Article

A Wireless-controlled 3D printed Robotic Hand Motion System with Flex Force Sensors

Fazil Salman [1], Cui Yuanhui [1,2*], Zafar Imran [2], Liu Fenghua [3], Wang Lijian [3], Wu Weiping [1,*]

[1] Department of Electrical and Electronic Engineering, School of Mathematics, Computer Science and Engineering, City, University of London, Northampton Square, London, EC1V 0HB, United Kingdom; Salman.Fazil@city.ac.uk
[2] School of Information Science and Engineering, Dalian Polytechnic University, Dalian, 116034, China; Imran_zafar@live.com
[3] State Key Laboratory of Metal Matrix Composites, School of Materials Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China; fenghualiu@sjtu.edu.cn, 016050910037@sjtu.edu.cn
* Correspondence: cuiyh@dlpu.edu.com; Weiping.Wu@city.ac.uk

Received: date; Accepted: date; Published: date

Abstract: Hand gesture recognition is an emerging field of technology in robotics and human-computer interaction. It has tremendous applications in daily life activities and intelligent workplaces. In this study, a system which could help people to work and operate without directly using hands or contacting by hands, is proposed and demonstrated. This system composed of a glove with flexible force sensors and a 3D printed robotic forearm. The user wearing the glove could control the action of the 3D printed robotic forearm. The 3D printed forearm simultaneously acted following the motion of the glove. The 3D printed forearm was composed of 46 individual parts that were printed with white biodegradable polylactic acid (PLA). Electronic components in the system are five flex sensors, a master Arduino Nano, a slave Arduino Nano, a wireless NRF24L01 transmitter module banding on the glove, a second wireless NRF24L01 receiver module in the forearm and five motors. The five flex sensors on the fingers of the glove detected and collected the signals reflecting the movements of the hands. The Arduino Nano processed the signals from the flex sensors and sent them through the wireless transmitter module to the slave Arduino Nano. In order to control the action of the robotic forearm, it was embedded with a slave Arduino Nano as a control kernel, a wireless NRF24L01 receiver module and five actuators. The slave Arduino Nano received and processed the signals through the wireless receiver module. After that, the signals were sent to the actuators- servo motors. The fingers' action in the robotics arm was executed with the actuators. After carefully testing the system, the robotic arm followed the action correctly with a maximum 0.133 ms time delay all the time. This system could be really useful for the users who work in dangerous conditions, hazardous environment or require remote operation for safety reasons.

Keywords: Arduino Nano; Flex force sensors; NRF24L01 module; hand motion; 3D printed robotic arm;

1. Introduction

People can face high levels of risks and even life-threatening situations while doing dangerous works such as landmine demining or clearance in the military, driving hazardous vehicles in transportation, exploring the outer space, and working with high temperature melting metal in industrial manufacturing, etc. Some of these hazards can be deadly or life threatening [1]. It is vital and essential to create systems or devices as a substitution for human workers to do these works safely. Because the hand is uniquely important to daily life activities, hand function has been given increasingly concentrated and widely concerned during the past few decades [2-5].

Research publications on hand functions can be classified into hand gesture recognition and hand motion controls. For the hand gesture, some researches were about artificial muscle based human-machine interfaces [6-8]. Different technologies such as electromyography (EMG) [9], computer vision [10] and accelerometers [9, 11] have been used for the hand movements driven by the muscle contractions of the forearm. A systematic review with 148 articles was undertaken to identify the characteristics of touchless/in-air hand gestures used in interaction interfaces [12]. Some other researches were on the latest material technologies which were adopted to obtain hand motion by tactile sensors. Wearable soft artificial skin for hand motion detection with an array of embedded soft strain sensors was studied by measuring joint motions of fingers [13]. A newly-developed wearable hand motion capture and measurement system with multiple sensors and ultrasound imaging were introduced [14]. Besides, a user interface design for a human-hand simulation system was studied in a virtual environment that produces ground truth data and provides visualization support for experiments on computer vision-based hand pose estimation and tracking [15].

For the hand motion controls, a new method that demonstrated how to capture 3D arm motions with a stereo vision system and transfer to the robotic avatar with real-time in a meeting was represented [16]. A stylus-based haptic device was presented not only in 3D dynamic models, but also with accurate measurements [17]. The 3D arms mentioned above were virtual status in the computer made by specific software. On the topic of solid robotic arms, it was easy to find many publications in the web of science or google scholar. But these articles usually focused on how to design and analysis structures in the laboratory [18, 19]. Therefore, a system which could substitute people to do something in a real-time with the guide of the peoples' hand gesture was becoming the goal of this study. The initial plan was to create a fully functional humanoid robot. But this ambitious plan was abandoned quickly, because of the tedious work with an expansive cost. A robotic forearm was designed instead of the whole human-like robot body. Therefore, the hand motion capture system was selected to carry out the research.

Nowadays, it is easy to get a toy car controlled by a joystick, but a few advanced models use wireless hand gesture control mode. These toy cars could usually do functions of moving forward or backward, turning left or right. This enlightened us and the system came true with a low-cost and easy to customize prototype. For the system proposed in this article, a 3D printed forearm was designed to substitute the human hand to do dangerous tasks remotely. The 3D printed robotic arm was made with biodegradable material polylactic acid (PLA) and its fingers were electronically controlled by five flex force sensors. Flex sensors with flexible performance could measure the angle displacement, pressure and stress, have many applications in wearable electronics [20] and health monitoring [16, 21]. The system had two main hardware parts: glove with sensors and robotic forearm with motors. The flex sensors were used to measure the angle of the fingers and they were banded on a glove. The glove was made as a wireless controller to control the 3D printed robotic hands' actions.

In this paper, the system architecture was shown with the 3D printed arm and assembly process, the program of the control procedure in section 2, the measurement tests and results were represented in reliability and repeatability, sensitivity and time delay in hand motions in section 3. The discussion, suggestions and conclusions were in sections 4 and 5. Each stage of the development of the arm would be shown in this article with the results. The prototype is a promising solution for hazardous works that need to be dealt with by humans, which can be life threatening. The robotic arm with sensors can also help health carers to fight against disease, disabled people or elder people in their daily lives.

2. Proposed System

2.1. System Architecture and hardware Components

The proposed system should meet some requirements, such as the components of the system had to be carefully chosen, so that it has low cost with better accuracy to make the system affordable and reliable. The system should also have the capability of integrating multiple sensors. Thus, considering the above requirements and the comfort zone of visual challenges, a system with two separate parts a glove and a 3D printed forearm was designed.

Five flex force sensors were bonding on the glove. These sensors could collect the actions of the hand's motion and the resistance values from the sensors were transferred into voltages through a resister division circuit. The master controller with a transmitter module processed the inputs from the sensors and sent to the slave controller with a receiver module. The outputs from the slave controller would command the five servo motors to move and control the motion of the 3D printed fingers. The architecture of the system is shown in Figure 1.

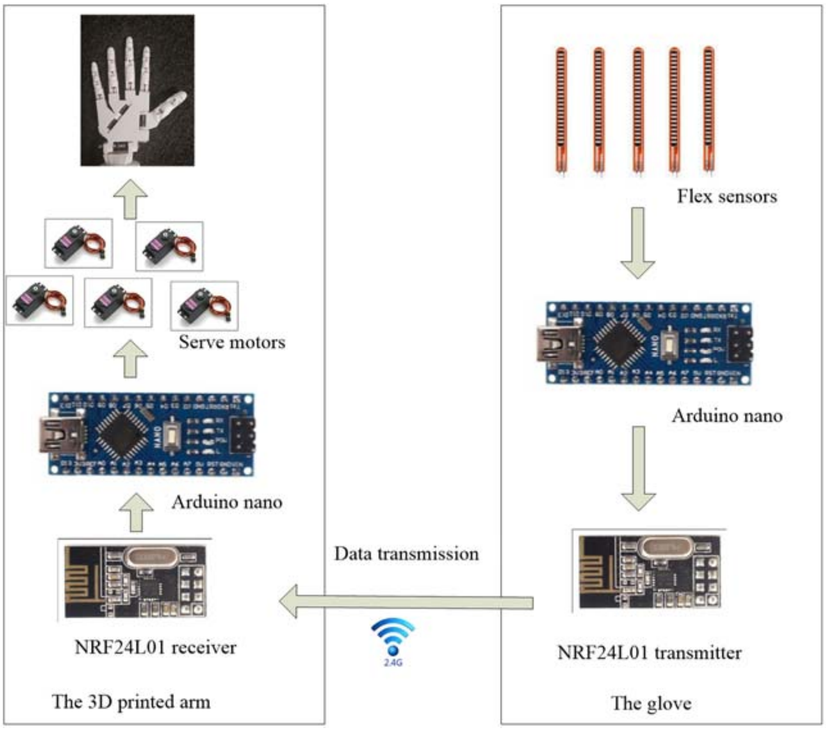

Figure 1. The system architecture of the whole system

2.1.1 3D Printing and Assembling of Forearm

The files of the printed arm were available online for free (http://www.youmagine.com). A white biodegradable PLA material was used to print the arm for its low printing temperature and smooth printing finish [22]. There were a total of 46 individual parts that were printed by 3D Printing. Each of the parts needed to be sanded down to obtain a smooth finish.

- Finger Construction

Each of the printed fingers had three rotational joints. Two different wires were used through the fingers before applying adhesive to the tip and the joints. One of the wires was a braided fishing line (red) used to resemble a tendon and the other was string rope which was used to make the fingers have a pulling force, back to open the hand. This was achieved by tying a knot on the tip of the finger using string rope two times and then tying a knot around the rope using the fishing line. After the knots were made, the adhesive was used to connect the tip of the fingers and hide the knots. To connect the three rotational joints, stainless steel wires of 1 mm and 2 mm in diameter were used. The 2 mm wire was used for the bigger fingers such as the middle finger and the thumb. The adhesive was used in order to make sure that the fingers did not fall apart and help keep its integrity as shown in Figure.2 (a).

- Palm and Wrist Construction

After the parts were printed, the first step was to connect the palm with the wrist which would be the foundation of the arm. Three M6 bolts (6 mm x 80 mm) were used for this part. One of the bolts was used to connect the thumb to the palm. The second bolt was used to connect the wrist to the palm and the final one was used to connect the lower portion of the palm to the upper half. Adhesive and a nut were used to make sure that the wrist did not move when force was applied. The palm and wrist were shown in Figure.2 (b) and the hand in Figure 2 (c).

- Forearm Construction

After the adhesives had dried and all the fingers were moving in the correct manner, the forearm needed to be connected to the wrist. This required connecting two additional pieces to the model and making minor adjustments to the pieces as bits of the filament were not needed, and more space was needed to house the electronics of the model. After the pieces dried again, the motor housing was ready to be placed on the model and the motors with it. The final arm construction is shown in Figure 2(d) and the system with the slave circuit is shown in Figure 2(e).

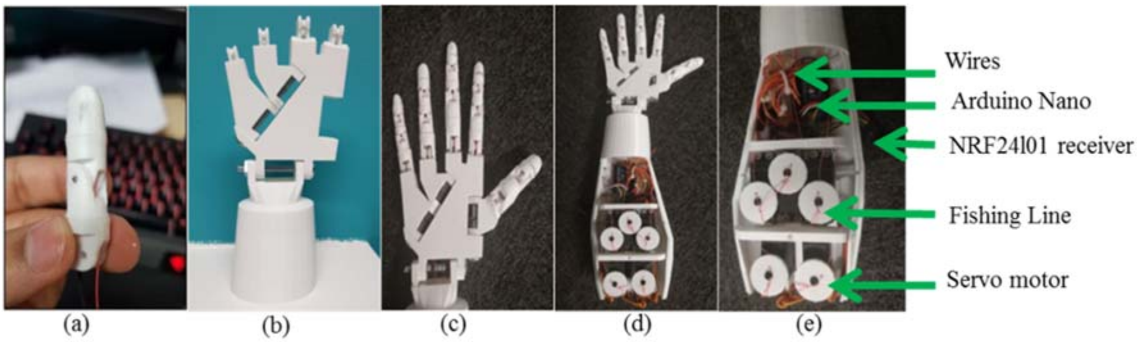

Figure 2. (a) Part of one assembling finger (b) Assembling palm and wrist (c) Assembling 3D printed hand (d) Assembling 3D printed forearm (e) Assembled circuit in the printed forearm

2.1.2. Control Kernel

Two Arduino Nanos acted as a master controller and a slave controller, respectively. Two NRF24L01 transceiver modules acted as a transmitter and a receiver. They worked together to make this system working well in achieving multiple applications.

- Arduino Nano

The Arduino Nano is a small, compatible and breadboard friendly microcontroller board with its dimensions being 45 mm × 18 mm. The analog pins came with a total resolution of 10 bits which measure the value from zero to 5 V. The Arduino Nano was breadboard friendly and had eight analog input pins, 14 digital I/O pins and six Pulse width modulation (PWM) pins. Each of these Digital and Analog Pins could be assigned with multiple functions but their main function was to be configured as input or output. In this study, no data needed storage. All data was used in real time.

- NRF24L01 Transceiver Module

The transceiver module was used for making wireless communication between two Arduino Nano systems. When researching components needed for this study, the main four things were the concerns about cost, range, ease of use and size. This module used a 2.4 GHz band and it could operate with baud rates from 250 kbps to 2 Mbps. The operating voltage of the module was between 1.9 and 3.6 V, which was convenient because the Arduino Nano had a pin that supplies 3.3 V. The current was around 12 mA. Thus, during transmission, the module had very low power consumption.

The transceiver module cost around £1 to £3 only and the range was more than enough. In open space, the transmission distance could work when the distance increased up to 100 meters or more. The size of the module was small because the space available in this system was limited. The module had eight pins which consisted of ground, Vcc, CE, CSN, SCK, MOSI, MISO, and IRQ (IRQ not needed). Three of the pins were SPI (serial Peripheral Interface) pins which were MOSI, SCK and MOSI. The SPI pins were connected to the Arduino pins depending on the type of Arduino used so for the Nano, MOSI was connected to pin 14, MISO was connected to 15 and SCK was connected to 16. The CSN and CE pins could be connected to any of the digital pins of the Arduino, here pin 9 and 10, and they were used for sitting the module in active and standby mode. They are also used to switch between transmit or command mode.

2.1.3. Flex forcer Sensors

The flex sensor in this study was used to measure the angle of the finger. Five flex sensors were bought from Spectrasymbol.com. Each of them was 2.2" (5.588 cm) in length and as the sensor is flexed, the resistance across the sensor increased. The price of each sensor is less than £3.

2.1.4. Other Components

- Actuator

The actuators used in the system were servo motors. The motors could be controlled and rotated 180 degrees. The servo motors were needed for being able to give motion to the model. This was done by connecting the tendon/fishing line to the servo motor and creating a pulley mechanism. The MG996r servo motors had an operating voltage between 4.8 V and 6 V.

- Power Supply

This system was two separate parts. So there were two batteries. Two Lithium batteries CR-V9 9 V (www.gpbatteries.com, China), could provide power lasting 5 times longer than the ordinary alkaline battery, as the power supply for the Arduino Nanos. The input voltage of the Arduino could vary from 7 V to 12 V.

2.2. System Signal Processing and Software Implementation

2.2.1 Flex force sensor calibration and executer

Although the flex force sensors are from the commercial supplier. But it is necessary to calibrate them first because bending sensors typically have different outputs. A test was designed to get the relatives between the angle measurement and the resistance values changes according to the supplier's datasheet. The test process is shown in Figure 3.

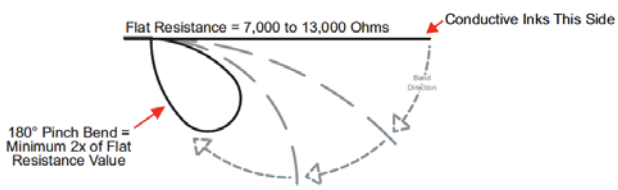

Figure 3. The test process of how the flex sensor works with angles

The flat resistance values were 10.2 kΩ, 11.8 kΩ, 10.9 kΩ, 10.4 kΩ and 11.5 kΩ from the first to the fifth. The sequence was also the order banding on the glove from the thumb, the index finger, the middle finger, the ring finger and the baby finger. And when the angle was 180°, the resistance values were 42.8 kΩ, 43.5 kΩ, 42.8 kΩ, 42.5 kΩ and 43.6 kΩ. There was a 0.1 kΩ stable increase when bending each of the flex sensors. Therefore, the angular accuracy of each flex sensor could be calculated as,

$$A_{cc} = \frac{R_{\max} - R_{flat}}{0.1 \times 180} \quad (1)$$

Here, R_{max} was the resistance value of 180°, and the R_{flat} was the resistance value of flat.

The minimum angular of each flex sensor could detect was 1.81°, 1.76°, 1.77°, 1.78° and 1.78°. These five flex sensors had different flat resistance values response, but each of flex sensor's resistance value was linear relation.

2.2.2 Flex force sensor voltage divider circuit convertor

The voltage divider circuit in Figure 4 was designed to convert the angle measurement to voltage. Here in this study, an op-amp was missed in this circuit. Because many tests were done for stability, and there was no difference whether there was an op-amp in the laboratory. But if this study would be used in the industry field with a noisy environment, an op-amp LM258 was necessary.

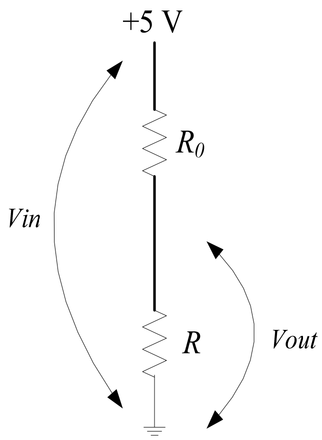

Figure 4. Voltage divider circuit of converting the resistance value to the voltage for a flex force sensor

The transmitter circuit had a supply voltage of 5 V regulated from pin 27 of the Arduino board (shown in Figure 6) which was then connected to R_0, the nominal resistance of the flex sensor. The voltage out was being measured across *R* with respect to ground and the flex sensor. By using the voltage divider formula, we could calculate the amount of voltage across the flex sensor and this would also give the amount of output voltage,

$$V_{out} = V_{in} \times \frac{R}{R_0 + R} \quad (2)$$

From the formula, the nominal value of the flex force sensor could be taken and calculated to find out how much voltage was being outputted. When V_{in}=5 V, R_0=R_{flat}=R, the output voltage V_{out}=2.5 V when the flex sensor was flat. This indicates that the more the flex force sensor was bent the more voltage was being outputted. This is very important because this helped to take the voltage outputted and feed the value to the Arduino analog pin which would convert it to digital. This could then be displayed on the serial monitor on the Arduino interface. Within the design, five voltage dividers were needed because five flex sensors were used to control the robot forearm.

2.2.3 The transmitter model

The main idea was to create a glove which could control a robot forearm wirelessly and to make the glove as comfortable as possible while operating. Using the circuit design, it would be implemented in a small area that can fit on a glove and not interrupt finger movement, thus the only place for this was the back of the hand. For the movement of the 3D model to work, the flex sensors needed to be mounted on the fingers of the glove. This was done by making incisions in the glove and using zip ties to fit them in place.

Due to the limitation of very little space on the back of the hand but all the circuitry needed to be mounted there, a PCB along with a mini breadboard were used. The mini breadboard was used for the Arduino Nano because the model was needed to test without soldering the pins and then cut copper off the PCB to separate the pins. This made the process a lot easier when conducting tests during the instances when the circuit was not working and debugging the code. For connections to be made, jumper cables were used in the transmitter model. Male to male connectors were needed to make connections between the breadboard and the PCB, while the male to female connectors were needed to make a connection between the NRF24L01 and the breadboard. The actual glove with sensors and circuits was shown in Figure 5.

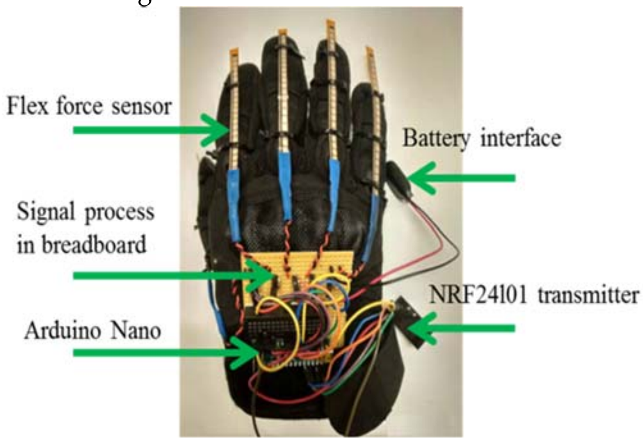

Figure 5. All the components of the master controller subsystem on the glove

2.2.4 Arduino Code Developed

Two individual sets of code were developed for the system. One set of code was used as transmitter and the other for the receiver. The code was introduced in detail below and for debugging the procedure, the code of the series port was presented.

- Transmitter

At the beginning of the sketch, the *#include* was used to add external library's which are available to use. In the study, the NRF24L01 was used which needed an external library for the module to work. With the module that signals were going to be sending a total of five numbers of data to another module, so an *Int msg[5]* was declared with this amount. Flex sensors were being used as input control for the forearm to move. For the flex sensor to work, they needed to be connected to the analog pins of the Arduino to get a measurement for the analog voltage. These were connected to the pins A1 to A5. These pins could measure the voltage between 0 to 5 V. In addition, due to getting a varying voltage, *int flex_5_val* was declared to store these values and this was done for all five sensors. The NRF24 object was defined and connected to the Arduino and the argument was the pins 8 and 9 which were connected to the digital pins of CE and CSN. This was written in both codes because there were two modules. Next, an unsigned integer c*onst uint64_t pipes[2]* was defined making it hold whole numbers and not negative numbers. The 64 refers to the number of N bits that can be stored, so 2^{64} followed by the variable name. This was done to create an address for a pipe for the transmitter and receiver. Next came to the setup, *radio.begin()* was needed for the code because it would start the object which was the module. *Serial. begin(9600)* was next, 9600 was the Baud rate for monitoring the data from the system on the serial port in a computer and it usually was used to transfer data in the industry field. The data from the flex sensors were displayed on the serial monitor and could be seen whether it changed in real time. Due to the flex sensor being connected to the analog pins of the Arduino board, *analogRead* was used. This was done for all the five pins of the Arduino and assigned to the variable *flex_x_val*. For the next line, the map function needed to be used because the values of the numbers would be ranging from a specific number to another when these numbers reach a certain value. When doing this, error codes were made. If the value of flex sensor 5 was less than 0° then made it equal to 0°. And if the value of each of the five sensors was greater than 180°, the value was set to 180°. And 0°~180° was the angle of the finger. Next, the values of the flex sensor were assigned to an integer in the message from *msg[0]* to *msg[4]*. The rest of the code was used to get the values of the transmitter on the serial monitor. The pin wiring diagram was in Figure 6 and the software flowchart of this procedure was shown in Figure 8(a).

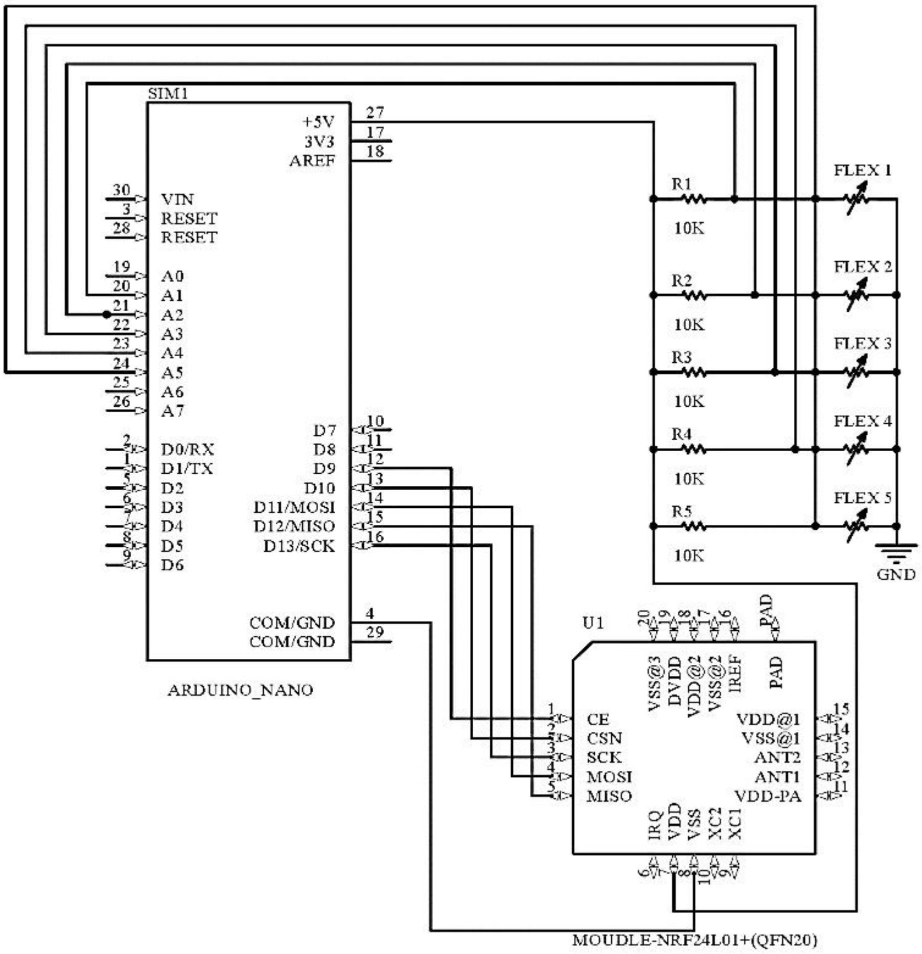

Figure 6. The pin wiring diagram including the transmitter module and the flex force sensors with the master Arduino Nano controller

- Receiver

Two libraries were needed in the receiver side of the code, one was for the NRF24l01 and the other was for the servo motor so that they could be controlled. After this, the servo motors were defined and assigned with names. For the setup, to define the servo inputs that were connected to the Arduino such as *myservo1.attach(A1)* means that the first servo named myservo1 was connected to the argument A1 (analog pin 1). To activate the module in this code had to use *radio.begin()* again. The lines allow for two-way communication, but here only needed one-way communication. *Radio.startListening()* enabled the module to begin receiving data from the other module. Within the loop, an *if statement* was written so that if the radio was available, it would begin to send the message from one of the ports to another via the NRF. This was done for all the ports by assigning a message value to each port from 0 to 4. For the result to be displayed, a serial monitor was created within the loop. During the testing, commenting out the serial print lines needed to be done. The pin wiring diagram was in Figure 7 and the software flowchart of this procedure was in Figure 8(b).

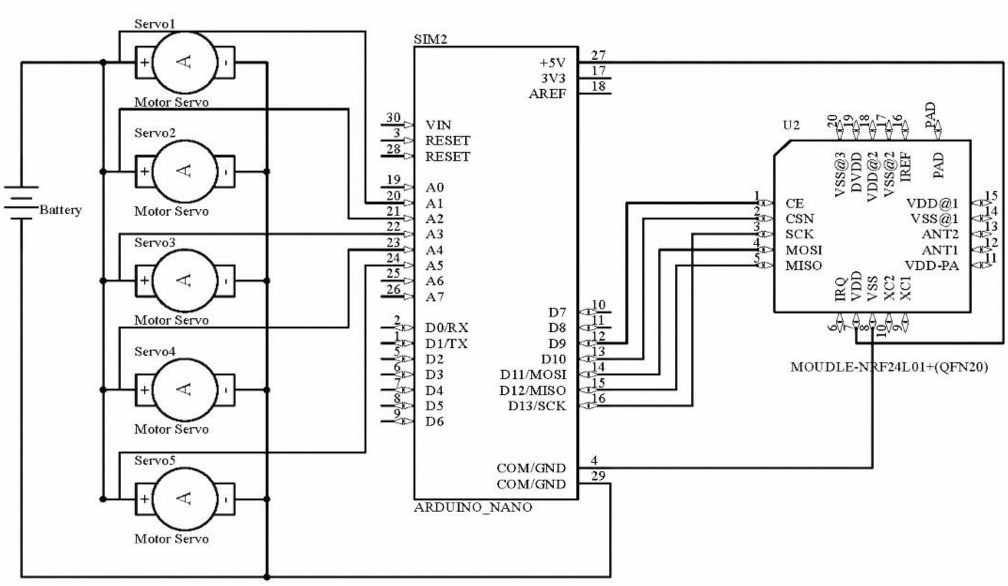

Figure 7. The pin wiring diagram including the receiver module and the servo motors with the slave Arduino controller

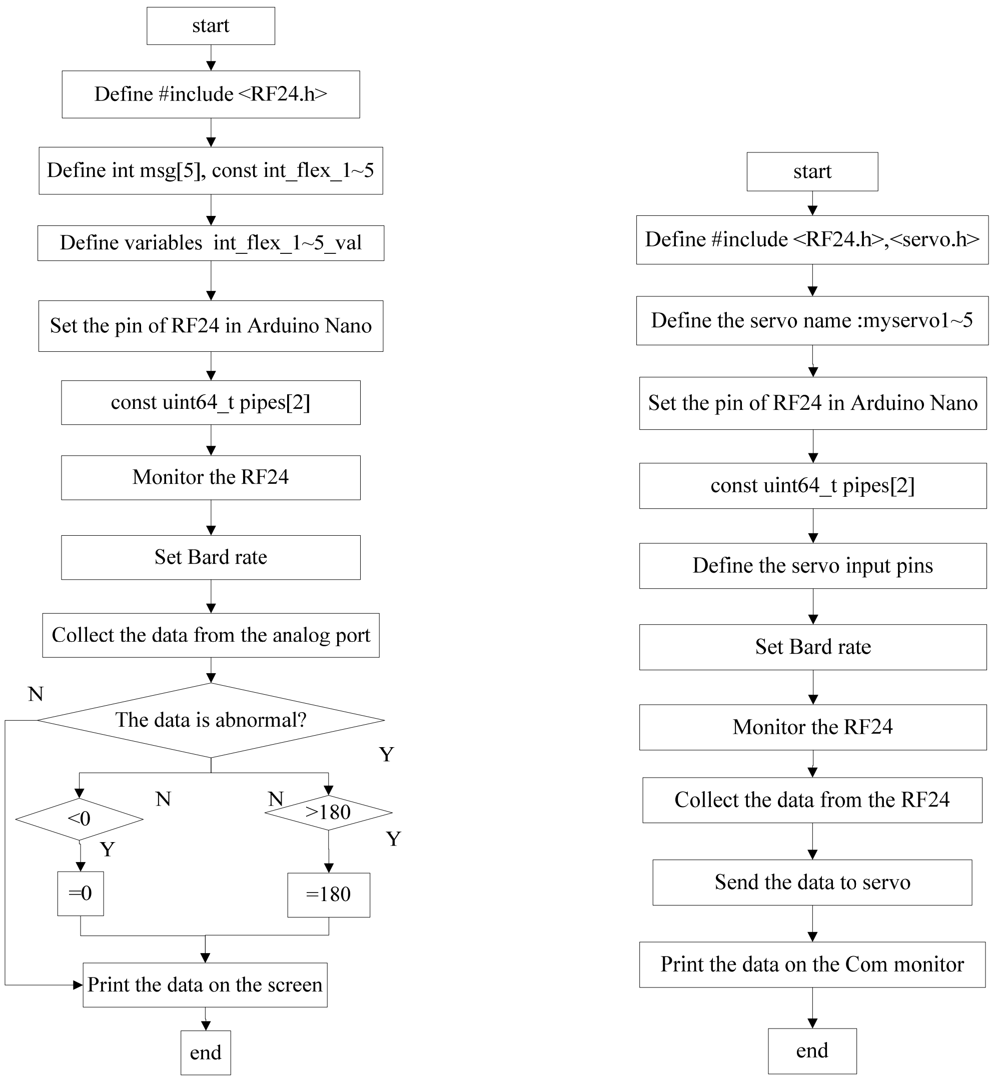

Figure 8. (a) Flowchart of data processing in the master controller subsystem (b) Flowchart of the data processing in the slave controller subsystem

3. Tests and Results

The robotic system could detect the movement from the sensor. For the prototype system, low cost and robustness were the two most important things. Using less components with high quality and low prices would be able to reach the first goal. This had been discussed in Section 2. When the system was implemented, lots of tests were made to detect the repeatability, stability, sensitivity and time delay of the system to guarantee the robustness. The outputs from the assigned Serial port mentioned above showed the status of each finger.

3.1. Repeatability and stability tests

3.1.1 Tests for action initialization

Because the glove material was stiff as seen in Figure 5, not every finger's output was 0°. And when using the glove, the initial value of each sensor was different. A test was provided by putting the gloved hand gently on the desktop shown in Figure 9. 0° was the data from the thumb, the first finger. And the red bar was from the index finger, the second finger, 17° or 16°. The blue bar was from the middle finger, the third finger, 25°. The purple bar was from the ring finger, the fourth finger, 41° and the green bar was from the baby one, the fifth finger, 36°. There were five groups data in about one minute in this chart and there was only 1° angle difference in the second finger, red bar, the angle of the finger was 16° not 17° at the fifth group.

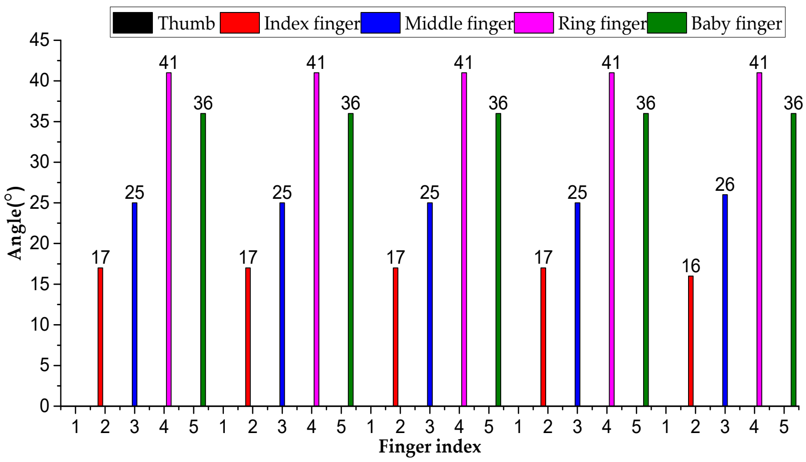

Figure 9. The flex force sensors' signals response banding on the glove and put on the desktop

3.1.2 Dynamic tests

Data were monitored between the transmitter from the glove and the receiver from the 3D printed forearm. The user wearing the glove bent their fingers randomly. In total, 30 samples of data were collected shown in Figure 10. The green bar was from the glove and the red bar was from the printed arm. The blue point was the percentage error between the transmitter and the receiver. The angle difference was ±1°. It was the same as the static tests. But the probability of the angle difference was higher than the static tests. The percentage error was

$$R = \frac{A_{trans} - A_{rec}}{A_{trans}} \times 100\% \tag{3}$$

Here, A_{rec} was the angle from the receiver; A_{trans} was the angel from the transmitter. R was the percentage error ratio, according to the test results, the percentage error ratio R was between (-0.91%~+1.07%).

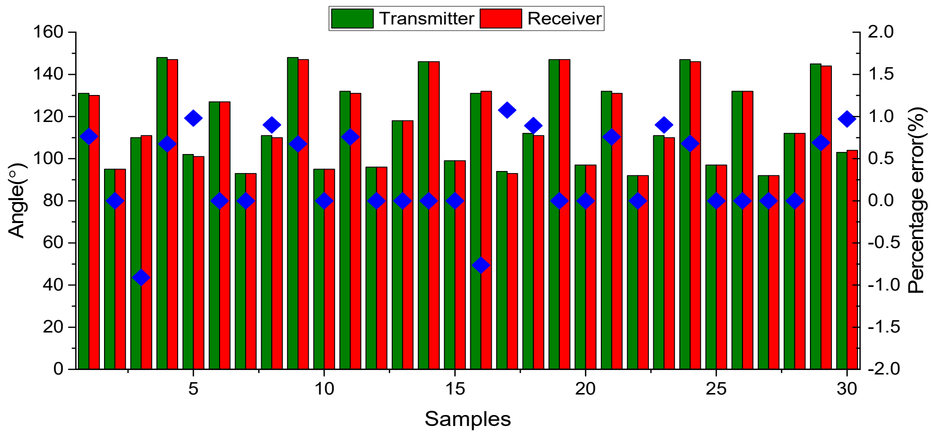

Figure 10. The angle of the fingers, the percentage errors between the transmitter subsystem and the receiver subsystem when moving

3.1.3 Repeatability of each of the data glove sensors

The repeatability of each of the data from the glove sensors was vital for the system. Lots of same tests were done repeatedly to evaluate this parameter. After bending each finger, the data was collected from the series port. The repeatability results were shown in Figure 11. This result showed that each sensor of the glove sensors was linear with a different resolution. The resolution was 2.64 °, 2.28 °, 2.34 °,1.78 ° and 1.93 °.

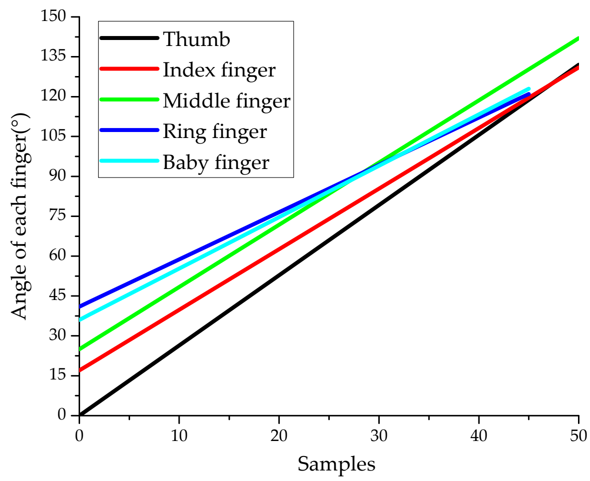

Figure 11. The repeatability of each of the data glove sensors

3.2 Sensitivity tests

Two different tests were designed to test the sensitivity. One was to get the minimum angle of each finger would bend. The other one was to test the response time when there was a change in the finger's movement.

When testing the minimum angle of each finger, the movement of the finger was kept as slowly as possible. Testing was done from holding the hand in its natural state to closing it like a fist. The maximum of each finger was 130°, 131°, 142°, 121° and 123° in order. After that, tests were made to get how quickly the values could react to the flex sensor when the finger was moved as fast as possible. The data with 28 values collected from 15:38:37:737 to 15:38:38:79 in Figure 11 from the transmitter in the black colour. Six group data were obtained. They were 3 values in 37:737 which was the start time, called 0, 5 values after 133 ms, another 5 values after 206 ms, 6 values after 240 ms, 4 values after 304 ms, and 5 values after 343 ms. The minimum resolution was 3 times per millisecond, and the maximum was 6 times per millisecond. From the terminal of the receiver, 28 values were also collected from 37:870 to 38:144 in Figure 12 in the red colour. But this time, seven group data were monitored. They were 3 values at 69 ms, 2 values at 167 ms, 3 values at 202 ms, 5 values at 236 ms, 274 ms, 342 ms, and 373 ms, respectively. In this time, the minimum resolution was 2 times per millisecond, and the maximum was 5 times per millisecond.

It was easy to find the data transferred asynchronously. In 133 ms, 5 data transmitted, but 2 were collected after 34 ms at 167 ms and the other three were collected after 69 ms at 202 ms. Another, in 240 ms, 6 data were transmitted, 5 were collected at 240 ms, the last one was collected at 342 ms which were collected with the other 4 data together. And these 4 data were transmitted at 342 ms.

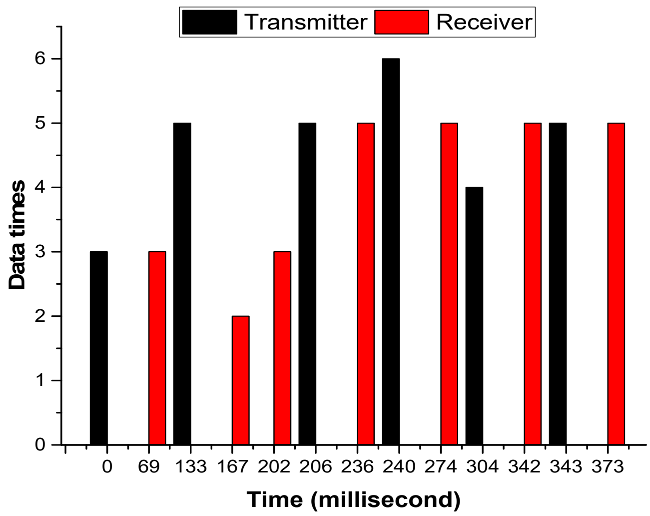

Figure 12. The collecting data numbers in the transmitter subsystem and the receiver subsystem

3.3 Time delay tests

It was observed that there was a time delay when the data transferred from the transmitter to the receiver. The data send from 15:38:37:737 to 15:38:38:79, but the receiver obtained the data from 15:38:37:870 to 15:38:38:144. The time delay was calculated as follows,

$$D = | T_{rec} - T_{trans} | \tag{4}$$

here, T_{rec} was for the receiver time, T_{trans} was for the transmitter. D was the absolute error of the time delay. The time delay was about 0.03 ms~0.133 ms during this period of time shown in Figure 13.

Though the sequences of the data collected in the receiver were in the same order with the data from the transmitter, the receiver times and the transmitter times were different, there was 8 delay time. Two special positions were shown in this figure. One was in the rectangle, these two different delay times received data from one same transmitter time. Here, there was the biggest delay time, 0.133 ms. Another was in the oval, the receiver collected data from two different transmitter time. The smallest time delay was not on these two special times, it was 0.03 ms happened at 236 ms.

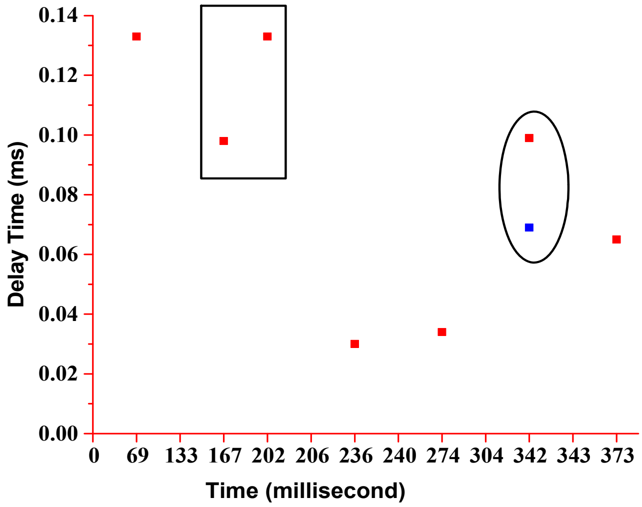

Figure 13. The delay time between the transmitter subsystem and the receiver subsystem

4. Discussions

First of all, each component was experimented individually by running the code for each function to analyze the results and performance. Then codes of each function were combined, executed simultaneously, and all the components were integrated into the glove and the 3D printed forearm for implementation. In order to reduce cost and complexity, electronic components small in size were used.

The system was simple and easy to duplicate. It was useful if designing an aid device to do some rough tasks with limited functions. Due to using a whole flex force sensor to detect the finger's movement, it was impossible to detect the tiny movement of the hand.

5. Conclusions and future work

A wireless-controlled 3D printed robotic hand motion system has been successfully designed, fabricated and tested. The system is composed of a gesture control glove and a 3D printed forearm, where the control kernel was performed by two Arduino Nano chips. The electronic embedded system was implemented and succeeded in integrating five flex sensors, five servo motors, a transmitter module, a receiver module, a master Arduino Nano controller and the slave Arduino Nano controller in the system. All the components worked well and provided accurate data from the flex sensors. The system was easily controlled, fast response time, robust and low cost with a small number of small size electronic components. There was a maximum 0.133 ms time delay between the gesture control and the printed arm motion. This system would probably be helpful for people who work in a dangerous environment for their jobs, disabled people or elder people in their daily lives. But due to the delay time, this system could not be used in any place with very higher precise timing and long term reliability. In the future, a flex glove should be designed and fabricated to detect the gesture of the hands. That system would have much higher accuracy.

Acknowledgements: This research is supported by the Innovate UK (Grant 104013), the Science and Technology Commission of Shanghai Municipality (STCSM) (Grant 17230732700), EPSRC UKRI (EP/T024682/1), the China Scholarship Council Fund (Grant No. 2018 082 10359), the institutional strategic grant - Global Challenges Research Fund (GCRF) that City, University of London receives from Research England, UK Research and Innovation (UKRI). We thank the website http://www.youmagine.com.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] Health and Safety Executive, Workplace fatal injuries in Great Britain, (2019) 1-16, http://www.hse.gov.uk/statistics/pdf/fatalinjuries.pdf.

[2] Kawasaki H., Ito S., Ishigure I., Development of a hand motion assist robot for rehabilitation therapy by patient self-motion control, Proceedings of the 2007 IEEE 10th International Conference on Rehabilitation Robotics, Noordwijk, Netherlands, June 12-15. (2007) 234-240. https://doi.org/10.1109/icorr.2007.4428432

[3] Marco S., Martha F., John F.S., Patterns of hand motion during grasping and influence of sensory guidance, J. Neurosci. 4, 22 (2002) 1426-1435.

[4] Howard W. L., Dean P. A., Geoffrey M. G., Game with hand motion control. US patent 9317110B2. (2016).

[5] Daniel E.W., Resolved motion rate control of manipulators and human prostheses, Trans. Man-Mach. Syst. MMS-10 (1969) 47-53. https://doi.org/10.1109/TMMS.1969.299896

[6] Carter S. H., Marcio D. L., Na L., Artificial muscles from fishing line and sewing thread, Sci. (2014) 868-872. https://doi.org/10.1126/science.1246906.

[7] Lee W. W., Tan Y. J., Yao H.C., A neuro-inspired artificial peripheral nervous system for scalable electronic skins, Sci Robot. 4, eaax2198 (2019) 1-15. https://doi.org/10.1126/scirobotics.aax2198.

[8] Duduta M., Hajismaili E., Zhao H.C., Realizing the potential of dielectric elastomer artificial muscles, Proc. Natl. Acad. Sci. U.S.A. 1073 (2018) 1-6. https://doi.org/10.073/pnas.1815053116.

[9] Zhang X., Chen X., Li Y., A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors, Trans. Syst. Man Cybern. A. 41 (2011) 1064-1076. https://doi.org/10.1109/tsmca.2011.2116004.

[10] Rautaray S. S., Agrawal A., Vision based hand gesture recognition for human computer interaction: a survey, Artif Intell Rev. 43 (2015) 1-54. https://doi.org/10.1007/s10462-012-9356-9.

[11] Xu R., Zhao S. L., Li J. W., MEMS accelerometer based nonspecific-user hand gesture recognition, Sensors J. 12 (2012) 1166-1173. https://doi.org/10.1009/jsen.2011.2166953.

[12] Vuletic T., Systematic literature review of hand gestures used in human computer interaction interfaces, Int J Hum-Comput St. 129 (2019), 74-94. https://doi.org/10.1016/j.ijhcs.2019.03.011.

[13] Chossaty J. B., Tao Y. W., Duchaine V., Wearable soft artificial skin for hand motion detection with embedded microfluidic strain sensing, 2015 IEEE International Conference on Robotics and Automation (ICRA), Washington State Convention Center, Seattle, Washington. May 26-30 (2015) 2568-2573. https://doi.org/10.1109/icra.2015.7139544.

[14] Ma Z.H. C., Ling Y.T., Shea Q.T.K., Towards wearable comprehensive capture and analysis of skeletal muscle activity during human locomotion, Sensors. 19 (2019) 1-13. https://doi.org/10.3390/s19010195.

[15] Yi B.F., Harris F.C., Dascalu S.M., User interface aspects of a human-hand simulation system, Syst. Cyb. and Info. 3(5), (2006) 1-7. https://www.cse.unr.edu/~dascalus/JSCI2006_BY.pdf

[16] Shi Q.F., Zhang Z.X., Chen T., Minimalist and multifunctional human machine interface (HMI) using a flexible wearable triboelectric patch, Nano Energy. 62 (2019), 355-366. https://doi.org/ 10.1016/j.nanoen.2019.05.033.

[17] Fu J. M., Cavusoglu M. C., Human-arm-and -hand-dynamic model with variability analyses for a stylus-based haptic interface, Trans. Syst. Man. Cybern. B. 42 (2012) 1633-1644. https://doi.org/10.1109/tsmcb.2012.2197387.

[18] Eranki G. R. R. V. K.P., Design and Structural Analysis of a Robotic Arm, Master degree Thesis, Blekinge Institute of Technology, Swedish, (2016).

[19] Carmine D. B., Luca R. B., Vincenzo L., Design, modeling and control of a 5-DoF light-weight robot arm for aerial manipulation, The 23rd mediterranean conference on control and automation (MED), Torremolinos, Spain, (2015) 853-858. https://doi.org/10.1109/med.2015.7158852.

[20] Purohit K., Patidar K., Kushwah R. S., A wearable hand gloves gesture detection based on flex sensors for disabled people, IJRASET. 5 (2017) 1825-1833.

[21] Zamir A. A., Zhang Y.F., Chen N.L., A novel flex sensor-based flexible smart garment for monitoring body postures, J Ind Text. (2019) 1-13. https://doi.org/10.1177/1528083719832854.

[22] Bàrbara M.V., Esther P., Khalil H., Maximization of FDM-3D-objects gonio-appearance effects using PLA and ABS filaments and combining several printing parameters: a case study, Materials. 12 (2019) 1-13. https://doi.org/10.3390/ma12091423.