Makara Journal of Technology

Volume 27 | Issue 1 Article 4

4-30-2023

Restorative Hand Therapy Exercise using IoT-Based Flex Force Smart Glove

Yi Neng Chang
School of Electric and Electronic Engineering, Universiti Sains Malaysia, Penang 14300, Malaysia, chang.yineng@student.usm.my

Mohamad Khairi Ishak
School of Electric and Electronic Engineering, Universiti Sains Malaysia, Penang 14300, Malaysia, khairiishak@usm.my

Khalid Ammar
Department of Electrical Engineering, Ajman University, Ajman 346, United Arab Emirates, k.ammar@ajman.ac.ae

Recommended Citation
Chang, Yi Neng; Ishak, Mohamad Khairi; and Ammar, Khalid (2023) "Restorative Hand Therapy Exercise using IoT-Based Flex Force Smart Glove," *Makara Journal of Technology*: Vol. 27: Iss. 1, Article 4.
DOI: 10.7454/mst.v27i1.1540
Available at: https://scholarhub.ui.ac.id/mjt/vol27/iss1/4

Restorative Hand Therapy Exercise using IoT-Based Flex Force Smart Glove

Cover Page Footnote

We would like to thank Universiti Sains Malaysia for providing the resources and infrastructure and this work is partly supported by the Universiti Sains Malaysia RUI Grant (PELECT/8014049).

Makara J. Technol. 27/1 (2023), 25–32
doi: 10.7454/mst.v27i1.1540

Restorative Hand Therapy Exercise using IoT-Based Flex Force Smart Glove

Yi Neng Chang[1], Mohamad Khairi Ishak[1*], and Khalid Ammar[2]

1. School of Electric and Electronic Engineering, Universiti Sains Malaysia, Penang 14300, Malaysia
2. Department of Electrical Engineering, Ajman University, Ajman 346, United Arab Emirates

[*]*E-mail: khairiishak@usm.my*

Abstract

Stroke has long been a major health concern in Malaysia, with cerebrovascular disease accounting for one-third of all deaths. Patients who have had a stroke have varying degrees of movement impairment, which can be improved with physiotherapy. A smart glove system is proposed in this paper to measure and visualize patient finger flexes in real-time remotely and record these flexes for further analysis to monitor and assess patient physiotherapy sessions. The system is designed to help physiotherapists evaluate physiotherapy exercises of patients and provide detailed information regarding the rehabilitation process. This paper also includes performance analysis and data from physiotherapy sessions using the system. Overall, the flex force smart glove shows promise in aiding restorative hand therapy exercises.

Keywords: embedded system, hand therapy, internet of things, smart glove, stroke rehabilitation

1. Introduction

As of 2021, cerebrovascular disease remains to be the third highest cause of death among Malaysians, representing 8.0% of the principal cause of death in Malaysia [1]. The most common cerebrovascular disease is stroke, which occurs due to the lack of nutrient and oxygen supply to the brain [2], thus causing paralysis or loss of motor function of the body parts controlled by the damaged part in the brain. Stroke rehabilitation aims to combat this disease through physiotherapy by retraining the brain to control movements using other undamaged parts of the brain, thus regaining motor function [3].

Stroke is a medical condition where the blood supply to the brain is interrupted or a blood vessel bursts, which causes brain cell death due to the lack of nutrients and oxygen supplied by the blood flow [2]. Sarcopenia is prominent among stroke patients based on research on stroke patients [4]. The study highlighted the loss of skeletal muscle after stroke, which leads to disability. Disabilities among stroke patients are often attributed to brain lesions [5], but the current study focuses on inactivity and malnutrition. The article also mentioned the assessment of hand grip strength as a method to determine sarcopenia and confirms that rehabilitation in stroke patients should consider the prevention of muscle wasting.

A recent journal article has shown that stroke patients typically experience impaired finger movement, especially with finger idle time. The journal also stated that “rehabilitation after stroke should encourage the performance of functional tasks that involve movements at faster cadences and encourage more frequent movement of the digits with shorter periods of inactivity.” [6]. An experiment was conducted in another study to evaluate the effectiveness of rehabilitation on hand and finger functions of stroke patients, which concluded that a training procedure can improve hand dexterity in patients who suffered stroke and have reduced movement repertoire [7].

A team of researchers in Canada conducted a study using rats as stroke rehabilitation targets to explore the required intensity of rehabilitation to recover motor functions [8]. Rodents were used due to their similar recovery biomarkers to humans. An algorithm was then developed using a predictive model based on the data collected from the recovery of rats to calculate the necessary rehabilitation intensity for the recovery of function based on the severity of stroke impairment. The obtained results displayed the significance of adequate rehabilitation. The previously mentioned studies have shown the prevalence of stroke in Malaysia as well as the effectiveness of stroke rehabilitation in recovering deteriorated motor function among stroke patients. The current work aims to propose a simple and effective way to assist in studies and recovery processes among numerous patients suffering from stroke-induced motor function impairment.

2. Related Works

Many studies have shown the feasibility of interactive rehabilitation using electronic devices. One article depicted the use of grip sensors in a wearable device and a musical computer game in therapy among stroke patients [9]. Another article uses a mixed reality tabletop system in the presence of patients with an interactive environment [10]. A customizable experience is possible with the usage of the experimental system, which allows different training programs. The system "was reported as highly usable, enjoyable, and motivating."

An article by Intel documented the usage of Cyclone® V SoC FPGA in a Flex Force Smart Glove (FFSG), which is also used in stroke rehabilitation [11]. The FFSG can integrate data from five sets of pressure and flex sensors and five inertial measurement units to compute the position and orientation of each finger and generate a 3D representation of the hand. Figure 1 shows the Flex Force Smart Glove used in stroke rehabilitation. Another journal also presented a similar rehabilitation system using solely IMU sensors, one for each joint for 16 units. In this case, the system is used to determine range of motion loss in patients with rheumatoid arthritis [12].

The aforementioned studies reveal that physiotherapy has been proven to help patients recover part of the motor function lost due to stroke. However, additional tools may be desired to obtain an in-depth overview of the recovery process of patients and assess the efficacy of physiotherapy exercise. This study aims to solve the presented system in this paper. The smart glove system proposed in this paper can quantitatively measure and present the finger flexes of a patient, which allows physiotherapists to analyze the degree of effectiveness of physiotherapy exercises in helping patients regain finger motor function. This system can also be used to monitor the finger usage of patients during exercises remotely and study the finger coverage of the exercises. The recent epidemic has witnessed lockdown enforced in places, which caused inconveniences among patients seeking nonemergency medical procedures, because patients are denied face-to-face treatment to reduce contact or medical resources diverted to highly urgent cases. The remote features in this system provide a means for medical personnel to monitor and guide patients in real-time without direct contact. In addition, multiple devices can access the system through the Internet, leveraging the big data trend to allow collaboration among researchers across the globe in real-time.

In this work, an Arduino Mega works in conjunction with a 2 GB RAM Raspberry Pi 4 Model B [13]. The Arduino Mega interfaces with the data collecting gloves and then sends the data to the Raspberry Pi 4 for processing, which displays the data through a graphical user interface (GUI) on its desktop and sends the data to a dedicated Android App through UDP using data packaged in JSON format, logging the data in a CSV file. A performance analysis is then performed on the system along with transmission between the devices.

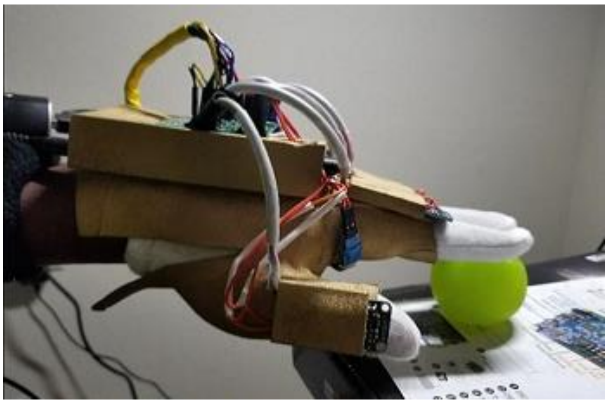


Figure 1. Flex Force Smart Glove [11]

3. System Design and Development

The development of the system comprises the following three general phases: hardware structure, software implementation, and IoT connectivity. The structure of the system includes the placement and usage of sensors, conditioning of raw sensor outputs, and physical connection between components. Software implementation in the system handles the development of a graphical user interface (GUI) and data processing, transaction, visualization, and storage. The IoT layer implements remote connectivity using Internet infrastructure and standard protocols.

System design. An overview of the entire system is shown in Figure 2. The sensors used in this work are the 2.2" Flexible Bend Sensors shown in Figure 3, which produce a change in resistance when bent. Five flex sensors are attached to the glove to observe the flexes of metacarpophalangeal joints and connected to five potential dividers (one for each sensor). The potential dividers are built with 47 k resistors in a pull-up position to convert the variable resistance obtained from the flex sensors into a voltage reading, which can be read by an Arduino Mega 2560. The Arduino Mega then concatenates the five analog readings into a single string, which is transmitted to a Raspberry Pi 4 through a USB cable using serial communication. Arduino plays the role of an ADC and multiplexer in the system.

Raspberry Pi acts as the main processor of the system. A GUI program written in Python is displayed on its desktop interface, which can be accessed through a VNC client [14] wirelessly or an external display connected to the Raspberry Pi through HDMI. In addition, an Android device connected to the local area network with the Raspberry Pi can access the reading output from the Raspberry Pi in real-time. Figure 4 shows the connection between devices in the system and the type of transmission established between them.

Implementation platform. Readings obtained by the Raspberry Pi from the sensors contain five integers ranging from 0 to 1023. The readings are then packaged into a single string separated with commas and sent to the Raspberry Pi. A simple handshake protocol is introduced between the Arduino and the Raspberry Pi due to the slower serial communication read speed at the Raspberry Pi compared to the Arduino. This condition fills up the buffer of the Raspberry Pi, which leads to substantial delays between the displayed and actual data. Therefore, the Arduino waits for a reply from the Raspberry Pi through the serial communication established through the USB before sending another set of data to avoid clogging up the communication.

The Raspberry Pi presents a GUI on the desktop within the Python Program as shown in Figure 5, which visualizes the data through sliders displaying the readings. Potential dividers are used; thus, the obtained readings will not be 0 when the flex sensors are straightened. Therefore, two offset buttons are added to the GUI to obtain the readings at rest and fully flexed as references for calibration of the readings from the Arduino. This strategy will remap the values to 0 when the fingers are relaxed and 100 when fully flexed, thus representing the readings as a flex percentage for each finger. A button for recording the readings in a CSV file is also available. These readings include the time the readings are taken since the record button is pressed and the calibrated readings of the flex bands. The inclusion of time taken for the readings allows for time-dependent analysis. The CSV file is saved on the desktop of the Raspberry Pi and is digitally stored in the microSD card containing the Raspberry Pi OS. The Raspberry Pi also broadcasts the processed readings to all Android applications installed with a dedicated application on the local network simultaneously.

Internet connectivity. The data output from the Raspberry Pi is encoded in JSON, where each finger is packaged as a name with the processed data as the values for the fingers. The Raspberry Pi then broadcasts the JSON package to all local devices using UDP by sending the datagram to 192.168.1.255 or a specific device with a known IP address. The datagram is sent to port 9033 of each IP address, where the Android application will be used for listening. The use of JSON instead of XML allows a small overhead because the same data packaged in JSON account for 87%–90% of the size of XML.

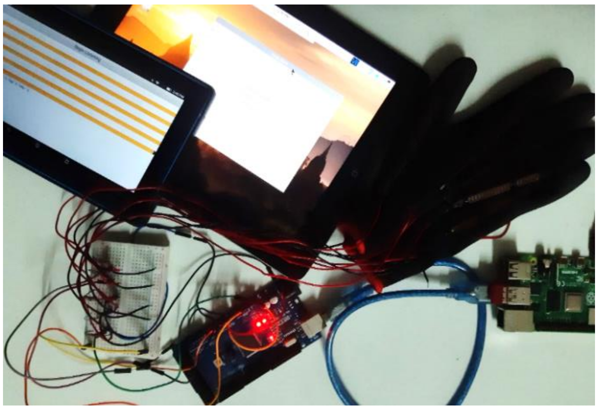


Figure 2. Devices Included in the System in Clockwise Order: (a) Data Glove, (b) Raspberry Pi 4 Model B, (c) Arduino Mega, (d) Potential Dividers, (e) Android Tablet with the Smart Glove Android Application, and (f) iPad with VNC Viewer

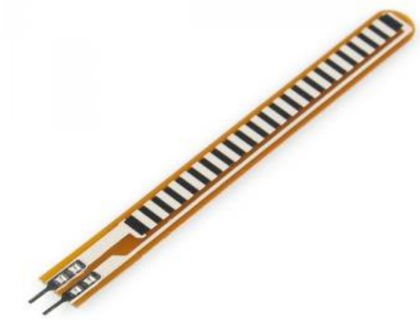


Figure 3. Flexible Bend Sensor

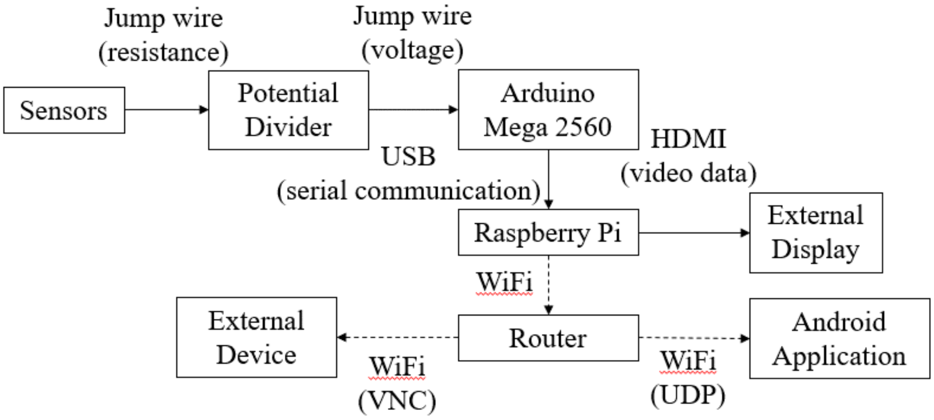


Figure 4. Topology of the System

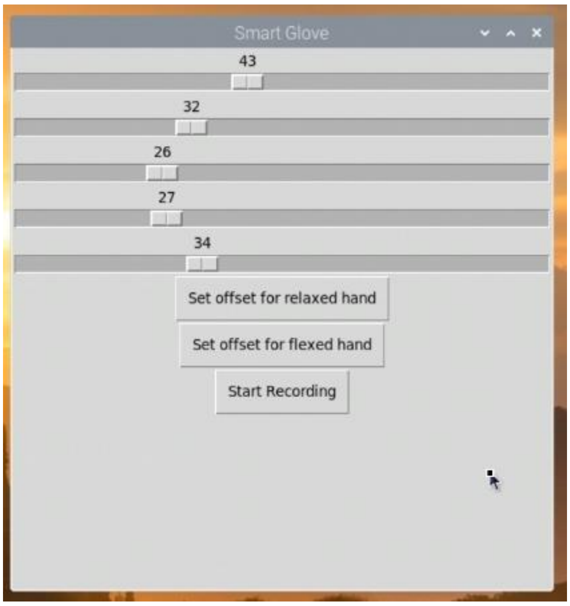
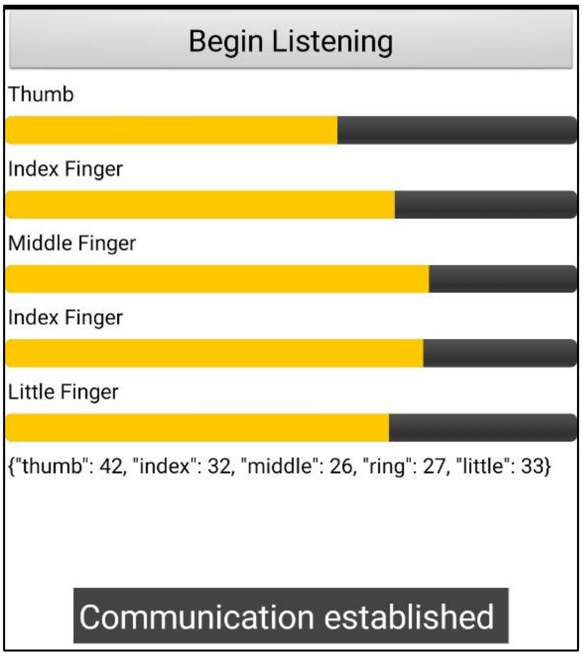


Figure 5. Graphical user Interface Displayed on the Raspberry Pi Desktop

Figure 6. FFSG Android Application



```
{"thumb": 42, "index": 32, "middle": 26, "ring": 27, "little": 33}
```

Communication established

Figure 7. Android Application

The Android application is made in MIT App inventor [15] (Figure 6) using a UDP extension [16] to access the datagram received on its IP address. Similar to the GUI, the application then decodes the JSON received into a dictionary and displays the readings on sliders, as shown in Figure 7. The visualization in the Android application is reversed; instead of effectively representing the state of the fingers, the length of the colored bar corresponds to the length of the straightened finger. In addition, alternative terminals can be easily developed to work with the system and access the data remotely by using a standard format for data packaging and protocol for transmission.

4. Results and Discussion

Results obtained from the implemented system can be split into three categories: Raspberry Pi considering the quantitative analysis of the performance, Android application performance based on the established communication between the Raspberry Pi and Android, and the application of the system in physiotherapy.

Raspberry Pi performance. Results obtained from the implemented system can be split into three categories: Raspberry Pi considering the quantitative analysis of the performance, Android application performance based on the established communication between the Raspberry Pi and Android, and the application of the system in physiotherapy. Performance analysis was performed on the Raspberry Pi, and the results are shown in Table 1. The data obtained from a bare-bones program, where the Raspberry Pi is set only to receive data and reply to the Arduino, are then fitted to Equation (1) to produce the graph shown in Figure 8.

$$Loop\ Time = \frac{Data\ Size}{Baud\ Rate} + Calculation\ Time \qquad (1)$$

Analysis of the fitted plot shows that the average data size transmitted is 221.275 bits. This finding accounts for the data output from Arduino and the reply from the Raspberry Pi. The attachment of start and short bits to each byte in the serial communication protocol results in 22 bytes. Comparatively, a generic data sent from an Arduino comprise five sets of three digits, separated by four commas in total, an additional end of line, or a newline character represented by “\n” appended at the end of the string. This phenomenon then adds up to 21 bytes of data sent from the Arduino, with the newline taking 2 bytes. The reply from the Raspberry Pi of a single “1” character takes another byte, thus summing to 22 bytes. A processing time independent of the baud rate at 3.141 ms per loop is also observed from the fitted plot. Implementation of the entire program with complete functions increases the loop time to 6.178 ms, which is a 20.64% or 1.057 ms increase from the bare-bones program. However, the full program only polls the serial input every 1 ms due to the usage of GUI; hence, an ext of delay of up to 1 ms may exist. Thus, the Raspberry Pi can process results at 161.86 Hz. A sample of a CSV file is visualized as a graph in Figure 9. The CSV file showed 918 data across 5.694 s, which yielded an average of 6.203 ms per loop, implying that writing to a CSV file does not impose noticeable performance reduction.

Android application performance. The UDP connection can transmit the JSON encoded readings through a router and broadcast the data to all devices on the local area network, which allows multiple devices to access the same data simultaneously, as shown in Figure 10. All three devices are connected to the same network as the Raspberry Pi, and the yellow bars represent the finger from thumb to little finger in top to bottom order, corresponding to the instantaneous flex matching of fingers to that of the gestures. All three devices show the same readings on their respective displays.

Table 1. Loop Time of Raspberry Pi

Baud rate	Average loop time (ms)
19,200	14.69121
38,400	8.886361819
57,600	6.863198
74,880	6.149198569
115,200	5.121108289

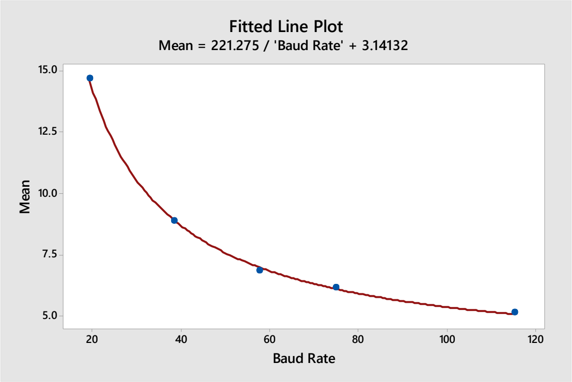


Figure 8. **Fitted Plot for Mean Loop Time and Baud Rate of Raspberry Pi**

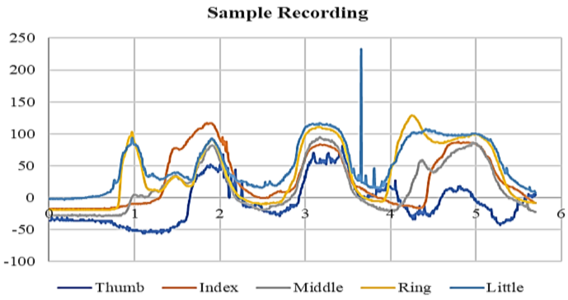


Figure 9. **Graph Generated with a CSV File**

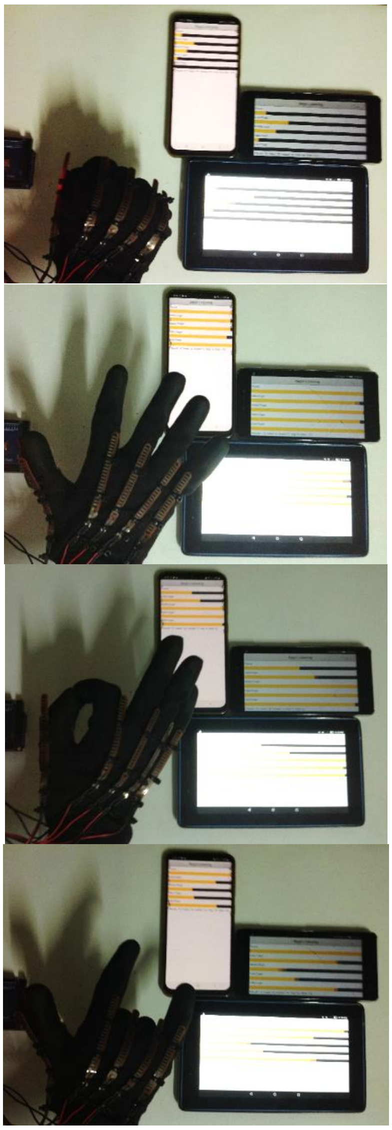


Figure 10. Data Transmission to Three Android Devices Simultaneously

Performance analysis on the Android application reveals that the application can receive data at an interval of 63.42 ms while the Raspberry Pi broadcasts on the network, which causes a significant drop in performance during data transmission to numerous devices. By contrast, the maximum rate at which the application can read datagrams was 8.96 ms, which is longer than the maximum rate at which the Raspberry Pi can send datagrams. This phenomenon causes noticeable delay at the receiving end on one-to-one communication, which can be mitigated by introducing an interval between packets sent from the Raspberry Pi that allows transmissions at 9.89 ms without noticeable latency (about 101.11 Hz). Notably, the maximum rate of transmission varies with device because a low-end Android device will require a low transmission rate to ensure a seamless data stream.

While the datagram can be broadcasted to devices on the same network, modifications must be made to transmit data due to the usage of routers and modems, where most devices are given a private IP address on the network due to Network Address Translation. Therefore, port forwarding must be performed on the router of the receiver, where the Android application is located, to realize transmission across the Internet. The Raspberry Pi can then send the datagram to the IP address of the router with the specified port. This setup is useful for a static receiver and dynamic transmitter, where the Raspberry Pi can be at different locations with various IP addresses. However, the Android device or alternative programs acting as the terminal of physiotherapists must be connected to the port-forwarded router and assigned a static IP address.

Another option is to implement a server with a public IP address as the target of the transmission, where remote terminals, such as the Android device will request data from the server. The server will then reply to the request with the latest received data from the Raspberry Pi, as shown in Figure 11. Alternatively, a VPN can also be used, which allows local area network transmission across the Internet through a simple setup on the Raspberry Pi and Android devices. The usage of a VPN server would also introduce an inherit security measure because VPN servers can be setup to login with credentials.

Physiotherapy application. A selection of physiotherapy exercises for the hand using a therapy ball (shown in Figure 12) [17] was performed and tested with the smart glove system. The finger flexes during the exercise were then recorded in CSV files and visualized with a scatterplot. The first exercise was power grip, where the therapy ball was squeezed and released using fingers and thumb. The collected data are then plotted in Figure 13.

The graph shows that all five fingers alternate between the maximum and minimum values with a period of around 1.5 s. A physiotherapist would be able to collect data on finger movement speed and flex range of patients to monitor and evaluate their recovery process. The second exercise performed was finger flexion (as shown in the scatterplot in Figure 14), which is similar to the power grip exercise but the thumb is not used. The ball is placed in the palm and pressed with the fingers.

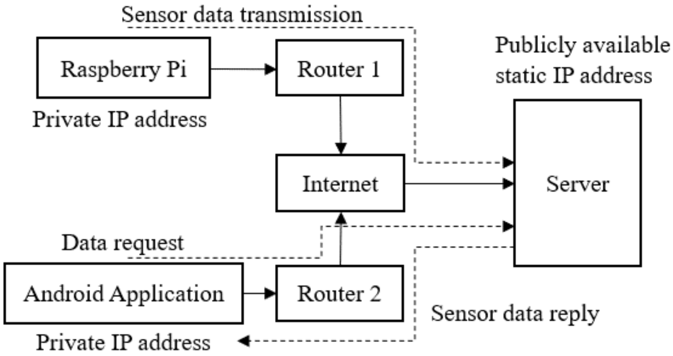


Figure 11. Topology of dedicated server

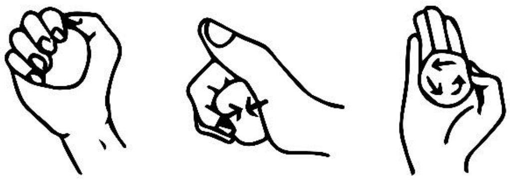


Figure 12. Three Finger Exercises Performed from Left to Right: Power Grip, Finger Flexion, and Thumb Roll

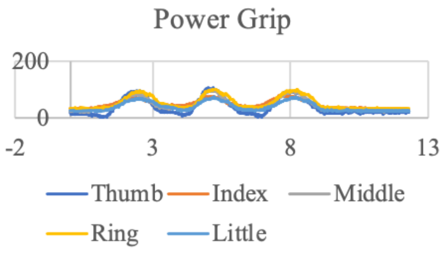


Figure 13. Scatterplot of Power Grip

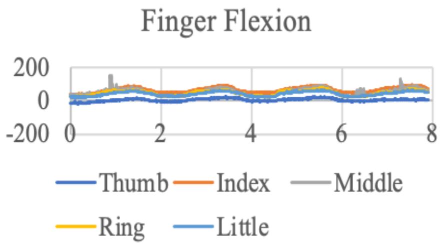


Figure 14. Scatterplot of Finger Flexion

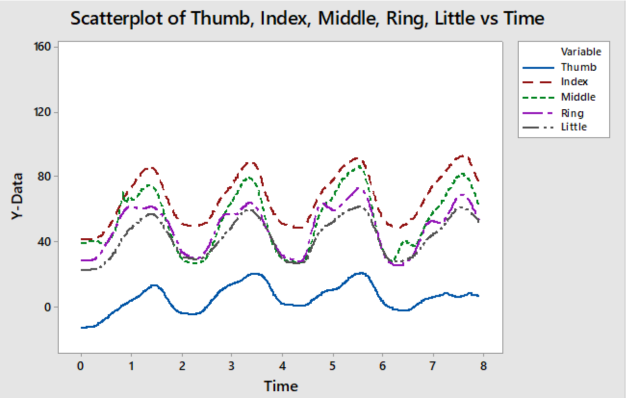


Figure 15. Smoothened Scatterplot of Finger Flexion

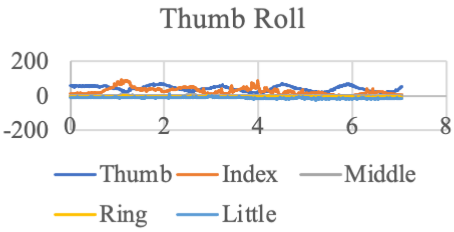


Figure 16. Scatterplot of Thumb Opposition

The graph shows a noticeable noise, including a few peaks reaching 150%. This noise can be easily reduced using various smoothing methods. A smoothed scatterplot using LOWESS with a step number of 2 and degree of 0.05 shown in Figure 15 markedly minimized noise and eliminated the extremities. This technique is useful while analyzing noisy data sets when a clear picture of finger movement is desired. In finger flexion exercise, the flex of the thumb is noticeably lower compared to the other fingers, which is consistent with the lack of thumb movement during the exercise.

A third exercise focusing on thumb movement, called the thumb roll, was also performed. In this exercise, the thumb is used to roll the ball in a circle on the palm, as shown in Figure 16. The graph reveals that the flex value of the thumb alternates between 60 and 20. However, the value for the index finger also fluctuates due to the close proximity between the thumb and index finger, inevitably flexing the index finger as well.

5. Conclusions

This work demonstrated the plausibility of a Raspberry Pi-centered data collection glove with IoT capabilities for physiotherapy among stroke patients. Readings from the flex sensors on the glove were read by the Arduino and transmitted to the Raspberry Pi, which displayed the processed data on its desktop GUI and transmitted them to Android devices in the form of JSON using UDP. The readings for all five fingers can also be recorded and tabulated in the form of CSV along with the time the readings were taken, allowing for time-sensitive analysis. The system has shown satisfactory results for intended usage, as illustrated by measured, displayed, and recorded data on flexes of five fingers at sampling rates of 161 and 101 Hz for CSV file recording and real-time remote monitoring, respectively. Data collected from physiotherapy exercises showed that the system can visualize finger flexes according to the performed exercise, which will help physiotherapists in analyzing and monitoring the recovery process of patients.

References

[1] Department of Statistics Malaysia, Statistics on Causes of Death, Malaysia, 2020, Department of Statistics Malaysia, Putrajaya, 2020. [*https://www.dosm.gov.my/portal-main/release-content/statistics-on-causes-of-death-malaysia-2020*].

[2] National Institute of Neurological Disorders and Stroke, Stroke Information Page, National Institute of Neurological Disorders and Stroke, 2019. [*https://www.ninds.nih.gov/Disorders/All-Disorders/Stroke-Information-Page*].

[3] National Institute of Neurological Disorders and Stroke, Post-stroke Rehabilitation Fact Sheet, National Institutes of Health, 2020. [*https://www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Fact-Sheets/Post-Stroke-Rehabilitation-Fact-Sheet*].

[4] N. Scherbakov, S. von Haehling, S.D. Anker, U. Dirnagl, W. Doehner, Int. J. Cardiol. 170/2 (2013) 89.

[5] G.J. del Zoppo, Stroke. 44/1 (2013) 263.

[6] H. Eschmann, M.E. Héroux, J.H. Cheetham, S. Potts, J. Diong, PloS One. 14/6 (2019) e0217969.

[7] R. Ranganathan, J. NeuroEng. Rehab. 14/1 (2017) 90.

[8] M.S. Jeffers, S. Karthikeyan, M. Gomez-Smith, S. Gasinzigwa, J. Achenbach, A. Feiten, D. Corbett, Neurorehab. Neural Re. 32/1 (2018) 73.

[9] Q. Sanders, V. Chan, R. Augsburger, S.C. Cramer, D.J. Reinkensmeyer, A.H. Do, IEEE Tran. Neur. Syst. Rehab. Eng. 28/6 (2020) 1363.

[10] C. Colomer, R. Llorens, E. Noé, M. Alcañiz, J. NeuroEng. Rehab. 13/1 (2016) 45.

[11] Intel, Flex Force Smart Glove: Use Intel® SoC FPGA to Measure Sensorimotor Data, 2018. [*https://software.intel.com/en-us/articles/flex-force-smart-glove-use-intel-soc-fpga-to-measure-sensorimotor-data*].

[12] J. Connolly, J. Condell, B. O'Flynn, J.T. Sanchez, P. Gardiner, IEEE Sens. J. 18 (2018) 1273.

[13] Raspberry Pi Trading Ltd., Raspberry Pi 4 Computer Model B Product Brief, 2021. [*https://static.raspberrypi.org/files/product-briefs/200521+Raspberry+Pi+4+Product+Brief.pdf*].

[14] RealVNC, VNC® Connect-Simple, Secure, Ready-to-Use Remote Access and Support Software, VNC® Connect, 2020.

[15] Massachusetts Institute of Technology, MIT App Inventor, Massachusetts Institute of Technology, 2020.

[16] U. Roboter, Ullis Roboter Seite/AI2 UDP, 2020. [*http://ullisroboterseite.de/android-AI2-UDP.html*].

[17] C. Maher, Hand Therapy Ball Exercises to Improve Fine Motor Skills, 2021. [*https://www.flintrehab.com/hand-exercise-ball-stroke-patients/*].