Design of Adjustable Rehabilitative-Assisted Mechanism for Rodents Suffering from Spinal Cord Injury

A Major Qualifying Project report

submitted to the faculty of

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

degree of Bachelor of Science

By:

Yanbo Dai, Xianhan Jia, Zhengrong Tang, Haojun Yan

Advisor:

Prof. Yuxiang Liu, Prof. Xiangnan Kong, WPI

External Collaborator:

Prof. Wei Wu, Indiana University

Date:

2023/3/24

Abstract

This report introduces a novel design of the rehabilitative-assisted mechanism that provides active locomotion training for rodents affected by spinal cord injury (SCI). This mechanism is adjustable and can accommodate rodents of different sizes, which distinguishes it from existing mechanisms that lack this feature. In addition, its modular design facilitates maintenance and updates. We believe that our work has the potential to encourage the future development of rehabilitative-assisted mechanisms for human patients suffering from SCI.

Acknowledgments

We are grateful for the invaluable assistance we have received throughout the course of this project.

First and foremost, we would like to extend our gratitude to Prof. Yuxiang Liu for his unwavering support and guidance on our project.

Then, we also wish to extend our heartfelt thanks to our co-advisor, Prof. Xiangnan Kong, for his invaluable guidance and professional advice on the computer science aspect of our project, particularly during the struggling A term.

Lastly, we would like to extend our sincerest thanks to our external collaborator from Indiana University, Prof. Wei Wu, for generously dedicating his time and expertise to our project.

Authorship

Chapter / Section #	Author(s)
Ch. 1	Xianhan & Yanbo & Zhengrong
Ch. 2	
2.1.1	Haojun
2.1.2	Zhengrong
2.1.3	Haojun
2.2	Haojun
Ch. 3	
3.1	Haojun
Sec. 3.2	
3.2.1	Haojun
3.2.2	Yanbo
Sec. 3.3	
3.3.1	Haojun
3.3.2	Haojun
Sec. 3.4	
3.4.1	Haojun
3.4.2	Haojun
3.4.3	Haojun
3.4.4	Yanbo
3.4.5	Haojun
Sec. 3.5	Haojun
Sec. 3.6	
3.6.1	Zhengrong
3.6.2	Yanbo & Xianhan
Sec. 3.7	Yanbo & Haojun
Ch. 4	Yanbo
Ch.5	Haojun
Ch.6	Xianhan

Table of Contents

Abstract I
Acknowledgments II
Authorship III
Table of Contents IV
List of Figures VI
1. Introduction 2
2. Literature Review 4
2.1. Rehabilitation-Assisted Exoskeleton 4
2.1.1. The Early Design with Flaws: The Rat Stepper 5
2.1.2. New Design, New Improvements in Mobility: The Iron Rat 6
2.1.3. Revolutionary Precision: Joint-Synergy-Based Exoskeleton 8
2.2. Summary 10
3. Design of the Mechanism 11
3.1. Statement of the Design Loop 11
3.2. Goal, Requirements, and Design Specifications 12
3.2.1. Goal, Requirements, and Design Specifications from the ME Side 12
3.2.2. Goal, Requirements, and Design Specifications from the CS Side 13
3.3. 1st Design Iteration 14
3.3.1. Mechanical Design 14
3.3.2. Evaluation of the Mechanical Design 21
3.4. 2nd Design Iteration 23
3.4.1. Exploration of Adjustable Design 23
3.4.2. Mechanical Design 26
3.4.3. Evaluation of the Mechanical Design 30
3.4.4. Electrical Design 31
3.4.5. Summary 34
3.5. 3rd Design Iteration 35
3.6. 4th Design Iteration 36
3.6.1. Implementation of the Encoder 36
3.6.2. Encoder coding 41
3.7. Summary 42

4. Review of Failure 43
4.1. Potential of Machine Learning 43
4.1.1. A Term Research Overview 43
4.1.2. Obstacles Encountered 43
4.1.3. Continuation of Research Direction 43
4.2. Delays in Manufacturing and Testing 43
4.3. Team Science 44
5. Social implications 45
6. Conclusion 45
References 46
Appendix A: Code for Encoder Control 48

List of Figures

Fig. 2–1: Some designs of the overground lower limb exoskeleton [12].----------4

Fig. 2–2: (a) The rat stepper system; (b) The robotic arm of the rat stepper system [14]. -5

Fig. 2–3: Conceptual sketch of the Iron Rat system [15].----------7

Fig. 2–4: The Rat Backpack is attached to the lower body part [15].----------7

Fig. 2–5: Planer covariation of the hind limb joints of the rat. The x, y, and z-axis of the coordinate system represent the real-time value of the knee, hip, and ankle angle respectively. The hip angle is the angle between the femur bone and the horizontal axis. The knee angle is the angle between the tibia bone and the femur bone. The ankle angle is the angle between the tibia bone and the foot [16]. ----------8

Fig. 2–6: 2D schematic of the joint-synergy-based exoskeleton [16].----------9

Fig. 2–7: Joint angle trajectories (Mechanical Reproduction vs. Live Rat Sample) [16]. 10

Fig. 3–1: 3D model of the support table (different view perspectives). ---------- 14

Fig. 3–2: Flowchart of the gait pattern extraction process. ---------- 15

Fig. 3–3: Graphical user interface (GUI) of Tracker. The two crossed purple lines represent the Cartesian coordinate system that all the extracted data w.r.t. The blue line segment represents the calibration stick that converts the extracted data from a pixel-based unit to a real-world length dimension unit (cm, inch, etc.).---------- 15

Fig. 3–4: Extracted position data of the mouse's hind limb joints.---------- 16

Fig. 3–5: Relative motion of the calf's midpoint w.r.t the hip | Frame 258-267. ---------- 17

Fig. 3–6: 2D sketch and simulation of the four-bar linkage system. Length of each link: 5; 15; 17,5; 12.5 (Crank; Coupler; Rocker; Ground) [mm]. (b) 1.0: 3.0: 3.5: 2.5 linkage model [24, p. 316].---------- 18

Fig. 3–7: Trajectory of the four-bar linkage system.---------- 18

Fig. 3–8: 3D model of the linkage system. ---------- 19

Fig. 3–9: 3D printed support table. (a) Isometric view; (b) Front view; (c) Side view; (d) Detailed view of the connection between the motor and the four-bar linkage mounting plate ---------- 21

Fig. 3–10: Size data of the typical experimental mouse. (A: 6-10 cm; B: 3-5 cm; C: No exact data due to the measurement difficulty D: around 2 cm) ---------- 23

Fig. 3–11: Sketch of the adjustable mechanism.---------- 24

Fig. 3–12: Isometric view of the 3D model of the adjustable mechanism. ---------- 26

Fig. 3–13: Screw fixation for the NEMA-8 stepper motor.---------- 27

Fig. 3–14: Screw fixation design. (a) between the four-bar mounting plate and the I-beam slider; (b) between the I-beam slider and the I-beam; (c) between the E-shaped end cap and the front/black plate. ---------- 27

Fig. 3–15: The adjustable range of the four-bar linkage module along x (horizontal) & y (vertical) directions. ---------- 29

Fig. 3–16: Design flaws of the second iteration.---------- 30

Fig. 3–17: Code for the function of angle replication.---------- 32

Fig. 3–18: Code for the speed control of the stepper motor.---------- 33

Fig. 3–19: Main contribution of the third design iteration. ---------- 35

Fig. 3–20: The switching module design contains the encoder. ---------- 37

Fig. 3–21: The gear system connects the motor and the encoder. The three gears have teeth counts of 24, 20, and 28, respectively, and are arranged from left to right. The gear ratios from left to right are 20:24 = 5:6 and 28:20 = 7:5. ---------- 37

Fig. 3–22: 3D printed well-assembled motor module with encoder design.---------- 38

Fig. 3–23: The fully assembled mechanism with wiring. ---------- 39

Fig. 3–24: Final wiring diagram between devices and breakout boards, using Fritzing. Since there are only two pins available on Arduino Uno Rev.3 for hardware interrupt, the two encoders also use interrupts, and the team didn't figure out a way to utilize it without the interrupt pin in time, the Qwiic Twist digital RGB Inter-Integrated Circuit (I2C) encoder was not included in the final design. ---------- 39

Fig. 3–25: 2D rendering of final wiring between devices and breakout boards, using Fritzing. The Qwiic Twist digital RGB I2C encoder was not included in the final design. ---------- 40

Fig. 3–26: Flowchart of final software design. The logic used to handle the input of the input knob (dotted line section) was not included in the final design.---------- 40

1. Introduction

Spinal Cord Injury (SCI) is a subcategory of body part injury with damage to any part of the spinal cord or nerves at the end of the spinal canal [1, p. 6]. Strength, sensation, and other body functions below the location of the damage are often permanently changed by the injury. People who injured their spinal cord might feel the effects of injury not only physically, but also mentally, emotionally, vocationally, and socially [2], [3]. These important issues are problematic around the world and may persist over several decades. SCI not only impacts the lives of patients but also impacts the family caregivers. Depression, anxiety, physical symptoms, and reduced satisfaction with life are issues for primary family caregivers of patients with SCI. Isolation, loss of identity, and role changes were also regularly reported as negative outcomes of caregiving for someone with an SCI [4].

Each year the number of SCI patients keeps increasing. Based on the World Health Organization, around 250,000 to 500,000 people suffer from SCI every year in the world [1]. According to the 2020 datasheet from National Spinal Cord Injury Statistical Center, approximately 183,000 to 231,000 of the US population experience lifelong disabilities due to SCI [5]. The two main causes of SCI are motor vehicle accidents and catastrophic falls among all reasons, 38%, and 30% respectively among patients in the United States [6]. What's more, nearly half (46.7%) of all injuries occurred between the ages of 16 and 30 [5].

Patients with SCI often face tremendous financial difficulties. Among SCI patients, the injuries are work-related. Of the 15,175 available records, 9.4% had a work-related SCI, which causes a loss of job opportunities and income while carrying a huge financial burden at the same time [5]. Total direct costs of caring for individuals with SCI exceed $7 billion per year in the USA [3].

Nearly half (48.5%) of the population with spinal cord injury suffered mental health problems of depression (37%), anxiety (30%), clinical-level stress (25%), or post-traumatic stress disorder (8.4%) [7]. Even with such devastating effects on daily living, many were injured very early in life. Over half of the patients were injured before 30 years old. Apart from age, SCI patients also often experience impacts on relationships with others. Although many patients found love after SCI, divorce rates are also increasing with the number of years after injury [5].

Recovery from SCI is difficult and often incomplete. Although there are various treatments for SCI, including surgery [8], locomotion training [9], and some experimental therapies like electrical simulation [10], gait training is still the primary way of rehabilitation [11]. Although proven to be beneficial, during the locomotion training process, the postures of the patient's lower limb are usually manually adjusted by the therapist on a per-patient basis, which is resource-consuming. To address such difficulties, the MQP team, which consists of undergraduate students from both Mechanical Engineering and Computer Science majors, aims to develop a rehabilitative-assisted mechanism that can provide repeated active gait training for paralyzed rodents suffering from SCI. We will combine

both mechanical and electrical design to manufacture the mechanism, with the goal of eliminating the need for manual adjustments to the patient's postures in traditional SCI treatment approaches. Our choice to fabricate a mechanism for rodents is based on the fact that medical research in general uses rodent experiments in the theoretical validation stage. Hopefully, the expected result of this project could inspire future interdisciplinary studies related to SCI.

The remainder of the page is intentionally left blank.

2. Literature Review

This chapter will be divided into two halves. The first half will review different types of rehabilitative-assisted mechanisms (exoskeleton) that scholars developed for paralyzed rodents from the perspective of mechanical design. The second half concentrate on the design principles of the control system that drives the exoskeleton.

2.1. Rehabilitation-Assisted Exoskeleton

The common protocol of locomotion training therapy always contains the rehabilitation-assisted exoskeleton [12]. Patients with lower limb movement disorders can gradually relearn how to move with the aid of the exoskeleton. Fig. 2-1 shows some designs of the overground lower limb exoskeleton, which is one major class of exoskeletons. The lower limb exoskeleton is a linkage mechanism that connects the joints of the lower limbs (hip, knee, ankle, etc.) and controls their movements [12]. In recent years, not only the gait-assisted exoskeleton was used in the recovery process of diseases affecting the movement of the lower limbs, but also served as a study tool for neuroscientists to better understand the neurorehabilitation process. Due to ethical reasons, biomedical researchers usually could only conduct experiments on animals. In this situation, rodents, i.e., rats and mouses, are typically the first choice because of their genetic similarity to humans [13]. Thus, a lot of scholars [14]–[16] have already designed and fabricated a variety of exoskeletons for either the rat or mouse in the course of their research. The following sections of this chapter will provide a general picture of the structure and the design logic of these exoskeletons.

Fig. 2–1: Some designs of the overground lower limb exoskeleton [12].

2.1.1. The Early Design with Flaws: The Rat Stepper

The rodent model has been commonly employed in the study of spinal cord injury [17]. However, it is difficult to manually adjust rodents' leg postures during locomotion training. To address this issue, in 2005, a neuroscience research team affiliated with IEEE developed a robotic device named "Rat Stepper" [14]. The structure of the device is shown in Fig. 2-2 (a), which consists of a treadmill, a body weight support system, and a robotic arm. The body weight support (BWS) system could offer an upward supportive force [18], which is extremely important for rats that just experienced SCI since they lack the strength to support their bodies. The treadmill could serve as an obstacle to the back of the rat, which stimulates it to move forward and thus quickens its walking. The most essential component of this apparatus is the robotic arm that connects to the lower shank of the hindlimb of the rat using alligator clips (Fig. 2-2 (b)). This robotic arm consists of two linkage mechanisms (one four-bar and one five-bar) which are driven by two mechanically grounded motors. Since these two linkage mechanisms share one common link, it could produce planer two degrees-of-freedom (DoF) motions.

Even though the rat stepper could be used to either passively track the hindlimb movement patterns of the rat or actively provide gait assistance during locomotion training, as an early design, it undoubtedly has some drawbacks. The main flaw of this robotic arm is that it connects to only one point of the rat's hind limb, making it unable to fully control both the position and orientation of the hind limb. In addition, this robotic arm constrains the rat to move within the para-sagittal plane. Finally, this body weight support system uplifts the upper body of the rat and forces it to move bipedally, which is against its natural quadrupedal movement logic.

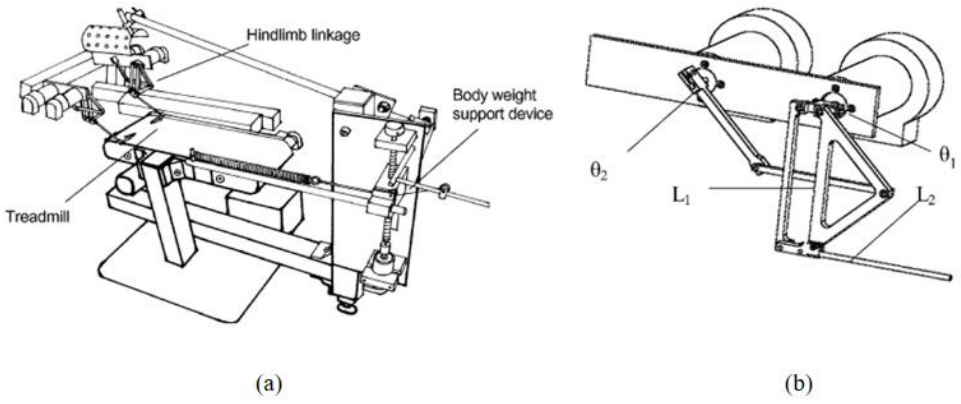

Fig. 2–2: (a) The rat stepper system; (b) The robotic arm of the rat stepper system [14].

2.1.2. New Design, New Improvements in Mobility: The Iron Rat

Although the exoskeleton provides forces to help the rodents' locomotion, it usually restricts their voluntary movement to some extent. Equipment that analyzes rodents' locomotion always creates intimidating artificial conditions such as inside a cage with a treadmill and subject to electric shock if it fails to move fast enough. Another new apparatus developed by an MIT Ph.D. student in 2015 is called the Iron Rat system, which allows maximal freedom of voluntary overground movement of the animal while providing forceful interaction to the hindlimbs. As shown in Fig. 2-3, a lightweight device is mounted on the lower back of the rat for controlling physical interaction. A tower with wiring is attached to the rat to provide weight support (Fig. 2-4). Other substantial parts, including the controller, the data storage, and the power source, are situated outside of this. The rat could move freely within the area of activity. In Fig. 2-4, two motors on each side controlled the rotation of two linkages to allow rat ankles to have a 2 DoFs movement relative to the pelvis. In general, the rat seemed to move as naturally in the mockup as when it was unconstrained. However, a partial tail dragging was observed while the rat was in the mockup. the rat is on open ground, the apparatus allows physical interaction with hindlimb movement, affecting not only the kinematics but also the dynamics of overground locomotion. Beyond what is conceivable with current equipment, the apparatus opens a wide unexplored area of potential experimentation, and the Iron Rat might help research pave the path for testing different robotic treatment methods in rats, which would ultimately lead to better physical therapy for humans [15].

The remainder of the page is intentionally left blank.

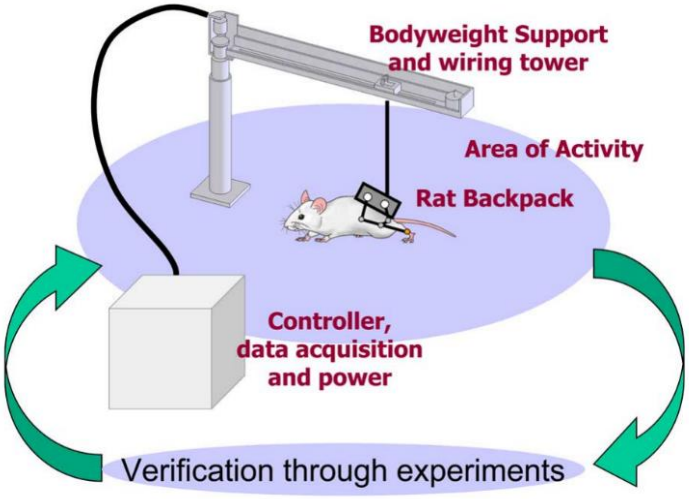

Fig. 2–3: Conceptual sketch of the Iron Rat system [15].

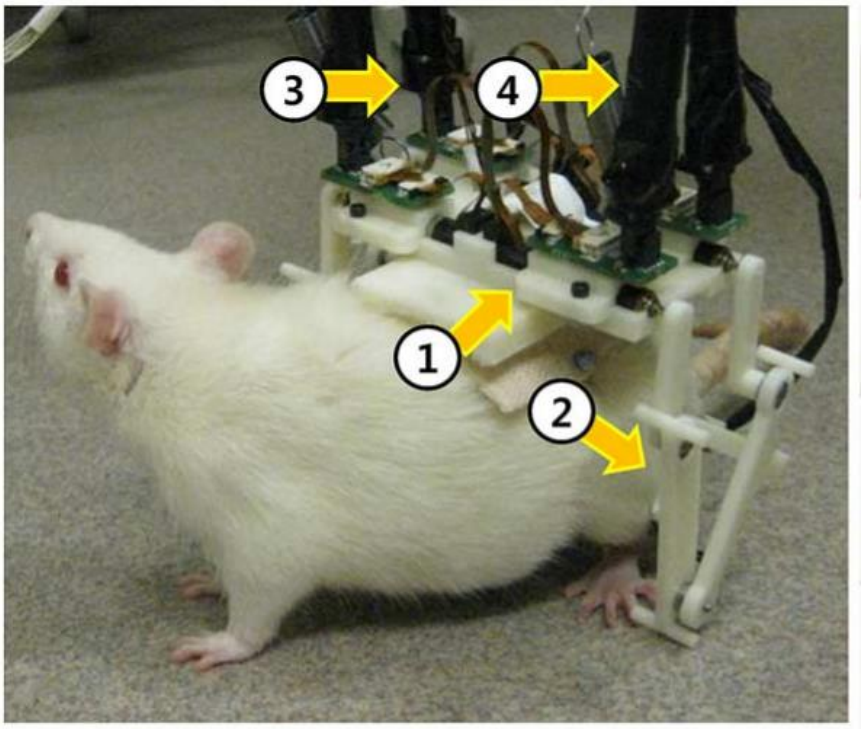

Fig. 2–4: The Rat Backpack is attached to the lower body part [15].

2.1.3. Revolutionary Precision: Joint-Synergy-Based Exoskeleton

Both the two exoskeletal mechanisms mentioned in sections 2.1.1 and 2.1.2, i.e., The Rat Stepper and Iron Rat, have an apparent flaw: they could only connect to one of the joints of the rat's hind limb, which results in their inability to fully control the position and posture of the rat's hind limb. The hind limb of the rodent contains three joints: the hip, the knee, and the ankle. Although the hind limb of the rodent has three degrees of freedom due to the combination of these three joints: translation along the x and y axes and rotation about the z-axis, it is not necessary to control all three joints to fully control the hind limb.

According to Catavitello's study, there is a kinetic synergy between the quadruped hind limb joints in their gait movements. It is possible to obtain a 2D covariance plane that depicts the relationship between the joint angles if the real-time values of the joint angles during gait are mapped onto a 3D Cartesian coordinate system (Fig. 2-5) [19]. This phenomenon is called planer covariation, which enables full control of the hind limb of the rodent by manipulating only two of its three joints. Consequently, a research team from the University of Tsukuba designed and manufactured an exoskeleton using the theory of planer covariation [16].

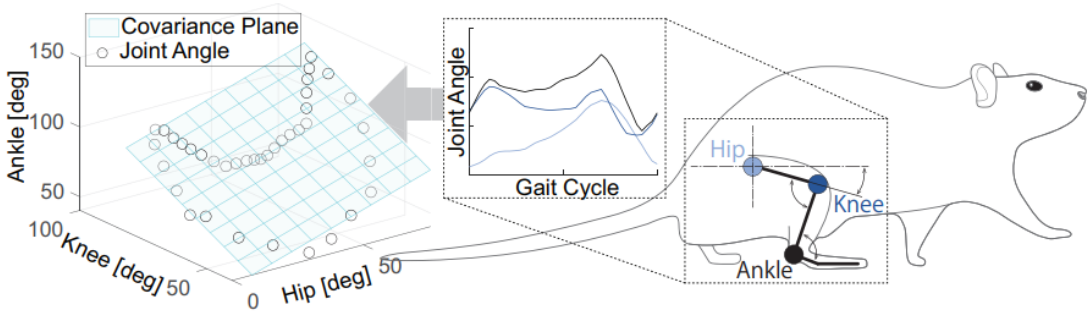

Fig. 2–5: Planer covariation of the hind limb joints of the rat. The x, y, and z-axis of the coordinate system represent the real-time value of the knee, hip, and ankle angle respectively. The hip angle is the angle between the femur bone and the horizontal axis. The knee angle is the angle between the tibia bone and the femur bone. The ankle angle is the angle between the tibia bone and the foot [16].

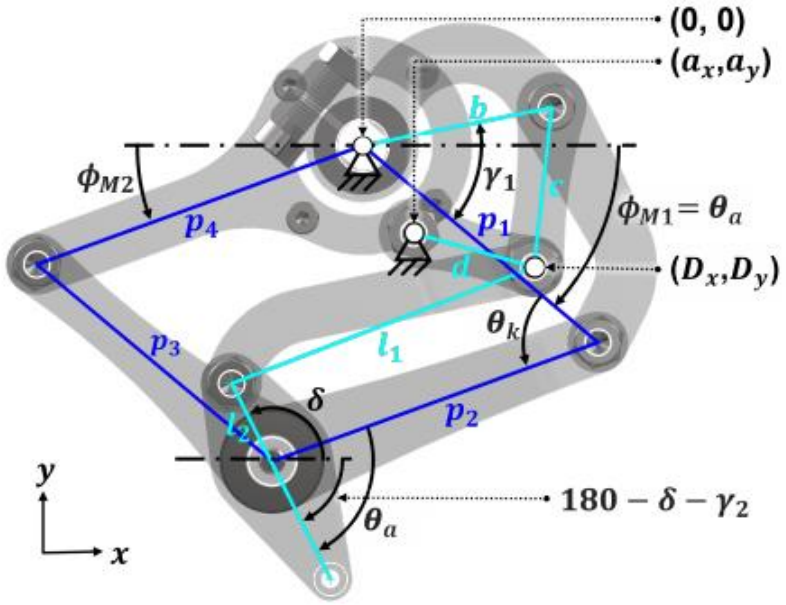

Fig. 2–6: 2D schematic of the joint-synergy-based exoskeleton [16].

As shown in Fig. 2-6, the joint-synergy-based exoskeleton is composed of two linkages: an input four-bar linkage (p_1, p_2, p_3, p_4) and an output six-bar linkage (b, c, d, l_1, l_2). The links p_1 and p_3 are set to the same length as the experimental rat's femur bone, while links p_2 and p_4 are set to have the same length as the tibia bone. In other words, the four-bar linkage has a parallelogram-like shape. By tracking the motion of the four-bar linkage, the real-time value of $\emptyset_{M1}$ and $\emptyset_{M2}$ could be obtained, which could then be used to derive the hip angle θ_h and the knee angle θ_k. These two angles (θ_h and θ_k) are then used by the output six-bar linkage to control the ankle's position. In a nutshell, this joint-synergy-based exoskeleton completely controls the hind limb of the rat by only adjusting the position of its hip and knee.

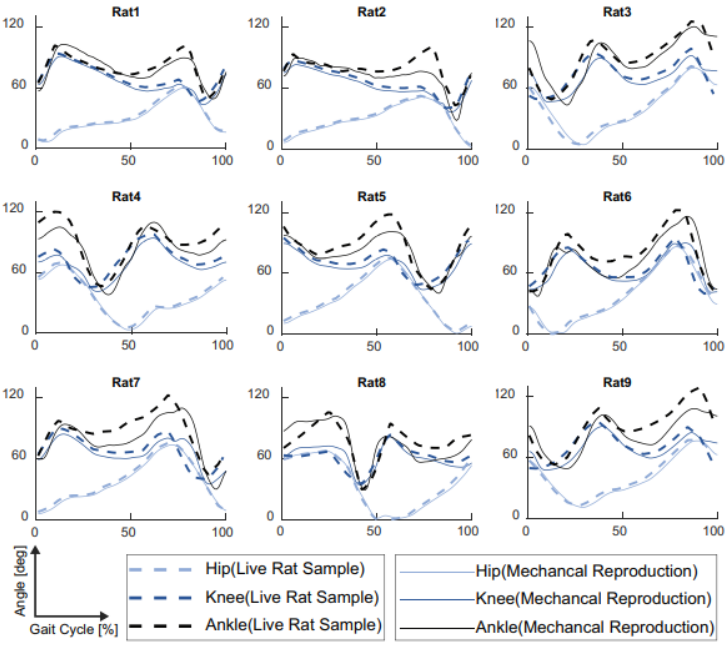

Fig. 2–7: Joint angle trajectories (Mechanical Reproduction vs. Live Rat Sample) [16].

The team then recorded the joint angle trajectories produced by the exoskeleton using a motion capture system and compared them to those of live rats. As shown in Fig. 2.7, the angle patterns of the hip and knee generated by the exoskeleton were very similar to those of the live rats. The ankle angle trajectory does, however, slightly deviate from that of the live rats. Overall, the angle patterns of the joints generated by the exoskeleton fit well with those of the living rat, demonstrating the validity of incorporating the theory of planer covariance into the exoskeleton design, which is a great inspiration for the design process of our linkage mechanism.

2.2. Summary

Based on the previous designs, we noticed the optimal design for the rehabilitation-assisted mechanism for rodents would be the one that assists the paralyzed rodent to move in a quadrupedal manner. The overground mechanism equipped on the back of the mouse is what the team is looking for because it could provide additional support to the rodent's back and help it to mimic the quadrupedal gait patterns.

However, a limitation of the current mechanical designs is their lack of adjustability. The mechanisms described in Sec. 2.1.1 to 2.1.3 were designed specifically for certain mice/rats, making it challenging to implement these mechanisms in other rodents due to various size differences. As a result, the primary focus of this project is on exploring the mechanical design for adjustability for the rehabilitation-assisted mechanism for paralyzed rodents.

3. Design of the Mechanism

3.1. Statement of the Design Loop

The mechanical design process, also known as the "design loop," is a systematic method for designing mechanisms that meet the needs and requirements of customers.

The classical design loop, clarified by [23], is summarized and divided into 10 steps:

1. Identification of Need
2. Background Research
3. Goal Statement
4. Performance Specifications
5. Ideation and Invention
6. Analysis
7. Selection
8. Detailed Design
9. Prototyping and Testing
10. Production

However, this classical design loop is suitable for the conventional design process where multiple design solutions needed to be examined and compared. Due to the project's time constraints, we decided to only concentrate on one design solution and continuously improve it through evaluations of each iteration. Hence, we adopted the following specific design loop throughout the course of the project:

1. Define the goal and requirements that the design needs to fulfill based on neuroscientists' needs.
2. Develop detailed design specifications for the design solution to meet the identified requirements.
3. Conduct the 1st design and prototyping iteration in accordance with the specifications and evaluate the pros and cons of the iteration.
4. Continuously iterate and improve the design based on the analysis of each iteration to make the design satisfies more specification.
5. Repeat this process until the design fully satisfies the pre-defined specifications.
6. Manufacture the final, optimal design and deliver it to our collaborator, Prof. Wu.

The next section will include the goal, requirements, and specifications from both the ME side and the CS side.

3.2. Goal, Requirements, and Design Specifications

3.2.1. Goal, Requirements, and Design Specifications from the ME Side

The goal of the ME side of the team is to design a mechanism that offers locomotion training for paralyzed rodents suffering from spinal cord injuries, while it can also be adjusted to suit rodents of different sizes. This mechanism is intended to aid neuroscientists in their study of the process by which rodents recover from SCI.

Specifically, the mechanism should satisfy the following requirements:

1. offer adequate support to the rodent during the locomotion training;
2. provide gait assistance during the locomotion training which enables the rodent to produce gait patterns that are similar to the healthy ones;
3. allow the rodent to move in a natural, quadrupedal manner, rather than forcing it to move bipedally.
4. provide adjustability for accommodating rodents of different sizes;
5. to help a completely paralyzed mouse regain its motor function, one can use a motor to control its movements and gradually reduce the motor's power while using sensors to detect the mouse's movement force, enabling it to move independently.

To fulfill the requirements outlined above, the machine must meet the following specifications:

1. The mechanism should consist of a support table (Size: $\leq 20*20*20\ cm^3$) that serves as the primary structure and provides adequate support to the rodent.
2. The mechanism should integrate two linkages that enable active gait training for both hind limbs of the rodent. Each linkage should satisfy the following specifications:
   (a) The linkage should have at least one Degree of Freedom (DoF), allowing the mouse to mimic one of its numerous healthy gait patterns.
   (b) The crank should be driven by a motor, providing the necessary power to drive the linkage.
   (c) Links should be connected by dowel pins or screws to ensure that the linkage can withstand the applied loads and forces during its movements.
   (d) The linkage should not include sliders to ensure its reliability and durability.
   (e) The endpoint of the coupler should be designed as a reserved hole that could be used to attach to the midpoint of the rodent's calf with a fishing wire.
3. The design of the mechanism should be modular to enable easy disassembly and reassembly, such that different modules could be installed to meet the needs of adjustability.
4. The device must include an encoder that can measure the motor's rotational speed and position and be able to provide real-time feedback on the motor's rotational speed and position.
5. A gear system must be designed to accurately transfer the motor's rotational motion to the encoder.

As described in Sec. 3.1, the mechanical design should gradually meet the above specifications through successive iterations. Since this is a cross-disciplinary project, the next section states the goal, requirements, and design specifications of the CS Side of the project team.

3.2.2. Goal, Requirements, and Design Specifications from the CS Side

The CS side of the team aims to develop a control and data collection system to accurately monitor and regulate the mechanism, with the specific objective of aiding the recovery of paralyzed mice. The CS team compiled a list of characteristics that the control and data collection system needed to meet to ensure that the mechanism is a comprehensive and functional device:

1. cover the automation part of the design;
2. efficiently collect necessary data from the mouse;
3. enable easy adjustments by the user.

To meet these characteristics, the CS team refined each step and listed a series of specifications that must be fulfilled, including:

1. Introduce a way to drive motors to achieve automation; ensuring that the motors used in the design are suitable for the specific requirements of the project, such as providing exact control over the movement and position of the linkage.
2. The accuracy of movement of a linkage system by a motor should be within a specified tolerance limit to ensure precise and consistent positioning.
3. Ensure that the power source for the device is enough for testing and data collection.
4. Record the speed and angle of the motor rotation output from the encoder, using a microcontroller and a programming system capable of effectively controlling motor movement.
5. Add a controlling system, such as a rotary encoder, to regulate the motor to produce slight movement in the legs of the mouse and gradually increase the training intensity as the mouse recovers.

By meeting these specifications, the CS team could ensure that the mechanism was comprehensive and functional, allowing it to effectively aid neuroscientists in studying the recovery process of rodents from SCI.

3.3. 1st Design Iteration

Our 1st design iteration focuses on the mechanical design aspect. Specifically, our goal is to define the general framework of the entire mechanism, including the structure of the support table (Spec. ME #1) and the four-bar linkage system (Spec. ME #2). Since the electrical design was not the focus of this design iteration, it would not be covered in this section.

3.3.1. Mechanical Design

To satisfy the Spec. ME #1, we designed a support table to satisfy the first design specification. The support table primarily consists of three components: the legs, the tabletop, and the linkage mounting slots. To provide sufficient support during the locomotion training, there exist four reserved holes on the tabletop that are used to secure the upper portion of the rodent's body by fixing it with Velcro, rope, etc.

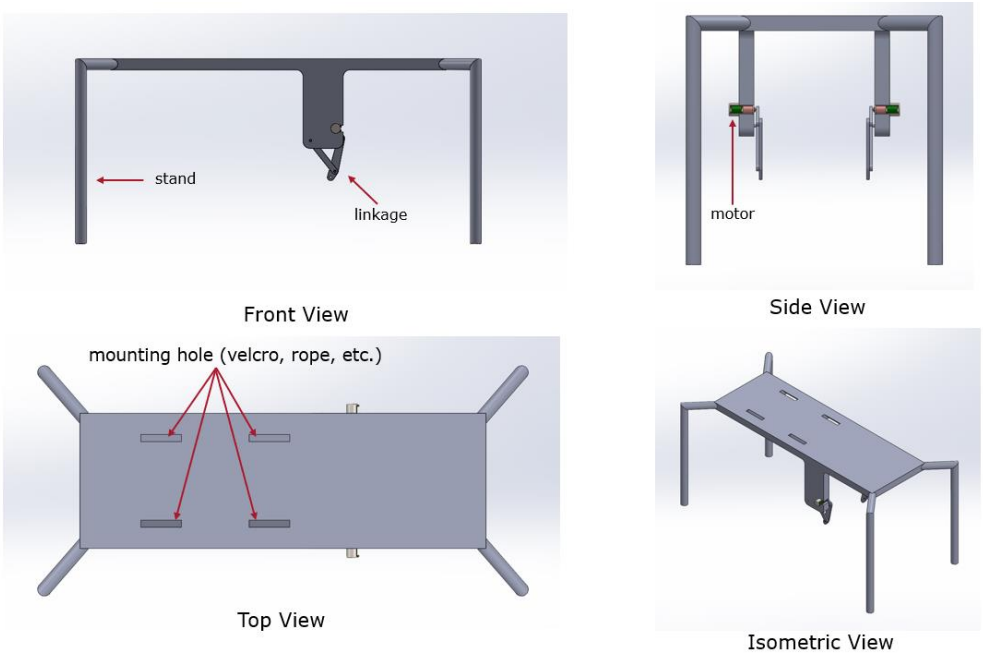

Fig. 3–1: 3D model of the support table (different view perspectives).

Following the second design specification, we started to design the linkage system. Due to the difficulty of integrating multiple linkage systems, we adopt only one four-bar linkage in our first design iteration.

The first step in designing the linkage system is to extract the gait patterns of the healthy mouse based on the pre-recorded video provided by Prof. Wu. Fig. 3-2 presents the basic logic of extracting the healthy mouse's gait patterns from the video:

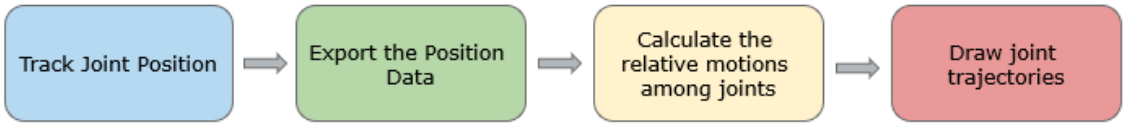

Fig. 3–2: Flowchart of the gait pattern extraction process.

To analyze the healthy mouse gait patterns, we needed to track the positions of the mouse's hind limb joints (hip, knee, and ankle) in each frame of a complete gait cycle. We handled this tack with the open-source software Tracker. Fig. 3-3 depicts the graphical user interface (GUI) of the software.

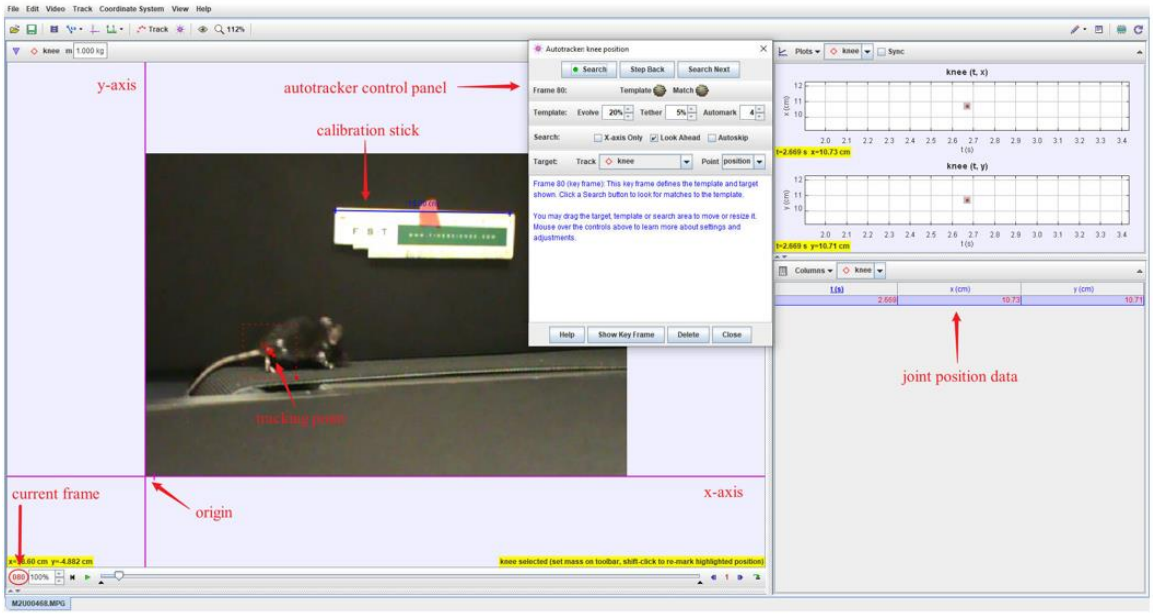

Fig. 3–3: Graphical user interface (GUI) of Tracker. The two crossed purple lines represent the Cartesian coordinate system that all the extracted data w.r.t. The blue line segment represents the calibration stick that converts the extracted data from a pixel-based unit to a real-world length dimension unit (cm, inch, etc.).

The first step in the tracking procedure is to specify the location of the Cartesian coordinate system's origin, which serves as the standard reference for all the extracted data. Since the resolution of the video is 720 * 480, we set the lower left corner of the video (0, -480) as the origin of the coordinates. Then, we established a calibration stick (15 cm) that translate the extracted data's unit from pixels to centimeters. After that, we identified the specific video frames that needed to be tracked (frames 258 to 267). To meet specification #2 (e), i.e., the endpoint of the coupler of the linkage needs to attach to the midpoint of the rodent's calf, we tracked both the positions of the hip and the midpoint of the calf. Finally, we saved and exported the obtained position data to a .txt file.

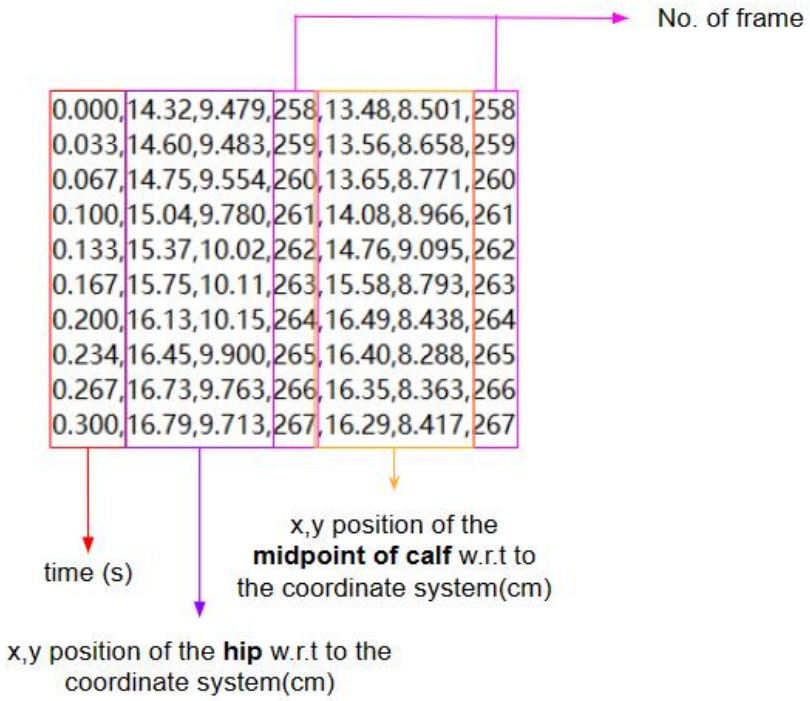

Fig. 3–4: Extracted position data of the mouse's hind limb joints.

Fig. 3-4 demonstrates the content of the exported .txt file. The data boxed in the purple and orange rectangle represents the x and y positions of the hip and the calf's midpoint w.r.t to the pre-defined coordinate system, which could be denoted as the vector $\bar{R}_{hip}$ and $\bar{R}_{calf,mid}$, respectively. Therefore, the relative position of the calf's midpoint w.r.t that of the hip, denoted as $\vec{R}_{hip\,|\,calf,mid}$, in each frame could be computed by the following equation:

$$\vec{R}_{hip\,|\,calf,mid} = \vec{R}_{calf,\,mid} - \vec{R}_{hip}. \tag{3.1}$$

Using MATLAB to do the calculations, the joint trajectory could be obtained by fixing the tail of each relative position vector to the origin of the Cartesian coordinate system and connecting their heads. Fig. 3-5 shows the relative motion of the calf's midpoint w.r.t the hip between frames 258 and 267.

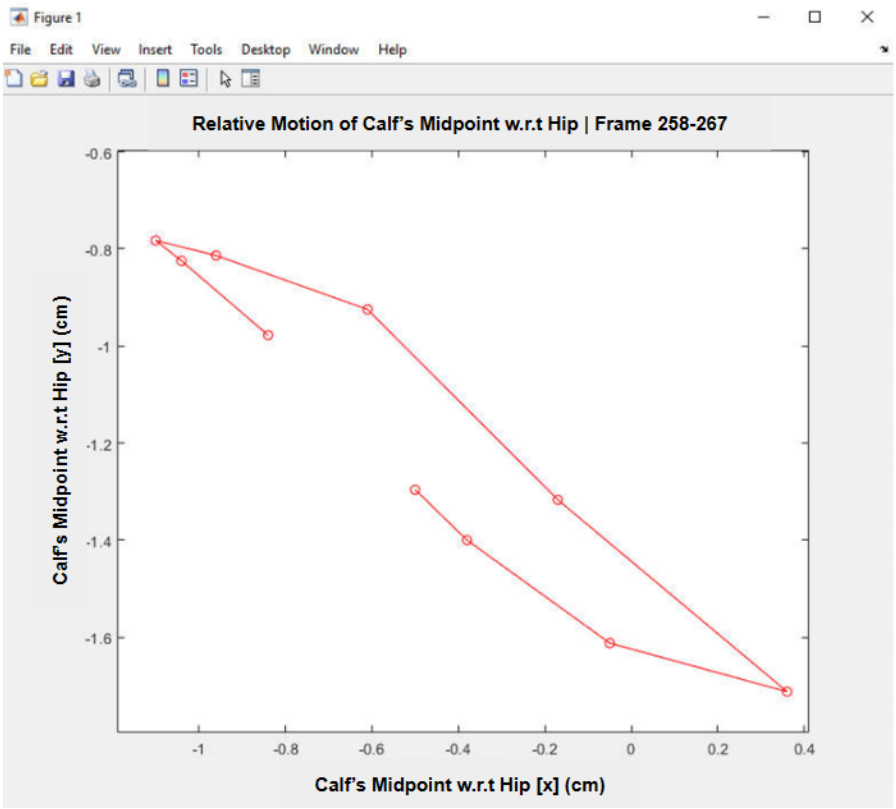

Fig. 3–5: Relative motion of the calf's midpoint w.r.t the hip | Frame 258-267.

Once the mouse's gait pattern has been extracted, we could determine the dimension of each link of our linkage system and create a 2D motion simulation based on the obtained joint trajectory. To determine the exact length of each link, we took the 1.0:3.0:3.5:2.5 linkage model (length ratio of crank, coupler, rocker, and ground) provided by [24, p. 316] as a reference. It was also worth mentioning that we did not adopt sliders in our design due to the consideration of the reliability of our system. Fig. 3-6 depicts the 2D linkage system that we designed utilizing the open-source software Linkage.

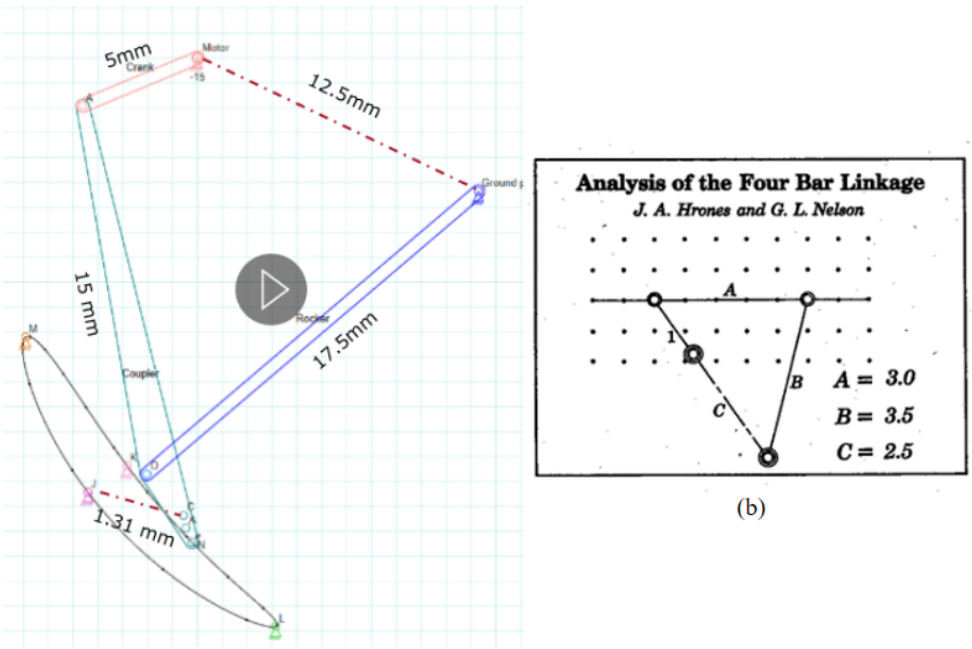

Fig. 3–6: 2D sketch and simulation of the four-bar linkage system. Length of each link: 5; 15; 17,5; 12.5 (Crank; Coupler; Rocker; Ground) [mm]. (b) 1.0: 3.0: 3.5: 2.5 linkage model [24, p. 316].

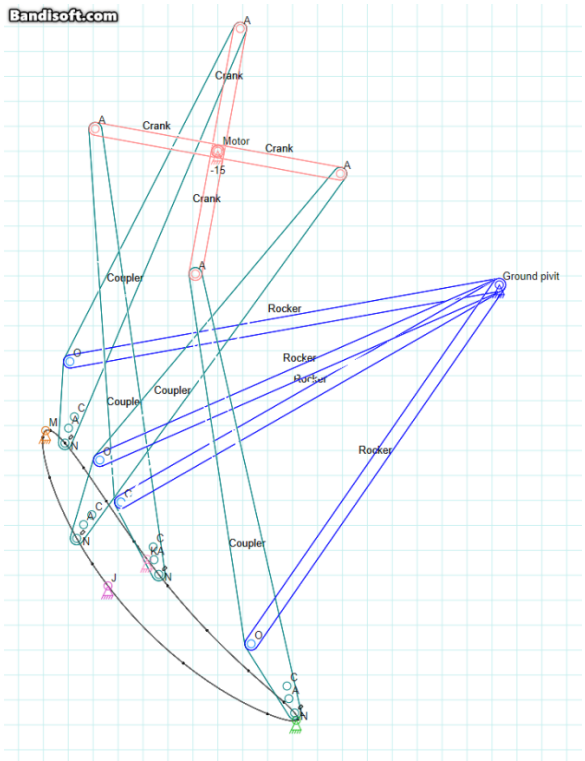

Fig. 3–7: Trajectory of the four-bar linkage system.

Based on the 2D linkage design, we then constructed the 3D model of each link in SolidWorks (Fig. 3-8). Table 3-1 summarizes the dimensions of each link's model. Since the dia. of the through hole that we reserved on the link is 1mm and the thickness of the link is defined as 1.5 mm, we chose to use the dowel pin with 1 mm dia. and 3 mm length. to fix and connect the links. So far, we have completed the design of the four-bar linkage system that provides gait assistance during the locomotion training for the paralyzed mouse.

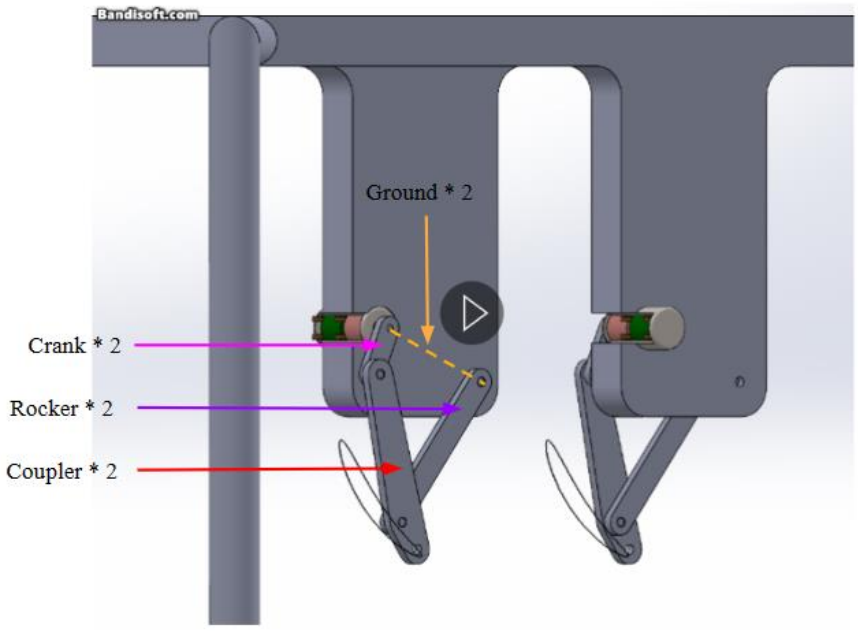

Fig. 3–8: 3D model of the linkage system.

Table 3-1: Dimension of each link's model.

	Length (mm)	Width (mm)	Thickness (mm)	Dia. of Reserved Hole (mm)
Crank	5	2	1.5	1
Coupler	15	2	1.5	1
Rocker	17.5	2	1.5	1

Based on Spec. ME # 3, we chose to use an ultra-tiny stepper motor to drive the linkage (Fig. 3-8), whose basic properties are summarized in Table 3-2.

Table 3-2: Spec. of the ultra-mini stepper motor.

Diameter (mm)	4.3
Dia. of Shaft (mm)	0.7
Length of Shaft (mm)	1.7
Overall Length (mm)	10.1
Working Voltage (V)	5
Working Current (A)	0.3
# of Phase	2
Horsepower (W)	14

Thus, we have completed the design of the first version of the rehabilitation-assisted mechanism for rodents. It performs passive gait rehabilitation training on mice through a linkage mechanism transmission. Table 3-3 summarizes all the components included in the mechanism.

Table 3-3: Component list of the mechanism.

# of Part	Name of Component
1	Support Table (Integrated with Four-bar Linkage Mounting Plate)
2	Acxico Ultra-Mini Stepper Motor
2	Crank Link
2	Coupler Link
2	Rocker Link
6	18-8 Stainless Steel Dowel Pins, 1 mm Diameter, 3 mm Long

3.3.2. Evaluation of the Mechanical Design

Due to cost considerations, we adopted 3D printing for the fabrication. We adopt two different 3D printing techniques during our manufacturing process. FDM (Fused Deposition Modelling) printing (material: PLA/ABS plastic) service offered by WPI's Innovation Studio was utilized for the fabrication of the support table since it is relatively large in comparison to the other components and does not necessitate high printing accuracy. Meanwhile, SLA (Stereolithography) printing (material: resin) offered by the WPI's rapid prototyping lab (RPL) was employed to manufacture link sets because they have through holes featuring 1 mm dia., which exceeds the lower bound of the FDM printing accuracy (2 mm) Also, we would like to ensure that the printed through-hole has a high-quality smooth finish, which reduces the friction and guarantees the smoothness of the linkage movement. Fig. 3-9 demonstrates different views of the 3D-printed support table.

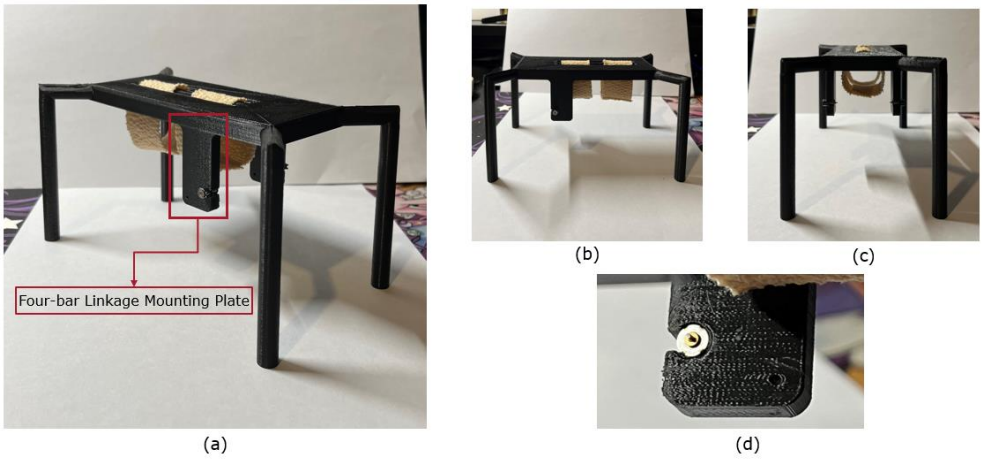

Fig. 3–9: 3D printed support table. (a) Isometric view; (b) Front view; (c) Side view; (d) Detailed view of the connection between the motor and the four-bar linkage mounting plate

Unfortunately, we did not receive the printed link sets due to the delay of the SLA printing service from WPI's RPL. As a result, we had to stop prototyping this design. Therefore, no testing data is yielded for this design iteration. However, we still discovered some issues with this design as we try to assemble other components:

1. The diameter of the through-hole we reserved on the link sets is too minute, which caused many limitations. For instance, we had to print these parts with the SLA technique and use 1mm dia. dowel pins for the assembly.
2. The connection between the motor and the four-bar linkage mounting plate relies on a merely press-fit, which is not stable and may cause the motor to fall out during the operation of the mechanism.
3. Due to the constraint of the motor's size, it cannot provide enough torque to drive the rat's hind limbs for gait training. In addition, the motor's pins were too close together, which increased the risk of connected solder joints and false soldering.

Overall, although the 1st iteration of mechanical design and prototyping was not perfect, it sets the stage for our future iterations. The following design iterations would use this design as a reference and fix the design flaws mentioned above. Meanwhile, we would also add new features to our following designs to fulfill any subsequent requirements.

The remainder of the page is intentionally left blank.

3.4. 2nd Design Iteration

3.4.1. Exploration of Adjustable Design

As specification #3 (Sec. 3.2.1) claims, the mechanism should be able to easily adjust to fit various sizes of rodents (Fig. 3-10). Since the adjustability of the mechanism is not a consideration of our 1st design iteration, we would like to explore the design idea that satisfies the adjustability in this design iteration.

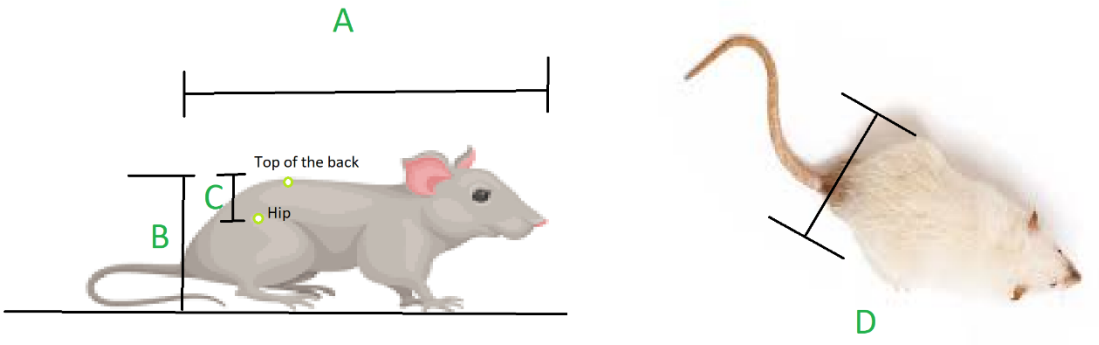

Fig. 3–10: Size data of the typical experimental mouse. (A: 6-10 cm; B: 3-5 cm; C: No exact data due to the measurement difficulty D: around 2 cm)

To achieve adjustability, we utilized a modular design approach by dividing the support table and four-bar linkage mounting plate into separate modules and incorporating sliders for position adjustments. This enables the easy modification of the mechanism to suit various sizes of mice. Fig. 3-11 presents the conceptual sketch of the revised design.

The remainder of the page is intentionally left blank.

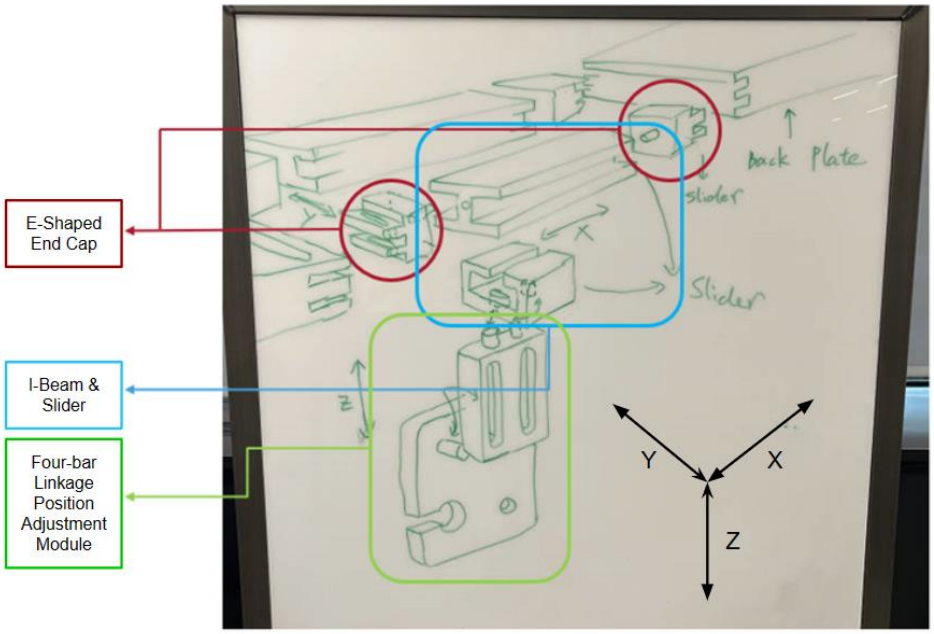

Fig. 3–11: Sketch of the adjustable mechanism.

Table 3-4 summarizes the conceptual design of the adjustable mechanism.

Table 3-4: Component list of the conceptual design of the adjustable mechanism.

# of Component	Name of Component
1	Front Plate of the Support Table (With E-shaped Slot)
1	Back Plate of the Support Table (With E-shaped Slot)
2	I-Beam
2	Slider of I-Beam
4	E-shaped End Cap (2 for each I-Beam)
2	Four-bar Linkage Position Adjustment Module (1 × Position Adjustment Slider, 1 × Four-bar Linkage Mounting Plate)
2	Four-bar Linkage (1 × Crank, 1 × Coupler, 1 × Rocker, 3 × Pin) [Not shown in Fig. 3-11]
4	Support Leg [Not shown in Fig. 3-11]

The I-beam and its slider, allow for position adjustment of the four-bar linkage along the x-axis. The E-shaped end cap is mounted at the end of the I-beam and engages with either the front plate or back plate of the support table to enable position adjustment along the y-axis. The four-bar linkage mounting module can split into two parts: the four-bar linkage

mounting slot and the slot attached to the I-beam slider, which allows for adjustment along the z-axis. In a nutshell, this design enables position adjustments of the four-bar linkage in all three dimensions.

One potential issue with this design is that it includes lots of components, which might lead to the difficulties in fabrication and implementation of the rodent. To simplify the design, we decided to focus on only x and y axes adjustments and eliminate the z-axis adjustment. This was possible because the wires connecting the rodent's hind limbs to the reserved hole of the coupler link could provide a certain amount of flexibility. The following section will present the actual mechanical design process for this conceptual adjustable mechanism.

The remainder of the page is intentionally left blank.

3.4.2. Mechanical Design

Based on the concepts of the adjustable mechanism outlined in Sec. 3.4.1, we constructed 3D models of each component in SolidWorks and assembled them into a complete structure. Fig. 3-12 presents the 3D model of the adjustable mechanism.

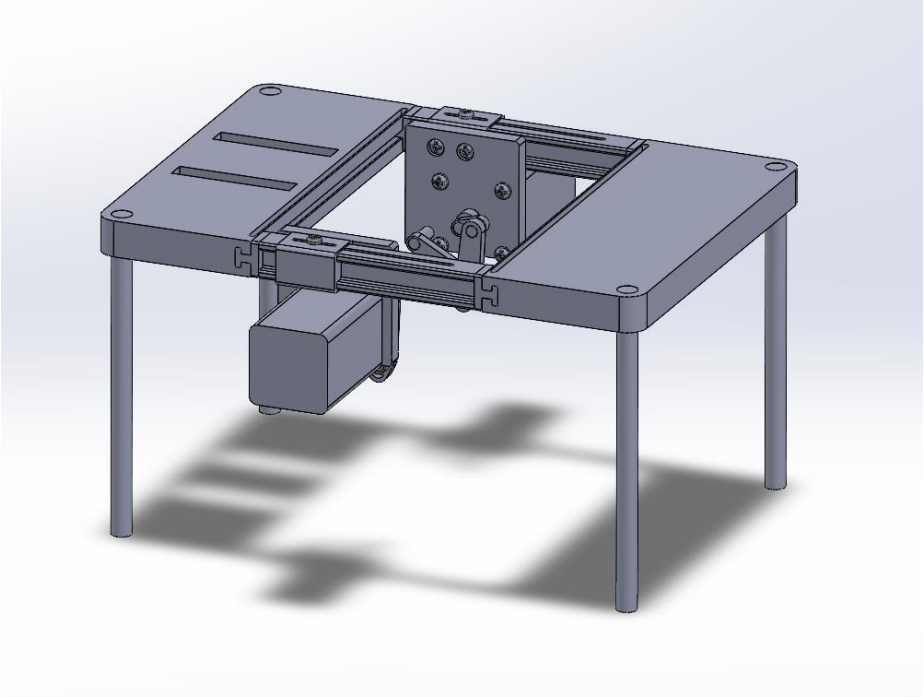

Fig. 3–12: Isometric view of the 3D model of the adjustable mechanism.

As shown in Fig 3.12, we replaced the ultra-tiny motors used in the first iteration with NEMA-8 stepper motors to address the third design flaw of the initial design mentioned in Section 3.3.2. Table 3-5 lists the specifications of the NEMA-8 motor. Meanwhile, four M2 screws, rather than a press fit, are used to secure the motors to the four-bar mounting plates (Fig. 3-13). In addition, the connections between the four-bar mounting plate and the I-beam slider (Fig. 3-14 (a)); the I-beam slider and the I-beam (Fig. 3.14 (b)); and the E-shaped end cap and the front/back plates are all secured with M2 screws with various lengths (Fig. 3-14 (c)). Compared to the first iteration of the design, which relies heavily on press fit for fixation, screw fixation is more reliable and minimizes the impact of tolerances issues brought up by 3D printing.

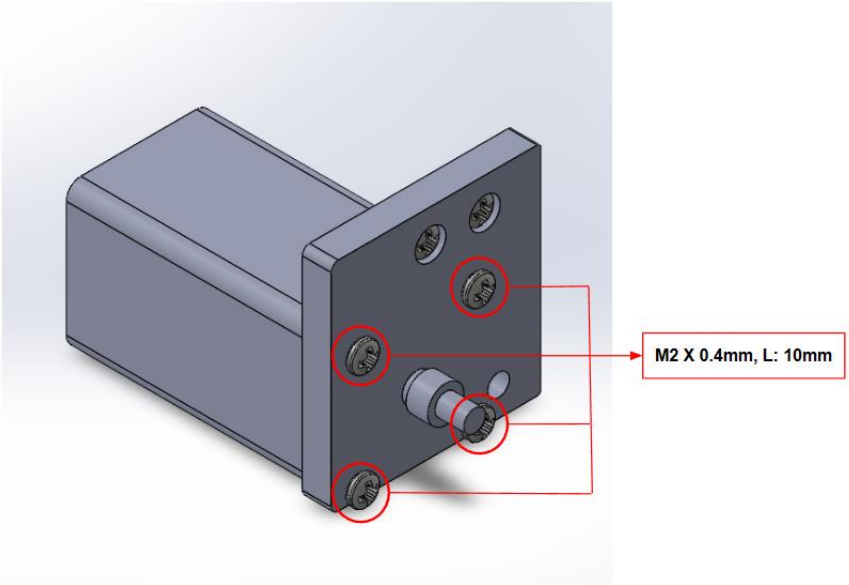

Fig. 3–13: Screw fixation for the NEMA-8 stepper motor.

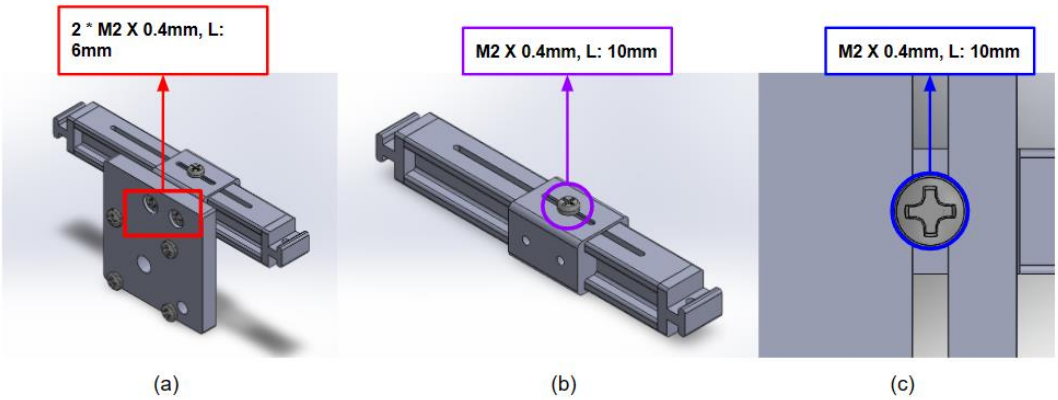

Fig. 3–14: Screw fixation design. (a) between the four-bar mounting plate and the I-beam slider; (b) between the I-beam slider and the I-beam; (c) between the E-shaped end cap and the front/black plate.

Table 3-5: Spec. of the NEMA-8 stepper motor.

L × W × H (mm)	34 × 20 × 20
Dia. of Shaft (mm)	4
Length of Shaft (mm)	9
Overall Length (mm)	10.1
Working Voltage (V)	3.9
Working Current (A)	0.6
# of Phase	2
# of Steps	200
Degree per Step (°)	1.8
Holding Torque (mN*m)	20

Regarding the design of the four-bar linkage, we widened each link of the four-bar linkage and adjusted the diameter of the through-hole from 1mm to mm to address the first design flaw mentioned in Sec. 3.3.2. Accordingly, the diameter of the pin was also increased to 3mm.

In a nutshell, throughout this design iteration, we focused on incorporating adjustability as a key feature. The position adjustment of the four-bar linkage was achieved by integrating sliders into the joints connecting each component. The adjustable range of the four-bar linkage along the x and y axes is shown in Fig. 3-15. Furthermore, the modular design approach employed in this design iteration ensures easy maintenance or updates to the mechanism in the future. The upcoming section will provide an evaluation of this design iteration.

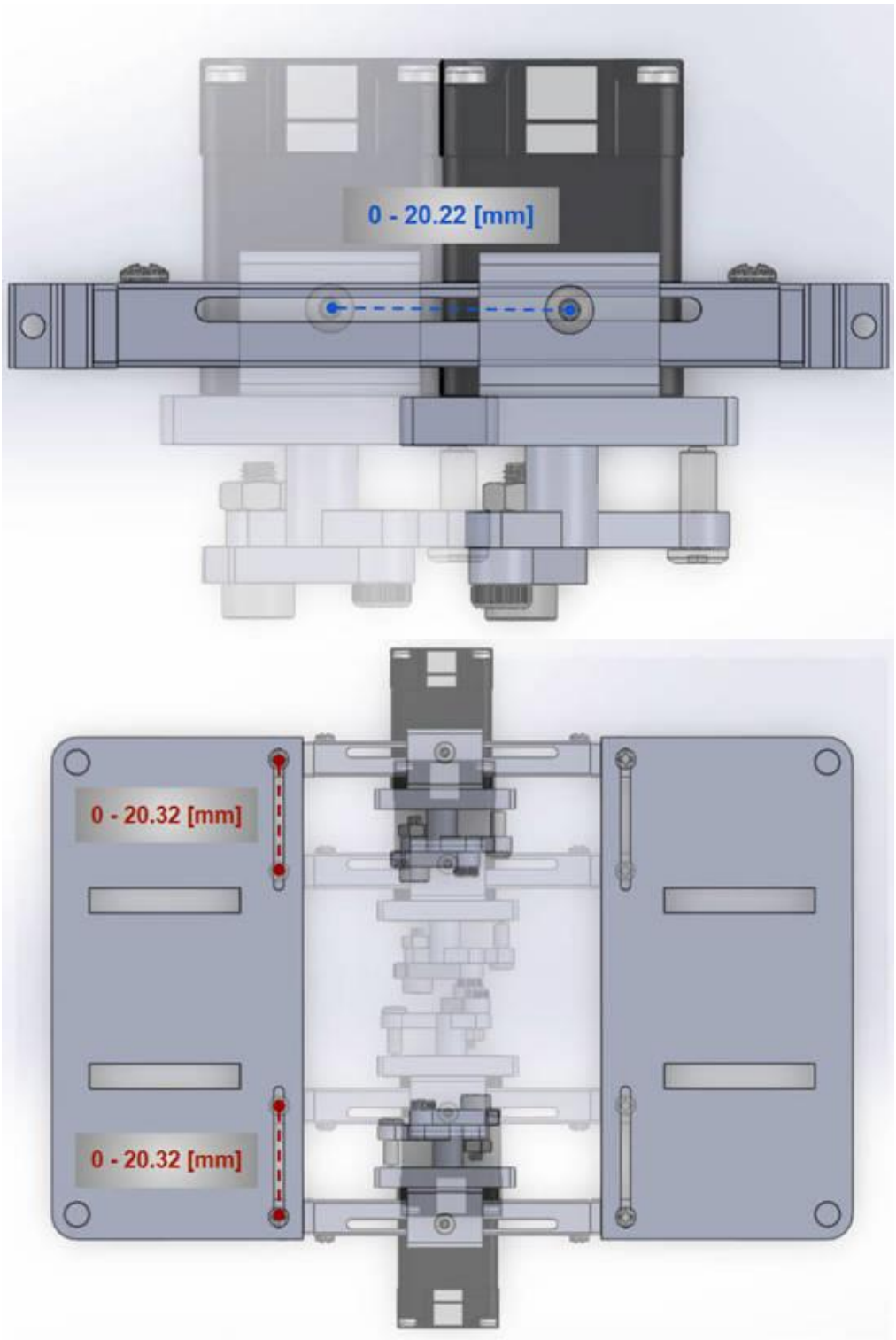

Fig. 3–15: The adjustable ranges of the four-bar linkage module along x (horizontal) & y (vertical) directions.

3.4.3. Evaluation of the Mechanical Design

Utilizing the 3D printing services provided by both WPI's Innovation Studio and Rapid Prototyping Lab, we printed all the components included in this design iteration and assembled them for evaluation. During the assembly and testing process, we identified the following design issues presented in this design iteration:

1. links might not align horizontally due to the connection between them relying on dowel pins that lack end-position restrictions (Fig. 3-16 (a));
2. links might be thrown off during the rotation due to the lack of position restrictions (Fig. 3-16 (b));
3. the reserved hole for the motor shaft of the four-bar mounting plate is too minute, which applies too much friction to the motor (Fig. 3-16 (c)).

Consequently, the next iteration of mechanical design mainly focuses on fixing the design flaws existing in the current iteration.

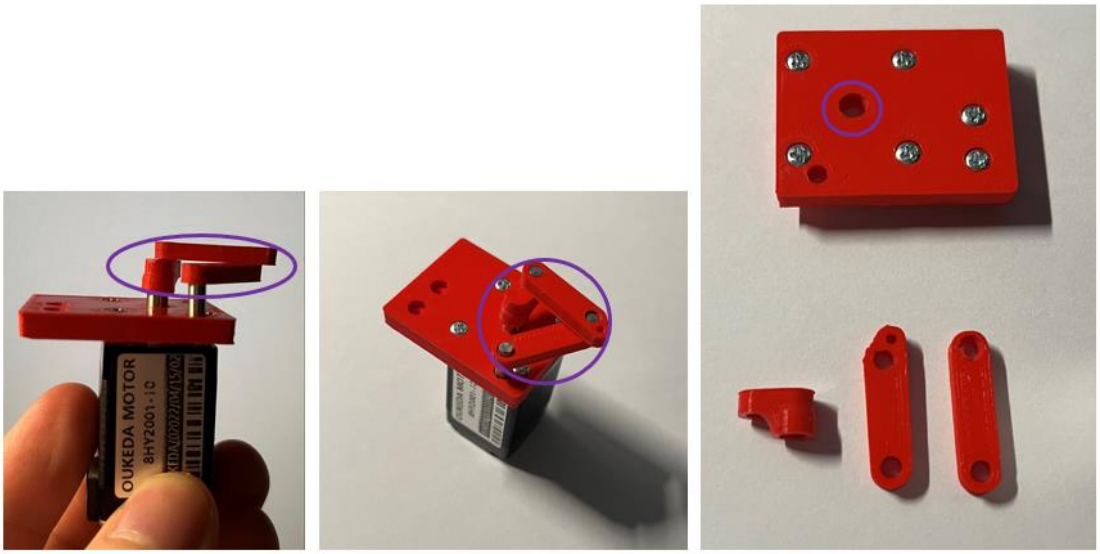

Fig. 3–16: Design flaws of the second iteration.

3.4.4. Electrical Design

A microcontroller is required to control the NEMA-8 bipolar stepper motor. After investigating various microcontrollers such as Raspberry Pi, PIC microcontroller, and Arduino, we decided to use an Arduino board as it was easily programmable and widely available. Furthermore, Arduino has a vast community and support, which makes it easy to find solutions for any problem that may arise during the development process.

We used the Arduino IDE to write C++-based codes, which could control the motion of the stepper motor in all three dimensions.

To fulfill the automation specifications (Spec. CS #1 & #3), we developed the code (Fig. 3-17) responsible for synchronizing the stepper motor's angle with the adjustable resistor's turn-knob angle. It begins with the inclusion of the Stepper library in the Arduino code. Then, the code defines two stepper motor objects, i.e., *slc* and *slcc*, which represent two stepper motors, one rotating in the clockwise direction *slc* and the other in the counterclockwise direction (*slcc*). The code specifies the number of steps per revolution for each stepper motor (200), as well as the four pins connected to the stepper motors (pins 3, 9, 10, and 11).

Next, the code defines several variables used in the control of the stepper motors. The variable *motorSpeed* specifies the speed at which the stepper motor rotates, and the variables *setAngle* and *curAngle* represent the desired angle and current angle of the adjustable resistor, respectively.

In the *setup()* function, the code sets up the serial communication with a baud rate of 9600 and a timeout of 25 ms. It also sets the *pinMode()* for the adjustable resistor's analog input (A0) and the built-in LED. The code sets the initial speed for the stepper motors and maps the initial angle of the adjustable resistor's turn-knob to the curAngle variable.

In the *loop()* function, the code reads the current angle of the adjustable resistor's turn-knob from the A0 pin using the *analogRead()* function and maps the value to the variable *setAngle*. The code then prints both *setAngle* and *curAngle* to the serial monitor. If the absolute difference between *setAngle* and *curAngle* is greater than one, the code turns on the built-in LED and rotates the appropriate stepper motor (*slc* or *slcc*) by one step to adjust the current angle. The variable *curAngle* is also adjusted to reflect the new angle of the adjustable resistor's turn-knob. If the difference between *setAngle* and *curAngle* is less than

or equal to one, the LED is turned off, and the stepper motors are not moved. This process is repeated in the *loop()* function until the Arduino is powered off or the code is stopped.

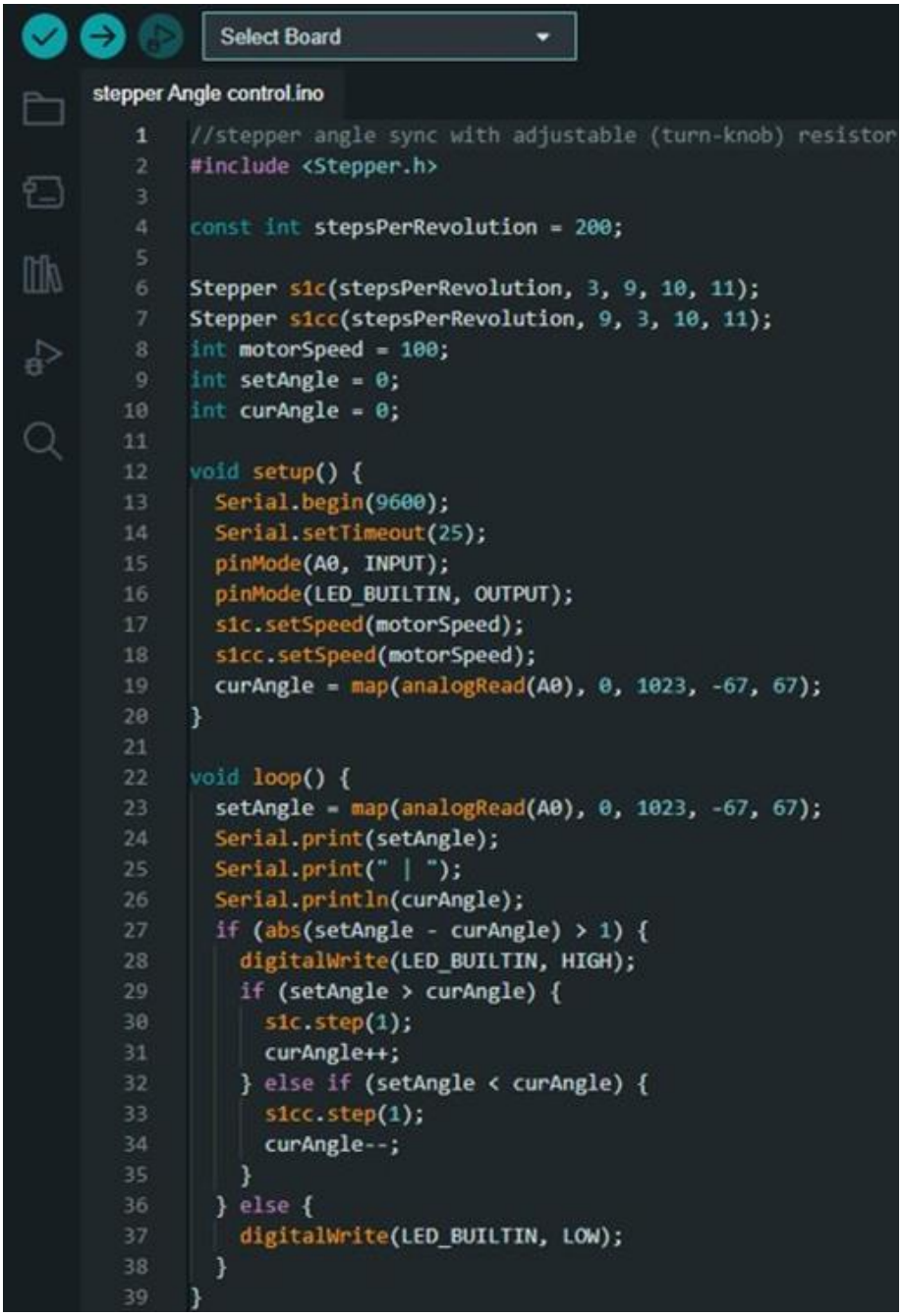

Fig. 3–17: Code for the function of angle replication.

To satisfy the accuracy specification (Spec. CS #2), we constructed the code (Fig. 3-18) which regulates the speed of a stepper motor. The motor is controlled through an Arduino board, which is connected to the computer via a serial or analog input.

As shown in Fig. 3-18, the code first defines the number of steps per revolution for the motor, which is set to 200 in this case. Two stepper motors are then defined, *stepper1c* and *stepper1cc*, with their respective pins specified in the parentheses.

The *motorSpeed* is set by taking in input from either the serial port or an adjustable resistor, as commented out in the code. The input is then parsed as an integer and assigned to the variable *motorSpeed*.

The *loop()* function then checks if the *motorSpeed* is greater than zero, in which case the *stepper1c* motor is set to the desired speed and rotated one step. If *motorSpeed* is less than zero, *stepper1cc* is set to the absolute value of motorSpeed and rotated one step. If motorSpeed is zero, the built-in LED is turned off.

This code can be used to control the motion of the stepper motor in real-time, allowing for precise movements and adjustments. In the exoskeleton design, this code could be used to control the exoskeleton's movement and record the mouse's motion.

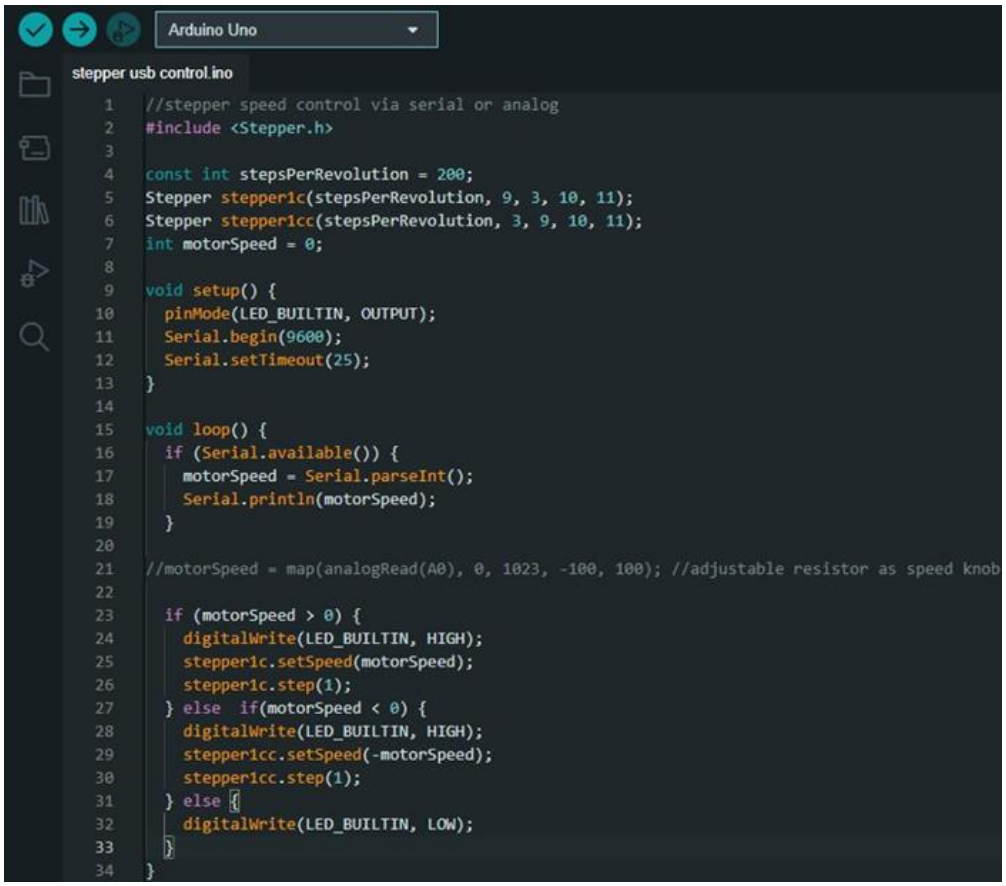

Fig. 3–18: Code for the speed control of the stepper motor.

We have tested and refined codes to ensure precise and efficient operation. This included testing the control system on a test bench and adjusting the code as needed. We also evaluated the performance of the control system in different scenarios and adjusted it as

needed to ensure optimal performance. We also tested the data collection system and made sure it can collect data efficiently. We made several adjustments and improvements to both the codes and the system until we were satisfied with the results.

Overall, the development of the control system was a critical aspect of the second design iteration. With the use of the Arduino board and our developed codes, we were able to achieve precise and efficient control of the stepper motor in all three dimensions. This allowed us to move forward with the project and focus on the next stages of development.t

3.4.5. Summary

In this design iteration, we considered both mechanical and electronic design. For the mechanical design, we focused on achieving adjustability through a modular design approach. For the electrical design, we utilized the Arduino board to control the stepper motor, which allows for precise speed control and synchronization of the stepper motor's angle with the turn-knob angle of the adjustable resistor.

The remainder of the page is intentionally left blank.

3.5. 3rd Design Iteration

As stated in Section 3.4.3, the third design iteration aims to resolve the three mechanical design issues identified in the second iteration. The following solutions were implemented:

1. To address design flaws #1 & #2, the dowel pins were replaced with shoulder screws (Fig. 3-19) This change provides two key benefits:
   (a) The non-threaded shaft under the head of the shoulder screw acts as a smooth-surfaced pin.
   (b) With the shoulder screw and the nut, the positions of links could be locked, preventing them from being thrown off by the rotational inertia while the motor is rotating.
2. To address design flaw #3, the diameter of the through-hole in the four-bar linkage mounting board was increased from 5mm to 16mm.

Overall, the third design iteration primarily aimed to address the mechanical design flaws of the second iteration without making significant changes to the mechanism's structure. After thorough testing, the solutions outlined above effectively addressed the design flaws. The following section would elaborate on the 4th design iteration that focuses on integrating the encoder into the mechanism, as Spec. ME #4 & #5 (Sec. 3.2.1) mentioned.

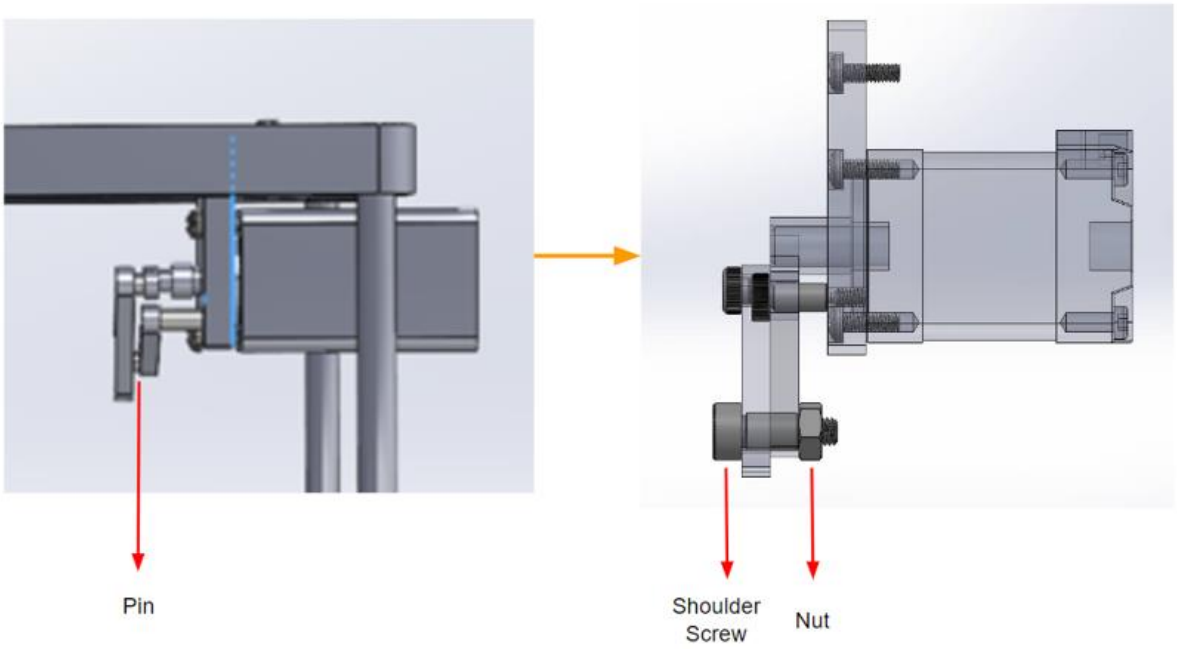

Fig. 3–19: Main contribution of the third design iteration.

3.6. 4th Design Iteration

3.6.1. Implementation of the Encoder

In order to meet Spec. ME #4 & #5, the fourth design iteration incorporates an encoder, which can capture the dynamic signals produced by the motor and convert them into digital codes for processing. Additionally, a gear system is also integrated to transfer motion from the motor to the encoder.

An encoder is a measurement device that converts physical motion into digital signals. In this application, the encoder can convert the dynamic signals of the motor into digital codes for recording and analysis. For example, the encoder can be used to measure the motor's speed, direction, angle, and other physical quantities, and convert them into digital codes for recording and analysis.

Table 3-6: Spec. of the E6A2-CW3C Rotary Encoder.

Supply voltage (V)	5 - 12
Current consumption (mA)	20 [max]
Number of responses (kHz)	30
Starting torque (mN·m)	1 [max]
Moment of inertia ($kg \cdot m^2$)	10^{-7} [max]
Shaft loading (N)	Radial: 10 Axial: 5
Slewing speed (r/min)	5,000 [max]

This design shown in Fig. 3-20 can help improve the accuracy and control capability of the motor system, as digital codes can more accurately represent the motor's status and motion. In addition, by analyzing the output signal of the encoder, a better understanding of the motor's operating conditions can be obtained, enabling the rodent to train by themselves.

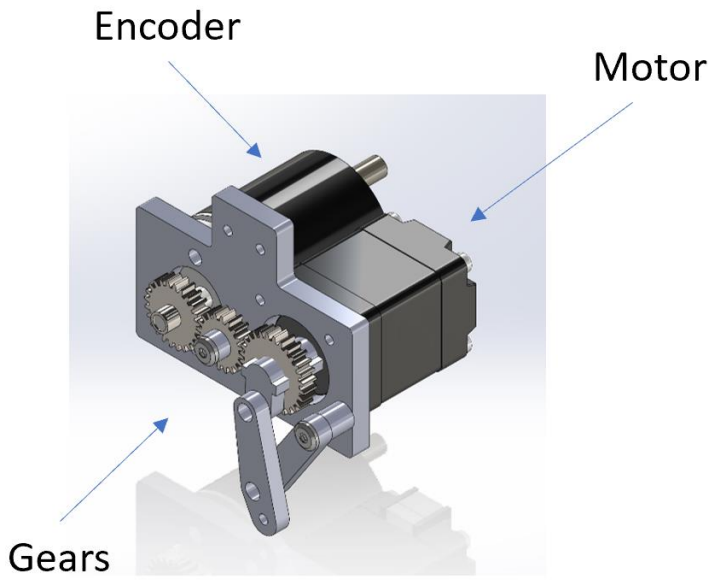

Fig. 3–20: The switching module design contains the encoder.

A gear structure can be used to connect the encoder and motor in a motor system. This allows the encoder to measure the rotation of the motor and provide accurate feedback to the control circuit.

By using this gear system depicted in Fig. 3-21, the encoder can accurately measure the motor's position, speed, and direction, allowing for precise control of the motor system.

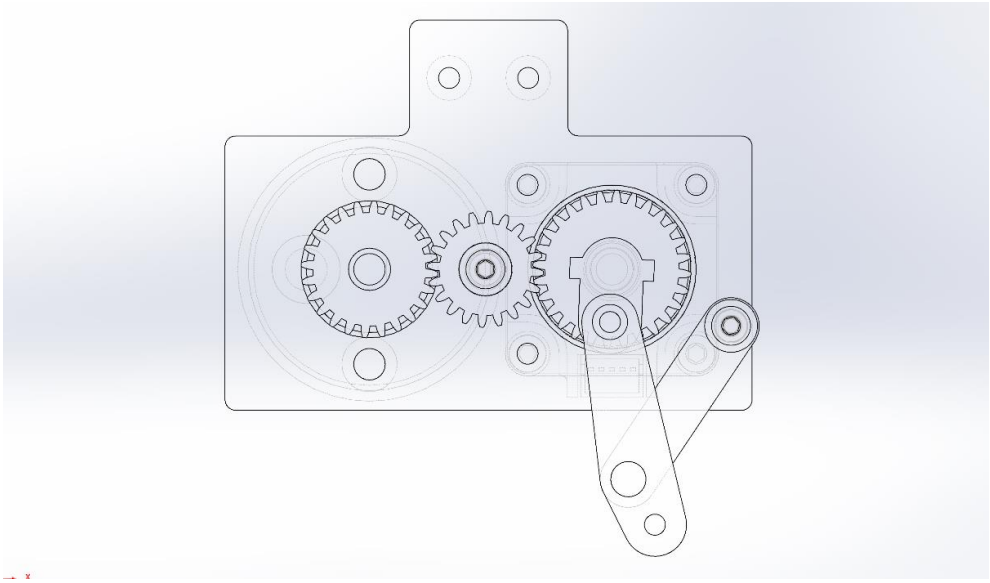

Fig. 3–21: The gear system connects the motor and the encoder. The three gears have teeth counts of 24, 20, and 28, respectively, and are arranged from left to right. The gear ratios from left to right are 20:24 = 5:6 and 28:20 = 7:5.

The gear ratio from the 24-tooth gear to the 28-tooth gear is 24:20 * 20:28 = 6:7. This is because the gear ratio can be calculated by multiplying the ratios of adjacent gears. Therefore, the ratio of the 24-tooth gear to the 20-tooth gear is 24:20 = 6:5, and the ratio of the 20-tooth gear to the 28-tooth gear is 20:28 = 5:7. Multiplying these two ratios gives the gear ratio from the 24-tooth gear to the 28-tooth gear as 6:7.

This 4th design iteration incorporates a gear system that meets the Spec. ME #5 which ensures precise transfer of the motor's rotational energy to the encoder.

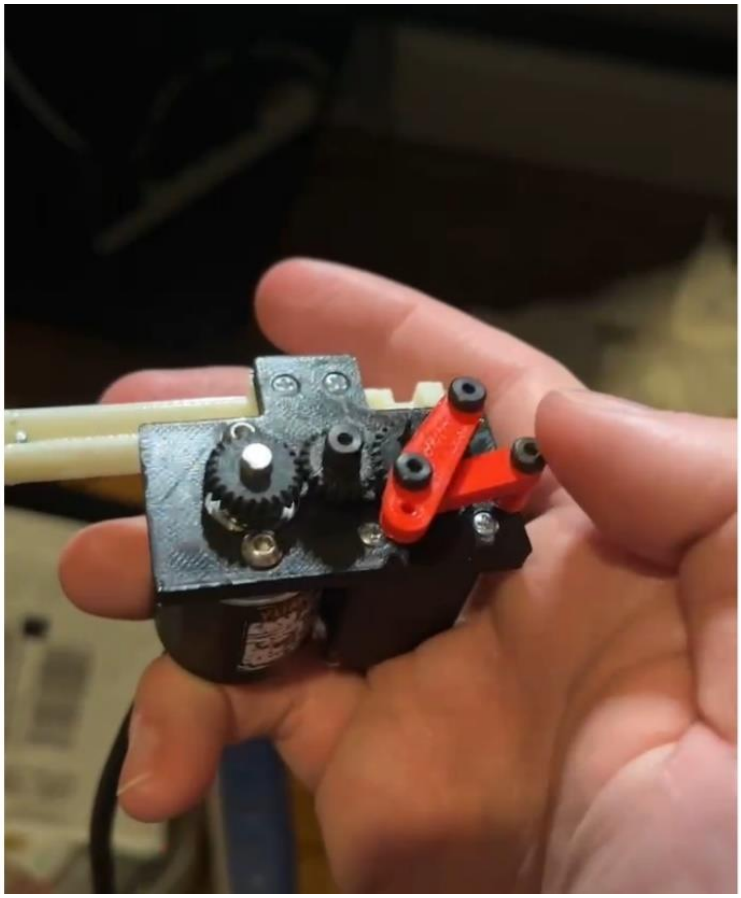

Fig. 3–22: 3D printed well-assembled motor module with encoder design.

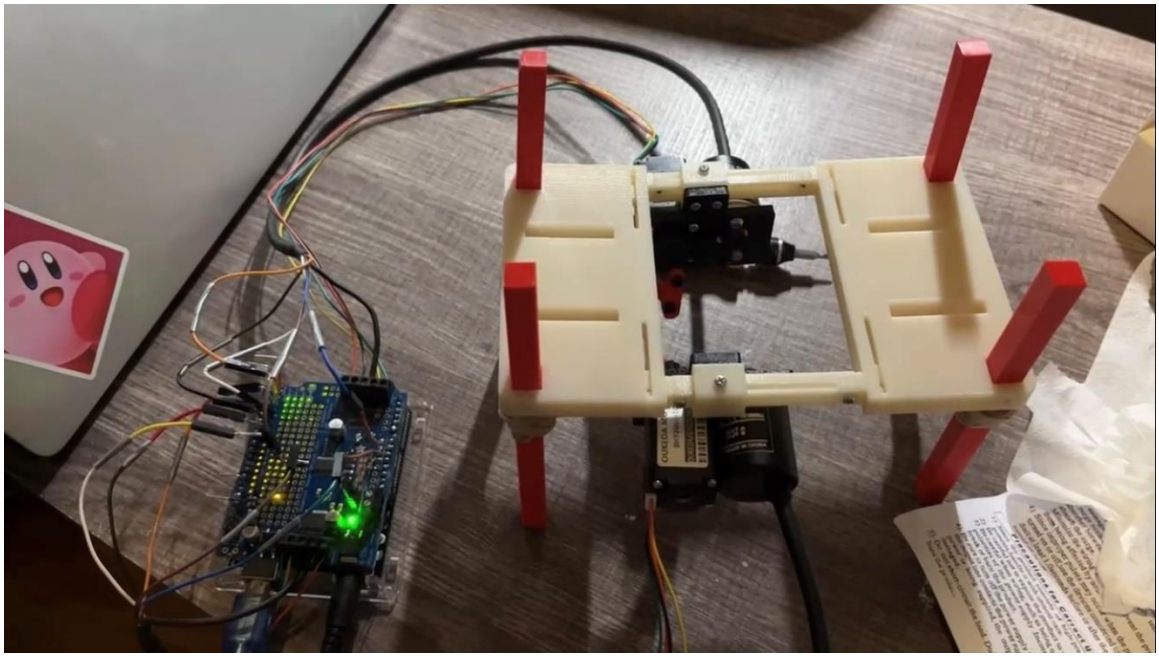

Fig. 3–23: The fully assembled mechanism with wiring.

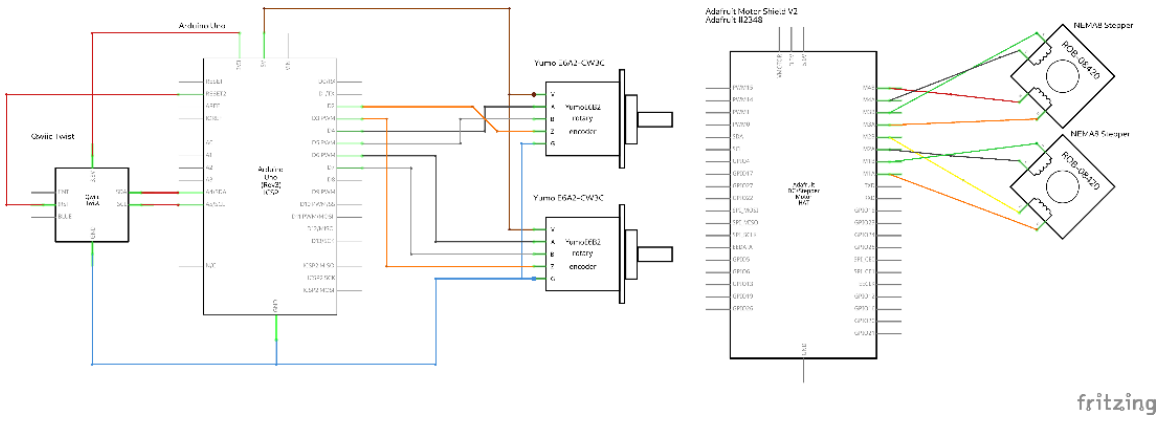

Fig. 3–24: Final wiring diagram between devices and breakout boards, using Fritzing. Since there are only two pins available on Arduino Uno Rev.3 for hardware interrupt, the two encoders also use interrupts, and the team didn't figure out a way to utilize it without the interrupt pin in time, the Qwiic Twist digital RGB Inter-Integrated Circuit (I2C) encoder was not included in the final design.

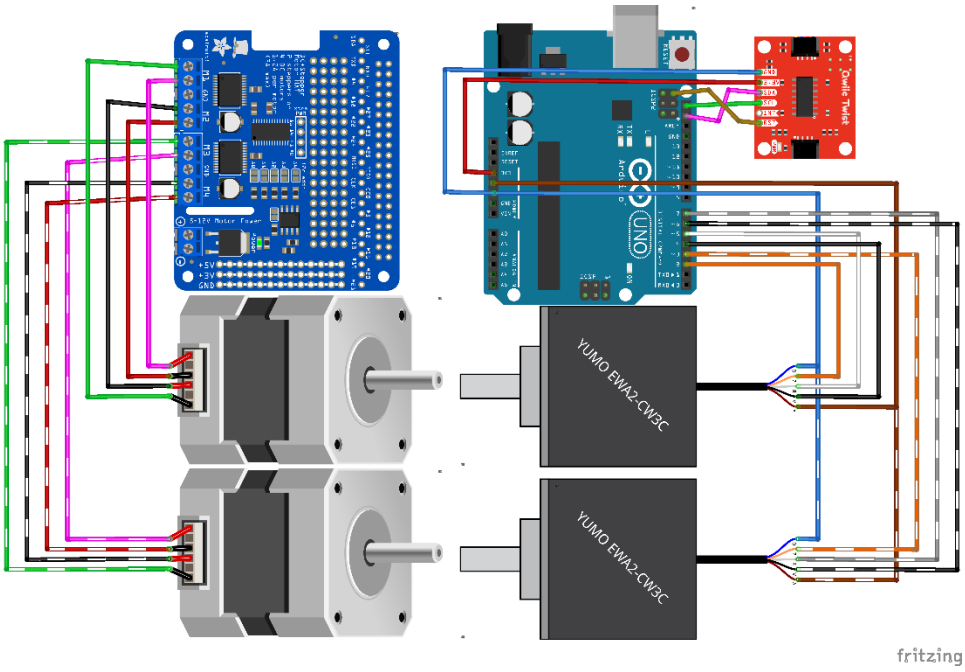

Fig. 3–25: 2D rendering of final wiring between devices and breakout boards, using Fritzing. The Qwiic Twist digital RGB I2C encoder was not included in the final design.

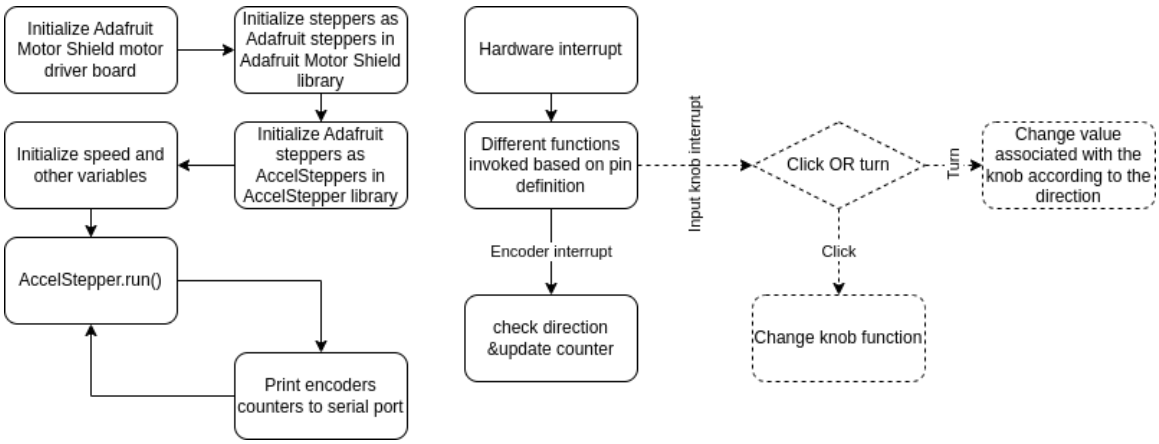

Fig. 3–26: Flowchart of final software design. The logic used to handle the input of the input knob (dotted line section) was not included in the final design.

3.6.2. Encoder Coding

To meet the need of data recording (Spec. CS #4 & 5), we developed the code (Appendix A) that is used to control the motion of a NEMA-8 bipolar stepper motor in all three dimensions. The code includes three libraries: <*digitalWriteFast.h*>, which is a library for high-performance reads and writes, <*Adafruit_MotorShield.h*>, which is used to control the motor shield object, and <*AccelStepper.h*>, which is used for acceleration and deceleration of the stepper motor.

The code defines quadrature encoders for both the left and right motors with the pins they are connected to and sets the encoders to increment or decrement depending on the direction of the motor. The code also defines the stepper motor pins and sets the maximum speed and acceleration of each motor.

The loop function controls the motion of the stepper motors by calling the *AccelStepper* run function for each motor. It also prints the encoder tick counts for both motors.

Finally, the code includes two interrupt service routines that handle the quadrature encoder inputs for both motors. When an encoder pulse is detected, the ISR adjusts the encoder tick count by one, based on the direction of the motor.

Overall, this code provides a means to precisely control the motion of the stepper motor and detect small movements using quadrature encoders.

The remainder of the page is intentionally left blank.

3.7. Summary

After several design iterations, we were able to manufacture the final version of the rehabilitative-assisted mechanism that met all the requirements and specifications outlined in Sec. 3.2. The ME side of the team successfully designed and fabricated the structure, linkages, and all other mechanical components, while the CS team developed the control and data collection system. The final product was delivered to our collaborator, Prof. Wu, for testing in his laboratory.

However, due to the time constraint of the project and unforeseen circumstances, the actual animal experiments could not be conducted as planned, which is the major limitation of the project since one of the primary goals of this project was to aid neuroscientists in their study of the recovery process of rodents from SCI. Without testing the mechanism on live rodents, we could not confirm the mechanism's effectiveness in providing proper active rehabilitative training.

Although our project did not yield experimental data, its main contribution lies in the exploration of the potential for adjustable design in rehabilitative-assisted mechanisms through the implementation of a modular design concept. We hoped that this could inspire the future development of the rehabilitative-assisted mechanism which is either for research or medical use.

The remainder of the page is intentionally left blank.

4. Review of Failure

The process of designing a mechanism for paralyzed rodents to aid neuroscientists in their study of spinal cord injuries was not without its challenges. Throughout the project, the team encountered various obstacles that hindered progress and ultimately resulted in some failure to achieve desired goals. In this chapter, we will review these failures and reflect on the lessons learned from them. We will discuss the challenges faced by both the CS and ME sections during the prototyping and testing phase, the difficulties in integrating the two sections, and the teamwork issues encountered. Through this review, we hope to identify areas for improvement and provide insights for future projects.

4.1. Potential of Machine Learning

4.1.1. A-Term Research Overview

In the early stages of the project, the CS team focused on using image recognition to identify the movement of mice, particularly the trajectory of their legs. The aim was to collect data about the subject's gait in a more efficient and accurate manner to help the ME team create more accurate models and reduce manual labor in future experiments.

4.1.2. Obstacles Encountered

The CS team encountered significant obstacles during the B term in the area of machine learning. They were unable to collect effective data for analysis due to the blurry and insufficient videos provided and a lack of valid data sources on the web. Additionally, there were significant teamwork issues between the CS and ME teams, leading to misunderstandings and a lack of collaboration. The CS team took a leadership role in addressing these issues by organizing more effective group meetings and explaining their design ideas to everyone.

4.1.3. Continuation of Research Direction

The CS and ME teams shifted their focus towards the motor sensor and Arduino programming for the control system in the C term. They selected the appropriate motor sensor and designed the control system for the motor, integrating it into the existing exoskeleton design. The teams also explored ways to improve the data collection system by incorporating pressure sensors to measure the force applied by the mice during locomotion, providing more accurate data for the CS team's algorithm to predict the speed and direction of mice movements in real-time. Through their continued efforts, the teams made considerable progress toward achieving the project goals and learned valuable lessons from the obstacles encountered.

4.2. Delays in Manufacturing and Testing

One of the major challenges the team encountered during the C term was delays in manufacturing and testing. The team was tasked with designing and building an exoskeleton that could be worn by paralyzed rats to help them regain mobility. However,

the manufacturing process, which included 3D printing and testing, took longer than expected, causing delays in the testing phase.

Moreover, the team had difficulty obtaining the necessary materials for the project. The process of purchasing project-related materials, such as motors and sensors, took a considerable amount of time, further contributing to delays.

In addition, the team was unable to meet Prof. Wu's testing schedule for the rats, as the manufacturing and testing process took longer than anticipated. As a result, the team was unable to test the exoskeleton on real rats and obtain results.

These delays were significant setbacks for the project and highlighted the importance of accurate project planning and scheduling. Moving forward, the team plans to implement strategies to improve the efficiency of the manufacturing process and streamline the procurement of necessary materials to avoid similar delays in the future. The team will also work closely with Prof. Wu to ensure that testing schedules are realistic and achievable.

4.3. Team Science

During the course of this project, the team learned some valuable lessons, including the importance of effective communication between team members, as well as the importance of clear declarations of each member's contributions. To foster a more collaborative team environment, the MQP team should schedule frequent meetings; establish clear expectations for each team member; and encourage open communications either within the team or between the team and the advisor.

The remainder of the page is intentionally left blank.

5. Social implications

As discussed in the Introduction, there are still numerous individuals suffering from SCI, not only because of the loss of mobility but also because of the emotional and psychological burdens that accompany their condition. Despite the availability of various treatments for SCI, including surgery and electrical stimulation, locomotion training is still the most widely used method. However, traditional locomotion training requires the therapy to frequently adjust the postures of the patient's posture, which is cumbersome and inefficient.

As students majoring in the fields of science and engineering, although we lack any medical or neuroscience-related knowledge or professional background, we are still able to contribute possible solutions from our own areas of expertise. With this in mind, we proposed a design concept for a rehabilitative-assisted mechanism. This mechanism is intended to provide repeated active locomotion training for paralyzed rodents suffering from SCI. Our aspiration is that our work can inspire neuroscientists to view SCI research from a new perspective, ultimately resulting in the development of more effective methods for treating SCI patients.

6. Conclusion

Overall, the MQP team presented the design principle of the adjustable rehabilitative-assisted mechanism for rodents suffering from SCI, which aims to active locomotion training for paralyzed rodents, in order to help them recover and regain mobility.

However, due to the time constraint of the project and unforeseen circumstances, we did not conduct any animal experiments to validate the performance of the mechanism, which is the major limitation of this project. Nevertheless, the design principle of the adjustable rehabilitative-assisted mechanism still serves as a promising starting point for future studies. The sequel project could focus on validating and enhancing the current design or exploring its potential applications for human patients.

References

[1] W. H. Organization and I. S. C. Society, *International perspectives on spinal cord injury*. World Health Organization, 2013.

[2] C. S. Ahuja *et al.*, "Traumatic spinal cord injury," *Nat. Rev. Dis. Primer*, vol. 3, no. 1, p. 17018, Dec. 2017, doi: 10.1038/nrdp.2017.18.

[3] J. W. McDonald and C. Sadowsky, "Spinal-cord injury," *The Lancet*, vol. 359, no. 9304, pp. 417–425, 2002, doi: https://doi.org/10.1016/S0140-6736(02)07603-1.

[4] J. Lynch and R. Cahalan, "The impact of spinal cord injury on the quality of life of primary family caregivers: a literature review," *Spinal Cord*, vol. 55, no. 11, pp. 964–978, Nov. 2017, doi: 10.1038/sc.2017.56.

[5] Y. Chen, "National Spinal Cord Injury Statistical Center," Aug. 2022, doi: 10.17605/OSF.IO/NP24C.

[6] C. H. Merritt, M. A. Taylor, C. J. Yelton, and S. K. Ray, "Economic impact of traumatic spinal cord injuries in the United States," *Neuroimmunol. Neuroinflammation*, vol. 2019, Jul. 2019, doi: 10.20517/2347-8659.2019.15.

[7] C. Migliorini, B. Tonge, and G. Taleporos, "Spinal Cord Injury and Mental Health," *Aust. N. Z. J. Psychiatry*, vol. 42, no. 4, pp. 309–314, Apr. 2008, doi: 10.1080/00048670801886080.

[8] S. Dalbayrak, "Current and future surgery strategies for spinal cord injuries," *World J. Orthop.*, vol. 6, no. 1, p. 34, 2015, doi: 10.5312/wjo.v6.i1.34.

[9] J. Mehrholz, J. Kugler, and M. Pohl, "Locomotor training for walking after spinal cord injury," *Cochrane Database Syst. Rev.*, Nov. 2012, doi: 10.1002/14651858.CD006676.pub3.

[10] S. Hamid and R. Hayek, "Role of electrical stimulation for rehabilitation and regeneration after spinal cord injury: an overview," *Eur. Spine J.*, vol. 17, no. 9, pp. 1256–1269, Sep. 2008, doi: 10.1007/s00586-008-0729-3.

[11] A. Kaku, A. Parnandi, A. Venkatesan, N. Pandit, H. Schambra, and C. Fernandez-Granda, "Towards data-driven stroke rehabilitation via wearable sensors and deep learning," p. 28, 2020.

[12] D. Shi, W. Zhang, W. Zhang, and X. Ding, "A Review on Lower Limb Rehabilitation Exoskeleton Robots," *Chin. J. Mech. Eng.*, vol. 32, no. 1, p. 74, Dec. 2019, doi: 10.1186/s10033-019-0389-8.

[13] E. C. Bryda, "The Mighty Mouse: the impact of rodents on advances in biomedical research," *Mo. Med.*, vol. 110, no. 3, pp. 207–211, 2013.

[14] J. A. Nessler *et al.*, "A robotic device for studying rodent locomotion after spinal cord injury," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 4, pp. 497–506, Dec. 2005, doi: 10.1109/TNSRE.2005.858432.

[15] Y. S. Song and N. Hogan, “A Novel Interactive Exoskeletal Robot for Overground Locomotion Studies in Rats,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 4, pp. 591–599, Jul. 2015, doi: 10.1109/TNSRE.2015.2396852.

[16] T. Miyamoto, L. Ccorimanya, M. Hassan, S. Puentes, and K. Suzuki, “Joint Synergy-Based Rehabilitative Exoskeleton for Rodents,” in *2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Sapporo, Japan, Jul. 2022, pp. 1225–1230. doi: 10.1109/AIM52237.2022.9863343.

[17] M. Sharif-Alhoseini *et al.*, “Animal models of spinal cord injury: a systematic review,” *Spinal Cord*, vol. 55, no. 8, pp. 714–721, Aug. 2017, doi: 10.1038/sc.2016.187.

[18] S. Hesse, C. Werner, A. Bardeleben, and H. Barbeau, “Body weight-supported treadmill training after stroke,” *Curr. Atheroscler. Rep.*, vol. 3, no. 4, pp. 287–294, Jul. 2001, doi: 10.1007/s11883-001-0021-z.

[19] G. Catavitello, Y. P. Ivanenko, and F. Lacquaniti, “Planar Covariation of Hindlimb and Forelimb Elevation Angles during Terrestrial and Aquatic Locomotion of Dogs,” *PLOS ONE*, vol. 10, no. 7, p. e0133936, Jul. 2015, doi: 10.1371/journal.pone.0133936.

[20] N. Mort, “Comparative study of fuzzy DC servo motors and stepper motors for mechatronic systems,” in *IEE Colloquium on `Innovations in Manufacturing Control Through Mechatronics*, Newport, UK, 1995, vol. 1995, pp. 6–6. doi: 10.1049/ic:19951359.

[21] T.-H. Hsu, Y.-C. Chiang, W.-T. Chan, and S.-J. Chen, “A Finger Exoskeleton Robot for Finger Movement Rehabilitation,” *Inventions*, vol. 2, no. 3, p. 12, Jul. 2017, doi: 10.3390/inventions2030012.

[22] W. L. Johnson, D. L. Jindrich, R. R. Roy, and V. R. Edgerton, “Quantitative metrics of spinal cord injury recovery in the rat using motion capture, electromyography and ground reaction force measurement,” *J. Neurosci. Methods*, vol. 206, no. 1, pp. 65–72, Apr. 2012, doi: 10.1016/j.jneumeth.2012.02.008.

[23] R. L. Norton, *Design of machinery: an introduction to the synthesis and analysis of mechanisms and machines*, Sixth edition. New York, NY: McGraw-Hill Education, 2020.

[24] Hrones, John A. and Nelson, George L., *Analysis of the Four-Bar Linkage Its Application to the Synthesis of Mechanisms*. Mit Pr, 1951.

Appendix A: Code for Encoder Control

encoderSource.ino

```
#include <digitalWriteFast.h>  // library for high performance reads and writes by jrraines
#include <AdafruitMotorShield.h>
#include <AccelStepper.h>
// see http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1267553811/0
// and http://code.google.com/p/digitalwritefast/

// It turns out that the regular digitalRead() calls are too slow and bring the arduino down when
// I use them in the interrupt routines while the motor runs at full speed creating more than
// 40000 encoder ticks per second per motor.

// Quadrature encoders
// Left encoder
#define c_LeftEncoderInterrupt 0
#define c_LeftEncoderPinA 19
#define c_LeftEncoderPinB 25
#define LeftEncoderIsReversed
volatile bool _LeftEncoderBSet;
volatile long _LeftEncoderTicks = 0;

// Right encoder
#define c_RightEncoderInterrupt 2
#define c_RightEncoderPinA 4
#define c_RightEncoderPinB 5
volatile bool _RightEncoderBSet;
volatile long _RightEncoderTicks = 0;

// Create the motor shield object with the default I2C address 0x60
Adafruit_MotorShield driverBoard = Adafruit_MotorShield();

// Connect a stepper motor with 200 steps per revolution (1.8 degree)
// to motor port #1 (M1 and M2)
Adafruit_StepperMotor *stepper1 = driverBoard.getStepper(200, 1);
// Connect a stepper motor with 200 steps per revolution (1.8 degree)
// to motor port #2 (M3 and M4)
Adafruit_StepperMotor *stepper2 = driverBoard.getStepper(200, 2);

#define STEPPER_MODE DOUBLE //SINGLE, DOUBLE, INTERLEAVE, MICROSTEP

void forward1() {
  stepper1->onestep(FORWARD, STEPPER_MODE);
}
void backward1() {
  stepper1->onestep(BACKWARD, STEPPER_MODE);
}
void forward2() {
  stepper2->onestep(FORWARD, STEPPER_MODE);
}
void backward2() {
  stepper2->onestep(BACKWARD, STEPPER_MODE);
}
AccelStepper accStepper1(forward1, backward1);
AccelStepper accStepper2(forward2, backward2);

void setup() {
  Serial.begin(115200); //default speed 9600 bps
//  while (!Serial);

// Quadrature encoders
// Left encoder
  pinMode(c_LeftEncoderPinA, INPUT_PULLUP);      // sets pin A as input, and turn on pullup resistors
  pinMode(c_LeftEncoderPinB, INPUT_PULLUP);      // sets pin B as input, and turn on pullup resistors
  attachInterrupt(digitalPinToInterrupt(c_LeftEncoderInterrupt), HandleLeftMotorInterruptA, RISING);

// Right encoder
  pinMode(c_RightEncoderPinA, INPUT_PULLUP);      // sets pin A as input, and turn on pullup resistors
  pinMode(c_RightEncoderPinB, INPUT_PULLUP);      // sets pin B as input, and turn on pullup resistors
  attachInterrupt(digitalPinToInterrupt(c_RightEncoderInterrupt), HandleRightMotorInterruptA, RISING);

  driverBoard.begin(1600); //default frequency 1600 Hz
//  stepper1->setSpeed(60);  //rpm, Ada function
//  stepper2->setSpeed(60);  //rpm, Ada function
  accStepper1.setMaxSpeed(400);
  accStepper2.setAcceleration(5);
  accStepper1.setSpeed(100);
  accStepper1.move(800);
  accStepper2.setMaxSpeed(400);
  accStepper2.setAcceleration(5);
  accStepper2.setSpeed(100);
  accStepper2.move(800);
}

void loop()
{
  //FORWARD - BACKWARD
  //SINGLE - DOUBLE - INTERLEAVE - MICROSTEP
//  stepper1->step(150, FORWARD, SINGLE); //Ada function
//  stepper2->step(150, FORWARD, SINGLE); //Ada function
  accStepper1.run();
  accStepper2.run();

  Serial.print(_LeftEncoderTicks);
  Serial.print("\t");
  Serial.print(_RightEncoderTicks);
  Serial.print("\n");
}

// Interrupt service routines for the left motor's quadrature encoder
void HandleLeftMotorInterruptA()
{
// Test transition; since the interrupt will only fire on 'rising' we don't need to read pin A
  _LeftEncoderBSet = digitalReadFast(c_LeftEncoderPinB);   // read the input pin

// and adjust counter + if A leads B
  #ifdef LeftEncoderIsReversed
    _LeftEncoderTicks -= _LeftEncoderBSet ? -1 : +1;
  #else
    _LeftEncoderTicks += _LeftEncoderBSet ? -1 : +1;
  #endif
}

// Interrupt service routines for the right motor's quadrature encoder
void HandleRightMotorInterruptA()
{
// Test transition; since the interrupt will only fire on 'rising' we don't need to read pin A
  _RightEncoderBSet = digitalReadFast(c_RightEncoderPinB);   // read the input pin
  Serial.print(digitalReadFast(c_RightEncoderPinA) + "\t");
  Serial.println(digitalReadFast(c_RightEncoderPinB));

// and adjust counter + if A leads B
  #ifdef RightEncoderIsReversed
    _RightEncoderTicks -= _RightEncoderBSet ? -1 : +1;
  #else
    _RightEncoderTicks += _RightEncoderBSet ? -1 : +1;
  #endif
}
```