Received: 15 October 2020 | Revised: 11 February 2021 | Accepted: 25 March 2021

DOI: 10.1049/cth2.12140

IET Control Theory & Applications

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

A control structure for ambidextrous robot arm based on Multiple Adaptive Neuro-Fuzzy Inference System

Mashood Mukhtar[1] | Dhayaa Khudher[2] | Tatiana Kalganova[1]

[1] Electronic and Computer Engineering, Brunel University London, London, UK

[2] Computer Engineering, Basra University, Basra, Iraq

Correspondence
Mashood Mukhtar, Howell Building 307, Electronic and Computer Engineering, Brunel University London, London, UB8 3PH, UK.
Email: Mashood.Mukhtar@brunel.ac.uk

Abstract

This paper presents the novel design of an ambidextrous robot arm that offers double range of motion as compared to dexterous arms. The arm is unique in terms of design (ambidextrous feature), actuation (use of two different actuators simultaneously: Pneumatic Artificial Muscle (PAM) and Electric Motors)) and control (combined use of Proportional Integral Derivative (PID) with Neural Network (NN) and Multiple Adaptive Neuro-fuzzy Inference System (MANFIS) controller with selector block). In terms of ambidextrous robot arm control, a solution based on forward kinematic and inverse kinematic approach is presented, and results are verified using the derived equation in MATLAB. Since solving inverse kinematics analytically is difficult, Adaptive Neuro Fuzzy Inference system (ANFIS) is developed using ANFIS MATLAB toolbox. When generic ANFIS failed to produce satisfactory results due to ambidextrous feature of the arm, MANFIS with a selector block is proposed. The efficiency of the ambidextrous arm has been tested by comparing its performance with a conventional robot arm. The results obtained from experiments proved the efficiency of the ambidextrous arm when compared with conventional arm in terms of power consumption and stability.

1 | INTRODUCTION

A robot arm plays an important role in determining a robot's capability as most of the tasks require some kind of end-effector to complete the task. The adroitness of the human arm to perform complicated operations has resulted in high demand across various industries. Literature reveals much work already completed on the design and control of robotic arms [1, 2]. Robotic arms that can offer clever manipulating, grasping, lifting and sense of different objects have always been highly desirable in industry due to their wide scope in many applications such as teleoperation, mobile.

Numbers of robotic arms have been developed in the past few decades to offer solutions to industry and humankind [3]. DLR lightweight robot III developed in 2003, KUKA robot arm LBR iiwa in 2013, Robonaut arm by NASA [4]. Delft robot arm developed by TU Delft University of Technology is low power and low mass safe manipulator offering four DOF. The Delft arm won the Amazon Picking challenge in 2016 [5]. The OpenArm v.2.0 is a low cost 7 DOF robotic arm that is actuated by servo motors. This arm is made keeping human safety in mind and comes with teleoperation control scheme [6]. The WAM arm developed in 2010 by Barret Technology Inc. is highly dexterous. Naturally back-drive manipulators also known as the most advanced robotic arm in the world by Guinness World Record Millennium Edition. It is available in two main configurations, four degrees-of-freedom and seven degrees-of-freedom.

RE2 offers innovative end effectors ranging from small to large arms. Their most famous product is a highly dexterous manipulation system (HDMS) is capable of doing complicated tasks. The arm itself is highly dexterous, efficient and affordable. More than two decades of experience make SCHUNK one of the most important developers of manipulator and gripper systems. Five-finger gripper hand SVH is designed for higher productivity in service robot applications [7]. ST robotics developed an R12 collaborative robot arm and R17 robot arm. Both arms are low cost five-axis articulated using servos. R12 arm offers fast, quiet and fantastic performance for the price [8]. KUKU offers tailor-made automation solution for the industry. They have a wide range of products that suit industry needs. KUKU arms are quite reliable. KUKU KR1000 titan is one of the powerful robots built for heavy loads. This six-axis robots move heavy parts safely and precisely [9]. Bionic arm developed by a Bristol start-up company called open Bionics released a new range of hero arms that could be fitted to patients from nine years old to an adult of any age. It is the world's first medically certified 3D printed arm and costing around £10,000 is considered one of the cheapest on the market [10].The AMO arm developed by Ryerson University is controlled by brain signals. The AMO arm enables amputees to avoid invasive surgeries and could potentially save money in the long run. It is controlled by the user's brain signals and powered by a pneumatic system [11]. Since none of the robot arms discussed is capable of ambidextrous movement, this research aims to propose an ambidextrous robot arm design that could offer a much greater range of movement than a conventional arm.

Precision control of a robotic arm is a challenging task especially when the design of the arm does not meet conventional parameters. Different mechanical designs naturally lead to different control solutions. As a result, many control solutions have been proposed over the last few decades. For instance, in [12] the author proposed to determine the joint motion of the end effector by evaluating the feasibility of the joint motion. The determined joint motion is called an inverse kinematic solution with singularity robustness because it denotes a feasible solution even at or in the neighbourhood of singular points. The robust singularity inverse (SR-inverse) is introduced as an alternative to the pseudoinverse of the Jacobian matrix. Several simulation results are also shown to illustrate the singularity problem and the effectiveness of the inverse kinematic solution with singularity robustness.

In [13] a novel algorithm for the adaptive control of a robot manipulator containing kinematic loops is presented. The algorithm identifies the mass properties of each link and the viscous friction coefficients for each joint of the manipulator. It is similar to the Newton-Euler inverse dynamics algorithm and, hence, obtains its computational efficiency through the recursive nature of the algorithm. Simulation results presented show the effectiveness of the controller. Similarly, in [14], the author considered the adaptive control of robotic manipulators in task space or Cartesian space. A general Lyapunov-like concept is used to design an adaptive control law. From the results, it is verified that global stability and convergence can be achieved for the adaptive control algorithm. The algorithm has the advantage that the inverse of the Jacobian matrix is not required. A robust control method using a switching-sliding algorithm for a planar dual-arm manipulator system is developed in [15]. The proposed controller is useful for modelling imprecision and disturbances, inertial-based problems, as well as space-based free-floating platforms.

Most of the research on robot trajectory control has assumed that the kinematics of the robot are known precisely [16, 17]. However, when a robot picks up tools of uncertain lengths, orientations, or gripping points, the overall kinematics becomes uncertain and changes according to different tasks. To overcome this problem, a new adaptive Jacobian tracking controller for robots with uncertain kinematics and dynamics is presented, and experimental results justify the performance of the proposed controller in [18].

A neural network controller for a mobile manipulator to track the given trajectories is introduced in [19]. The dynamics of the mobile manipulator are assumed to be unknown and learned online by the Radial Basis Function Network (RBFN) with weight adaptation rule derived from the Lyapunov function. Generally, an RBFN can be used to approximate a non-linear function accurately. However, there remains some approximation error inevitably in a real application. An additional control input to suppress this kind of error source is also used. Simulation results confirmed the effectiveness of the system in an unknown workspace.

In [20], the authors investigated the implementation of inverse kinematics and a servo controller for a robot manipulator using a Field Programmer Gate Array (FPGA). They have evaluated the performance of the proposed circuit design through an experimental system that consisted of the FPGA-based motion controller and a Mitsubishi RV-M1 micro-robot and collected the experimental results to evaluate correctness and effectiveness. Similarly, in [21] an inverse kinematics method to control the servo angles of five DOF arm joints to get the desired tip position controlled by teleoperation is proposed. A strategy for solving the inverse kinematics equations of a six DOF robot arm system, using the robot arm assembled by seven Artificial Intelligence (AI) servos is proposed in [22]. A five DOF robotic arm driven by servo motors is controlled using an SSC-32 control board in [23]. The author added another servo to rotate the gripper and proved the concept of controlling all actuators using a single board.

The main contributions of this paper are twofold. Firstly, we present a new MANFIS controller to get rid of the singularity problem associated with the inverse kinematics. Secondly, we proved the efficiency of the ambidextrous robot arm in term of power consumption by comparing the robot with other conventional robot arm. This paper is focused on presenting the novel design structure of a robotic arm with ambidexterity, a unique actuation system and control of such a complex system using an artificial intelligence based method.

The rest of the paper is organized as follows: the kinematics modelling of the ambidextrous arm is summarized in Section 2. In Section 3 the inverse kinematics of the ambidextrous arm is presented. In Section 4 the adaptive network fuzz inference system is discussed. The ANFIS controller design for the ambidextrous arm is explained in Section 5. In Section 6 the efficiency of the ambidextrous robot arm is proved. Finally, the paper is concluded in Section 7.

2 | KINEMATICS MODELLING OF THE AMBIDEXTROUS ARM

The robot arm has five revolute joints as shown in Figure 1. The length of each link is defined to be the distance between adjacent

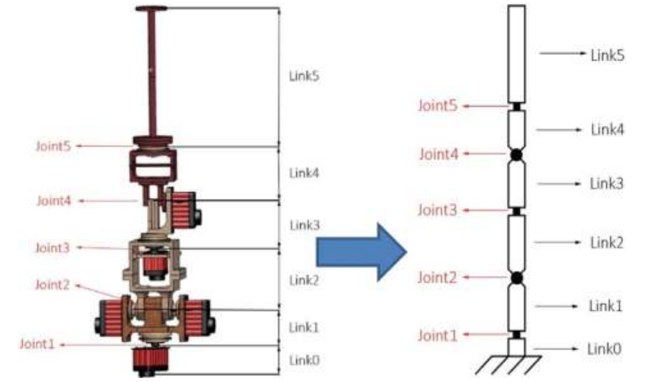


FIGURE 1 Ambidextrous robot arm mechanical structure translated into links and joints

joint axes. Servo motor that is driving link one is permanently fixed to the base of an ambidextrous arm, θ_1 represents the angle between the x-axis and link 1. θ_2 is the angle between link 1 and link 2, θ_3 is the angle between link 2 and link 3. An actuator driving link 4 generating θ_4 and θ_5 is generated by actuator driving link 5. Robotic manipulators are designed to perform various tasks mostly using end effectors. So in order to perform such tasks, the position and orientation of the end effector must be known. Then, the position and orientation of the end effector in terms of a joint variable are calculated. This technique is called forward kinematics. Calculating forward kinematics of a robot is often considered the very first step in robotic research. Denavit Hardenberg (DH) approach has been used to determine the forward kinematics and to assign the axis to movable joints. There are various approaches to model the robot arm such as Screw Theory representation [24]; Hayati Roberts [25] and geometric modelling DH approach [26] are suitable for the ambidextrous robotic arm structure. Other approaches may be beneficial only in the case where the DH approach does not handle parallel z-axis.

The DH convention describes a systematic way to develop the forward kinematics of any robot. The kinematic analysis allows the designer to obtain key information on the position and orientation of each joint and link within the mechanical structure. This information is necessary for subsequent dynamic analysis along with control paths. The transformation set given below can be used to locate $i-1$ axes of a point x_i (revolute joint) placed on the ith link (Figure 2).

Using the DH convention, θ_i describes joint angle of x_i axis relative to x_{i-1} axis defined according to the right-hand rule about Z_{i-1} axis, distance from the origin is denoted by d_i of the $i-1$ axes to the intersection of the Z_{i-1} axis with the x_i axis and measured along the Z_{i-1} axis, $a_i$is minimum distance between Z_{i-1} and Z_i and α_i describes an offset angle of Z_i axis relative to Z_{i-1} axis measured about the x_i axis using right-hand rule to obtain the forward kinematics transformation matrix T_n^0 based on homogenous transformations and DH convention. All the reference systems are located that are required in making sure the DH coordinate frame assumptions are satisfied. Then the table of link parameter is created as shown in Table 1.

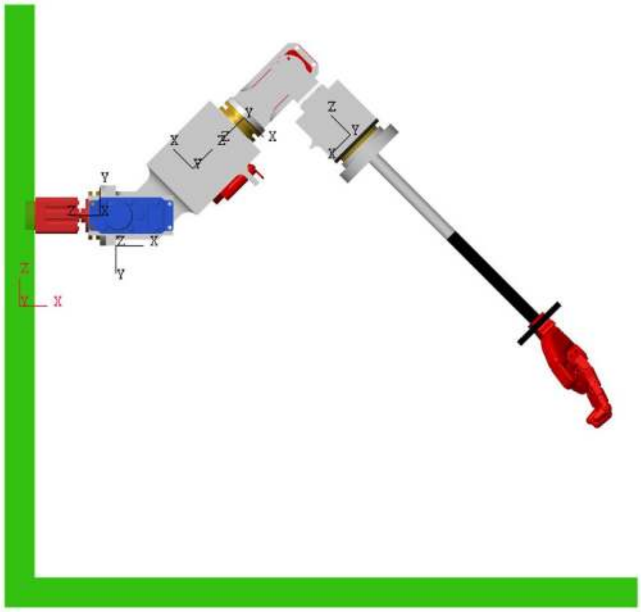


FIGURE 2 Kinematic configuration and joint model of the five-joint ambidextrous robotic arm

TABLE 1 DH parameter of a 5-DoF robot arm

Link	a_i	d_i	α_i	θ_i
1	l_1	0	90°	θ_1
2	0	0	−90°	θ_2
3	$l_2 + l_3$	0	90°	θ_3
4	0	0	−90°	θ_4
5	$l_4 + l_5$	0	0	θ_5

The relative translation and rotation between ith and $i-1$ coordinate systems (adjacent links) can be represented as a homogeneous transformation matrix:

$$T_1^0 = \begin{bmatrix} C\theta_1 & 0 & S\theta_1 & 0 \\ S\theta_1 & 0 & -C\theta_1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$T_2^1 = \begin{bmatrix} C\theta_2 & 0 & -S\theta_2 & 0 \\ S\theta_2 & 0 & C\theta_2 & 0 \\ 1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$T_3^2 = \begin{bmatrix} C\theta_3 & 0 & S\theta_3 & 0 \\ S\theta_3 & 0 & -C\theta_3 & 0 \\ 0 & 1 & 0 & l_2 + l_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

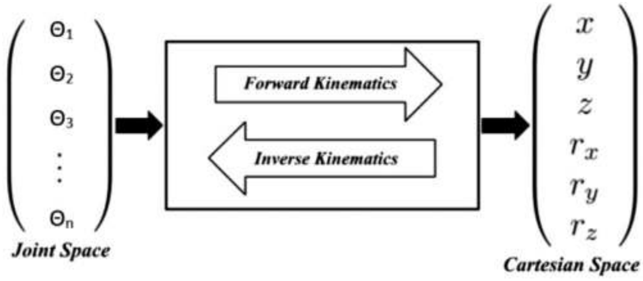


FIGURE 3 Forward and inverse kinematics relationship

$$T_4^3 = \begin{bmatrix} C\theta_4 & 0 & -S\theta_4 & 0 \\ S\theta_4 & 0 & C\theta_4 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$T_4^5 = \begin{bmatrix} C\theta_5 & -S\theta_5 & 0 & 0 \\ S\theta_5 & C\theta_5 & 0 & 0 \\ 0 & 0 & 1 & l_4 + l_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

where T is the transformation matrix, C and S are short for cos and sin respectively.

The end effector reference frame can be expressed with respect to the arm base frame as Equation (6):

$$T_5^0 = \prod_{i=1}^{5} T_i^{i-1} \quad (6)$$

3 | INVERSE KINEMATICS OF THE AMBIDEXTROUS ARM

Unlike forward kinematics, finding an inverse kinematic solution is relatively hard in particular when dealing with multiple DOF robots. Usually, there is always more than one inverse solution and choosing the best solution by specifying the type of configuration the user prefers is key to moving the robot arm to the desired position. For instance, a one revolving joint robot arm will have only one possible inverse solution to define the position of the end effector while a 6 revolving joint robot may have 16 different solutions to define the same position of the end effector. The relationship between joint space and Cartesian space as well as forward kinematics and inverse kinematics is shown in Figure 3.

In inverse kinematics the most challenging task is to solve the complicated equations and to deal with multiple possible solutions. The complexity of this problem and possible alternative approaches are discussed in [27]. Some simulation based platforms exist that do all the calculations but sometimes selecting the best one is difficult. In order to explain the difficulty of solving the inverse kinematics problem, the position of the hand (d_x, d_y, d_z) will be formulated with respect to arm base frame as Equation (7):

$$x\left(d_x,\ d_y,\ d_z\right) = f(\theta) \quad (7)$$

By differentiating both sides of Equation (7) with respect to θ, the velocity in the task space of the hand will result as Equation (8).

$$\dot{x} = J\dot{\theta} \quad (8)$$

where $\dot{x}$ is the velocity at task-space, J is the robot arm Jacobian matrix that map the position and the orientation of the hand to the joint-space, and $\dot{\theta}$ is joints velocity. The problem of the inverse kinematic is to get the velocity in configuration space ($\dot{\theta}$) by giving the velocity in task space ($\dot{x}$). Therefore, Equation (8) should be inverted to get a linear form as Equation (9).

$$\dot{\theta} = J^{-1}\dot{x} \quad (9)$$

It is clear from Equation (9) that the matrix of Jacobian is not square. So, the inversion process is not straightforward. Many methods in literature have dealt with this problem either analytically or numerically [28]. One of the most commonly used methods is an ANFIS.

4 | ADAPTIVE NETWORK FUZZY INFERENCE SYSTEM (ANFIS)

AN is a network structure consisting of nodes and directional links, and in practice, AN is considered a superset of multilayered feed forward Neural Network (NN) with supervised learning capabilities. The basic rule of AN is based on gradient descent [29] and the chain rule [30]. ANFIS utilises network topology to reduce the optimisation search space. The design objective of the fuzzy controller is to learn and improve in terms of performance despite the system facing disturbances. An ANFIS is similar to an NN that is based on Takagi-Sugeno fuzzy interference system. The objective of ANFIS is to integrate both fuzzy logic and NN principles. It could offer the benefit of both in a single framework and be considered as a universal estimator. ANFIS is best option to choose between neural network and fuzzy systems, providing smoothness (due to fuzzy control) and adaptability (due to neural network back propagation). ANFIS corresponds to a set of fuzzy if-then rules that have learning capability to approximate non-linear functions. Fuzzy if-then rules express conditions IF A THEN B, where A and B are fuzzy set labels characterised by appropriate Membership Function (MF). If then rules help the user make decisions in an uncertain and imprecise environment. Thus, a hypothesis is created from the parameterised mathematical model and data is generated using forward kinematics (due to quick and straightforward outcomes). The NN is used to tune the MF of Sugeno type fuzzy interference system.

There are two types of fuzzy systems: Mamdani and Sugeno models. Fuzzy interference system is mainly composed of five functional blocks as shown in Figure 4.

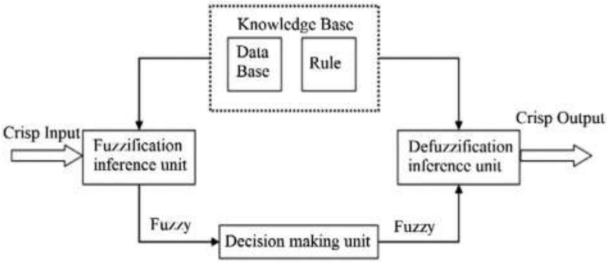


FIGURE 4 Schematic diagram of a fuzzy inference system

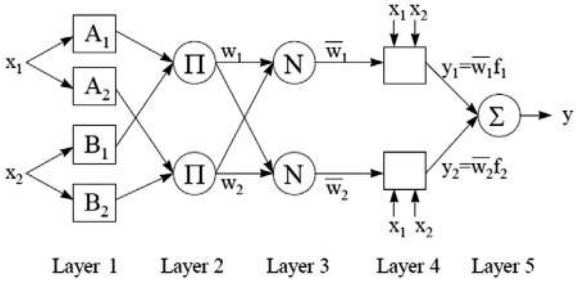


FIGURE 5 ANFIS architecture [31]

There are three main blocks of FIS as listed below

(1) A fuzzification interference unit, which transforms the input into degrees of a match with linguistic values works by comparing the input variables with the MFs to obtain membership values of the linguistic label.
(2) A knowledge-based block is composed of the two units called database unit (define MFs of the fuzzy set) and rule-based unit (contains If-Then rule).
(3) A defuzzification interference unit, which transforms the fuzzy results into an output.

ANFIS is multi-layered feed forward network in which each node performs a particular node function as shown in Figure 5. To represent different adaptive networks both circle node (fixed node) and square nodes (adaptive node) are used. The formula of function may vary node to node as it depends on the overall input-output function. For simplicity, consider a first order Sugeno fuzzy model with two inputs, x and y and one output z. Imagine rule base contains two if-then rules of Takagi and Sugeno:

$$\text{Rule } 1: \text{ IF } (x \text{ is } A_1) \text{ and } (y \text{ is } B_1) \text{ then } (f_1 = P_1\ x + Q_1 y + R_1)$$

$$\text{Rule } 2: \text{ IF } (x \text{ is } A_2) \text{ and } (y \text{ is } B_2) \text{ then } (f_2 = P_2\ x + Q_2 y + R_2)$$

where x and y are the inputs, A_i and B_i are the fuzzy sets, f_i are the outputs within the fuzzy region specified by fuzzy rules: P_i,Q_i and R_i are the design parameters that are determined during the training process.

Layer 1: Every node is a square node (adaptive nodes) in layer 1. Parameters in the layers are called premise parameters. The output of layer 1 is a fuzzy membership grade of the inputs, which are given by Equations (10) and (11)

$$O_i^1 = \mu\, A_i\ (x)\ \ i = 1,2 \tag{10}$$

$$O_i^1 = \mu\, B_i\ (y)\ \ i = 1,4 \tag{11}$$

where $\mu\, A_i\ (x)$, $\mu\, B_i\ (y)$ can adopt any fuzzy MF.

Layer 2: Every node is a circle node in layer 2 as Equation (12).

$$O_i^2 = \mu\, A_i\ (x)\ \mu\, B_i\ (y) \tag{12}$$

where $i = 1 : 2$ the output of this layer can be represented as a firing strength of the rules.

Layer 3: Every node is a circle node in layer 3 as Equation (13).

$$O_i^3 = \overline{W_i}\ f_i = \frac{W_i}{\left(\left|\overline{W_1 + W_2}\right|\right)} \tag{13}$$

where $i = 1 : 2$ the ith node calculates the ratio of ith rule's firing strength to the sum of all firing strength.

Layer 4: Every node is a square node in layer 4. The output of each node in this layer is a square is a product of the normalized firing strength and first-order polynomial as Equation (14).

$$O_i^4 = \overline{W_i}\ f_i = W_i\,(P_i x + Q_i y + R_i) \tag{14}$$

Layer 5: This layer has only one node that sums all incoming signals and represents the overall output of the model. This node performs the summation of all incoming nodes as Equation (15).

$$O_i^5 = \overline{W_i}\ f_i = \frac{\sum_{i=1}^{2} W_i f_i}{(W_1 + W_2)} \tag{15}$$

ANFIS uses a two-pass learning cycle: feed forward and backward pass. In feed forward pass nodes outputs go forward until layer 4, S1 is unmodified, and S2 is computed using LSE [32]. On the other hand in backward pass, error signal propagates backwards, S2 is unmodified, and S1 is computed using gradient descent algorithm [29]. It is apparent from Figure 6, when the values of premise parameters are fixed, the output can be represented as a linear combination of consequent parameters.

The Neuro-Fuzzy Designer app (as shown in Figure 8) is used to design, train, and test adaptive ANFIS using input/output training data. It is useful in modifying interference system before tuning MF of Sugeno type FIS, based on training data generate an initial inference system, prevent over fitting to the training data, using testing data test the generalisation ability of tuned system and export tuned data to MATLAB workspace.

The process of using ANFIS technique involves data generation (hypothesis a model structure) of all possible angles lying within the possible joint range of movement. The forward kinematic formula is used at this stage to deduce a combination of

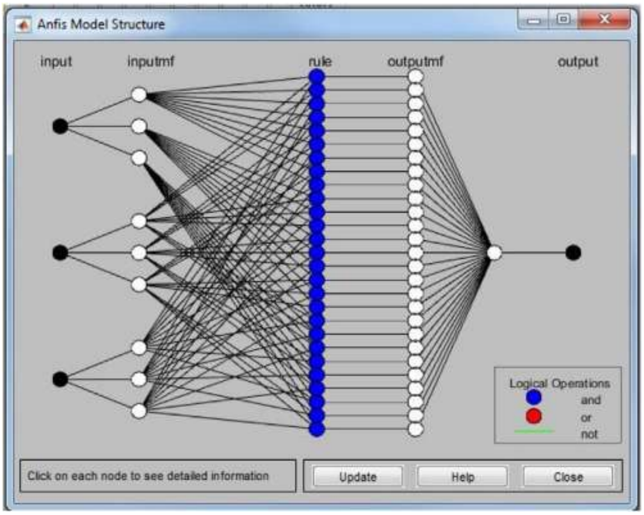
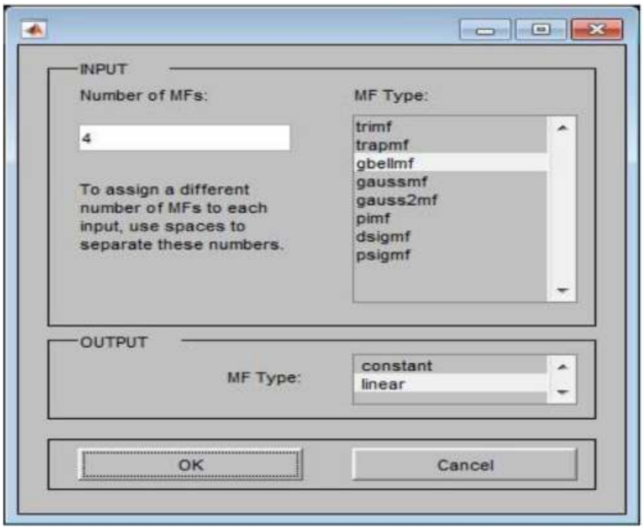


FIGURE 6 ANFIS model structure



INPUT

Number of MFs: 4

To assign a different number of MFs to each input, use spaces to separate these numbers.

MF Type: trimf, trapmf, gbellmf, gaussmf, gauss2mf, pimf, dsigmf, psigmf

OUTPUT

MF Type: constant, linear

FIGURE 7 MF input window in MATLAB

all theta values. Then, the ANFIS solution is built specifically to address the problem in question. An ANFIS network can only predict angles when they are trained with sample input-output data. After training the network an important step is to validate the network to determine how well the ANFIS network would perform inside a large control system. Until a satisfactory solution is found, parameters to the ANFIS function may be tweaked. Finally, in the large control system, the trained ANFIS network is used as a reference to determine what angles of the arm should be given to reach the desired location of the manipulator. The system will apply appropriate force on each joint to make a move once knowledge of desired angles and the current angle are known.

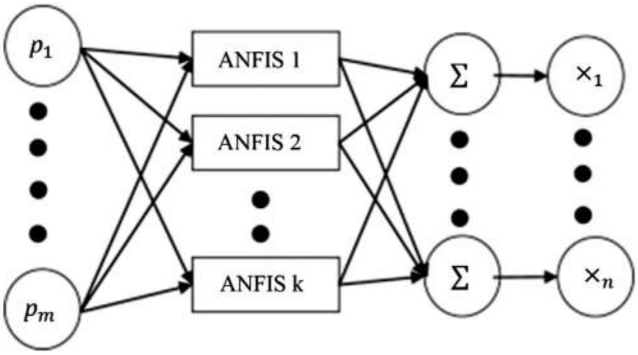


FIGURE 8 General MANFIS architecture

5 | ANFIS CONTROLLER DESIGN FOR THE AMBIDEXTROUS ARM

This section describes an ANFIS network developed and trained to control the ambidextrous robot arm. For simplicity, the inverse kinematic problem is further divided into the end-effector orientation problem and end-effector position problem. It is apparent from Figure 2 that Joint 1, Joint 2 and Joint 4 can drive the end-effector to any given position with the workspace of the ambidextrous robot arm. Joint 5 is only important if the problem in question requires end–effector orientation to be considered. Any application of ANFIS requires detailed knowledge of fuzzy logic as ANFIS demands a careful choice of suitable shape and MFs. Choice of careful selection of such parameters affects not only the efficiency of the ANFIS model but also computational cost. Various MFs can be used to solve any given problem as shown in Figure 7. A Gaussian shape of the MF is a very popular choice due to its smooth representation of input space. Some tests were performed to find the ANFIS network that is most suitable for the problem in question.

The training error can be reduced by changing the key parameters such as membership function, increasing the number of the input membership function, increasing the number of epochs or increasing the training data. However, the key to determining the most suitable ANFIS network lies in a balancing exercise to find the mean point between network computing time and training error.

ANFIS has one output, and in order to move multiple joints, multiple ANFIS networks are required as shown in Figure 8. For the ambidextrous robot arm specifically five ANFIS networks namely ANFIS-1, ANFIS-2, ANFIS-3, ANFIS-4, and ANFIS-5 are used to solve the problem of inverse kinematics. Multiple ANFIS also known as (MANFIS) is modelled in Simulink software as shown in Figure 9. The MANFIS maps the input in task space to the joint angles in joint space, and joint angles are used to determine the desired trajectory.

Figure 9 shows a Simulink diagram for the controller. The controller contains five ANFIS with six inputs (x, y, z, Rx, Ry and Rz) and five output (θ_1, θ_2, θ_3, θ_4 and θ_5).

In order to evaluate the ability of the controller to solve the inverse kinematics problem, the controller has been tested with three paths. The first path in 2D plane (y-z) and the other paths in 3D space (z-y-z) as shown in Figure 10.

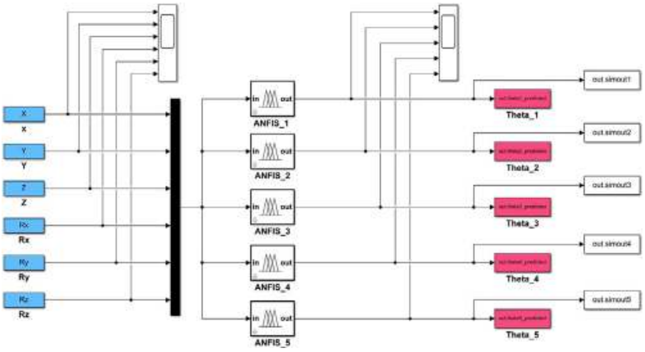


FIGURE 9 MANFIS modelled in Simulink software

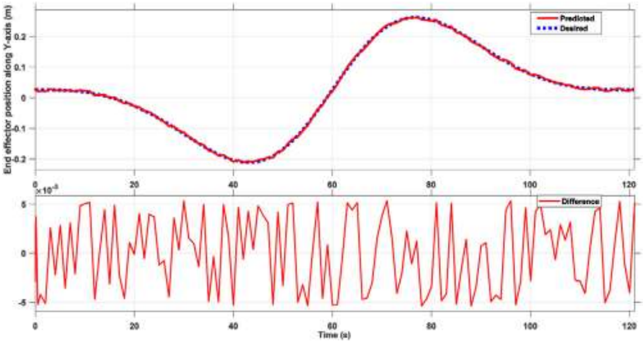


FIGURE 12 The hand position along *y*-axis for the circle path in *y*-*z* plane

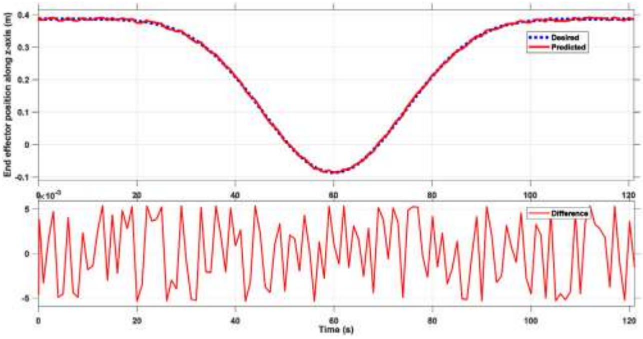


FIGURE 13 The hand position along *z*-axis for the circle path in *y*-*z* plane

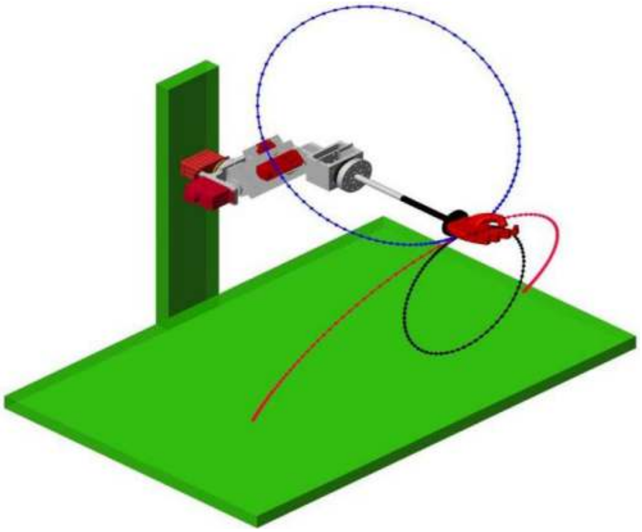


FIGURE 10 The desired paths in the task space. The black circle represents a circular path in *y*-*z* plane. The two other paths in *x*-*y*-*z* are illustrated by a red arc and a blue circle respectively

The results will be presented in the following figures for both the desired and the response of the controller in same figure. Further, the desired path and the predicted one along each axis (*x*, *y* and *z*) combined with difference between the two paths. The first evaluated path will be the circle in *y*-*z* plane. Figure 11 depicts the desired and the predicted path. The red colour path is produced by the robot hand in the operational space. The average differences between the two path in *y*-axis and *z*-axis are illustrated in Figures 12 and 13 respectively. The maximum difference is about 0.5 cm in both axes. A short video for this experiment is available in [33].

In general, solving inverse kinematic problem in 2D workspace is easy due to the limited effect for the orientation parameters at the end effector of the robot. Therefore, the next evaluation will exploit a circular path in *x*, *y* and *z* axes. Figure 14 shows the desired and predicted paths. The difference between the two paths along *x*, *y* and *z* axes are shown in Figures 15–17 respectively. The maximum error between the two paths is approximately 4 cm. Higher error is expected as

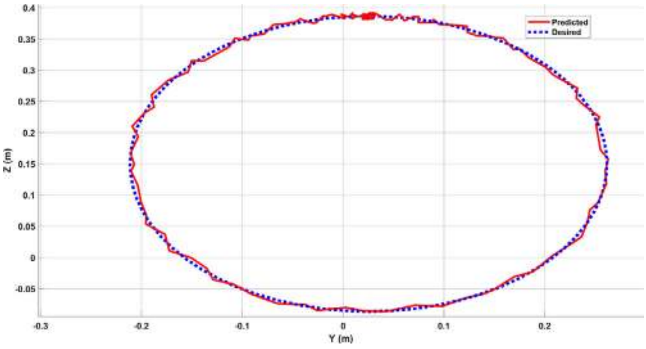


FIGURE 11 The desired path (blue colour) and the predicted path (red colour) in the task space produced by the robot hand

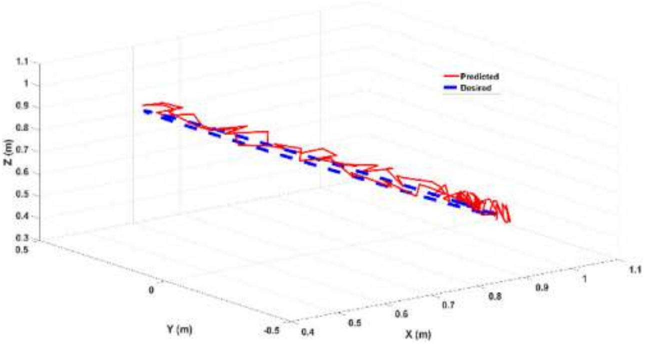


FIGURE 14 The desired path (blue colour) and the predicted path (red colour) in the task space produced by the robot end effector

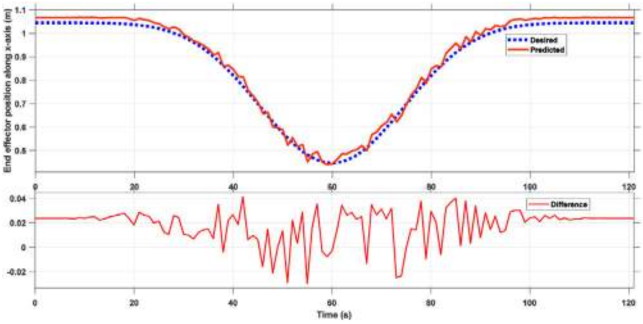


FIGURE 15 The hand position along *x*-axis for the circle path

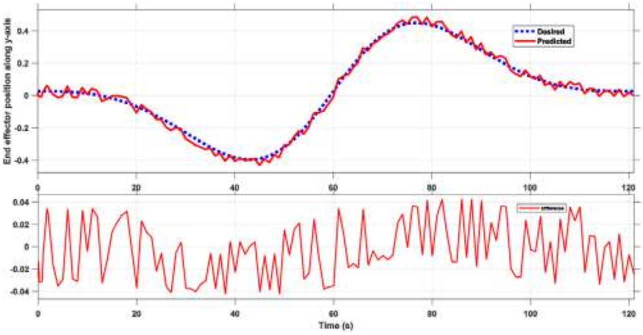


FIGURE 16 The hand position along *y*-axis for the circle path in *y*-*z* plane

more constraints are imposed by the orientation parameters. This suggests a different approach is required for controller to work efficiently in an ambidextrous environment. Therefore a new controller is shown in Figure 20 was designed in Simulink to achieve the ambidexterity element. Five ANFIS networks formed each MANFIS and were driven by a selector block. The idea of using if block (selector) comes from the observation of results where the MANFIS-1 controller can produce a satisfactory result within the specified ranges it is trained for, and the same can be applied to the MANFIS-2 controller. So by having a selector, it is possible to select the best possible controller for the given axis.

From the results shown in Figures 15–17 it is apparent that MANFIS-1 controller that is used to produce the robot path did not produce the satisfactory results. It is noted that the controller failed to produce the trajectory in an all axes. This suggests a different approach is required for controllers to work efficiently in an ambidextrous environment. Therefore a new controller is shown in Figure 18 was designed in Simulink to achieve the ambidexterity element. Five ANFIS networks formed each MANFIS and were driven by a selector block. The idea of using if block (selector) comes from the observation of results where the MANFIS-1 controller can produce a satisfactory result within the specified ranges it is trained for, and the same can be applied to the MANFIS-2 controller. So by having a selector, it is possible to select the best possible controller for the given axis.

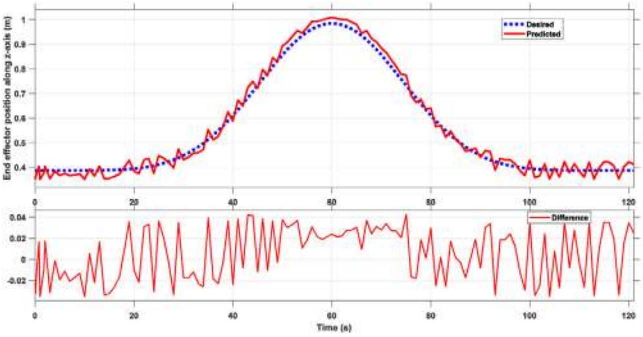


FIGURE 17 The hand position along *z*-axis for the circle path in *x*-*y* plane

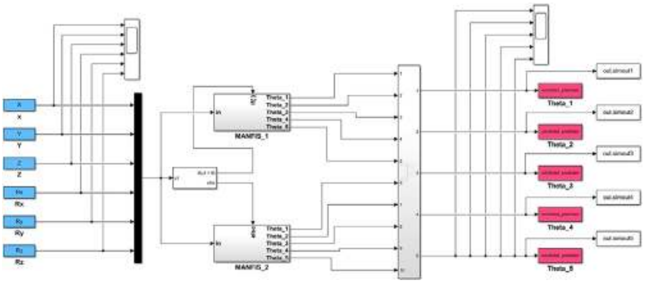


FIGURE 18 Ambidextrous robot arm controller designed in Simulink

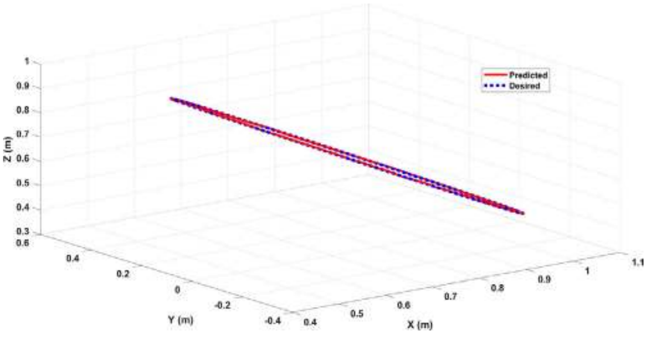


FIGURE 19 The desired path (blue colour) and the predicted path (red colour) in the task space produced by the robot end effector

The previous desired path (circular path in 3D) will be utilized to evaluate the proposed controller. Figure 19 shows the desired and predicted paths. It's obvious that the controller gave a perfect response to produce the joint angles of the robot arm. The maximum error is about 0.2 cm as shown in Figures 20–22. A short video for this experiment is available in [34].

For further validation for the proposed controller, an arc path in 3D will be chosen for this experiment. The diameter of arc path is 105 cm. The wide range of the path will impose more complexity on the controller to generate the joints angles. The results of the desired and the predicted path are presented in Figure 23. Although the desired path has a wide range of motion, the response of the controller to produce the joints angles of the robot was typical. In term of the difference between the two paths, Figures 24–26 clearly illustrate that the maximum error is approximately 0.2 cm, which is acceptable in

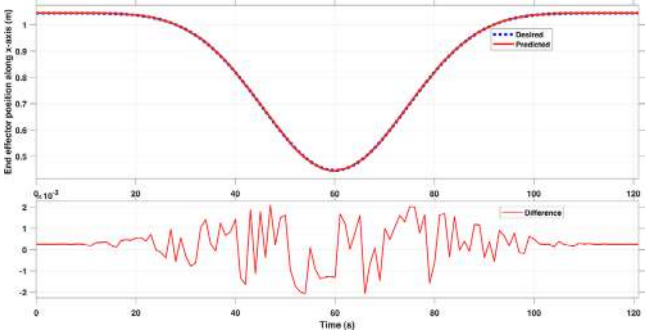


FIGURE 20 The hand position along *x*-axis for the circular path

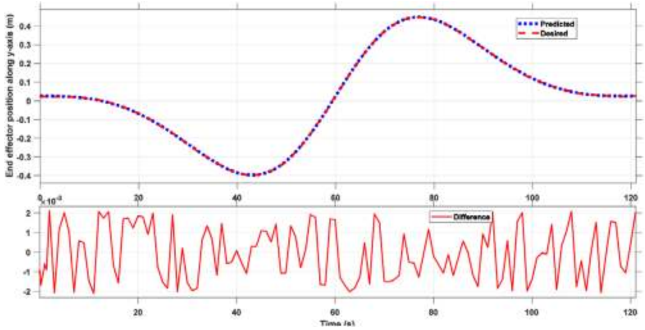


FIGURE 21 The hand position along *y*-axis for the circular path

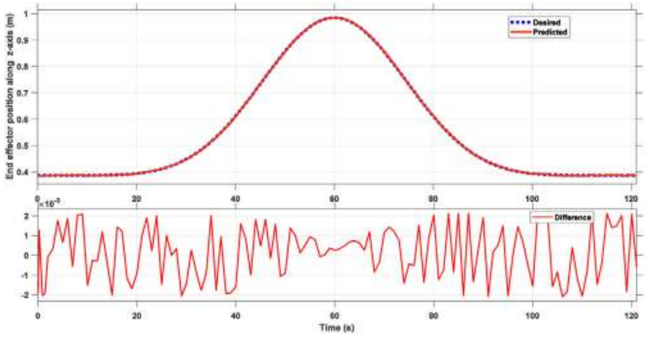


FIGURE 22 The hand position along *z*-axis for the circular path

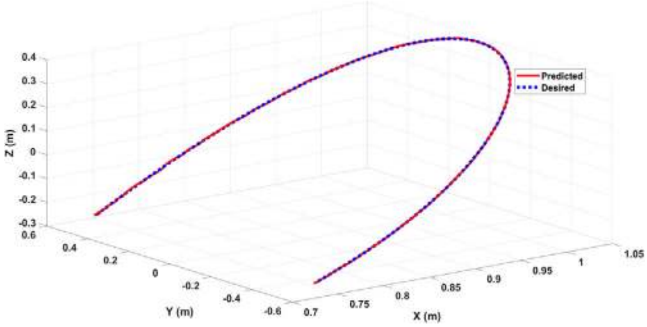


FIGURE 23 The desired path (blue colour) and the predicted path (red colour) in the task space produced by the robot end effector

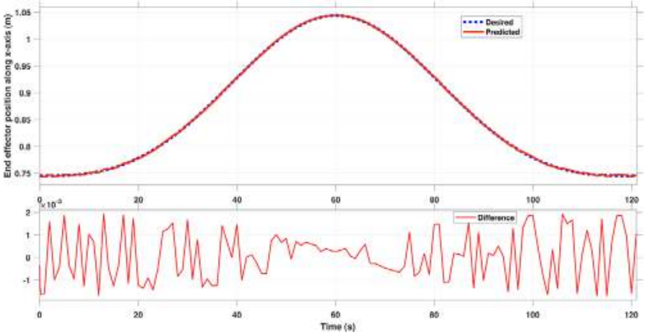


FIGURE 24 The hand position along *x*-axis for the arc path

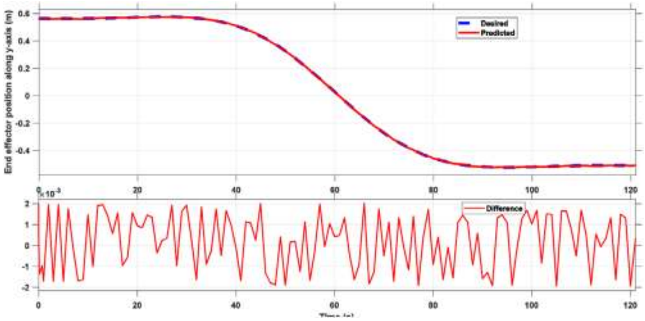


FIGURE 25 The hand position along *y*-axis for the arc path

many applications. A short video for this experiment is available in [35].

In the following experiment, velocity parameter will be inserted to the trajectory by differentiating the input of each ANFIS. The selected path for this evaluation is generated by combining two curves. Figure 27 shows the robot environment and the path. The distances between adjacent nodes of the generated path are not same, which means the robot will move in different speed along the whole trajectory.

Figure 28 presents the desired and the predicted trajectory for the combined curves path. Although the parameters of the velocity have been added to the controller, the robot followed the desired trajectory perfectly. The maximum difference between the desired and the predicted paths is approximately

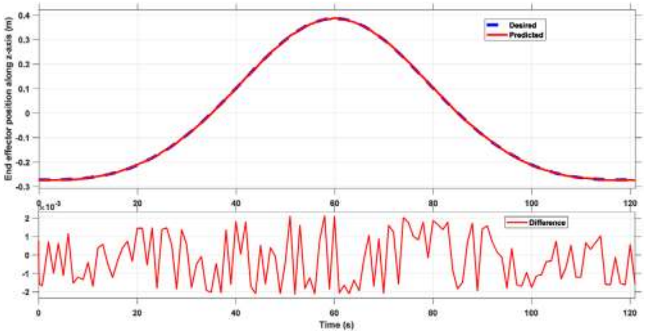


FIGURE 26 The hand position along *z*-axis for the arc path

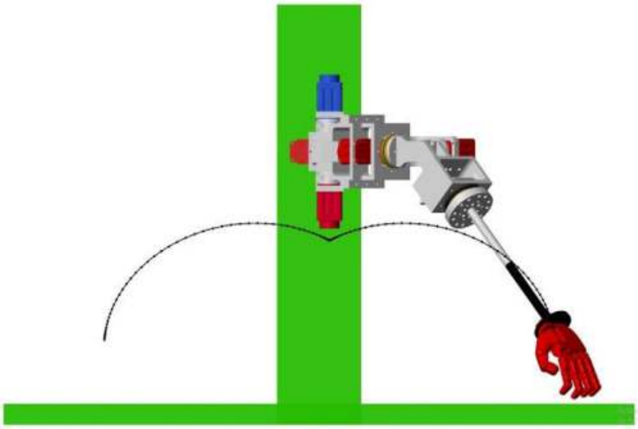


FIGURE 27 The desired path in the task space

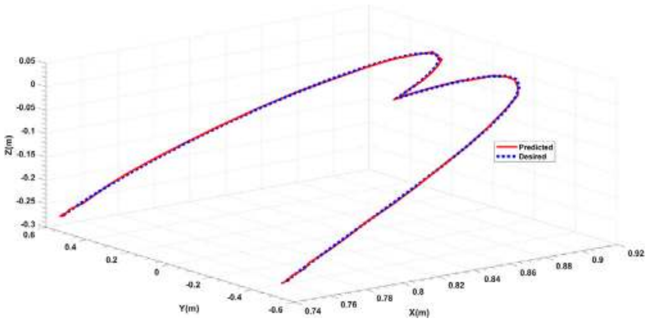


FIGURE 28 The desired paths in the task space for the combined curves path

0.2 cm for very short time (see Figures 29–31). A short video for this experiment is available in [36].

Figure 32 represents the produced joints angles for the ambidextrous robot arm. The joints transitions are very smooth. Further, it's clear from the figure that the velocity of the robot has been dropped slightly in time 50 s for 20 s and then resumed after time 70 s. This period represents the space where the two curves have been combined.

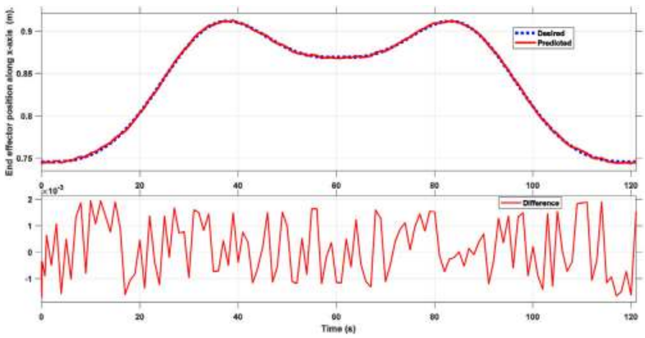


FIGURE 29 The hand position along *x*-axis for the combined curves path

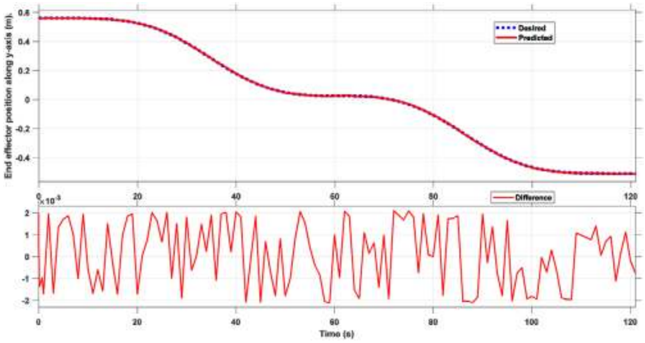


FIGURE 30 The hand position along *y*-axis for the combined curves path

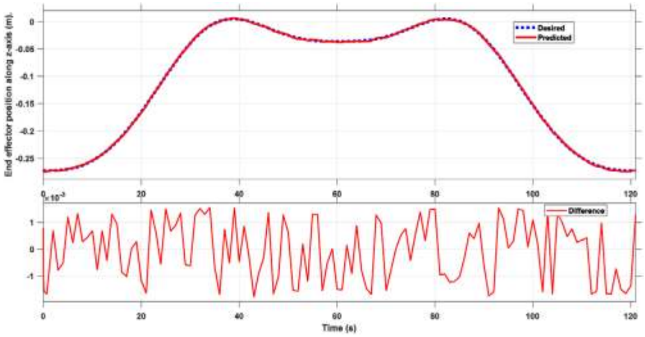


FIGURE 31 The hand position along *z*-axis for the combined curves path

6 | THE EFFICIENCY OF THE ROBOT ARM

Robots are widely used in industry due to their efficiency and high performance. Many of them are employed in industry where the highest percentage of energy is consumed. Therefore, completing tasks with minimal energy consumption has become point of interest for many researchers [37]. Nevertheless, the optimization of the power consumption is still a challenging task. In this section aim is to verify the efficiency of the robot arm in term of the power consumption. The performance of the ambidextrous robot has been compared with a conventional robot. Figure 33 shows a robot arm that used in

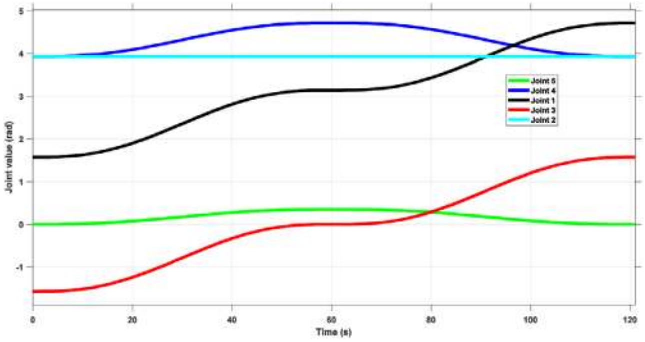


FIGURE 32 The resulted joint values to produce the combined path at the robot hand

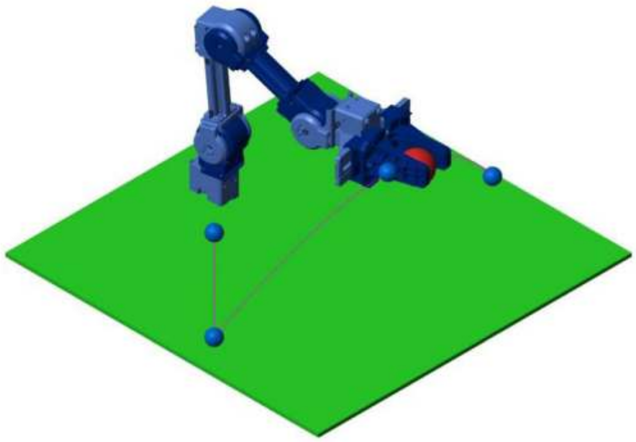


FIGURE 33 A conventional robot arm that is used in the experiments

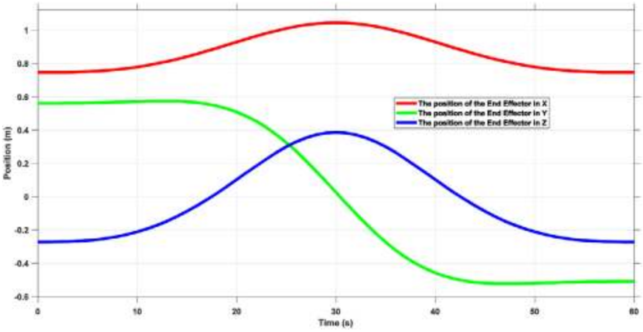


FIGURE 34 The position of the ambidextrous arm in *x*, *y* and *z* axis

the experiments. The model of this robot has been designed in SolidWorks 2018 and the model has been exported to SimMechanics environment.

The two robots follow a semi-circular path. This path is generated using three points in *xyz* plane and the intermediate points have been interpolated using quantic polynomial to get continuous velocity and acceleration for both robots [38].

Figures 34 and 35 show the position of the hand of the ambidextrous arm and the end effector of the conventional arm in all three axes (*X*, *Y* and *Z* axes) respectively.

The experiment is divided in two parts. The first part each robot follows the specified path with no load. The robots carried a load with weight of 0.5 kg by the hand for the ambidextrous arm and by the gripper for the normal robot. A snapshot of ambidextrous arm and conventional arm while passing through the path is depicted in Figures 36 and 37.

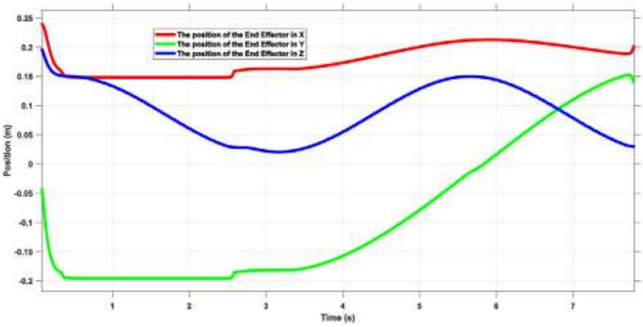


FIGURE 35 The position of conventional arm in *x*, *y* and *z* axis

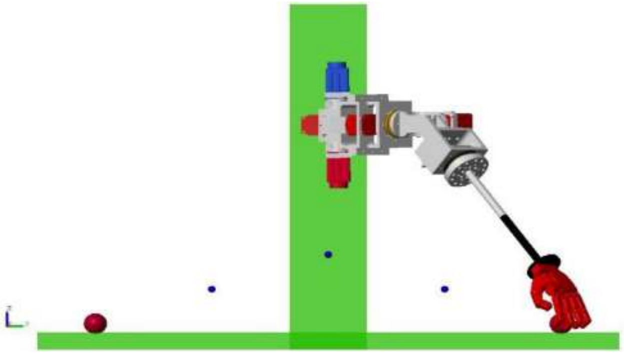


FIGURE 36 Snapshot for the ambidextrous arm and the semi-circular path

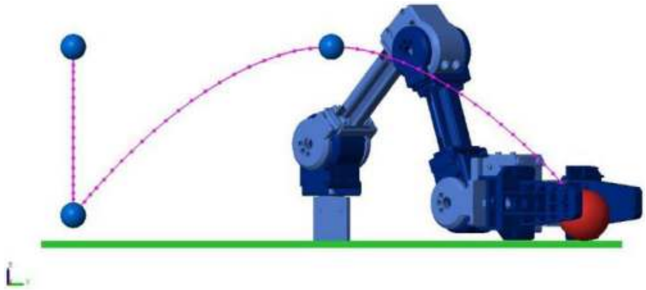


FIGURE 37 Snapshot for the conventional robot arm and the semi-circular path

Two tasks are performed on the ambidextrous arm to validate the design. First torque exerted by the arm on each joint is noted while no load is placed and then with 0.5 kg load. Purpose of this experiment was to see the torque exerted on each joint and then by comparing it with conventional arm performance while doing exactly the same tasks. The torque for both robots has been calculated depending on Euler–Lagrange equation as represented in Equation (16) [39].

$$M(\theta)\ddot{\theta} + H(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau \tag{16}$$

where $M \in \mathfrak{R}^{(5 \times 5)}$, is the inertia matrix of the system, $\ddot{\theta} \in \mathfrak{R}^{(5 \times 1)}$ is the joint acceleration, $H(\theta, \dot{\theta}) \in \mathfrak{R}^{(5 \times 1)}$ is a vector of Coriolis and centrifugal forces, $G \in \mathfrak{R}^{(5 \times 1)}$ is vector of gravity forces, $\tau \in \mathfrak{R}^{(5 \times 1)}$ is a vector of joint torques.

Figures 38–42 depict the torque exerted by the ambidextrous arm.

Similarly, two tasks are assigned to conventional arm to see the torque exerted on each joint in both scenarios. The torque exerted by the conventional arm is shown in figure 43–47.

It is clear from this experiment that the exerted torque at only joint one and two in the ambidextrous arm have been changed

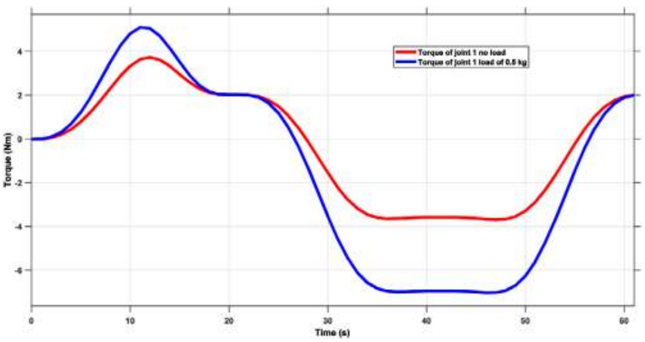


FIGURE 38 The torque exerted by the ambidextrous arm in joint one

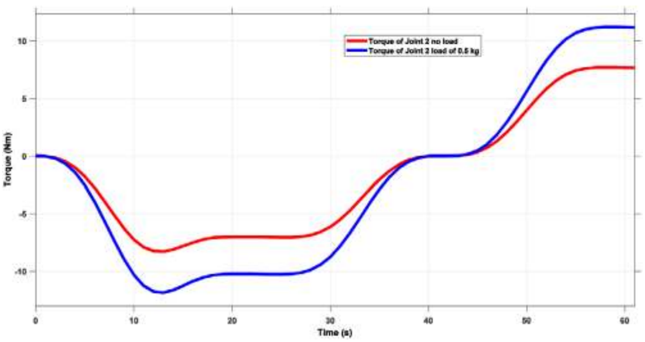


FIGURE 39 The torque exerted by the ambidextrous arm in joint two

when carrying a load. Whereas, the exerted torque at all joint of the conventional robot has been changed. This verifies the effectiveness of the ambidextrous arm design.

In order to verify the efficiency of ambidextrous arm design, acceleration graphs are obtained through experiment. If acceleration stays same in both scenarios (without load and with load) it proves the goodness of the design. By comparing the results how the arm behaves with load and without load, efficiency of the design can be deduced. The acceleration of the ambidextrous arm in x-axis and y-axis is shown in Figures 48 and 49, respectively.

Similarly, the acceleration of the conventional arm in x-axis and y-axis is shown in Figures 50 and 51 respectively. Conventional arm is also tested in both scenarios first without load then

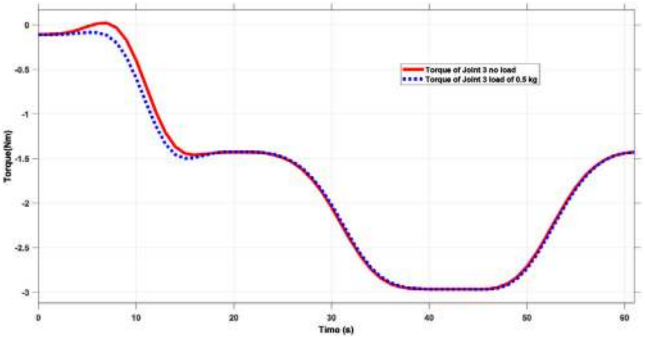


FIGURE 40 The torque exerted by the ambidextrous arm in joint three

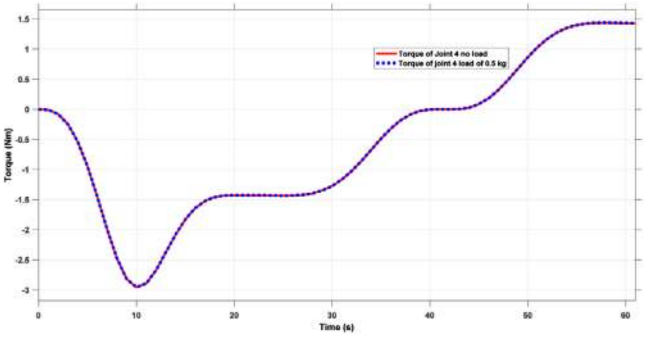


FIGURE 41 The torque exerted by the ambidextrous arm in joint four

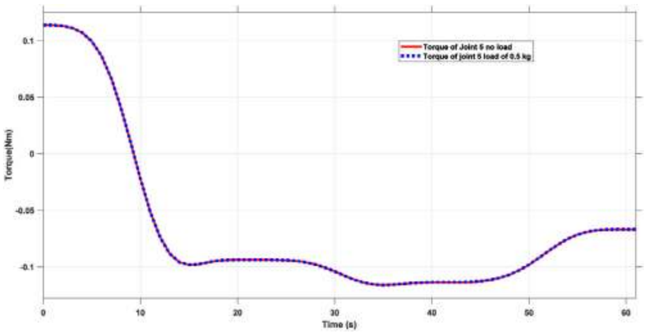


FIGURE 42 The torque exerted by the ambidextrous arm in joint five

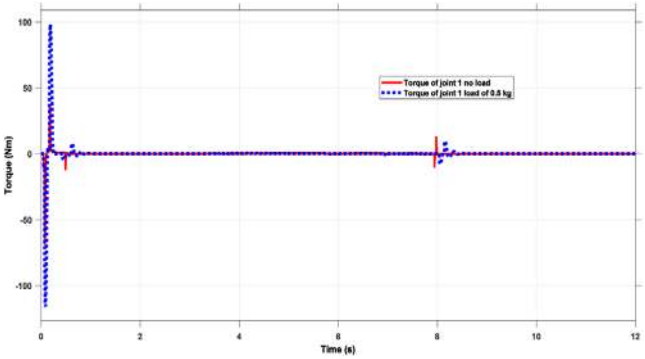


FIGURE 43 The torque exerted by the conventional arm in joint one

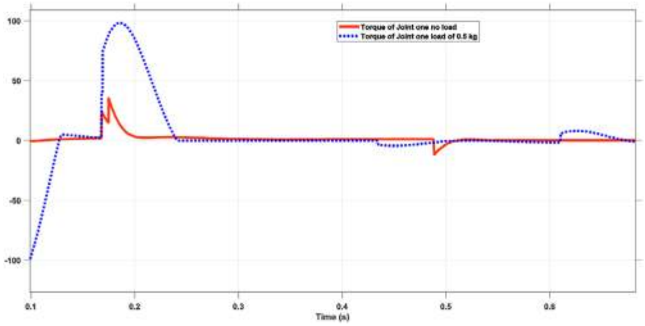


FIGURE 44 Magnified part of the torque exerted in joint one for time from 0 to 0.7 s

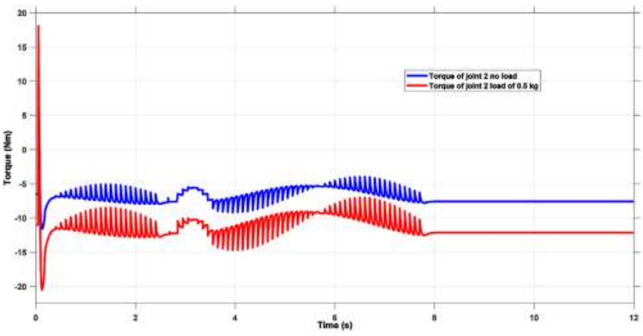


FIGURE 45 The torque exerted by the conventional arm in joint two

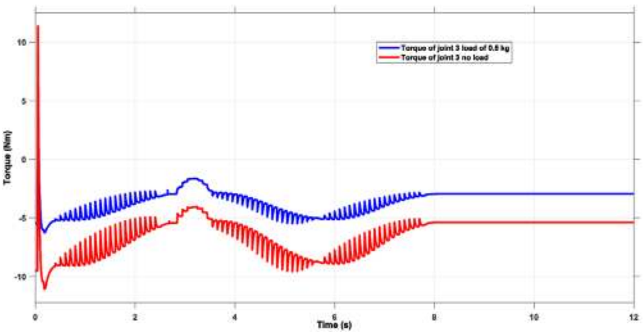


FIGURE 46 The torque exerted by the conventional arm in joint three

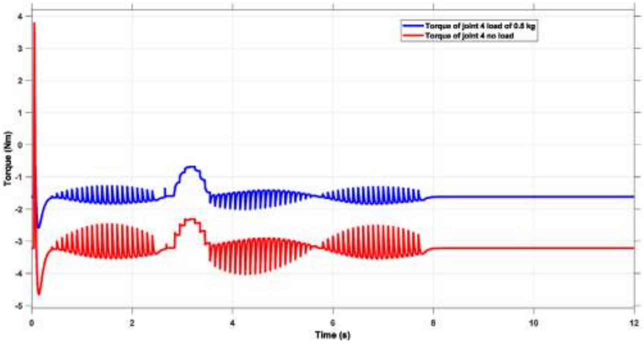


FIGURE 47 The torque exerted by the conventional arm in joint four

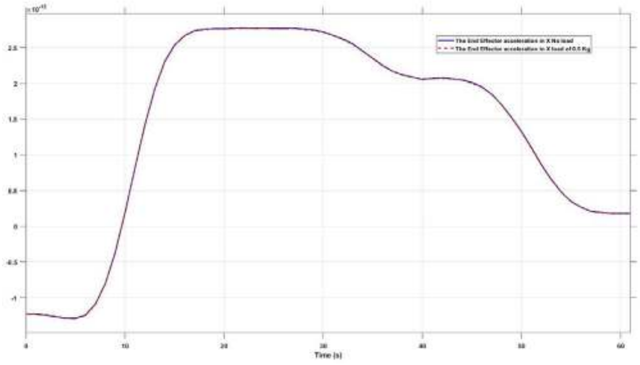


FIGURE 48 The acceleration of the EE in *x*-axis (ambidextrous arm)

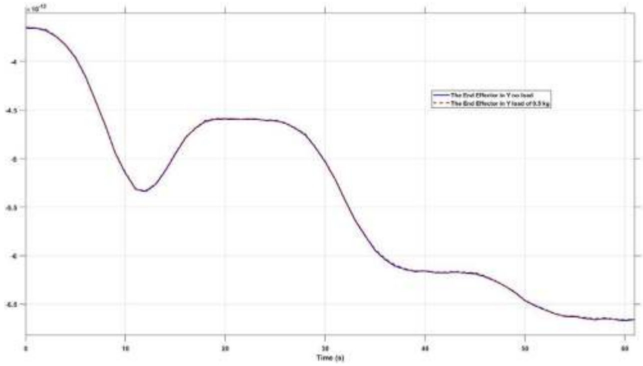


FIGURE 49 The acceleration of the EE in *y*-axis (ambidextrous arm)

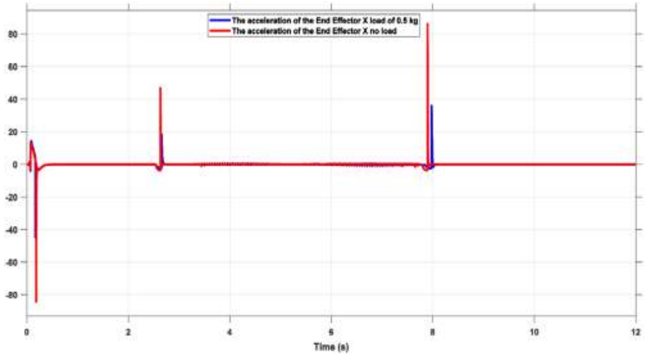


FIGURE 50 The acceleration of the EE in *x*-axis (conventional arm)

with 0.5 kg load. It can be observed from both figures that acceleration has changed.

By comparing Figures 48 and 49 with Figures 50 and 51, goodness of ambidextrous design is evident.

7 | CONCLUSIONS

In this paper, controlling and modelling of the ambidextrous robotic arm were presented. The inverse kinematic problem was discussed in great detail. Due to the complexity of computing inverse kinematic for an ambidextrous robot arm, an artificial

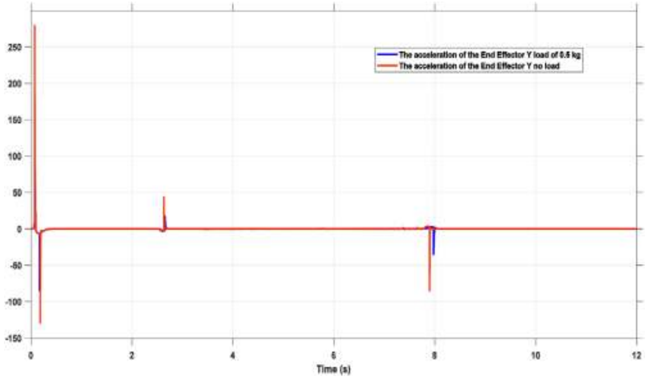


FIGURE 51 The acceleration of the EE in *y*-axis (conventional arm)

neural fuzzy interference system (ANFIS) was employed. The ANFIS controller was trained to control a simple dexterous arm effectively. After satisfactory results were obtained, the difficulty level was increased to control the ambidextrous arm. It is apparent that ambidextrous trajectories cannot be performed using a simple ANFIS-based controller. In the end, it was decided to combine two simple ANFIS controllers to form a controller design for both right and left arm movements. A conventional ANFIS network failed to work alone in a space where the ambidextrous movement is required. Multiple ANFIS were then used and merged with each other for better accuracy. Use of an 'IF' block as a selector to switch between appropriate MANFIS was implemented. A Simulink/MATLAB based model specifically for an ambidextrous robot arm was designed, modelled and tested to confirm the goodness of work. From the trajectory simulation analysis, it was confirmed that the combined use of MANFIS with selector block may be the most appropriate solution for controlling ambidextrous systems in general.

The efficiency of the ambidextrous arm has been tested by comparing its performance with a traditional robot arm. Both the exerted torque and the acceleration at the end effector (in x and y direction) have been used to accomplish the tasks in two cases (load and no load). The experiments have been proved that the effectiveness of the ambidextrous arm over the traditional robot arm in terms of power consumption and the stability.

From the control side, conventional controllers were tested, and their performances are compared to find the most appropriate one. The combined MANFIS controller seems to be the best option to control ambidextrous trajectories. The training time arises when the ANFIS architecture has more than five MFs per each input. Future research should focus on strategies like having ANFIS with multiple outputs with minimal computing time.

Furthermore, a balanced approach needs to be identified in future research that can help the arm to use the shortest possible route to reach the desired point while energy consumption is kept to a minimum.

REFERENCES

1. Gama Melo, D., et al.: Anthropomorphic robotic hands: A review. Ing. Desarrollo 32(2), 279–313 (2014)
2. Bos, H.D., Wassink, M.: Evolution of Robotic Hands. University of Twente (2010)
3. Moran, M.E.: Evolution of robotic arms. J. Rob. Surg. 1(2), 103–111 (2007)
4. Bluethmann, D., et al.: Robonaut: a robot designed to work with humans in space. Auton. Rob. 14(2), 179–197 (2003)
5. Hernandez, C., et al.: Team Delft's Robot Winner of the Amazon Picking Challenge (2016) https://arxiv.org/pdf/1610.05514.pdf
6. Quigley, M., Asbeck, A., Ng, A.: A low-cost compliant 7-DOF robotic manipulator. In: IEEE International Conference on Robotics and Automation. Shanghai, China, pp. 6051–6058 (2011)
7. SCHUNK Mobile Gripping Systems for higher Productivity in your Service Robotics Application (2016). http://www.schunk-modularrobotics.com/en.html. accessed: 2020-09-18
8. Robotics, S.: R17 5-Axis Robot Arm. 2010 (2010). http://www.strobotics.com/. accessed: 2020-08-16
9. KR 1000 Titan (2013). https://www.kuka.com/engb/products/robotics-systems/industrial-robots/kr-1000-titan. accessed: 2020-09-21
10. Hero Arms (2018). https://openbionics.com/press/. accessed: 2020-07-26
11. Trout, C.: AMO Arm Pneumatic Prosthetic does Mind-Control on the Cheap (2011). https://www.engadget.com/2011/04/05/amo-arm-pneumaticprosthetic-does-mind-control-on-the-cheap/?guccounter=1. accessed: 2020-09-14
12. Nakamura, Y., Hanafusa, H.: Inverse kinematic solutions with singularity robustness for robot manipulator control. ASME J. Dyn. Syst. Meas. Control 108(3), 163–171 (1986)
13. Walker, M.W.: Adaptive control of manipulators containing closed kinematic loops. IEEE Trans. Rob. Autom. 6(1), 10–19 (1990)
14. Kheirikhah, M.M., Rabiee, S., Edalat, M.E.: A review of shape memory alloy actuators in robotics. In: Robot Soccer World Cup. pp. 206–217, Springer, New York (2010)
15. Liu, J.N., Abdel-Malek, K.: Robust control of planar dual-arm cooperative manipulators. Rob. Comput. Integr. Manuf. 16(2–3), 109–119 (2000)
16. Manjunath, T.: Kinematic modelling and maneuvring of a 5-axes articulated robot arm. Int. J. Mech. Mechatron. Eng. 28, 364–370 (2007)
17. Serrezuela, R.R., et al.: Kinematic modelling of a robotic arm manipulator using MATLAB. ARPN J. Eng. Appl. Sci. 12(7), 2037–2045 (2017)
18. Cheah, C.C., Liu, C., Slotine, J.J.E.: Adaptive jacobian tracking control of robots with uncertainties in kinematic, dynamic and actuator models. IEEE Trans. Autom. Control 51(6), 1024–1029 (2006)
19. Lee, C.-Y., Jeong, K., Lee, I.-h.: Motion control of mobile manipulator based on neural networks and error compensation. Proc. - IEEE Int. Conf. Rob. Autom. 5(5), 4627–4632 (2004)
20. Kung, Y.-S., Shu, G.-S.: Development of a FPGA-based motion control IC for robot arm. In: IEEE International Conference on Industrial Technology. Hong Kong, China, pp. 1397–1402 (2006)
21. Tunwannarux, A., Tunwannarux, S.: Design of a 5-joint mechanical arm with user friendly control program. Int. J. Mech. Aerosp. Ind. Mechatron. Manuf. Eng. 2(1), 75–80 (2008)
22. Huang, G.-S., et al.: Inverse kinematics analysis trajectory planning for a robot arm. In: 8th Asian Control Conference (ASCC). Kaohsiung, Taiwan (2011)
23. Szabo, R., Gontean, A.: Remotely commanding the lynxmotion AL5 type robotic arms. In: 21st Telecommunications Forum (TELFOR). Belgrade, Serbia (2013)
24. Su, H.-J., Dorozhkin, D.V., Vance, J.M.: A screw theory approach for the conceptual design of flexible joints for compliant mechanisms. J. Mech. Rob. 1(4) (2009)
25. Ajwad, S.A., et al.: Modelling robotic arms– a review and derivation of screw theory based kinematics. In: International Conference on Engineering & Emerging Technologies. Lahore, pp. 66–69 (2014)
26. Siciliano, B., Khatib, O.: Springer Handbook of Robotics. Springer, Switzerland (2016)
27. Salem, F.A., Rashed, A.A.: PID controllers and algorithms: selection and design techniques applied in mechatronics systems design - Part II. Int. J. Eng. Sci. 2(5) 191–203 (2013)
28. Tolani, D., Goswami, A., Badler, N.I.: Real-time inverse kinematics techniques for anthropomorphic limbs. Graphical Models 62(5), 353–388 (2000)
29. Cai, B., Zhang, Y.: Different-level redundancy-resolution and its equivalent relationship analysis for robot manipulators using gradient-descent and zhang's neural-dynamic methods. IEEE Trans. Ind. Electron. 59(8), 3146–3155 (2011)
30. Tarn, T., Bejczy, A., Yun, X.: Design of dynamic control of two cooperating robot arms: closed chain formulation. In: Proceedings. 1987 IEEE International Conference on Robotics and Automation. Raleigh, NC, pp. 7–13 (1987)
31. Mathur, N., Glesk, I., Buis, A.: Comparison of adaptive neuro-fuzzy inference system (ANFIS) and Gaussian processes for machine learning (GPML) algorithms for the prediction of skin temperature in lower limb prostheses. Med. Eng. Phys. 38(10), 1083–1089 (2016)

32. Strang, G., Introduction to Linear Algebra: Fifth Edition. Wellesley-Cambridge Press, Wellesley, MA (2016) http://math.mit.edu/~gs/linearalgebra/
33. Mukhtar, M.: Ambidextrous robot arm follow a circular path in Y-Z plane. (2020). https://youtu.be/Zs0Hu-36v1U. accessed: 2020-08-11
34. Mukhtar, M.: Ambidextrous robot arm follow a circular path in X-Y-Z. (2020). https://youtu.be/cylZKJxEQmo. accessed: 2020-07-14
35. Mukhtar, M.: Ambidextrous robot arm follow an Arc path in X-Y-Z. (2020). https://youtu.be/LCLuUSQlPRw. accessed: 2020-09-11
36. Mukhtar, M.: Ambidextrous robot arm follow a combined curves path in X-Y-Z. (2020). https://youtu.be/J-TC2Tavzb0. accessed: 2020-08-15
37. Mohammed, A., Schmidt, B., Wang, L., Gao, L.: Minimizing energy consumption for robot arm movement. Procedia CIRP 25(25), 400–405 (2014)
38. Kroger, T.: Online trajectory generation: Straight-line trajectories. IEEE Trans. Rob. 27(5), 1010–1016 (2011)
39. Falkenhahn, V., et al.: Dynamic modelling of constant curvature continuum robots using the EulerLagrange formalism. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. Chicago, IL, pp. 2428–2433 (2014)

How to cite this article: Mukhtar M, Khudher D, Kalganova T. A control structure for ambidextrous robot arm based on Multiple Adaptive Neuro-Fuzzy Inference System. IET Control Theory Appl. 2021;15:1518–1532. https://doi.org/10.1049/cth2.12140