

# Sensor-Based Exoskeletal Mirroring System

Vishnu Iyengar

12th Grade, Rancho Bernardo High School  
San Diego, CA, USA

Leo Sitaraman

12th Grade, Homestead High School  
Cupertino, CA, USA

## Abstract

Upper-limb impairment after stroke or motor nerve injury most often leaves patients without fine control of the hand. This paper presents a low-cost, sensor-based exoskeletal hand mirroring system in which a measurement glove worn on the intact hand records finger joint angles with magnetic rotary encoders, and a cable-driven exoskeleton reproduces those motions on the impaired hand. Unlike designs that couple the finger joints, the exoskeleton actuates the metacarpophalangeal (MCP) and proximal interphalangeal (PIP) joints of the index, middle, and ring fingers independently, plus a single-degree-of-freedom thumb, while keeping all actuators on the forearm. The glove tracked finger position with a mean absolute error of  $1.19^\circ$  and a lag of about 312 ms, introduced mainly by smoothing. The exoskeleton followed commanded motions with 72–140 ms of lag but had reduced joint angle accuracy for certain motions due to the nature of a differential string drive, with a mean absolute error of  $15\text{--}25^\circ$ . The exoskeleton exerted a maximum force per finger of around 1.2 N. The results establish the feasibility of independent dual-joint mirroring with actuation hardware kept off the hand.

## Index Terms

Hand exoskeleton, rehabilitation robotics, bilateral mirroring, cable-driven actuation, stroke

## I. INTRODUCTION

In the United States alone, roughly 7–8 million adults live with the lasting aftermath of stroke, and between 55 and 75 percent of them report persistent impairment of the upper limb. A second group is composed of people whose motor nerves have been damaged in vehicle collisions, industrial accidents, and combat. Both groups continue to grow, and both are disproportionately composed of underserved populations, such as older adults and lower- to middle-income communities, where limb function is most essential to daily life and rehabilitation is least available.

Impairment of the hand is where disability is felt most acutely. In less severe cases, patients can often recover gross arm movement to some degree, but finer control, the ability to shape the hand in anticipation of an object, returns far more rarely. Under the current standard of care, the response to these cases is repetition-based therapy. Patients are often left without restored bilateral function, and most respond by using the impaired hand less and less, allowing the disability to progress.

### A. Related Work

Several research groups have developed mirroring systems in which a sensor glove worn on the unaffected hand drives an actuated device on the impaired hand. Triwiyanto et al. [1] built an open-source, 3D-printed exoskeleton in which five IMU sensors on a glove wirelessly command five micro servos, one per finger, reaching a mean tracking error of  $9.04^\circ$ . In this design, each servo pushes the proximal phalanx through a rod, and the distal segment follows through a fixed linkage. Bryantono et al. [2] paired a glove with bend sensors on the unaffected hand with pneumatic soft silicone actuators on the affected hand, one per finger. The actuators are injection-molded in a single shot from liquid silicone rubber and can bend up to  $180^\circ$ , while the sensing glove uses low-cost graphite strain gauges drawn in pencil on paper. Both systems are designed for rehabilitation exercises rather than functional use in daily activities, and both drive each finger with a single actuator, so the joints within a finger cannot move independently. Other research systems [3]–[7] are similarly focused on rehabilitation. Of the solutions that are commercially available [8]–[11], most ignore dual joint measurement, instead working like a tentacle, and do not feature a large range of measurement.

This paper presents a Sensor-Based Exoskeletal Hand Mirroring System: a wearable orthosis that derives actuation commands from sensors mounted on the contralateral limb, restoring grasping capability to individuals with mild

to severe hand impairment while minimizing the cost of achieving a form factor suited to everyday use. Each finger bends primarily at two joints: the metacarpophalangeal (MCP) joint, the knuckle where the finger meets the palm, and the proximal interphalangeal (PIP) joint, the middle joint of the finger. Unlike the systems described above, which drive each finger with a single actuator and are intended primarily for rehabilitation exercises, the proposed system (1) measures and actuates the MCP and PIP joints of the fingers independently, enabling hand postures that coupled designs cannot reproduce; (2) relocates all finger actuators from the hand to the forearm, keeping the hand unencumbered for comfort and ergonomics; and (3) is built from low-cost, off-the-shelf components augmented by custom 3D-printed parts. The goal of this device is to be capable of capturing subtle motor kinematics through joint-angle sensors on the intact hand and accurately recreating them on the paralyzed hand via lightweight actuators. The system would need to achieve real-time motion synchrony within sufficiently small latency, wearability, and force safety output in order to work well in everyday activities. Furthermore, it must be able to be developed at a cost which is affordable to the general public.

## II. SYSTEM

### A. System Architecture

To achieve high dexterity and mirroring accuracy, the aim was to measure and actuate both MCP and PIP joints on the index, middle, and ring fingers. Many existing exoskeletons only control the MCP joint, or have a single-actuator coupling between MCP and PIP joints, creating a curling motion of the finger as it contracts and releases [1], [2], [5], [6]. The central design choice of this work is to break this coupling: by using two actuators per finger with independent control of both joints, mirroring the two joint angles measured independently on the intact hand, more complex hand positions can be achieved, facilitating many Activities of Daily Living (ADLs). Rather than restricting the movement of both joints in a certain way, independent actuation enables gripping of many oddly shaped objects by allowing fingers to adjust to the shape. One practical example is extracting a book from a bookshelf. This is much more difficult to do when PIP joints are mechanically restricted to bend with MCP joints. To keep the hand unencumbered, all of these actuators are low-cost, off-the-shelf micro servos mounted on the forearm, with power transmitted to the finger joints through cables.

Additionally, many ADLs require asymmetric motions of each hand, e.g., opening a water bottle requires one hand to hold the bottle and the other to unscrew the cap. The nature of a hand-mirroring system prevents asymmetric motions such as this from being executed. For this reason, our system includes a mirroring switch that couples and decouples the two hands. When the mirroring switch is on, the exoskeleton mirrors the joint positions measured by the glove. When it is turned off, the exoskeleton ignores input from the measurement glove and maintains the position it was in until the switch is turned on again. This allows the impaired hand to be positioned in a certain state and held there, allowing for asymmetric motions.

### B. Design Evolution of the Measurement Glove

1) *Mechanical Design:* Early on, effort was placed on determining the suitability of magnetic rotary encoders to measure finger joint angles consistently and to a high accuracy while remaining compact. Mechanical testing was completed with an arrangement made from plastic connectable bricks, so that quick adjustments to alignments and controlled trials of rotation could occur without committing to an established structural design.

Prior to using the rotary encoder and open-source hinge design [12], initial testing indicated that the hinge design would restrict the maximum joint angle in flexed position. As the number of degrees of flexion increased on the finger joint, the rotating arm interfered with functionality and stability by contacting nearby components before the full range of motion was reached at the finger joint. This interference limited the range of finger motion the glove could measure, preventing accurate replication of natural finger movement. To alleviate some of the problems associated with the hinge design, a small metal L-bracket was mounted to the hinge assembly (Fig. 1). The intention of this modification was to provide additional clearance between the rotating finger arm and the hinge assembly during the flexed position of the finger joint. With additional clearance provided by extending the rotating finger arm out from the hinge axis, the L-bracket modified the angle formed by the two surfaces (L-bracket and rotating finger arm) causing no premature surface contact and thus allowing the finger joint to have a greater angular displacement without interference (i.e., hitting adjacent components).

The initial biomechanical studies on range of motion conducted with paper and balsa wood models indicated an important mechanical limitation; as each segment of the digit moves from extension to flexion, the three segment positions change substantially. Because of this, there are compressive and expansive forces acting on the segment position that would create difficulties for rigid transmission mechanisms and could misalign joints. This led to the development of a 3D-printed slider mechanism (Fig. 1), which is designed to accommodate translation while maintaining smooth rotational motion.

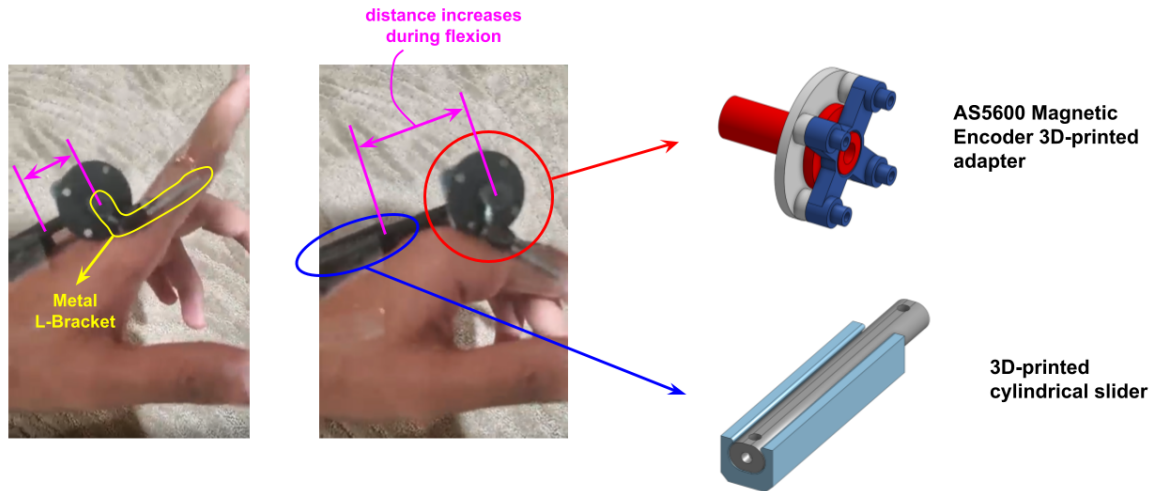


Fig. 1. Joint mechanism of the measurement glove. Left: the metal L-bracket on the finger. Center: the distance between joint mounts increases during flexion, which the slider accommodates. Right: CAD models of the 3D-printed adapter for the AS5600 magnetic encoder and the 3D-printed cylindrical slider.

After validating the initial single-joint sensing device, a motion sensing device capable of monitoring the full hand was designed and built by incorporating a total of six magnetic rotary encoders, measuring flexion at the MCP and PIP joints of the index and ring fingers, the MCP joint of the middle finger, and a single joint of the thumb (Fig. 2). A PIP sensor on the middle finger was omitted because of limited space on the back of the hand. The encoders were attached to an off-the-shelf rubber glove using custom 3D-printed mounts. Each encoder was centered on the rotation axis of the joint it measures and secured so that it stays aligned with that axis as the finger moves. The wiring for each encoder was routed along the glove's structural components to help minimize the amount of mechanical strain and eliminate the introduction of any opposing torque or friction that could interfere with natural finger movement.



Fig. 2. Dual-joint measurement glove with six magnetic rotary encoders measuring the MCP and PIP joints

2) *Code and Electronics*: A significant technical challenge was encountered when integrating all six encoders, as they share a single fixed I<sup>2</sup>C address [13]. Instead of adding additional hardware, such as a dedicated I<sup>2</sup>C multiplexer, the problem was solved in software using bit-banging, a technique in which the I<sup>2</sup>C protocol is implemented by toggling microcontroller pins directly in software rather than with a dedicated hardware peripheral [14]. Each encoder was isolated on its own stand-alone, software-defined I<sup>2</sup>C bus, which was implemented by controlling the clock and data signals with standard general-purpose input/output (GPIO) pins on the Arduino Mega 2560 (Fig. 3). By doing this, the Mega 2560 simulates two independent I<sup>2</sup>C buses for each of the six encoders and allows each encoder to be polled independently, thus eliminating address conflicts. Avoiding the multiplexer also keeps the glove compact and inexpensive, since no additional components need to be mounted on the hand. Although bit-banging carries a much greater processing overhead than the hardware I<sup>2</sup>C peripheral, the Mega 2560 has sufficient processing resources to maintain a reliable sample rate for real-time motion sensing.

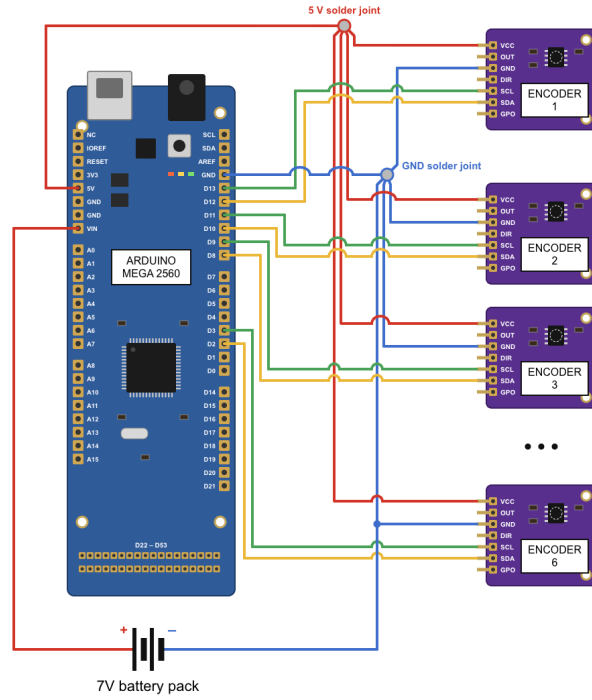


Fig. 3. Encoding glove wiring diagram

With reliable multi-sensor acquisition established on the Arduino Mega 2560, the joint angle data next needed to be transmitted off the board for visualization and control. Because the two subsystems were developed roughly 450 miles apart, the measurement glove in San Diego, CA and the exoskeleton in Sunnyvale, CA, the data link also needed to work over the internet so that the complete system could be developed and tested remotely. Instead of creating a traditional serial monitor or a custom socket server, a relay system was created using the Discord API [15] as a lightweight communication bridge. The Mega 2560 sent formatted joint angle data over serial to a host computer, where a Python script was used to parse the incoming data in real time. The script interfaced with the Discord API via a bot token, posting well-formed data messages to a private channel on a server (Fig. 5). On the receiving end, a second Python client subscribed to that channel, extracted the numeric data from incoming messages, and translated them into control inputs for the actuation system. The result was a cloud-based message relay for sensor data flowing from the Arduino Mega 2560 to the actuation system (Fig. 4), allowing the glove in San Diego to drive the exoskeleton in Sunnyvale.

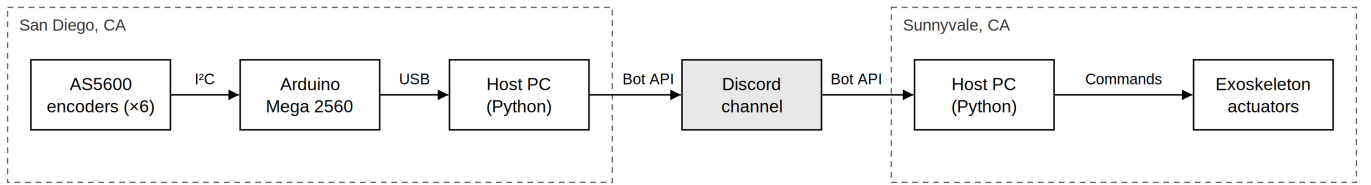


Fig. 4. Remote testing pipeline

Although unconventional, this approach offered several advantages for prototyping. Because messages were relayed through a Discord text channel, no direct socket configuration or firewall reconfiguration was required. Every message was also automatically logged in the channel history, providing a built-in record of each test. During testing, latency measurements indicated that the relay system's performance was adequate for proof-of-concept synchronization but not for the final low-latency implementation.

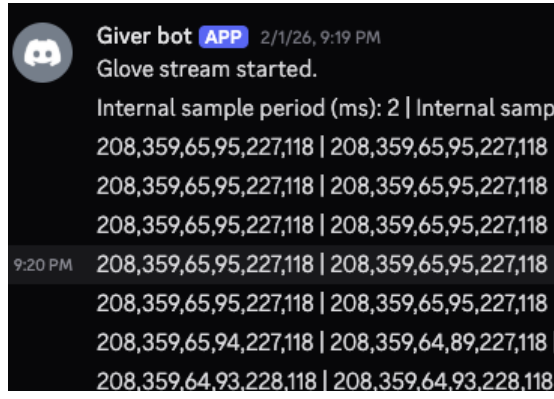


Fig. 5. Discord bot posting glove joint-angle data to a private channel

When the glove was wired directly to the actuators, the lag dropped significantly as the data no longer passed through the Discord relay. However, since the encoders have over 4000 unique positions, significant jitter appeared even when the hand was completely still. To solve this issue, a moving-average filter was applied, so that each reported joint angle is the average of the 10 most recent encoder readings. Although this filter eliminated the jitter, it introduced a lag of about 300 ms between the finger's motion and the measured angle, as shown in Section III.

### C. Design Evolution of the Actuation System

1) *Mechanical Design:* To approach the design of a complete exoskeleton with many degrees of freedom, the process began by developing a single two-stage finger with independent actuation of the MCP and PIP joints, the core capability that distinguishes this system from single-actuator designs. The main challenge faced when designing this finger was achieving decoupled motion for the PIP stage without an actuator mounted on the MCP stage. Due to space constraints with many fingers side-by-side, and to keep the hand light and comfortable for the user, it is not feasible to place an actuator on the MCP section of the finger, moving up and down with MCP actuation. Instead, both actuators must reside on the fixed frame, keeping all actuator mass off the moving finger, and power must be transferred to both joints accordingly. This cannot be achieved with traditional power transfer methods due to the changing relative positions of the PIP joint location and the PIP actuator, as shown in Fig. 6.

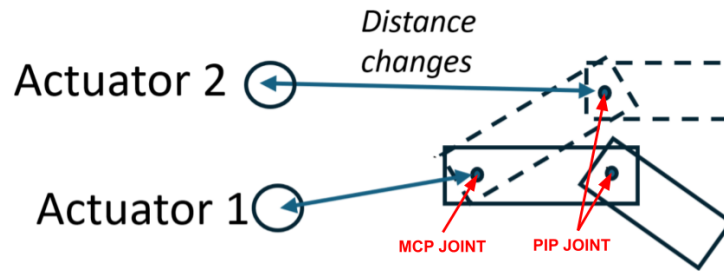


Fig. 6. Non-constant actuator distances during two-stage finger motion

To address this, a cable routing system was developed that allowed fully decoupled power transfer from the PIP actuator to the PIP joint in all positions of the MCP joint. The mechanical components used were mainly 3D-printed with low-cost PLA, allowing for rapid prototyping and accurate, custom parts at a small scale, while actuation was provided by inexpensive, off-the-shelf MG90S micro servos. To achieve independent PIP motion, a cable pulled by the PIP actuator was directed through the center of the MCP stage, then along the body of the MCP stage, and to a pulley on the PIP joint (Fig. 7). Because the cable runs exactly along the axis of rotation of the MCP joint, it is not affected by MCP rotation. For MCP actuation, a cable was used to connect the servo arm to a pulley. In

this first iteration, the cables were made of upholstery thread. Cable actuation was used due to its capability for direction changes in power transfer and the flexibility to place actuators in any location, including away from the hand on the forearm, a task that would be significantly more challenging and complex to achieve with other forms of power transfer such as linkages, gears, or belts.

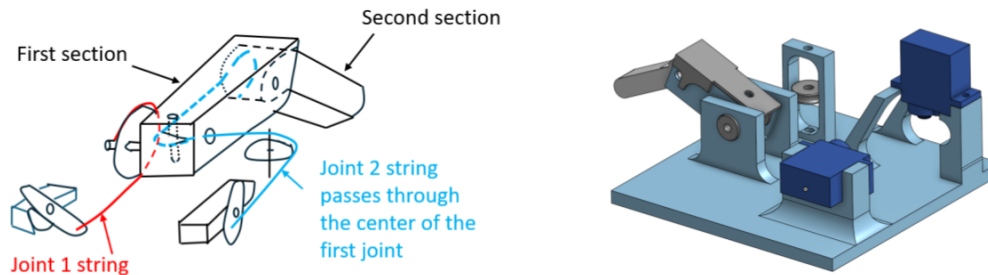


Fig. 7. Axial cable routing through the MCP pivot

While the design performed well on the test bench, the testing base used was very large in size and could not be scaled to handle three fingers side by side. Because each finger had a cable exiting horizontally from the MCP axis, significant width-wise clearance between fingers was required, which is not achievable with human fingers being less than 2.5 cm apart from each other. Additionally, the design required springs to return the fingers to extension, as cables can only provide tension, not compression.

To address these issues, a differential cable routing system was developed that allowed for independent actuation of both stages, with both cables exiting the finger toward the wrist rather than horizontally, so that adjacent fingers can sit side by side and both cables can be routed to the actuators on the forearm. As shown in Fig. 8, both cables wrap around the MCP pulley in the same direction but around the PIP pulley in opposite directions. Pulling both cables flexes the MCP joint and releasing both allows it to extend, while pulling one cable and releasing the other rotates the PIP joint in either direction, so two actuators fully control both joints.

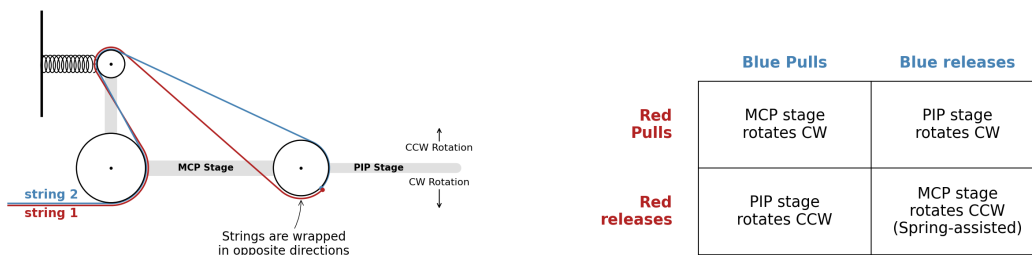


Fig. 8. Differential cable routing system. Both cables wrap the MCP (1st stage) pulley in the same direction and the PIP (2nd stage) pulley in opposite directions; the table gives the resulting joint motion for each combination of cable pulls.

Because the two cables wrap the PIP pulley in opposite directions, the PIP joint no longer requires a return spring: in both actuation directions, there is at least one cable in tension providing the force to rotate the joint. This, however, is not the case for the MCP joint, since both cables wrap its pulley in the same direction and can only flex it. A return spring, implemented as a rubber band in this prototype, is therefore still required to return the joint to its initial position when both actuators release their cables. This spring is not an ideal solution: the actuators must work against it throughout MCP flexion, reducing the force available to the finger, and its elasticity

makes the joint less rigid and harder to position precisely. It was later eliminated by adding a dedicated extension cable to the MCP joint, as described below (Fig. 12).

Rather than the upholstery thread used for the previous iteration, a high-strength fishing line was used to reduce flexibility. The fishing line also has a smoother surface, reducing friction when rubbing against 3D-printed channels during PIP actuation.

When scaling to multiple fingers, it is no longer feasible to keep all actuators at each MCP joint. Instead, they were mounted further up on the forearm, which also keeps the hand itself light and unencumbered. To transfer power effectively, flexible silicone tubes were used to guide the cables from the actuators to the mechanical fingers. This allowed the cables to flow through slight bends and take non-linear paths without significant resistance, similar to the Bowden cables commonly used in bicycle shifters and brakes. Since each finger required two actuators, each driving a cable pulley, a compact 3D-printed housing was designed to hold both servos (Fig. 9).

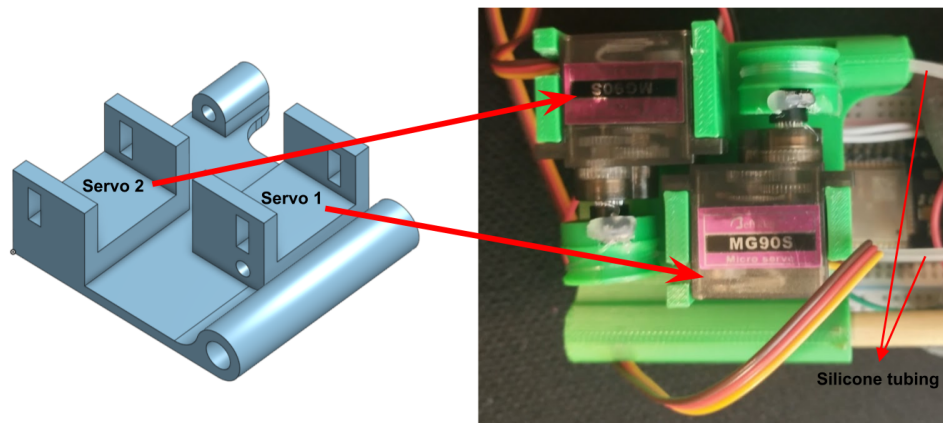


Fig. 9. Compact 3D-printed housing for the two servos of one finger: CAD model (left) and assembled unit, with silicone tubing guiding the cables (right)

In order to avoid relative motion between the actuators and finger stages, it was necessary to construct a rigid frame. Flexibility would allow tension in the cables to be used in bending the system, taking away from the force transferred to the fingers themselves. Inexpensive wooden dowels were used to ensure rigidity and were inserted into slots in a wrist brace to facilitate mounting to a human arm and hand. For efficient and reliable testing, the system was mounted on a wooden hand, serving as an articulated anatomical proxy without introducing confounding human variables, such as active muscle resistance.

In addition to the challenge of achieving independent, decoupled MCP and PIP actuation, the exoskeletal system must actuate physical human fingers, unlike a standard robotic hand. This is a significant obstacle. Just as with the measuring glove, it is not possible to match the axis of rotation of mechanical components with the axis of rotation of the human fingers, meaning there will exist relative motion as components push and pull on the MCP and PIP segments. This relative motion causes friction and contact between the knuckles and the exoskeleton. This issue was addressed by modifying the geometry of each mechanical stage. Various iterations were tested to optimize range of motion and reduce opposing forces. Because of the complex geometries with gaps and overhangs, multiple 3D printing orientations and support structures were tested to avoid sagging filament and layer-line failure under tension. CAD diagrams of the first and second stage finger designs are shown in Fig. 10.

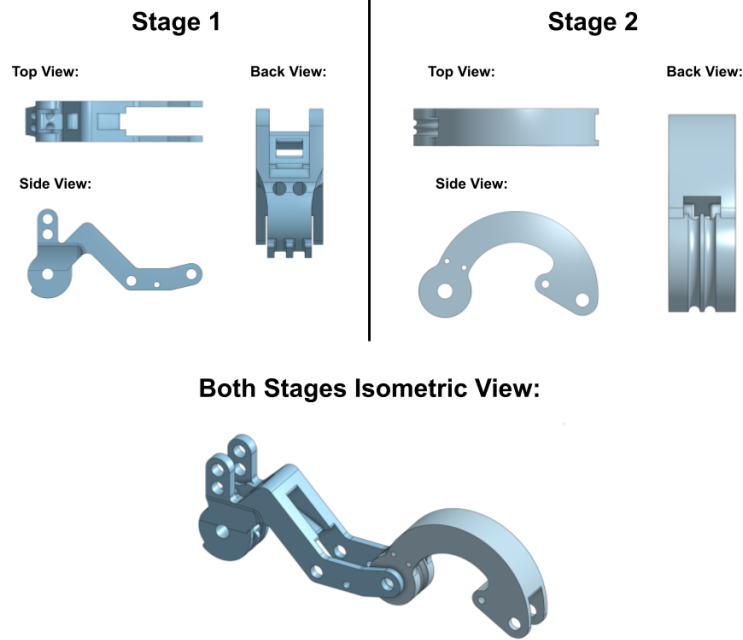


Fig. 10. Top, side, back, and isometric views of finger stage 3D-printed parts in CAD

By creating large divots in the 3D-printed parts, contact with MCP and PIP knuckles could be avoided. Flexible rubber cord was used to fasten the human finger to the mechanical exoskeleton, and free-rolling wheels were added to each stage at contact points to reduce friction. The resulting motion of the system is shown in Fig. 11.

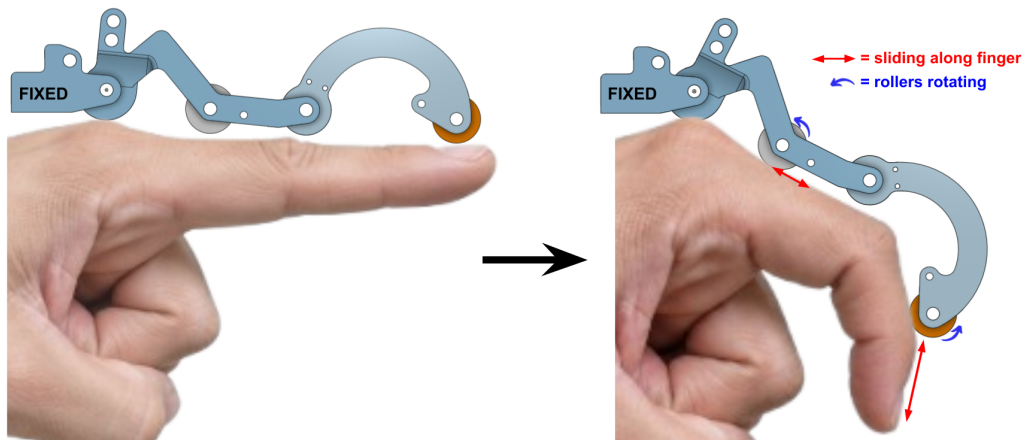
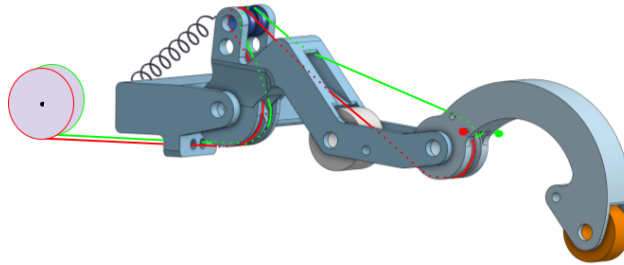


Fig. 11. Exoskeleton sliding against human finger during joint rotation

Due to friction in the system, finger force output was highly limited. As stated previously, this was worsened by the rubber band used for MCP extension, which further increased the resistance faced by actuators during MCP flexion. To eliminate this rubber band, a modified cable routing system was developed, removing the need for any springs to return the fingers to extension. This routing system added a single cable with each end coming from the top side of the servo pulleys and pulling through a loop on the top side of the MCP pulley. This cable served to reverse the initial bottom cables' behavior, pulling back up on the MCP joint when the bottom cables release.

This cable could be tightened to ensure all cables were properly taut and no force was wasted in flexibility. This change more than tripled the force output and made motion much more rigid and accurate. The routing system improvement is shown in Fig. 12.

**Initial design with spring-powered passive extension:**



**New cable-routing system for active extension:**

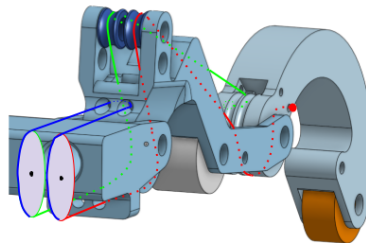


Fig. 12. New cable routing system to replace spring for MCP extension

Once a single finger was working adequately, the system was scaled to include three fingers with independent MCP and PIP actuation: the index, middle, and ring fingers. The pinky finger is not fully necessary for most ADLs, so to reduce complexity, it was simply fastened to the ring finger and restricted to move with it. To store all actuators compactly, three copies of the single-finger servo housing were mounted around the forearm, keeping all six finger actuators off the hand, with a 3D-printed fastening bracket to keep them together. This is shown in Fig. 13.

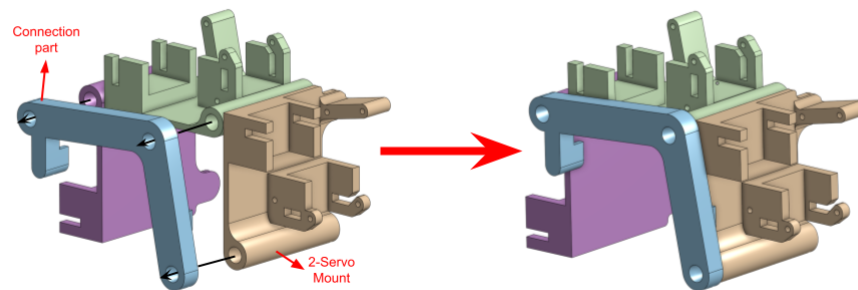


Fig. 13. Three servo housings mounted around the forearm with a 3D-printed connecting bracket

Many ADLs require the thumb for grip and dexterity. For this reason, an actuation system for the thumb was also developed. To reduce complexity and weight, the thumb was only given a single degree of freedom, avoiding the need for an intricate cable-routing arrangement. Instead, the thumb actuator was mounted much closer to the

hand on the wrist brace, while still remaining off the hand itself, and a four-bar linkage was used to transfer power from the servo arm to a ring fastened to the thumb (Figure 14)



Fig. 14. Image of thumb linkage

2) *Code and Electronics*: An ESP32 development board was used to run software receiving Bluetooth signals and moving the corresponding servos. A 5-volt battery pack was used to provide power to the system, which included the ESP32 and seven servos. As shown in Fig. 15, the battery pack feeds a common 5 V rail that powers the ESP32 through its VIN pin and each servo directly, with all components sharing a common ground. Each servo receives its control signal from a dedicated GPIO pin, and the mirroring switch described in Section II-A connects an additional GPIO input to ground. The firmware checks this input before applying each incoming command (Listing 1): while the mirroring switch is on, the input reads low and each command is applied to the servos; when the switch is turned off, the input is pulled high by the ESP32's internal pull-up resistor, joint commands received from the glove are discarded, and every servo maintains its last commanded position until the switch is turned on again. This allows the user to shape the impaired hand with the glove, hold it around an object, and then use the unimpaired hand freely for an asymmetric task. 100  $\mu$ F capacitors was placed across the supply of each servo to handle sudden voltage fluctuations caused by servo current spikes and prevent damage to components.

For testing purposes, a Python program was written providing a graphical user interface (GUI) to send commanded joint angles in the form of Bluetooth data to the ESP32, which converts them into servo positions. The GUI is an interactive Matplotlib [16] plot of the finger in which each joint can be dragged with the mouse; as a joint is moved, the corresponding actuation command is sent to the exoskeleton. This provides immediate feedback on how the system responds to each command (Fig. 16) and enabled control and testing of the exoskeleton.

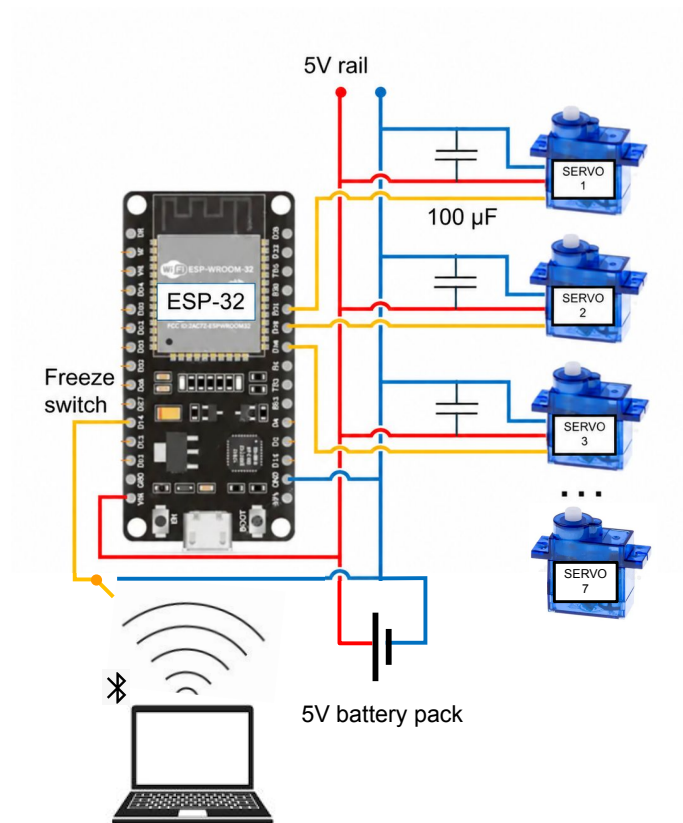


Fig. 15. Exoskeleton wiring diagram: ESP32, seven servos with 100 µF decoupling capacitors, mirroring switch, and 5 V battery pack, with Bluetooth commands from a host computer

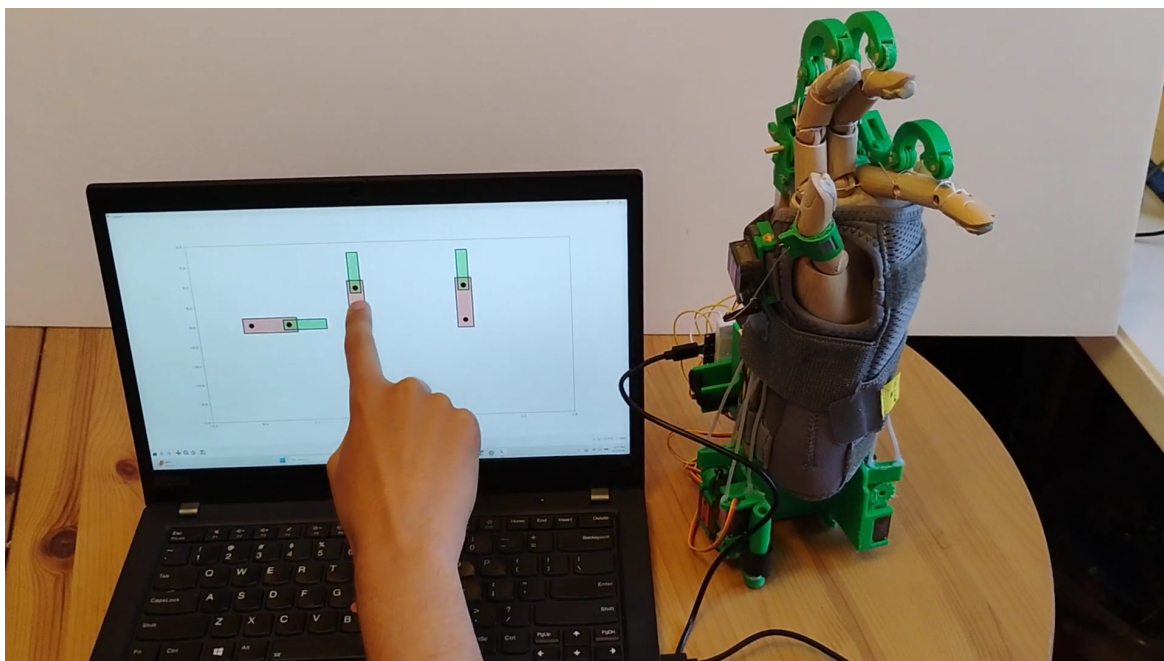


Fig. 16. Interactive Matplotlib GUI used to command the exoskeleton by dragging each joint

3) *Joint-to-Servo Mapping*: Because of the differential routing in Fig. 8, neither servo controls a single joint; each commanded pair of MCP and PIP angles must be converted into a pair of servo angles (Fig. 17). When a joint rotates by an angle  $\theta$ , a cable wrapped around its pulley of radius  $r$  is displaced by the arc length  $r\theta$ . For commanded MCP and PIP angles  $\theta_M$  and  $\theta_P$ , the required cable displacements at the two joints are

$$\Delta_M = r_M \theta_M, \quad \Delta_P = r_P \theta_P, \quad (1)$$

where  $r_M$  and  $r_P$  are the radii of the MCP and PIP pulleys. Both cables wrap the MCP pulley in the same direction, so each must be displaced by  $\Delta_M$ ; they wrap the PIP pulley in opposite directions, so  $\Delta_P$  adds to one cable and subtracts from the other. Each servo winds its cable on a pulley of radius  $r_S$ , so the servo angles are

$$\phi_1 = s_1 \frac{r_M \theta_M + r_P \theta_P}{r_S} + \phi_1^0, \quad (2)$$

$$\phi_2 = s_2 \frac{r_M \theta_M - r_P \theta_P}{r_S} + \phi_2^0, \quad (3)$$

where  $\phi_1^0$  and  $\phi_2^0$  are the servo angles at which both joints are fully extended, and  $s_1, s_2 \in \{-1, +1\}$  account for the mounting direction of each servo. Inverting (2)–(3) shows how the differential decouples the joints:

$$\theta_M = \frac{r_S}{2r_M} \left( \frac{\delta\phi_1}{s_1} + \frac{\delta\phi_2}{s_2} \right), \quad \theta_P = \frac{r_S}{2r_P} \left( \frac{\delta\phi_1}{s_1} - \frac{\delta\phi_2}{s_2} \right), \quad (4)$$

with  $\delta\phi_i = \phi_i - \phi_i^0$ . Rotating both servos so that both cables are pulled equally flexes only the MCP joint, while pulling one cable and releasing the other by the same amount rotates only the PIP joint, matching the table in Fig. 8. Since only ratios of arc lengths appear, all angles can be expressed in degrees.

The single-degree-of-freedom thumb is driven through the four-bar linkage, whose servo angle is interpolated linearly between its fully extended and fully flexed positions:

$$\phi_T = \phi_T^{\text{ext}} + \frac{\theta_T}{90^\circ} \left( \phi_T^{\text{flex}} - \phi_T^{\text{ext}} \right), \quad (5)$$

with  $\phi_T^{\text{ext}} = 180^\circ$  and  $\phi_T^{\text{flex}} = 45^\circ$  for the prototype. The measured pulley radii of the prototype are  $r_M = 0.25$  in (6.4 mm),  $r_P = 0.22$  in (5.6 mm), and  $r_S = 0.30$  in (7.6 mm). Listing 1 shows the corresponding firmware for the three fingers and the thumb.

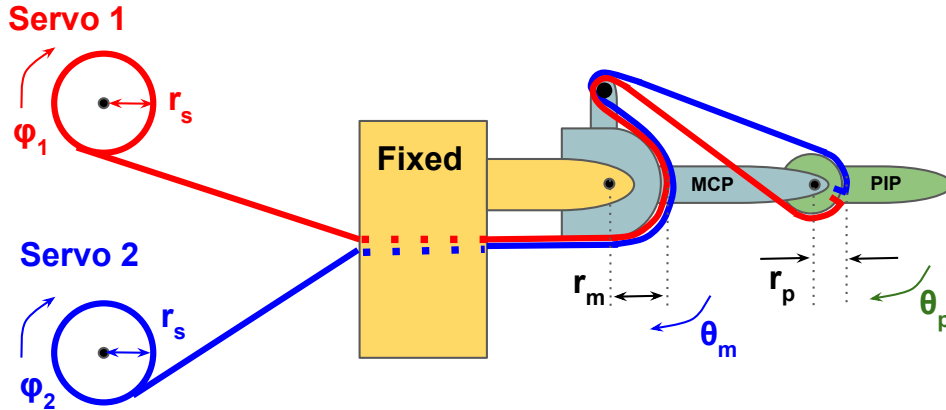


Fig. 17. Schematic of the differential cable routing used to derive the joint-to-servo mapping. Both cables wrap the MCP pulley in the same direction and the PIP pulley in opposite directions, so cable 1 is displaced by  $r_M \theta_M + r_P \theta_P$  and cable 2 by  $r_M \theta_M - r_P \theta_P$ .

Listing 1. Excerpt from the ESP32 firmware (Arduino C++) that converts the commanded MCP and PIP angles received over Bluetooth into servo positions

```

... // include libraries

// constants (pulley radii in inches)
const double servo_rad = 0.30; // r_S
const double mcp_rad = 0.25;   // r_M
const double pip_rad = 0.22;   // r_P
const int pos1_reverse = -1; // Constant to reverse servo rotation direction
const int pos2_reverse = 1;
const int thumb_out = 180; // Servo angle for thumb fully extended (angle = 0)
const int thumb_in = 45;   // Servo angle for thumb fully flexed (angle = 90)

... // Pin initializations, setup, bluetooth serial begin, etc.

// Called repeatedly:
void loop() {
    if (bluetooth.available()) {
        message += (char) bluetooth.read(); // read and append message fragments as they are received
        ↪ through bluetooth
    }

    if (message.charAt(message.length() - 1) == ']') { // end token received, message complete

        if (digitalRead(4) == LOW) { // mirroring switch on

            ... // Message is parsed and substrings stored in the 'subs' array

            // Looping through 3 main fingers (6 servos):
            for (int i = 0; i < 6; i += 2) {
                int n1 = subs[i].toInt(); // commanded MCP angle
                int n2 = subs[i+1].toInt(); // commanded PIP angle

                // Calculations for servo positions given desired joint angles
                double mcp_string_pull = mcp_rad * n1;
                double pip_string_pull = pip_rad * n2;
                ↪ pos1_zero;
                int pos1 = (int) lround((mcp_string_pull + pip_string_pull) / servo_rad) * pos1_reverse +
                ↪ pos2_zero;
                int pos2 = (int) lround((mcp_string_pull - pip_string_pull) / servo_rad) * pos2_reverse +
                ↪ pos2_zero;

                servos[i]->write(pos1); // Send signal to servo 1
                servos[i+1]->write(pos2); // Send signal to servo 2
            }

            // Thumb handled separately:
            int thumb_angle = subs[6].toInt();
            int thumb_servo_pos = (int) ((thumb_angle / 90.0) * (thumb_in - thumb_out) + thumb_out); //
            ↪ linear interpolation
            thumb.write(thumb_servo_pos); // Send signal to thumb servo
        }

        message = ""; // reset message for next data reception
    }
}
...

```

---

### III. VALIDATION

This section validates the two subsystems independently. First, the measurement glove is evaluated by comparing its sensor output with the joint angles observed on the user's finger (Section III-B). Second, the exoskeleton is evaluated by comparing the joint motions it produces with the commanded joint angles (Section III-C). In both cases, the observed joint angles are obtained from video using the same computer vision pipeline, described next.

#### A. Measurement Method

To obtain quantitative measurements of the observed joint motion, video analysis was used. For each test, a camera captured a side view of the finger motion at 30 frames per second (FPS). Brightly colored paper markers

were placed on each joint (Fig. 18), and a computer vision pipeline written in Python detected the location of each marker in every frame and calculated the angle between adjacent finger segments. These joint angle values over time could then be plotted along with the expected values to measure both the accuracy of positioning and the time delay. Each test was repeated several times to confirm that the video-based measurements were consistent across trials.

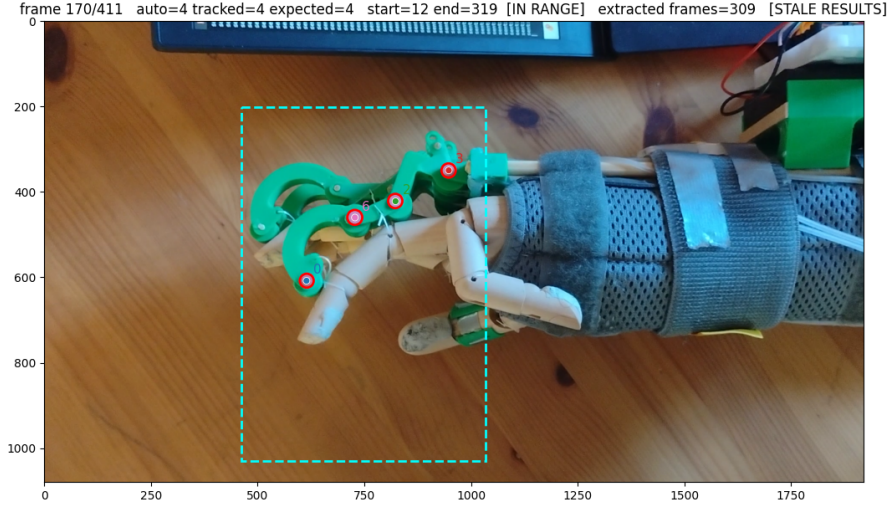


Fig. 18. Computer vision pipeline tracking the colored markers on the exoskeleton in a video frame

### B. Measurement Glove

Fig. 19 compares the encoder output of the measurement glove with the joint angle observed from video as the MCP joint of the index finger was repeatedly flexed and extended, and Table I lists the values at each peak and trough. The encoder tracked the observed angle with a mean absolute error of  $1.19^\circ$ , while the measured signal trailed the finger by 0.312 s on average, a consequence of the moving-average filter described in Section II-B2.

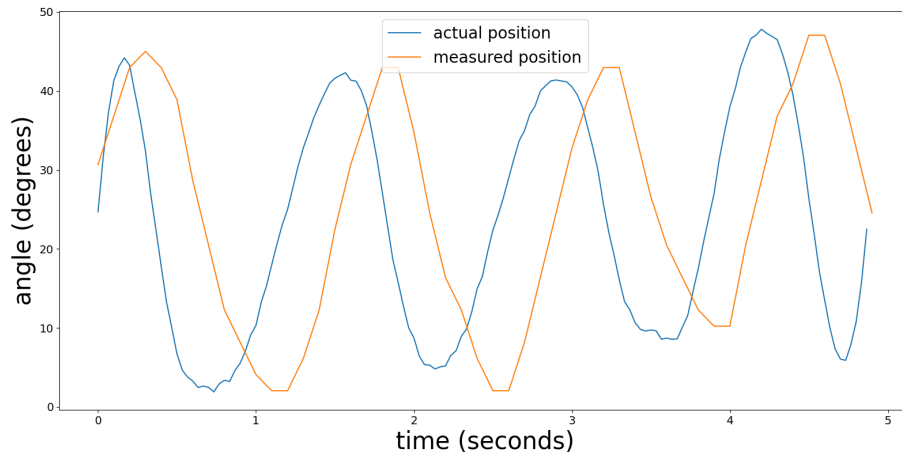


Fig. 19. Encoder output compared against the actual finger position. The encoder replicates the finger's position within 1.19 degrees on average and the measured signal trails the position by about 300 ms, a consequence of the moving-average filter applied to reduce jitter.

TABLE I  
COMPARISON OF ACTUAL AND MEASURED JOINT ANGLE AT EACH PEAK AND TROUGH (DATA FROM FIG. 19)

| Point               | Time (s) |          |       | Angle (°) |          |       |
|---------------------|----------|----------|-------|-----------|----------|-------|
|                     | Actual   | Measured | Lag   | Actual    | Measured | Error |
| Peak 1              | 0.167    | 0.294    | 0.127 | 43.85     | 45.00    | +1.15 |
| Trough 1            | 0.698    | 1.148    | 0.450 | 2.00      | 2.13     | +0.13 |
| Peak 2              | 1.575    | 1.852    | 0.277 | 42.48     | 43.00    | +0.52 |
| Trough 2            | 2.161    | 2.543    | 0.382 | 4.66      | 2.13     | -2.53 |
| Peak 3              | 2.920    | 3.237    | 0.317 | 41.87     | 43.03    | +1.16 |
| Trough 3            | 3.624    | 3.942    | 0.318 | 8.14      | 10.33    | +2.19 |
| Peak 4              | 4.228    | 4.541    | 0.313 | 47.81     | 47.13    | -0.68 |
| Mean                |          |          | 0.312 |           |          | +0.28 |
| Std. dev.           |          |          | 0.100 |           |          | 1.53  |
| Mean absolute error |          |          |       |           |          | 1.19  |

### C. Exoskeleton

The exoskeleton was evaluated on the wooden hand described in Section II-C. Two oscillation studies were performed to test independent actuation. In the first, the MCP joint was commanded to move linearly from 0° to 90° over 10 s while the PIP joint was commanded to oscillate between 0° and 90° with a period of 2.5 s (Fig. 20 and Table II). In the second, the roles were reversed: the PIP joint moved linearly while the MCP joint oscillated (Fig. 21 and Table III). In both studies, the oscillating joint continued to cycle while the other joint followed its own ramp, demonstrating that the two joints can be commanded independently. The observed motion lagged the command by only 0.072–0.140 s on average, but neither joint reached the full commanded 90°, peaking at about 50–59°, and the MCP range of motion decreased as the PIP joint flexed. Two factors limit the range of motion: physical obstruction inherent in actuating a limp hand through components mounted above the fingers, and flexibility in the system that prevents the cables from retracting the full expected distance. In the first study, the MCP joint moved most at low PIP angles and plateaued at about 54°. In the second, the MCP range of motion fell from about 50° (6–56°) in the first cycle to about 20° (30–50°) in the last, a reduction of about 60%, while the PIP joint followed its ramp with roughly linear motion.

TABLE II  
COMPARISON OF COMMANDED AND ACTUAL PIP JOINT ANGLE AT EACH TROUGH AND PEAK (DATA FROM FIG. 20)

| Point               | Time (s) |        |        | Angle (°) |        |        |
|---------------------|----------|--------|--------|-----------|--------|--------|
|                     | Expected | Actual | Lag    | Expected  | Actual | Error  |
| Trough 1            | 0        | 0.058  | +0.058 | 0         | -6.3   | -6.3   |
| Peak 1              | 1.25     | 1.449  | +0.199 | 90        | 58.8   | -31.2  |
| Trough 2            | 2.5      | 2.549  | +0.049 | 0         | -1.4   | -1.4   |
| Peak 2              | 3.75     | 3.957  | +0.207 | 90        | 55.9   | -34.1  |
| Trough 3            | 5        | 5.082  | +0.082 | 0         | -0.1   | -0.1   |
| Peak 3              | 6.25     | 6.473  | +0.223 | 90        | 56.5   | -33.5  |
| Trough 4            | 7.5      | 7.606  | +0.106 | 0         | 0.2    | +0.2   |
| Peak 4              | 8.75     | 8.948  | +0.198 | 90        | 55.5   | -34.5  |
| Mean                |          |        | +0.140 |           |        | -17.61 |
| Std. dev.           |          |        | 0.073  |           |        | 16.94  |
| Mean absolute error |          |        | 0.140  |           |        | 17.66  |

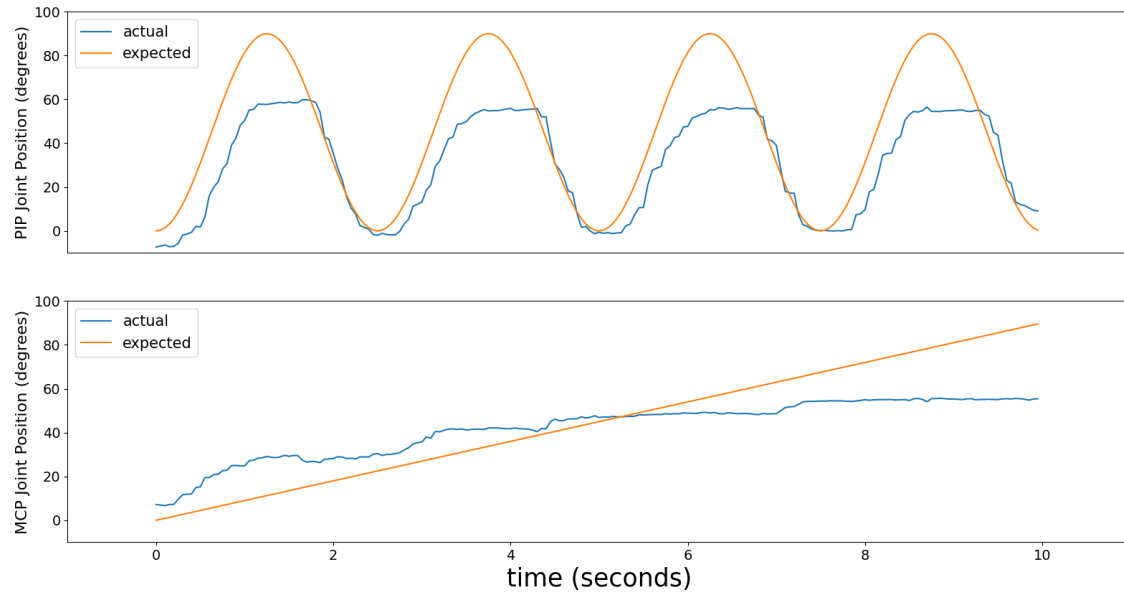


Fig. 20. Exoskeleton response with the PIP joint commanded to oscillate between  $0^\circ$  and  $90^\circ$  while the MCP joint ramps linearly from  $0^\circ$  to  $90^\circ$ . Top: PIP; bottom: MCP. “Expected” is the commanded angle and “actual” the angle observed from video.

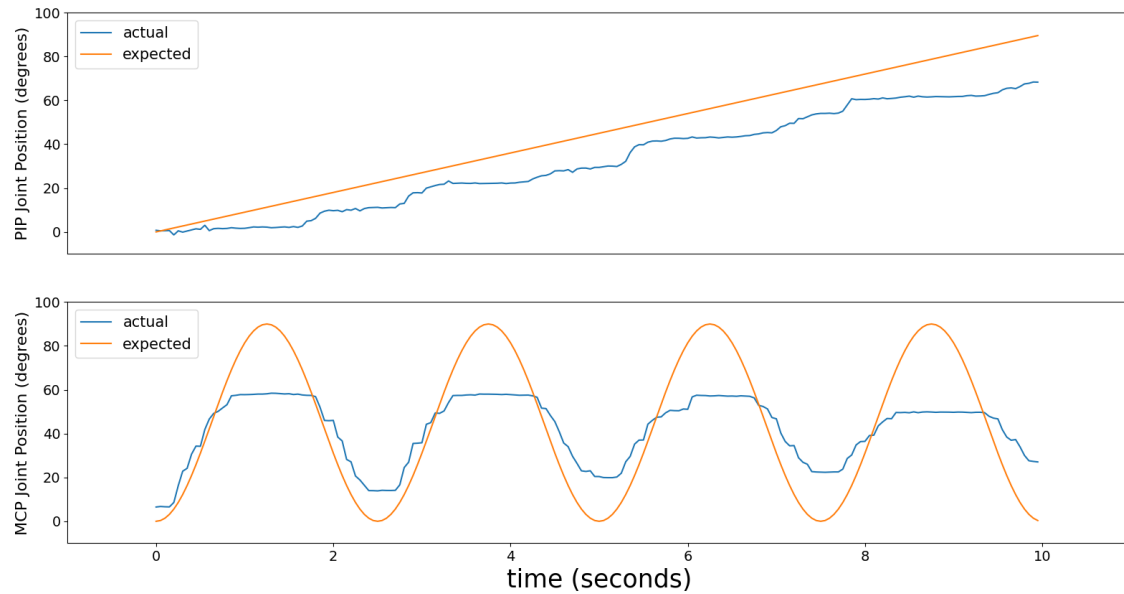


Fig. 21. Exoskeleton response with the MCP joint commanded to oscillate between  $0^\circ$  and  $90^\circ$  while the PIP joint ramps linearly from  $0^\circ$  to  $90^\circ$ . Top: PIP; bottom: MCP. “Expected” is the commanded angle and “actual” the angle observed from video.

A test was then conducted to determine the force delivered by the index finger during MCP and PIP flexion. The test used a two-phase procedure: the MCP joint was commanded from  $0^\circ$  to  $90^\circ$  over 3 s with the PIP joint held at  $0^\circ$ , after which the PIP joint was commanded from  $0^\circ$  to  $90^\circ$  over the next 3 s with the MCP joint held at  $90^\circ$ . Fig. 23 plots the measured force against the commanded angle of the joint moving in each phase. During MCP flexion, the force remained near zero until about  $30^\circ$  and then rose almost linearly to about 0.8 N. As the PIP joint then flexed, the force increased further to a maximum of about 1.2 N at  $25\text{--}35^\circ$  of PIP flexion before falling

TABLE III  
COMPARISON OF COMMANDED AND ACTUAL MCP JOINT ANGLE AT EACH PEAK AND TROUGH (DATA FROM FIG. 21)

| Point               | Time (s) |        |        | Angle (°) |        |        |
|---------------------|----------|--------|--------|-----------|--------|--------|
|                     | Expected | Actual | Lag    | Expected  | Actual | Error  |
| Trough 1            | 0        | 0.05   | +0.05  | 0.0       | 7.7    | +7.7   |
| Peak 1              | 1.25     | 1.312  | +0.062 | 90.0      | 58.8   | -31.2  |
| Trough 2            | 2.5      | 2.533  | +0.033 | 0.0       | 2.5    | +2.5   |
| Peak 2              | 3.75     | 3.811  | +0.061 | 90.0      | 57.9   | -32.1  |
| Trough 3            | 5        | 5.057  | +0.057 | 0.0       | 20.3   | +20.3  |
| Peak 3              | 6.25     | 6.408  | +0.158 | 90.0      | 57.6   | -32.4  |
| Trough 4            | 7.5      | 7.557  | +0.057 | 0.0       | 22.5   | +22.5  |
| Peak 4              | 8.75     | 8.851  | +0.101 | 90.0      | 50.0   | -40.0  |
| Mean                |          |        | +0.072 |           |        | -10.34 |
| Std. dev.           |          |        | 0.040  |           |        | 26.13  |
| Mean absolute error |          |        | 0.072  |           |        | 23.59  |

to about 0.7 N at full flexion. This decrease is attributed to flexibility in the system, which reduces cable tension as the PIP joint approaches full flexion. For this test, the finger pressed on a digital scale, and the scale display, in grams, was recorded on video and read automatically by a digit-recognition script (Fig. 22) before being converted to newtons.

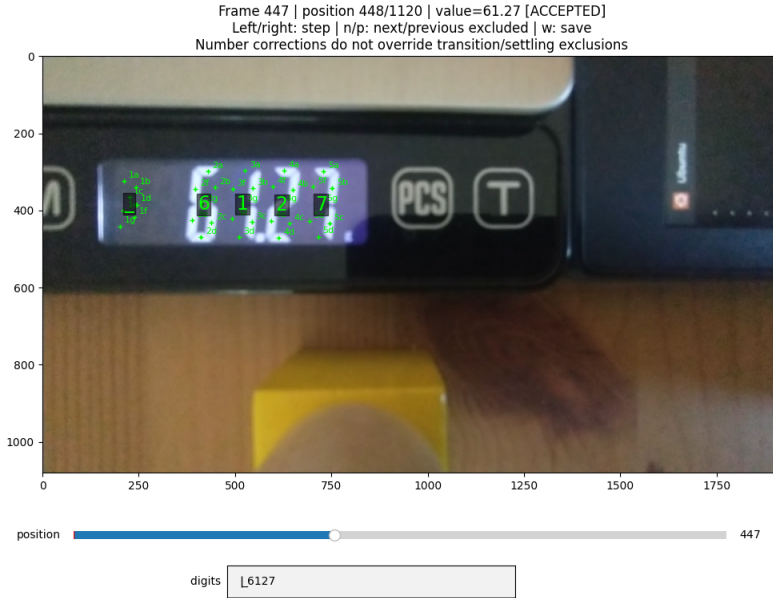


Fig. 22. Digit-recognition script reading the digital scale display (in grams) from a video frame during force measurement

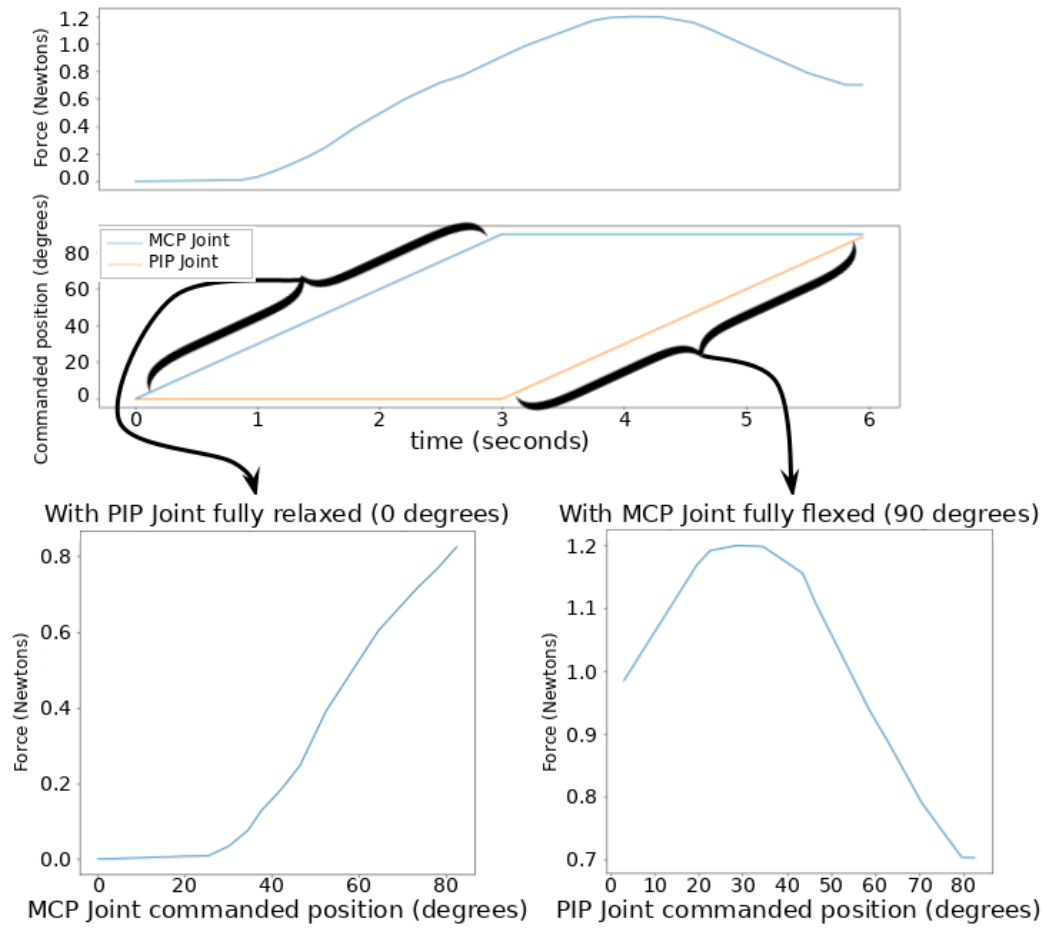
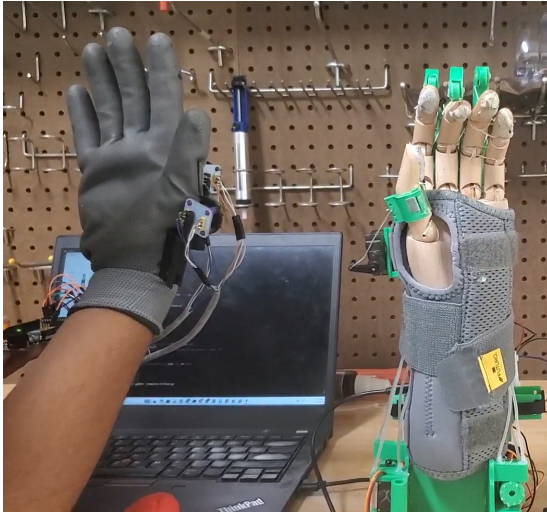


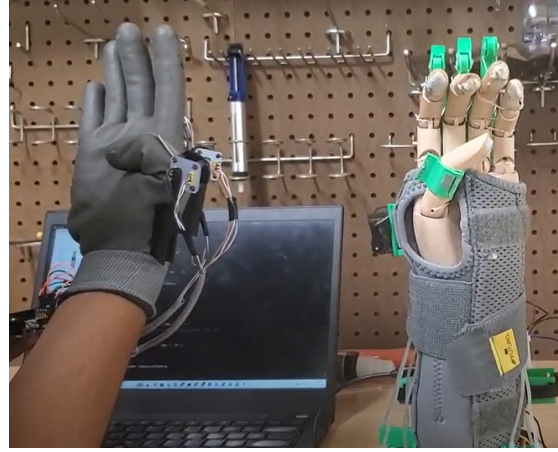
Fig. 23. Measured force evolution during MCP and PIP contraction. PIP contraction begins once MCP contraction is complete. Bottom left and right subplots show relationship between MCP and PIP joints and force output.

#### D. Integrated System

Finally, the complete mirroring system was tested end to end, with the measurement glove worn on a human hand driving the exoskeleton mounted on an off-the-shelf wooden hand. Fig. 24 shows frames from a video of the integrated system as it mirrors several hand poses.



(a) Open hand



(b) Thumb flexion



(c) MCP/PIP actuation with the thumb forming a ring



(d) Closed fist

Fig. 24. The integrated system mirroring hand poses from the glove (left in each frame) onto the exoskeleton-driven wooden hand (right).

Fig. 25 shows the MCP angle of the index finger on both hands over a 30 s trial: the exoskeleton-driven finger reproduces each flexion and extension of the gloved finger, trailing it by a visible lag that is consistent with the glove's moving-average filter. To quantify agreement across all sensed joints, the gloved hand was held at target angles from  $15^\circ$  to  $90^\circ$  and the corresponding angles of the wooden hand were measured (Fig. 26). A linear fit over the six joints gives  $R^2 = 0.782$ . Agreement is closest for the index and ring MCP joints and weakest for the ring PIP joint, which consistently undershot the glove, and the middle MCP joint, which overshot it.

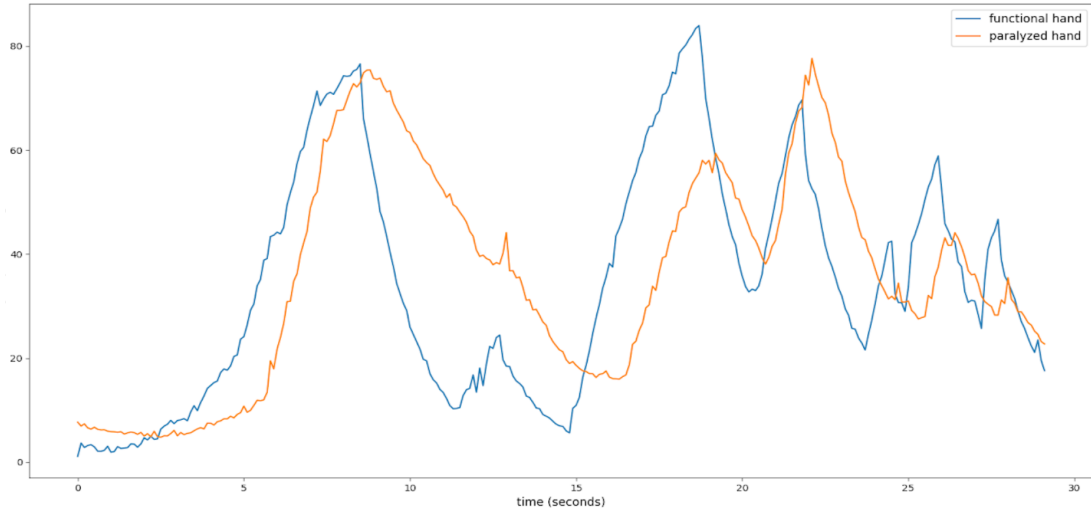


Fig. 25. Index-finger MCP angle of the gloved (functional) hand and the exoskeleton-driven (paralyzed) hand over time.

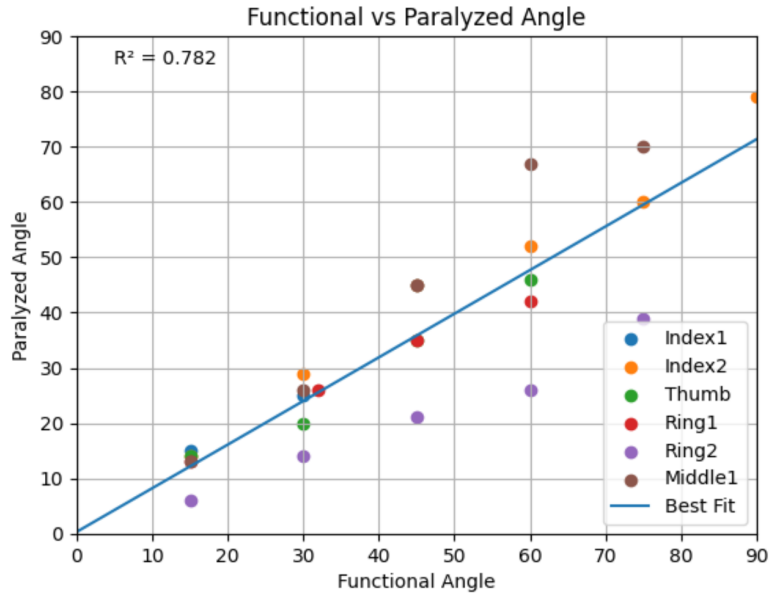


Fig. 26. Observed paralyzed-hand angle versus functional-hand angle for the six sensed joints (1 = MCP, 2 = PIP), with linear best fit ( $R^2 = 0.782$ ).

#### IV. CONCLUSION

The prototype developed demonstrates the feasibility of a wearable exoskeletal system capable of measuring and actuating both the MCP and PIP finger joints independently while also maintaining all of the actuators off the hand and on the arm.

Throughout the development process of the sensing glove, several mechanical and sensory challenges were addressed through prototype iterations. The initial limitation of the range of motion with the hinge was solved with the addition of a bracket that distances it from the finger. A cylindrical slider mechanism was also developed to accommodate the translation of each segment while maintaining the rotational motion. The sensing system was subsequently expanded from a single finger to a full hand encoding glove incorporating six magnetic encoders. Each encoder was aligned to either the MCP or PIP joint. Because the encoders shared fixed I<sup>2</sup>C addresses, software-defined I<sup>2</sup>C buses were implemented instead of adding a multiplexer to save crucial space on the hand. These buses

were created through bit-banging on GPIO pins. This allowed for all six of the encoders to relay data simultaneously. A remote testing system was implemented through a Discord API relay allowing for encoded data to be sent from encoders to actuators while cities apart.

Regarding the actuation system, a differential cable routing system was implemented to achieve 7 degrees of freedom across three fingers and the thumb (two per finger, one for the thumb). The system initially suffered from low force output, inaccurate joint angles and a lack of compactness, but extensive testing and design modifications significantly improved performance. The current design features two actuators each powering 4 cables such that the release of one pair is coupled to the retraction of the other pair. These 4 cables pass over pulleys in specific configurations that allow for active actuation for both MCP and PIP joints in both flexion and extension.

Despite these results, some limitations still remain in the current system. The prototype was unable to meet the 100 ms latency target, primarily due to the smoothing filter. The filter averages multiple data points to produce a more jitter-free output, allowing stable data to be transmitted to the actuators. Although it solves the issue of stability, it introduces latency into the system. Future prototypes could investigate alternative approaches that provide noise suppression while also reducing latency. Additionally, the encoding glove does not capture all degrees of freedom for each finger. Due to spatial constraints at the top of the hand, adding 12 encoders, 2 for each finger (MCP and PIP joints), was not possible using our current system. Lateral finger movements were also not captured. Additional sensors on the sides of each finger would be necessary to detect the width of the hand. In general, the prototype establishes the feasibility of measuring and actuating both joints on the finger, while also reducing bulkiness by relocating actuation hardware off the hand itself. Future development could entail completely new measuring systems that would be able to accurately sense all 12 joints as well as lateral movement and improvements to the actuation system allowing a higher force output.

#### ACKNOWLEDGMENT

The two authors divided the development of the system between its two halves. V. Iyengar developed the measurement glove and its data-acquisition software. L. Sitaraman developed the exoskeleton and its control software, carried out the CAD design for both the measurement glove and the exoskeleton, and developed the software that integrates the two into a complete mirroring system. Both authors contributed equally to the writing of the manuscript. This work is part of an ongoing research effort. An earlier version of the system was presented at the Greater San Diego Science and Engineering Fair (GSDSEF) in March 2026, where it won First Award and received special awards from Kaiser Permanente and the California Nurses Association.

#### REFERENCES

- [1] T. Triwiyanto, S. Luthfiyah, I. P. A. Pawana, A. A. Ahmed, and A. Andrian, "Bilateral mode exoskeleton for hand rehabilitation with wireless control using 3D printing technology based on IMU sensor," *HardwareX*, vol. 14, 2023, art. no. e00432, doi: 10.1016/j.ohx.2023.e00432.
- [2] H. D. Bryantono, C.-Y. Su, J.-K. Huang, T.-W. Xin, and S.-C. Tseng, "Soft gripper gloves with mirroring system design for hand rehabilitation," *Engineering Proceedings*, vol. 103, 2025, art. no. 29, doi: 10.3390/engproc2025103029.
- [3] C. Liu, J. Lu, H. Yang, and K. Guo, "Current state of robotics in hand rehabilitation after stroke: A systematic review," *Applied Sciences*, vol. 12, no. 9, 2022, art. no. 4540, doi: 10.3390/app12094540.
- [4] M. Sandison, K. Phan, R. Casas, L. Nguyen, M. Lum, M. Pergami-Peries, and P. S. Lum, "HandMATE: Wearable robotic hand exoskeleton and integrated Android app for at home stroke rehabilitation," in *Proc. 42nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, 2020, pp. 4867–4872, doi: 10.1109/EMBC44109.2020.9175332.
- [5] V. Nazari, M. Pouladian, Y.-P. Zheng, and M. Alam, "A compact and lightweight rehabilitative exoskeleton to restore grasping functions for people with hand paralysis," *Sensors*, vol. 21, no. 20, 2021, art. no. 6900, doi: 10.3390/s21206900.
- [6] L. Zhao, F. Guo, F. Yang, C. Li, Y. Zhao, J. Qu, K. Li, Y. Liu, L. Wang, and L. Bu, "A novel underactuated exoskeleton rehabilitation glove for hand flexion and extension training," *Biomimetic Intelligence and Robotics*, vol. 5, no. 4, 2025, art. no. 100248, doi: 10.1016/j.birob.2025.100248.
- [7] J. Palacios, A. Deli-Ivanov, A. Chen, L. Winterbottom, D. M. Nilsen, J. Stein, and M. Ciocarlie, "Grasp force assistance via throttle-based wrist angle control on a robotic hand orthosis for C6–C7 spinal cord injury," arXiv:2402.08020, 2024.
- [8] Myomo Inc., "What is a MyoPro orthosis?" [Online]. Available: <https://myomo.com/what-is-a-myopro-orthosis/>
- [9] Broadened Horizons, "PowerGrip orthosis — powered grasp exoskeleton glove," Inclusive Inc. [Online]. Available: <https://inclusiveinc.org/products/powergrip>
- [10] Shanghai Siyi Intelligent Technology Co., Ltd., "Syrebo exoskeleton hand." [Online]. Available: <https://www.syrebo.com/rehabilitation-system-for-clinic/exoskeleton-hand>

- [11] MED-EL Medical Electronics, “Functional electrical stimulation (FES) for hand injuries — STIWELL.” [Online]. Available: <https://stiwell.medel.com/application-areas/fes-for-a-hand-injury>
- [12] “Open-source hinge design,” Onshape public document, accessed: Oct. 4, 2026. [Online]. Available: <https://cad.onshape.com/documents/858b448abbcbd73cbd124787/w/6b9f8d622a570a12cd88565e/e/6e8bdc995e46c82b3b7f4225>
- [13] *UM10204: I<sup>2</sup>C-bus Specification and User Manual*, NXP Semiconductors, Oct. 2021, rev. 7.0. [Online]. Available: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>
- [14] Microchip Technology Inc., “AN1488: Bit banging I<sup>2</sup>C on mid-range MCUs with the XC8 C compiler,” 2013. [Online]. Available: <https://ww1.microchip.com/downloads/en/AppNotes/01488A.pdf>
- [15] Discord Inc., “Discord developer documentation.” [Online]. Available: <https://discord.com/developers/docs>
- [16] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.