Journal of Electrical Engineering & Technology (2022) 17:665–683
https://doi.org/10.1007/s42835-021-00903-5

ORIGINAL ARTICLE

On the Implementation of a Low-Cost Mind-Voice-and-Gesture-Controlled Humanoid Robotic Arm Using Leap Motion and Neurosky Sensor

Quan K. Pham[1] · Truong V. Vo[1] · Phuong T. Tran[2]

Received: 13 February 2021 / Revised: 24 August 2021 / Accepted: 3 September 2021 / Published online: 13 September 2021

Abstract

In this paper, a low-cost prototype for humanoid robotic arm that is controlled by either voice, gesture or brain signals is introduced. The prototype consists of a robotic arm that can rotate around a vertical axis and a hand with five fingers, which animates approximately the movements of a human hand. This model can be operated by brainwave signals from a Neurosky MindWave™ Mobile 2 Brainwave Sensing Headset, by human gesture using Leap Motion device, by voice command using Google Assistant, or by web-based interface. All operations can be monitored by a user interface on web server, with the support from a Raspberry Pi board. The experiments show that the average accuracy of our mindwave control prototype is about 83%, which is similar to most of previous high-cost solutions, and can go up to nearly 100% for the case that the data acquisition time is larger than 10 s (which improves 2–8% compared to previous works). This model would be highly useful in practice, especially, for people with disabilities who cannot use their hands, as well as in academia when it is used for undergraduate students' experiments on robotics.

Keywords Brainwave control · Gesture control · Humanoid robot · Leap motion · Neurosky · Robotic arm · Voice control

1 Introduction

This research is inspired by disabilities and patients with motor deficit due to stroke, spinal cord injures, etc., who cannot control their arms and hands to do normal tasks. Independent living is essential for such people [1] to live their life. In order to assist the above patients to live independently, intelligent robotics technologies such as voice, gesture, or mind control are attractive solutions [2–4]. Especially, humans always have a desire to communicate and interact with machines through the thought and also create devices that work with human mind and thought. Recently, cognitive neuroscience and brain imaging technologies have started to provide people with ability to interface with human brain. From the effort to interact human brain with the machines around us in order to make them functional with brain signals, the field of brain-machine interface (BMI) has come to existence, and brain-robot interaction is a special topic of this field.

Brain-Robot Interaction (BRI) refers to the ability to control a robot system via brain signals and is expected to play an important role in the application of robotic devices in many fields [4, 5]. Among a variety of robotic devices, humanoid robots are more advanced, as they are created to imitate some of the same physical and mental tasks that humans perform on a daily basis [6]. In this section, we focus only on interaction between brain and humanoid robot. Brain signals are acquired by different approaches, including magnetoencephalogram (MEG), near infrared spectroscopy (NIRS), electrocorticogram (ECoG), functional magnetic resonance imaging (fMRI), and electroencephalogram

Phuong T. Tran at Ton Duc Thang University, Ho Chi Minh City, Vietnam.

Phuong T. Tran
tranthanhphuong@tdtu.edu.vn

Quan K. Pham
41502080@student.tdtu.edu.vn

Truong V. Vo
41801275@student.tdtu.edu.vn

1 Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam

2 Wireless Communications Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam

(EEG) [7]. On the other hand, brain signal acquisition devices are usually classified into two categories: invasive and noninvasive devices [8]. For invasive Brain-Computer Interface (BCI) systems, arrays of microelectrodes are permanently implanted in the cerebral cortex to record the brain signals from ensembles of single brain cells or the activity of multiple neurons. This technique was first applied on rats by [9] and mostly knew by the succesful experiment of [10] that allowed two patients with long-standing tetraplegia to control a robotic arm for drinking. However, this method requires operation on users in advance, so it's not a good choice for most people. Therefore, noninvasive technology, which uses EEG devices to record the electrical activity occurring at the surface of the brain by using electrodes or sensors placing on the scalp, has attracted more attention because it's easy to implement, low-cost, and low-damage to human bodies.

EEG signals are nothing but the brain activity in the form of electro-voltaic waves. Mostly the signals are used to monitor the brain activity of people. But nowadays EEG contains a fully developed sector, which is known as BCI or specifically, BRI. Several of EEG-based BRI algorithms and systems have been developed and experimented in recent years. Based on the types of electrical potentials that can be acquired through EEG for the development of control models, the BRI paradigms can be categorized into motor imagery (MI), the steady-state visual evoked potentials (SSVEPs), and the P300 potentials [4]. Among them, the first one is an active paradigm, while the last two are passive paradigms [11]. MI potentials, also known as mu/beta rhythms, are induced by the motor cortex through the spontaneous imagining of body movements. Some reports on using MI to control the humanoid robots have been found, such as robotic hand [12] and walking gait of a simulated humanoid robot [13]. This method is easier to implement than the other methods, but has limited classifiable states and relatively low classification accuracy. The SSVEP is the potential that naturally responds to visual stimuli at specific frequencies. SSVEP-based model for directing a humanoid robot in performing a pick-and-place task has been reported by Tidoni et al. [14]. Zhao et al. [15] developed a behavior-based SSVEP hierarchical architecture for effecive telepresence control of a humanoid robot to achieve full-body movement. P300 is another BRI method that does not require subjects to learn how to modulate their EEG. The P300 potential is an event-related potential (ERP) with a positive deflection that is time-locked to auditory or visual stimuli. P300 and its combination with SSVEP have been found in many studied on BRI systems with humanoid robots recently. In [16], the authors applied SSVEP and P300 to develop a new brain-robot interaction system by fusing human and machine intelligence to improve the real-time control performance. SSVEP and P300 were compared in a review by Zhao et al. [4], in which the authors showed the advantages of each method in BRI systems.

SSVEP, P300, and their hybrid solutions have shown the effectiveness in controlling humanoid robots. However, they requires more complicated devices and equipment that may not be handled in academic institutions such as universities, etc. Furthermore, in order to improve the accuracy and information transmission rate, these two methods require efficient classification algorithms such as neural networks or machine learning, which also raise some problems in implementation and testing [17, 18]. In this paper, we introduce a very-low-cost mind-controlled humanoid robotic arm design using the simple MI method. We record and analyze the brain waves concentration and meditation values, which are acquired by the Neurosky Mindwave EEG Sensor, to trigger the process on Raspberry Pi control module to make the arm rotate around its axis and move the fingers of its hand.

In order to make our robotic arm model become more practical and can simulate the basic operations of a real human arm, other functionalities have been added. First, not only the robot hand and fingers can be controlled by brain waves, but they can also be controlled by voice command and by human gesture. The voice-controlled capability is implemented by using Google Assistant (GA), while the gesture control is applied by using a Leap Motion device. All operations of the robotic arm as well as the control parameters are shown on a web-based user interface. Google Assistant is a voice assistant tool, which is developed by Google, thanks to the ability to quickly perform voice transcription and natural language processing. Google also provides software development kit (Google Assistant SDK) for hardware manufacturers and hobbyists to embed this technology in their own Internet of Things (IoT) devices [19]. In our work, Google Assistant is integrated in a Raspberry Pi board. On the other hand, Leap Motion (LM) controller is a device that allows the creating of a 3D space to identify the position of the hand and the fingers' movements in the space, based on the control gestures of users' hands in real time. LM has been used in many robotic applications such as in [20, 21]. In [20], the authors tried to establish how similar or different a query dynamic hand gesture is in comparison to a reference dynamic hand gesture whilst compensating for differences in the duration of gestures, rotation of the hand, reasonable distance from LM sensor using Multi-dimensional Dynamic Time Warping (MDTW) approach. In [21], LM controller is applied to perform the sign language recognition.

The main functionality of our prototype, which is the control of the robotic arm using brain wave signals, is performed by using a Neurosky sensor. NeuroSky was one of the earliest players that took up the production and market distribution of consumer-grade EEG devices [22]. Among the consumer-grade EEG measuring sensors, Emotiv products are popular in cognitive study and gaming while NeuroSky products dominate the educational field. NeuroSky has been used in experimental research, especially in the recent years, such as [23–25]. The applications presented in these above works were on evaluation e-learning activities, wheelchair control, and brain exercise for elderly people. More applications of Neurosky can be found in a review by Sawangjai et al. [22]. With Neurosky sensor, handicapped peoples and speechless ones can control the robotic arm and hand to do simple operations. This model would be highly useful in practice as long as it has more upgrades in hardware and mechanics, because it can support human working in dangerous, hazardous environments or helping the elderly and people with disabilities in their daily activities.

2 System Design

Figure 1a illustrates a classical robot arm model, which is the typical type of robotic arm in factories or workshops. This model is suitable for performing repetitive operations according to a certain procedure, but for direct control from humans they have certain limitations as follows. Since this construction is fixed, it can only handle objects within a certain range, which is limited in size as well as material of the objects to be held, and is not suitable for use within a family. The robotic arm model made from our work is a part of the Humanoid Robotic project [26], which is similar in size and shape to a real human arm. Therefore, this model will overcome the above-mentioned disadvantages of the classical model. Figure 1b shows the 3D model of the robot arm designed for this work. In fact, the robotic arm model in this work is a launch pad for the development of the overall design project about humanoid robotic model that can walk, hold objects, think and behave like humans.

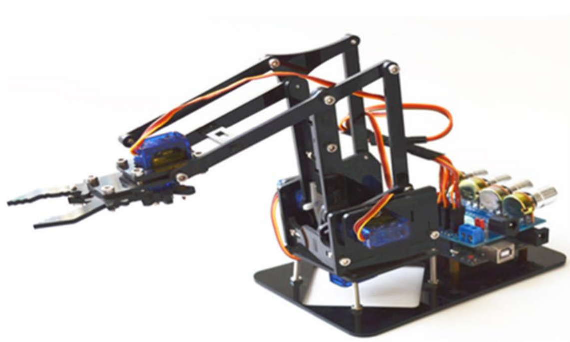
2.1 System Structure

The robotic arm in this paper can be controlled by one of four methods: web-based control, voice control, gesture control, and mindwave control, as shown in Fig. 2. Gesture control is implemented by two methods: using Leap Motion controller or using MPU6050, a six-axis accelerometer and gyroscope that is attached to a special glove worn by the user.

Leap Motion is a device that uses three monochrome infrared cameras and three infrared LEDs to create a spherical space with a radius of about 1 meter in order to support users' interaction with computers similar to mice or keyboards through hand gestures such as pointing, holding, touching, etc. [27]. For the Leap Motion block, after connecting the Leap Motion to the USB port from a laptop to record the human hand gesture, we use a processing software to identify position of the hand in space, as well as to determine whether the fingers' state is closed or opened via an identification (ID) assignment algorithm. From there, we take the data after processing and transfer it into the microcontroller and robotic model according to the corresponding gestures. Another gesture control technique is to use Flex sensors and MPU6050. The intended user wears a glove that has MPU6050 and Flex sensors on it. Hand gestures of the user can be received in terms of sensing data, then we need a controller (we use an Arduino Nano board in our design) to convert them to the corresponding commands and send to the central microcontroller via RF signals. Two RF HC12 modules attached to the glove controller and the central controller will do the transceiver task.

For web-based control, we use a Raspberry Pi module (model 3B+, see Fig. 3) as a web server to build a web-based user interface to control the robot by user's commands. Raspberry Pi is a compact-size embedded computer handled by the Linux operating system. This is a great choice for IoT

Fig. 1 Robotic arm models



(a) Classical robot arm model

(b) Robotic arm for our work

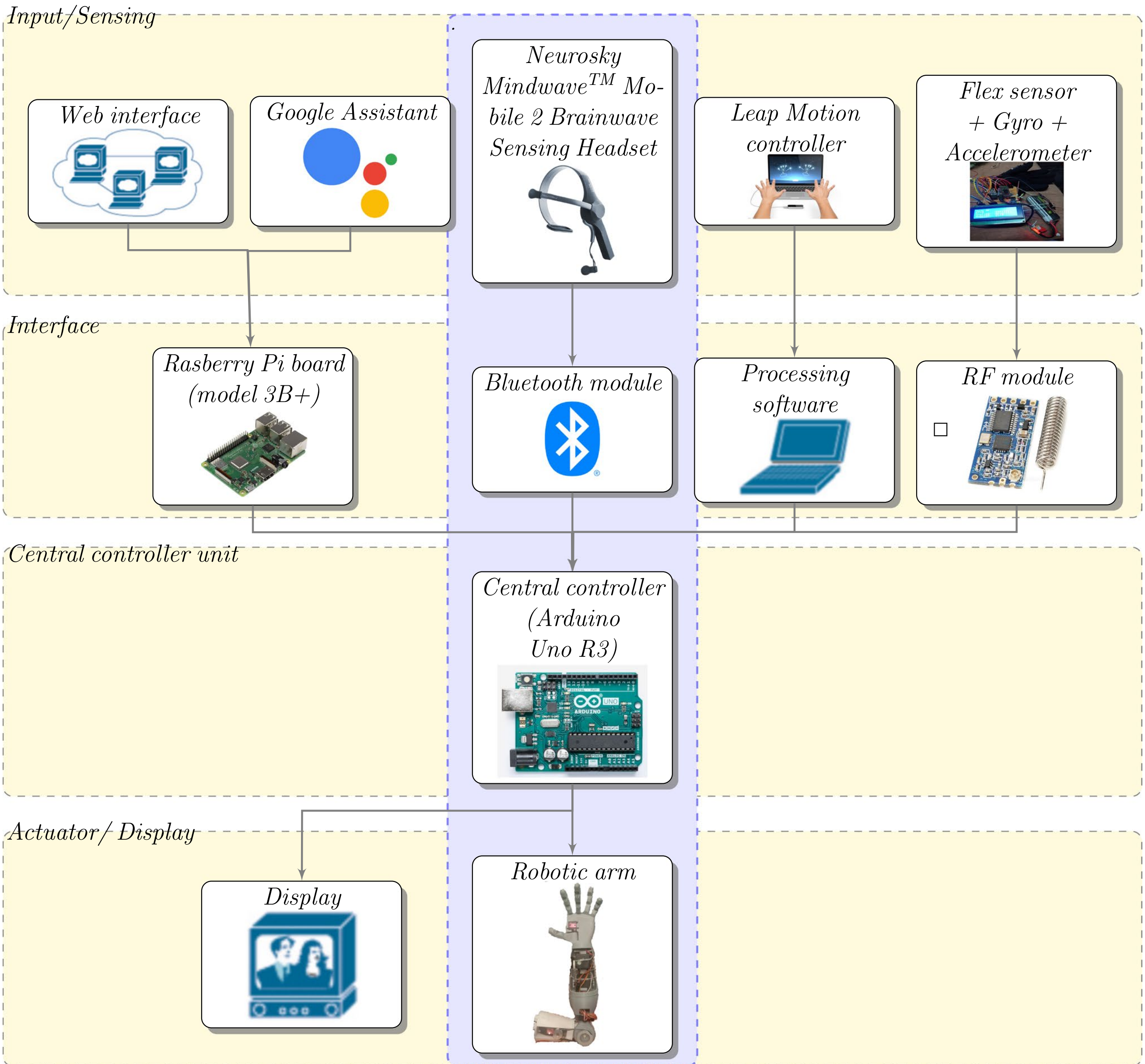


Fig. 2 System structure of the designed model

projects, because Raspberry is equipped with support functions similar to a mini computer with a memory card used to make hard drives, 4 USB ports, ethernet ports, HDMI port, camera port, audio port along with WiFi connectivity, Bluetooth and also integrated pins for GPIO peripheral control as well. To observe the motion of the robotic arm in real time, a camera is connected to Raspberry Pi via USB port, which continuously records the images of the robotic arm as well as the nearby environment and display on the web interface.

For voice control, we integrate the Google Assistant (GA) into the web interface provided by Raspberry Pi module to process voice control function. Voice commands are input from a USB microphone connected to the Raspberry Pi board. An A5 mini speaker is also connected to Raspberry Pi to help users listen to the feedback from GA. In addition, we also use IFTTT to configure the response statement from GA and use Webhook to bring data to the web server. Then, we split the data series and push it back to the microcontroller, which handles the rest. In another aspect, voice commands may also be retrieved from Android smartphones. Voice commands from smartphones are sent to the central controller via Bluetooth.

Finally, mindwave control is implemented using a Neurosky MindWave^{TM} Mobile 2 Brainwave Sensing Headset [28], which transmits information from EEG signals to the control center by Bluetooth. For the Brainwave Sensor block,

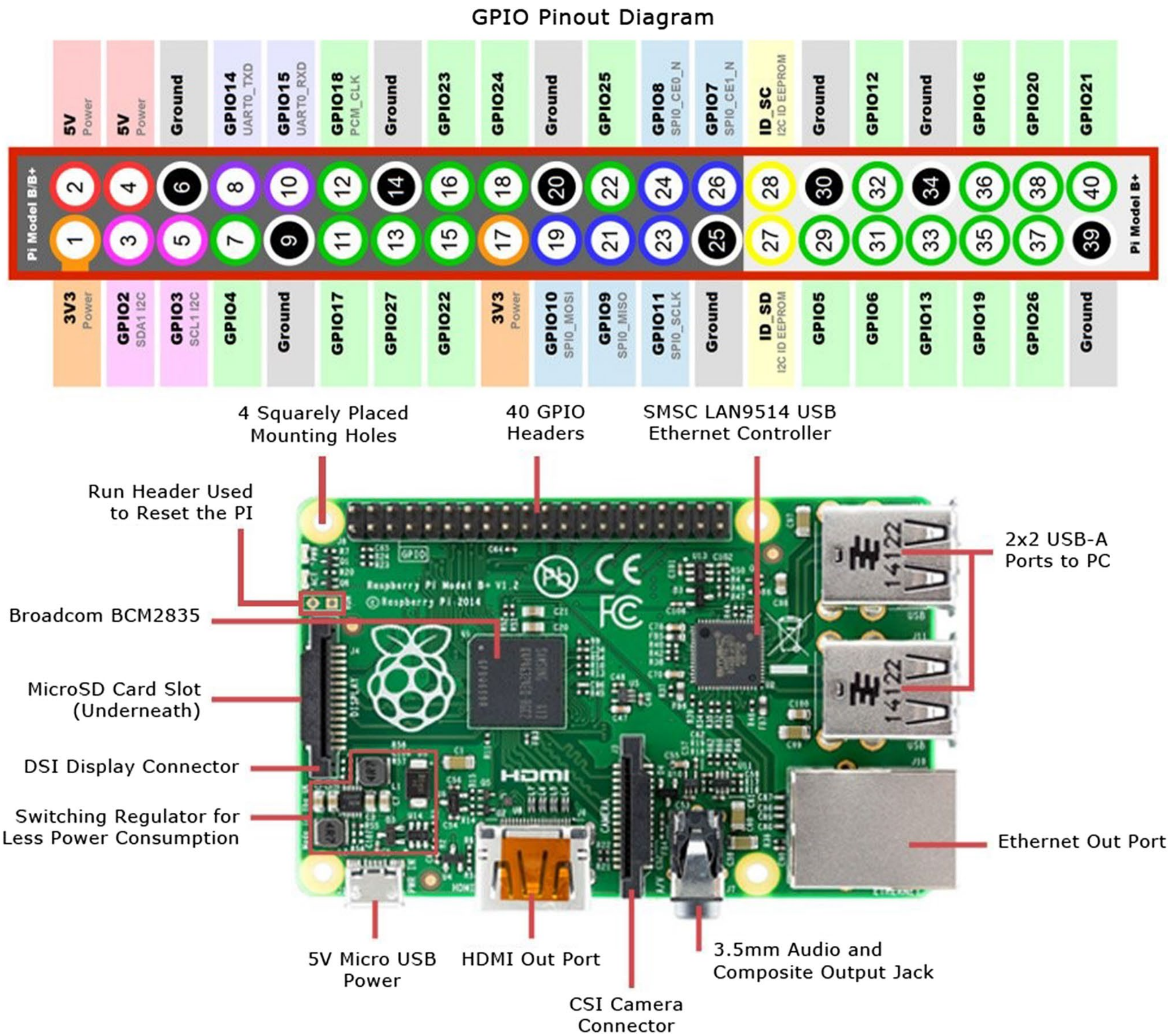


Fig. 3 Raspberry Pi model 3B+ embedded computer

we set up a Bluetooth module HC05 for an ID address of the Neurosky sensor by configuring AT commands. That means, this bluetooth module can only connect to the Neurosky sensor if it detects that the brainwave sensor is on, while it cannot be connected to any other device. This operation also avoids interference and loss of connection during data transmission. As soon as the microcontroller receives a signal from the Bluetooth module, some filters and data partitioning algorithm is applied to get two values: Attention Level and Meditation Level. These two data values stand for the concentration and the relaxation of our brain, respectively.

2.2 Hardware Components

Table 1 lists all main components for the prototype of our robotic arm. The total cost is about $500, which is considered as quite low for an academic robotic arm with multiple functions including mind-control. The cost may be lower if we reduced some components such as the extra Arduino module or RF module for gesture control.

The schematic of main hardware circuits is shown in Fig. 4. Figure 4a displays the hardware circuit for the transmitter side, which consists of the control gloves, Arduino Nano board, and RF module. A LCD is used to display the parameters from the Flex sensors and MPU6050 module, which allows us to recognize the errors in the design and fix them. Figure 4b presents the schematic of the receiver

Table 1 Bill of materials

Component	Block	Quantity	Cost (USD)
RC Lipo battery 3S	Power supply	1	11.42
USB voltage divider 5V 3A	Power supply	1	1.22
5V DC power supply	Power supply	1	8.56
Arduino nano	Transmitter controller	1	5.43
Flex sensor	Gesture control	5	82.55
MPU6050	Gesture control	1	1.04
Leap motion device	Gesture control	1	99.91
Neurosky mindwave^{TM} Mobile 2 brain-wave sensing headset	Mind control	1	152.03
Bluetooth module HC05	Bluetooth transmission	2	8.25
RF module HC12	RF transmission	2	9.56
Raspberry Pi model 3B+	Web interface	1	70.15
USB camera	Web interface	1	6.52
USB microphone	Voice control	1	4.34
Mini speaker	Voice control	1	8.47
Arduino Uno R3	Central controller	1	5.65
LCD 20×4	Transmitter display	1	3.21
3D robotic arm	Actuator	1	39.09
Servo motor MG996R	Actuator	7	21.28
Total	–	–	542.68

side, i.e. the central controller and the actuators. The servo motors, which include two at the thumb, one at each remaining finger and one at the elbow, are used to control the rotation of the arm. Three single LEDs are used to indicate the active control mode of the controller, i.e. either mind, voice, or gesture control.

3 Control Algorithms and Protocols

In this section, the protocol for control the operation of the robotic arm is presented.

3.1 Control Algorithms and Protocols for Gesture, Voice, and Web-Based Control

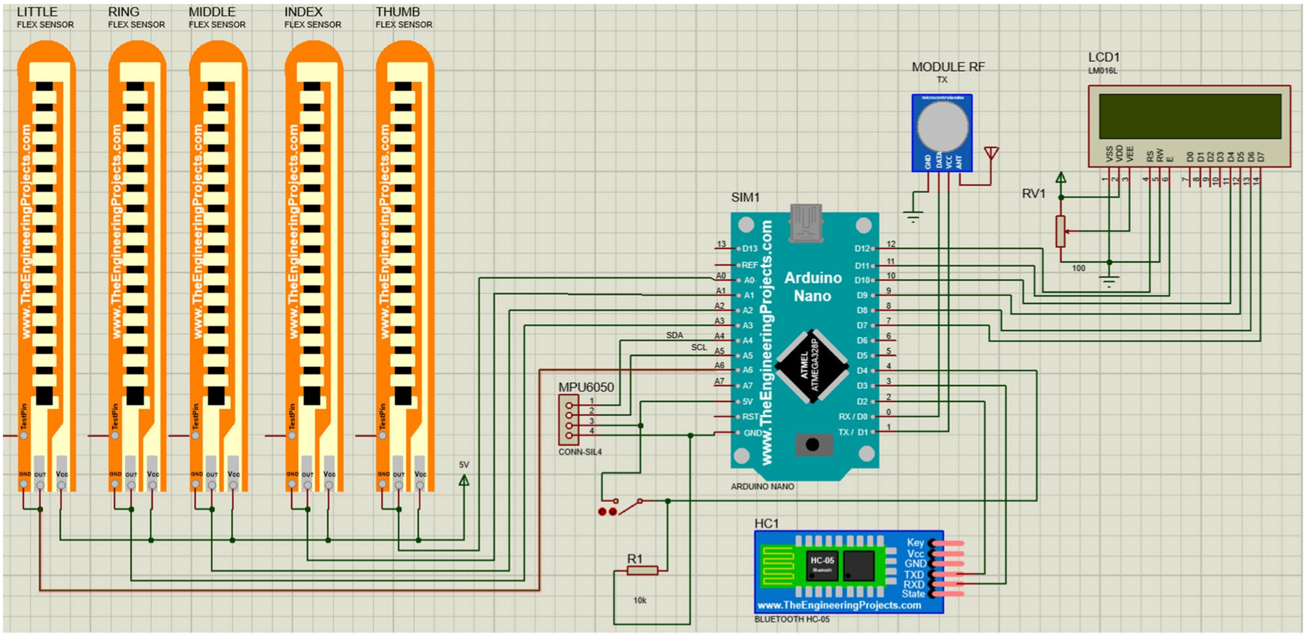
Let's start with the gesture control function by using Flex sensors and MPU6050. At the transmitter side, an Arduino Nano board is used to handle the sensing data. There is a pushbutton to select the source of input: from the "smart gloves" or from the voice commands. If the pushbutton is in mode 1, i.e. voice input, Arduino Nano reads the data from the Bluetooth module that is connected to Arduino and continuously receives voice data from smartphone. Then the Arduino processes the received data and send it to the RF module for transmission to the central controller. If the pushbutton is in mode 2, i.e. "gloves" mode, the Arduino module reads the analog values from the sensors, and converts it to a word of 10 bits, which expresses a rotation angle in the range from 0 to 180°. Then the data will be embedded into a message with the following format:

$< finger\_name >< rotation\_angle >< . >$.

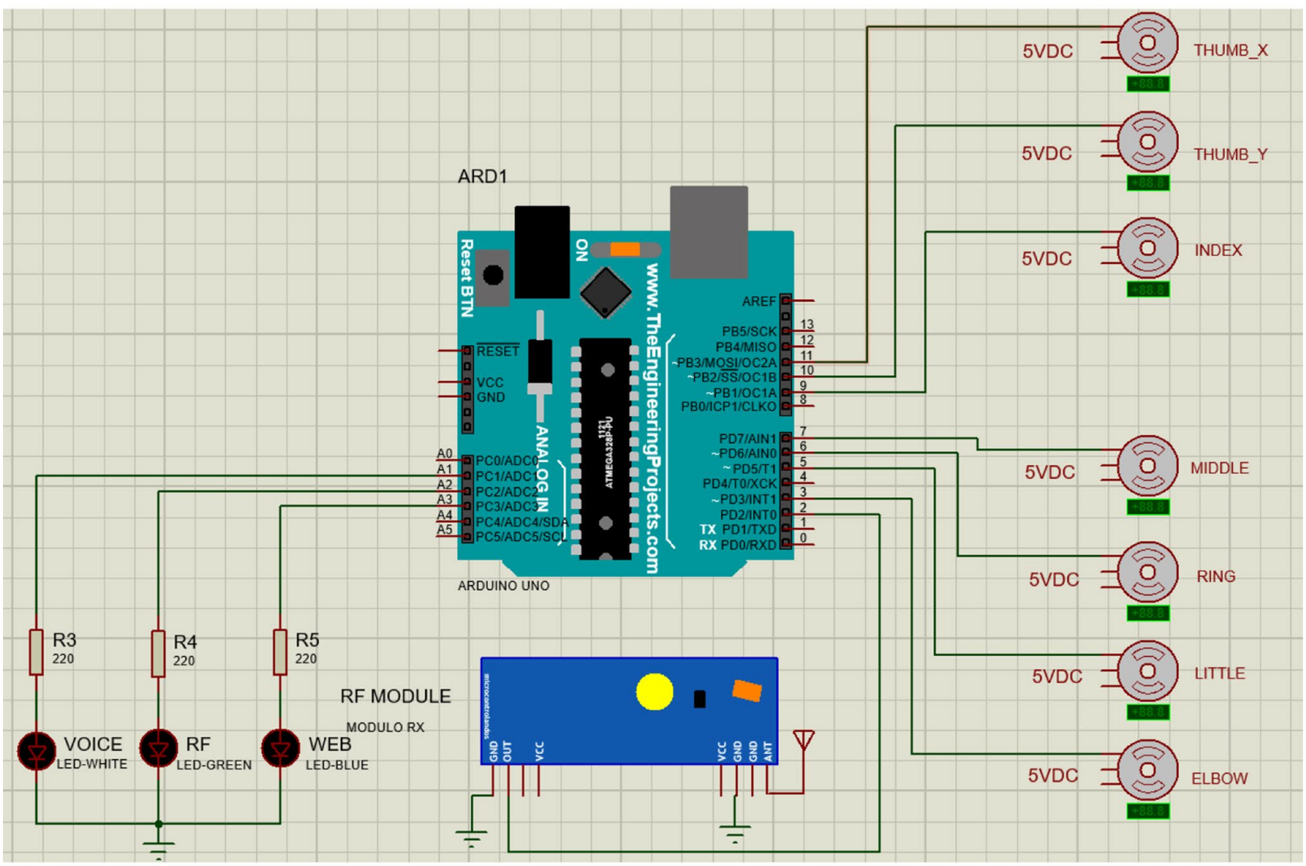
This message is then transmitted to the central controller by the help of RF module. The information in this message is also displayed on an LCD at transmitter side.

At the receiver side, the Arduino Uno module is able to process the received message and identify the command source. If the format of the received data consists of finger (or elbow) information and corresponding values, the "GLOVES" LED is turn on. Then the message is decoded to retrieve the neccessary data for controlling the servo motors.

Secondly, the algorithm to implement the gesture control using Leap Motion device is presented in Algorithm 1 below. At first, we need to check the connectivity of Leap Motion device to the laptop. After plugging the USB cable, the user needs to put the hands above the device about 15 – 20 cm. If the image of the hands appear on the screen, and the motion of the images on the screen follows the real motion of the human hands, the connectivity is good. In that case, the data of hands and fingers will also be displayed on the graphical user interface of Processing software. For the next step, with the support of Leap Motion library, the Processing software can determine whether the control hand is the left hand or righthand, and correct the *handRoll* value (a value that reflects the tilt angle when we rotate our hands horizontally) to adapt to the correct hand. Finally, an ID assignment algorithm provided by Processing is executed to assign an appropriate ID for each finger. Specifically, if the



(a) Transmitter



(b) Receiver

Fig. 4 Schematic of control hardware at transmitter and receiver sides software detects any finger that is in open position, it will return the ID with the value corresponding to the current position of that finger. The data is then embedded to a message including the finger name and an angle (in the range from 0° to 180°) and then sent to the central controller. The algorithm for the central controller is the same as the one applied for "gloves" mode.

As shown in Fig. 5a, we can see that the GUI shows the images of both hands recognized by Leap Motion, including information such as the names of the fingers, hand side, and displaying time. By using the ID assigning algorithm for each finger, we can determine which finger is raised or held. We actually receive the result in the bottom-left corner of the GUI as 11111, that means all five fingers are raised. If any finger is held, the corresponding bit position should become zero. Note that the thumb is corresponding to the left-most bit. As shown in Fig. 5b, we only let the left hand above LM device and have the thumb up, while the remaining fingers are held. According to that gesture, we get the return data as 10000, which means the first bit 1 indicates that the thump is up and the other 0 bits indicate the closed state of the remaining fingers.

Algorithm 1: Control the Leap Motion device

Result: Angle values of 5 fingers

begin
- Check connectivity of Leap Motion and computer via USB port;
- Display image and data of hand and fingers via Processing;
- **if** *Right hand* **then**
  - Redefine the handRoll value to correct the angle of the elbow to the right hand;
- **else**
  - Redefine the handRoll value to correct the angle of the elbow to the left hand;
- **end**
- Read the data of 5 fingers (open and closed) using ID assignment algorithm, convert these values into corresponding angle values and send them to the microcontroller.;

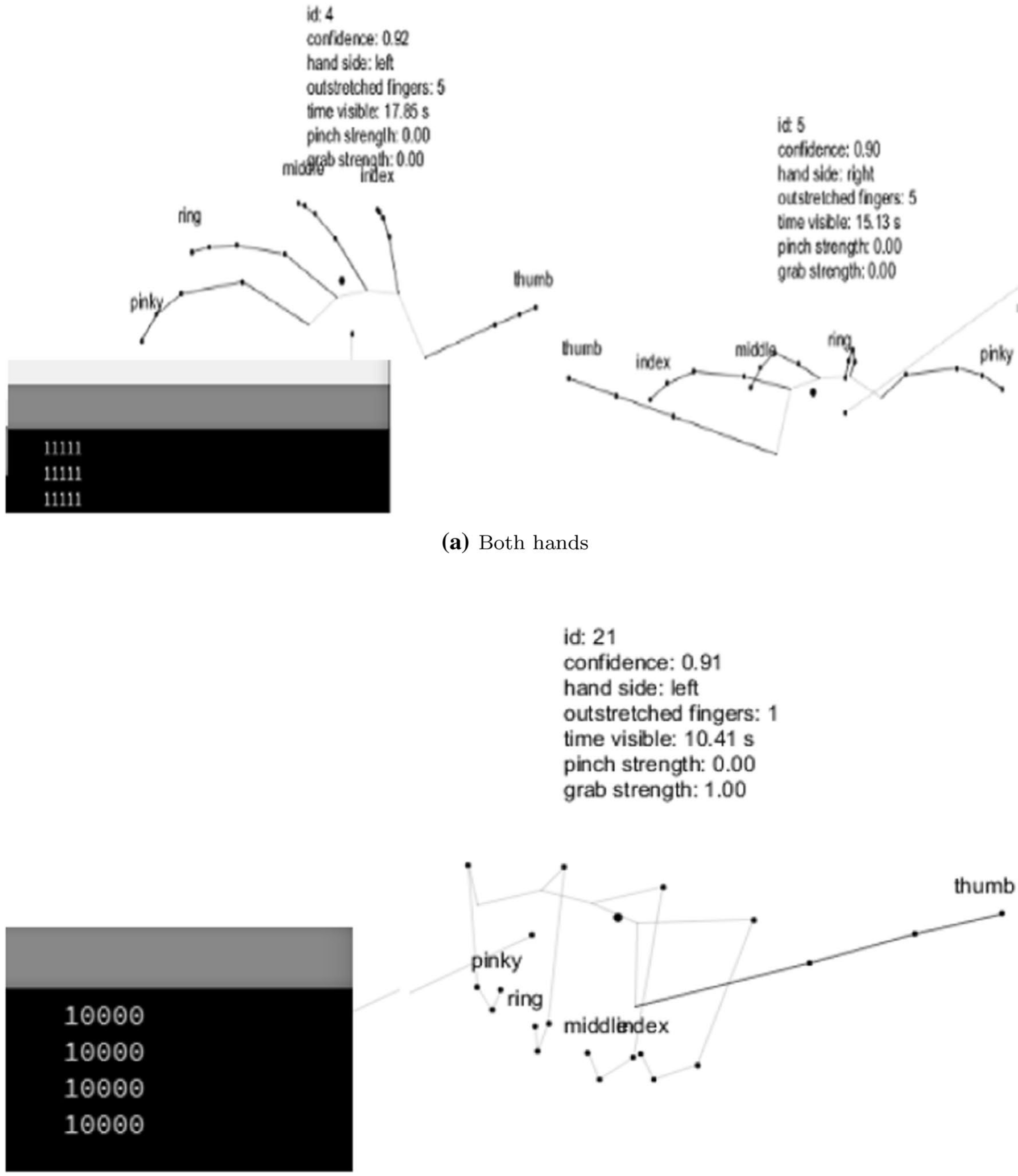
end

Thirdly, we consider the control protocol for web interface using Raspberry Pi. We use Node-Red, which can provide vast library of tools that can be used to visualize and handle data, to implement web services for Raspberry Pi. Node-RED is an open-source programming tool used for wiring together hardware devices, APIs, and online services smartly. It can be installed on a Linux-based platform, and it provides a browser-based editor that makes it easy to wire together flows using various nodes in the palette that can be deployed to its runtime [29].

Figure 6 illustrates the dashboard, i.e. the web interface, of the robotic arm that controls the rotation angles of all fingers as well as of the elbows from 0° to 180°. In addition, the dashboard also integrates a camera observation system to visualize the robotic arm when being remotely controlled, which may serve as a platform to set up image processing algorithms for the robot in the future.

Raspberry Pi is an embedded computer and quite expensive compared to other circuit components. Moreover, it has some limitations, for example, it cannot read/write analog data. So operations such as PWM cannot be implemented unless an ADC converter is integreated with the Raspberry. Reversely, Arduino is a circuit board designed to control hardware-intensive devices, so it can complement for the drawbacks of Raspberry Pi. However, to build an IoT web server with Arduino is not as simple as using Raspberry Pi, which possesses important communication features of a computer like Wi-Fi, Bluetooth, USB port, etc. to easily transmit and receive data from the Internet. Being aware of the advantages and disadvantages of these two modules, we make them work together via a serial protocol with the purpose of eliminating the disadvantages of both. With this connection, the web-input data on Node-Red can be sent perfectly to Arduino for processing. At the central controller (Arduino Uno), once it receives the data from Raspberry Pi, it can detach the necessary information and output the proper commands for the servo motors, while the LED indicating web interface mode is turn on.

Voice control protocol is implemented by using Google Assistant, which is developed by Google and integrated on almost smart phones. This application, which integrates artificial intelligence algorithms, allows us to talk to Google



(a) Both hands

(b) Left hand only

Fig. 5 Data from leap motion via processing

like a real person to help us answer our questions. For those who are tech-savvies, GA can be used to control devices in a house, robots, cars, etc. The flow chart of voice control protocol using GA is illustrated in Fig. 7.

In recent years, IF-This-Then-That (IFTTT) services, which allow users to create workflows with "triggers" and "actions" by using Web Application Programming Interfaces (APIs), are becoming more and more popular [30]. The first step in voice control process is to configure the commands from GA on IFTTT to trigger some actions corresponding to each command. The phrase "If This Then That" means that if an event occurs, this service will execute an event that follows, respectively. For example, we can create an event when I say "Open your hand", IFTTT will create a feedback event from Google that is "The hand robotic is fully opened" as shown in Fig. 8a.

Webhook is used as a data transfer station from GA to Node-red with the information of the command configured

Fig. 6 Web user interface for controlling the robotic arm

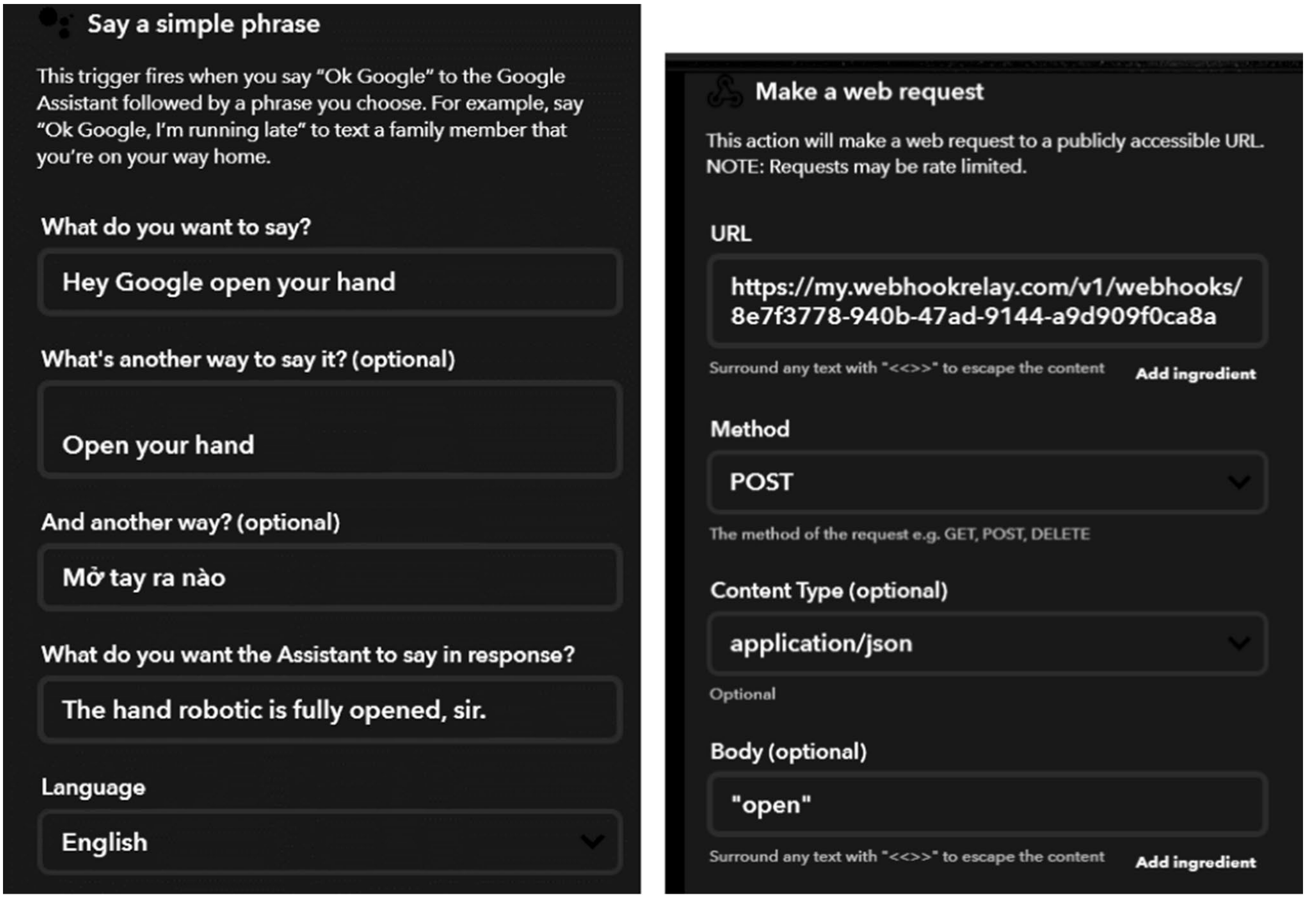
Fig. 7 Flow diagram of voice control protocol

on IFTTT. Please notice the URL box in Fig. 8b. This is the Webhook's address that both IFTTT and Node-red will base on to exchange data. In our design, the POST method is used, which applies only one-way transfer from Webhook to Node-red. The message "open" is encrypted by Webhook via an access key code (including the access key name and password). Only when the Node-red server has exact configuration with these contents, it would be able to read the data. This is a very good security measure of Webhook to avoid risks during data transmission on the Internet.

Figure 9 is shown as the process to check if the Google Assistant function is working properly or not. We see that when Google Assistant receives the "Open your hand" command, IFTTT and Webhook will immediately work to send the "open" content, which was previously configured on Webhook. We can see the content "open" in the debug box on Node-red. As soon as Node-red gets the data, the remaining task is to transmit this information to the Arduino, which connects directly to the Raspberry via USB port. At the central controller unit, the Arduino Uno can identify the messages in voice control mode (either from Raspberry Pi or from the RF module), decode the messages, and send corresponding commands to the robotic arm.

3.2 Brainwave Control Protocol

Finally, this section introduces the protocols for mindwave control. For having a better understanding about how the Neurosky brainwave sensor works and the basic principles for determining Attention Level and Meditation Level data, let's introduce some background on EEG. The acquired EEG data includes an assortment of components, which are called brainwaves and can be moreover classified based on various parameters such as the waveform shape and frequency. These components don't have an individual presence and the emersion of a single component can be even higher than others depending on the subject mental status [31]. There are totally five brainwave frequencies including Gamma, Beta,



(a) Configuration of Google Assistant response on IFTTT

(b) Configuration of Webhook on IFTTT

Fig. 8 Configuration of Google Assistant and Webhook

Alpha, Theta, and Delta. Figure 10 shows the different brainwave frequencies and characteristics of each [32].

- Gamma wave, which frequency is from 31 to 100 Hz, is the highest brainwave frequency. In this state, one can experience a range of increased emotions, insight, and high level of information processing.
- Beta wave, which frequency is between 16 Hz and 30 Hz, represents our normal wakefulness. It is associated with high levels of alertness, logical thinking, concentration and problem solving.
- Alpha wave, which has frequency in the range 8–15 Hz, is a bit shorter in wavelength than Beta wave. Alpha represents the brain that is in a state of rest and relaxation. This type of brainwave is popular in people who are always relaxed and highly creative.
- Theta wave, which frequency is from 4 to 7 Hz, appears when the body is in a state of deep relaxation and meditation, shallow sleep or dreaming. This wave is associated with the unconscious, where the mind is capable of deep understanding, intuitively developing, physically and mentally united.
- Delta wave, which frequency varies from 0.1 to 3 Hz, has the lowest frequency but the highest amplitude. Delta brainwaves have long been used for healing, because a deep sleep is always needed for the body's regeneration and repair mechanisms.

The sensing data from the Neurosky brainwave sensor is transmitted on a packet basis. Each packet of data from the Neurosky brainwave sensor always follows the same structure to transmit data to other devices. By analyzing the data structure, we can extract the necessary data we need in this research. Figure 11 shows an overview of each part of the packet structure [33]. Each package includes Header, Payload and Checksum.

The Header structure contains a total of three bytes: two synchronized bytes (standard values are 0xAA and 0xAA) and one odd byte Payload Length. These two synchronized bytes ensure that any data packets with two synchronized bytes different from 0xAA will be discarded. Besides,

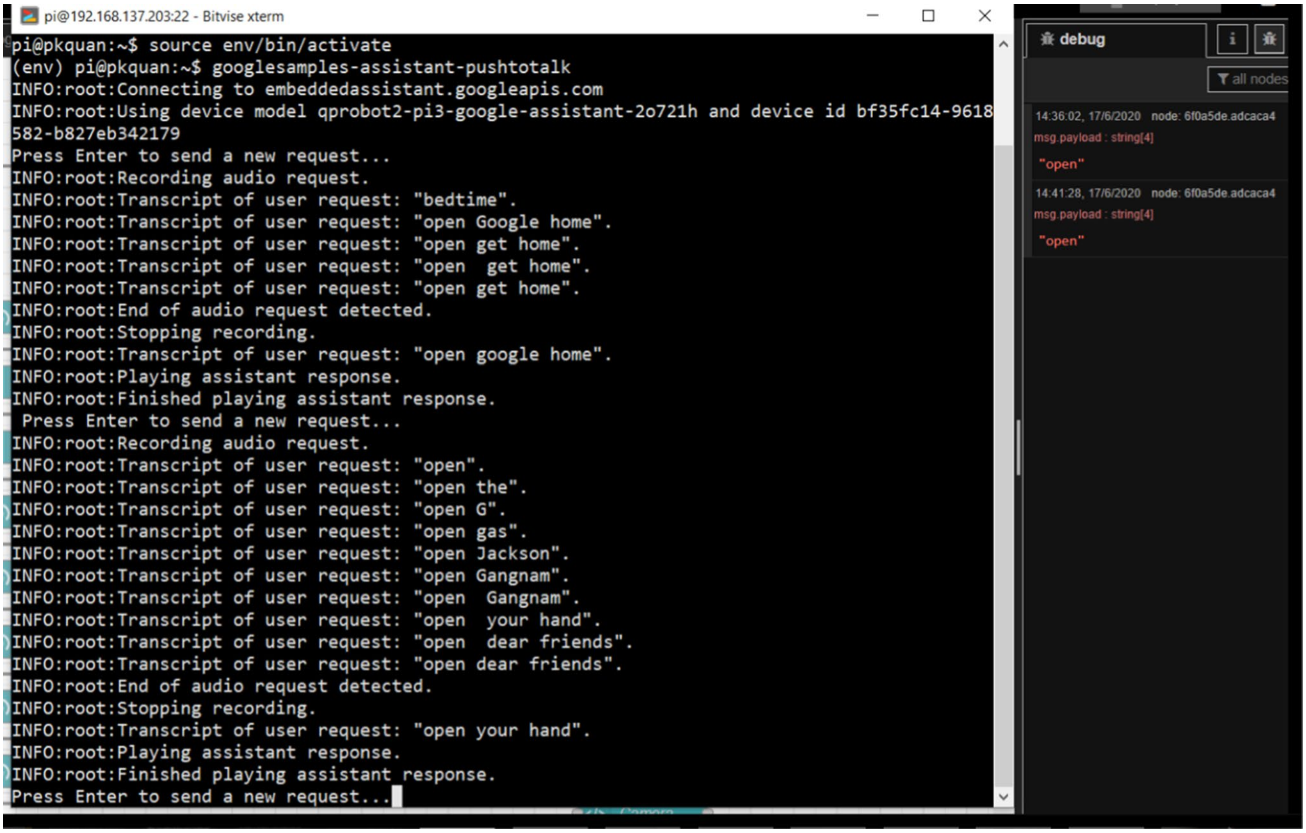


Fig. 9 Testing Google Assistant function on Raspberry Pi and Node-red

NeuroSky also uses Checksum byte at the end of the packet to detect the error data packets.

The Payload field in the data packet format is a sequence of bytes and the number of bytes is specified by the PLENGTH byte in the Header (the maximum value of this byte is 169 (decimal value) = 0xA9 (hexadecimal value)). The payload structure contains information about the data value, the length of the data value, and the position of bytes of the data value. The structure of payload is illustrated in Fig. 12:

- EXCODE (Extended code) has a hexadecimal value of 0x55 and is often used with CODE bytes to determine what type of value the Payload structure has.
- CODE byte indicates the data type of the payload. There are two main categories of data: single-byte and multi-byte types. If CODE byte ranges from 0x00 to 0x7F, it represents single-byte. In that case, there will be no VLENGTH byte. If CODE byte is greater than 0x7F, it represents a multi-byte type. In this case, VLENGTH byte indicates the length of data expressed in bytes. Multi-byte data type is used to transmit arrays of values or values that cannot fit in the size of a single byte.
- VALUE byte contains the hexadecimal value of the data, which type is determined by CODE byte.

Tables 2 and 3 show the possible values of CODE byte and the corresponding data type and the range of the VALUE byte for both single-byte and multi-byte cases.

Two data values that are used in this research are Attention Level and Meditation Level. In principle, Attention Level is extracted from Beta waveforms, while Meditation Level is extracted from Alpha waveforms. The algorithm for brainwave control of the robotic arm is presented as the Algorithm 2 below. First, the data packets from Neurosky are sent to the Arduino Uno board via Bluetooth modules. The packets that fail the Check Sum test will be eliminated, as well as the noise in data by applying some filtering algorithms. Then, three data values are extracted from the data packets: Attention Level, Meditation Level, and Blink Strength. In this work, the central controller mainly bases on the Attention Level to determine the proper commands to sent to the robotic arm.

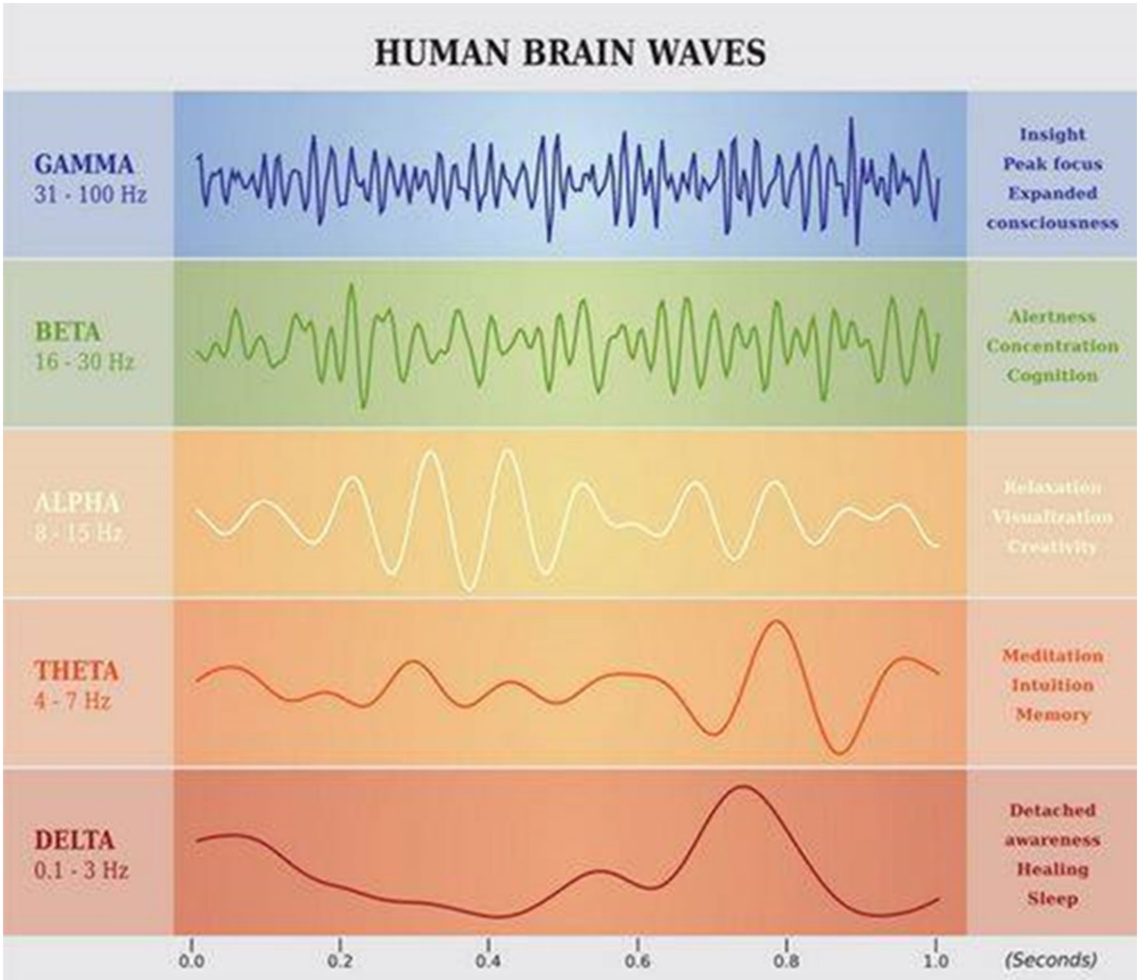


Fig. 10 Types of brain waves from a human brain

Fig. 11 Structure of Neurosky brainwave data

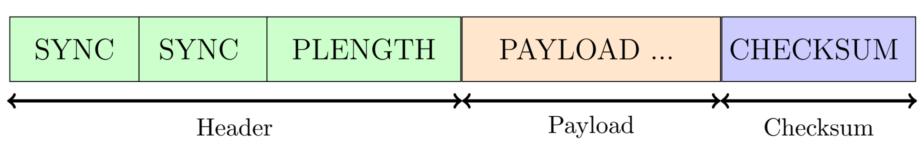


Fig. 12 Structure of payload data

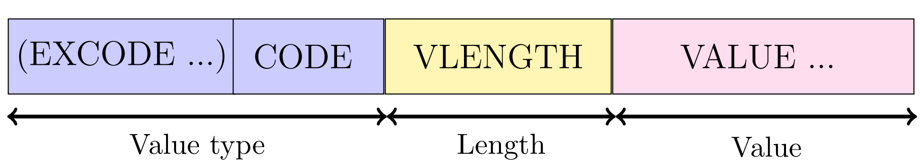


Table 2 CODE values for single-byte type

Extended code	Code	Vlength	Value
0	0x02	–	Poor signal (0–255)
0	0x04	–	Attention level (0–100)
0	0x05	–	Meditation level (0–100)
0	0x16	–	Blink strength (0–255)

Table 3 CODE values for muti-byte type

Extended code	Code	Vlength	Value
0	0x80	2	Raw wave value (– 32768 to 32767)
0	0x83	24	Power spectrum values of 5 brainwaves: delta, theta, alpha, beta, gamma.
Any	0x55	–	Byte reverse of [EXCODE]
Any	0xAA	–	Byte reverse of [SYNC]

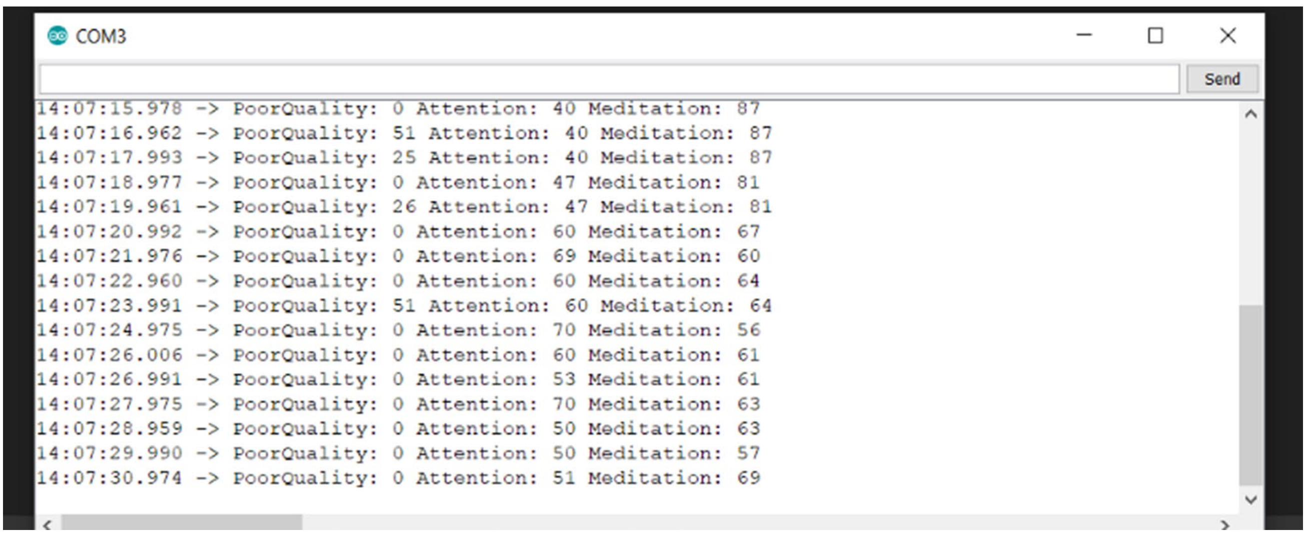


Fig. 13 Test results of attention, meditation data

Fig. 14 Robotic arm model in 3 modes: closed fingers, open fingers, and grasping an object

Algorithm 2: Control the Neurosky brainwave sensor

```
Result: Control command for robotic arm
begin
    Receive brainwave data from Neurosky sensor via connection with
      Bluetooth HC05;
    Apply filter to eliminate noise and failed packets based on Header and
      Checksum data structure;
    Extract Blink Strength, Attention Level, and Meditation Level based
      on data from the Payload structure;
    Embed the extracted data into microcontroller, mainly based on the
      Attention Level to control the robotic arm model;
end
```

Table 4 Evaluation of web interface functionality

No.	Control value	Received value	Accuracy (%)
1	Thumb, 100	Thumb, 100	100
2	Index, 0	Index, 0	100
3	Middle, 63	Middle, 63	100
4	Ring, 126	Ring, 126	100
5	Little, 180	Little, 180	100
6	Elbow, 90	Elbow, 90	100
7	Hand, 0	Hand, 0	100

Table 5 Evaluation of leap motion control functionality

No.	Control value	Robotic value	Accuracy (%)
1	00000	00000	100
2	10000	10000	100
3	11000	11000	100
4	11100	11100	100
5	11110	11110	100
6	11111	11111	100
7	01100	01100	100

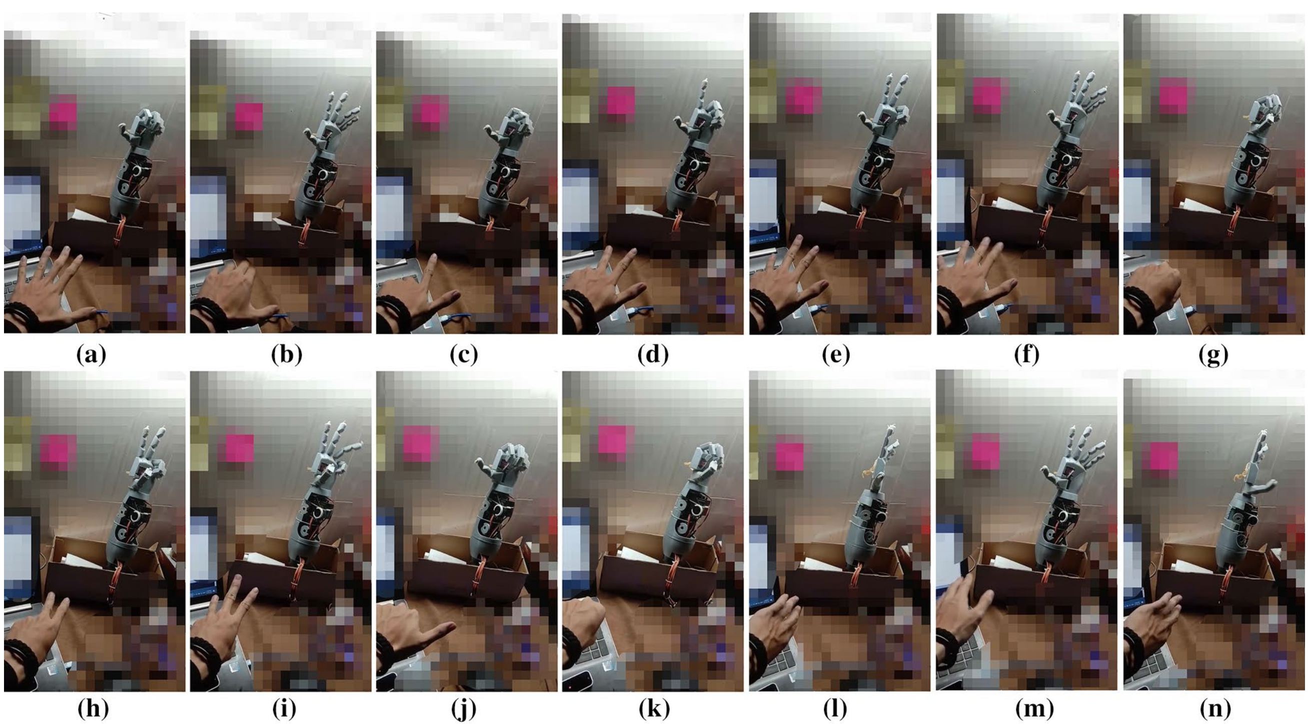


Fig. 15 Video frames of testing leap motion control

Table 6 Evaluation of voice control functionality

No.	Voice-controlled command	Robotic arm gesture	Accuracy (%)
1	Open	Open	100
2	Test	Test	100
3	Close	Close	100
4	Hi	Hi	100
5	Thumb	Thumb	100
6	Index	Index	100
7	Elbow	Elbow	100

Table 7 Evaluation of brainwave control functionality

Nos.	Mind-controlled command	Robotic arm gesture	Duration (s)	Accuracy (%)
1	Open hand	Close $\rightarrow$ open $\rightarrow$ close	< 3.5	45
2	Open hand	Open	> 5.9	79
3	Close hand	Close	> 10	100
4	Close hand	Close	< 8.3	83
5	Close hand	Open $\rightarrow$ close	< 5.6	56
6	Close hand	Close	< 3.7	37

As shown in Fig. 13, this is the result after applying all the filter algorithms and processing algorithms to extract the brainwave data, which are mentioned above. Attention data shows the concentration of the brain and Meditation data shows the relaxation of the brain.

4 Final Prototype and Testing Results

Figure 14 is a robotic arm model after fully assembling and connecting hardware components. The model consists of 7 servo motors including: 1 MG996R servo motor located at the thumb of the arm, 5 servo motors controlling 5 fingers, and 1 elbow control motor. Note that the thumb uses up to 2 motors for the X and Y axis, given that the range of the X-axis is from 0° to 75° and the range of the Y-axis is between 100° and 180°. Therefore, after receiving the value from the Leap Motion of the thumb, we only select the values within the appropriate range to be written into the engine. The model can move fingers as well as rotate the arm back and forth. Moreover, the model is currently able to hold a few small-size and mass objects, which fit the reach of the fingers as well as within the range of the drag of the servo such as phones, batteries, pliers, markers, etc.

4.1 Results of Web-Based Control

The testing of Web User Interface protocol shows that this function works very well. In addition, the web-based interface can also handle the sequence of control signals if the user changes the rotation value continuously in one touch. This helps the robotic model work smoothly with this GUI. The results is shown as Table 4.

4.2 Results of Leap Motion Control

The video frames for testing Leap Motion control function is shown in Fig. 15. It can be observed that the robotic arm responds accurately to the hand gestures of the user. Figure 15a–k illustrate the openning and closing of the robotic hand fingers, while the remainings show the rotation of the hand. In Fig. 15a–e, it can be seen that there is a slight delay between the user gesture and the motion of the robotic hand due to rapid change in the gesture, but in Fig. 15f–n, we can see that the robotic hand follows exactly the gestures of the intended user.

The accuracy of Leap Motion control is confirmed by the testing results in Table 5. Although the Leap Motion Control protocol works very well, it is still limited in close distance. In the future, if it is possible to use the RF band or integrate IoT to support long-distance signal transmission, the application of this function would be more efficient. Note that the control Value is 00000 means that all 5 fingers are closed; while 10000 means only the thumb is raised, while the remaining fingers are still closed.

4.3 Results of Voice Control

First we test the voice control function with a simple commad "test"—all servo motors take turn to rotate from 0° to 180°. This test is repeated at different distances, including 1 m, 5 m, 10 m, and 20 m, 10 times for each case. The result is perfectly good with the accurracy of 100%. After that, the testing is continued with the commands to control some specific motions of the fingers and the elbow of the robotic arm. The results are reported in Table 6, which the accuracy 100% for all cases.

Although the accuracy of voice control is perfect, there is a disadvantage of using GA capabilities on Raspberry Pi. Especially, it should depend on the language and the pronunciation of the user. Sometimes the texts recognized by GA and the actual spoken words are not the same. However, this problem can be fixed. During the process of installing GA on Raspberry, we conduct the Google account synchronization on devices using GA. In other words, we just need to use any smart phone and login into my Google account that have been set up before, then we can use smart phone to control the robotic arm anywhere in the world as long as Internet connection and account synchronization are available. Using

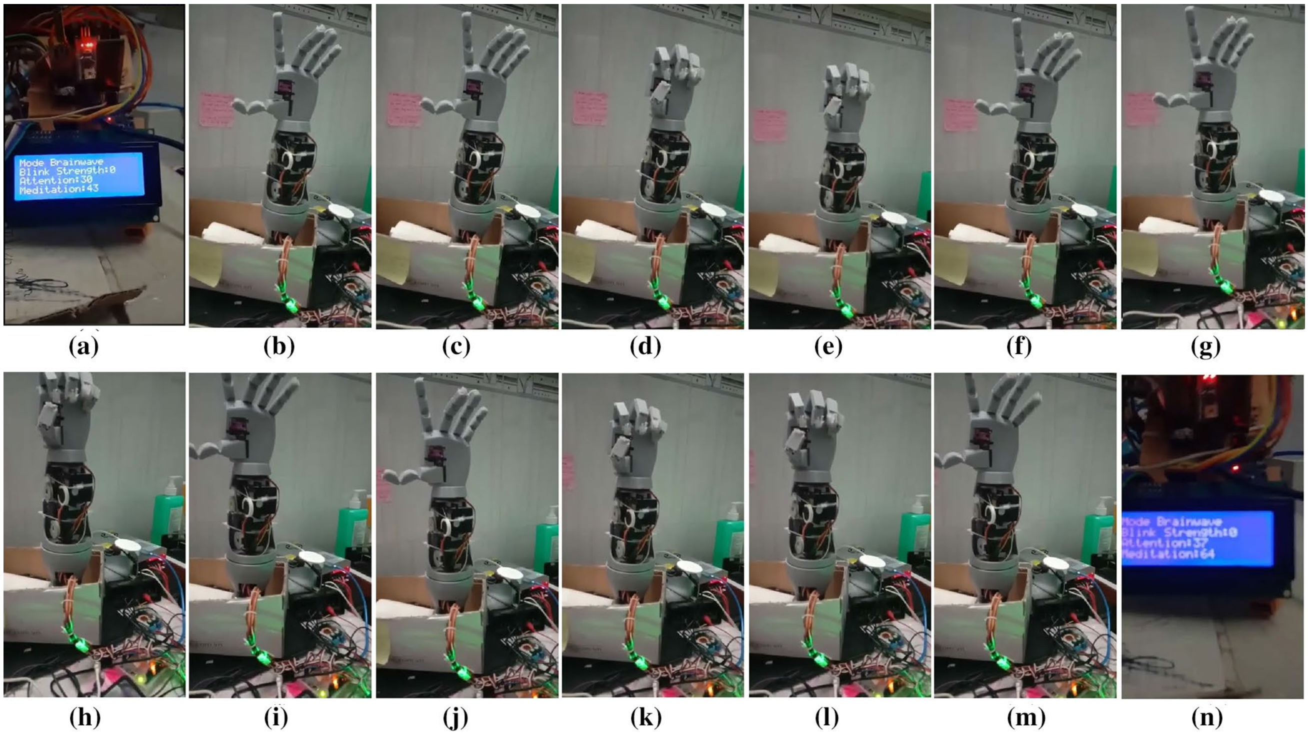


Fig. 16 Video frames of testing mind control function

a smart phone for GA instead of using Raspberry can fix the problem of unmatched command recognition on Raspberry Pi.

4.4 Results of Brainwave Control

For the brainwave control function, two motions of robotic hand are tested. The first motion is openning the hand, i.e. the fingers from the closed position stretch out to the open position. If the duration of data acquisition process is less than 3.5 s, the robotic hand can respond to the command with the accuracy of around 45 %. Howevever, if the data acqusition is longer, the accuracy can increase up to nearly 80 %. The second motion is the reverse of the first one, i.e. the hand moves from the open position to the “fist” position. Again the accuracy of control is improved when the duration of data retrieving process is increased. Especially, the accuracy can go up to nearly 100 % percent for the case that the data acquisition time is larger than 10 s. All results of the test are shown in Table 7. Images extracted from the video clip of this experiment are shown in Fig. 16.

It’s worth noting that although the accuracy can go up to nearly 100 % in “closed hand” test, this is not guaranteed for other motions of the robot. Furthermore, the results still depend on each specific user. However, with the accuracy up to 80 % in many cases, this mind control solution is simply too good for its price. Another disadvantage of our prototype is the battery problem. The brainwave sensor uses a 1.5 V battery, so the operating time is quite low, only about 10 minutes for one usage. This can be improved if we provide the system with DC power supply to continuously supply power to the sensor, but of course the cost will increase a little.

In terms of advantages, the Neurosky sensor works well on various operating systems such as Windows, Android, etc., and the return value is relatively accurate and stable. We certainly can increase the accuracy of brainwave control by applying the machine learning algorithms, which is our future work.

5 Conclusion

This research has successfully achieved all the predefined requirements. The most contribution of this work is a comprehensive solution to control a robotic arm model by using gesture, voice, web, and brainwave, but with a low cost. The accuracy of mind control can achieve up to 80 % in most cases and can go nearly 100 % in some specific case. Compared to previous works such as [31], our prototype achieves the same level of accuracy, which is too good for such a low-cost solution. Furthermore, if the data acquisition time is longer than 10 s, it can reach the accuracy of nearly 100 %, which improves 2 % compared to the work in [11] and 8 % compared to the work in [4]. This would be a big step forward in robot control applications at academic level, all the extended functions work well and stable, completely overcome the disadvantages of the old functions.

This work can be extended further in the future by considering the following approaches. Firstly, if artificial intelligence (AI) can be applied to this project like voice control, we can train the model to understand the sentences with similar meanings of the command. There is no need for us to remember exactly what command we have set up before. Robots can now learn, understand and follow commands. In addition, using AI to process data of brainwave frequencies, we can use our thinking to control the operation of this model with more gestures and having greater accuracy. Secondly, it is a natural idea to extend the mechanical parts of our prototype, including increasing the degree of freedom for the robotic arm so that it can ultimately function as similar to human arm as possible and then adding other parts to complete the full humanoid robot model as in [26].

Authors' contribution Conceptualization: QKP; Methodology: QKP, PTT; Formal analysis and investigation: QKP, TVV; Software: QKP, TVV; Validation: TVV, Writing-original draft preparation: PTT, QKP; Writing-review and editing: PTT; Resources: TVV; Supervision: PTT.

Funding No funds, grants, or other support was received.

Availability of data and material All data has been provided in this paper.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Code availability Not applicable.

References

1. Susko T, Swaminathan K, Krebs HI (2016) Mit-skywalker: a novel gait neurorehabilitation robot for stroke and cerebral palsy. IEEE Trans Neural Syst Rehabil Eng 24(10):1089–1099. https://doi.org/10.1109/TNSRE.2016.2533492
2. Tran P, Jeong S, Wolf SL, Desai JP (2020) Patient-specific, voice-controlled, robotic flexotendon glove-ii system for spinal cord injury. IEEE Robot Autom Lett 5(2):898–905. https://doi.org/10.1109/LRA.2020.2965900
3. Jackowski A, Gebhard M, Thietje R (2018) Head motion and head gesture-based robot control: a usability study. IEEE Trans Neural Syst Rehabil Eng 26(1):161–170. https://doi.org/10.1109/TNSRE.2017.2765362
4. Zhao J, Li W, Li M (2015) Comparative study of ssvep- and p300-based models for the telepresence control of humanoid robots. PLOS ONE 10(11):1–18. https://doi.org/10.1371/journal.pone.0142168
5. Ortiz-Rosario A, Adeli H (2013) Brain–computer interface technologies: from signal to action. Rev Neurosci 24(5):537–552. https://doi.org/10.1515/revneuro-2013-0032
6. Hirai K, Hirose M, Haikawa Y, Takenaka T (1998) The development of honda humanoid robot. In: Proceedings. 1998 IEEE international conference on robotics and automation (Cat. No.98CH36146), vol 2, pp 1321–1326 vol.2. 10.1109/ROBOT.1998.677288
7. Amiri S, Rabbi A, Azinfar L, Fazel-Rezai R (2013) A Review of P300, SSVEP, and hybrid P300/SSVEP brain–computer interface systems, IntechOpen. 10.5772/56135. https://app.dimensions.ai/details/publication/pub.1039292662 and https://www.intechopen.com/citation-pdf-url/44907
8. Mao X, Li M, Li W, Niu L, Xian B, Zeng M, Chen G (2017) Progress in eeg-based brain robot interaction systems. Compit Int Neurosci 2017:1742862
9. Chapin JK, Moxon KA, Markowitz RS, Nicolelis MA (1999) Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. Nat Neurosci 2:664–670
10. Hochberg LR, Bacher D, Jarosiewicz B, Masse NY, Simeral JD, Vogel J, Haddadin S, Liu J, Cash SS, van der Smagt P, Donoghue JP (2012) Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. Nature 485:372–375
11. Zhang Z, Huang Y, Chen S, Qu J, Pan X, Yu T, Li Y (2017) An intention-driven semi-autonomous intelligent robotic system for drinking. Front Neurorobot. https://doi.org/10.3389/fnbot.2017.00048
12. Reust A, Desai J, Gomez L (2018) Extracting motor imagery features to control two robotic hands. In: 2018 IEEE international symposium on signal processing and information technology (ISSPIT), pp 118–122, 10.1109/ISSPIT.2018.8642627
13. Bouyarmane K, Vaillant J, Sugimoto N, Keith F, Furukawa JI, Morimoto J (2014) Brain-machine interfacing control of whole-body humanoid motion. Front Syst Neurosci 8(138):001–010. https://doi.org/10.3389/fnsys.2014.00138
14. Tidoni E, Gergondet P, Kheddar A, Aglioti SM (2014) Audio-visual feedback improves the bci performance in the navigational control of a humanoid robot. Front Neurorobot 8:20. https://doi.org/10.3389/fnbot.2014.00020
15. Zhao J, Li W, Mao X, Hu H, Niu L, Chen G (2017) Behavior-based ssvep hierarchical architecture for telepresence control of humanoid robot to achieve full-body movement. IEEE Trans Cognit Dev Syst 9(2):197–209. https://doi.org/10.1109/TCDS.2016.2541162
16. Mao X, Li W, Lei C, Jin J, Duan F, Chen S (2019) A brain–robot interaction system by fusing human and machine intelligence. IEEE Trans Neural Syst Rehabil Eng 27(3):533–542. https://doi.org/10.1109/TNSRE.2019.2897323
17. Chandrasekaran G, Karthikeyan P, Kumar NS, Kumarasamy V (2021) Test scheduling of system-on-chip using dragonfly and ant lion optimization algorithms. J Intell Fuzzy Syst 40(3):4905–4917. https://doi.org/10.3233/JIFS-201691
18. Chandrasekaran G, Periyasamy S, Panjappagounder Rajamanickam K (2020) Minimization of test time in system on chip using artificial intelligence-based test scheduling techniques. Neural Comput Appl 32:5303–5312. https://doi.org/10.1007/s00521-019-04039-6
19. Sheppard D, Felker N, Schmalzel J (2019) Development of voice commands in digital signage for improved indoor navigation using google assistant sdk. In: 2019 IEEE sensors applications symposium (SAS), pp 1–5, https://doi.org/10.1109/SAS.2019.8706120
20. Walugembe H, Phillips C, Requena-Carrion J, Timotijevic T (2020) Comparing dynamic hand rehabilitation gestures in leap motion using multi-dimensional dynamic time warping. IEEE Sens J 1:21. https://doi.org/10.1109/JSEN.2020.3047268

21. Enikeev D, Mustafina S (2020) Recognition of sign language using leap motion controller data. In: 2020 2nd international conference on control systems, mathematical modeling, automation and energy efficiency (SUMMA), pp 393–397. https://doi.org/10.1109/SUMMA50634.2020.9280795
22. Sawangjai P, Hompoonsup S, Leelaarporn P, Kongwudhikunakorn S, Wilaiprasitporn T (2020) Consumer grade eeg measuring sensors as research tools: a review. IEEE Sens J 20(8):3996–4024. https://doi.org/10.1109/JSEN.2019.2962874
23. Avudaiammal R, Mystica KJ, Balaji A, Raja B (2020) Brain sense controlled wireless robot: interfacing neurosky brainsense to a wheelchair prototype. In: *2020 third international conference on smart systems and inventive technology (ICSSIT)*, pp 276–280. https://doi.org/10.1109/ICSSIT48917.2020.9214100
24. Lancheros-Cuesta DJ, Arias JLR, Forero YY, Duran AC (2018) Evaluation of e-learning activities with neurosky mindwave eeg. In: 2018 13th Iberian conference on information systems and technologies (CISTI), pp 1–6. https://doi.org/10.23919/CISTI.2018.8399316
25. Chaipakornwong T, Djohan, Chang S, Sittiprapaporn P (2019) Brain exercise in elderly people by neurosky smarter kit. In: 2019 16th international conference on electrical engineering/electronics, computer, telecommunications and information technology (ECTI-CON), pp 490–493. https://doi.org/10.1109/ECTI-CON47248.2019.8955239
26. Gross R (2018) Humanoid robotic torso PROTO1. https://www.myminifactory.com/es/object/3d-print-humanoid-robotic-torso-proto1-48754
27. Niechwiej-Szwedo E, Gonzalez D, Nouredanesh M, Tung J (2018) Evaluation of the leap motion controller during the performance of visually-guided upper limb movements. PLOS ONE 13(3):e0193639. https://doi.org/10.1371/journal.pone.0193639
28. Technologies NBCI (2018) Neurosky mindwave mobile 2 transition doc. https://www.robotshop.com/media/files/pdf/datasheet-80013-001.pdf
29. Zare A, Iqbal MT (2020) Low-cost esp32, raspberry pi, node-red, and mqtt protocol based scada system. In: 2020 IEEE international IOT, electronics and mechatronics conference (IEMTRONICS), pp 1–5. https://doi.org/10.1109/IEMTRONICS51293.2020.9216412
30. Liu L, Bahrami M, Chen WP (2020) Automatic generation of ifttt mashup infrastructures. In: 2020 35th IEEE/ACM international conference on automated software engineering (ASE), pp 1179–1183
31. Ali HA, Goga N, Marian CV, Ali LA (2020) An investigation of mind-controlled prosthetic arm intelligent system. In: The 16th international scientific conference elearning and software for education, pp 17–26. https://doi.org/10.12753/2066-026X-20-088
32. Muse (2018) A deep dive into brainwaves: brainwave frequencies explained. https://choosemuse.com/blog/a-deep-dive-into-brainwaves-brainwave-frequencies-explained-2/
33. Program ND (2017) Thinkgear serial stream guide. http://developer.neurosky.com/docs/doku.php?id=thinkgear_communications_protocol

Quan K. Pham was born in Ho Chi Minh City, Vietnam. He received B.Eng. electronics and telecommunications engineering from Ton Duc Thang University, Ho Chi Minh City, Vietnam in 2020, where he was working as a student research assistant of Wireless Communications Research Group, under the supervision of Dr. Phuong T. Tran. He is currently pursuing his M.S. degree in information and communication engineering at Chungbuk National University, South Korea. His major interests include robotics, embedded system design, and IoT applications.

Truong V. Vo was born in Ho Chi Minh City, Vietnam. He is currently an undergraduate student in control and automation engineering at Ton Duc Thang University, Ho Chi Minh City, Vietnam, where he is also working as a student research assistant of Wireless Communications Research Group, under the supervision of Dr. Phuong T. Tran. His major interests include embedded system design, IoT applications, industrial automation and digital data management.

Phuong T. Tran was born at Ho Chi Minh City, Vietnam. He received B.Eng. and M.Eng degrees in electrical engineering from Ho Chi Minh University of Technology, Ho Chi Minh City, Vietnam in 2002 and 2005, respectively. In 2007, he became a Vietnam Education Foundation Fellow at Purdue University, U.S.A., where he received his M.S. degree in mathematics and Ph.D. degree in electrical and computer engineering in 2013. In 2013, he joined the Faculty of Electrical and Electronics Engineering of Ton Duc Thang University, Vietnam and served as the Vice Dean of Faculty since October 2014. He is currently a Senior Member of IEEE. His major interests are in the area of wireless communications and network information theory.