s^2 + B_m n^2s) + K_b sn}
\]

For this model, the motor parameters are as listed in table 3:

**Table 3.** List of Motor Parameters [22].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moment of inertia of the rotor (J)</td>
<td>(3.2284 \times 10^{-6} \text{ kgm}^2)</td>
</tr>
<tr>
<td>Motor viscous friction constant (B)</td>
<td>(3.5077 \times 10^{-6} \text{ Nms})</td>
</tr>
<tr>
<td>Electromotive force constant ((K_b))</td>
<td>(0.0274 \text{ V/rad s}^{-1})</td>
</tr>
<tr>
<td>Motor torque constant ((K_t))</td>
<td>(0.0274 \text{ N m/A})</td>
</tr>
<tr>
<td>Electric resistance ((R))</td>
<td>(4 \Omega)</td>
</tr>
<tr>
<td>Electric inductance ((L))</td>
<td>(2.75 \times 10^{-6} \text{ H})</td>
</tr>
</tbody>
</table>
Table 4 gives the dimensions of the 1 DoF Simscape model.

**Table 4. Dimensional Parameters of Single Degree-of-Freedom Robot Model.**

<table>
<thead>
<tr>
<th>Part/Distance</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radius of Pulley</td>
<td>0.5 cm</td>
</tr>
<tr>
<td>Height of Pulley</td>
<td>0.5 cm</td>
</tr>
<tr>
<td>Length of Link</td>
<td>10 cm</td>
</tr>
<tr>
<td>Breadth of Link</td>
<td>1 cm</td>
</tr>
<tr>
<td>Thickness of Link</td>
<td>1 cm</td>
</tr>
<tr>
<td>Distance between the centers of Driver Pulley and Driven Pulley</td>
<td>10 cm</td>
</tr>
</tbody>
</table>

Here, the gear ratio is 140. Using these dimensions, the inertia of the single degree-of-freedom arm is 0.0010 kg m² and the open-loop transfer function of the plant is,

\[
G_{sys} = \frac{3.836}{s\left((1.768 \times 10^{-7})s^2 + 0.2571s + 14.99\right)}.
\]  

Equating the denominator to zero gives the poles of this system, which are 0, −58.31 and −1.454x 10⁶. The system has no zeros.

The characteristic equation representing a PID controller is given by,

\[
G_c = K_p + \frac{K_i}{s} + K_ds,
\]  

where, \(K_p\) is the proportional gain and gets multiplied directly with the error signal, \(K_i\) is the integral gain and is multiplied with the integral of the signal, \(K_d\) is the derivative gain and takes the derivative of the signal. This characteristic equation can be further simplified as shown.
\[ G_c = K_p + \frac{K_i}{s} + K_d s = \frac{s^2 + K_p s + K_i + K_d s^2}{s} = \frac{K_d(s^2 + K_p s + K_i)}{s}. \] (38)

which can be equivalently written in a generic factorized form, that is,

\[ G_c = \frac{K_d(s^2 + \frac{K_p}{K_d} s + \frac{K_i}{K_d})}{s} = K(s + a)(s + b). \] (39)

The open-loop transfer function of the control system is obtained as,

\[ G_{open} = G_c \cdot G_{sys} = \frac{3.836K(s + a)(s + b)}{s^2((1.768x10^{-7})s^2 + 0.2571s + 14.99)}. \] (40)

The open-loop transfer function is used to determine the type of control system. In the above equation, the highest power of the \(s\) term in the denominator is 2. This is referred to as a type 2 system. This kind of a system has no SSE for a step input. The parameters \(a, b\) and \(K\) need to be evaluated to obtain the PID equation.

Using the root locus method, the parameters for a PD controller are first found, followed by finding values for a PI controller, and then these are combined. Designing a PD controller implies adding a zero to the system. In the generic transfer function equation for the controller, this refers to the \(K(s + a)\) part, where \(K\) is for proportional control and \(-a\) is the zero that is added to the system. Hence, to design the PD controller, the first step is to find \(a\). The method used here is the graphical method. The open loop poles and desired pole locations are plotted on the \(s\)-plane, and \(a\) is plotted on the negative real axis at a random point. Figure 13 is a diagram with a rough representation of the roots of the system and is used to solve for the unknown variables \(a\) and \(b\).
**Figure 13.** Figure representing the problem of finding PID parameters for a single degree-of-freedom system using root locus. The red crosses on the real axis indicate the poles. The green circle represents the zero to be added. Pole and zero angles are represented with $\theta$ and pole and zero lengths are marked by $L$. The figure is not drawn to scale.

A property of the root locus plot is that at any point on the root locus, the difference between all the zero angles and all the pole angles is $-180^\circ$. All angles are measured anti-clockwise from the real axis. Therefore, from figure 13, this implies that,

$$\sum \theta_z - \sum \theta_p = \theta_1 - (\theta_2 + \theta_3 + \theta_4) = -180^\circ. \quad (41)$$

From the above diagram,

$$\theta_2 = \tan^{-1} \left( \frac{42}{(1.454 \times 10^6) - 40} \right) = 1.655 \times 10^{-3}^\circ, \quad (42)$$

$$\theta_3 = \tan^{-1} \left( \frac{42}{58.31 - 40} \right) = 66.45^\circ, \quad (43)$$
\[ \theta_4 = 180 - \tan^{-1} \left( \frac{42}{40} \right) = 133.60^\circ. \]  

(44)

Therefore,

\[ \theta_1 = \tan^{-1} \left( \frac{42}{a - 40} \right) = (-180 + 1.655 \times 10^{-3} + 66.45 + 133.60)^0, \]  

(45)

\[ \theta_1 = 20.05^0. \]  

(46)

Therefore, \( a = 155.06 \), so a zero has been added at -155.06. This is represented on a root locus plot using MATLAB. Figure 14(a) – 14(c) are the resulting root locus diagrams from MATLAB.

(a)
Figure 14. Root locus plots from MATLAB. (a) Root locus plot for open-loop system. (b) Root locus plot with PD controller (where $a$ is found). (c) Root locus plot of system with PID controller. Similarity to (b) satisfies pole-zero cancellation.
At the point closest to the desired pole location on the root locus plot of the PD controlled system, the gain $K$ is found to be approximately 1.454. This validates the value of gain found through manual calculations in the following steps.

By the graphical method, the overall gain of the system or $K_{overall}$ is given by

$$K_{overall} = \frac{\pi L_p}{\pi L_z},$$  \hspace{1cm} (47)

here, $L_p$ and $L_z$ are the lengths of the poles and zeros from the desired pole locations. From figure 13,

$$K_{overall} = \frac{L_2 L_3 L_4}{L_1}.$$  \hspace{1cm} (48)

Using Pythagoras theorem, the lengths obtained are

$$L_1 = 122.49, L_2 = 1.454 \times 10^6, L_3 = 45.82, L_4 = 58.$$  \hspace{1cm} (49)

Then,

$$K_{overall} = \frac{L_2 L_3 L_4}{L_1} = 3.154 \times 10^7.$$  \hspace{1cm} (50)

On comparing this with equation (8),

$$K_{overall} = \frac{3.836 K}{1.768 \times 10^{-7}} = 3.1435 \times 10^7,$$

$$K = 1.449.$$  \hspace{1cm} (51)

Thus, the PD controller is designed for the single degree-of-freedom model and the value of gain has been validated by comparing with the value of $K$ obtained from the root locus plot, which is approximately 1.45. Now, the PI controller is designed. This is done by introducing a pole at the origin and a zero very close to the origin so that pole-zero cancellation can take place. From the
above root locus diagram, it is established that the desired pole location does in fact lie on the root locus plot, hence further modification of the plot is not necessary.

A very small value for the zero is chosen, let \( b = -0.001 \). Thus, the transfer function of the PID controller now becomes,

\[
G_c = \frac{1.45(s + 155.06)(s + 0.001)}{s}.
\]  

(53)

The root locus for the closed loop system with PID controller is plotted on MATLAB and it is found to be very similar to the system with PD controller, confirming pole-zero cancellation. The PID parameters for the single degree-of-freedom manipulator are therefore, \( K_p = 225.46, K_i = 0.2255, K_d = 1.45 \).

### 3.5.2 For 2 DoF System

The 2 DoF system consists of two links of the same lengths connected to each other in the manner described before. Identical motors control both the links. Therefore, the motor connected to the first driver pulley handles the inertia of the entire arm, while the second motor only experiences the inertia of the second link and attached pulleys. Thus, to find the PID gains of the first motor, we repeated the above process by doubling the length of the link for the calculation of \( J_L \). Although this causes an increase in the inertia, due to the high gear ratio and a microgravity environment, this does not have significant effect on the PID values obtained. The following chapter includes the results of these simulations, which will further clarify this point.
Chapter 4 - Simulation

The PID controller parameters were calculated for each of the actuators used in the 1 DoF and the 2 DoF model as described in the previous section. The open-loop models have step input voltage, and the closed-loop models have step input angular positions. The simulation results of these models were compared with plots obtained from analytical models on MATLAB for validation.

In each case, input was provided to the revolute joint of the driver pulleys. The joint receiving the input is connected to an electromechanical DC motor model which is PID controlled. Each motor has a separate input, and the response of the model is then observed by plotting the angular position, velocity, voltage and current outputs. As with modeling and calculations, the simulations start with a basic 1 DoF model, later upgraded to a model with two degrees-of-freedom.

4.1 Simulation of 1 DoF Model with Regular Dimensions

Figures 15 and 16 are the open-loop and closed-loop responses of the 1 DoF model with the original dimensions from table 2.
Figure 15. Open-loop response plots of 1 DoF model with original dimensions. (a) Angular position vs time. (b) Angular velocity vs time. (c) Effective voltage input vs time ($\Delta V$). (d) Current vs time.
Figure 16. Closed-loop response plots of 1 DoF model with original dimensions. (a) Angular position vs time. (b) Angular velocity vs time. (c) Effective voltage input vs time. ($\Delta V$). (d) Current vs time.
4.2 Simulation of 1 DoF Model with Reduced Dimensions

Figures 17 and 18 are the responses of the 1 DoF arm after we reduced the dimensions and increased the gear ratio from 40 to 140 in order to lower the voltage and current spikes. The reduced dimensions are listed in table 2.

![Graph (a)](image)

![Graph (b)](image)
Figure 17. Open-loop response plots of 1 DoF model with reduced dimensions. (a) Angular position vs time. (b) Angular velocity vs time. (c) Effective voltage input ($\Delta V$) vs time. (d) Current vs time.
(a) 

(b)
Figure 18. Closed-loop response plots of 1 DoF model with reduced dimensions. (a) Angular position vs time. (b) Angular velocity vs time. (c) Effective voltage input vs time. (ΔV). (d) Current vs time.
4.3 Simulation of 1 DoF Model with Low-Pass Filter

Next, we added a low-pass filter with a time constant of 1 ms to the step input of the closed-loop 1 DoF model to smoothen the signal and observed that the resultant voltage and current now had much lower spikes. Figure 19 shows the effect of the addition of the filter.
**Figure 19.** Plots of changes in step input and resulting responses. (a) Effect of addition of low-pass filter on step input. (b) $\Delta V$ vs time. (c) Current vs time.

### 4.4 Simulation of 2 DoF Model

Figures 20 and 21 are the responses of the two links of the 2 DoF open and closed-loop models. These models have low-pass filters and a gear ratio of 140 for each motor.
Figure 20. Responses of both links of the 2 DoF open-loop model to step input voltage. Angular displacement ($\theta_L$), angular velocity ($\dot{\theta}_L$), $\Delta V$ and current are plotted with respect to time.
Figure 21. Responses of both links of the 2 DoF closed-loop model to step input voltage. Angular displacement ($\theta_L$), angular velocity ($\dot{\theta}_L$), $\Delta V$ and current are plotted with respect to time.
Chapter 5 - Results and Discussion

From the plots obtained from simulations of the models in the previous section, it can be inferred that the models are easily controlled with a PID controller. The results obtained from the analytical models on MATLAB are in agreement with those obtained from Simulink. We also observe that the open-loop model has a slower response as compared to the closed-loop model. While this is also true of the 1 DoF model which had the original dimensions, the high spikes in voltage and current rendered this model impractical. The first step that we took to reduce these spikes was to reduce the dimensions of the parts in the model. This reduced the spikes to some extent, but this measure alone was not enough. In addition to this, increasing the gear ratio and introducing a low-pass filter to the input helped in making these fluctuations more manageable. The low-pass filter incorporated here using a transfer function is synonymous to an RC circuit in an experimental setup. Moreover, raising the time constant, which would be equivalent to adding a larger capacitor, helps in the further reduction of such voltage spikes. Figure 22 helps demonstrate this.

![Figure 22. Voltage and Current response of 1 DoF model with time constant \( \tau = 10 \) ms.]

There are certain challenges and limitations when modeling only using Simulink. The main challenge in this thesis was to ensure appropriate spatial and rotational arrangement of all the components with respect to each other and to the world frame. Such spatial relations are much
harder to describe in a simulated environment and the design had to be modified to accommodate this. While the Belts and Cables toolbox provided a method for modeling the system of cables and pulleys implemented in these models, a major limitation of this toolbox is that the properties of the cables cannot be controlled or altered in any way, and so physical properties such as cable diameter, elasticity, specific weight, and others could not be specified. The model presented here is an ideal model, and some deviations are expected when the tests are conducted on an experimental setup. For example, we would have to consider friction, slippage of the cables and vibrations.

The main objective of this project was to put forward a viable design of a robot arm that is lightweight as compared to ones that are currently available. Table 5 shows a comparison of the weights of the lightest serial manipulators of some manufacturers with the weight of the experimental model designed in this project.

**Table 5.** Comparison of weights of various serial robots [4, 23-26].

<table>
<thead>
<tr>
<th>Model</th>
<th>FANUC LR Mate 200iD/4SH</th>
<th>YASKAWA MotoMini</th>
<th>KUKA LBR iiwa 7 R800</th>
<th>KAWASAKI RS003N Robot</th>
<th>EPSON Synthis T3 Scara Robot</th>
<th>Model presented in this thesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mechanical weight (kg)</td>
<td>19</td>
<td>7</td>
<td>22.3</td>
<td>20</td>
<td>16</td>
<td>5</td>
</tr>
</tbody>
</table>
Chapter 6 - Conclusion and Future Work

The objective of this thesis was to design and simulate a cable-driven robot arm, which could be a lighter and safer alternative for traditional serial manipulators. The thesis began with the conceptual design of a model with three degrees-of-freedom. An experimental model was built based on this design, but due to the pandemic, experimentation with this model could not be pursued. We changed the approach to this thesis by using virtual models and simulations to achieve the task. This was done with the help of MATLAB and Simulink. Following this, the design of the model had to be modified in order to accommodate for certain limitations of constructing it completely virtually. The model is built in a microgravity environment. It is seen from simulations that the manipulator easily controlled using a simple PID controller. The controllers were tuned by using the root locus method, which was observed to yield better results for the models as compared to the PID tuner app on Simulink.

Although the final Simscape model uses dimensions that are much smaller than the human arm, this method of modeling on Simscape can be used as a design tool to test the responses of prototypes, especially for microgravity environments. This model also helps us study the influence of changing properties such as inertia, gear ratio and the addition of a filter.

In the future, its functionalities can surely be extended by the inclusion of the effects of gravity, using different end effectors on the final link, and accurate control of such end effectors to perform different tasks such as pick-and-place, aid to differently abled people, for household chores, etc. A further challenge would be to improve the accuracy of the motion when performing micro-scale applications such as in surgery or micromanufacturing, since the precision depends not only on the control strategy of the motors but also the elastic behavior of the cables and the friction between the pulleys and cables. The previous point also provides the next scope of improvement in such a model, which is by developing better and more accurate modelling of the elastic properties of the cables and thereafter integrating them into the Simscape model.
References


Appendix

MATLAB code for open and closed-loop 2 DoF system including PID calculation

%Motor Parameters
Jm = 3.2284e-6; %rotor inertia (kg m^2)
Bm = 3.5077e-6; %viscous friction constant (Nm/(rad/s))
Kt = 0.0274; %motor torque constant (Nm/A)
Kb = 0.0274; %electromotive force constant (Nm/V)
R = 4; %resistance (Ohms)
L = 2.75E-6; %inductance (H)
n = 140; %gear ratio

%Link Parameters - motor 1
JL1= 8.28733e-05 + (0.0248*0.095*0.095) + (0.000486947)*2; %Based on dimensions in Simulink

%Range of Time for Simulation
Ts=1e-4;
Tf=5;
T=0:Ts:Tf;

%Open loop transfer function
s=tf('s')
Gsyst=1/((L*s+R)/(n*Kt)*((JL1+Jm*n^2)*s^2+Bm*n^2*s)+Kb*n*s)
step_amp = 0.1;
%Plot of angle

[Y1,T]=step(step_amp*Gs1,T);
figure
plot(out.two_Position1.time-1,out.two_Position1.signals.values,'c','LineWidth',2)
hold on
plot(T,Y1,'r--','LineWidth',1.5)
xlim([-1 4])
xlabel('Time (s)','FontSize',15)
ylabel('\theta_L (rad)','FontSize',15)
legend('\theta_L from Simulink','\theta_L from MATLAB')

%Plot of velocity

[Ydot1,T]=impulse(step_amp*Gs1,T);
figure
plot(out.two_Velocity1.time-1,out.two_Velocity1.signals.values,'c','LineWidth',2)
hold on
plot(T,Ydot1,'r--','LineWidth',1.5)
xlim([-1 4])
xlabel('Time (s)','FontSize',15)
ylabel('$\dot{\theta}_L$ (rad/s),','interpreter','latex','FontSize',15)
legend('$\dot{\theta}_L$ from Simulink','$\dot{\theta}_L$ from MATLAB')
%Voltage and current from code

%DeltaV = Vapp - Kb*s*(theta_m)

%Define V app

t=0:Ts:Tf;
Vapp = step_amp*heaviside(t);

Vsub1 = Kb*Ydot1'/n;

DelV1 = Vapp - Kb*Ydot1'*n;

Current1 = DelV1/R;

%Voltage and Current plots

figure
plot(out.two_Voltage1.time-1,out.two_Voltage1.signals.values,'LineWidth',2)
hold on
plot(t,DelV1,'--','LineWidth',1.5)
xlim([-1 4])
xlabel('Time (s)','FontSize',15)
ylabel('\Delta V (V)','FontSize',15)
legend('\Delta V from Simulink','\Delta V from MATLAB')

figure
plot(out.two_Voltage1.time-1,out.two_Voltage1.signals.values/R,'LineWidth',2)
hold on
plot(t,Current1,'--','LineWidth',1.5)
xlim([-1 4])
xlabel('Time (s)','FontSize',15)
ylabel('Current (A)','FontSize',15)
legend('Current from Simulink','Current from MATLAB')

%Link Parameters - motor 2
JL2 = 1.04367e-05 + (0.0124*0.045*0.045) + (0.000486947)*2; %Based on dimensions in Simulink

%Open loop transfer function
s=tf('s')
Gsys2=1/((L*s+R)/(n*Kt)*((JL2+Jm*n^2)*s^2+Bm*n^2*s)+Kb*n*s)

%Plot of angle
[Y2,T]=step(step_amp*Gsys2,T);
figure
plot(out.two_Position2.time-1,out.two_Position2.signals.values,'c','LineWidth',2)
hold on
plot(T,Y2,'r--','LineWidth',1.5)
xlim([-1 4])
xxlabel('Time (s)','FontSize',15)
ylabel('	heta_L (rad)','FontSize',15)
legend('	heta_L from Simulink','	heta_L from MATLAB')
%Plot of velocity

[Ydot2,T]=impulse(step_amp*Gsys2,T);

figure

plot(out.two_Velocity2.time-1,out.two_Velocity2.signals.values,'c','LineWidth',2)

hold on

plot(T,Ydot2,'r--','LineWidth',1.5)

xlim([-1 4])

xlabel('Time (s)','FontSize',15)

ylabel('$\dot{\theta}_L$ (rad/s)','interpreter','latex','FontSize',15)

legend('$\dot{\theta}_L$ from Simulink',[$'$\dot{\theta}_L$ from MATLAB'],$'interpreter','latex')

%Voltage and current from code

%DeltaV = Vapp - Kb*s*(theta_m)

%Define Vapp

t=0:Ts:Tf;

Vapp = step_amp*heaviside(t);

Vsub2 = Kb*Ydot2'/n;

DelV2 = Vapp - Kb*Ydot2'*n;

Current2 = DelV2/R;

%Voltage and Current plots

figure
plot(out.two_Voltage2.time-1,out.two_Voltage2.signals.values,'LineWidth',2)
hold on
plot(t,DelV2,'--','LineWidth',1.5)
xlim([-1 4])
xlabel('Time (s)','FontSize',15)
ylabel('
Delta V (V)','FontSize',15)
legend('
Delta V from Simulink', '
Delta V from MATLAB')

figure
plot(out.two_Voltage2.time-
1,out.two_Voltage2.signals.values/R,'LineWidth',2)
hold on
plot(t,Current2,'--','LineWidth',1.5)
xlim([-1 4])
xlabel('Time (s)','FontSize',15)
ylabel('Current (A)','FontSize',15)
legend('Current from Simulink', 'Current from MATLAB')

%Root locus and PID - motor 1
poles1 = pole(Gsys1)
figure
rlocus(Gsys1) %open loop root locus

% with PD controller
a1 = 153.3; %adding a zero to the system for PD control. 'a' is obtained through trigonometry
G_PD1 = Gsys1*(s+a1); %open loop system with a PD controller
figure
rlocus(G_PD1)
title("With PD controller")

b1 = 0.001; %adding a zero to the system for PI control. Value of 'b' is close to zero to ensure pole-zero cancellation
G_PID1 = Gsys1*(s+a1)*(s+b1)/s ;%defining a transfer function which has the new zero and pole accounted for
figure
rlocus(G_PID1) %this root locus should be similar to the one obtained when the system has a PD controller to confirm pole-zero cancellation
title("With PI controller")

%%
K1 = 1.48; %this can be obtained both graphically and from the root locus diagram
%PID parameters
Kp1 = K1*(a1+b1)
Ki1 = K1*a1*b1
Kd1 = K1
G_cont1 = Kp1 + Kd1*s + (Ki1/s);
G_open1 = G_cont1*Gsys1;
G_closed1 = feedback(G_open1,1);

%%

%Plot of angle
[Y_PID1,T]=step(step_amp*G_closed1,T);
figure
plot(out.twoclosed_Position1.time-1,out.twoclosed_Position1.signals.values,'c','LineWidth',2)
hold on
plot(T,Y_PID1,'r--','LineWidth',1.5)
xlim([-1,4]);
ylim([0,0.15]);
xlabel('Time (s)','FontSize',15)
ylabel('
theta_L (rad)','FontSize',15)
legend('
theta_L from Simulink','
theta_L from MATLAB')

%%

%Plot of velocity
[Ydot_PID1,T]=impulse(step_amp*G_closed1,T);
figure
plot(out.twoclosed_Velocity1.time-1,out.twoclosed_Velocity1.signals.values,'c','LineWidth',2)
hold on
plot(T,Ydot_PID1,'r--','LineWidth',1.5)
xlabel('Time (s)','FontSize',15)
xlim([-1,4])
ylabel('$\dot{\theta}_L$ (rad/s)','interpreter','latex','FontSize',15)
legend('$\dot{\theta}_L$ from Simulink','$\dot{\theta}_L$ from MATLAB','interpreter','latex')

%%

%Voltage and Current plots
figure
plot(out.twoclosed_Voltage1.time-1,out.twoclosed_Voltage1.signals.values)
xlabel('Time (s)','FontSize',15)
ylabel('\Delta V (V)','FontSize',15)
legend('\Delta V with PID controller from Simulink')

figure
plot(out.twoclosed_Voltage1.time-1,out.twoclosed_Voltage1.signals.values/R)
xlabel('Time (s)','FontSize',15)
ylabel('Current (A)','FontSize',15)
legend('Current with PID controller from Simulink')

%%

%Root locus and PID - motor 2
poles2 = pole(Gsys2)
figure
rlocus(Gsys2) %open loop root locus

% with PD controller
a2 = 155.06; %adding a zero to the system for PD control. 'a' is obtained through trigonometry
G_PD2 = Gsys2*(s+a2); %open loop system with a PD controller
figure
rlocus(G_PD2)
title("With PD controller")

b2 = 0.001; %adding a zero to the system for PI control. Value of 'b' is close to zero to ensure pole-zero cancellation
G_PID2 = Gsys2*(s+a2)*(s+b2)/s; %defining a transfer function which has the new zero and pole accounted for
figure
rlocus(G_PID2) %this root locus should be similar to the one obtained when the system has a PD controller to confirm pole-zero cancellation
title("With PI controller")

%%
K2 = 1.454; %this can be obtained both graphically and from the root locus diagram
%PID parameters
Kp2 = K2*(a2+b2)
Ki2 = K2*a2*b2
Kd2 = K2
G_cont2 = Kp2 + Kd2*s + (Ki2/s);
G_open2 = G_cont2*Gsys2;
G_closed2 = feedback(G_open2,1);

%%
%Plot of angle
[Y_PID2,T]=step(step_amp*G_closed2,T);
figure
plot(out.twoclosed_Position2.time-1,out.twoclosed_Position2.signals.values,'c','LineWidth',2)
hold on
plot(T,Y_PID2,'r--','LineWidth',1.5)
xlim([-1,4]);
xlabel('Time (s)','FontSize',15)
ylabel('	heta_L (rad)','FontSize',15)
legend('	heta_L from Simulink','	heta_L from MATLAB')

%%
%Plot of velocity
[Ydot_PID2,T]=impulse(step_amp*G_closed2,T);
figure
plot(out.twoclosed_Velocity2.time-1,out.twoclosed_Velocity2.signals.values,'c','LineWidth',2)
hold on
plot(T,Ydot_PID2,'r--','LineWidth',1.5)
xlabel('Time (s)','FontSize',15)
xlim([-1,4])
ylabel('$\dot{\theta}_L$ (rad/s)','interpreter','latex','FontSize',15)
legend('$\dot{\theta}_L$ from Simulink','$\dot{\theta}_L$ from MATLAB','interpreter','latex')

%%

%Voltage and Current plots
figure
plot(out.twoclosed_Voltage2.time-1,out.twoclosed_Voltage2.signals.values)
xlabel('Time (s)','FontSize',15)
ylabel('$\Delta V$ (V)','FontSize',15)
legend('$\Delta V$ with PID controller from Simulink')

figure
plot(out.twoclosed_Torque2.time-1,out.twoclosed_Torque2.signals.values/(n*Kt))
hold on
plot(out.twoclosed_Voltage2.time-1,out.twoclosed_Voltage2.signals.values/R)
xlabel('Time (s)', 'FontSize', 15)
ylabel('Current (A)', 'FontSize', 15)
legend('Current - Torque/Kt', 'Current - Voltage/R')