

Consistent Control Framework for Ambidextrous Robot Arm Using MANFIS Controller

Mashood Mukhtar¹ Dhayaa Khudher² and Tatiana Kalganova³

¹ & ³Electronic and Computer Engineering, Brunel University London,
Kingston Lane, UB8 3PH, London, UK

Mashood.Mukhtar@brunel.ac.uk

Tatiana.Kalganova@brunel.ac.uk

²Computer Engineering, Basra University, Basra, Iraq

Basra University, Basra, Iraq

Dhayaa.Khudher@uobasrah.edu.iq

Abstract - This paper presents a control strategy to control an Ambidextrous robot arm. The idea is based on combining of Proportional Integral Derivative (PID) with Neural Network (NN) and Multiple Adaptive Neuro-fuzzy Inference (MANFIS) controller with selector block. This unique set of combination is used to solve the problem of inverse kinematics and to get a stable dynamic response in different situations. The results obtained from the experiments proved the effectiveness of the controllers in term of performing multiple tasks and dynamic stability.

Keywords: Dynamic Control, Task and Motion Planning, Kinematic Analysis, ANFIS Controller, 3D Printed

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 robotics, industrial robots and biomedical robotics [3], [4].

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 [5] 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 [6] 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 [7],

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 [8]. 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 [9], [10]. 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 [11].

A neural network controller for a mobile manipulator to track the given trajectories is introduced in [12]. 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.

From the survey of robot arm control, it is apparent that none of the researchers employs two complete sets of separate actuators in such a setting as to control the whole system and use of MANFIS controller with selector block to drive ambidextrous arm.

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 effectiveness of the robot arm by integrating the robot design and the robot controller. It is expected that authors will submit carefully written and proofread material. Careful checking for spelling and grammatical errors should be performed. The number of pages of the paper should be from 4 to 8.

2. Design of the Ambidextrous Robot Arm

The human arm has always been considered the perfect model to start designing robotic arms. An arm is mainly divided into two parts; upper limb that connects shoulder joint and the elbow joint and forearm refers to a part between hand and elbow. The most common feature of the arm is to act as an extender for the hand and help in effective completion of various tasks. Human arm offers a total of seven degrees-of- freedom; three in the shoulder (Pitch, Yaw and Roll), two in the elbow (Pitch and Roll) and two in wrists.

All 3D printed parts were assembled to make the ambidextrous robotic arm. Fig. 1 shows the assembled version of the ambidextrous robot arm. The assembly process took place in the SEP lab at Brunel University London.

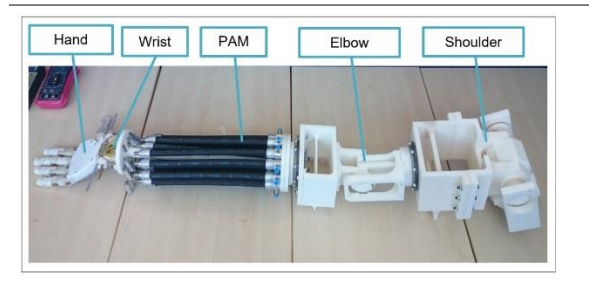


Fig. 1: The assembled version of the ambidextrous robot arm

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.

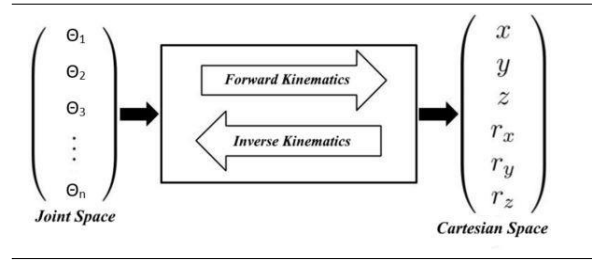


Fig. 2: Forward and inverse kinematics relationship.

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 six 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 Fig. 2.

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 [13]. 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 (dx, dy, dz) will be formulated with respect to arm base frame as Equation (1):

$$x(dx, dy, dz) = f(\theta) \quad (1)$$

By differentiating both sides of Equation (1) with respect to θ , the velocity in the task space of the hand will result as Equation (2)

$$\dot{x} = J\dot{\theta} \quad (2)$$

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 (2) should be inverted to get a linear form as Equation (3).

$$\dot{\theta} = J^{-1}\dot{x} \quad (3)$$

It is clear from Equation (3) 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 [14]. One of the most commonly used methods is an ANFIS.

4. Adaptive Network Fuzzy Inference System (ANFIS)

The basic rule of AN is based on gradient descent [15] and the chain rule [16]. ANFIS utilizes network topology to reduce the optimization 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 a NN that is based on Takagi-Sugeno fuzzy inference 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 nonlinear functions. Fuzzy if-then rules express conditions IF A THEN B, where A and B are fuzzy set labels characterized 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 parameterized 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 inference system.

ANFIS is multi-layered feed forward network in which each node performs a particular node function as shown in Fig. 3. 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:

$$\begin{aligned} \text{Rule1 : } & \text{IF } (x \text{ is } A_1) \text{ and } (y \text{ is } B_1) \\ & \text{then } (f_1 = P_1x + Q_1y + R_1) \end{aligned}$$

$$\begin{aligned} \text{Rule2 : } & \text{IF } (x \text{ is } A_2) \text{ and } (y \text{ is } B_2) \\ & \text{then } (f_2 = P_2x + Q_2y + R_2) \end{aligned}$$

Where x and y are the inputs, Ai and Bi are the fuzzy sets, fi are the outputs within the fuzzy region specified by fuzzy rules: Pi, Qi and Ri are the design parameters that are determined during the training process.

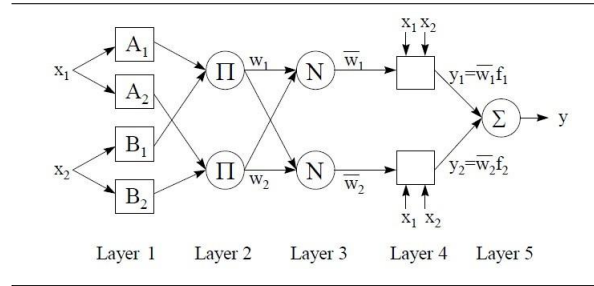


Fig. 3: ANFIS architecture [17].

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.

$$O_i^1 = \mu A_i(x) \quad i = 1, 2 \quad (4)$$

$$O_i^1 = \mu B_i(y) \quad i = 3, 4 \quad (5)$$

Where $\mu A_i(x)$ and $\mu B_i(y)$ can adopt any fuzzy MF.

Layer 2: Every node is a circle node in layer 2.

$$O_i^2 = \mu A_i(x) \mu B_i(y) \quad (6)$$

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.

$$O_i^3 = \bar{w}_i f_i = \frac{w_i}{w_1 + w_2} \quad (7)$$

Where $i = 1: 2$ the i th node calculates the ratio of 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 product of the normalized firing strength and a first-order polynomial.

$$O_i^4 = \overline{w}_i f_i = w_i (P_{ix} + Q_{iy} + R_i) \quad (8)$$

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.

$$O_i^5 = \sum_{i=1}^2 \overline{w}_i f_i = \frac{\sum_{i=1}^2 \overline{w}_i f_i}{w_1 + w_2} \quad (9)$$

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 [18]. On the other hand in backward pass, error signal propagates backwards, S2 is unmodified, and S1 is computed using gradient descent algorithm [15].

The process of using ANFIS technique involves data generation (hypothesise 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 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.

For simplicity, the inverse kinematic problem is further divided into the end-effector orientation problem and end effector position problem. It is apparent from Fig. 4 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.

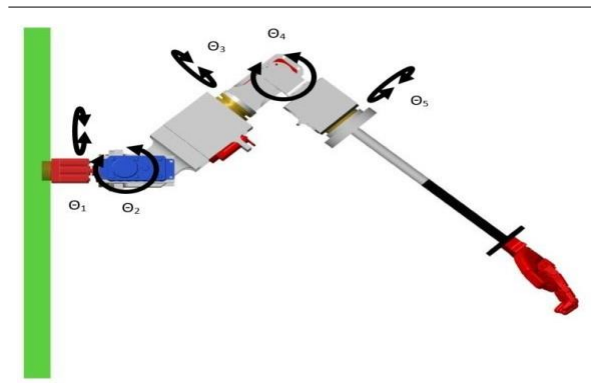


Fig. 4: Kinematic configuration and joint model of the five- joint ambidextrous robotic arm.

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. Multiple ANFIS also known as (MANFIS) is modelled in Simulink software as shown in Fig. 5. The MANFIS maps the input in task the joint angles in joint space, and joint angles are used to determine the desired trajectory. Fig. 5 shows a Simulink diagram for the controller. The controller contains five ANFIS with six inputs (X, Y, Z, Rx, Ry and Rz) and five ($\theta_1, \theta_2, \theta_3, \theta_4$ and θ_5).

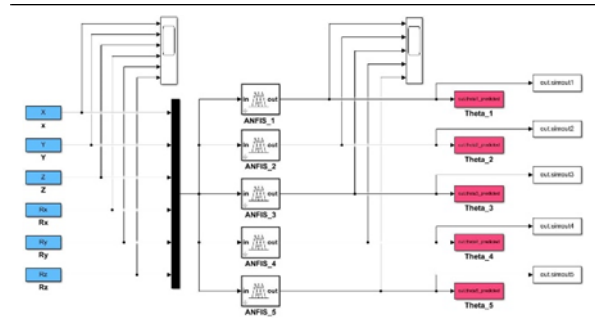


Fig. 5: MANFIS modelled in Simulink software.

5. Experiments

In order to demonstrate the effectiveness of the controller, two experiment have been accomplished. The first experiment is carried out to demonstrate the ability of the robot to transit between multiple tasks. The second task is issued to focus on the dynamic stability while performing tasks in different situation.

5.1 Smooth Transition between Tasks

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. Fig. 6 shows four views for 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.

Fig. 7 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 0.2 cm for very short time as shown in Figures 8, 9 and 10. A short video for this experiment is available in [19].

Fig. 11 represents the produced joints angles for the ambidextrous robot arm. The joints transitions are very smooth. Further, it is 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 represent the space where the two curves have been combined.

5.2 Dynamic Evaluation for the Robot

The following experiments are performed to evaluate the exerted torque and the acceleration at the end effector of the robot. The experiment is divided in two parts. In the first part, the robot will follow a specified semi-circular path carrying no load. A load of 0.5kg will be carried by the robot hand in the second part. The path is generated using three points in xyz plane. The intermediate points have been interpolated using quintic polynomial to get continuous velocity and acceleration [20].

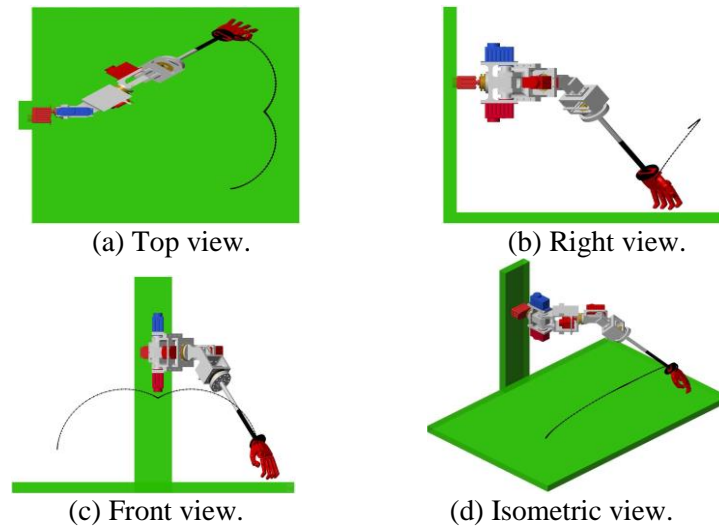


Fig. 6: The desired paths in the task space.

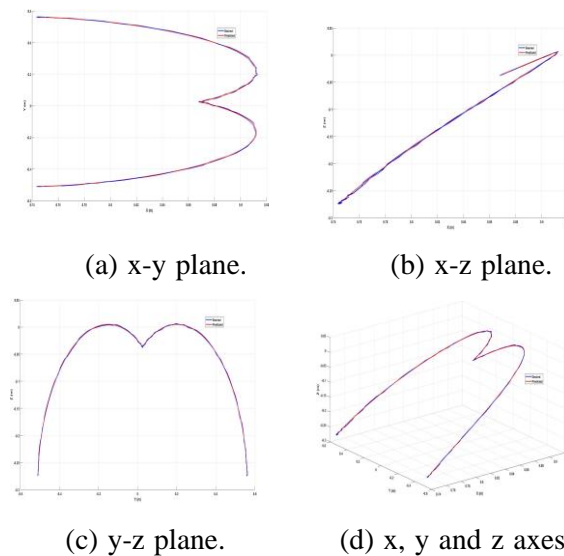


Fig. 7: The desired paths in the task space for the combined curves path.

Figures 12 - 16 depict the torque exerted by the Ambidextrous arm. It's clear from this experiment that the exerted torque at only joint one and two in the Ambidextrous arm have been changed when carrying a load. Whereas the exerted torque at the other joints of the robot remained the same. 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.

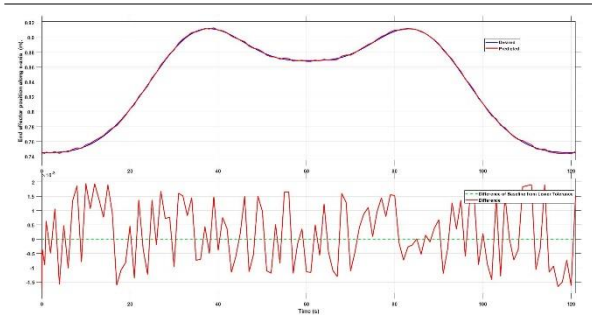


Fig. 8: The hand position along x-axis for the combinedcurves path.

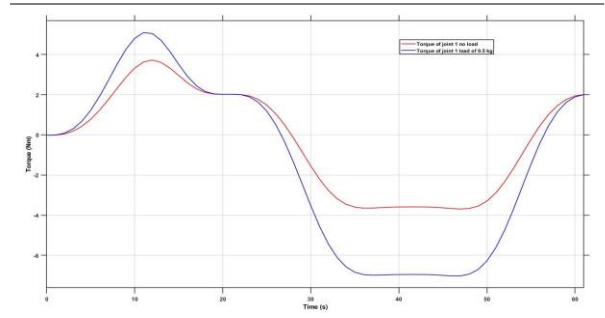


Fig. 12: The torque exerted by the Ambidextrous arm in jointone.

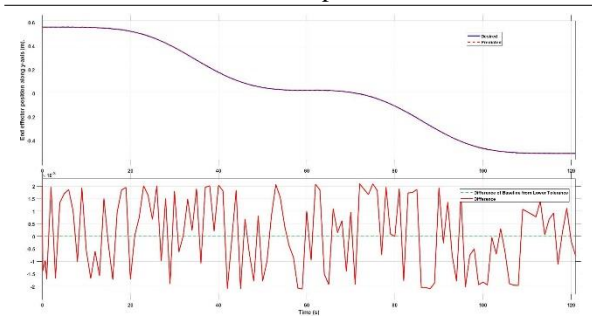


Fig. 9: The hand position along y-axis for the combinedcurves path.

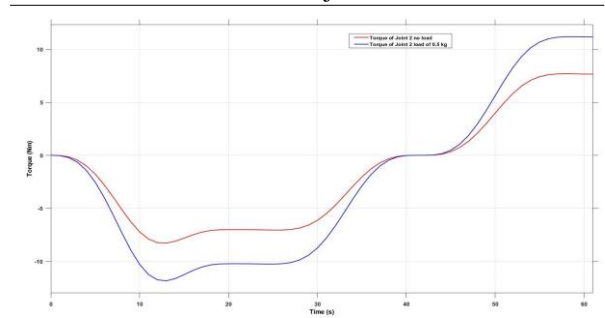


Fig. 13: The torque exerted by the Ambidextrous arm in jointtwo.

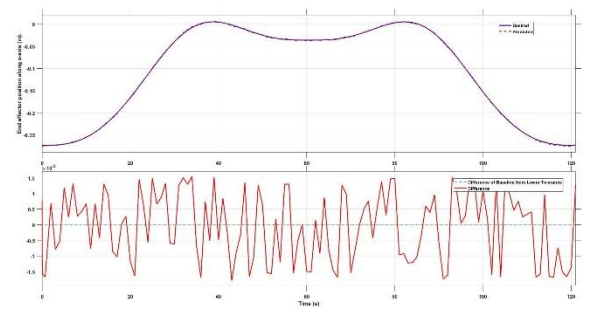


Fig. 10: The hand position along z-axis for the combinedcurves path.

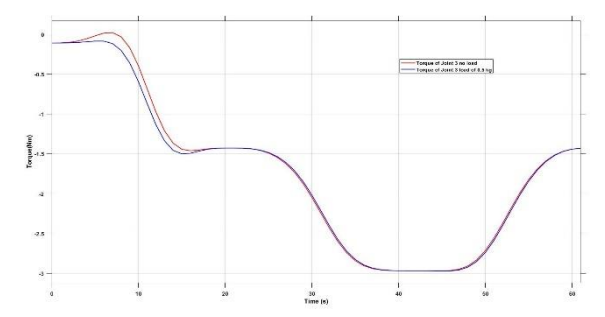


Fig. 14: The torque exerted by the Ambidextrous arm in jointthree.

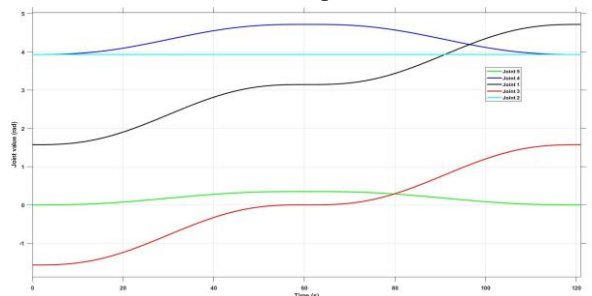


Fig. 11: The resulted joint values to produce the jointfour.

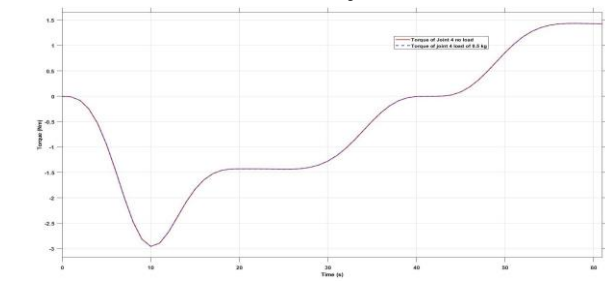


Fig. 15: The torque exerted by the Ambidextrous arm in Combined path at the robot hand.

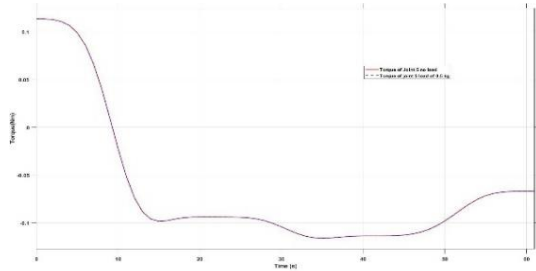


Fig. 16: The torque exerted by the Ambidextrous arm in joint five.

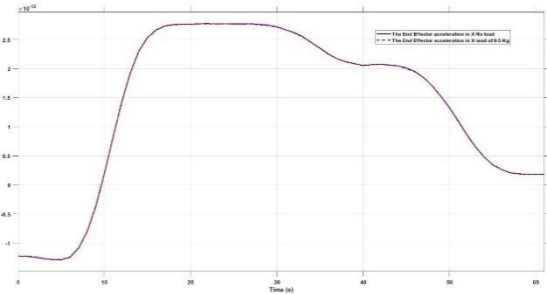


Fig. 17: The acceleration of the EE in X-axis (Ambidextrous)

The acceleration of the ambidextrous arm in X-axis and Y-axis shown in Fig. 17 and Fig. 18 respectively.

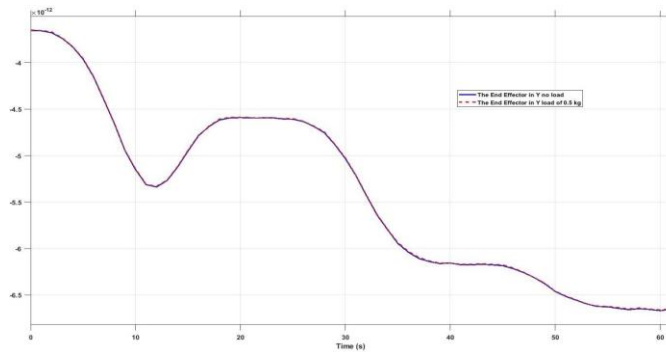


Fig. 18: The acceleration of the EE in Y-axis (Ambidextrousarm).

6. Conclusion

The paper relates to the design and control of a novel robot arm named ambidextrous robot arm. An integration between the control system and the robot design has been achieved in order to get a consistent control framework. In the design part, two types of actuators have been used to drive the robot joints. Whereas, in the control part, two MANFIS controller are used to get rid of the singularity problem that arisen during the inverse kinematics. The performance of the robot has been evaluated while accomplishing two tasks. The resulted joints angle were smooth. Further, both the exerted torque at each joint and the acceleration at the end effector have been tested. According to the obtained results, the effectiveness of the controller has been proved.

References

- [1] D. Gama Melo, Erika Nathalia and Aviles Sanchez, Oscar Fernando and Amaya Hurtado, "Anthropomorphic robotic hands: a review," *Ingeniería y Desarrollo*, vol. 32, no. 2, pp. 279–313, 2014.
- [2] Y. Tsuneo, "Multifingered robot hands: Control for grasping and manipulation," *Annual Reviews in Control*, vol. 34, pp. 199–208, 2010.
- [3] B. Antonio and K. Vijay, "Robotic Grasping and Manipulation: Chapter 1, pp. 1-22," 2000. [Online]. Available: <http://www.centropiaggio.unipi.it/sites/default/files/surveys-ramsete01.pdf>
- [4] S.-W. Park, J.-H. Bae, J.-H. Park, and M.-H. Baeg, "Development of an anthropomorphic robot hand aimed at practical use for wide service robot application," in *IEEE International Conference on Automation Science and Engineering (CASE)*, Seoul, South Korea, 2012.
- [5] Y. Nakamura and H. Hanafusa, "Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control," *ASME. J. Dyn. Sys., Meas., Control*, vol. 108, no. 3, pp. 163–171, 1986.
- [6] M. W. Walker, "Adaptive Control of Manipulators Containing Closed Kinematic Loops," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 1, 1990.
- [7] M. M. Kheirikhah, S. Rabiee, and M. E. Edalat, "A review of shape memory alloy actuators in robotics," in *Robot Soccer World Cup*. Springer, 2010, pp. 206–217.
- [8] J. N. Liu and K. Abdel-Malek., "Robust Control of Planar Dual- Arm Cooperative Manipulators," *Robotics and Computer-Integrated Manufacturing*, vol. 16, no. 2-3, pp. 109–119, 2000.
- [9] T. Manjunath, "Kinematic modelling and maneuvering of a 5-axes articulated robot arm," *world academy of science, engineering and technology*, vol. 28, pp. 364–370, 2007.
- [10] R. R. Serrezuela, A. F. C. Chavarro, M. A. T. Cardozo, A. L. Toquica, and L. F. O. Martinez, "Kinematic modelling of a robotic arm manipulator using matlab," *ARNP Journal of Engineering and Applied Sciences*, vol. 12, no. 7, pp. 2037–2045, 2017.
- [11] C. C. Cheah, C. Liu, and J. J. E. Slotine, "Adaptive Jacobian Tracking Control of Robots With Uncertainties in Kinematic, Dynamic and Actuator Models," *IEEE Transactions on Automatic Control*, vol. 51, no. 6, 2006.
- [12] C.-Y. Lee, K. Jeong, and I.-h. Lee, "Motion control of mobile manipulator based on neural networks and error compensation," *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 5, no. 5, pp. 4627–4632, 2004.
- [13] F. A. Salem and A. A. Rashed, "PID Controllers and Algorithms: Selection and Design Techniques Applied in Mechatronics Systems Design - Part II," in *International Journal of Engineering Sciences*, 2013, pp. 191–203.
- [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] B. Cai and Y. Zhang, "Different-level redundancy-resolution and its equivalent relationship analysis for robot manipulators using gradient-descent and zhang's neural-dynamic methods," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3146–3155, 2011.
- [16] T. Tarn, A. Bejczy, and X. Yun, "Design of dynamic control of two cooperating robot arms: Closed chain formulation," in *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, vol. 4. IEEE, 1987, pp. 7–13.
- [17] A. Mathur, Neha and Glesk, Ivan and Buis, "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," *Medical engineering & physics*, vol. 38, no. 10, pp. 1083–1089, 2016.
- [18] G. Strang, *Introduction to linear algebra*. Wellesley-Cambridge Press Wellesley, MA, 1993.
- [19] M. Mukhtar, "Ambidextrous robot arm follow a combined curves path in X-Y-Z." 2020. [Online]. Available: <https://youtu.be/J-TC2Tavzb0>
- [20] T. Kroger, "Online trajectory generation: Straight-line trajectories," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 1010–1016, 2011.