International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 11, No.1 (Jan-2022)

https://dx.doi.org/10.12785/ijcds/110132

# Design, Implementation, Interfacing and Control of Internet of Robot Things for Assisting Robot

**Zeyad A. Karam** [1] **and Mustafa Abdulkadhim Neamah**[2]

[1,2]*College of information engineering, Al-Nahrain University, Iraq*

*Received 13 Jun. 2021 , Revised 6 Nov. 2021, Accepted 13 Nov. 2021 , Published 15 Jan. 2022*

**Abstract:** Robotic arm plays a great role in assisting the human hand in current life. This project involves the design and implementation of two degrees of freedom (DOF's) assistant arm robot for aiding humans in the medical recovery for the upper left limb motion. The contribution in this work involves three parts: the first one, design special interfacing circuits based on (Arduino card) for controlling the designed arm motion, where the motion execution and record is done by designed reading and writing circuits, these circuits controlled by a designed MATLAB fuzzy proportional derivative (PD) controller, this control structure implemented by assisting of special Arduino Simulink library In MATLAB Simulink, where it will be used for control the signals transition, a new Simulink design proposed and investigated to interfacing the controller with the arm motors and encoders via the circuits. The second contribution is adding an assisting spring system to the arm first motor joint, where this addition rises the motion stability through the control phase by increasing the required torque in the arm first joint, especially after the human hand load is attached to the robot links, where its tested effectively without needing for replacing the motor with increased torque. The results illustrate the power of these circuits design and the error reduction occurred after supporting the system's spring. The Third contribution, An Internet of robotic thing (IoRT) model is designed, and then the robot arm is connected using our designed internet –based control system using IoT protocols for communication.

**Keywords:** 2DoF's assistant robot, IoRT, Arduino board as interfacing card, Fuzzy controller.

## 1. Introduction

The assistant robot is a program used for therapy for the persons who have suffered from injuries in the nervous system, stroke, brain trauma and sports. International Federation of Robotics (IFR) defines a disability arm as a robot implemented for help injures who have physical disordered that impede life activities [1] . The area of expertise that robots is known as "disability robotics". Disability arm used to help people who are reordered them limbs after strokes of injuries that affect their life tasks. Several types of research deal with modeling and design of controlled upper limb assistant robots that are used in human limb assisting program. Upper limb exoskeleton robot is another name for the assisting robot, these robots are proposed with several designs.: these exoskeleton robots have been used as an assisting device, a human amplifier or a haptic interface. Most of them have less than 7 DoF's. The following research deal with upper limb assisting robots that were implemented previously. For modeling or controlling assisting robot, many ideas were proposed based on the motion geometry and controller type. The illustrated literature present verity kinds of ideas that used to model and control of many structures of assisting arm: De-la-Torre, Ruben, et al. in 2020, illustrated study for therapy by assisting robots which involved in the activity of daily living for the patient [2]. The drawback was in the workspace limitations in motion with all human hand joints. In 2021, Zhang et al., proposed impedance-based controller for assist-as-needed in robot motion that can be used in assisting robot arm for rehabilitation training in human upper extremity dysfunction [3]. Argall and Brenna present in 2018, the introduction of clinically feasible autonomy solutions for rehabilitation robots, and opportunities for autonomy within the rehabilitation domain, where they presented the full required designs for upper or lower limbs and what the robot autonomy required with each case whether wearable or non-wearable [4]. Aitziber et al, proposed in 2018 the effective sensors, were used as a substitute for the actual force and motion sensors of the proposed assisting robot that used for upper limb recover. These virtual sensors established the required stimulates of force and motion at the contact points, where the person interacts with the assisting arm using the arm mathematical model [5]. In 2018, Monica, et al, provide a control structure used the electromyography signals for drive the joint motion and control the assisting arm. The relation analysis for electromyography signal and the subject force exertion was noticed. The authors provide surface of electromyography force-based effective control structure that can control the force exerted by the assisting arm through the training [6]. In 2019, research concerning

*E-mail address: zyad.karam@nahrainuniv.edu.iq, mustafa.kadhim@coie-nahrain.edu.iq*

design rehabilitation robots witnessed major breakthroughs. For example, Mohammad Hossein, et al., presented a under lab implemented robot and testing the accuracy by evaluation of five structures for testing of human-assisting arm interaction torques. These structures involve the simplest idea of using direct control for motor torques with consideration of the arm dynamics, also involves advanced ideas with full arm dynamics such as inverse dynamics [7]. Edwin Daniel, et al., present an ordered review of arm-based systems focused on upper human body rehabilitation, where these systems adoption in clinical training remained limited. To understand the reasons of these limitation's, they created method for self-adaptation used for personalizing the training, of injured–assisting arm interaction, where it contributes in training of movement generation factors [8]. Brock, et al., presented an authenticated investigation based on machine vision with deep neural networks, where it's used for read the environment to assist the predictive structure for control the robotic lower-limb [9]. Leiyu, et al., developeda prototype of ankle rehabilitation assisting robot with three degrees of freedom's, around a virtual fixed ankle joint as center. The ankle center should harmonize with the virtual fixed center through the rehabilitation training. Also, a full information acquisition design was implemented to provide the human-machine interactive for the assisting arm, patients, and therapists [10]. In 2014 and 2015, another creative design was established by Hassan and Zeyad, where they designed a 3 Dof's assisting robot arm for upper left limb to assist the injured persons who suffer from losing control in their arms. The designed robot is a wearable one and it has a superior mechanism for adjusting the robot with multi human arms lengths, also it interfaced with PC using Advantech card via specially designed controlling circuits. The presented robot was managed by using a novel control strategy of force position controlling algorithm that incorporated a fuzzy type one as a position controller. The controller parameters were optimized using a modified particle swarm optimizer PSO. The presented robot was tested with multi human hand weights and entered the phase of injured person training and showed superior results in the execution of the medical trajectories [11], [12], [13]. This work includes the design and implementation of 2 DoF's exoskeleton robot for assisting and training the human upper left limb as a non-wearable assisting robot. The mechanical design and implementation are accomplished. The robot involves 1 DoF for shoulder motion and 1 DoF for elbow motion. Two interfacing circuits based on Arduino card will be designed and implemented; one for reading a DC motor's position from the shaft encoder sensor and another to drive the motors to the desired position using control signals. APD fuzzy type 1 controller architecture in Matlab Simulink will be designed to control the robot motion according to the desired treatment trajectories via the connection with the Arduino card interfacing simulation library. The other sections of this article are arranged as follows: section 2 discusses the IoRT paradigm, while section 3 deals with the mathematical model of the proposed 2 DoF's assisting robot. Section 4 highlights the proposed mechanical design and its implementation. Section 5 explains the proposed interfacing circuits design in two parts: one for reading the motor position sensors and the other for driving the motors based on the controller signals. The section also presents the interfacing with the Arduino card simulation library. Section 6 shows the design of the fuzzy type 1 controller and its simulation with robot kinematics, while section 7 aims at showing the results of a simulated robot kinematics response and the real-time robot position response for the implemented robot. Section 8 explains IoRT implementation model. Finally, section 9 is dedicated to the conclusion of the design. IEEE technical committee of automation and robotics defined the robotic system connected via network as a group of controlled devices that are linked together via wireless or wired communication network [14], [15]. Also, robotic networked applications can be one of the two cases: It can either be 'teleoperated', which means they are remotely controlled by a person from a network to achieve the given task. One example of such systems is the Mars rover robot. Its picture shown in 1 below:

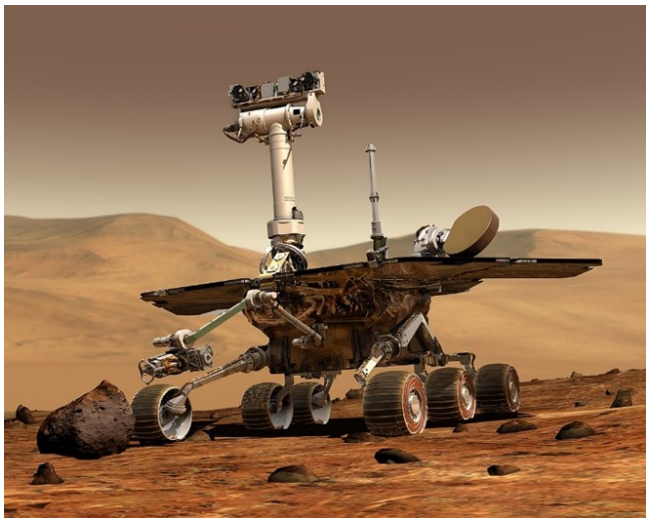

Figure 1. Mars rover robot

Or it can be a 'multi-robotic system', which is a collection of network-accessed robots that are self-cooperate by exchanging data collected using sensors mounted on the robots. A good example of a multi-robotic system is a swarm of drones used in Shanghai to welcome the new year 2020 as shown in 2

One of the problems of the network operated robots was the inherited latency because they relied on network operation, and the command execution time would be affected by that latency. Another problem that the network operated robots' faces is the physical constrains of the size of the processing chip the robot carries on board, which limit the speed and efficiency of the robots. One current solution for the problems is to use what is called 'Cloud-based' robotic system, in which the robotic system will

Figure 2. drones working together to form a walking man in the sky of Shanghai.

depend on the processing power of the cloud to overcome the physical limitation that used to limit the processing power of the robot. an application of such systems is the Google car with self-driving system used to index Google maps. Although cloud-based robotic system makes use of the big data processing capability of the cloud and the cloud computing paradigm, it still suffers from many other issues such as security, latency, and standardization, to name a few. Because of all the previously mentioned issues regarding the implementations of a network or cloud based robotic system, we use the Internet of robotic things paradigm. It can efficiently attack and address each of the mentioned issues in both previous implementations because it is based on Internet of Things (IoT) architecture [16].

## 2. THE IORT PARADIGM

Internet of robotic things is positioned as a top layer over the cloud robotic paradigm. Basically, it consists of five distinct layers of operation. A brief description for each layer will be given below:

### *A. The Hardware Layer*

These are the sensors, smartphones, vehicles, actuators, and other physical real-life components.

### *B. The Network Layer*

In this layer cellular connectivity and other communication and networking technologies are described, such as 3G, 4G, WIFI, NFC and LORA [17]

### *C. The Internet Layer*

In this layer specific IoT protocols are added such as MQTT, XMPP and LLAP, to name a few. This is also the layer that defines the IoRT paradigm.

### *D. The Infrastructure Layer*

This layer consists of many sub-layers, when working together they form the whole infrastructure for the IoRT paradigm. These sub-layers are robotic platform, M2M2A, IoT business cloud sub-layer and IoT cloud infrastructure robot.

### *E. Application Layer*

This is the highest layer of the IoRT paradigm, it includes applications that the robots can perform in the following domains: healthcare, datacenters, business and many more. The below 3shows our implementation using the mentioned IoRT architecture.

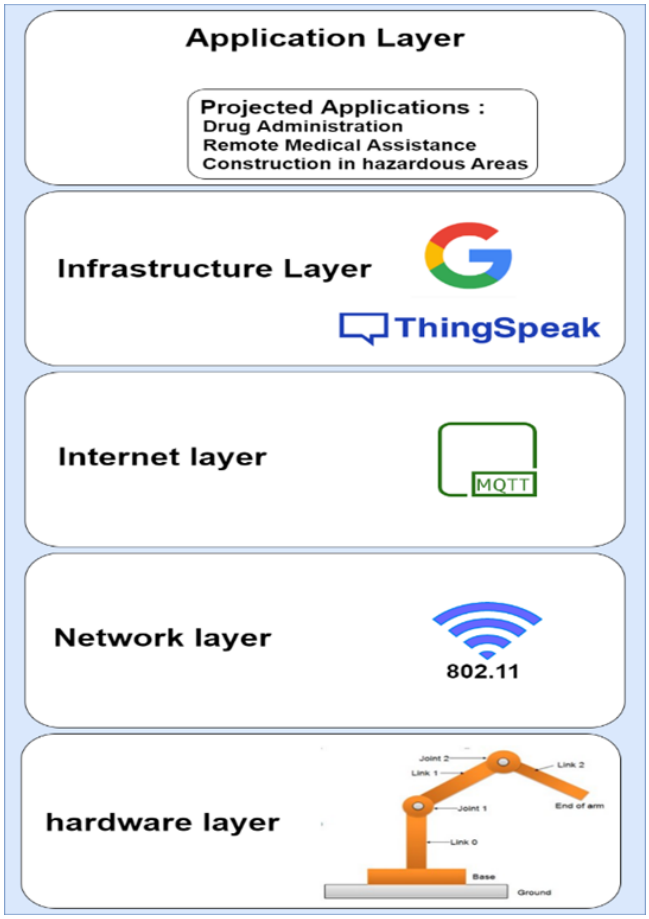


Figure 3. The proposed IoRT implementation architecture

## 3. MATHEMATICAL MODELING

Kinematics modeling is the basic representation of the robot motion. Kinematics known as the principle of the robot motion. Where its useful in calculate the end tip positions, joint velocity, and arm acceleration without considering of the forces bringing through movement. Their schematic in the x-y plane is shown in 4 [18].

For the kinematics investigation, there exist two general inquiries to assess the end-effector position. With every joint angle and arm link parameters, the position and orientation can be assessed for the end tip with respect to the base reference frame. This case is presented as forwarding kinematics.

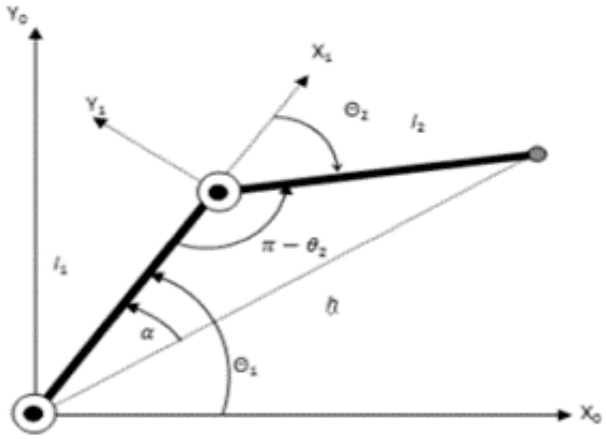


Figure 4. The Forward and Inverse Kinematic of 2 DoF robot arm

Therefore, for a calculating the position with orientation of the end tip of the robotic arm, one can calculate the joint angles. This case is presented as inverse kinematics. Utilizing the "Denavit–Hartenberg" transformation, by starts in processing the case of forwarding kinematics, when any link changes are causing transformation depicts the position and orientation of all arm geometry. The general kinematics of any arm is:

### A. Forward Kinematics

The D-H theorem contains four parameters which are: angle, the link/phalanx offset *di*, the link /phalanx length, and the link/phalanx twist *ai*, These factors are used to constrict the position and orientation of the arm end tip. The forward kinematics inputs are the joint angle vectors (i) with the parameters of links length. The Output is the direction control and the position control of the end arm tip [18]. Table (1) summarizes the D-H parameters. Also, the D-H equations are:

Table (1): D-H parameters of the links.

| Joint | Twist Angle ai | Link Length Li | Offset Distance di | Angle $i$ |
|---|---|---|---|---|
| 1 | 0 | L1* | 0 | 1 |
| 2 | 0 | L2* | 0 | 2 |

$$P_x = L_1 \cos \quad + L_2 \cos(\ _1 + \ _2) \tag{1}$$

$$P_y = L_1 \sin \ _1 + L_2 \sin(\ _1 + \ _2) \tag{2}$$

$$pz = 0 \tag{3}$$

### B. Inverse Kinematics

Inverse kinematics, where the link orientation and position considered as input with parameters of links length. The Output will be the vectors of joint angles ( i). The Inverse kinematics depends directly on the forward kinematics, to obtain the inverse we will use the equations 1 and 2 of the forward kinematics [19].

$$_1 = \arctan 2\ \ p_y; p_x \quad \arctan 2\ \frac{L_2 \sin \ _2}{L_1 + L_2 \cos \ _2} \tag{4}$$

$$_2 = \arctan 2\left[\frac{p\sqrt{1 - D^2}}{D}\right] \tag{5}$$

### C. Third-Order Polynomial Trajectory Planning

In this application, the initial location and orientation of the robot are known and using the inverse kinematics, which finds the final joint angles for the desired position and orientation. Now consider one of the joints, which at the beginning of the motion segment at a time $t_i$ is at $_i$. that desire to have the joints move to a new value of $_f$ at time $t_f$. These four pieces of information allow us to solve for four unknowns (or a third-order polynomial) in the form of:

$$(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 \text{At} = 0 \quad t = t_j zero \tag{6}$$

$$c_0 = \ _i \tag{7}$$

$$c_1 = 0 \tag{8}$$

$$c_2 = \frac{3(\ _f \quad _i)}{t^2 f} \tag{9}$$

$$c_3 = \frac{2(\ _f \quad _i)}{t^3 f} \tag{10}$$

## 4. DESIGN AND IMPLEMENTATION

This part shows the design and connection of the assisting robot as shown in figures 5, 6. Manipulator Design: The mechanical robotic arm is made up of series combinations for the links and joints. The links are the solid parts that linked to the joints. The motion of these links based on joints, which are the movable components of the arm. Joint Design: The joints (also called axes) are the movable components of the robot that cause relative motion between adjacent links. For the joint motors to be able to move efficiently while carrying the subsequent links, joints and even the human arm, we choose a motor with high torque output due to its powerful gearbox the DC motor operates on 12-volt DC input. The two motors used in this work are brushed DC motors with a gearbox for high torque. The motor of the first link is highly inertia motor with a gearbox ratio of 1:99, rated current of 6 Amp and 12-volt DC supply. This motor is connected to 3900Pulse/rotation incremental shaft encoder in the back-motor shaft. Its body is connected to the robot base that is located on the head of the adjustable mechanism and its shaft is connected to the iron flange to transfer the motion to the first robot link. Link Design: The links are the solid parts that linking the joints. The link is designed using variable-length lightweight aluminum to fit the various lengths of the patient arms and be light enough as not to pose a burden on the patient that is

attached to it or add extra weight on the motor and the robot that is carrying it. The assisting robot links are made from Aluminum metal. The use of Aluminum of rectangular bare shape of dimensions (2cm*4cm) is for its lightweight that should be considered in robot links design. The links design is non-wearable to avoid adding additional stress to the patient whose arm is already disabled. Figure 7 shows the implemented hardware design. For the first joint, two springs are added for assisting the first motor in carrying the weight of the human hand and the rest of the robot parts, and the result of this addition will be presented in the results section. As shown below, Link 0 with a fixed length of 60 cm, link1,2 (L1, L2) with variable length range (30-45 cm). , Link 0 with a fixed length of 60 cm, link1,2 (L1, L2) with variable length range (30-45 cm).

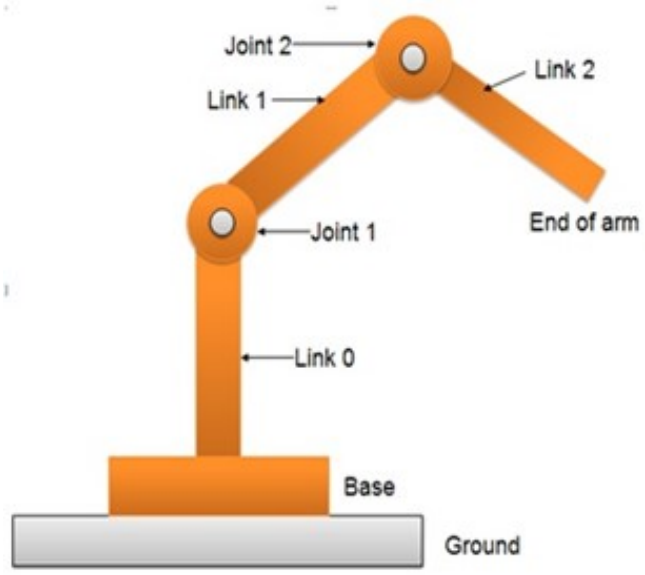


Figure 5. Robot Part

## 5. INTERFACING CIRCUIT DESIGN

The interfacing of the designed circuit for robot sensing and controlling its motor will be implemented via Arduino Mega.fig. 8 shows the general interfacing block diagram between the robot motors and sensors with the simulated controller in MATLAB simulator via the Arduino interfacing card and its special interfacing library.

The Arduino Mega is an open-source embedded system based on easy-to-use in hardware interfacing and software programing. The Mega 2560 is designed for more complex projects interfacing which are implemented via 54 digital input/output pins (of which 15 can be used as PWM (pulse width modulation) outputs) 16 analog inputs. The Arduino Mega 2560 is a microcontroller board based on the ATmega 2560. It has 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB port, a power jack, an ICSP header, and a reset button. The Atmega microcontroller instilled on the board of Arduino mega is programmed using IDE

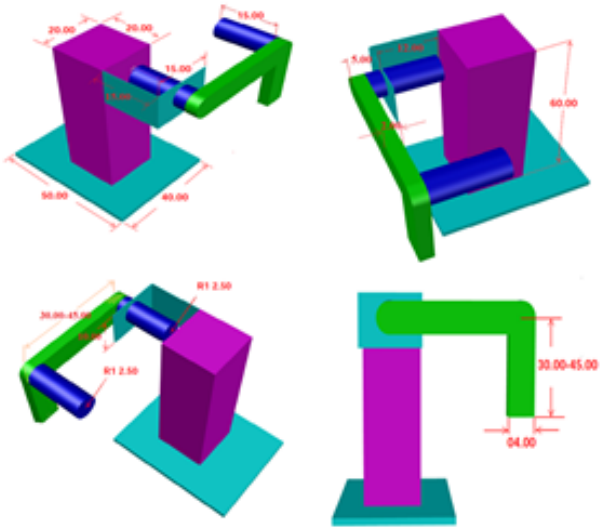

Figure 6. Design of assisting robot parts with auto cad from different angles.

Figure 7. 2 DoFs assisting robot prototype.

language [19] and [20]. The interfacing circuits are divided for two circuits; one for reading the motor sensors and feed its data to the Simulink controller via Arduino, the second one is the controlling and driving circuit used to send the Simulink controller control signals via Arduino card.

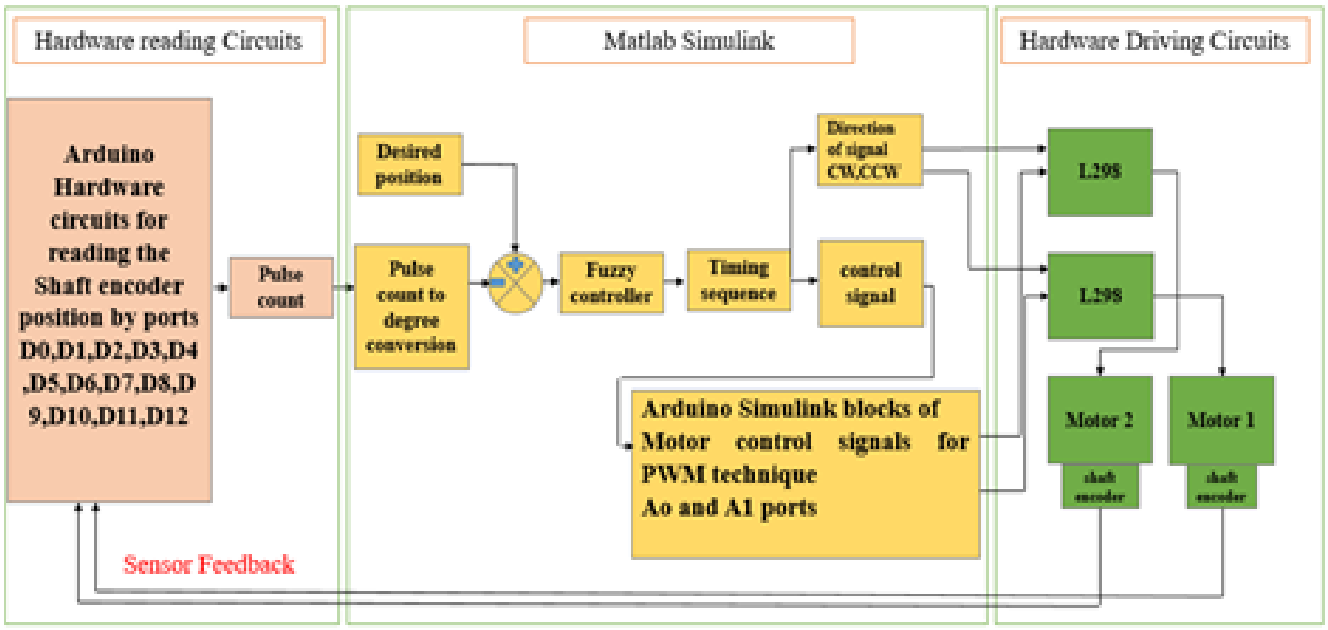


Figure 8. Block diagram of the designed interfacing system.

### A. *Sensors Reading Circuits*

The reading circuit design involve a D-flip-flop chip used for convert the two input phases that counted by the incremental shaft-encoder, (A and B), where it utilized to read the motor shaft rotation in clockwise CW or counter-clockwise CCW. Also, the AND gate's purpose for convert the sensed direction pulses to the up/down counters. The timing diagram, shown in fig. 9, explains this operation. The profuse simulation for this circuit is shown in fig. 10.

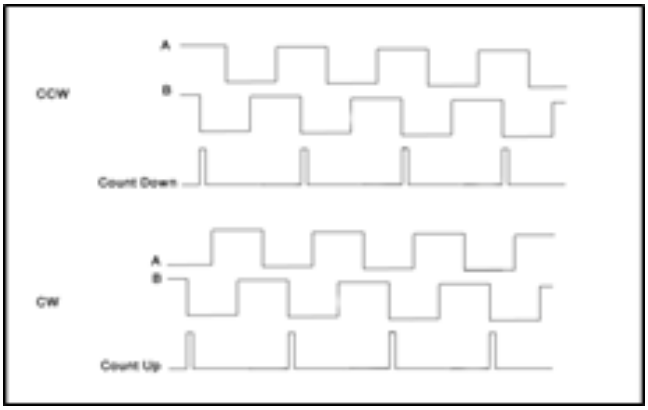


Figure 9. The PWD used to determine clockwise or counterclockwise motor movement

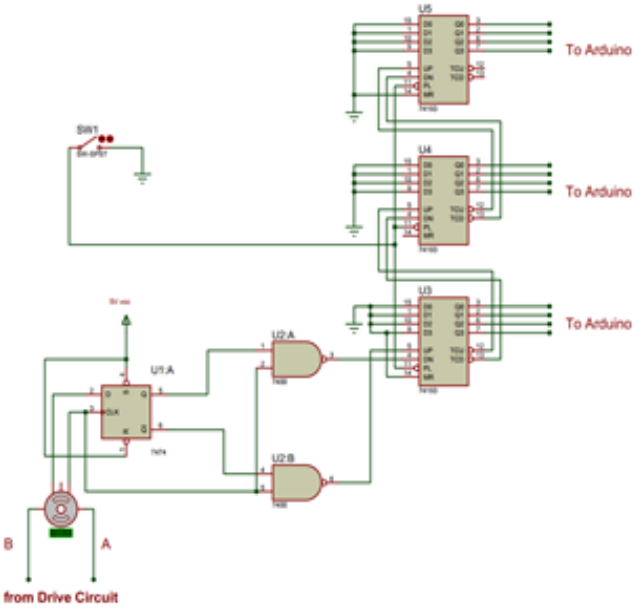


Figure 10. Simulation of reading circuit for each motor

### B. *Driving circuit*

The driving circuit consists of two L298 drivers controlled by the Arduino Mega needed to drive the two 12V motors. The L298 drivers are used to provide the needed voltage, because the Arduino Mega cannot provide 12V to drive the two motors e. Each driver circuit is connected to H-bridge with freewheeling diodes to save drivers from induction current that is generated in motors after cutting of driving (control) signal. Fig. 11 shows the design of these circuits using a proteus simulator. Fig. 12 shows the flowchart of the wire connection between Reading (sensor circuit) and Driving (controlling) Circuits [20].

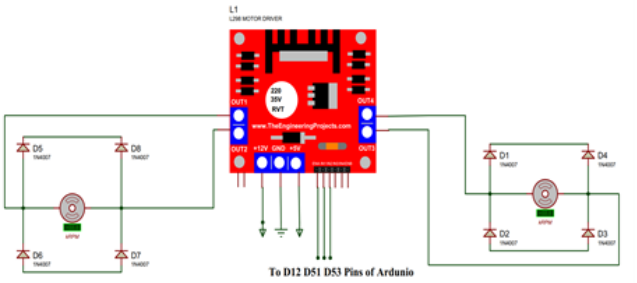


Figure 11. Simulation wire connection by Proteus of the Driving Circuit

## 6. FUZZY CONTROLLER DESIGN AND SIMULATION OF THE ROBOT KINEMATICS

In this work, the fuzzy type one, PD like fuzzy logic position controller design for controlling the required trajectory of motion in each joint based to reference degree of rotation. The PD in FLC structure is chosen to achieve the minimized error in controller signal, that done by minimize the rise, peak, settling times and response overshoot. So, the proposed controller equation will be [21] and [22]:

$$u(t) = K_p e(t) + K_d \dot{e}(t) \tag{11}$$

The inputs to the FLC are e(t), which is the error and which is the change of error and $K_p$, $K_d$ which are the proportional and derivative gains of error signal respectively.

The FLC is the Mamdani type. The I/O puts a membership function used are seven, each one in shape of triangles and Gaussian, with universe of discourse of [-10 10] for both input and output, where this range will be adjusted by proposed gain $K_o$. The mechanism used in defuzzification process is the center of the gravity. fig 13 shows these memberships for I/O puts, table 2 lists the chosen rules for building the FLC position controllers which are selected by trial and error. The used linguistic for memberships are NB, NM and NS, where N is Negative, M is Medium and S is Small. Z is Zero. PS, PM PB, where P is Positive, B is Big. The designed Simulation modelled by MATLAB, which involves the PD in FLC structure as the controller for the 2 DoFs assisting arm [23] and [24]. The designed robot will be used for controlling the real-time implemented robot via the designed interfacing circuits.

Table (2): Rules of PD-like position FLC.

| e | NB | NM | NS | Z | PS | PM | PB |
|---|---|---|---|---|---|---|---|
| NB | NB | NB | NB | NB | NM | NS | Z |
| NM | NB | NB | NB | NM | NS | Z | PS |
| NS | NB | NB | NM | NS | Z | PS | PM |
| Z | NB | NM | NS | Z | PS | PM | PB |
| PS | NM | NS | Z | PS | PM | PB | PB |
| PM | NS | Z | PS | PM | PB | PB | PB |
| PB | Z | PS | PM | PM | PB | PB | PB |

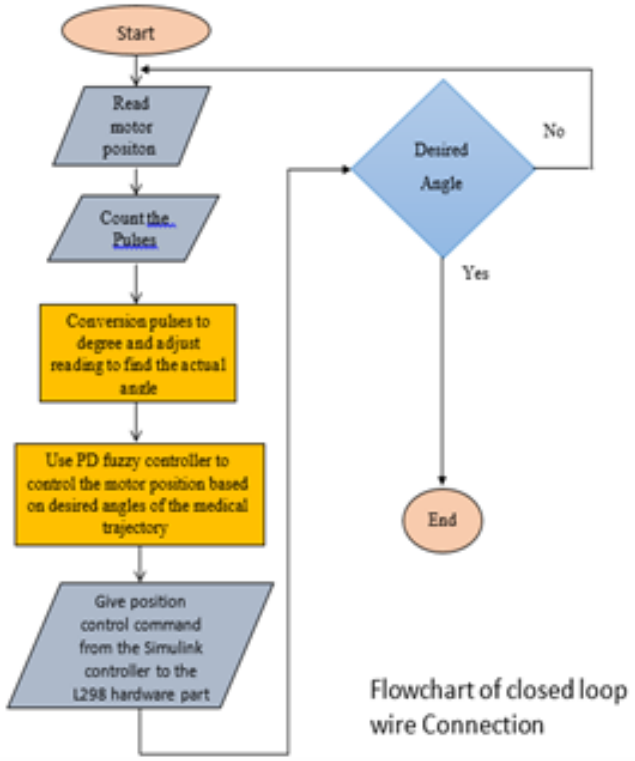


Figure 12. Flowchart of wire connection for the interfacing circuits

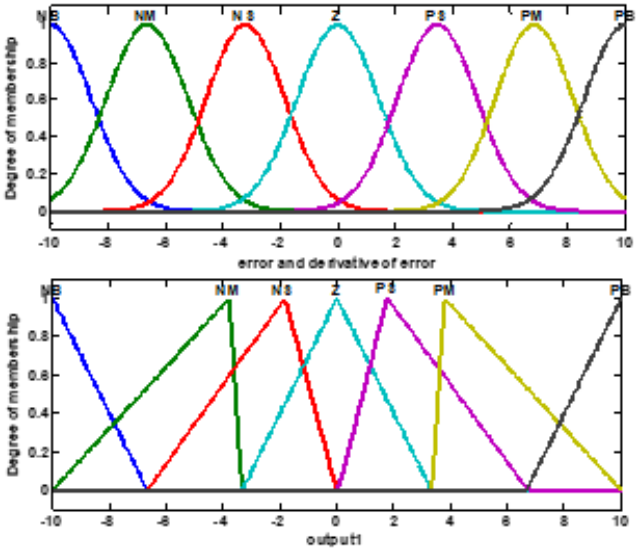


Figure 13. Inputs and output Membership functions.

The kinematics model simulation of the forward and inverse kinematics also involves checking the proposed robot motion before designing the robot for the assisting job. This testing is done based on a specified joint angle and is related to the presented trajectory equations. Figure 14 illustrates the simulation of the forward and inverse kinematics based on specific desired joint angles' trajectory. robot motion before designing the robot for the assisting

job. This testing is done based on a specified joint angle and is related to the presented trajectory equations. Figure 14 illustrates the simulation of the forward and inverse kinematics based on specific desired joint angles' trajectory.

## 7. SIMULATION RESULTS OF THE SIMULATED AND IMPLEMENTED ROBOT

This part shows the Simulink simulation result of kinematic model based on the reference trajectory, where the proposed trajectory represents part of the medical training movements or any other desired trajectory to be executed by the robot as an assisting process. Also, this trajectory is applied to the implemented robot with applying load of real human hand of 3.78 Kg. Where this load represents the patient hand that requires to be assisted by the robot as a disordered person or as a helper to the person how execute specific process. Kinematic Simulation Results Table (3) and (4) show the result of the forward and inverse kinematics [25].

Table (3): Test of the kinematics of forward and inverse for the robot when l1=45, l2=45.

| Inverse Output | | Forward Output | | Theta | |
|---|---|---|---|---|---|
| 2 | 1 | $P_y$ | $P_x$ | 2 | 1 |
| 0 | 90 | 35 | 35 | 90 | 0 |
| -90 | 0 | -70 | 4.286e-15 | 0 | -90 |
| 0 | 45 | 24.75 | 59.75 | -45 | 45 |
| 45 | 90 | 49.5 | 3.553e-15 | 90 | 45 |
| -135 | 45 | -59.75 | -24.75 | -45 | -90 |

Table (4): Test of the kinematics of forward and inverse for the robot when l1=35, l2=35.

| Inverse Output | | Forward Output | | Theta | |
|---|---|---|---|---|---|
| 2 | 1 | $P_y$ | $P_x$ | 2 | 1 |
| 90 | 0 | 45 | 45 | 90 | 0 |
| 0 | -90 | -90 | 6.858e-15 | 0 | -90 |
| -45 | 45 | 31.82 | 76.82 | -45 | 45 |
| 90 | 45 | 63.64 | 7.105e-15 | 90 | 45 |
| -135 | 45 | -76.82 | -31.82 | -45 | -90 |

Where $_1$ and $_2$ are the angles of the first and second joints respectively, Px and Py are the end-effector position. These results provide the modeling for the forward kinematic through the inverse kinematics results. Also, the kinematics is tested based on the proposed reference trajectory of a third-order parabolic polynomial. The values of theta result show in figures (15 and 16) respectively. The simulation time takes 100 seconds in this test.

The upper part of figures 15 16 represents the desired angle trajectory planning of the first and second robot joints by 90[U+F0B0] and the lower part represents the motion of the end effectors in (X, Y) planes through the trajectory execution. The implemented robot test without adding the human hand load with references to the desired trajectory is presented in fig. 17 The testing time chose to be 20 seconds as an appropriate time for reference medical therapy, the testing angle for the first motor is 120°. The required controller signal presented in fig. 18

The presented results for the robot arm response without human hand load, shows a type of a smooth response in figure 17, where its signal with acceptable range of oscillation by (4.5°), the presented control signal in fig. 17 shows a ranged signal with actuator range of ( 10 volt). The second phase of the experiments is adding the human hand weight to the robot by attached to the designed arm links, by a weight of 3.78 Kg, the results for both joints are presented in figures 19 and 20 respectively. The testing angles are chosen to be 90°for both motors.

From both figures, one can notice a part of oscillation in the joint motor responses after being attached to the human hand. This oscillation in robot joint results from the addition of the external weight of the human hand, which consumes the ability of the controller to track the desired trajectory. Specially the range of oscillation in some segments being ( 6°). To solve this problem, there is need for increasing the torque required by motors, which means using highly torque motors where that results in extra costs and the device becomes unavailable for all patients. Accordingly, a two spring's system are added to the first joint motor through movement, where it adds an extra tension in robot first link that results in enabling the first joint motor from overcome the required torque. These springs are added between the base of the first motor and the end of the first joint along the length of the first link. Furthermore, these springs result in damping the retraction phase when the first motor return to its initial position. The following mathematical equations will be modeled after adding the springs and will be presented in fig. 21

$$K = K_A + K_B \tag{12}$$

$$I_t = I_{l1} + I_{l_2} + m_{l2}\,(l_1 + l_2)^2 \tag{13}$$

$$I_e = \frac{1}{I_t}\left[ I_{l1} I_{l_2} + m_{l2}\ \ I_{l1} l_2^2 + I_{l2} l_1^2 \right] \tag{14}$$

$$\begin{bmatrix} I_{l1} + m_{l2} l_1^2 & l_1 l_2 m_{l2} \\ l_1 l_2 m_{l2} & I_{l2} + m_{l2} l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\ } \\ \ddot{\ } \end{bmatrix} + \begin{bmatrix} 2K & 2K \\ 2K & 2K \end{bmatrix} \begin{bmatrix} \ _1 \\ \ _2 \end{bmatrix} = \begin{bmatrix} T \\ 0 \end{bmatrix} \tag{15}$$

$$G_1(s) = \frac{_1(s)}{T(s)} = \frac{I_{l2} + m_{l2} l_{l2}^2\ \ s^2 + 2K}{I_t s^2\ \ I_e s^2 + 2K} \tag{16}$$

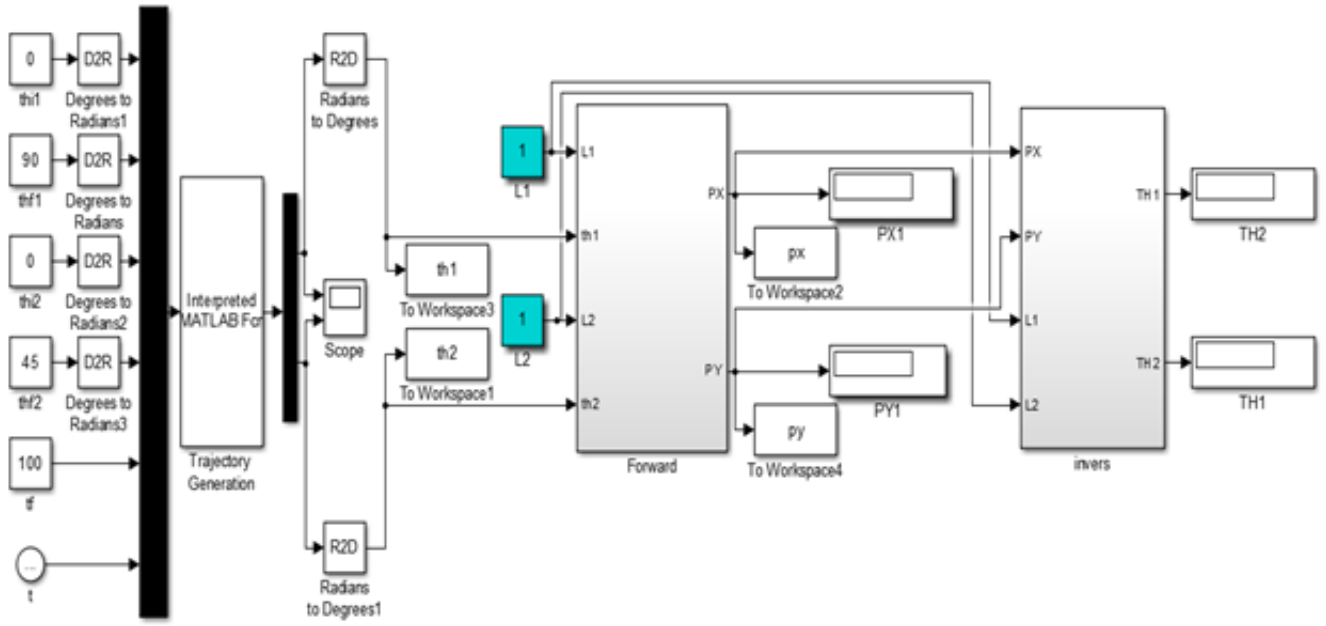


Figure 14. The assisting robot trajectory Simulink.

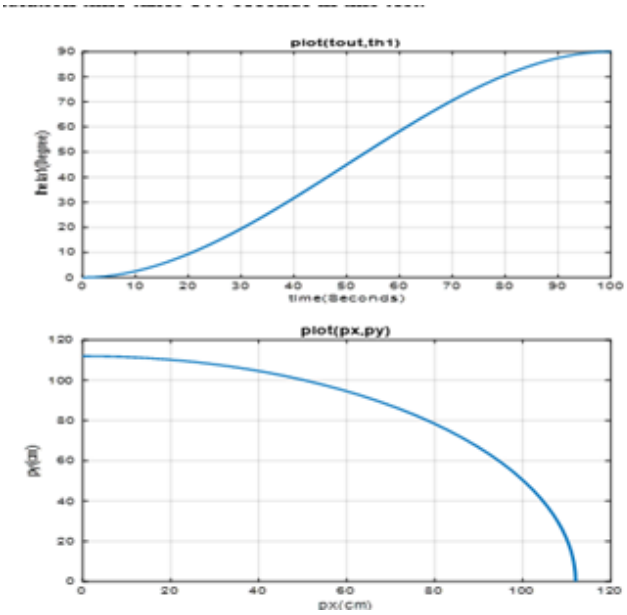


Figure 15. Trajectory plot when $_i1$=0, $_f1$=90, $_i2$=0, $_f2$=0

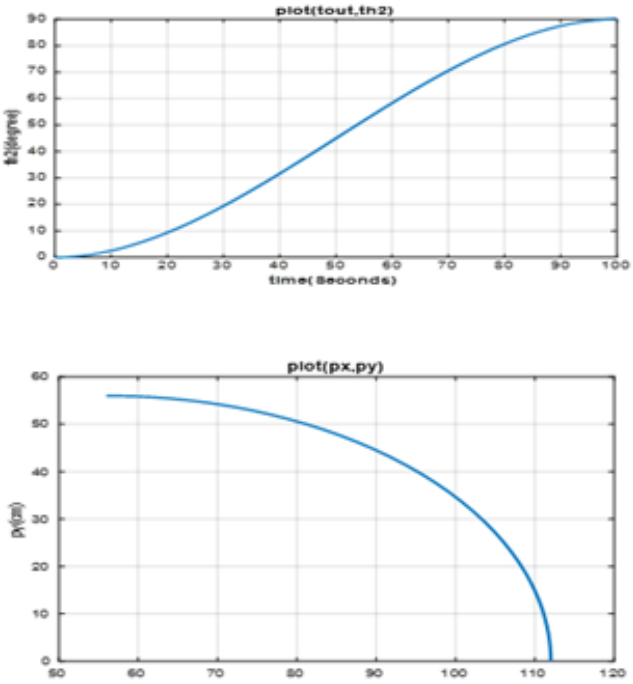


Figure 16. Trajectory plot when $_i1$=0, $_f1$=0, $_i2$=0, $_f2$=90

$$G_2(s) = \frac{_2(s)}{T(s)} = \frac{I_{l1}I_{l2}m_{l2}s^2 + 2K}{I_t s^2 \ I_e s^2 + 2K} \qquad (17)$$

Where 2K is the spring coefficient of two springs positioned in parallel with a length of 25 cm and coefficient value 1.177 99 N ·m. It is moment of inertia for actuators mechanical parameters, I.e., is the moment of inertia of the actuator electrical parameters, $I_{11}$ $I_{12}$ are the moments of inertia of the two links respectively, $m_11$ $m_12$ are the masses of the two links respectively and $i_1$ and $i_2$ are the lengths of the two links, respectively. Fig. 21 presents the robot joint motors response after spring system addition for the first joint, where it assists the first motor tightening in the return phase through the execution. **??** presents the

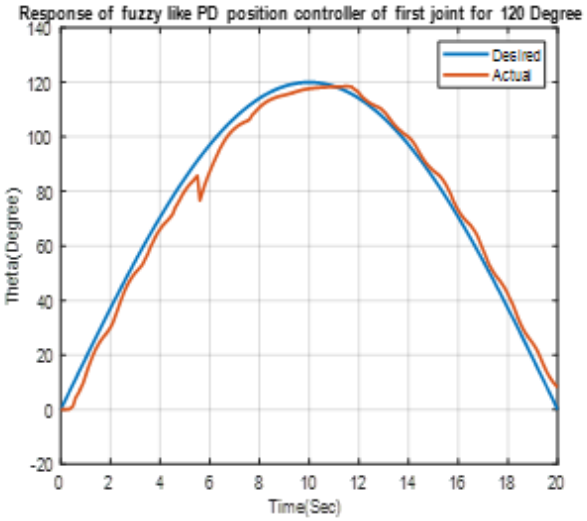


Figure 17. Response of fuzzy PD like position controller of the first joint with the input of 120°

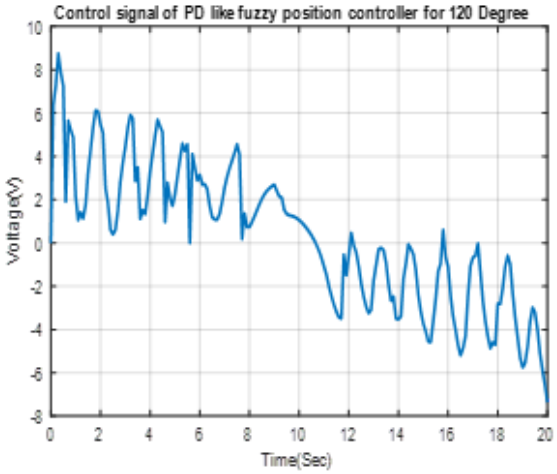


Figure 18. Response of fuzzy PD like position controller of the first joint with the input of 120°

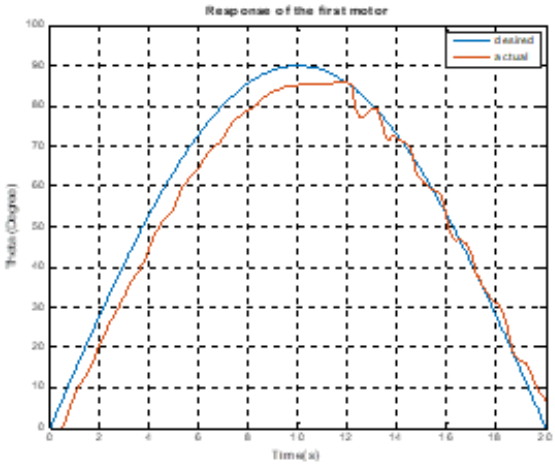


Figure 19. First joint response after attaching the human hand with the input of 90°

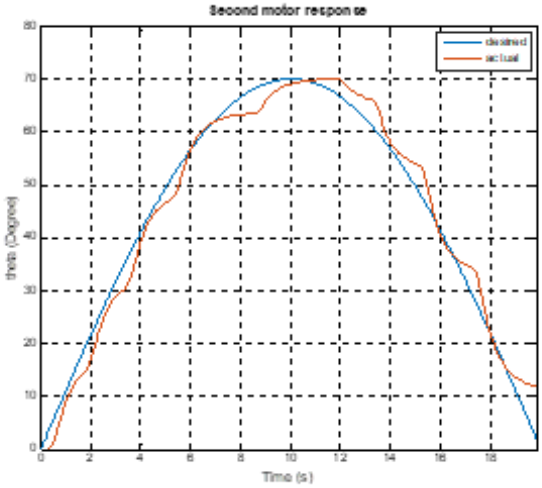


Figure 20. Second joint response after attaching the human hand with the input of 70°

robot joint motors response after spring system addition for the first joint, where it assists the first motor tightening in the return phase through the execution of the medical trajectory. This assisting results in strengthen the first link to overcome the required force to hold the rest of robot mechanical parts with attached human hand. The adapted results after adding the springs system in presence of the human hand load to the robot links are presented in figures 23 and 24 respectively.

After comparing Figures 19 and 20 with Figures 23 and 24, one can find an adapted response in both joint motors while a human hand is attached to these links. Where the error minimization result in acceptable therapy trajectories. Table (5) illustrates the error presented in joint motor responses before and after using the spring system.

Table (5): Error percentage before and after adding the spring system.

| Case | | Error in response (degree) | Percentage of enchantment |
|---|---|---|---|
| Before adding the spring system | First | 5.89 | |
| | Second | 5.28 | After addition of the spring system |
| After adding the spring system | First | 4.12 | 29.572% |
| | Second | 4.07 | 22.916% |

The above table illustrate the oscillation minimization in robot joints through the presence of the load, that means the designed spring system is working effectively, where it increases the required torque by the first joint motor in the execution phase and damping the links motion through the retraction phase, all of these results overcome via the power of the PD fuzzy controller that can overcome the oscillation with wide range of universe of discourse in input and output signals. By compering this work with

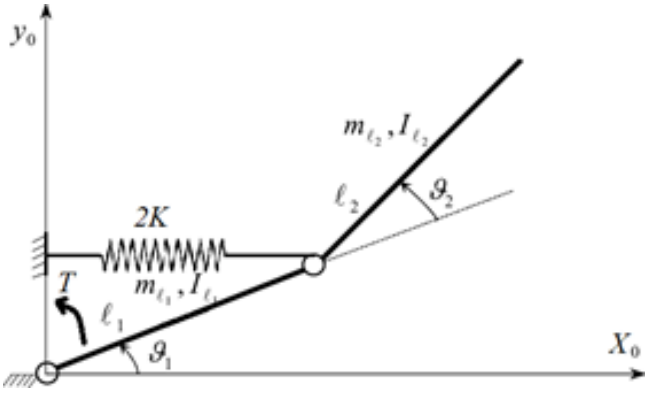


Figure 21. Analytic representation of the two-link robot with spring addition°

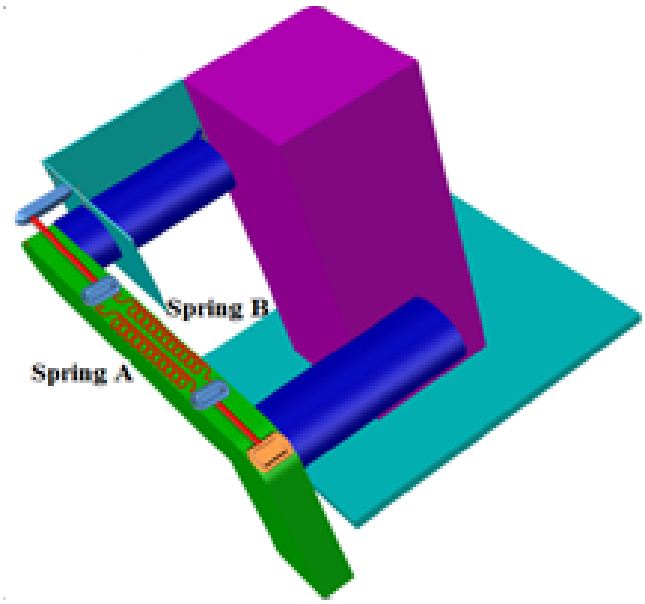


Figure 22. The proposed mechanical spring system for assisting the first motor

works of ref. 11, 12 and 13 one can find an optimized results in in robot joints error as shown in table (6). Where the compared works used the fuzzy system and particle swarm optimization technique for controller fitting based on highly torque motors, while the proposed work depends on manually fitted fuzzy controller with normal motor torque augmented by the designed spring system.

Table (6): Percentage of enchantment in proposed robot joints in compare with references 1, 2 and 3.

| Joint angle | Error in response (degree) Proposed work | Error in response (degree) compared work | Percentage of enchantment |
|---|---|---|---|
| First | 4.12 | 5.930 | 30.522% |
| | 4.07 | 6.023 | 32.425% |

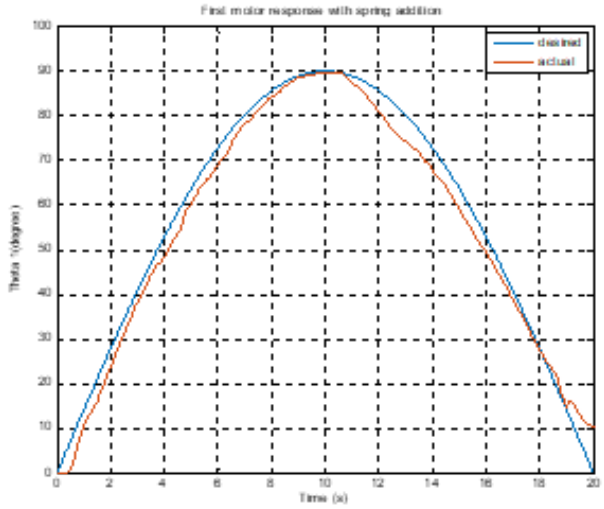


Figure 23. First joint response after attaching the human hand and spring system via 90°trajectory input

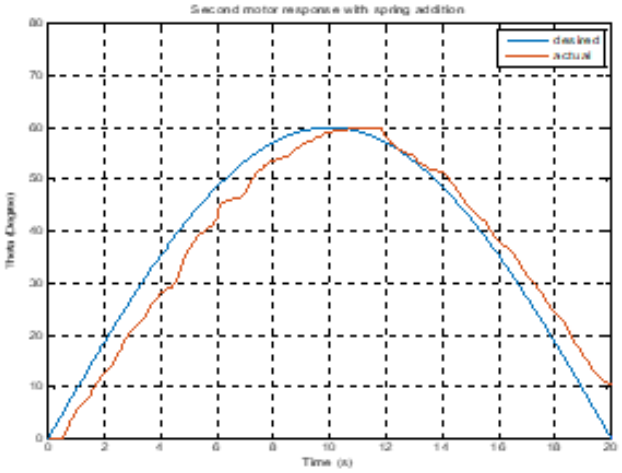


Figure 24. Second joint response after attaching the human hand and spring system via 90°trajectory input

## 8. IORT IMPLEMENTATION MODEL

A webcam is connected to the robot model to give visual feedback and the robotic system is connected to the wireless router using Wi-Fi shield for Arduino. The data is sent from the robotic system using MQTT as a lightweight protocol designed for IoT applications. The cloud model is implemented using thing speak for data aggregation and visualization and JSON and REST for commanding the robotic arm as an API with the controlling web application. Finally, the control web app is designed using bootstrap, JS and node. The complete IoRT model implementation is shown in fig. 25 below.

## 9. CONCLUSIONS

In this work, the mechanical with electronic design has been implemented for two DoF's assisting robot. The

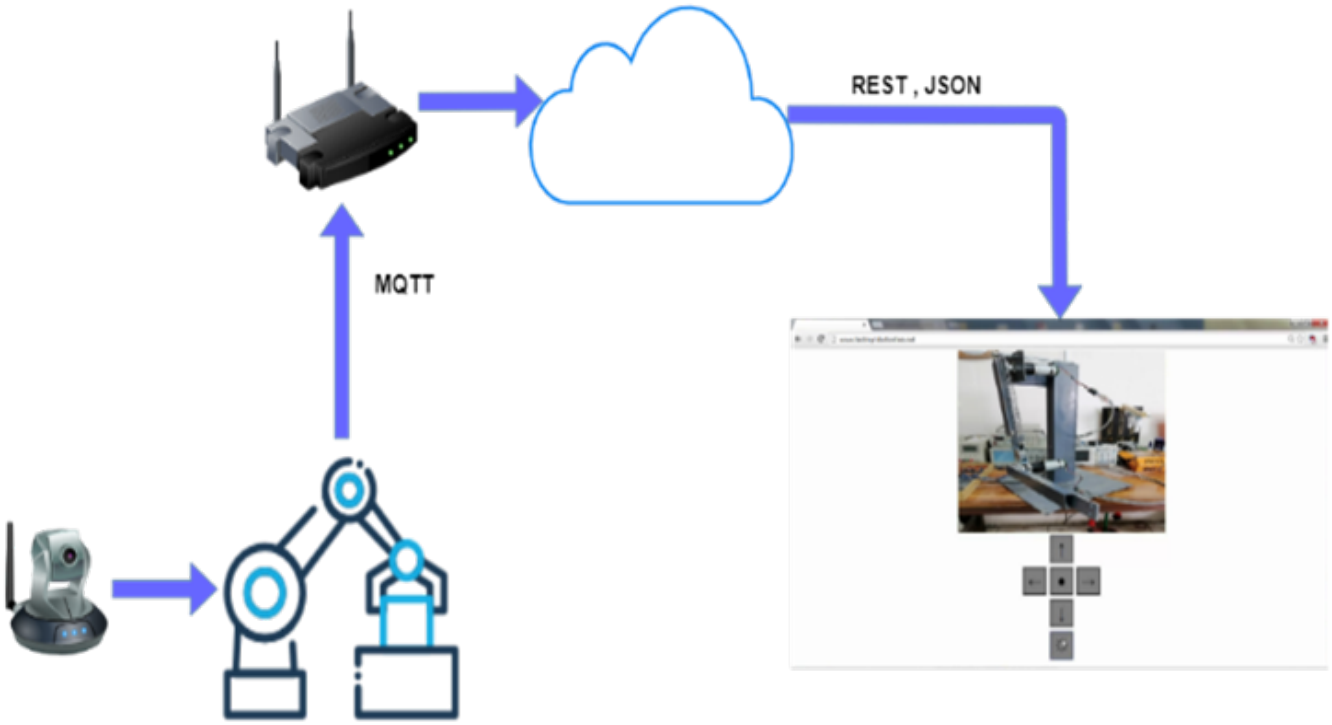


Figure 25. the proposed IoRT implementation model

designed robot used for assisting the persons who suffer from losing muscle control in their upper left limb. The implemented robot is controlled via a PD like fuzzy logic controller designed by Matlab Simulink and interfaced through proposed Simulink design, where it designed by special Arduino Simulink library, the Simulink design connected by two electronically designed interfacing circuits for actuating the robot joint motors. One of these interfacing circuits is used for motor encoders reading with a resolution of 212 bits and the other is used for motors control with [0 -12] volts. Both circuits are interfaced with PC Matlab controller via Arduino (UNO) card. The implemented circuits and the controller show a powerful control and accuracy in controlling the robot links via the desired medical trajectories without human hand load. After attaching the human hand of weight 3.78 Kg, a percentage of occultation appeared according to this load addition. That means that there has been a need for more torque power in the first joint, and for that reason a spring system has been added to the first link for compensating the required torque. The results show a percentage of enhancement after adding the spring system where the first joint error was enhanced in percentage of 29.572% and the second joint error was enhanced in percentage of 22.916%. The next work in this project will involve the design of the nonlinear controller and a special damping system to increase the stability in robot joint through the execution of the required trajectories. After operating the robot as a standalone, we have connected the robot to the internet using IoT protocols and we have developed a web application to control the robotic hand via the internet using the IoRT paradigm as our design model. IoRT is the future of remote-controlled robotic systems, and more focus should be given to the topic now especially in the development of very low latency high speed 5G and in the future 6G technologies.

## References

[1] M. A. Doshi, "Wireless robotic hand using flex sensors," *International Journal of Scientific Engineering Research*, vol. 6, pp. 1471–1474, 2015.

[2] R. De-la Torre *et al.*, "Robot-aided systems for improving the assessment of upper limb spasticity: A systematic review," *Sensors*, vol. 20, no. 18, pp. 1–23, 2020.

[3] L. Zhang, S. Guo, and Q. Sun, "An assist-as-needed controller for passive, assistant, active, and resistive robot-aided rehabilitation training of the upper extremity," *Applied Sciences*, vol. 11, no. 1, pp. 1–24, 2021.

[4] B. D. Argall, "Autonomy in rehabilitation robotics: An intersection," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 441–463, 2018.

[5] A. Mancisidor, "Virtual sensors for advanced controllers in rehabilitation robotics," *Sensors*, vol. 18, p. 785, 2018.

[6] M. Tiboni, "et al, "robotics rehabilitation of the elbow based on surface electromyography signals," advances in mechanical engineering vol. 10," *Advances in Mechanical Engineering*, vol. 10, 2018.