International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 12, No.1 (Oct-2022)

https://dx.doi.org/10.12785/ijcds/120197

Smartly Control, Interface and Tracking for Pick and Place Robot Based on Multi Sensors and Vision Detection

Zeyad A. Karam[1] **and Elaf A. Saeed**[2]

[1,2]*Department of Systems Engineering, College of Information Engineering, Al-Nahrain University, Jadriya, Baghdad, Iraq*

Received 30 Oct. 2021, Revised 15 Sep. 2022, Accepted 27 Oct. 2022, Published 31 Oct. 2022

Abstract: A great role played by the robots during the current pandemic. So far, they contributed a large effect on the medical sector. However, their valuable uses could be developed in many endeavors. Thus, this work involves the assembly of a six degree of freedom (DoF) robotic, DoF's arm, and multi motion control. Also, designed control algorithms with interfacing circuits are proposed for this robot motion where the implemented robot can replace human activity in the industry, by teaching the robot for the required specific process and re-execute it again with the same precision based on the power of the designed controllers. This controlling behavior is known as open-loop control; it's implemented by redesigning a joystick button as a manual controller, then implementing the wireless controller based on a specifically designed mobile application and a Bluetooth model. The re-repeating process can include some workspace obstacles. Therefore, the implemented robot will be equipped with environmental sensors, which can sense the object's presence via sensors or a vision system or it can analyze the object's color by sensors. Where the controlling idea via sensing environment is known as closed-loop control. The power of the designed controllers shows trajectory tracking accuracy of 95% and 97% in the open-loop control idea-based wire control by joystick, and by wireless application respectively, while the accuracy of the trajectory tracking by the closed-loop control system equipped with presence sensors and vision system is 100%.

Keywords: 6 DoF's Robotic Arm, Interfacing Circuits, Color Recognition, and Vision Detection.

1. INTRODUCTION

Robots utilize large roles in our life activities and execute processes which hard to humans, because of robots' fast, exactness, and heavy load. Robots can compensate the human efforts, where they can apply them to the skills that make the robot to acquiring a required job [1]. Robotic jobs can be in various environments and fields to imitate human activities and motions [2]. Therefore, robot design should involve main factors such as the work concept and techniques, building a 6 degree of freedom (DoF) manipulator is not a modern idea, but multi-ideas in the design and the techniques can make difference. The design roles are for perform many actual trajectories close to a human's usual efforts, like, assembling and clasping under the regulations in the roles of the human hand is not workable. Where in 2021, Fomin et al., proposed a novel study involving the analysis of inverse and forward kinematic for a newly designed 6DoF parallel manipulator augmented by a circular guide. Where the proposed structure excludes the clashing of carriages while moving in the circular guide. Where cranks play the role of archiving that in the robotic arm kinematic chains [3]. Karam and Neamah, designed and implemented two (DoF) assistant robots for helping the injured humans with upper limbs activity, where the design includes the mechanical design and analysis using a new novel spring mechanism for compensating the required force in the robot joints based on intelligent designed controllers [4].

Farzam et al., 2020, proposed a 6DOF robot, with linear manipulators, designed and controlled. The proposed controller deals with the inherent nonlinearity of the robot dynamics, where a type of adaptive PID manipulator was proposed to employ and validate the nonlinearities of the model [5]. Another work done in 2020 by Chu et al., proposed teaching devices based on parallel wire-type augmented by force sensors, a kind of hybrid teaching for position-force and a kind of control method were developed to proficiently manipulators teaching based on human hands instead of teaching pendants. Where these methods will be used for teaching the manipulator the required position and force trajectories [6]. Chintan et al., derived 2020 as a general Scara Arm (5DoF) dynamic model. Where the robot model is nonlinear, controlled by new nonlinear controller with fuzzy system to reach the larger error enhancement [7]. Karam, and Awad, analyze the dynamic performance of a car's suspension system for enhancing the comfort and safety of traveling. The work focuses on enhancing the dynamic system specification, where the proposed controller minimizes the car body oscillation after effected to the road obstacles which tried to raising the car fastness [8]. In 2016,

E-mail address: : zyad.karam@nahrainuniv.edu.iq, elaf.ahmed@ced.nahrainuniv.edu.iq

Karam derived the Electrohydraulic servo actuators model and controlled it with hybrid fuzzy controllers to reach the optimal position control specification [9].

Recently, deep-learning technology has been selected as the most presented work among the techniques in the domain of grip synthesis. Where the past works used many kinds of convolutions neural networks (CNNs) to classify the shape of gripe. Therefore, the ideas which depended on the deep-learning technique are hard to apply to the robotic grasping process with multi-target objects, moving angles, and a dynamic environment [10] and [11]. Therefore, deep learning technologies for object detection took a wide scope in robot vision in recent years.

This work analyzes the prototype implementation, and control of a six DoF's Arm, which execute industrial processes, like holding and place of brittle objects or object analysis by robot sensors and vision. This 6 DoF Arm robot will controlled by Arduino of Mega 2560 type. These robots operate using 6 servo motors.

This work aims to implement and control a 6DoF's robot structure. The implemented robot will control by the wire and wireless control strategies. The wire position control algorithm (open-loop) will do by using a redesigned joystick. The wireless position control algorithm (open-loop) will do by using a smartphone-based designed Android application. With each control phase, the motion trajectory is recorded by Arduino card memory with a new recording algorithm. The recorded motion trajectories can be executed for testing the controller's ability.

Also, automatic position control (closed-loop) by multi-sensors will be implemented and the action for that sensing will be executed as trajectory motion to a new specific location.

Also, shapes and object detection will be implemented based on computer vision using a camera and analyzed by a python language algorithm. The recognized shape will be transferred via a determined trajectory to a new location. This transfer tests the controller's ability to locate the object in the new location after recognition.

Also, interfacing circuits will be designed and implemented to interface the wire and wireless motions via Arduino card. The card interface will involve the sensors and robot motors. The forward kinematics will be derived to test the reachable point in the motion area of the robot by using the MATLAB program.

This work is structured as follows. In Section II, control circuits and algorithms design are introduced, In Section III, simulation of the forward-kinematics for the Arm is presented, Section IV, will discuss the results, Section V, shows the error reduction in the playback method, In Section VI, we evaluated the simulation results, Finally, in Section VII, we draw up our Conclusion.

2. CONTROL CIRCUITS AND ALGORITHMS DESIGN

To analyze any robot motion and then kinematics or dynamics, it is required to model the robot structure in schematic form with link length and joint motion limits. And this will be processed and will be performed in two forms i.e 2D and 3D design. Each joint and robot link, with the servo motors, is designed by a solid work program to present the links and joint details. The gripping mechanisms in this robot are also designed in detail, as shown in Fig.1. To control all the Servos motors for the implemented 6 Dof's arm, it is required for interfacing them with a microcontroller or embedded system, in this work Arduino mega card with full embedded digital and analog signals will be used. Therefore, servos (1,2,3,4,5 and 6) are connected to Arduino ports (1,2,3,4,5 and 6) respectively via the PCA9685 Servo Driver. Where the driver enables the user to control the servo motor in steps of motion. Digital sensor is interfaced with the DC regulator circuit to reach the optimal voltage range for the PCA9685 motor Driver, which enables it to drive the motor in a smooth motion. For reaching the desired voltage for the Bluetooth work, we designed a voltage splitting circuit to change the voltages values into 3.3v. Fig.2 illustrates the overall interfacing required between the Arduino and the robot servomotors. While Fig.3 presents the fully interfacing required.

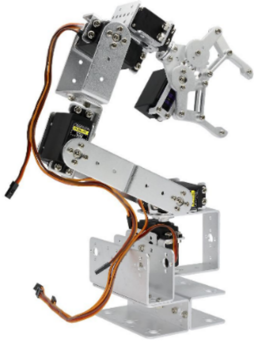


Figure 1. Hardware Model of the Implemented Robot.

The presented circuit in Fig.3 presents all required electronic parts for all implemented ideas in this work. For a better understanding of the fully interfacing TableI illustrate the interfacing pins between Arduino and used motors and sensors. The implementation of circuits in Fig.3 as shown in Fig.4 a set of methods for 6DoF's Manipulator Robot structure Position control was executed based on this

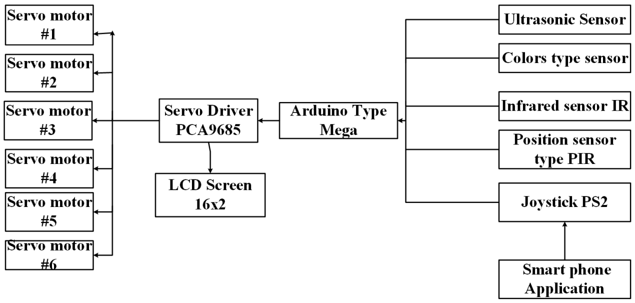


Figure 2. Schematic of the Manipulator Robot control.

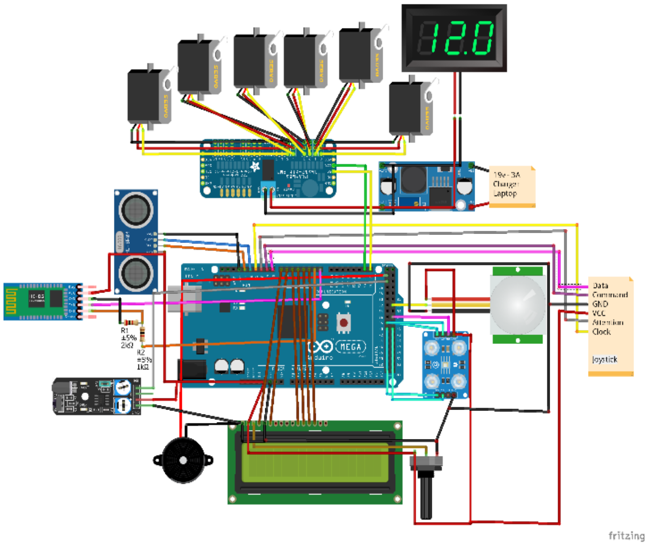


Figure 3. Electrical Stimulation of the Control Circuits.

interfacing as follows:
1-Wire Control for the Position (open-loop method) By using the Joystick .
2-Wireless Control for the Position (open-loop method) By Smartphone Application.
3-Auto Control for the Position (closed-loop method) By using Color ,Ultrasonic, IR, and PIR sensors.
4-Automatic Position control based on object recognition method (closed-loop control system). These methods were programmed by Arduino IDE, MATLAB, and MIT App.

Figure 4. The Designed 6DOF's Manipulator Robot is Electronic Components.

A. *Wire Control using Joystick with Record and Playback Method*

In the Joystick implemented approach, the Joystick will control the motion of the arm, where it used for controls the position angles that transmitted to the servo motors and then presented on LCD. Therefore, in the joystick design all the keys are programmed to control single motor, while the other keys used for other control methods like the record and playback algorithm. By using the record and playback approach, it's recorded a set of steps, these steps are the position angle for each motion, then the approach store them in an array format by using a specific keys in the joystick, after that, the user can re-run these saved positions as trajectory in automated form y the playback approach. Fig.5 shows the schematic for the equipment's that used in Joystick working with record and playback approach. To use a PS2 Joystick as a controller, there is a need to introduce the action of each control key to be known when for the Arduino as shown in Fig.6

TABLE I. Pins Configuration for the Arduino Board and other Electrical Parts.

The electronic element	The components pin Type	Pins to Arduino
Wire control	Clock	pin 11
	Command	pin 9
	Attention	pin 10
	Data	pin 8
LCD 2x16	RS sign	pin 2
	Enable sign	pin 3
	D4 sign	pin 4
	D5 sign	pin 5
	D6 sign	pin 6
	D7 sign	pin 7
	R/W sign	pin GND
Type of Ultrasonic	Tirg sign	pin 13
	Echo sign	pin 12
Type of PIR	Output sign	pin 33
Type of IR	Output sign	pin 24
Alarm	VSS sign	pin 22
Color detector	Switch 0	pin 40
	Switch 1	pin 41
	Switch 2	pin 38
	Switch 3	pin 39
	Color output sign	pin 37
Driver type PCA9685	SDA sign	pin 20
	ACL sign	pin 21
Bluetooth Signals	Transmit data	pin 0
	Receive data	pin 1

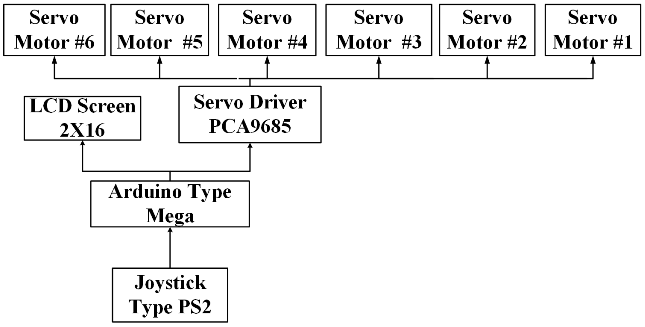


Figure 5. Schematic of working by the Designed Joystick and Record and Playback Approach.

TableII emulates all the motion and action buttons required for moving and controlling the arm, where each one of the signals is interfaced with an Arduino card to read the signals and take the required control action for the robot motors. Fig.7 and Fig.8 shows the control ideologies presented in

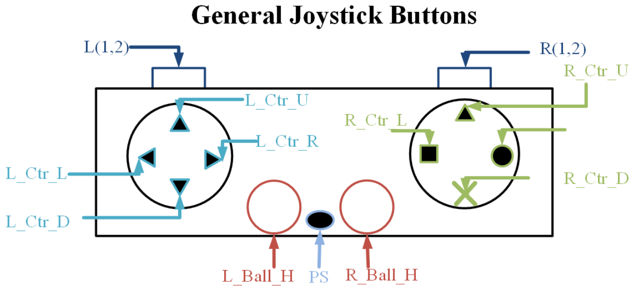


Figure 6. The used Joystick Buttons.

TABLE II. Function of the used Joystick Buttons.

The electronic element	The components pin Type	Pins to Arduino	
L Ball H	Angle of $Servo_1$	Analog	Joystick Method
L Ball V	Angle of $Servo_2$	Analog	
R Ball H	Angle of $Servo_3$	Analog	
R Ball V	Angle of $Servo_4$	Analog	
L Ctr U	Angle of $Servo_5$	Analog	
L Ctr D	Angle of $Servo_5$	Analog	
L Ctr L	Angle of $Servo_6$	Analog	
L Ctr R	Angle of $Servo_6$	Analog	
L_2	$Setup_servo_5$ to 0 angle (direct)	Analog	
R_2	$Setup_servo_5$ to 180 angle (direct)	Analog	
L_1	Angle of $Servo_6$	Analog	
R_1	Angle of $Servo_6$	Analog	
R Ctr L	Automatic run 6	Analog	Record and Playback method
R Ctr R	Save current servo position	Analog	
R Ctr U	Run saved trajectory(steps)	Analog	
R Ctr D	RESET the saved data to 0	Analog	
L Ball Btn	return to the initial servo point	Digital	Automatic method
R Ball Btn	find the saved path	Digital	
PS	using the PIR and IR Sensors	Digital	

flowcharts as a path for idea of direct position control based Joystick and direct Position Control by Record and rerun method.The overall moving ideologies presented in the above figures are about moving the implemented arm in the required joint angles position, where the movement done by open-loop control signals referenced to the servo motor signal comes from the joystick. Through this control all movement steps (motors joints angles) are recorded in array form in the controller memory, then the user can remove the recorded path from a specific joystick button, where the recorded movement execution is done in high performance with minimum position error.

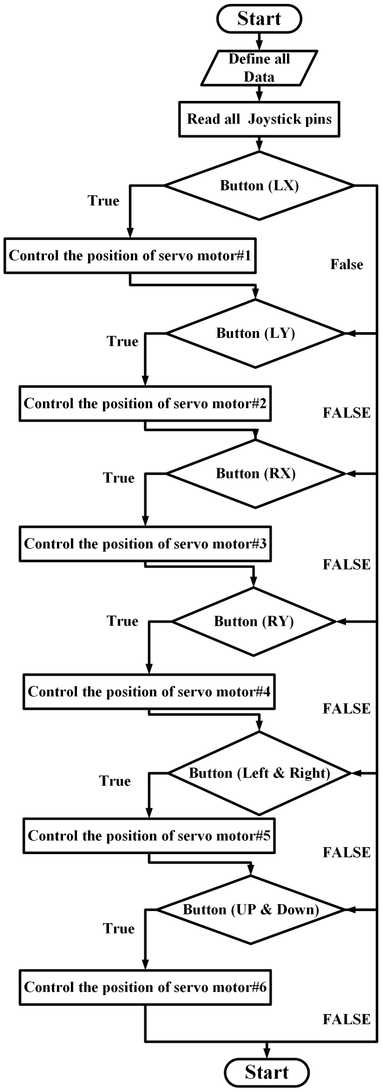


Figure 7. Wire Control using Joystick.

B. Wireless Control based Application with Record and Playback Method

Using the sliders in the phone application enables the user to reach the manual controlling of motion for both servo and axis of the arm. The use of such a method, to improve the idea of IORT (internet of robotics thing), where the robot wireless motion control via a mobile application will be implemented. Where this experiment exposes the ability in controlling the robot trajectory via a Bluetooth module in the most accurate way. Where each slider in the phone application enables the user to control the motor angle from 0° till 90° ,however the using the "Save" button can record the arm motion as steps, then the robotic arm can auto-move and re-perform these saved steps (recorded positions). then, using the same push one can stop the automatic process with the ability of reset or delete all recorded positions, then the user can record a new one. Fig.9 showed the Block Diagram of components that work in this approach. This application is programmed using the Massachusetts Institute of Technology Application Inventor and Java Studio for Android which supported by Google. At top side there are two buttons in joystick used for authenticating the controlling phone with the Bluetooth device of type HC-05, as shown in Fig.10, Fig.11 and Fig.12 shows the schematic of the wireless position control by mobile application-based android method and the idea of the wireless position control by record and playback method referenced to the saved executed trajectory by the user application. Where these figures present the control methodology for controlling the robotic arm wirelessly based on IORT technology, where the motion is recorded based on moving each joint angle in series-parallel motions. This recording is done by the controller memory also in array form. The execution of the recorded trajectories presents a minimized error in motion.

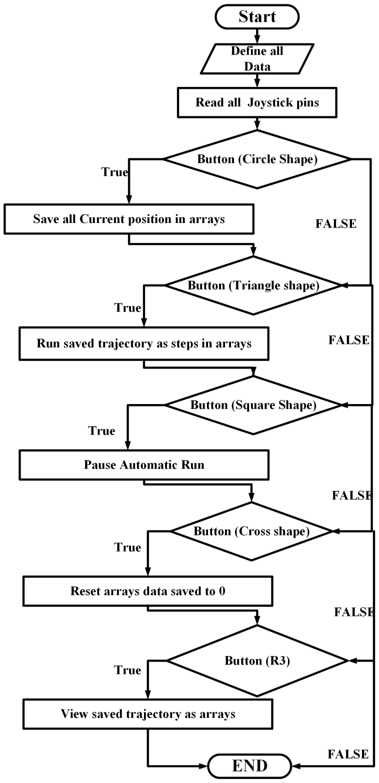


Figure 8. Wire Control by Record and Playback Method.

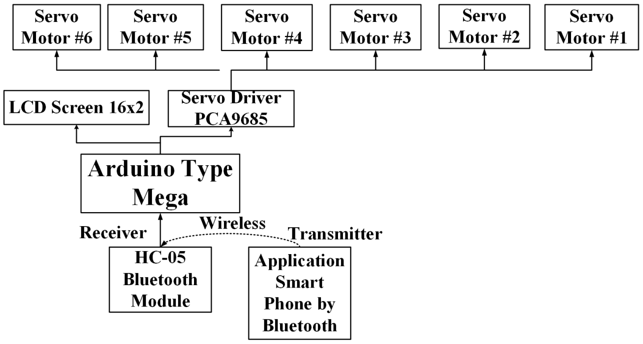
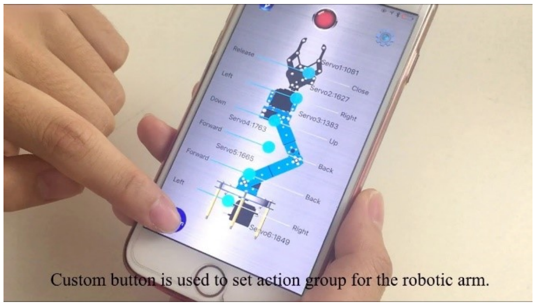


Figure 9. Application methods-based record and playback trajectory.



Custom button is used to set action group for the robotic arm.

Figure 10. Android Application for Arms Control.

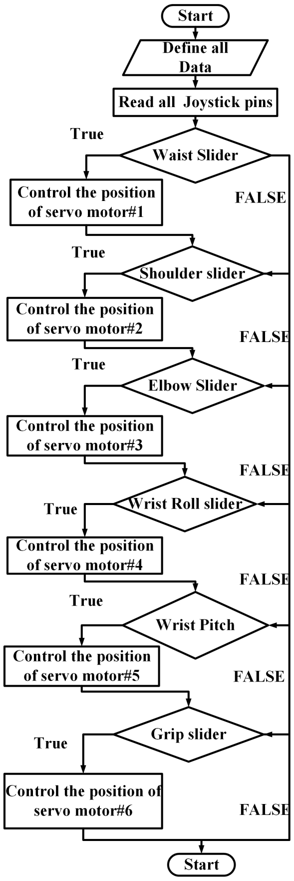


Figure 11. Wireless Control by Smartphone Application.

C. Auto-control for Position by Multi Sensors (closed-loop system)

In this part, a Set of Sensors of type PIR, Colour, Ultrasonic and IR sensor interfaced to the Arduino, Fig.13 shows the Block Diagram for the components of this idea. The color will be working with ultrasonic sensor for supporting the production line. suppose in the production line a conveyor belt conveys a group of products in different colors and it's required to sort them based on color. So, the robot will pick the product up after the controller senses the product position via the ultrasonic sensor, then the robot will move the product to the color sensor position to put it in a perpendicular position for recognition after the color recognition process is done, the robot will decide to move the product to the container position of that color. Fig.14 presents this sorting process.

The following is the procedure of moving when ultrasonic and color sensors are used:

1- When the key of symbol of start is used in the Joystick. The Arduino program will calls the restart algorithm and fire the Ultrasonic sensor.

2- The ultrasonic sensor will transport specific signals within one meter range started from the arm position, and when these signals detect a obstacle, it re-back to the

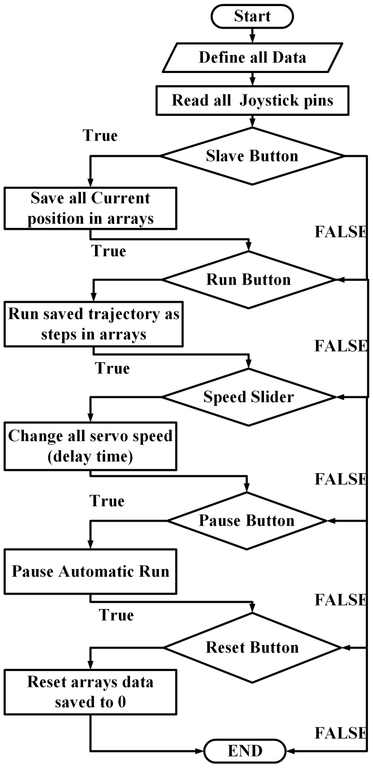


Figure 12. Wireless Control by Record and Playback method.

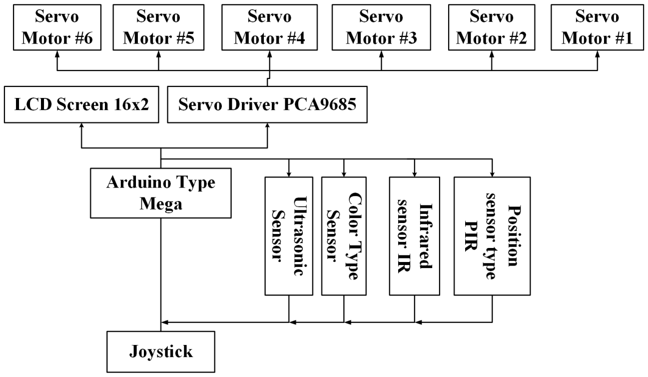


Figure 13. Schematic of Working with Sensors Method.

ultrasonic .

3- The detected distance of the obstacle will calculate by the ultrasonic and then presented by the LCD. then, the Arduino will calls the build (servo-onc-time) program to firing the recalculated path that moves the arm to find the detected obstacle and hold it up by an arm-gripper.

4- After picking the object, the arm moves by depending on the recorded path toward the color detector to optimize the color, then the Arduino program will use the (color) function to optimize the color of such object.

5- Now, the color is recognized. The color type will be displayed on the LCD, then Arduino will repeat running of the function of (servo one time) to make the arm move toward the color container (red, blue, green) as response to the same product color that sensed and analyzed by sensor and Arduino.

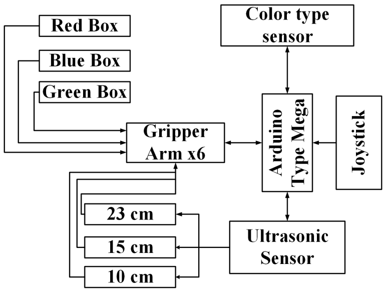


Figure 14. Schematic for Color and Ultrasonic Sensors Method.

6-Now, the robotic arm return back to the starting point and the control mode by joystick is fired. The users can repeat this process by pressing the start key again.

For firing the mode of IR and PIR sensors, the user can presses the (select) key, then the IR sensor is fired and start searching for any object nearest to the circuits of the arm. If there is an detection, the controller program will move the arm manipulator depending on a calculated path to protect the interfacing circuit from any object and alarming a massage on the LCD. This action is programmed for protect the arm interfacing circuits from damage that can happen while the production process.

To detect the close loop controller ability, the arm are tested by hello trajectory form. The user can presses the select key again, the PIR sensor is fired on and start searching for any object that approximate to the arm with in two meters distance, if any object detected like human arm that makes hello. Then the arm will execute a pre-saved trajectory for hello motion, this is a review for the arm's controllers abilities only. Fig.15 shows Flowchart of Automatic Position control method by sensors.

All the executed trajectories with sensors show highly tracking performance with minimum positioning error, where each process was tested more than one hundred times, with each process 97% of them performing the trajectory exactly. That means the determination of motion and recording process based on motor steps are executed optimally.

D. Auto-control based on Object Recognition

In this work, a fast, and lightweight technique, based google teachable machine is a web-based tool that makes it fast and accessible. it's enabled the user to train on the picture without writing any code. Using the supported camera one the robot gripper to take a picture and then training it by this technique, then the shape is detected, then sent to the robot controller as specific characters represent the object name, which enables the robot to execute a predetermined trajectory for picking and placing the detected object in the specified position.

This idea can be used in factories to carry items from the conveyor belt (10,11). So, in this work, the object is detected

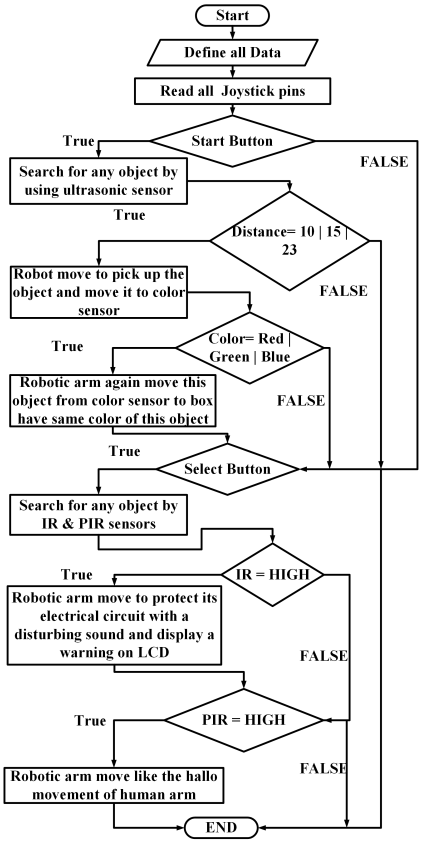


Figure 15. Auto- control by Set of Sensors.

using the camera that was placed on the robot and interfaced to the computer which involves the Record and remove method to calculate all the predetermined paths. Training ideas for any taken image will pass the following process:
1- In this research, the type of teaching was used tech based on images, as shown in Fig.16.
2- Training the model through a google teachable machine website by using the sample image of size 224x224 pixels. As shown in Fig.17.
3- Taking a set of images of the required objects using the webcam supported in the robot gripper and giving each object a specific name.
4- Pass the images to the training step.
5- Convert the model to a Keras [12] model. Keras is the library that is used in programming.
6- Download it as a file to use it in the project and recognize the object. This file was downloaded in the code to make compression between the samples that were used in this file and video frames.
Fig.18 shows the flowchart for the training method steps.
The process of recognizing the objects started by instilling the webcam in the robot gripper. When the object recognizes the robot, but the object is in a specific place and the robot returns to the start point to recognize another object. The process block diagram is presented in Fig.19.
These steps of the recognition process can be emulated in

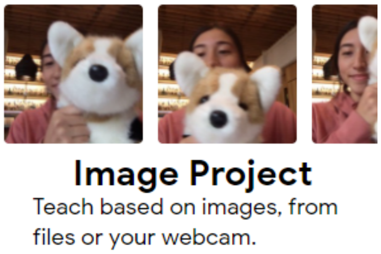
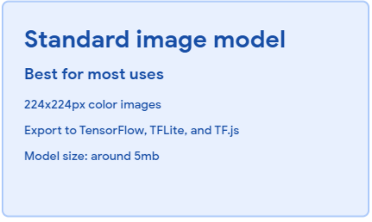


Image Project

Teach based on images, from files or your webcam.

Figure 16. Teaching based on Image.



Standard image model

Best for most uses

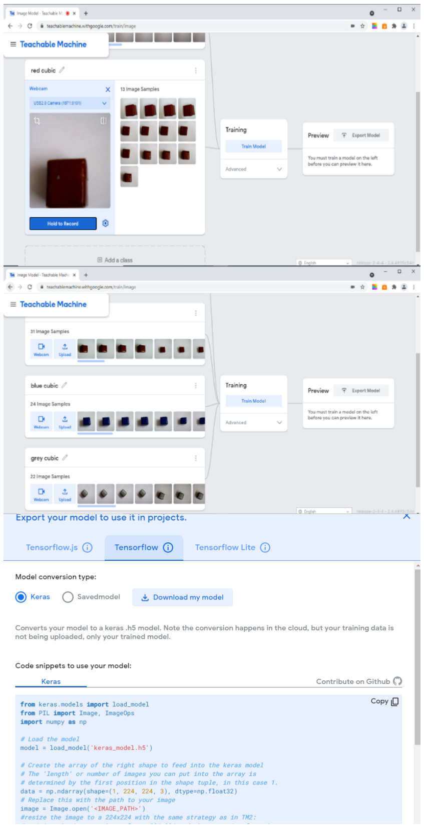
224x224px color images

Export to TensorFlow, TFLite, and TF.js

Model size: around 5mb

Figure 17. The model that is used in Teachable Machine.

the following points:
1- The robot is installed at the start points. Then the instilled webcam robot checks whether (i) there is an object or not.
2- If there is no object, then no object is written on the display area and the letter (n) is sent by the computer to the robot analyzer. The robot remains firmly in its place.
3- If there is an object, the object is checked by the training process whether is red cubic or blue cubic or gray cubic, or pen. Or you could write it like that: the object is checked by the training process based on the color state i.e. red cubic.
4- In case of the object is a red cubic, then the program print (will display the state on the screen) the letter (r). Then, the robot will execute a saved trajectory for transferring the cubic toward the red containers.
5- After the object is delivered, the robot will return to the starting point (s).
6- In the other case of blue cubic, the program print on the display screen the letter (b). Then the robot will execute a saved trajectory for transferring the cube toward blue containers.
7- Then, the robot will return to the point (s).
8- The robot will execute in the same way for a gray cubic to move it toward point gray container and a pen to point en container.
9- The letters that are sent to the robot which print on the LCD.
Fig.20 shows the flowchart of the object recognition method. Where the video frames resize to 224x224 pixels because the model was trained in a teachable machine on 224x224 pixels. When taking photo samples, it took them in size 224 x 224 pixels and does the training on them, but after predicting the object it resizes the frames of video to 500x500 pixels size to display the video.
If the color or shape is required, the robot will be moving from the starting point (s) to the desired place based on



You must train a model on the left before you can preview it here.

Export your model to use it in projects.

Model conversion type: ● Keras ○ Savedmodel

Converts your model to a keras .h5 model. Note the conversion happens in the cloud, but your training data is not being uploaded, only your trained model.

Code snippets to use your model:

```
from keras.models import load_model
from PIL import Image, ImageOps
import numpy as np

# Load the model
model = load_model('keras_model.h5')

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1.
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
# Replace this with the path to your image
image = Image.open('<IMAGE_PATH>')
#resize the image to a 224x224 with the same strategy as in TM2:
```

Figure 18. Steps 3,4,5 and 6 Training Method Steps.

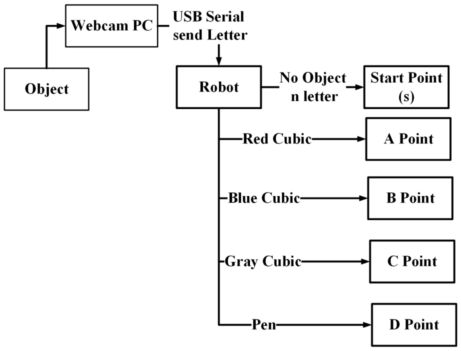


Figure 19. Schematic of the Steps for Object Recognition.

the recognition process result, where the predetermined trajectories for the target motion were trained via the joystick by moving the robot from the starting point (s) to the required points A, B, C, D, and saving them in array forms in the controller memory. So, when recognition is done, the robot controller will execute the stored trajectory in a playback manner.

3. FORWARD KINEMATICS SIMULATION FOR THE ARM

For kinematics calculation, there are two general issues to assessed in the arm end-effector position with each joint,

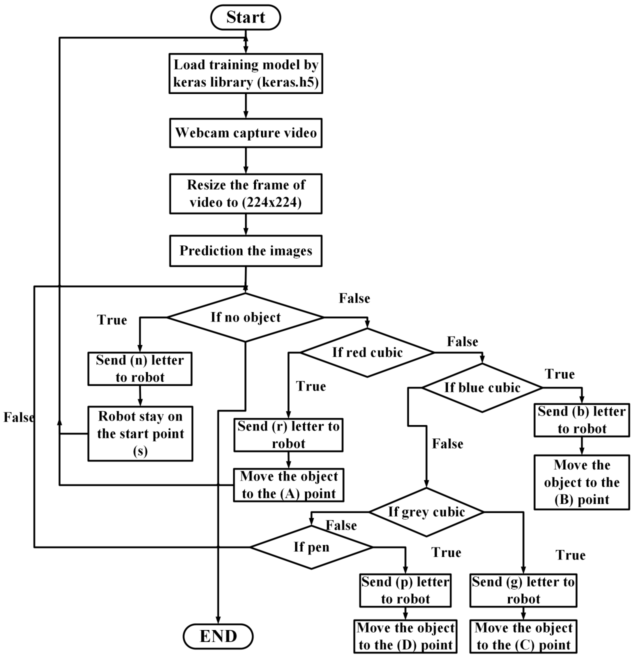


Figure 20. Flowchart of the Steps for Object Recognition.

it's the joint-angles and the link-arguments, the position and direction can calculated for the end robot point using the reference idea for frames. This idea is named as forwarding kinematics. Also, if the present point and direction of the end robot point were available, then the angles in each joint can be measured. This approach is named as inverse kinematics. In this work, just the forward kinematics were analyzed for ensuring that the applied physical trajectories were executed precisely. Fig.21 presents the forward kinematic technique.

The used manipulator has a six-axis. Three of them which simulate the foundation, joint of the shoulder, and elbow joint are responsible to transfer the robot toward the target point. The design depends six rotational parameters of joint [13] and [14]. Fig.22 present the exact coordinates diagram for the link frames for each axes and joints.

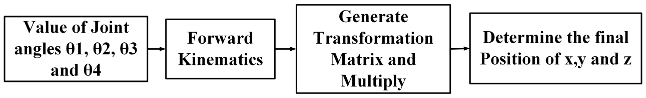


Figure 21. Forward kinematic block diagram.

The DH (Denavit–Hartenberg) theorem will use for this analysis, which include four arguments: Angle for each joint (Θ_i), link-offset (d_i), link-distance (a_i), and the twisted-angle for the links (α_i). These four arguments are used to compute the end-effector point and direction. Then, the forward-kinematics, the input is the vectors of the joint-angles (Θ_i) and the link-length argument (a_i). TableIII illustrates the all investigated parameter based on the employed referenced frame presented in Fig.22. Then the Output is

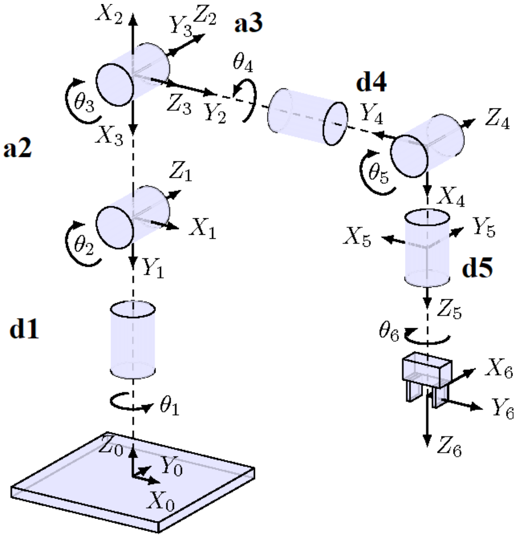


Figure 22. Schematic for the 6 DoF's Robot based on Frame Analysis.

the direction and the localization of the robot end point [15] and [16].

The transformation matrix T_0^6 that named the Homogeneousar matrix, will calculated based on table III with 6 joints, while the symbols P_x, P_y and P_z represent a general axes for actual position of the robotic manipulator end-point, which calculated from the forward kinematic of matrix's multiplication. The MATLAB program used to multiply the matrices A_{1-6} toward finding the end effector position, then the resulted position equation will be:

TABLE III. D-H Table for the used 6DOF's Manipulator Robot

Joint No.	offset d_i	Length of links a_i	Twisted orientation Angle α_i	Rotational joint Angle Θ_i
1	d_1	0	90	Θ_1*
2	0	a_2	0	Θ_2*
3	0	a_3	0	Θ_3*
4	0	a_4	90	Θ_4*
5	d_5	0	90	Θ_5*
6	0	0	90	Θ_6*

$$T_0^6 = A_1A_2A_3A_4A_5A_6 \tag{1}$$

Then the resulted end effector position by the forward kinematic transformation matrix T_0^6 will be:

$$P_x = \cos\theta_1(a_2\cos\theta_2 + a_3\cos(\theta_2+\theta_3) + a_4\cos(\theta_2+\theta_3+\theta_4) + d_5\sin(\theta_2+\theta_3+\theta_4)) \tag{2}$$

$$P_y = \sin\theta_1(a_2\cos\theta_2 + a_3\cos(\theta_2+\theta_3) + a_4\cos(\theta_2+\theta_3+\theta_4) + d_5\sin(\theta_2+\theta_3+\theta_4)) \tag{3}$$

$$P_z = d_1 + a_2\sin\theta_2 + a_3\sin(\theta_2+\theta_3) - d_5\cos(\theta_2+\theta_3+\theta_4) + a_4\sin(\theta_2+\theta_3+\theta_4)) \tag{4}$$

To emulate and simulate the motion of the robot joint and the end effector, it is required to move the joints and the gripper referenced to a specific motion trajectory. The derived position equations will be used in this simulation. The simulation enables the user to ensure the robot motion space dimension and the required torques by the motors for the specified motions in the physical tests. This positioning test is done by a proposed third-order polynomial trajectory planning method. However, if any joints at the motion starting path has initial time t_i and initial position θ_i.Then needed to move toward new final position of points θ_f during final time t_f [17], [18] and [19], so the trajectory will be:

$$\theta(\mathrm{t}) = \mathrm{c}_0 + \mathrm{c}_1\mathrm{t} + \mathrm{c}_2\mathrm{t}^2 + \mathrm{c}_3\mathrm{t}^3\mathrm{At} = 0\left(\mathrm{t} = \mathrm{t_j}zero\right) \tag{5}$$

The following four equations of arguments allow the user to find the four unknowns parts as follow:

$$\mathrm{c}_0 = \theta_\mathrm{i} \tag{6}$$

$$\mathrm{c}_1 = 0 \tag{7}$$

$$\mathrm{c}_2 = \frac{3(\theta_\mathrm{f} - \theta_\mathrm{i})}{\mathrm{t}^2\mathrm{f}} \tag{8}$$

$$\mathrm{c}_3 = \frac{-2(\theta_\mathrm{f} - \theta_\mathrm{i})}{\mathrm{t}^3\mathrm{f}} \tag{9}$$

Fig.23 presented below the simulation of the forward kinematic based equations (2 to 4). The forward kinematic simulation results is presented in Fig.24, the input is θ_i, θ_f for the equations (6 to 9) and the output is the joint angle trajectory for each joint. By this method, each joint angle tracks a specific term (initial and final angles, initial and final operation time), and the result is an ordered joint angle

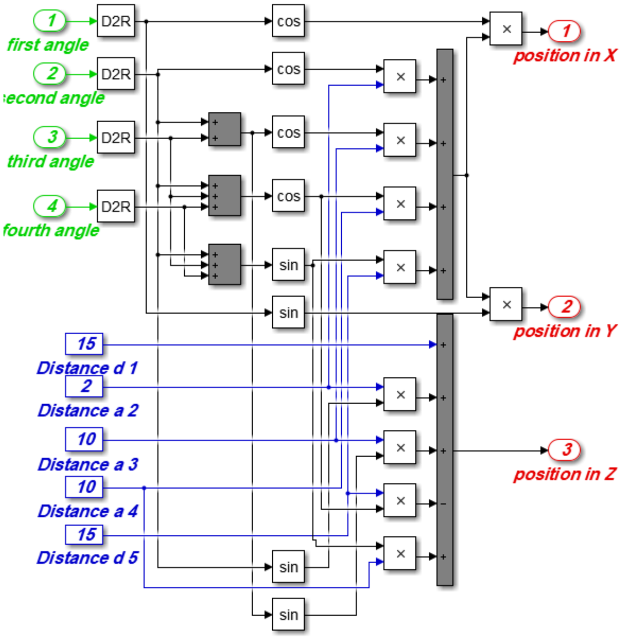


Figure 23. Forward Kinematic Simulation by MATLAB Program.

that activates the forward kinematic equations (2, 3, and 4), so the robot end-effector will move in ordered motion from a specified position to another.

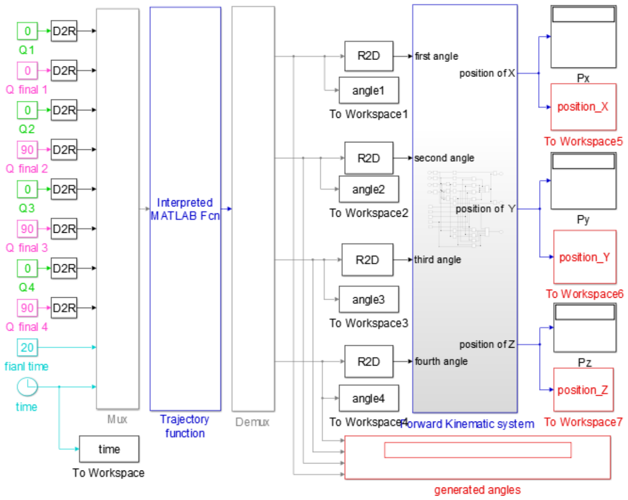


Figure 24. Manipulator Robot trajectory Simulation.

4. RESULTS AND DISCUSSION

The following paragraphs will discuss the physical and simulation results presented in this work

A. *Position Control Directly Using Joystick for Record and Re-motion Path*

Fig.25 shows the result motion by open-loop control method-based joystick control. These results show a high performance in positioning operation, where the overall carry load is sample one because it's required to test this prototype robot. This is related to the fine torque of the used servo motors. The motion accuracy was measured after running the playback method for repositioning the robot head in the same location that was recorded by the joystick control phase within more than a hundred tests. Where it found 95%. This percentage gives a conclusion about the optimally of the used recording execution process.

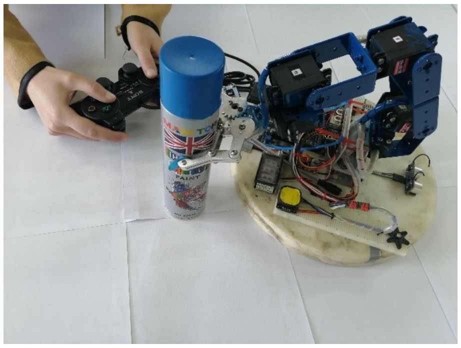


Figure 25. Results of Movement by Wire Controller.

B. *Indirect Position Control Using Phone Application with Record and Re-motion Idea*

Fig.26 shows the result motion by open-loop control method-based Smartphone Application control. The tested method presented in the above figures satisfies the principle of IORT, where the required motion for each robot joint is sent by the Bluetooth module from the smartphone application and received by the HC-05 module, which sends the received data to the Arduino controller. the tests involve more than one hundred, where this motion is recorded in the moving phase in the controller's memory and re-run in the playback phase. This operation showed an accuracy in motion percent of 97%.

Figure 26. Results of Movement by Wireless Control-based Application Smartphone.

C. *Automatic Position control by Set of Sensors*

Fig.27 shows this result for the color and ultrasonic sensor test. More than one hundred tests were done in this closed-loop control method. The proposed control algorithm for the execution of the predetermined motion trajectory-based color shows a highly accurate percentage of 99%. Where the tests were done in three colors (red, green, and blue). The robot accurately positions the analyzed color in the target container. The robotic arm motion that resulted by the IR sensor is fired to detect any object close to the electrical circuit. Fig.28 shows this result of motion. Fig.29 shows the result of motion if any action is detected by the PIR sensor. Fig.28 and Fig.29 present the results of PIR and IR sensor sense and robot action, using such sensors to detect the robot environment and make the robot take any action against it. Where this behavior can reach by using the closed-loop technique (sensing, analyzing, and action). This testing enables the user to test the robot in various positions and orientations, which gives a wide induction about such kinds of robot kinematics in its environment.

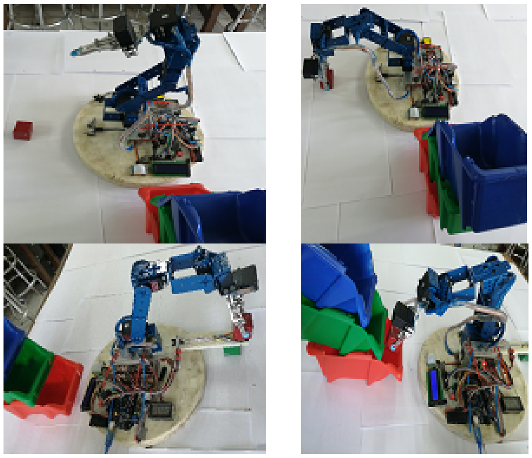


Figure 27. Results of the Movement-based Color and Ultrasonic Sensors (using red cubic).

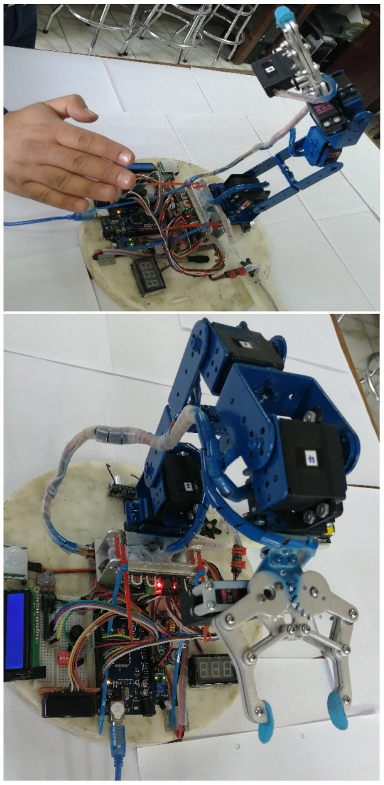


Figure 28. Results for Protection using IR Sensor to Detect Any Nearest Action.

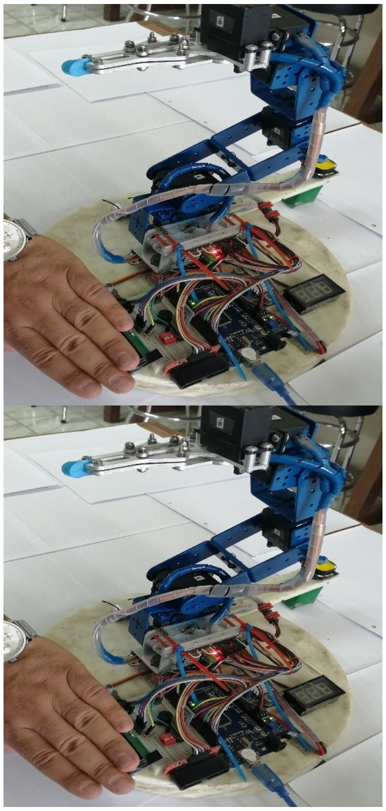


Figure 29. Results for Protection using IR Sensor to Detect Any Nearest Action.

D. *Automatic Position control based on object recognition method*

Fig.29 shows the result of recognition will present in the following steps: For teachable machines Fig.30 shows the result of the training of the taken image by the webcam. As shown in the Fig.30 the training machine trained shapes for (pen and gray cubic), where the result of recognition for selecting the detected shape shows a 100% of accuracy. Fig.31 showed the result of object recognition by using a webcam with python coding for recognition.

5. THE ERROR REDUCTION IN THE PLAYBACK METHODS

Before reducing the error percentage in the method (playback), as shown in Fig.32. The measuring of error is done by monitoring the re-runed trajectories and it's

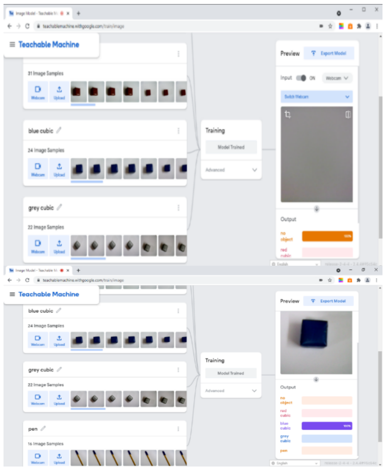


Figure 30. Results for Protection using IR Sensor to Detect Any Nearest Action.

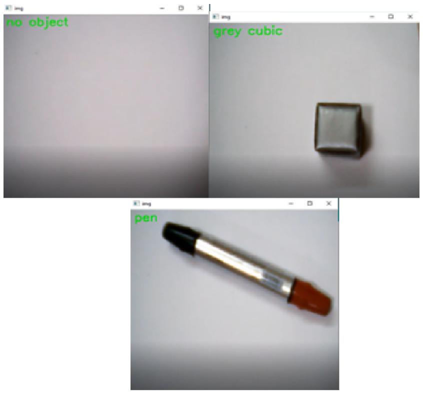


Figure 31. The result of object recognition by using a webcam with python coding.

found in the range of 50%. So, it's required to minimize this error percentage. The error rate has been reduced by placing a statistical process that takes more than one reading within the code, and whoever takes the average readings as shown in Fig.33, the error rate has been reduced from 50% to 5%. Also, the error percentage in the method of the color recognition with the color sensor, shown at 60%, this percent calculated from re-running the process many times, and shown in Fig.34. After reducing the error in the same way above the rate has been reduced from 60% to 3%. Fig.35. present this reducing algorithm. In comparison with the previously implemented work [20], by using the same robot kinematic and vision system. Table IV illustrates the degree of coolers detection between the proposed work with the previous one. These results were obtained after more than 30 tests. It's clear from the Table IV that the used recognition algorithm-based python language provides more accurate detection results. The accuracy of the proposed system is more accurate was its working space with three axes, while the proposed one works in two axes workspace. Which gives power to the proposed work.

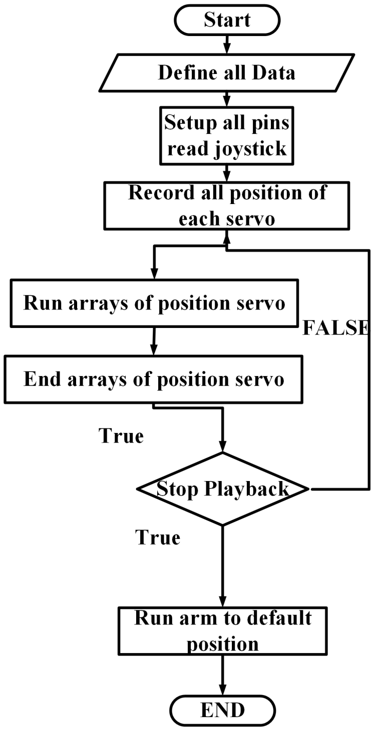


Figure 32. Algorithm of Reducing Error in the Playback Method Before Reducing.

TABLE IV. Comparison between the proposed work and the previous one based on the degree of color recognition.

Color	Previous/ + Rate	Previous/ - Rate	Proposed/ + Rate	Proposed/ - Rate
Blue	95.83%	4.17%	98%	0%
Green	100%	0%	100%	0%
Red	100%	0%	100%	0%

Color	Previous/ Precision	Proposed/ Precision
Blue	90%	96%
Green	100%	100%
Red	100%	100%

6. SIMULATION RESULTS

Kinematics simulation results in forward phase for the manipulator arm are illustrated in Table V, where the test is done by testing a group of required angles and finding the robot arm end-point position. Where these results can satisfy the required tests for the implemented robot in touching all reachable points in its real physical environment. The second test is done by applying the third-order polynomial trajectory instead of a specific joint angle in each robot joint. Table VI presents the required parameters for each

TABLE V. Tests of Path motion for the six DOF's Manipulator Arm.

Trajectory No.	Desired input				Forward Output		
	Θ_{f1}	Θ_{f2}	Θ_{f3}	Θ_{f4}	P_x	P_y	P_z
1	0	60	0	0	13	0	33
2	0	0	0	90	27	0	25
3	60	60	60	60	-11	-21	9
4	30	30	30	30	4.3	3.33	46.22
5	0	60	90	60	-21	6	11
6	60	0	30	90	-14	-10	43

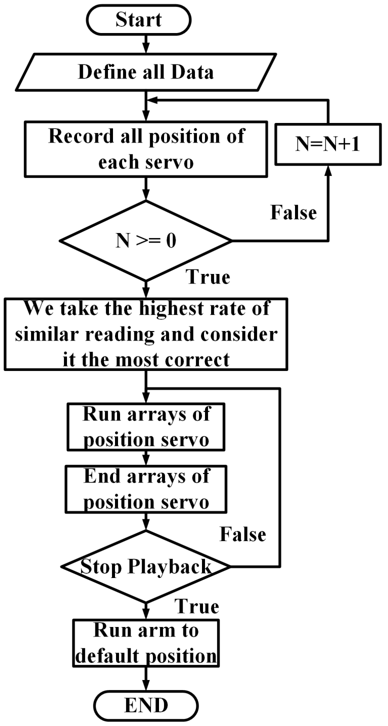


Figure 33. Algorithm of Reducing Error in the Playback Method After Reducing.

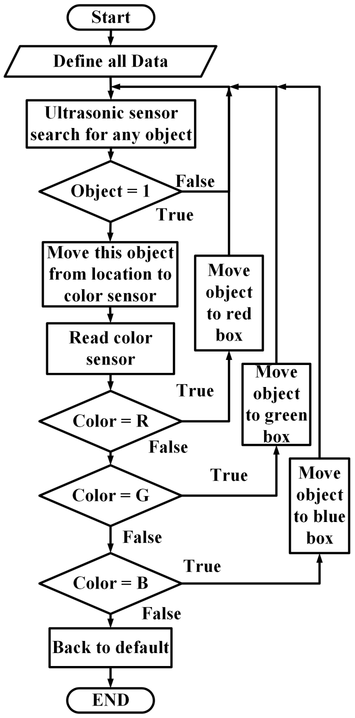


Figure 34. The Error Calculation in the Colour Sensing Method, Before Reducing.

joint angle trajectory and Fig.36, Fig.37 and Fig.38 presents the trajectory results for the end effector after calculation of the theta inputs, $\theta_1 i, f$, $\theta_2 i, f$, $\theta_3 i, f$, and $\theta_{4i,f}$.

TABLE VI. Path Tests by joint angle Path motion for the six DOF's Manipulator Arm.

Trajectory No.	Desired input				Forward Output		
	Θ_{f1}	Θ_{f2}	Θ_{f3}	Θ_{f4}	P_x	P_y	P_z
1	85	0	0	0	0	23	0
2	0	80	0	0	15	0	35
3	0	0	75	0	17.22	0	33.56
4	0	0	0	90	27	0	23
5	45	35	45	30	20.76	9.81	32.56
6	90	45	35	45	4	37	43.95

A comparison with previous work [21], that depending on the simulated angle response of robot servo motors is made and illustrated in Table VII. From the TableVII, one can conclude that the proposed control idea gives a better response in each joint link; where in the first joint there is an enhancement in tr by 4.82% and the ts by 12.5%, and in the second joint there is an enhancement in tr by 4.61% and the ts by 10% in the third joint there is an enhancement in tr by 7.14%, in ts by 10% and in Mp by 29.54%, in the fourth joint there is an enhancement in tr by 5.71% and the ts by 0%, in the fifth joint there is an enhancement in tr by 11.25% and the ts by 13.63%, and in the sixth joint there is an enhancement in tr by 8% and the ts by 6.66%.

TABLE VII. Performance analysis between the proposed and previously used techniques by joint angles responses.

Trajectory No.	Previous work			Proposed work		
	t_r	t_s	$Mp\%$	t_r	t_s	$Mp\%$
1	1.45	4	0	1.38	3.5	0
2	0.65	2	0	0.62	1.8	0
3	0.7	2	0	0.65	1.8	9.3
4	0.7	2	13.2	0.66	2	0
5	0.8	22	0	0.71	19	0
6	2.5	3	0	2.3	2.8	0

7. CONCLUSIONS AND FUTURE WORK

The artificial 6DOF's Manipulator Robot was implemented. Arduino-based interfacing circuits are designed for controlling the robot via specific control algorithms written for each specific job.

At first, the robot motion was controlled using the wire control idea by a Joystick, and then wirelessly by an Android application and Bluetooth module as open-loop controllers. For each controller type, the picking and place operation is done while the motion is recorded via the controller memory, then it's re-executed by the joystick play button or wirelessly by the application scroll. The accuracy of the playback method was adopted to be around 95% and 97% for wire and wireless motions respectively.

Secondly, the closed-loop control ideology was involved for control the robot's motion based on a real-time environment interaction. Where the robot environment is tested by PIR,color ,Ultrasonic and IR sensors, where these sensors exhibit the ability of the robot to reach the desired locations based on saved predetermined trajectories using

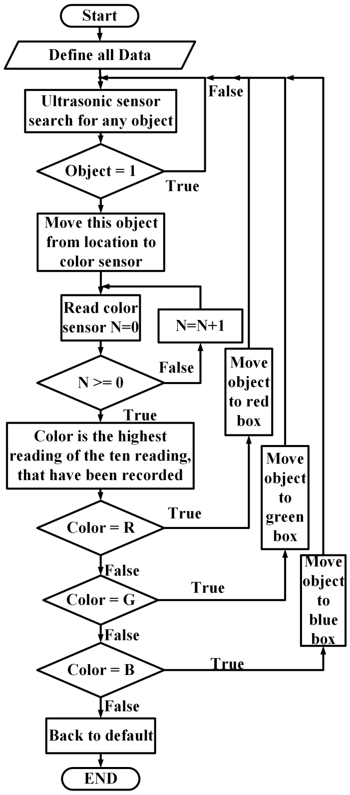


Figure 35. The Error Calculation in the Colour Sensing Method, After Reducing.

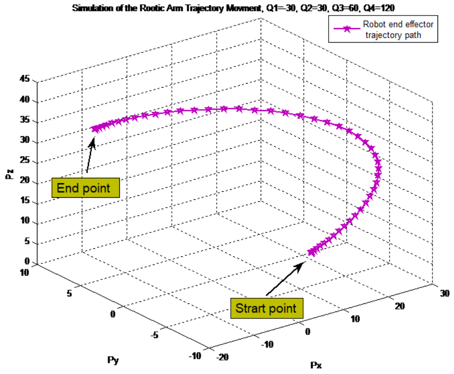


Figure 36. Path Tests by forwarding-Kinematics for the Arm/first trajectory.

the playback method with an accuracy of 100%. Where for each sensing method the required action is executed as a pre-saved motion.

Thirdly, using vision technology to control the robot's motion based on object recognition is a powerful technique via webcam camera, which is concerned with putting each detected object in a predetermined location. Where the used technique detected the objects based on the pre-optimized shapes. After each detection, the robot will transfer the shape to the new predetermined location as a container.

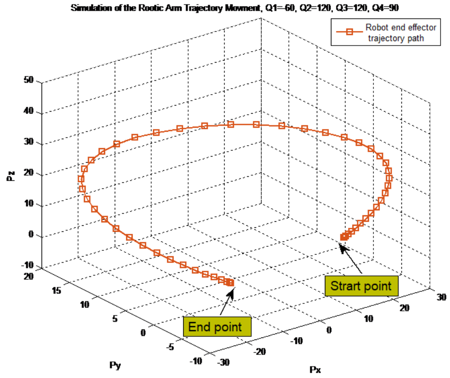


Figure 37. Path Tests by forwarding-Kinematics for the Arm/second trajectory.

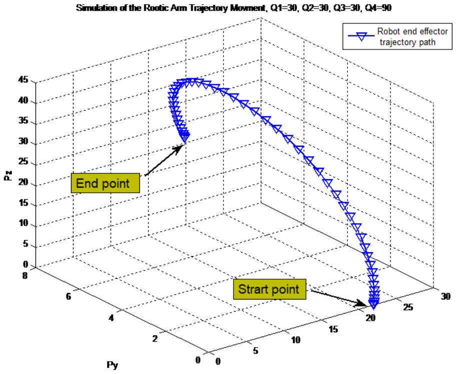


Figure 38. Path Tests by forwarding-Kinematics for the Arm/third trajectory.

Fourthly, a simulation of forwarding kinematics was investigated with a third-order polynomial motion trajectory to test all the reachable points in the robot workspace. The simulation gives the user the required image of the implemented robot's physical workspace.

One can conclude that the constructed six DOF's Arm is beneficial in perform multi-task operations from long distance points. Also, the implemented arm can record the entire movements using the controller memory. This function enables the user to perform tasks first, and then the Arm will be able to perform the recorded task by its own, this point considered the power of this robot.

This work contribution is; design open and closed-loop controllers in matrices form using a new idea for recording the robot motion-based Arduino card. The motion was tested in response to multi-sensors detection, and the robot acted for this detection by tracking the motion trajectories referenced to the saved locations by 100% present. Also, the recognition by vision process is done exactly by the python algorithm and the robot picks the recognized object and placed it in the required location exactly.

References

[1] H.-I. Lin, Y.-C. Liu, and C.-L. Chen, "Evaluation of human-robot arm movement imitation," in *2011 8th Asian Control Conference (ASCC)*. IEEE, 2011, pp. 287–292.

[2] S. K. Saha, *Introduction to robotics*. Tata McGraw-Hill Education, 2014.

[3] A. Fomin, A. Antonov, V. Glazunov, and Y. Rodionov, "Inverse and forward kinematic analysis of a 6-dof parallel manipulator utilizing a circular guide," *Robotics*, vol. 10, no. 1, p. 31, 2021.

[4] Z. A. Karam and M. A. Neamah, "Design, implementation, interfacing and control of internet of robot things for assisting robot." *International Journal of Computing and Digital Systems*, vol. 11, no. 1, pp. 387–400, 2022.

[5] F. Tajdari, N. E. Toulkani, and N. Zhilakzadeh, "Semi-real evaluation, and adaptive control of a 6dof surgical robot," in *2020 11th Power Electronics, Drive Systems, and Technologies Conference (PEDSTC)*. IEEE, 2020, pp. 1–6.

[6] Q.-T. Chu, H. Tanaka, H. Inuzuka, Y. Morita, and M. Sakai, "Study on hybrid position/force teaching and control method for 6 dof manipulator utilizing f-pawted." *J. Robotics Netw. Artif. Life*, vol. 7, no. 4, pp. 222–226, 2021.

[7] S. S. Ghintab, Z. A. Karam, and S. Hasan, "Highly-error enhanced smartly-algorithmic structured impedance fuzzy controllers for a scara redundant manipulator," *Solid State Technology*, vol. 63, no. 6, pp. 1727–1746, 2020.

[8] Z. A. Karam and O. A. Awad, "Design of active fractional pid controller based on whale's optimization algorithm for stabilizing a quarter vehicle suspension system," *Periodica Polytechnica Electrical Engineering and Computer Science*, vol. 64, no. 3, pp. 247–263, 2020.

[9] Z. A. Karam, "Pi-like fuzzy logic position controller design for electro-hydraulic servo-actuator based on particle swarm optimization and artificial bee colony algorithms," *Al-Nahrain Journal for Engineering Sciences*, vol. 19, no. 2, pp. 395–406, 2016.

[10] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4461–4468.

[11] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[12] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[13] Y. Nones, "Heterogeneous modeling & design of a robot arm control system," *Illinois Tech Robotics (ITR)*, pp. 1–6, 2003.

[14] D. Tolani, A. Goswami, and N. I. Badler, "Real-time inverse kinematics techniques for anthropomorphic limbs," *Graphical models*, vol. 62, no. 5, pp. 353–388, 2000.

[15] M. W. Spong, S. Hutchinson, M. Vidyasagar *et al.*, *Robot modeling and control*. Wiley New York, 2006, vol. 3.

[16] Z. A. Karam, "Hybrid fuzzy congestion controllers for computer networks tuned by modified particle swarm optimization," *International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems*, vol. 7, no. 2, pp. 17–26, 2018.

[17] Z. A. Karam, A. M. Al-Kadhimi, and E. A. Saeed, "Design and implementation of a wireless robotic human hand motion-controlled using arduino," in *2018 International Conference on Advanced Science and Engineering (ICOASE)*. IEEE, 2018, pp. 518–523.

[18] Z. A. Karam, S. S. Ghintab, and S. Hasan, "Rapidly-implementable optimizely-sizable fuzzy controller architectures: A performance analysis for semiconductor packaging two axes table," *Periodicals of Engineering and Natural Sciences*, vol. 7, no. 2, pp. 644–659, 2019.

[19] S. S. Ghintab, Z. A. Karam, and S. Hasan, "Artificially-intelligent robotic space manipulator using fuzzily-architected nonlinear controllers," *International Journal of Advanced Technology and Engineering Exploration*, vol. 9, no. 87, p. 180, 2022.

[20] A. C. Abad, D. D. Ligutan, E. P. Dadios, L. J. S. Cruz, M. C. D. Del Rosario, and J. N. S. Kudhal, "Fuzzy logic-controlled 6-dof robotic arm color-based sorter with machine vision feedback," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 5, 2018.

[21] A. N. A-Amir and H. A. Akkar, "Artificial intelligent fuzzy logic controller applied on 6dof robot arm using labview and fpga," *European Journal of Engineering and Technology Research*, vol. 3, no. 5, pp. 1–8, 2018.

Zeyad A. Karam he is a control and automation engineer, had B.Sc. in Control Engineering and, M.Sc. in Robotics Engineering from University of Technology, Iraq. Now, he is member in the staff Lecturer for the COIE college, Al- Nahrain University. With more than 8 years of experience in research and teaching, he had a research of seventeen specialist articles. His research area involves Nonlinear control, Robot control and dynamic, intelligent controllers, optimization algorithms, intelligent model design and control.

Elaf A.Saeed is a systems and control engineer at the University of Al-Nahrain, college of information engineering, Iraq. Information Engineering College. she works in the fields of control, embedded systems, Artificial Intelligent, and web design and has a programming talent, had eleven books published in lambert academic. Elaf was ranked as the first student on her stage for all years in B.Sc.