Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #45503

# Expanding Prosthetic Applications: A Multidisciplinary Approach to Assistive Technology for Bedridden Patients

**Prof. Bala Maheswaran, Northeastern University**

Bala Maheswaran, PhD COE Distinguished Professor Northeastern University 367 Snell Engineering Center Boston, MA 02115

**Meghna Sridhar, Northeastern University**
**Alexander Sebastian Aschinberg, Northeastern University**
**Reid Beland, Northeastern University**
**Siddharth Athota, Northeastern University**

# Expanding Prosthetic Applications: A Multidisciplinary Approach to Assistive Technology for Bedridden Patients

Meghna Sridhar, Alexander Aschinberg, Reid Beland, Siddharth Athota, and Bala Maheswaran
College of Engineering
Northeastern University

## Abstract

The field of prosthetics has seen significant growth in recent years, particularly in aiding bedridden individuals. We aimed to expand prosthetic applications to a broader range of patients, including those with autoimmune diseases or conditions causing them to be bedridden. We emphasized the educational value of core engineering principles throughout this project, including system integration, iterative design, and problem-solving.

Our structured engineering process began with defining requirements, followed by conceptual design, detailed analysis, and iterative prototyping. We focused on mechanical functionality, electrical integration, and user-centered design to ensure the device met practical needs. Our solution—a remote-controlled car with a prosthetic arm—was designed to grasp and deliver items to bedridden individuals. This project offered invaluable learning experiences in electrical engineering, programming, and mechanical design, while adhering to systematic engineering practices.

We faced several challenges, such as resolving electrical issues with the Raspberry Pi, wiring the fingers to a central control point, and ensuring a secure grip on objects. Despite these obstacles, the prototype demonstrated both efficiency and user-friendliness. In this paper, we will discuss potential improvements for future iterations and highlight the educational benefits of the engineering process and iterative testing.

## Introduction

Prosthetic technology, which dates to around 950 BC, has long played a crucial role in improving the lives of individuals with physical disabilities, including those with missing limbs. Over the centuries, prosthetics have evolved from basic designs to highly sophisticated systems that enhance mobility and functionality. Recent developments have led to the creation of fully functional prosthetic robotic arms that closely mimic natural limb movements, significantly enhancing user autonomy [1]. Additionally, the incorporation of multimodal embedded sensor systems has improved the dexterity and responsiveness of prosthetic hands, bridging the gap between human and robotic interaction [2]. Today, the integration of robotics with prosthetic limbs represents a major leap forward, offering the potential to assist individuals with limb loss, physical disabilities, conditions such as multiple sclerosis (MS) and dysautonomia, as well as the elderly who struggle with daily movement [3-4].

Patients with MS and dysautonomia experience a variety of complex, fluctuating symptoms that require multifaceted, personalized care. MS often results in mobility impairments, muscle weakness, and coordination difficulties, demanding not only physical assistance but also modifications to the home environment. In addition, cognitive challenges such as memory loss or concentration issues may require support with medication management and routine scheduling. Dysautonomia, with symptoms like heart rate and blood pressure instability, fatigue, and difficulty regulating body temperature, demands constant monitoring and timely intervention to manage sudden physiological changes. For these individuals, a multifunctional healthcare assistant could provide invaluable support, offering both physical aid and assistance with the nuanced task of symptom management, ultimately enhancing their quality of life. This inspiration stems from the experience of a friend who lives with dysautonomia and occasionally suffers from fainting episodes, where immediate access to life-saving treatments, such as Klaralyte Salt Capsules, is critical. A robotic healthcare assistant could significantly improve her ability to manage these episodes independently, reducing reliance on caregivers and improving her overall well-being.

The development of robotic systems for healthcare has made significant progress in recent years. However, the challenge we aimed to address was combining the agility and versatility of robotic mobility with the functionality of prosthetic limbs. Most modern prosthetics are specialized for tasks and lack the flexibility to adapt to a wide range of daily activities. By combining prosthetic arms with full-range robotic movement, we aim to create a system that can adapt to diverse environments and tasks, offering greater mobility and independence to individuals with physical limitations—whether those limitations are temporary or permanent. This evolution from basic prosthetics to advanced robotic systems reflects the ongoing improvement in assistive technologies and their potential to enhance the lives of people with disabilities. By merging the capabilities of prosthetics and robotics, we envision a future where individuals with severe mobility challenges, including those with degenerative diseases like MS and dysautonomia, can regain independence and autonomy.

Our system targets individuals who retain partial arm and hand mobility but experience significant difficulty moving independently [5]. This innovation in assistive technology aims to provide a solution for people who face both physical and cognitive challenges in everyday life, offering practical support in tight spaces, completing a variety of tasks, and ultimately improving the user's independence. Through this project, we seek to demonstrate the potential of assistive robotics and pave the way for future innovations in supporting individuals with disabilities, particularly in medical and healthcare contexts.

## Engineering Educational Values

This project underscores several core engineering educational values, particularly the importance of problem-solving, interdisciplinary collaboration, and human-centered design. The integration of robotics with prosthetic limbs to assist individuals with mobility challenges demonstrates the application of engineering principles to real-world medical issues. Students engaged in this project develop a deeper understanding of how engineering innovations can improve lives, particularly for those with complex and fluctuating health conditions like multiple sclerosis (MS) and dysautonomia.

Through the design and testing of a multifunctional healthcare assistant, this project emphasizes the importance of creativity, adaptability, and systems thinking. Engineering students learn to consider not only the technical aspects of robotics and prosthetics but also the human factors—such as patient needs, environmental constraints, and user experience—that make these technologies effective in practice. The project also illustrates the significance of interdisciplinary knowledge, blending fields like mechanical engineering, robotics, healthcare, and user-centered design.

Moreover, the focus on accessibility and independence for people with disabilities highlights the ethical responsibility engineers bear to create technologies that improve quality of life. This project exemplifies how engineering can foster innovation in assistive technology, advancing the field while making a tangible impact on society's most vulnerable populations.

## Method and Approach

Our first step in our design process was to study existing robots that can move in all directions and grab objects. This research, helped us understand what features were helpful and what could be improved. We focused on making sure our robot would meet the specific needs of patients, helping them become more independent and comfortable. We had several mini prototypes created including a temporary paper prototype (depicted in the top left image in Figure 3a) in which we realized that we wanted to implement the ability to strafe when driving.

When deciding on the best way to make the robot move, we looked at different types of wheels. At first, we thought about using omni wheels, which let a robot move in any direction. But after some tests and comparisons, we chose mecanum wheels instead. These wheels are not only good at moving around in all directions but also handle different types of floors better and are more stable. We used software such as SolidWorks to make sure these wheels would work well in our design. We also used SolidWorks to create a robotic arm that could help patients do things like grab personal items, help with dressing, or move around more easily. After designing our project on AutoCAD (Figure 1a) and SolidWorks (Figure 1b), we were able to find mechanical pieces that could replicate our initial design. For specific parts that were more unique to our project, such as the palm of the prosthetic hand and the base, we 3D printed them using our SolidWorks design, which can be seen in Figure 1b.

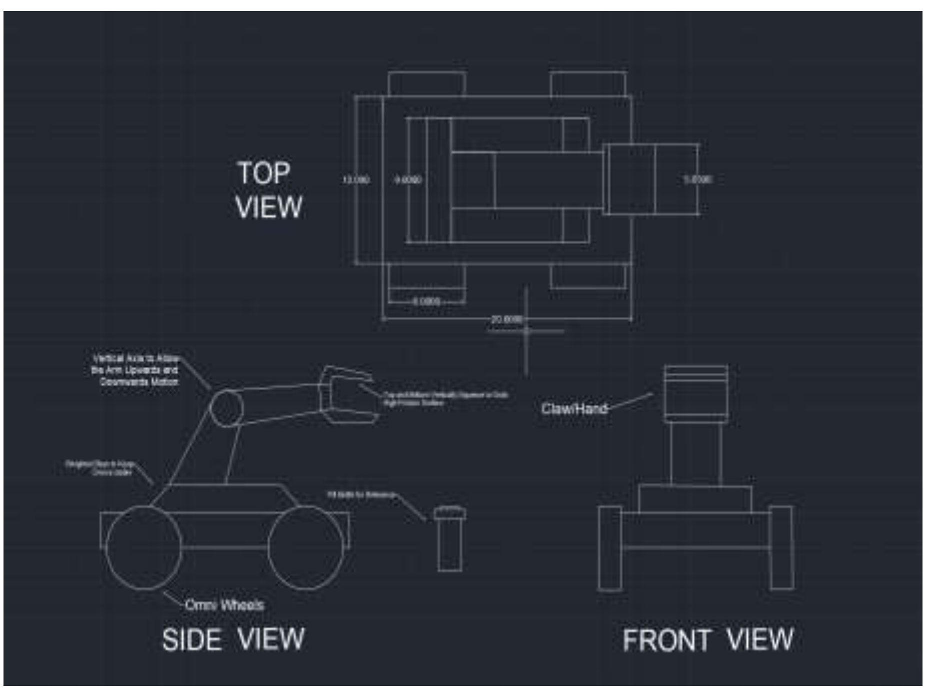



*Figure 1a: Shows AutoCAD drawing for the three views*

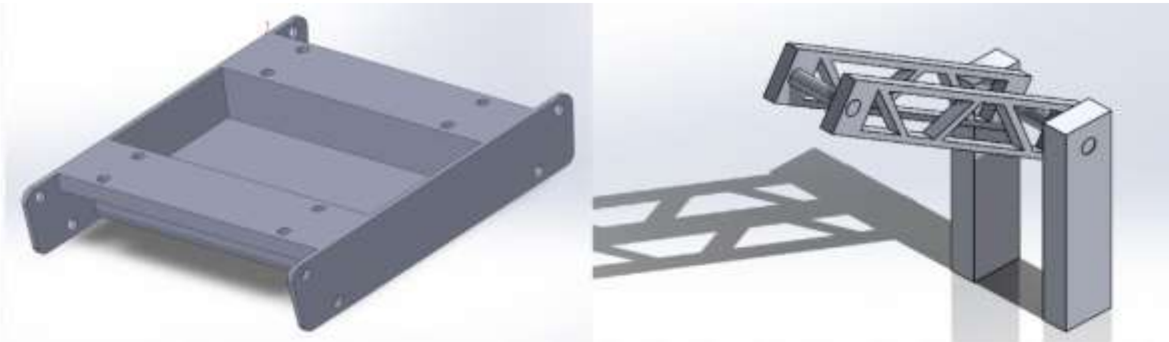


*Figure 1b: Shows two Solidworks designs: the base piece which was 3D printed and used in our final design (left) and an arm design which helped us model the one we were going to use (right).*

Lastly, we made sure that the robot could be controlled easily by patients, even those who aren't good with technology. We designed a simple remote control and tested it with real users to make sure it was easy to use and safe. Our approach aligns with contemporary designs that emphasize scalability and integrated sensor feedback to achieve responsive and user-friendly prosthetic devices [2]. Our goal was to make a robot that could help patients move around and do everyday tasks on their own, making their time in healthcare settings a little easier.

## Prototyping

Our current model uses a program from the developer we bought our mecanum wheels from, as the package came with five 12-volt motors. We used four motors to control each wheel individually to allow the robot to turn on a dime, and then we used the last one to control the arm's up-and-down movement. The pieces we used are depicted below in Figure 2.

*Figure 2: An image of the pieces collected/used in the project. The motors, mecanum wheels, controller, and finger pieces can all be seen. Our Sparkfun kit and 3D printed pieces were utilized as well.*

We chose to get mecanum wheels specifically as they are specialized wheels with rollers oriented at 45-degree angles, allowing movement in any direction without rotating the chassis. This enables precise maneuverability, including lateral (side-to-side) movement, which traditional wheels cannot achieve. These wheels were particularly beneficial for our prototype as they allowed for seamless strafing and fine-tuned control, making our design more versatile.

We then wired a SparkFun motor into our central processing unit to allow for the hand's closing movement. This presented us with some challenges at first, trying to figure out how the fingers would close in sync and if the motor would provide enough torque to do so, but with the correct placement, the motor pulled through and we were able to use the motor for our final.

The controls for the model are simple. A controller contains two joysticks and an array of buttons, similar to any gaming controller. The left joystick controls forward and backward movement, and left and right turns on the spot. The right joystick allows the user to strafe left and right without turning, taking advantage of the mecanum wheels. Buttons on the controller's directional pad allow the user to press and hold to move the arm up and a separate button for down, then separate left and right buttons that control the ability to open and close the hand. The other directional pad has no operational functions used in this project. Our power source is a rechargeable 3.7-volt lithium battery wired to the processing unit,  providing power to each motor.

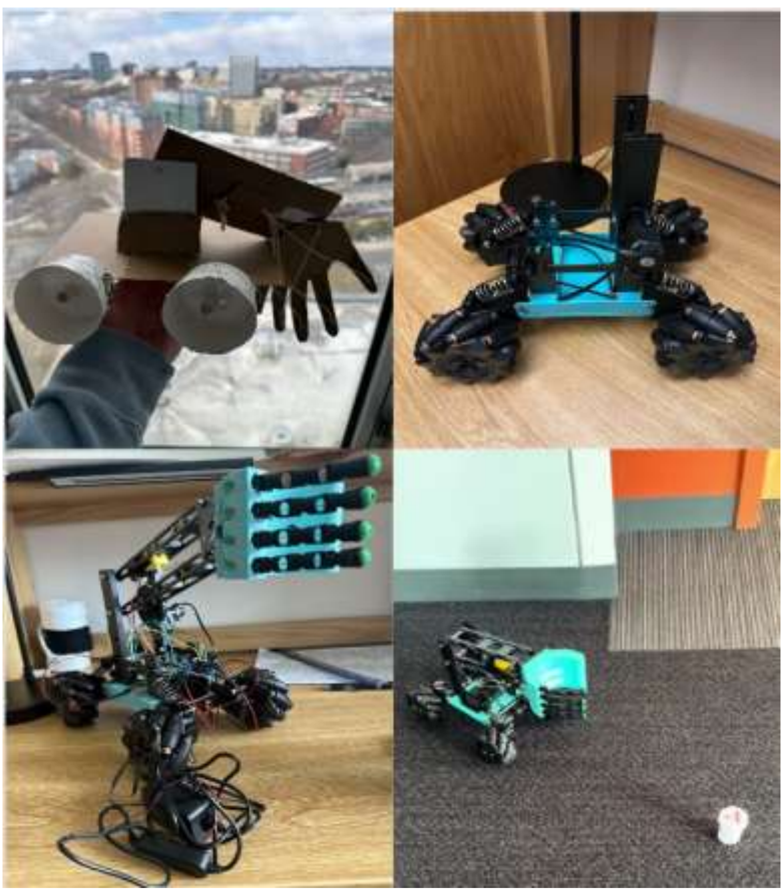


*Figure 3a: The figure above shows 4 different stages starting with the paper prototype (top left), then the base with wheels (top right), the Raspberry Pi wiring and hand (bottom left, and finishing with the final design (bottom right).*

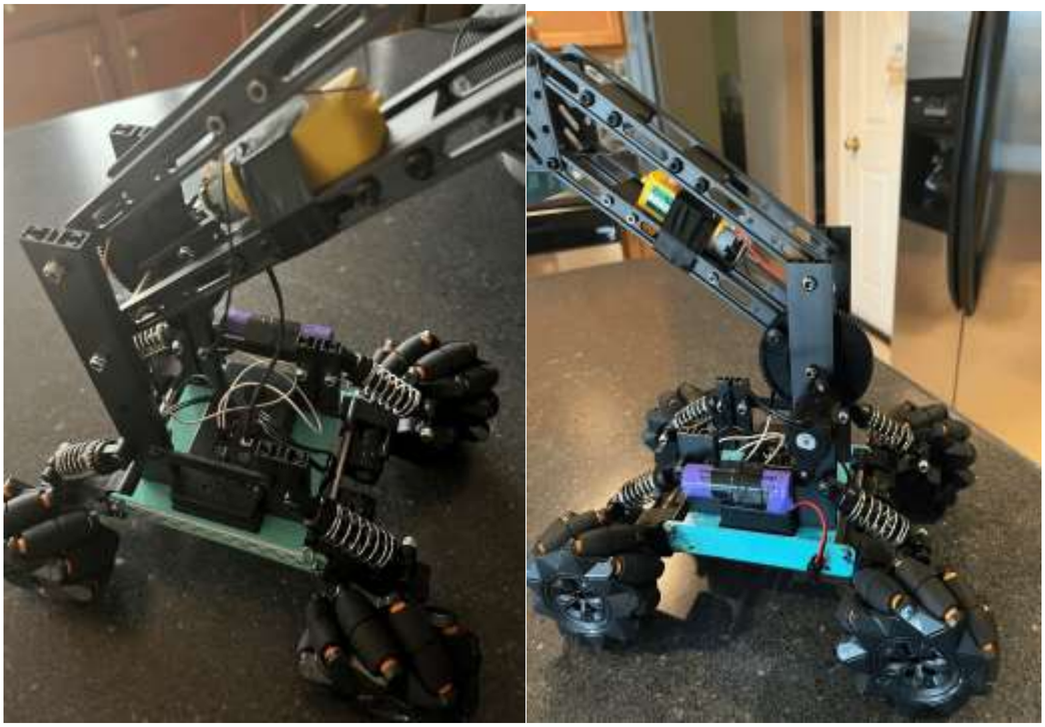


*Figure 3b: Final Prototype*

The original wiring for the Raspberry Pi included the same six motors. Initially, we attempted to wire the motor's logical control and power straight through the Raspberry Pi. Still, we realized that the Raspberry Pi doesn’t transmit enough current to obtain this result. So, instead, we opted to use three motor-controlling units controlled by the Pi's logical power and operated through the electrical power of an individual battery pack. Each unit, named the L298n, controlled two motors, and operated off of four AA batteries in a SparkFun battery pack. To connect the SparkFun battery pack, the barrel jack was removed and the wires were stripped to be directly connected to the ground and 5-volt ports on the L298n module. This was then wired to the GPIO pins on the pi through the iN1, 2, 3, and 4 inputs on the L298n module.

We researched and identified a code (Appendix 1) to control each motor using the Raspberry Pi [6], assigning the outputs to specific GPIO pins. The code utilized simple Python commands, such as forward and reverse. To enhance user accessibility and control, we considered developing an app that would Bluetooth-connect to an Xbox controller, allowing the user to easily direct the robot's movements. Unfortunately, we could not reach the implementation stage for this feature.

For unknown reasons, our motors would not operate despite our efforts. The code was successfully uploaded, and the Raspberry Pi and motors received power [6], as confirmed by the power indicators. However, after many hours of troubleshooting, including changing the GPIO pins in the code to align with our wiring, the motors still did not function. We ultimately concluded that the issue was likely due to a problem with the wiring or an incompatibility between the Raspberry Pi and the motor drivers we used.

Therefore, we looked for alternate control sources, which brought us back to the one which came with the wheel set. This one had a simpler connection since it was designed to integrate with the wheels, but we still needed to rework the wiring to link the arm functions to something functional. We did this by utilizing the robotic arm controlled by an Arduino-based platform with the kit. Fortunately, after some tweaking with the wires from our Sparkfun kit, we were able to get the whole project functional. This allowed us to leverage the Arduino programming language already used with the kit and modify it to work for our prosthetic hand. Using the Arduino Integrated Development Environment (IDE), we were able to write, test, and upload our code to the microcontroller, effectively managing the robotic arm's movements and functionalities (Figure 4).

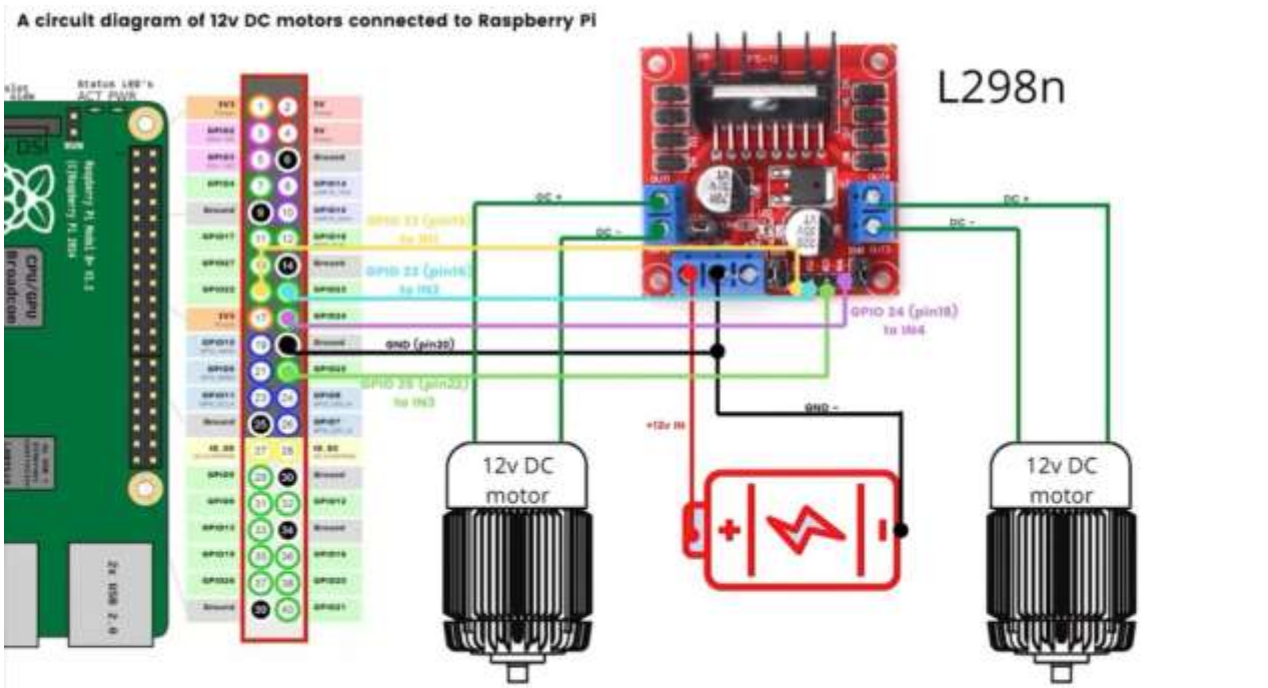
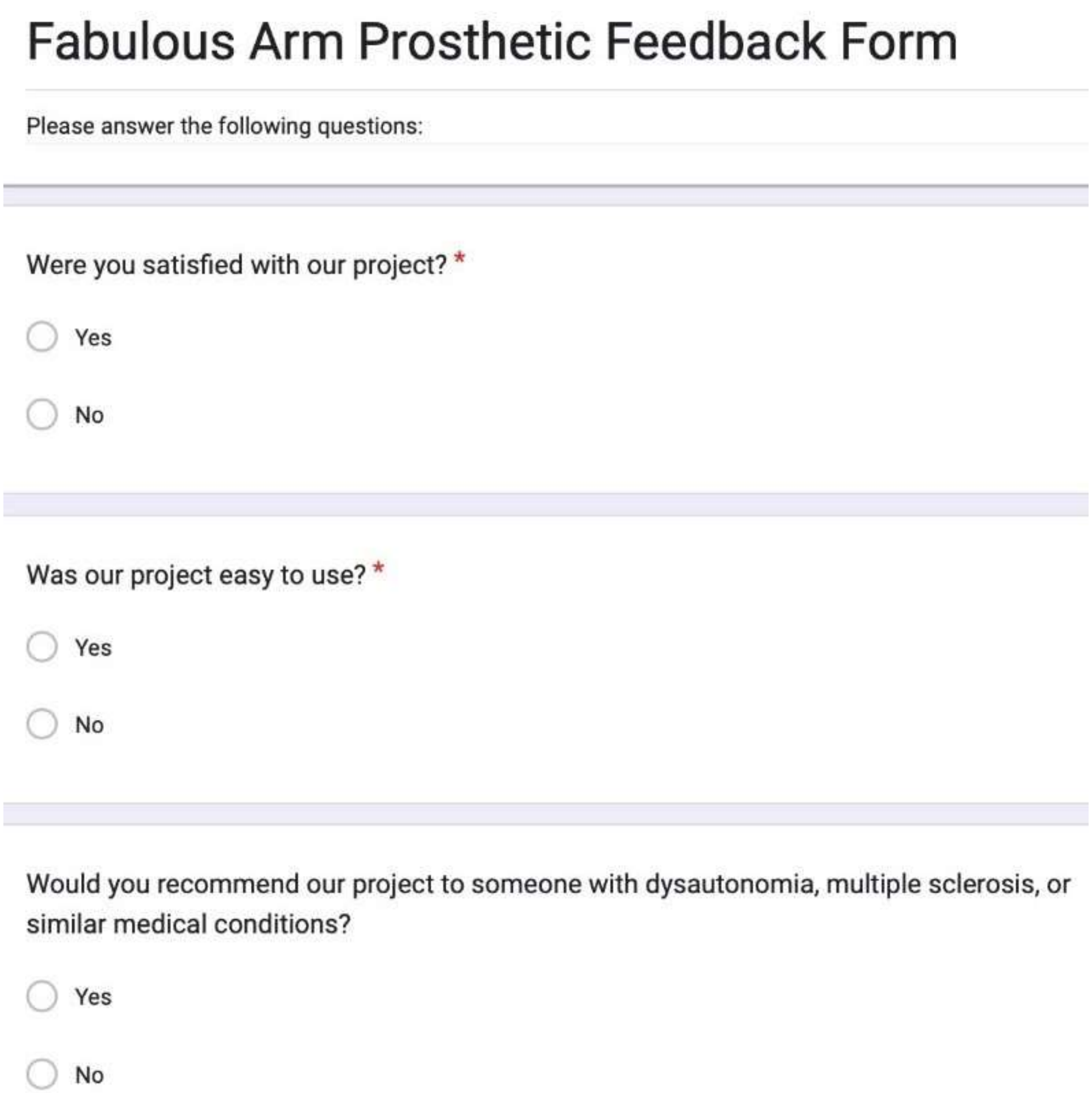



*Figure 4: Circuit Component Images for the Raspberry Pi as well as the motor drivers and power.*

**Results and Analysis**

For the data collection aspect of our project, we conducted a series of tests to evaluate the performance of our healthcare robot. These tests were carried out in an even-terrain environment where the robot was tasked with completing specific actions, such as navigating the space and handling objects like pill bottles. We recorded the robot's movements and task execution, gathering detailed quantitative data on aspects like speed, accuracy, and reliability with multiple test runs. To test speed, we ran a series of tests with our robot driving forward on a treadmill. We observed that five miles per hour was where our robot capped in speed.

In addition to the quantitative data, we also collected qualitative feedback from potential end-users, which included a range of elementary school students, college students and teachers, and middle-aged and elderly adults. The survey we used can be seen in Figure 5a. After the testing sessions, we had our users fill out feedback surveys after these participants could interact with the robot and share their thoughts on its performance and usability. This combination of numerical data and user feedback was crucial for identifying areas where the robot needed improvement.



Fabulous Arm Prosthetic Feedback Form

Please answer the following questions:

Were you satisfied with our project? *

- Yes
- No

Was our project easy to use? *

- Yes
- No

Would you recommend our project to someone with dysautonomia, multiple sclerosis, or similar medical conditions?

- Yes
- No

*Figure 5a: This is what our survey looked like, and the questions which we asked the participants.*

**Discussion**

The practicality and ease of use of our new robot make it an excellent addition to any home environment, which is its primary setting for usage. According to user feedback, the setup, start, and control processes are intuitive and quick, drawing comparisons to the familiar interfaces of video games for younger users and car controls for older users. This approach to design, combining elements from both gaming and existing automotive technologies, appeared to allow users of all ages to quickly become comfortable and proficient in operating the robot. The control system ensures that whether for daily chores or special tasks, the robot enhances home life without adding complexity [5].

Thirty-seven users participated in our survey after testing the project. The feedback, as can be seen in Figure 5b, was overwhelmingly positive. All thirty-seven respondents indicated satisfaction with the project, achieving a 100% approval rating for the question "Were you satisfied with our project?" Similarly, everyone affirmed they would recommend our project to individuals who are bedridden or have physical disabilities, maintaining a 100% score in this category. However, responses to the question "Was our project easy to use?" varied: 31 users found it easy, while six did not, giving us an 83.8% satisfaction rate. The challenges mentioned included the precision required to handle objects without causing disruptions, difficulties in controlling the robot's speed, issues with the robot's hand grip when holding certain items, and some younger users struggling with the controls.

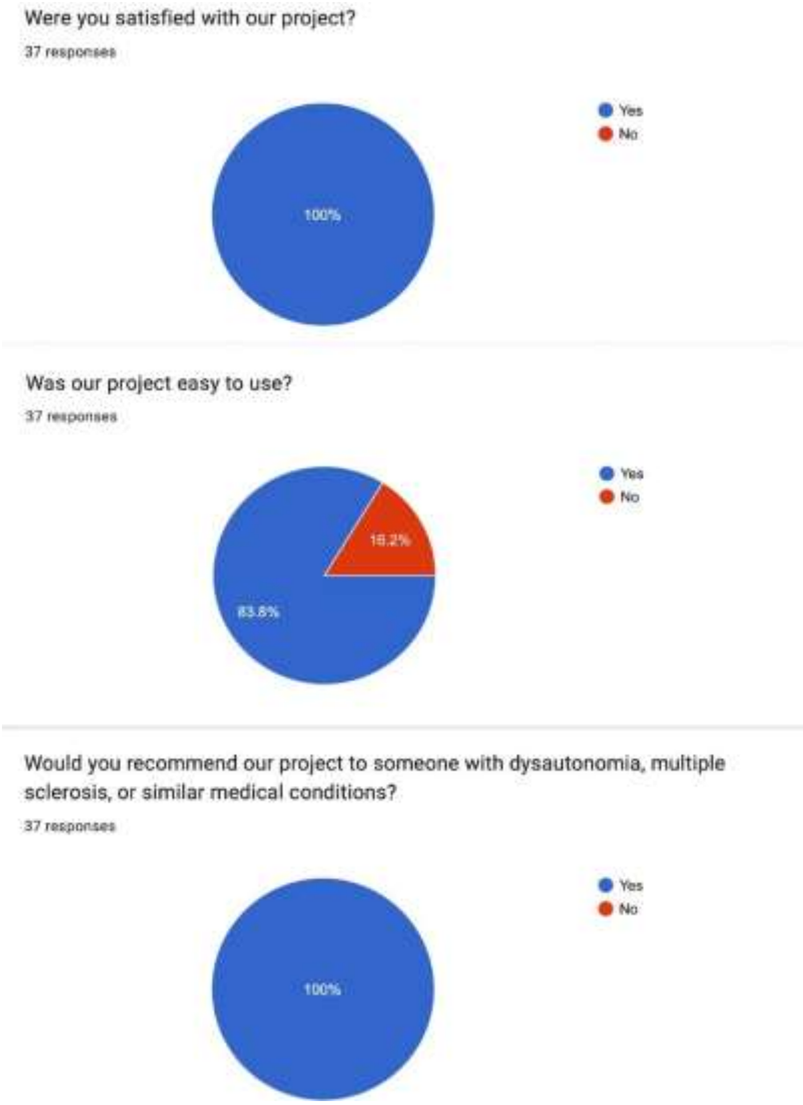



*Figure 5b: Responses*

Overall, our survey findings emphasize that the robot is relatively user-friendly and has an appeal to most, from kids to seniors. Its compact and portable design were highly appreciated, which makes it easy to move and store in any size home or apartment. The robot reached speeds to five miles per hour, which is ideal for quickly responding in emergencies—a major goal in its design. Its mecanum wheels provide excellent maneuverability, and the flexible robotic arm improves its effectiveness in critical situations, ensuring it performs reliably when needed most. However, an important aspect to note is that the users tested the product on flat surfaces. In different terrains, the robot would not have driven as well.

Moreover, the design and technology incorporated into the robot were appreciated by users. Features such as Bluetooth connectivity have gained the interest of our users with many stating they would want this at their houses just as an addition, not just to be used as a medical assistant. Overall, our survey data strongly supports the robot's position as a dependable healthcare assistant and potential household appliance.

## Conclusion and Future Improvements

Our project was initially developed as a proof of concept focusing on achieving a simple and practical design. As we sought to demonstrate the basic feasibility of our idea, we chose to implement a straightforward approach. This initial phase allowed us to establish a solid foundation. Still, it also highlighted areas that could benefit from further refinement to enhance our product's overall functionality and performance.

One significant area for improvement in our project involves the wheel system. Currently, our design includes mecanum wheels, which offer robust maneuverability but come with trade-offs. These wheels require higher power consumption due to the independent control of four motors, and they rely on complex control algorithms using vector-based calculations to coordinate movement. However, these wheels struggle with irregular terrain and are best suited for flat surfaces. Their angled rollers depend on constant ground contact, meaning they lose traction on uneven surfaces, reducing control and making movement unreliable. This limitation is a particular consideration for environments like homes and hospitals, where floors may have rugs, door thresholds, or slight inclines that could disrupt smooth movement. The mecanum wheels' inability to effectively handle bumps or uneven ground suggests that an adjustment of this component may be necessary in future designs. By exploring alternative wheel designs or adjustments, we could increase the robustness and versatility of our product, allowing it to operate more effectively in various environmental conditions.

Furthermore, enhancing the adaptability of the prosthetic hand is necessary. We observed that the hand struggled to pick up smaller items because the palm was too large and fingers were too small. Making these components more adjustable to enable a variety of grasping capabilities would greatly increase the effectiveness of our project.

Additionally, we observed that the gripping strength of the robotic hand needs improvement. The current mechanism in the fingers is weak, making it difficult to grasp and hold heavier objects. The fingers are operated by a single string that connects them all, limiting their range of motion and reducing their functionality. To overcome this limitation, we would redesign the fingers to allow for a fuller range of motion, enabling the hand to perform more complex and varied tasks. This modification will enhance the utility and efficiency of our robotic hand, broadening its potential applications.

## Acknowledgments

The authors would like to express their gratitude to the First Year Engineering Learning and Innovation Center (FYELIC) at Northeastern University, as well as the Red Vest Assistants, for their invaluable support and for providing the resources necessary to develop this product.

**References:**

*These references include authors, titles, source titles, volume and issue numbers (where applicable), pages, publication dates, and DOIs or URLs to access the documents.*


1. Rahman, M. A., Islam, M. S., Hossain, M. A., and Akhand, M. A. H. "Design and Development of Fully Functional Prosthesis Robotic Arm." 2023 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), IEEE, 2023. https://ieeexplore.ieee.org/document/10142370

2. Markovic, M., Dosen, M., Farina, D., and Aszmann, O. C. "A Multimodal Embedded Sensor System for Scalable Robotic and Prosthetic Applications." IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 26, no. 6, IEEE, 2018, pp. 1121–1131. https://ieeexplore.ieee.org/document/8624955

3. Mayo Clinic. (n.d.). Multiple sclerosis (MS) is a potentially disabling disease of the brain and spinal cord (central nervous system). Retrieved from [Mayo Clinic website](https://www.mayoclinic.org/)

4. Kumar, N. N., et al. (2022). Robot arm with gesture control. Journal of Physics: Conference Series, 2325(1), 012012. https://doi.org/10.1088/1742-6596/2325/1/012012

5. Pepe, A., et al. (2016). A hybrid teleoperation control scheme for a single-arm mobile manipulator with omnidirectional wheels. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). https://doi.org/10.1109/IROS.2016.7759236

6. Xu, L. (2018). Remote control and monitoring system of robotic arm using Raspberry Pi. In Advances in Intelligent Systems and Computing (pp. 642–648). https://doi.org/10.1007/978-3-030-00214-5_81

**Appendix 1:**

The python code used to run the Raspberry Pi. This code was taken online, however our code was different as we changed the GPIO numbers as well removed some of the code that didn't have a use to our project. This code was repeated three times to control each L298n

```
import curses
from gpiozero import Robot, Motor
robot = Robot(left=Motor(4, 14), right=Motor(17, 18))
actions = {
    curses.KEY_UP:      robot.forward,
    curses.KEY_DOWN: robot.backward,
    curses.KEY_LEFT:  robot.left,
    curses.KEY_RIGHT: robot.right,
}

def
    main(window):
    next_key        =
    None        while
    True:
        curses.halfdelay(1)
        if next_key is None:
            key =
            window.getch()
        else:
            key = next_key
            next_key = None
        if key != -1:
            # KEY PRESSED
            curses.halfdelay(3)
            action =
            actions.get(key) if
            action is not None:
                action()
            next_key =
            key
            while next_key == key:
                next_key = window.getch()
            # KEY RELEASED
            robot.stop()
curses.wrapper(mai)
```

**Appendix 2:** Source Code for the Python

Python Code:

```
import curses
from gpiozero import Robot, Motor, Servo

# Setup for the robot's wheels
robot = Robot(left=Motor(4, 14), right=Motor(17, 18))

# Setup for the prosthetic arm servo
arm_servo = Servo(22)

# Define actions for both robot and prosthetic arm actions
= {
    curses.KEY_UP: robot.forward,
    curses.KEY_DOWN: robot.backward,
    curses.KEY_LEFT: robot.left,
    curses.KEY_RIGHT: robot.right,
    ord('o'): arm_servo.min,  # Open the prosthetic arm ord('c'):
    arm_servo.max,  # Close the prosthetic arm ord('r'):
    arm_servo.mid  # Return arm to neutral position
}

def
    main(window):
    next_key        =
    None        while
    True:
        curses.halfdelay(1)
        if      next_key      is
        None:
            key                         =
        window.getch() else:
            key  =  next_key
            next_key        =
            None

        if key != -1:
            # KEY PRESSED
            curses.halfdelay(3)
            action =
            actions.get(key)
            if action is not None:
                action()
            next_key = key
            while next_key == key:
                next_key = window.getch()
            # KEY RELEASED
            if key in (curses.KEY_UP, curses.KEY_DOWN, curses.KEY_LEFT,
```

curses.KEY_RIGHT):
robot.stop()  # Only stop the robot's movement, not the arm curses.wrapper(main)

**Appendix III: Source code used with the Arduino IDE and modified with the kit**

```
#include <Servo.h>

// Pin assignments
int pins[] = {4, 14, 17, 18}; // {leftForward, leftBackward, rightForward, rightBackward}
int servoPin = 22;

// Create servo object
Servo armServo;

void setup() {
  for (int pin : pins) pinMode(pin, OUTPUT); // Set motor pins as outputs
  armServo.attach(servoPin);                // Attach the servo
  Serial.begin(9600);                       // Setup Serial communication
}

void loop() {
  if (Serial.available() > 0) executeCommand(Serial.read());
}

void executeCommand(char cmd) {
  int states[4]; // Motor pin states: {leftForward, leftBackward, rightForward, rightBackward}
  switch (cmd) {
    case 'w': states[0] = states[2] = HIGH; states[1] = states[3] = LOW; break;   // Forward
    case 's': states[1] = states[3] = HIGH; states[0] = states[2] = LOW; break;   // Backward
    case 'a': states[1] = states[2] = HIGH; states[0] = states[3] = LOW; break;   // Left
    case 'd': states[0] = states[3] = HIGH; states[1] = states[2] = LOW; break;   // Right
    case 'o': armServo.write(0); return;    // Open arm
    case 'c': armServo.write(180); return;  // Close arm
    case 'r': armServo.write(90); return;   // Neutral arm
    default: memset(states, LOW, sizeof(states)); break; // Stop motors
  }
  for (int i = 0; i < 4; i++) digitalWrite(pins[i], states[i]); // Apply states
}
```