Project Report on

"Wireless Dexterous Anthropomorphic, Gesture Controlled Robotic Arm"

ABSTRACT

Gesture Controlled Robotics stands for using anthropomorphic movements as input signals to control the robot dynamics. Generally, robot arms can be of various end effectors, each having its significances. This project plans to develop a robotic arm with a better Degree of Freedom and grasping abilities, which acts in a manner depending on the fingers and hand gestures from a distance. The wireless hand gesture-controlled robotic arm is a wireless system that can convey the information between two or more points, and any electrical conductors do not physically connect it. It will share movement information in real-time, from human hand gestures to the robotic arm wirelessly. The objective is to ease the robotic arm's control with hand gestures using flex bend sensors and servo motors for humanitarian aid purposes and even in the prosthesis.

Two- Arduino NANO (ATMEGA328) are used, one for the human hand and the other on the robotic arm as the microcontroller in this project. It is easy to change/modify, and further, the programming language of this microcontroller is more familiar than other microcontrollers. The pair of the NRF24L01 transceiver module is used for wireless communication between the Arduino NANO (ATMEGA328) microcontrollers.

Ultimately, the proposed ideology is meant to resemble the human arm's size and physical abilities to function in human-oriented environments and work through gestures mainly, but to pose no physical threat to humans.

Keywords— Anthropomorphic; End effectors; Degree of Freedom; Arduino NANO (ATMEGA328); NRF24LO1;

Table of Contents:

Table of Contents: ... iii

List of Figures: ... iv

List of Tables: ... v

1. Introduction ... 1

1.1 Scope of the project ... 2

1.2 Problem Statement ... 3

2. Literature Survey ... 5

3. Objective ... 9

3.1 Scope of objective: ... 9

4. Methodology ... 11

4.1 Interconnections between the components: ... 11

4.2 Transmitter - Flex Sensor configuration: ... 12

4.3 Receiver - Servo Motor configuration: ... 14

4.4 Hardware Implementation: ... 16

5. Hardware/ Software Specifications ... 18

5.1 Hardware Specification ... 18

5.1.1 NRF24L01+ SI24R1 ... 18

5.1.2 Microcontroller ATmega328 ... 20

5.1.3 MG995R Digital Servo ... 20

5.1.4 Flex Sensor ... 22

5.2 Software Specification ... 22

5.2.1 Why is Arduino advantageous? ... 24

5.3 Software Program ... 25

6. Experimental Results and Discussion ... 28

7. Conclusion ... 32

8. Future Implications ... 33

Reference ... 34

List of Figures:

Figure 1.1. This shows how simple hand gestures can be implemented for various tasks. 4
Figure 4.1. Block diagram of the system 11
Figure 4.2. Block diagram of the transmitter system 12
Figure 4.3. A brief flowchart of the Transmitter system 13
Figure 4.4. Block diagram of the receiver system 14
Figure 4.5. A brief flowchart of the Receiver system 15
Figure 4.6. (a)Transmitter system hardware prototype along with (b)Receiver model 16
Figure 4.7. Communication system hardware prototype of (Tx)& (Rx) 17
Figure 5.1. NRF24L01+ SI24R1 19
Figure 5.2. Microcontroller ATmega328 20
Figure 5.3. MG995R Digital Servo 21
Figure 5.4. Flex Sensor 22
Figure 5.5. Basic structure view of Arduino ID 23
Figure 6.1. All fingers unflexed 29
Figure 6.2. Grip open 29
Figure 6.3. Finger movement 29
Figure 6.4. Respective mimic of finger gesture 29
Figure 6.5. Grasping action for different objects 30
Figure 6.6. Grasping action performed for the unknown object 30
Figure 6.7. All fingers flexed 30
Figure 6.8. Grip Closed 30
Figure 6.9. An overall view of the project components making and assembling 31

List of Tables:

Table 3.1. RELATION OF PARTS OF HUMAN HAND (TX) CONTROLLING RESPECTIVE PARTS OF THE ROBOTIC ARM (RX). 9
Table 4.1. VARIOUS DIFFERENT OBSERVATIONS OF INDIVIDUAL FINGERS (Tx). 14
Table 4.2. VARIOUS DIFFERENT OBSERVATIONS OF INDIVIDUAL FINGERS (Rx). 16
Table 5.1. TRANSMITTER SYSTEM PSEUDO CODE 25
Table 5.2. RECEIVER SYSTEM PSEUDO CODE 26

1. Introduction

Humanoid robots are designed and built to mimic human form and movement. The number of degrees of freedom endowed in any robotic arm is sufficient for executing a wide range of tasks. With the latest technologies nowadays, the interaction between the human and the automated system can be done in a wired system using a wireless system. With the integration of wireless communication over the wired communication, the benefit is significantly increased in the robotic system's mobility, and there are no cables to mess up the system, cheap and controllable from a distance to handle tasks. The mechanical design is used in many situations and for a lot of purposes. They are used in dangerous environments such as manufacturing processes, military purposes, and even where humans cannot survive, like in space. For centuries, humans have been fascinated by the idea of robotics and automation. Until today, universities, private companies, and military powerhouses work tirelessly to create human-like machines to do everyday tasks and maintain absolute precision and provide unique entertainment.

Over the last years, the field of robot hand design has received increased attention. Anthropomorphic characteristics (e.g., appearance, links lengths), lightweight, low-cost, and flexible materials, and synergistic actuation are some of the current trends. The interest, as mentioned earlier, is motivated by the fact that robot hands can be used for several everyday life applications, from myoelectric prostheses and teleoperation to human-robot interaction and humanoids. How can we define "anthropomorphism"? Is it possible to discriminate if a robot hand is more anthropomorphic than another? How can anthropomorphism be helpful? Those are some of the fundamental questions that will be raised and addressed in the upcoming future. Over the last years, a lot of effort has been put into building fully-actuated, highly dexterous, multi-fingered robot hands. But these hands require multiple motors for the various Degrees of Freedom (DoF), sophisticated sensing elements and electronics, and complicated software and control laws. Thus, they have increased complexity, weight, and cost. Thus, increasing system skill can be accomplished either by adding hand complexity or arm kinematic redundancy.

Since the beginnings of robotics, mapping human to robot motion was necessary for a series of applications ranging from teleoperation and telemanipulation studies to closed-loop, anthropomorphic grasp planning. Anthropomorphic robot motion extraction is useful for robots that collaborate, interact, and co-exist with humans in dynamic or human-centric environments.

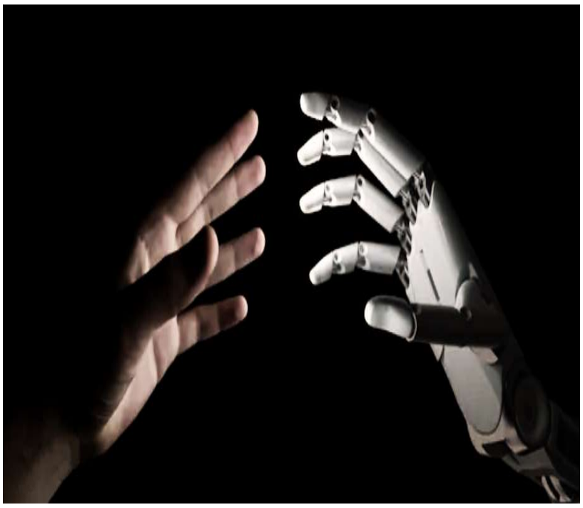
Anthropomorphism is derived from the Greek words Anthropos (human) and morphe (form). A robot may be characterized as anthropomorphic or human-like if it mimics the human condition. The primary purpose of anthropomorphism is "to imbue the imagined or real behavior of nonhuman agents with human-like characteristics, motivations, intentions, and emotions." Regarding the different classes of anthropomorphism, a clear distinction between functional and perceptional anthropomorphism.

Functional anthropomorphism has its first priority to guarantee the execution of a specific functionality in task-space and only after accomplishing such a prerequisite to optimizing anthropomorphism of structure (minimizing a "distance" between the human and robot poses). This project intends to implement human-like machine movements by controlling a real-time robotic arm in a wireless system using a pair of the NRF24L01 transceiver module. The mechanical arm movement is done by servo motors positioned on each finger on the robotic arm that rotates according to a person's hand gesture detected by flex bend sensors placed on each finger embedded on the glove. Simultaneously, the NRF24L01 transceiver module acts as the transmitter and receiver module for the robotic arm system and the human hand (glove), respectively. The NRF24L01 transceiver module can communicate with each other over a distance of 100 meters, allowing users to handle various tasks from a distance, thus making the wireless system dynamic. Furthermore, by using the NRF24L01 transceiver module, the user can control more than one particular system, meaning that the human hand gesture may contain a few robotic arms simultaneously and for the same movements.

1.1 Scope of the project

An important question is: why has anthropomorphism become significant and necessary? Nowadays, we experience an increasing demand for human-robot interaction (HRI) applications. It is believed that anthropomorphism of robot motion is essential in these applications. It increases safety in human and robot interactions and facilitates establishing a solid social connection between the human and the robots. More precisely, regarding the social relationship, the more human-like a robot is in terms of motion, appearance, expressions, and perceived intelligence, the more quickly it will manage to create meaningful "relationships" with human beings as robot-likeability is increased.

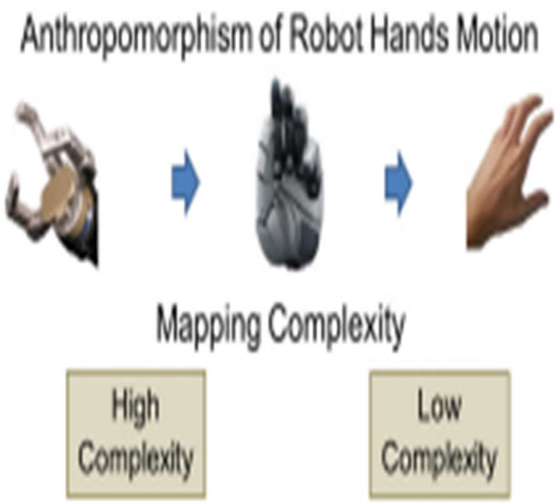
Regarding safety in HRI scenarios, when robots move anthropomorphically, users can more easily predict their motion and comply with their activity, thus avoiding injuries. **Gielniek et al.** support this idea, discussing in their work that: "human-like motion supports natural human-robot interaction by allowing the human user to more easily interpret movements of the robot in terms of goals. This is also called motion clarity." **Beetz et al**. first elaborated the idea of creating legible and predictable robot motions. In contrast, the concept of the legibility of robot motion goes back to **Alami et al**. **Dragan, and Srinivasa** proposed a methodology based on gradient optimization techniques for autonomously generating legible robot motion (e.g., the motion that communicates its intent to a human observer). This study's motivation comes from the fact that when humans can predict the outcome/goal of robot motions, they may comply with their movements and avoid injuries or enhance collaborations. Similarly, deriving anthropomorphic robot motions can be significant not only for aesthetic but also for practical reasons.



1.2 Problem Statement

In general, developing robotic hands with mechanisms has to be dealt with by using rigid palm configurations and articulated fingers. Frequently, this approach would restrict the workspace and give a very complex mechanical design for the fingers and the need to use many actuators to obtain adequate grasping ability.

Henceforth, in this project, it is planned to work on human hand gestures, which purely takes advantage of its foldable and flexile palm to achieve practical and versatile grasping of objects through an anthropomorphic robotic arm perform dexterous hand movements.



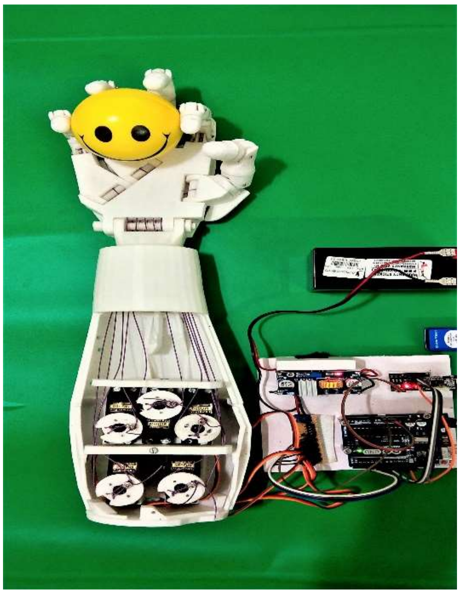


Figure 1.1. This shows how simple hand gestures can be implemented for various tasks.

2. Literature Survey

[1]. Minas Liarokapis and Aaron M. Dollar, "Combining Analytical Modeling and Learning to Simplify Dexterous Manipulation with Adaptive Robot Hands." This paper focused on the formulation of a hybrid methodology that simplifies the execution of dexterous, in-hand manipulation tasks using adaptive robot hands by the formulation of a mixed-method that combines analytical models, constrained optimization schemes, and machine learning techniques to simplify the execution of dexterous, in-hand manipulation tasks with adaptive robot hands. More precisely, the constrained optimization scheme is used to describe the kinematics of adaptive hands during the grasping and manipulation processes, unsupervised learning (clustering) is used to group similar manipulation strategies, dimensionality reduction is used to either extract a set of representative motion primitives (for the identified groups of manipulation strategies) or to solve the manipulation problem in a low-d space and finally, an automated experimental setup is used for unsupervised, automatic collection of large datasets. It also assesses the derived manipulation models' capabilities and primitives for both model and everyday life objects. We analyze the resulting manipulation ranges of motion (e.g., object perturbations achieved during the dexterous, in-hand manipulation). It showed that the proposed methods facilitate fingertip-based, within-hand manipulation tasks while requiring minimal sensory information and control effort. We demonstrate this experimentally on a range of adaptive hands.

[2]. Jianshu Zhou, Xiaojiao Chen, Yong Hu, and Zheng Wang, "A Soft-Robotic Approach to Anthropomorphic Robotic Hand Dexterity." It has been demonstrated that introducing passive compliance into robotic hands could impose drastic changes to specific dexterity measures. Soft robotics is quickly emerging in anthropomorphic mechanical hand design, with innovative soft robot hands reported to achieve a substantial subset of human hand dexterity, despite their substantially lower mechanistic sophistication than conventional rigid or underactuated robotic hands. More interestingly, soft robot hands are most successful in reproducing object grasping rather than in-hand manipulation tasks. Inspired by this notable advance, this paper investigated the gentle robotic approach on the influence of passive compliance to functional dexterity, offering insights into their efficacy and addressing the remaining gaps to fully

replicating human hand dexterous motions. This novel 26-DOF soft robotic hand design is the most dexterous soft robotic hand to date; besides, it was also demonstrated that passive compliance could substantially increase the range of motion for a finger, contributing to in-hand writing.

[3]. Minas Liarokapis and Aaron M. Dollar, "Deriving Dexterous, In-Hand Manipulation Primitives for Adaptive Robot Hands." This paper proposed a methodology for the automated extraction of dexterous, in-hand manipulation primitives for adaptive, underactuated, or compliant robot hands. Adaptive robot hands have changed the way we approach and think of robot grasping and manipulation. Traditionally, pinch, fingertip grasping, and dexterous, in-hand manipulation tasks were executed with fully actuated, rigid robot hands and relied on analytic methods, computation of the hand object Jacobians extensive numerical simulations for deriving optimal and minimal effort grasps. However, even insignificant uncertainties in the modeling space could render the extraction of candidate grasps or manipulation paths infeasible. Adaptive hands use underactuated mechanisms and structural compliance, facilitating by design the successful extraction of stable grips and the robust execution of manipulation tasks, even under significant object pose or other environmental uncertainties.

[4]. Minas V. Liarokapis, Aaron M. Dollar and Kostas J. Kyriakopoulos, "Human-like, Task-Specific Reaching and Grasping with Redundant Arms and Low-Complexity Hands." This paper proposed a closed-loop methodology, human-like, task-specific reaching and grasping with redundant robot arms and low-complexity robot hands. This paper presents a closed-loop procedure, human-like, task-specific reaching and grasping with redundant robot arms and low-complexity robot hands. Human demonstrations are utilized in a learn by demonstration fashion to map human to human-like robot motion. Principal Components Analysis (PCA) is used to transform the human-like robot motion in a low-dimensional manifold, where appropriate Navigation Function (NF) models are trained. A series of grasp quality measures and task compatibility indexes are employed to guarantee the robustness of the computed grasps and task specificity of goal robot configurations. The final scheme provides anthropomorphic robot motion, task-specific robot arm configurations and hand grasping

postures, optimized fingertips placement on the object surface (that results in robust grasps), and guaranteed convergence to the desired goals.

[5]. Minas Liarokapis, Panagiotis K. Artemiadis, and Kostas J. Kyriakopoulos, "Quantifying Anthropomorphism of Robot Hands." This paper proposed a systematic approach to quantify the anthropomorphism of robot hands. A comparison is performed, taking into account those specifications that make the human hand the most agile and versatile end effector known (e.g., opposable thumb, palm mobility, etc.). Here, it discusses an index for the quantification of anthropomorphism of robot arms. The index is defined as a weighted sum of specific metrics which evaluate the similarities between the human and robot arm workspaces, providing a normalized score between 0 (non-anthropomorphic artifacts) and 1 (identical human artifacts). The human arm workspaces were extracted using data reported in anthropometry studies. The formulation is general enough to allow utilization in various applications by adjusting the weighting factors according to the specifications of each task. The proposed methodology can be used to assess the human likeness of existing robot arms and provide specifications for the design of new anthropomorphic robots and prosthetic devices.

[6]. George P. Kontoudis, Minas V. Liarokapis, Agisilaos G. Zisimatos, Christoforos I. Mavrogiannis, and Kostas J. Kyriakopoulos, "Open-Source, Anthropomorphic, underactuated Robot Hands with a Selectively Lockable Differential Mechanism: Towards Affordable Prostheses". This paper presented an open-source design to develop anthropomorphic, underactuated robot hands that utilize a selectively lockable differential mechanism. The robot hand user can switch to different grasping postures using the finger bases' buttons intuitively.

[7]. Minas Liarokapis, Charalampos P. Bechlioulis, George I. Boutselis, and Kostas J. Kyriakopoulos, "A Learn by Demonstration Approach for Closed-Loop, Robust, Anthropomorphic Grasp Planning." Human-inspired optimization principles were proposed to derive task-specific, robust grasp configurations. Tactile sensors mounted on the robot's fingertips were used to confront uncertainties regarding the joint displacements. This paper presents a series of design directions for developing affordable, modular, lightweight, intrinsically compliant, underactuated robot hands that can be easily reproduced using off-the-

shelf materials. The proposed robot hands, efficiently grasp a series of everyday life objects and are considered general-purpose, as they can be used for various applications. The efficiency of the proposed robot hands has been experimentally validated through a series of experimental paradigms, involving:
grasping multiple everyday life objects with different geometries, myoelectric (EMG)control of the robot hands in learning tasks, preliminary results on a grasping capable quadrotor, and autonomous grasp planning under object position and shape uncertainties.

[8]. Saurabh A. Khajone, Dr. S. W. Mohod, V.M. Harne, "Implementation of a Wireless Gesture Controlled Robotic Arm." A low-cost computer vision system that can be executed in a typical PC with low power USB webcam was one of the main objectives. The idea is to change the perception of remote controls for actuating manually operated Robotic-Arm. This paper presents a thought and a way to eradicate the buttons and joysticks and replace them with some of the more intuitive techniques, controlling the complete Robotic Arm by the operator's hand gesture. The proposed electronics system recognizes a particular hand gesture performed in front of webcam & transmitted respected signals wirelessly through the RF module. Depending on the received signals, the robotic arm, followed by the AVR microcontroller, performs the receiver section's receptive motions.

3. Objective

This project focuses on the challenging problems of the measure and the reproduction of the finger capabilities. However, these days, NO or SCARCE effort has been devoted to comprehending the feasibility limits due to the repeatability and reliability problems in measuring the operator's fingers' multiple joints and reproducing them by a robotic hand.
The intuitive nature of the system allows it to be handled quickly, without much guidance or practice. Further, to produce a wireless artificial robotic hand that mimics the human hand (via-glove) on manipulating the objects and contributing to robot end-effector grasping problems and robot reprogramming difficulty.

3.1 Scope of objective:

The main objective of this project is to investigate the characteristics and performance of the development of the robotic arm to mimic the human hand and to design control parts of the robot hand by midrange microcontroller and to use 3-D printed hand models as it is versatile and it is light in weight and perfect for human hand anthropomorphism. The same objectives that will be carried out in this project, have been explained below:

Table 3.1. RELATION OF PARTS OF HUMAN HAND (TX) CONTROLLING RESPECTIVE PARTS OF THE ROBOTIC ARM (RX).

HUMAN HAND	ROBOTIC ARM
All fingers unflexed	Grip open
Finger movement	Respective mimic of finger gesture
Grasping action for different objects	Grasping action performed for the unknown object
All fingers flexed	Grip Closed

In the future, the system has potential in fields where teleoperation may be required, including medicine, industrial operations, disaster recovery, bomb disposal, and undersea recovery. This project is implemented to control the movement by using the glove to integrate with hand and teleoperate by RF wireless module using simple human hand gestures.

4. Methodology

4.1 Interconnections between the components:

A total of 5 flex bend sensors, each is assigned to a specific finger of the hand (thumb, index, middle, ring, and pinky). Suppose any of the fingers moves or lifts. In that case, it will send resistivity reading to the receiver circuit via a transceiver module from the embedded Arduino NANO controller at the transmitter part.

The 1st Arduino NANO controller is placed at the transmitter circuit that will transmit the signal to the receiver via a transceiver module connected to it. The movement is wirelessly received by a receiver transceiver connected to the 2nd Arduino UNO controller at the receiver part. It sends the signals to the servo motors to do its task at particular fixed mapping functions. Accordingly, the robotic arm will do actions according to different human hand gestures; thus, human-robot interaction can be achieved.

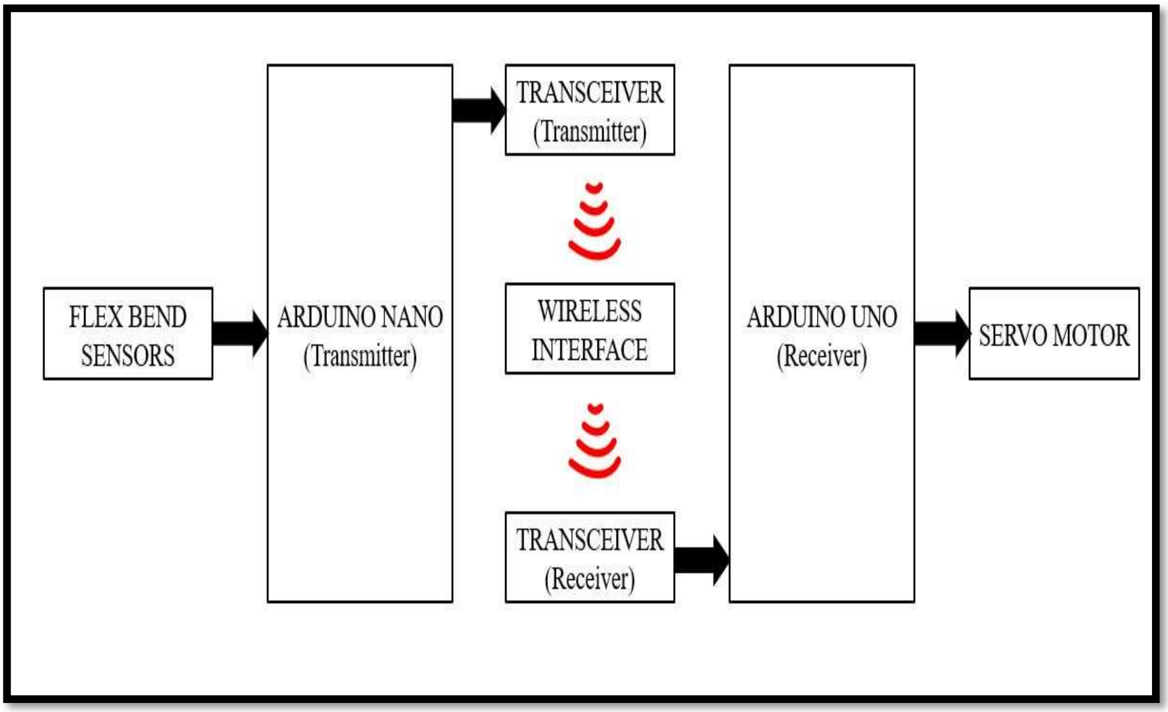


Figure 4.1. Block diagram of the system

Fig.4.1. shows the block diagram of the system. Flex bend sensors are set to be input, and servo motors are placed to be the output.

4.2 Transmitter - Flex Sensor configuration:

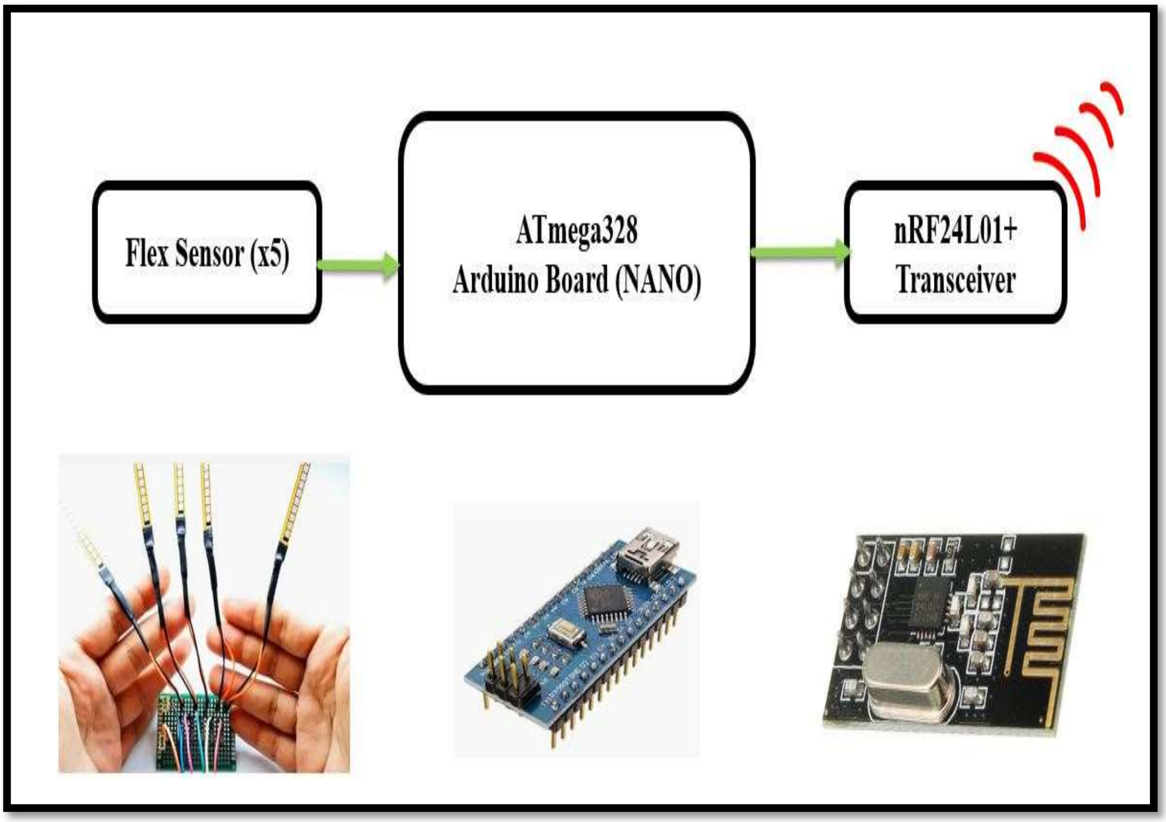


Figure 4.2. Block diagram of the transmitter system

A flex sensor or bend sensor is a sensor that measures the amount of deflection or bending. A flex sensor uses carbon on a plastic strip to act as a variable resistor—the resistance changes by flexing the component. The sensor bends in one direction; the more it turns, the higher the resistance gets. Usually, the sensor is stuck to the surface, and the sensor element's resistance is varied by bending the surface. Since the resistance is directly proportional to the amount of bend, it is used as a goniometer and is often called a flexible potentiometer.

Further in Fig. 4.3. shows the flowchart for the transmitter system from the human hand. When the circuit is ON, it will wait and standby for any possibilities in resistivity reading changes caused by moving fingers. A total of 5 flex bend sensors, each is assigned to a specific finger of the hand (thumb, index, middle, ring, and pinky). If any of the fingers move or lift, it will send resistivity reading to the receiver circuit via a transceiver.

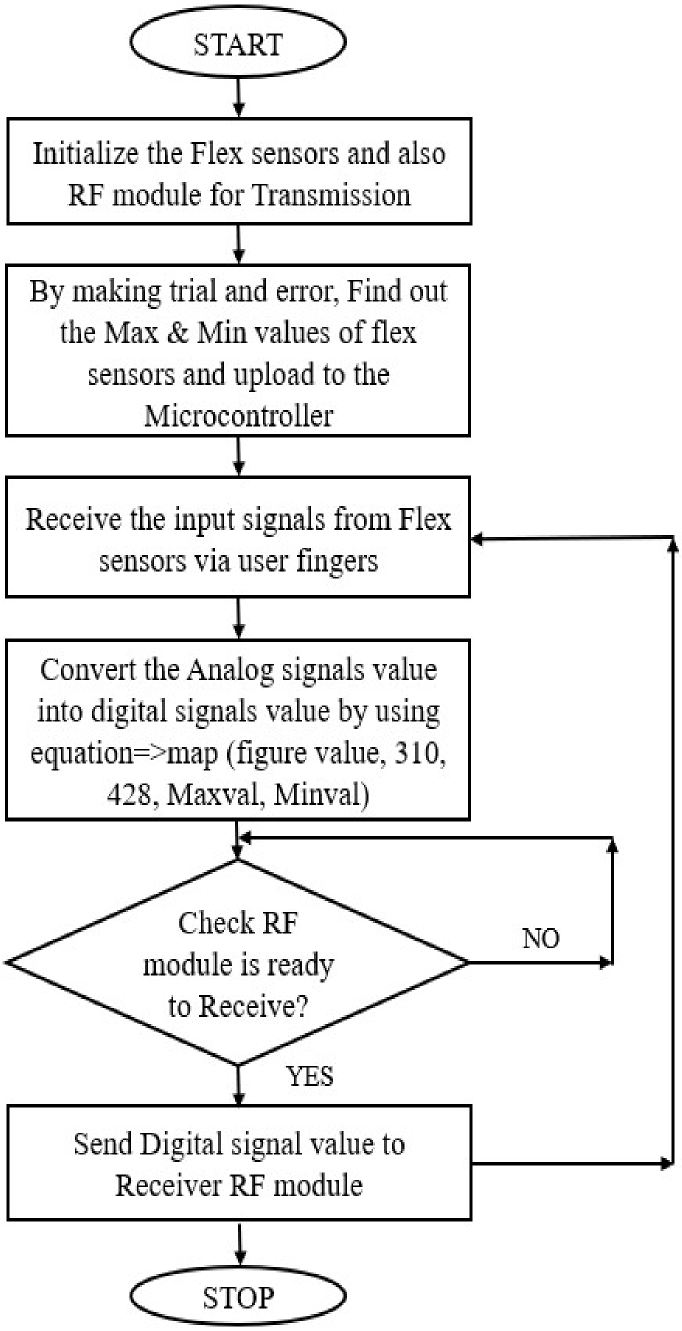


Figure 4.3. A brief flowchart of the Transmitter system

Further, the software development with the flex-sensors is configured. The max-min flexing values for each finger are carried out using the trial-error method, and different results are noted in table 4.1.

Table 4.1. VARIOUS DIFFERENT OBSERVATIONS OF INDIVIDUAL FINGERS (Tx).

Finger:	Max Flex (Digital Values)	Min Flex (Digital Values)
Thumb	428	310
Index	510	426
Middle	350	285
Ring	240	200
Pinky	447	350

4.3 Receiver - Servo Motor configuration:

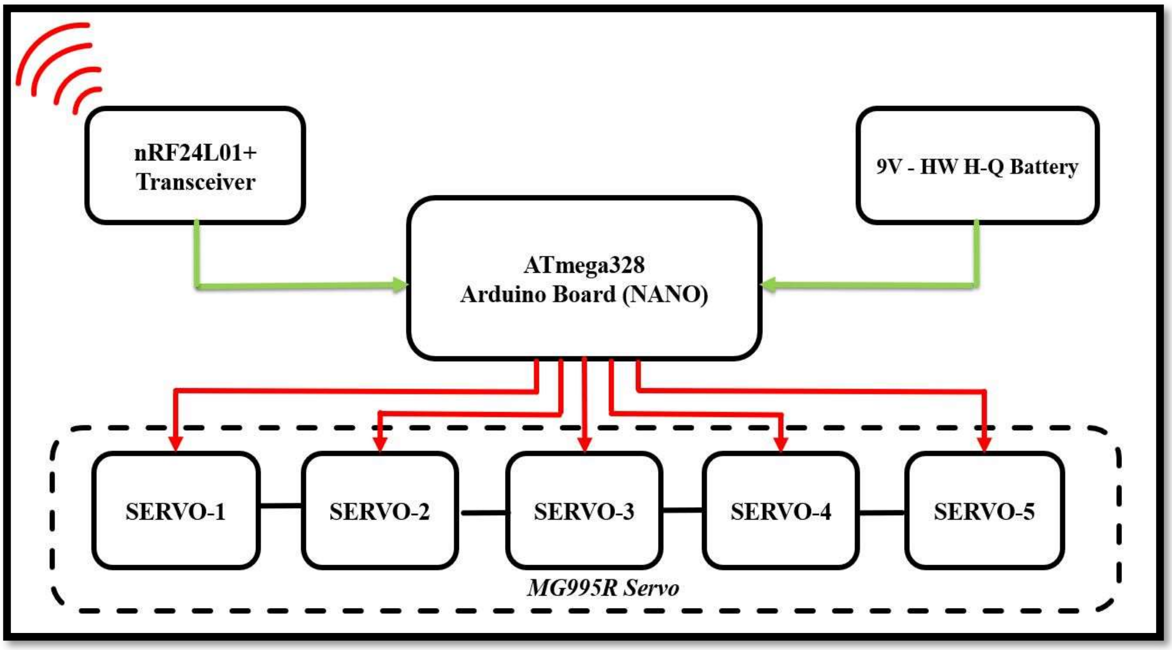


Figure 4.4. Block diagram of the receiver system

Servo sweeps the shaft of an RC servo motor back and forth. This project makes use of the Arduino servo library, which is a pre-defined function. Servo motors have three wires: power, ground, and signal. The power wire is typically red and should be connected to the 5V pin on the Arduino UNO board. The ground wire is usually black or brown and should be connected to a ground pin on the board. The signal pin is typically yellow, orange, or white and should be related to the board. The similar working and procedure of the same have been depicted below:

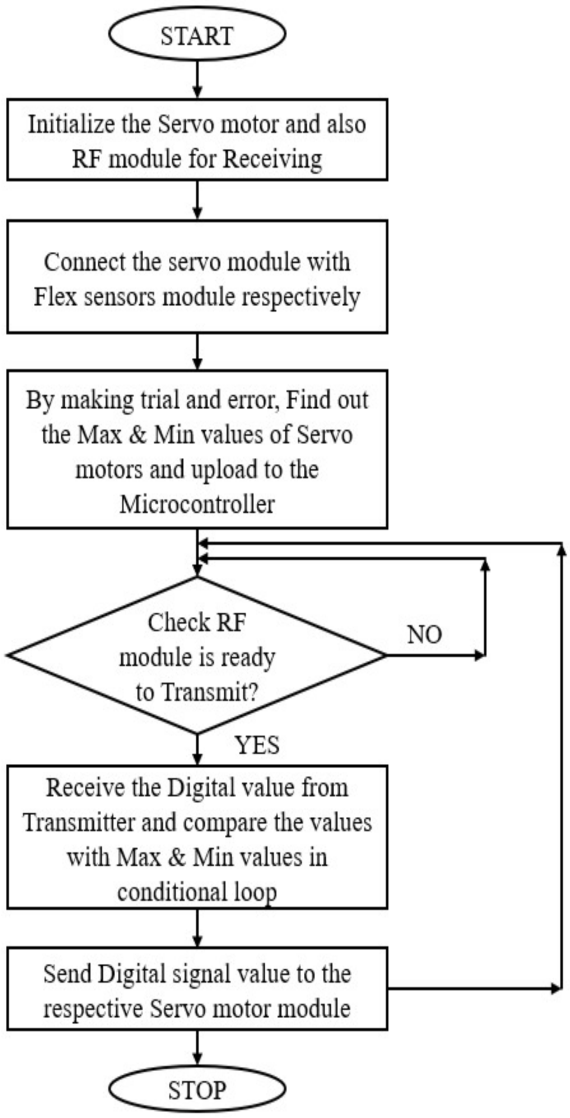


Figure 4.5. A brief flowchart of the Receiver system

Fig. 4.5. shows the flowchart for the receiver system at the robotic arm. When the circuit is ON, it will receive signals and readings from the transmitter circuit via a transceiver and send the servo motors' alerts to rotate at a correct degree. A total of 5 servo motors are used for fingers. Each engine has a specific task to lift its respective fingers. Further, similarly as recorded with flex observation, the software development with the servo-motors is configured

and the max-min servo calibration values. Each finger is carried out using the trial-error method; different results are noted, as shown in table 4.2.

Table 4.2. VARIOUS DIFFERENT OBSERVATIONS OF INDIVIDUAL FINGERS (Rx).

Finger:	Max Servo Value Displaced (in Degrees)	Min Servo Value Displaced (in Degrees)
Thumb	110	40
Index	100	30
Middle	95	05
Ring	90	25
Pinky	90	25

These values are unique w.r.t each user's work; as far as this project is concerned, the different values are recorded and manipulated to each independent finger as program arguments. Each variable is configured and checked within the Arduino compiler with valid executive logic and procedure.

4.4 Hardware Implementation:

Figure 4.6. (a)Transmitter system hardware prototype along with (b)Receiver model

Fig. 4.6. The user will wear the prototype for the transmitter system to control the robotic arm system. The sensors used to detect the human hand movements are flex bend sensors embedded onto the surface, and the sensor element's resistance is varied by bending the surface.

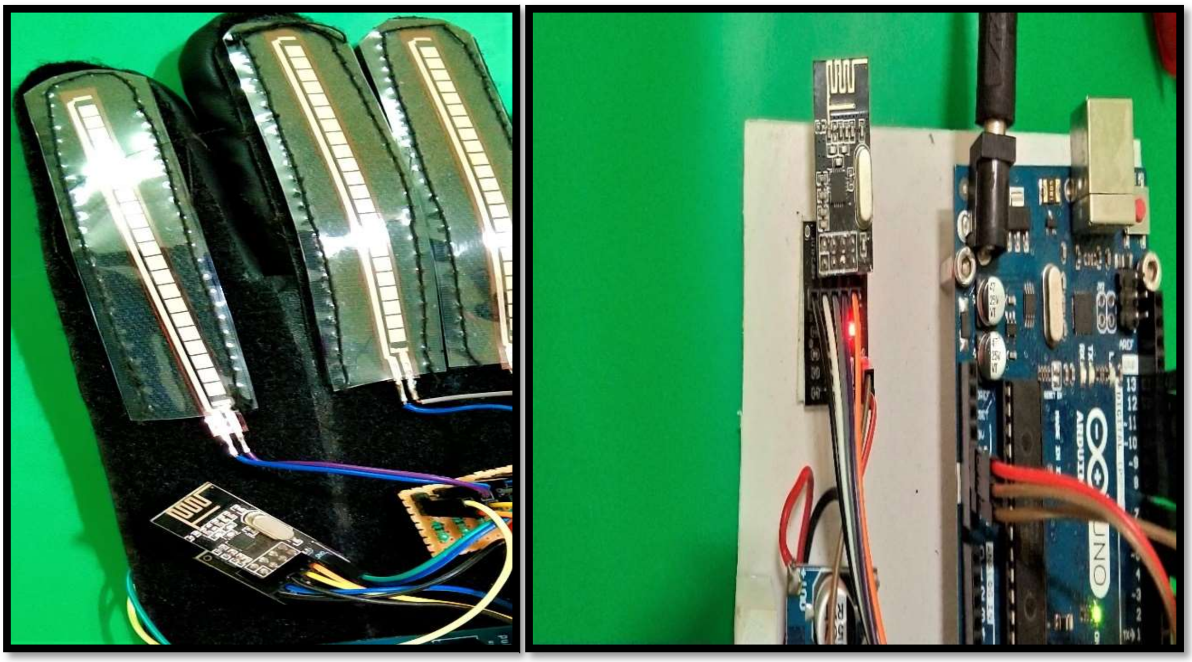


Figure 4.7. Communication system hardware prototype of (Tx)& (Rx)

Fig. 4.7. (Tx) and (Rx) shows the transmitter communication system hardware and its receiver communication system's hardware prototype. The communication between both circuits is connected with a transceiver module NRF24L01. This transceiver module is a wireless connection component that uses a 2.4 GHz frequency. It can operate from 250 Kbps up to 2 Mbps and ranges up to 100 meters. The power consumption is just around 12 mA. The controller for the system used is the Arduino NANO & UNO modules which allow input and output mechanisms to interact with the design and external environments. The Arduino microcontroller module provides input and output mechanisms to interact with the system and external environment. There are many types of Arduino boards, such as UNO, NANO, and MEGA. For this proposed project, Arduino UNO and NANO boards are selected as it is more convenient to use and more compatible with most components than other boards.

5. Hardware/ Software Specifications

The purpose of the Requirements & Specification is to outline the project's requirements and specifications—the advantages of the microcontroller and components such as sensors, transceiver module, etc. Specifications also contain pin descriptions, Key features, power ratings, and resolutions of the parts. Requirements and Specifications are divided into two kinds, they are:

- Hardware Requirements and Specifications.
- Software Requirements and Specifications.

5.1 Hardware Specification

Hardware Requirements and Specifications is a document or set of documentation that describes the features and behavior. It also specifies the prerequisites for the system to operate without any failure. It also determines the power ratings of the hardware. The hardware required for the project and the specification of the hardware are given below.

5.1.1 NRF24L01+ SI24R1

NRF24L01+ SI24R1, 2.4G is a wireless power enhanced communication receiver module. The nRF24L01 is configured and operated through a Serial Peripheral Interface (SPI). Through this interface, the register map is available. The register map contains all configuration registers in the nRF24L01 and is accessible in all operation modes of the chip. The embedded baseband protocol engine (Enhanced Shock Burst) is based on packet communication and supports various manual operation modes to the advanced autonomous protocol operation.
Internal FIFOs ensure a smooth data flow between the radio front end and the system's MCU. The front end uses GFSK modulation. It has user-configurable parameters like frequency channel, output power, and air data rate supported by the nRF24L01, which is configurable up to 2Mbps.

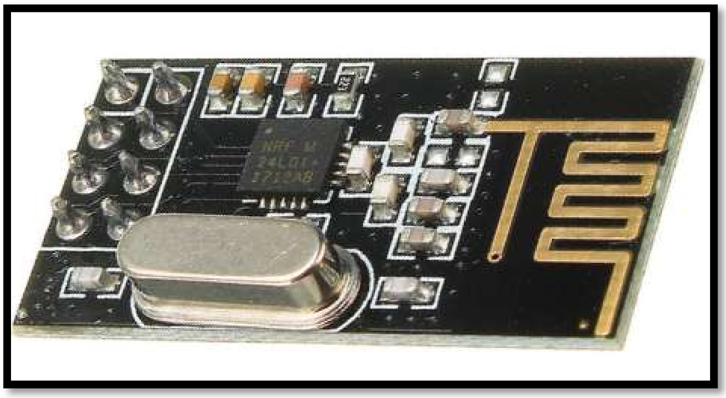


Figure 5.1. NRF24L01+ SI24R1

- A fully compatible (SPI Synchronous serial communication interface) that can communicate with each other and supports power-enhanced communications simultaneously.
- Works 2.4 GHz of frequency.
- Operates from 250Kbps up to 2Mbps
- Ranges to 100 meters.
- The power consumption is just around 12mA.

The below table explains the basic pin description of the transceiver module.

Pin description of nRF24L01 Transceiver

Pin No.	Pin Name Description
1	GND Ground
2	Vcc power (3.3v)
3	CE Chip Enable
4	CSN Chip Select Not
5	SCK Serial Clock
6	MOSI Master Out Slave In
7	MISO Master In Slave Out
8	IRQ Interrupt

5.1.2 Microcontroller ATmega328

The ATmega328 is a single-chip microcontroller created by Atmel in the megaAVR family (later, Microchip Technology acquired Atmel in 2016). It has a modified Harvard architecture 8-bit RISC processor core. It consists of two memories- Program memory and data memory. The code is stored in the flash program memory, whereas the data is stored in the data memory. The Atmega328 has 32 KB of flash memory for storing code.

- Operating Voltage (logic level): 5V
- Input Voltage (recommended): 7V ~ 12V

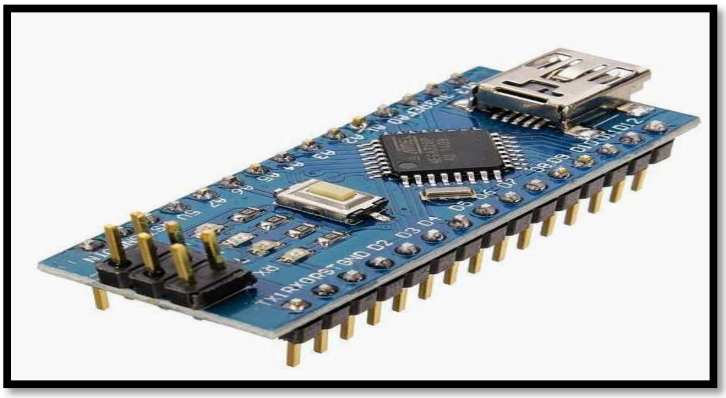


Figure 5.2. Microcontroller ATmega328

- Input Voltage (limits): 6V ~ 20 V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 8
- DC Current per I/O Pin: 40mA
- Flash Memory: 32KB (ATmega328) SRAM: 2KB (ATmega328)
- EEPROM: 1KB (ATmega328)
- Clock Speed: 16MHz

5.1.3 MG995R Digital Servo

Features upgraded shock-proofing and a redesigned PCB and IC control system that make it much more accurate. This high-torque standard servo can rotate approximately 120 degrees (60 degrees in each direction). The Servo motor works on the PWM (Pulse width modulation)

Principle; the applied pulse duration controls its angle of rotation to its Control Pin. Servo motor is made of DC motor controlled by a variable resistor (potentiometer) and some Gears. Servo motors can usually only turn 90'deg in either direction for a total of 180'deg movement. Both AC and DC motors can be used as Servo motors.

Figure 5.3. MG995R Digital Servo

- Weight: 55g
- Operating Speed: 0.20sec / 60 degrees (4.8V no load)
- Operating Speed: 0.16sec / 60 degrees (6.6V no load)
- Stall Torque: 10 kg-cm at 4.8V
- Stall Torque: 12 kg-cm at 6.6V
- Operation Voltage: 4.8 - 7.2Volts
- Temperature: -30°C to +60°C

Pin description of servo motor

Pin No.	Pin Name	Description
1	GND	Ground
2	VCC	Power (5v)
3	Input control	Control the servo motor

5.1.4 Flex Sensor

A flex sensor or bend sensor is a sensor that measures the amount of deflection or bending. Usually, the sensor is stuck to the surface, and the sensor element's resistance is varied by bending the surface. Since the resistance is directly proportional to the amount of bend, it is used as a goniometer and is often called a flexible potentiometer.

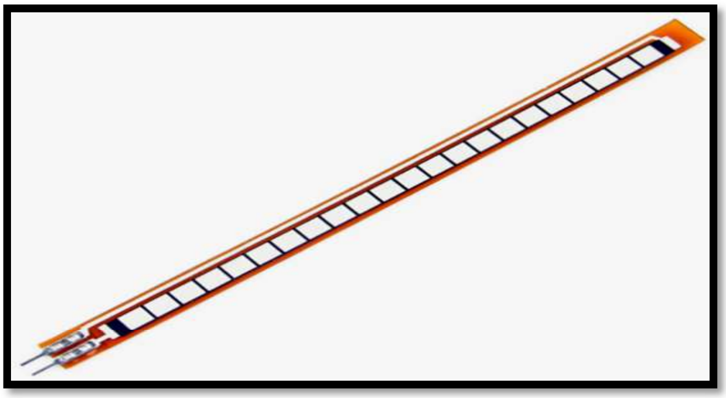


Figure 5.4. Flex Sensor

- As the sensor is flexed, the resistance across the sensor increases
- Flat Resistance: 10K Ohms ±30%
- Bend Resistance: minimum two times greater than the flat resistance at 180° pinch bend
- Power Rating: 0.5 Watts continuous; 1-Watt Peak

Mechanical Specifications

- Life Cycle: >1 million bends
- Temperature: -35°C to +80°C

5.2 Software Specification

Arduino IDE (Integrated Development Environment) software is used to integrate or program the system into the Arduino NANO & UNO (ATMEGA328) boards. It has its library functions for input and output interfacing. The program created in the software can be accessed quickly, and changes can be done at the user's preferences. The Arduino IDE software needs to be updated from time to time to ensure that the libraries are up-to-date to prevent any error while running the program.

The IDE environment mainly contains two essential parts: Editor and Compiler, where the former is used for writing the required code and is later used to compile and upload the code into the given Arduino Module. The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded to the board's controller. The IDE environment is mainly distributed into three sections:

- Menu Bar
- Text editor
- Output pane

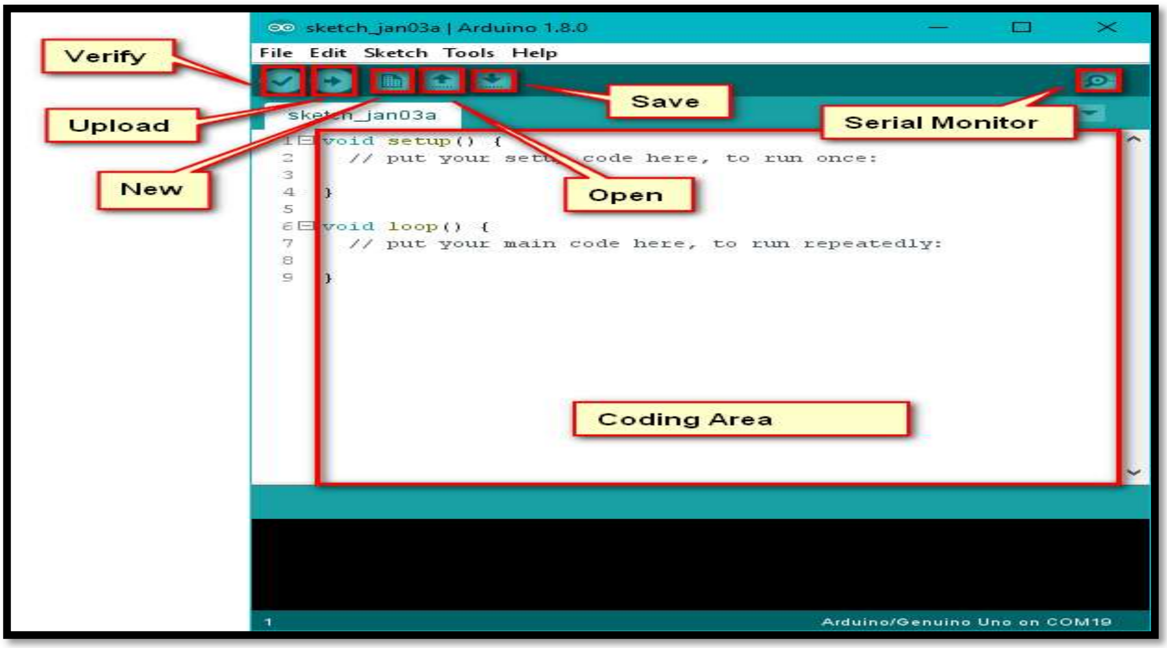


Figure 5.5. Basic structure view of Arduino ID

The bar appearing on the top is called Menu Bar that comes with five different options as follow:

- File: It is used to open a new window for writing the code or open an existing one
- Edit: Used for copying and pasting the code with further modification for font
- Sketch: For compiling and programming
- Tools: Mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller.
- Help: If you feel skeptical about software, complete support is available from getting started to troubleshooting.

The Six Buttons appearing under the Menu tab are connected with the running program as follows:

- The checkmark appearing in the circular button is used to verify the code. Click this once written the code.
- The arrow key will upload and transfer the required code to the Arduino board.
- The dotted paper is used for creating a new file.
- The upward arrow is reserved for opening an existing Arduino project.
- The downward arrow is used to save the current running code.
- The button appearing on the top right corner is a Serial Monitor. This separate pop-up window acts as an independent terminal and plays a vital role in sending and receiving the Serial Data. The Serial Monitor will help debug the written Sketches where the programmer can get a hold of how it operates. The menu board's main screen is a simple text editor used for writing the required code.
- The bottom of the main screen is described as an Output Pane that mainly highlights the running code's compilation status: the memory used by the code, and errors in the program.

5.2.1 Why is Arduino advantageous?

- **Inexpensive:** Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than 400 INR.
- **Cross-platform:** The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- **Simple, transparent programming environment:** The Arduino programming environment is easy to use for beginners yet flexible enough for advanced users. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino.

- **Open source and extensible software:** The Arduino software is published as open-source tools available for experienced programmers. The language can be expanded through C++ libraries.
- **Open source and extensible hardware:** The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers. The plans for the modules are published under a Creative Commons license, so experienced circuit designers can make their version of the module, extending it and improving it. Even relatively inexperienced users can build the module's breadboard version to understand how it works and save money.

5.3 Software Program

Table 5.1. TRANSMITTER SYSTEM PSEUDO CODE

Pseudo Code Transmitter	Description
#include <SPI.h> #include <RF24.h>	• The communication interface with the modem • The library which helps us to control the radio modem
int msg [1]; RF24 radio (pinCE_NRF, pinCSN_NRF); const byte address [6] = "00001";	• Initialize msg [1] for the receiver system to identify • Digital PINs to which signals CE and CSN is connected. • The address of the modem that will receive data from Arduino

void setup () {
radio.begin();
Serial.begin(9600);
radio.read(values, sizeof(values));
	• Activates the modem.
	• Serial Read with the baud rate of 9600
	• Sets the address of the receiver to which the program will send data.

void loop () {
thumbVal = values [0];
indexVal = values [1];
middleVal = values [2];
ringVal = values [3];
pinkyVal = values [4];}
	• Five flex bend sensors are connected at an analog input pins 0 to 4 per flex bend sensor on the Arduino UNO board.

thumbPos = map (thumbVal, 310, 428, thumbPosMin, thumbPosMax);
indexPos = map (indexVal, 426, 510, indexPosMin, indexPosMax);
middlePos = map (middleVal, 285, 350, middlePosMin, middlePosMax);
ringPos = map (ringVal, 200, 240, ringPosMin, ringPosMax);
pinkyPos = map (pinkyVal, 350, 447, pinkyPosMin, pinkyPosMax);
	• Linearization is made in the map Function. For example, (the thumbnail, 310,
	428, PosMin, PosMax) is ('from low', 'from high', 'to low', 'to high').
	Meaning the value of analog read (0) is between 310 to 428 and will be converted into 0 to 10.

Table 5.2. RECEIVER SYSTEM PSEUDO CODE

Pseudo Code Receiver	Description
#include <Servo.h> #include <SPI.h> #include "RF24.h"	• The library which helps us to control the servo motors • The communication interface with the modem • The library which helps us to control the radio modem
Servo srvThumb; Servo srvIndex; . .	• Defining the servo names
RF24 radio (pinCE_NRF, pinCSN_NRF); const byte address [6] = "00001";	• Digital PINs to which signals CE and CSN are connected. • The address of the modem that will receive data from Arduino

void setup () {
srvThumb.attach(THUMB_PIN);
srvIndex.attach(INDEX_PIN);
srvMiddle.attach(MIDDLE_PIN);
srvRing.attach(RING_PIN);
srvPinky.attach(PINKY_PIN);
radio.begin();
radio.openReadingPipe(0, address);
 	
radio.startListening();
}
	• Attach 5 servos to digital pin per The servo motor is connected to the Arduino UNO board.
	
	• Activates the modem.
	• Determines the address of the modem which receive data.
	• Enable receiving data via modem.

<table>
<tr>
<td>

```
void loop () {
if(radio.available()) {
radio.read(values, sizeof(values));

if (thumbPos < thumbPosMin) {
thumbPos = thumbPosMin;
} else if (thumbPos > thumbPosMax) {
thumbPos = thumbPosMax;
.
.
.
}
srvThumb.write(thumbPos);
srvIndex.write(indexPos);
srvMiddle.write(middlePos);
srvRing.write(ringPos);
srvPinky.write(pinkyPos);

}
```

</td>
<td>

• Receiver system starts to read inputs
From the transmitter system.

• Linearization is made in the map function.
Different linear positions obtained via flex
bending is compared w.r.t the max and min
values of different servo positions
It is connected to each respective finger.
This means the value of msg[0] is between
0 to 10 and will be converted into 0 to 180
in the term of angle for the servo motors to
Rotate.
• The coding follows throughout the other
analog values and repeated procedures are
They were carried out for different arguments.

</td>
</tr>
</table>

Table 5.1 & 5.2 shows the pseudo-code for the system's transmitter and receiver systems, with a brief description of the coding used for easier understanding.

6. Experimental Results and Discussion

In this paper, we proposed a systematic approach to quantify the anthropomorphism of robot hands. Once the systems are activated, both circuits attempt to connect via transceiver module NRF24L01. After the connection has been made, the user will use the hand gesture-controlled robotic arm. This project only covers the gripping part of the robotic arm and mimicking of human anthropomorphic gestures. These project designs were made using Solidworks software. The generated STL files were fed to Cura software for slicing and further printed with a 3-D printer. Presently this is a working robotic hand that can mimic commands as given by a human hand wirelessly. The hardware responded successfully to the interaction of both the transmitter system and the receiver system. This study has shown that it is much easier to control a robotic arm by using human hand gestures rather than using keyboards, joysticks, or mice. The following given images will explain the experimental results and provide a brief explanation of the proposed ideology and how it works:

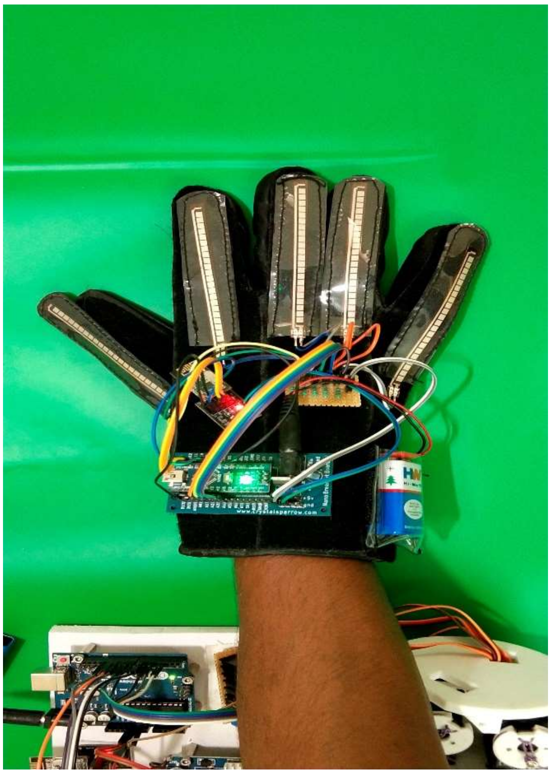


Figure 6.1. All fingers unflexed

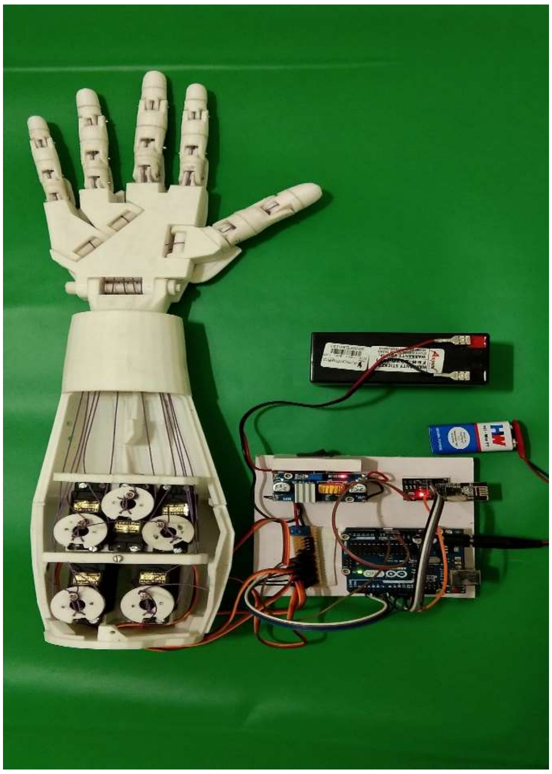


Figure 6.2. Grip open

Figure 6.3. Finger movement

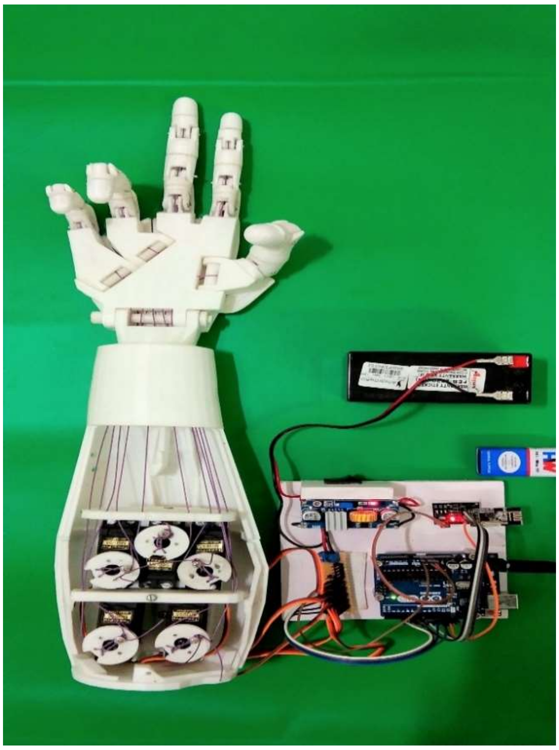


Figure 6.4. Respective mimic of finger gesture

Figure 6.5. Grasping action for different objects

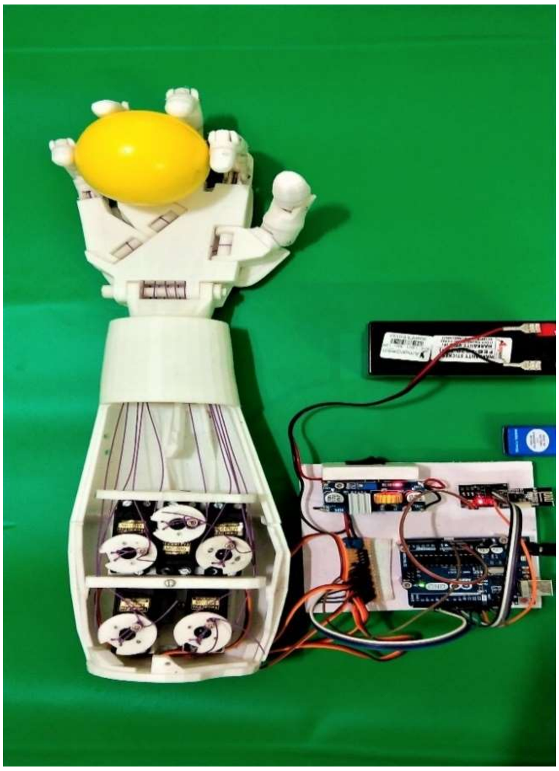


Figure 6.6. Grasping action performed for the unknown object

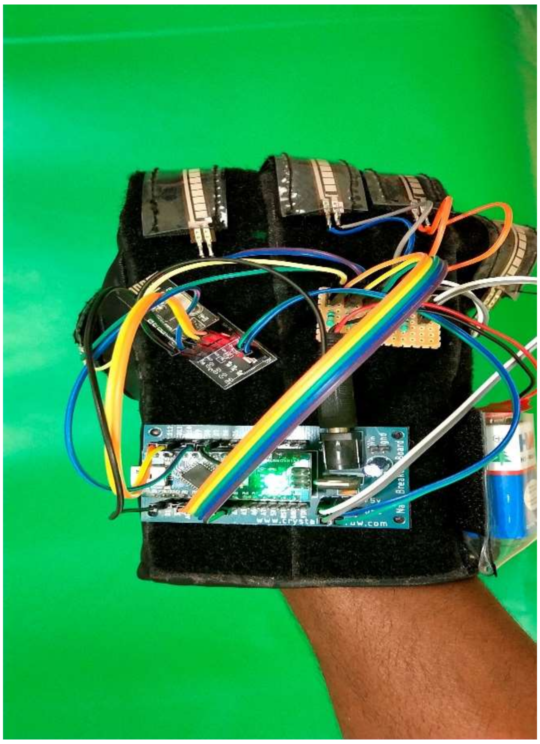


Figure 6.7. All fingers flexed

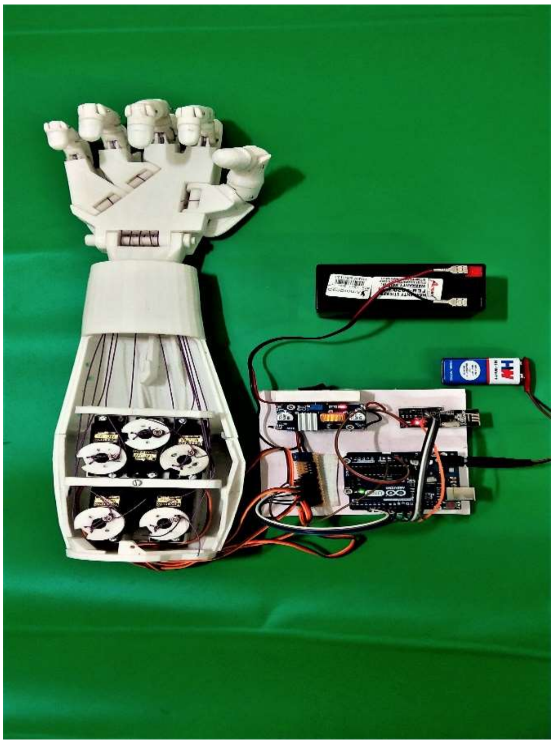


Figure 6.8. Grip Closed

A video containing the same working prototype of this model is attached to this report, here we may see the basic functioning and a brief idea of the proposed methodology at the following URL:

Wireless Dexterous Anthropomorphic, Gesture Controlled Robotic Arm

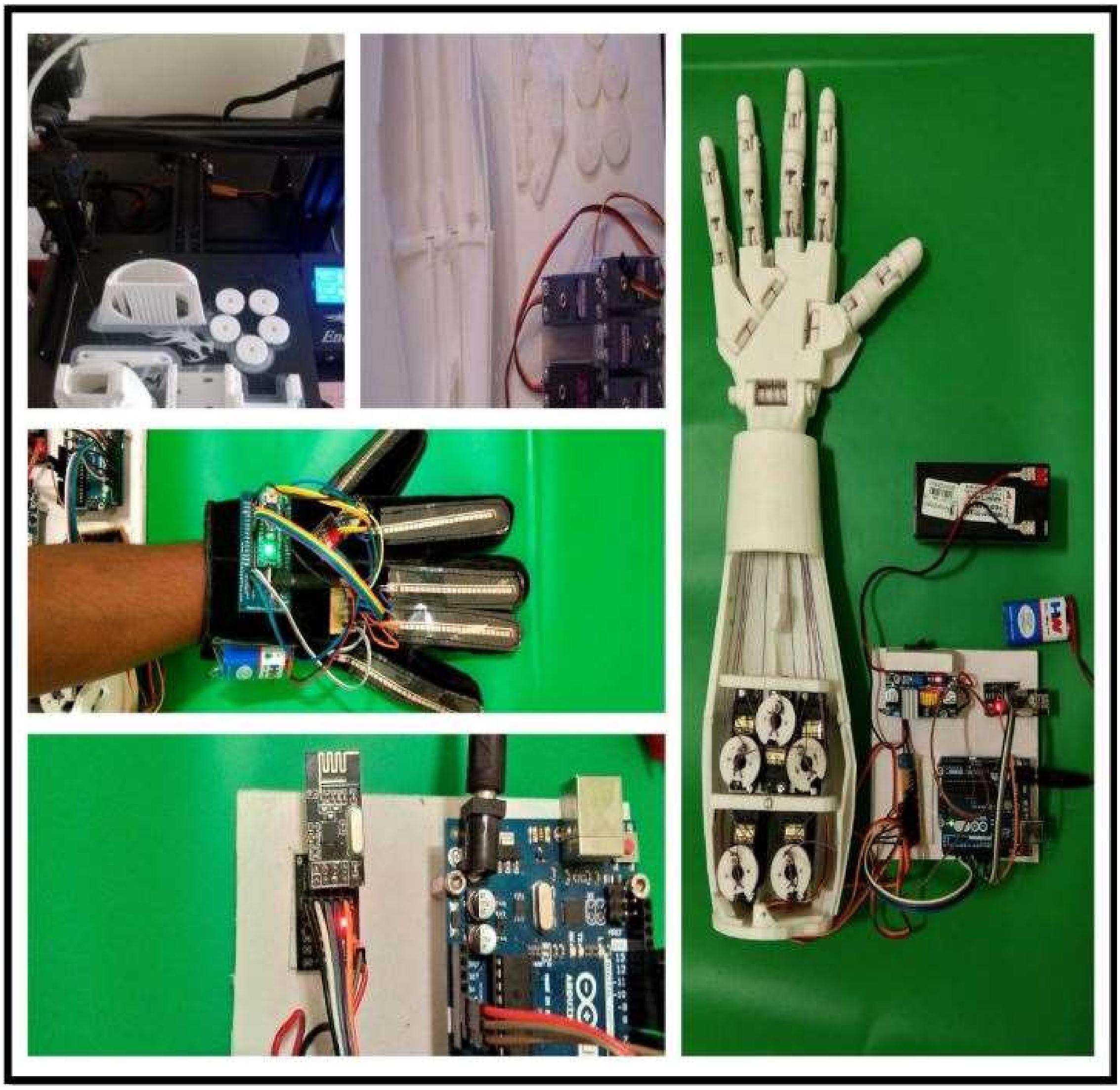


Figure 6.9. An overall view of the project components making and assembling

7. Conclusion

The components used to complete this project can be improved to a much higher level in the future. The transceiver module can be switched to a better module covering up to large distances, eliminating humans' limitations to reach places they can't. The microcontroller Arduino UNO is simple and is prone to be hacked. As for the prototype, a well-designed robotic arm is to be done to achieve the maximum output produced by this project.

The material used must be to grip, picking up, and holding objects. Many human parts of the body can be automated or machined to help humankind perform daily tasks. The gesture made in this method involves only one hand, which reduces the number of gestures that can be made using both hands. However, several parameters are a setback for the system, such as the robotic arm's material and the wireless communication modules used. These parameters will determine the limits of the robotic arm in terms of its strength and durability.

8. Future Implications

- Certain grasping and force closure identification and replication for various uniform and non-uniform objects may be carried out where robot AI will be tested.
- It may be implemented for easy daily use anthropomorphic prostheses applications.
- Medical uses by doctors to perform surgical operations at distant places.
- A combination of the traditional and gesture-controlled robotic arm may prove to be very handy, providing the component both flexibility as well as accuracy
- Such arms may prove handy in such sectors where the precision must be adjusted from time to time.
- It can be used in mines, space, fire-fighters, etc., where human intervention is impossible.
- It can be used for bomb disposal systems, offering as much accuracy as a human arm and saving a human life.

Reference

[1]. Minas Liarokapis and Aaron M. Dollar, "Combining Analytical Modeling and Learning to Simplify Dexterous Manipulation with Adaptive Robot Hands," IEEE transactions on automation science and engineering, 1545-5955 © 2018 IEEE.

[2]. Jianshu Zhou, Xiaojiao Chen, Yong Hu, and Zheng Wang, "A Soft-Robotic Approach to Anthropomorphic Robotic Hand Dexterity," Digital Object Identifier *10.1109/ACCESS.2019.2929690,* VOLUME 7, 2019.

[3]. Minas Liarokapis and Aaron M. Dollar, "Deriving Dexterous, In-Hand Manipulation Primitives for Adaptive Robot Hands," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) September 24–28, 2017, Vancouver, BC, Canada.

[4]. Minas V. Liarokapis, Aaron M. Dollar and Kostas J. Kyriakopoulos, "Human-like, Task-Specific Reaching and Grasping with Redundant Arms and Low-Complexity Hands," IROS-2015, NSF grant IIS-1317976 and the European Commission with the Integrated *Project no. 248587*, THE Hand Embodied, within the FP7-ICT-2009-4-2-1 program Cognitive Systems and Robotics.

[5]. Minas Liarokapis, Panagiotis K. Artemiadis, and Kostas J. Kyriakopoulos, "Quantifying Anthropomorphism of Robot Hands," European Commission with the Integrated *Project no. 248587*, THE Hand Embodied, within the FP7-ICT-2009-4-2-1 program Cognitive Systems and Robotics, IROS-2015.

[6]. George P. Kontoudis, Minas V. Liarokapis, Agisilaos G. Zisimatos, Christoforos I. Mavrogiannis, and Kostas J. Kyriakopoulos, "Open-Source, Anthropomorphic, underactuated Robot Hands with a Selectively Lockable Differential Mechanism: Towards Affordable Prostheses," European Commission with the Integrated *Project no. 248587*, THE Hand Embodied, within the FP7-ICT-2009-4-2-1 program Cognitive Systems and Robotics, IROS-2015.

[7]. Minas Liarokapis, Charalampos P. Bechlioulis, George I. Boutselis, and Kostas J. Kyriakopoulos, "A Learn by Demonstration Approach for Closed-Loop, Robust, Anthropomorphic Grasp Planning," M.V. Liarokapis et al.

[8]. Saurabh A. Khajone, Dr. S. W. Mohod, V.M. Harne, "Implementation of a Wireless Gesture Controlled Robotic Arm," IJIRCCE- Vol. 3, Issue 1, January 2015.