

Model Reference Controller Approach for Robot Arm Tracking Using Neural Networks

N. Najva* and Abdul Saleem

Department of Electrical and Electronics Engineering, Government Engineering College, Thrissur, Kerala 680009, India; najvahassan@gmail.com, abdulsaleempk@gmail.com

Abstract

Objectives: This study proposes a neural network (NN)-based model reference controller (MRC) for robot arm trajectory tracking. **Methods/statistical analysis:** The proposed methodology uses two NNs: a reference model and a controller. The NN-based reference model is initially trained such that it follows any desired reference trajectory. The position of the robot arm is controlled by changing the joint angles, which is achieved by applying the desired torque. The NN controller provides the desired torque, and the controller is trained until the error between the outputs of the actual plant and the reference model is driven to a value which is approximately zero. The trained NN controller is employed for the actual trajectory tracking. **Findings:** Our NN-based reference model is capable of approximating the nonlinear model of the robot arm motion, and it is expected to minimise the effect of model uncertainties. Simulations are done to validate the proposed method, which found that the NN-based MRC is capable of following the desired trajectories with approximately zero tracking error. **Application/improvements:** The proposed controller effectively tracks any desired trajectory with least tracking error and minimum control input. Simulation results illustrate that the total control effort and maximum control input required to track the desired trajectory are very less compared to that required for a PID controller.

Keywords: Model Reference Controller, Robot Arm, Tracking, Neural Networks, Tracking Error, Control Input

1. Introduction

Robot manipulators are electromechanical devices which are capable of performing a variety of functions in a more flexible working environment and at a lower production cost. The various applications include industrial, educational and medical fields apart from their applications in farm, home, hospital and military. The robots can work in unpredictable and hazardous situations where a human might find very difficult to reach. One of the objectives of the robot is its arm movement from an initial position to a target position. Planning robot arm trajectory that satisfies obstacle avoidance and developing an appropriate method to control trajectory tracking are important research problems in robotics. An effective trajectory tracking controller should be able to track the desired trajectory with minimum tracking error, lesser control input and minimum total control effort.

The objective of robot arm tracking is to develop a controller such that the robot arm can move along the desired trajectory. This can be achieved by controlling the joint angles of the arm. The joint angle for a particular end effector orientation and position is calculated by inverse kinematics. Since the robot arm dynamics are highly nonlinear, a closed-form solution of joint angles may not be feasible. Neural networks (NNs) are capable of handling uncertainties and unexpected parameter variations in nonlinear plants, and so NN-based controllers can be considered as a better solution to robot arm trajectory control problems.

In literature, numerous methods are suggested for effective trajectory tracking of robot arm. Bingol et al.¹ proposed a pattern search algorithm-based PID controller for the trajectory control of robot arm. The pattern search algorithm uses a set of vectors called patterns to

*Author for correspondence

determine which points to search at each iteration; and for each iteration, this algorithm searches a point, from the pattern around the current point, such that the value of the objective function is improved. The optimisation algorithm presented in this paper requires more time to complete trajectory tracking. Youns et al.² proposed a PID controller whose control parameters are tuned using a genetic algorithm. This method controls the robot arm with less tracking error compared with a conventional PID controller. An adaptive computed torque control algorithm is presented by ref.³ for trajectory tracking. The adaptive control is capable of compensating the parametric uncertainties, but is limited to robots with 2-DOF. Linear estimation techniques together with a computed torque control law are used in Guo and Zhang.³ Tracking performance was improved by Lin and Chen⁴ who developed a model reference adaptive control which considered variations at end effector load. Dynamic uncertainties are considered in the work proposed by Fan and Joo.⁵ A fractional order fuzzy-based PID controller, where parameter tuning is done using fuzzy logic, was developed by Mohammed et al.⁶ Sinha and Mishra⁷ developed a controller based on sliding mode, which provides robustness against uncertainties to control the robot arm. The controller guarantees stability in Lyapunov sense, but the chattering effect could not be eliminated completely. The chattering effect was reduced by Boudoua et al.⁸ by using a twisting sliding mode controller for the control of pneumatic artificial muscles in robot arm. However, this method does not guarantee the stability of the overall system. Parlakçi et al.⁹ proposed a robust PD-sliding mode controller for avoiding the instability problem. Even though this method avoids the instability problem, it produces tracking error. For under-actuated robots, a model-based adaptive variable structure control scheme was proposed by Su and Stepanenko.¹⁰ A simple sliding mode control scheme for robot arm was presented used by Bailey and Arapostathis.¹¹ Design complexity is reduced greatly here. A trajectory controller which uses the variable structure theory was proposed in Yeung and Chen.¹² This method considered parameter variations, and the controller can be used for a larger number of links.

Due to the uncertainties in the dynamic model and possible variations in system parameters, an adaptive controller is highly appreciated. The effects of modelling uncertainties decrease the performance of the controller in terms of tracking accuracy and control input.

Conventional methods fail to model the nonlinearities and uncertainties in robot arm dynamics. An effective trajectory tracking controller should possess the least tracking error and low control input. Also, its maximum control input requirement and total control effort should be minimised.

This study proposes a model reference NN-based trajectory controller which gives the best performance in terms of tracking error, control input, total control effort and maximum torque compared to conventional PID controllers. Herein, two NN architectures are used: one as the controller and the other as the reference model. The reference model is trained such that it can follow any reference trajectory. The controller is trained using a large number of randomly generated reference trajectories, and the weights are updated to minimise error between reference trajectory and the output of NN-based plant model. After training, the NN structure is connected to the robot arm.

The rest of this paper is organised as follows. Section 2 formulates the tracking problem. A brief introduction of NNs is presented in section 3. The proposed methodology is described in section 4. Simulation results are presented and analysed in section 5. Finally, concluding remarks are presented in section 6.

2. Problem Formulation

A robot arm with 2-DOF consists of two links as shown in Figure 1. L_1 and L_2 are link lengths of arm 1 and arm 2, and their joint angles are θ_1 and θ_2 , respectively. The two links are connected together to a revolute joint known as elbow, and the first link is fixed to a revolute joint called shoulder.

The relation between position and orientation of a robot arm is obtained from its forward kinematics which can be derived using Denavit-Hartenberg (DH) convention as:

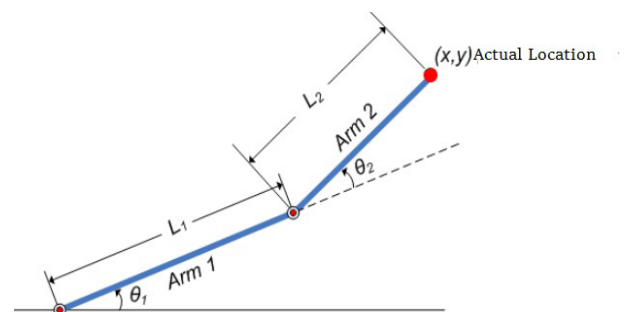


Figure 1. Robot arm.

$$x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) \quad (1)$$

$$y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) \quad (2)$$

where θ_1 and θ_2 are joint angles.

Equations (1) and (2) give the position of robot arm at any instant. These two equations imply that the position of the arm can be controlled by changing the angle θ_1 or θ_2 which can be achieved by applying torque. Thus, the torque acts as the control input of robot arm tracking. The robot dynamics is represented by the equation

$$\frac{d^2 \phi}{dt^2} + g \sin \phi + \alpha \frac{d\phi}{dt} = u \quad (3)$$

where u is the control signal, ϕ is the angle to be controller and a, g are constants.

The dynamic equation considers frictional and gravitational forces. The objective of robot arm trajectory tracking is to design a control input u such that the robot arm moves along the desired trajectory as close as possible such that total control effort is minimised. In this work, artificial NN (ANN) is used as the controller to solve the trajectory tracking problem.

3. Neural Networks

ANN is a mathematical model that mimics the structure and function of biological NNs. The basic building block of an ANN is neurons. A neuron receives the weighted sum of outputs of all neurons connected to it as its input, which is passed to an activation function to compute the neuron output. A multi-layer NN consists of an input

layer, one or more hidden layers and an output layer. The commonly used activation functions are sigmoid function, ReLu function hyperbolic tangent, leaky RELU, softmax and swish. ANNs have the ability to learn and model nonlinear and complex relationship and are capable of generalisation. An NN learns by updating its weights. Back propagation algorithm is one method used for NN training. In back propagation, the weights are adjusted such that a cost function is minimised. The sum of squares of errors is a proper choice of cost function. Here, error is the difference between the desired output and actual output. An NN structure is shown in Figure 2.

Net input by a neuron is given by

$$net_j = \sum_{i=1}^n w_{ij}x_j + b_j \quad (4)$$

where x is the output of the incoming neuron of the previous layer, w is the weight, b is the bias and n is the number of neurons in the previous layer. If a sigmoid function is used as the activation function, then the output of each neuron can be computed as

$$out_j = \frac{1}{1 + e^{-net_j}} \quad (5)$$

A typical cost function can be the sum of squares of errors defined as

$$E = \sum \frac{1}{2} (\text{target} - \text{output})^2 \quad (6)$$

The weights are adjusted such that the cost function is minimised. A common method used for updating the weights is to find how much changes occurs to the cost function for a given change in the weight, which is computed as

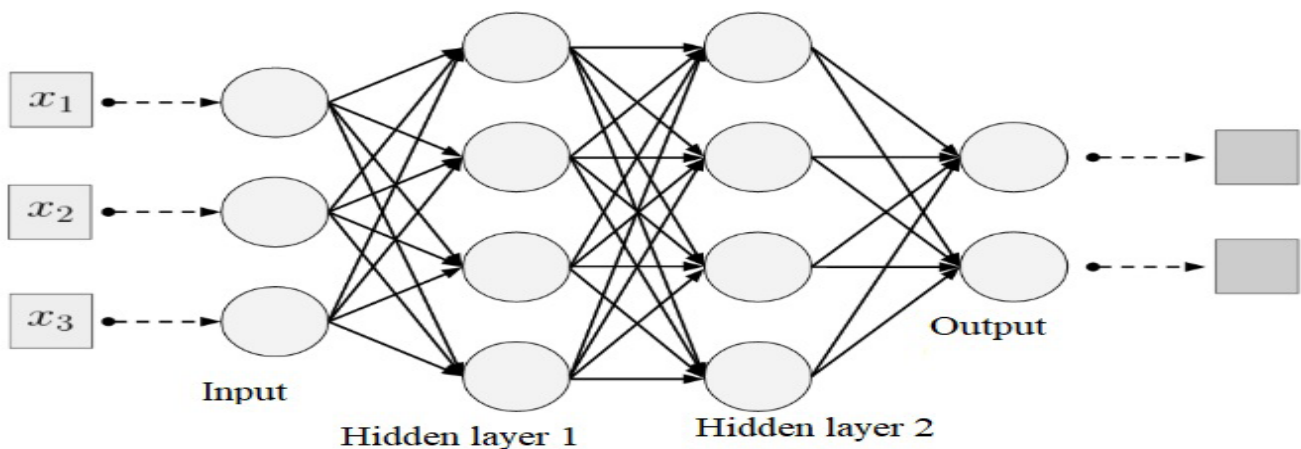


Figure 2. NN structure.

$$\frac{\partial E_{total}}{\partial w_i} = \frac{\partial E_{total}}{\partial out_{01}} * \frac{\partial E_{out_{01}}}{\partial net_{01}} * \frac{\partial net_{01}}{\partial w_i} \quad (7)$$

and the weight is updated using the formula

$$w_{n+1} = w_n + \alpha \frac{\partial E_{total}}{\partial w_i} \quad (8)$$

where α is the learning factor. Training is continued until the cost function is reduced to an acceptable value. The proposed method uses an NN-based controller to control robot arm trajectory.

4. Proposed Methodology

The proposed methodology uses the concept of model reference controller (MRC) presented in Youcef-Toumi and Ito.¹³ Herein two NNs, the reference plant and the controller, are used.

Figure 3 represents the NN structure of the reference plant. Inputs to the network are desired trajectory and output of the model. Initially, the reference model is trained until the error between output of the model and its input attains zero or a very small value. A trained plant reference model is capable of following any desired trajectory. Hence the output of this model can be used to train the controller.

The next step is to train the controller. Figure 4 shows the schematic structure during controller training. It consists of two subnetworks: the controller and the reference plant. Inputs to the controller are desired trajectory, controller output, and output of the plant reference model. Inputs

to the reference model are controller output and model output.

A number of randomly generated desired trajectories are given to the controller input for training, and the weights of the NN are adjusted until the difference between desired trajectory and reference plant is zero or very small. The advantage of this technique is that there is no need of an actual plant for training the network. Back propagation algorithm is used for training. Once the NNs are trained, then the controller can be connected to the robot arm, as shown in Figure 5. The desired trajectory and output of the robot arm are inputs to the MRC which consists of the two subnetworks of Figure 4. The controller output is connected to the plant.

5. Results and Discussion

The performance of the proposed methodology is tested by simulation experiments. SIMULINK setup is used for simulation. The robot arm is modelled using SIMULINK. Values of $a=2$ and $g=10$ are considered for the coefficients of the dynamic model given by equation (3). Three different trajectories are considered. First, the reference model is trained and then the controller is trained by giving a set of randomly selected desired trajectories as input. After training, error between the desired trajectory and the output of reference plant is zero. The three cases discussed here use this trained MRC. To illustrate the effectiveness of the proposed method, the results are compared with that obtained for a PID controller-based robot arm trajectory tracking.

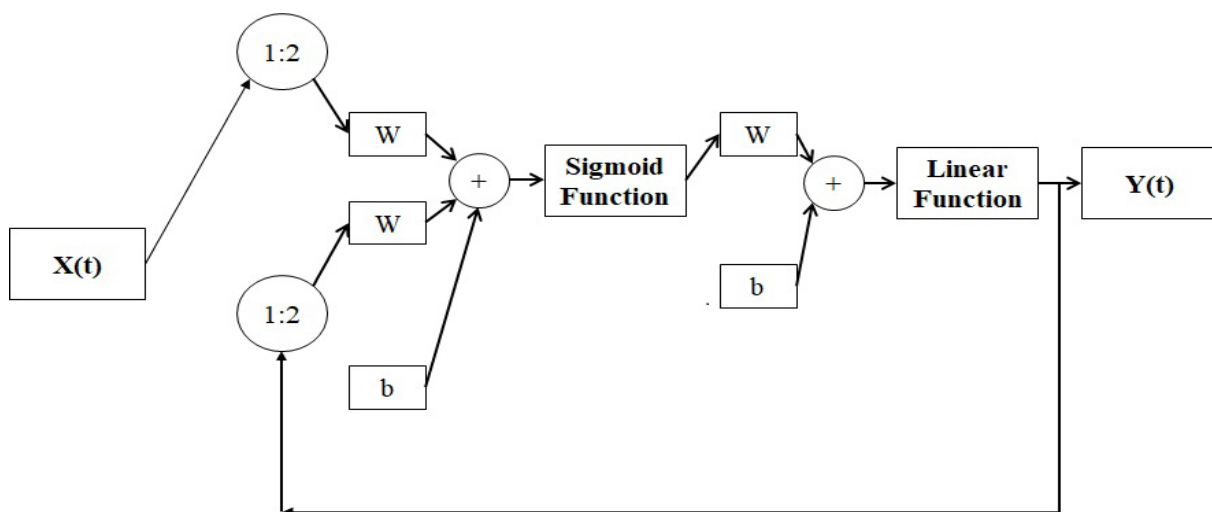


Figure 3. Plant reference model.

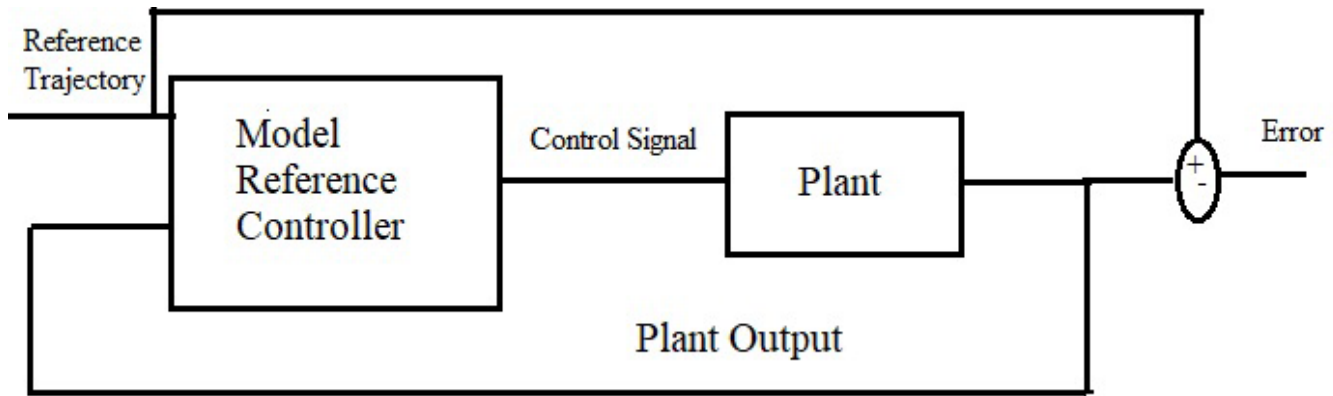


Figure 4. Plant reference model plus NN controller.

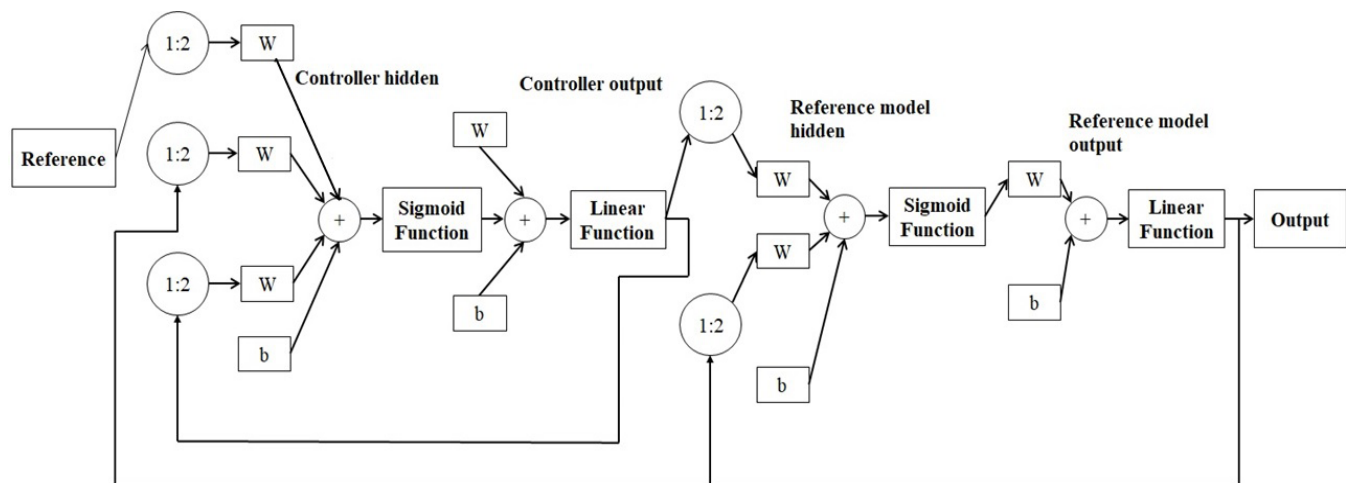


Figure 5. Simulation setup.

5.1. Case 1

A straight line given by equation

$$y = 0.02 + 0.02 * t \quad (9)$$

is considered as the reference trajectory. The objective is that the robot arm should track this trajectory with minimum tracking error and by applying minimum control. The NN controller is trained using the reference trajectory. Figure 6 shows the tracking of straight line trajectory by our NN-MRC.

It shows that the robot arm is capable of closely following the desired trajectory. However, the trajectory tracked by the PID controller is not so close to the desired trajectory which can be observed from Figure 7. For this case, the PID parameters are chosen as $K_p = 12$, $K_d = 8$, $K_i = 5$.

The error vs. time plot, shown in Figure 8, illustrates that tracking error of the NN controller is less than that of the PID controller. Tracking error of the NN controller

is near zero. The control signal u is plotted against time t , for both PID and NN controller, as shown in Figure 9. It is obvious from the figure that the PID controller requires more torque. Also, at all instants of time, torque requirement of the NN controller is less than that required for the PID controller.

Another achievement is that maximum torque requirement is also small for the NN controller. The total control effort for the PID controller is very large than that needed by the NN controller.

5.2 Case 2

In practical applications, the robot arm is hardly required to move in a sinusoidal path. For a theoretical interest and to illustrate the effectiveness of the NN controller, we considered a sinusoidal trajectory. Trajectory tracking plots with the NN controller and the PID controller are shown in Figures 10 and 11.

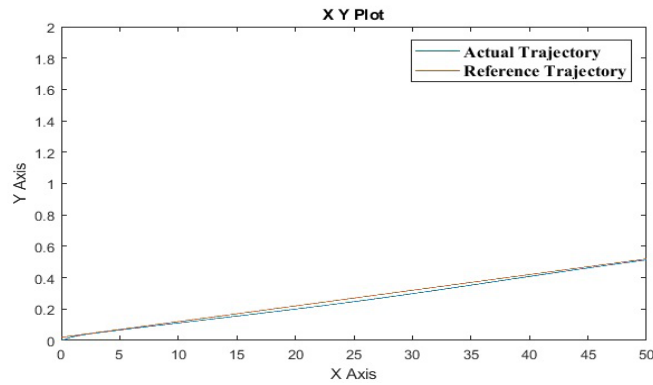


Figure 6. Trajectory tracking using an NN controller (case 1).

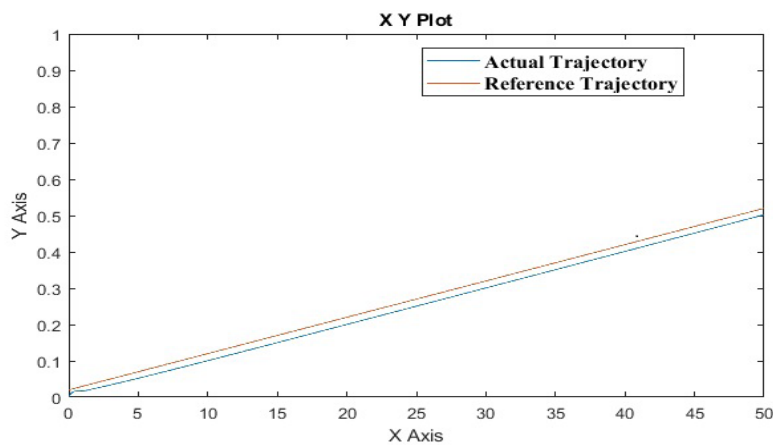


Figure 7. Trajectory tracking using a PID controller (case 1).

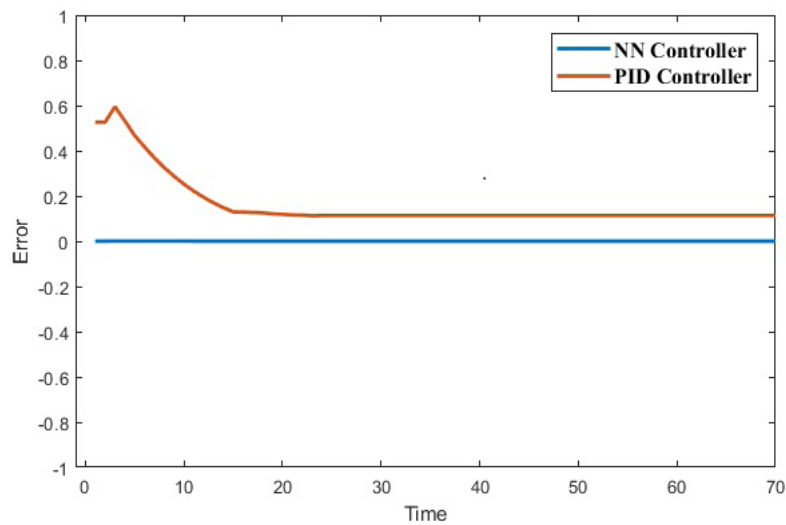


Figure 8. Time history of tracking errors: comparison of controllers (case 1).

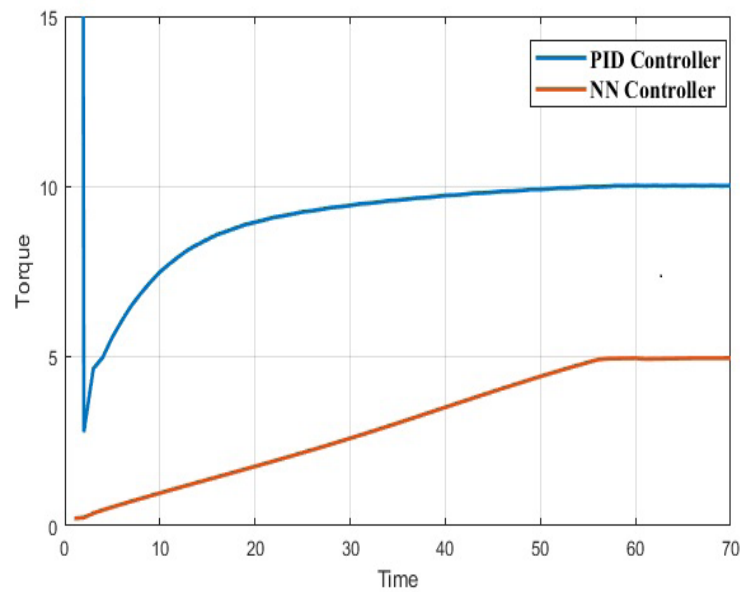


Figure 9. Time history of control input: comparison of controllers (case 1).

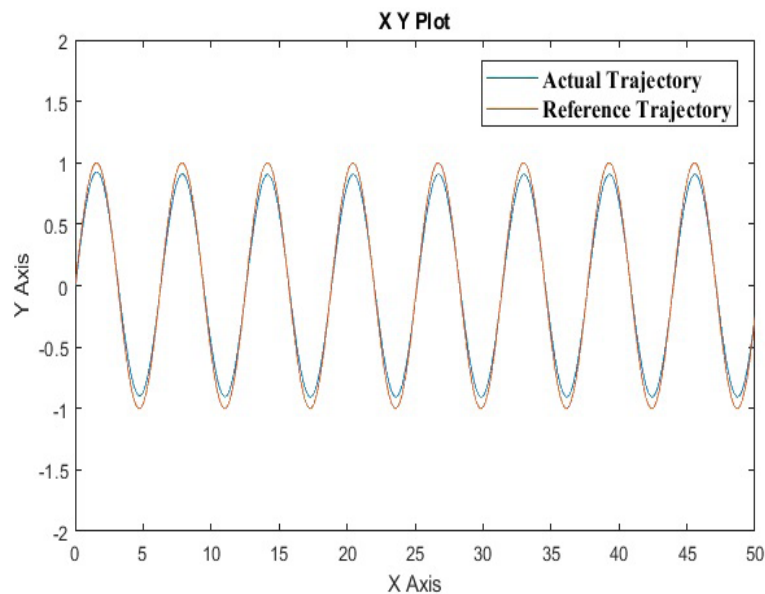


Figure 10. Trajectory tracking using an NN controller (case 2).

The plots clearly illustrate that the NN controller gives a much smaller tracking error compared to the PID controller. K_p , K_d , K_i parameters of the PID have the values of 25, 8 and 15, respectively.

5.3. Case 3

Consider a case where the objective of the robot arm is to pick an object from an initial position to a target

position along a reference trajectory as shown in Figure 12.

Simulations are done for both NN and PID controllers, and tracking trajectories are depicted in Figures 12 and 13. PID parameters are taken as $K_p = 53$, $K_d = 10$, $K_i = 25$.

Table 1 shows the total control effort required by the two controllers for the three cases. It clearly illustrates

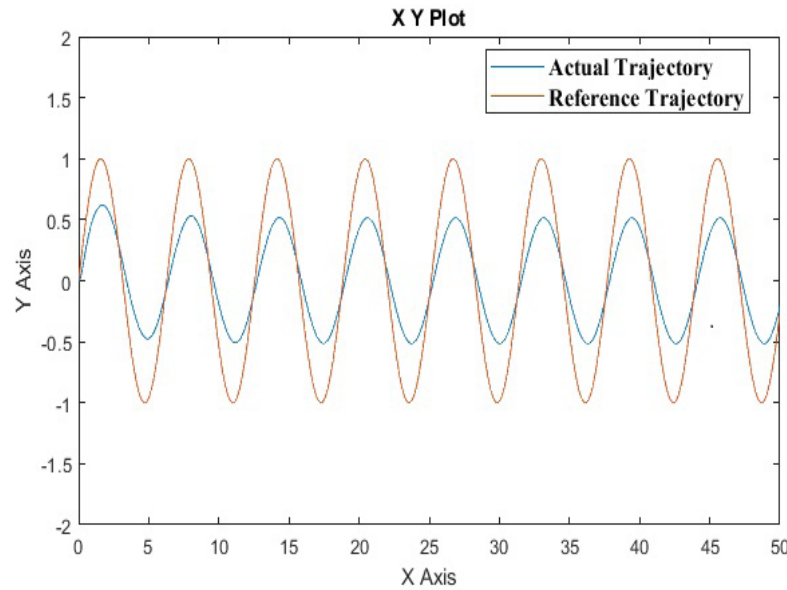


Figure 11. Trajectory tracking using a PID controller (case 2).

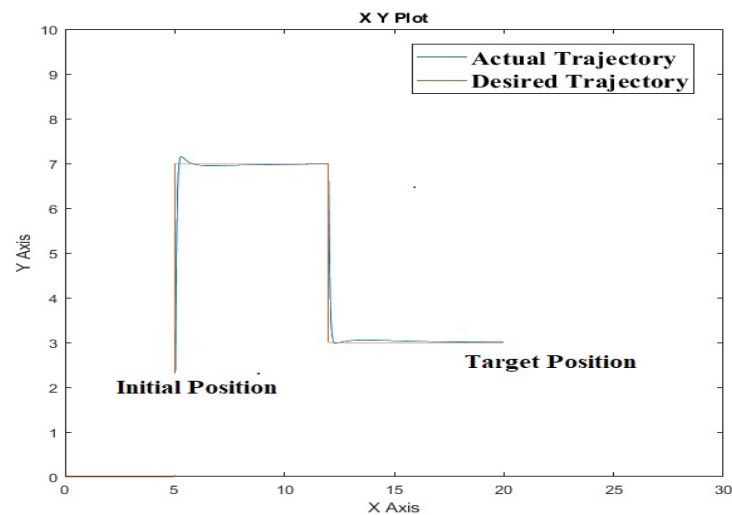


Figure 12. Trajectory tracking using an NN controller (case 3).

Table 1. Total control effort: comparison of two controllers

	Case 1	Case 2	Case 3
NN controller	584.76	4.19×10^4	3.09×10^4
PID controller	3575.67	5.88×10^4	5.13×10^4

that our NN-based controller requires lesser control effort compared to the PID controller.

Simulation results illustrate that our NN-MRC has more accuracy and requires less torque and minimum

control effort. The robot dynamics is nonlinear and behaves in an unpredictable way. The adaptive property of NNs makes it very suitable for approximating any nonlinear dynamics and thus nullifying the effect of model uncertainties. Also, the controller is trained to adapt to all possible reference trajectories. Hence, our model reference NN controller is capable of tracking any desired trajectory with least tracking error. In addition, the NN controller minimises control input to the plant. The total control effort required is also less with the NN controller.

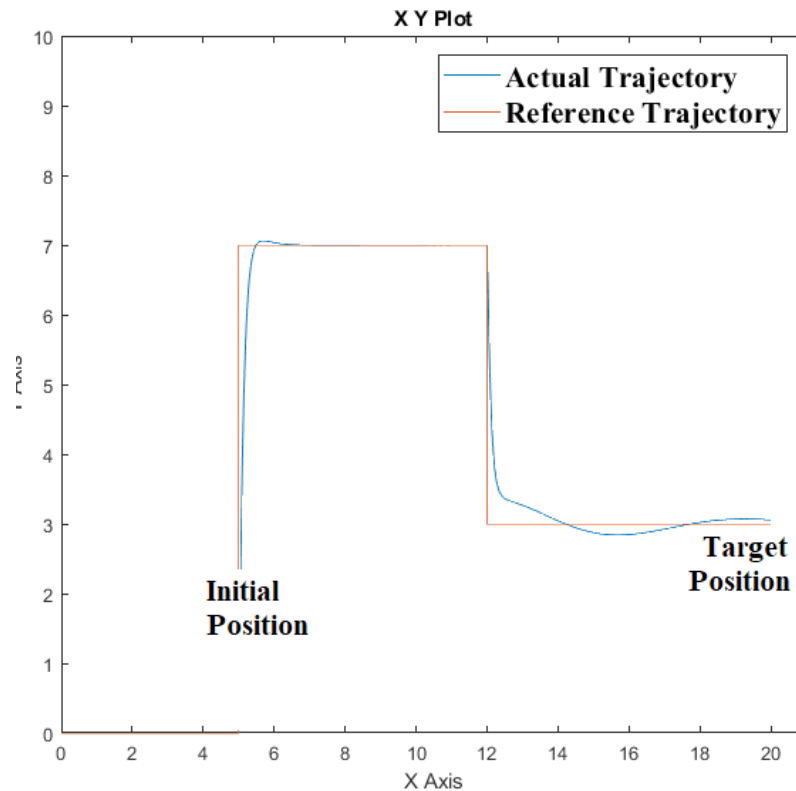


Figure 13. Trajectory tracking using a PID controller (case 3).

6. Conclusion

This study presented an NN-based MRC for trajectory control of robot arm. The proposed NN-based controller consists of two NNs: the plant model and the controller. First, the reference model is trained such that it can follow any reference trajectory, and then the controller and reference model is trained for a number of different random reference trajectories such that tracking error is near zero. The trained NNs are connected to the plant. Simulations are done for three different cases for both the proposed NN controller and a conventional PID controller. Simulation results illustrated that, compared to a PID controller, the proposed NN-based controller exhibited superior performances with respect to tracking error, total control effort and maximum torque requirement.

References

1. Bingol MC, Akpolat ZH, Koca GO. Robust control of a robot arm using an optimized PID controller. International Conference Mechatronics; Cham: Springer, 2017. p. 484–92.
2. Youns MD, Attia SM, Abdulla AI. Position control of robot arm using genetic algorithm based PID controller. AL Rafdain Eng J. 2013;21(6):19–30.
3. Guo, L. S., Q. Zhang. Adaptive trajectory control of a two DOF closed-chain robot. Proceedings of the 2001 American Control Conference (Cat. No. 01CH37148). Vol. 1. IEEE; 2001.
4. Lin Ming-Chang, Chen Jian-Shiang. Experiments toward MRAC design for linkage system. Mechatronics. 1996;6(8):933–53.
5. Fan L, Joo M. Linear and nonlinear PD-type control of robotic manipulators for trajectory tracking. IEEE Conference on Industrial Electronics and Applications; 2009.
6. Mohammed RH, Bendary F, Elserafi K. Trajectory tracking control for robot manipulator using fractional order-fuzzy-PID controller. Int J Comput Appl. 2016;134(15):8887.
7. Sinha A, Mishra RK. Smooth sliding mode controller design for robotic arm. International Conference on Control, Automation, Robotics and Embedded Systems (CARE); 2013.
8. Boudoua S, Hamerlain M, Hamerlain F. Intelligent twisting sliding mode controller using neural network for pneumatic artificial muscles robot arm. International Workshop on Recent Advances in Sliding Modes (RASM); 2015.

9. Parlakçi MNA, Jafarov1 EM, İstefanopulos Y. Design of robust PD-sliding mode controller for robot position systems with parameter perturbations. IFAC Proceedings Volumes. 2003;36(7):105–10.
10. Su CY, Stepanenko Y. Adaptive variable structure set point control of under actuated robots. IEEE Trans Automat Control. 1999;44(11):2090–93.
11. Bailey E, Arapostathis A. Simple sliding mode control scheme applied to robot manipulators. Int J Control. 1987;45(4):1197–09.
12. Yeung KS, Chen YP. A new controller design for manipulators using the theory of variable structure systems. IEEE Trans Automa Control. 1988;33(2):200–206.
13. Youcef-Toumi K, Ito O. A time delay controller for systems with unknown dynamics. J Dyn Syst Meas Control. 1990;112(1):133–42