Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 44 No. 9
September 2023

Identifying The Dynamic Model of the 2-Link Robots in A Functional Study

Hamid Reza Eslami[1*], Abbas Chatraei[2], Maede Narimani[3]
[1] MSc. Electric Control Engineering, Faculty of Electrical Engineering, Islamic Azad University, Najafabad, Iran, Email: hamidrezaeslamii@gmail.com
[2] PhD, Electric Control Engineering, Faculty of Electrical Engineering, Islamic Azad University, Najafabad, Iran, Email: abbas.chatraei@gmail.com
[3] MSc, Electric Control Engineering. Faculty of Electrical Engineering, Islamic Azad University, Najafabad, Iran, Email: maede.narimani@gmail.com
***Corresponding author:** Hamid Reza Eslami, Email: hamidrezaeslamii@gmail.com

Abstract- Over the past few years, neural network algorithms have led to the development of new and powerful tools for identifying and modeling nonlinear systems. This study aims to introduce a new recognition tool based on neural networks and strengthen its efficiency as a new recognition method. Also, the developed neural network-based tool is used to extract the dynamic model of the guard robot in a functional study. One of the goals of this study is to implement the system recognition algorithm on the 2-link robot to achieve its dynamic model. To do this, least squares and recursive least squares algorithms and neural networks were used and the results were compared with each other. First, the characteristics of the robot under study are presented, and the modeling and recognition steps are performed on a 2-link robot in MATLAB software. Based on the information and the resulting dynamic equations, the correctness of the performance of the algorithms for use in the recognition of the robot under study was confirmed. Subsequently, all the proposed steps were implemented again for the guard robot.

Keywords-2-link robot, dynamic model, least squares algorithm, recursive least squares algorithm, neural networks.

1. Introduction

Over the past few years, the development and dominance of robotics have accelerated in various fields such as automation, medicine, space, rescue, and service provision. According to the predictions of scientists in this field, the development process will improve in the coming years in more advanced categories such as inference and cognitive science, nano-robots, and interaction with humans. As a result, more and more investments will be made in this field. Reaching an accurate mathematical model to control the subject under discussion is the goal of many researchers. The system recognition process is a tool to achieve this goal. Investigating recognition algorithms and how to use neural networks to identify the mathematical model of the system as a new, advanced, and intelligent method is one of the key goals of this research. Dynamic models are functions of the geometric parameters of robots (length of links, the angle between the axis of joints, etc.) and dynamic parameters (inertia, mass, friction, etc.). Robot dynamics study the mapping of the forces applied to the parts of the robot as well as the movement of the robot, i.e. the position, speed, and acceleration of the joints. These maps are obtained by using a series of mathematical equations describing the dynamic behavior of the robotic arm. The dynamic model is a tool for calculating the position, speed, and acceleration of the joints due to the torque applied to each joint (direct dynamics). Also, in the inverse dynamics section, the key goal is to calculate the moments of the joints to reach the desired position, speed, and acceleration for the joints.

In [1], neural network controllers are designed and implemented for a mechanical handy arm of a 2-link robot. In this study, a static perceptron neural network model is used. First, the dynamic controller generates the inverse of the input torque. Then recognition is taught offline using the back-propagation algorithm. Researchers in [2] have presented a new recognition method using neural networks based on the compensation of uncertain dynamics. Subsequently, to prove the performance, they used the proposed method on a SCARA industrial robot arm in a practical study. In [3], the detection of discrete-time nonlinear systems using neural networks with a hidden layer is investigated. A wide class of discrete-time nonlinear systems can be represented by the NARMAX model. Researchers in [4] have used a predictive model control method to identify and optimize uncertain nonlinear dynamic systems based on two recurrent neural networks. In [5], a wavelet fuzzy neural network (WFNN) is presented to identify and control the dynamics of nonlinear systems.

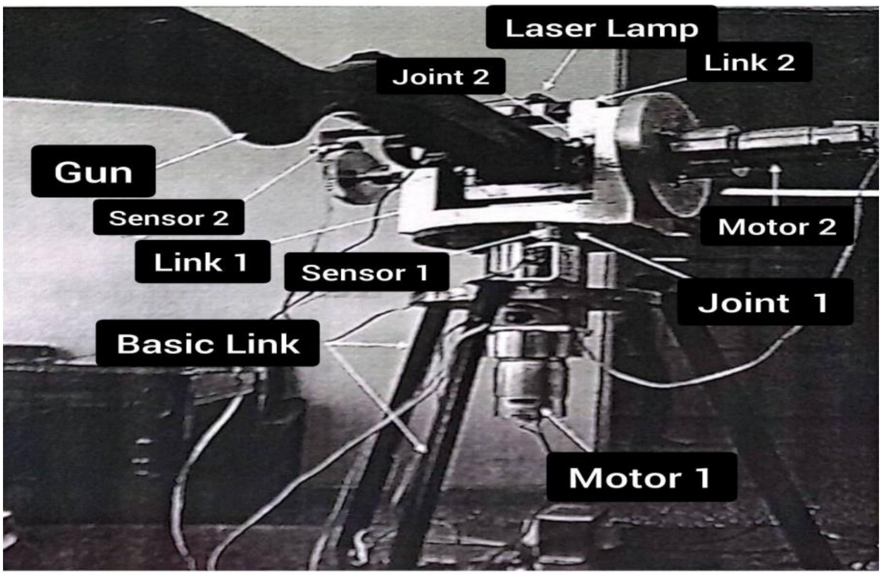
In [6], how to solve problems such as uncertainty and non-linear properties caused by different loads and un-modeled dynamics is studied. To achieve the mentioned goal, the neural network is combined with a linear robust controller. In [7] four multilayer perceptron neural networks (MLP), Elman network, NARXSP networks, and radial basis function (RBF) network are described. The above neural networks have been applied to control and identify DC servo motors and non-linear systems. In [8], neural network detection capability is used for the online diagnosis of closed-loop control systems. The neural network is trained online using a back-propagation optimization algorithm with an adaptive learning rate. In [9], it is proposed to identify and model the dynamics of the Staubli RX-60 robot based on the use of least squares and particle swarm methods. Then, several recognition tests have been performed in which the position, speed, acceleration, and torque of the robot's joints have been measured by the encoders installed on the motors. In [10], researchers have proposed a new method called the direct and inverse dynamic recognition model (DIDIM) to identify the dynamics of the robot, which only measures the torque of the joint force. In this method, the position of the joint is placed with the torque of the joint force. The whole process is based on a closed-loop simulation of the robot using a direct dynamic model. In [11], a novel method is presented that is computationally efficient and follows optimality criteria for robot path design. Experiments have also been performed on the Staubli TX-90 robot.

A comprehensive review of different recognition methods of series and parallel robot systems has been done in [12]. Dynamic parameter estimation methods are summarized and the advantages and disadvantages of each method are examined. In [13], a fuzzy neural network (FNN) is used to identify and control nonlinear dynamic systems such as a three-membered robot arm. The motion equation of the three-membered robot arm is derived from Lagrange's equations. Also, to realize control, direct, and inverse adaptive control methods based on FNN have been presented.

The main purpose of this research is to apply appropriate experimental recognition methods on a 2-link robot with rotary joints to estimate the dynamic parameters of the system and find the dynamic model of the robot under study.

2. Guard Robot

The title is centred on the page and is CAPITALIZED AND SET IN BOLDFACE. Please limit the title to a maximum length of 14 words. Author names, affiliations, email addresses, etc. should not appear on the initial manuscript anywhere. Footnotes are indicated within the text by an asterisk sign "*" in superscript, which is used to document author information and the date the manuscript was received. Do NOT separate footnotes by a horizontal line from the body text.

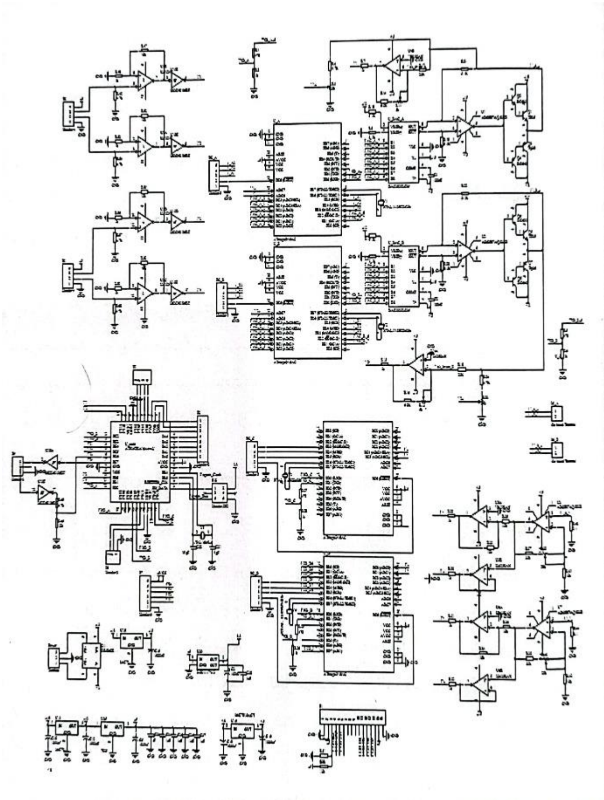


The robot under study in this research is called the guard robot, which contains the following characteristics:

2.1. Mechanical Structure and its Components

Based on the functions defined for the project, the specialists in the field of mechanics presented the current design with several different designs and fixed the defects. However, further modifications are needed to achieve better performance. The mechanical structure used in this project according to (1) includes the following components.

Figure 1. The final mechanical structure of the research plan.



2.2. Basic Links and Connections

As shown in Fig. 1, the robot has a base link, links 1 and 2. To keep and connect the main structure to a fixed environment such as the ground, three relatively strong metal bases have been used. The first joint connects the base link and the first link. The second joint is also used to connect link 1 and link 2. It should be noted that the rigidity of connections is also observed in this robotics project.

2.3. Joints

To connect two consecutive links in a robot, three types of joints, i.e. rotary, sliding, and composite, can be used. According to the intended application and performance of this project, we need two rotary joints. The first joint connects links 1 and 2 and has a coordinate system $x_0y_0z_0$. The axis z_0 is the axis of joint 1. It can be said that joint 1 is the generator of the horizontal movement of the gun. That is, the movement is formed in the plane x_0y_0 and the movement of this joint is a sector of 360 ° in this plane. The second joint has a coordinate system $x_1y_1z_1$. The axis z_1 is the axis of joint 2. In this section, the second joint is used as a rotary joint for the vertical movement of the gun. The movement of the joint in the plane x_1y_1 is about 45° vertical.

2.4. Electric motors

In this robot, two compound DC motors with gearboxes 1 and 2 are used according to Fig. 1. The motors are of single-axis type. Also, the horizontal motor and the vertical joint motor have the specifications of 12_v, $10R.P.M$ and 12_v, and $5R.P.M$, respectively.

2.5. Sensors

Two volume sensors are used in this robotic system. Sensor 1 is a multi-turn volume to calculate joint angle 1 (according to Fig. 1). Also, Sensor 2 is a conventional volume to calculate joint angle 2. However, after conducting numerous experiments, we concluded that using an OptoCounter leads to much better measurement accuracy. Therefore, instead of volume sensors, OptoCounters mounted on the motor shaft were used.

2.6. Gun

Finally, the gun is mounted on the final arm. Due to military restrictions, a simple air rifle with a relatively low rate of fire has been used.

2.7. Electronic Starter Circuit

A general representation of the schematic of the electronic board designed to operate the guard robot is reflected in Fig. 2.

Figure 2. A schematic representation of the electronic board designed to operate the guard robot.

3. Block Diagram of Project Control System

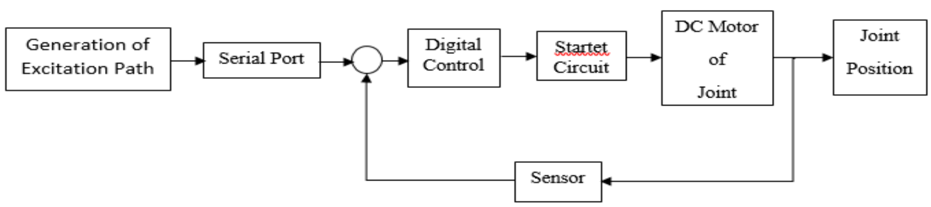


Figure 3. Block diagram of the control system used in this project.

To explain the design for the guard robot system, the diagram shown in Fig. 3 is used. As shown in the figure, the ultimate goal is to move the entire structure to a position where the tip of the gun used is in the right position and direction. To move the tip of the gun to a point in a certain position and direction, we need to calculate the corresponding angle for the joints of the robot through direct kinematics for the position and direction of the tip of the gun. Then we will direct the joints to the desired angle by giving the appropriate command to the motors.

3.1. Estimation of Non-Deterministic Parameters with Least Squares Technique

The steps for performing least squares calculations are as follows.

The equations compiled in the form of Equation (1) are applied to l time samples (l<11n), on a movement path (sinusoidal path with different frequencies) and lead to the creation of the following regression equation:

$$\Gamma = W\theta_d \quad (1)$$

where $\theta_f \in R^{2n}$ is called the observation matrix or regressor and is represented as follows:

$$W = \begin{bmatrix} Y(q(t_1), \dot{q}(t_1), \ddot{q}(t_1)) \\ \vdots \\ Y(q(t_1), \dot{q}(t_1), \ddot{q}(t_1)) \end{bmatrix} \quad (2)$$

Where

$$\Gamma = [\tau_d(t_1), \dots, \tau_d(t_1)]^T \quad (3)$$

According to the general structure obtained for the linear regression equations for a 2-link robot, we use the least squares parameter estimation formula and use it to run the MATLAB simulation.

$$\hat{\theta}_{LS} = (W^T W)^{-1} W^T \Gamma = \left(\sum_{k=1}^{N} Y(k) Y(k)^T\right)^{-1} \left(\sum_{k=1}^{N} Y(k)\Gamma(k)\right)$$

The robot modeling shown in Fig. 4 has been used to collect the necessary information in the MATLAB simulation environment, which is reflected in the Subsystem 2 block in Fig. 4. As shown in the figure, the combination of sinusoidal inputs has been used to determine the desired signals for the angles of the joints q1 and q2, and the desired information has been extracted as feedback from the robot model. Also, $\tau_2, \tau_1, \ddot{q}_2, \ddot{q}_1, \dot{q}_2, \dot{q}_1, q_2, q_1$ which respectively represent the position, speed, acceleration, and torques of joints 1 and 2, which are used in the LS parameter estimation algorithm [1,2

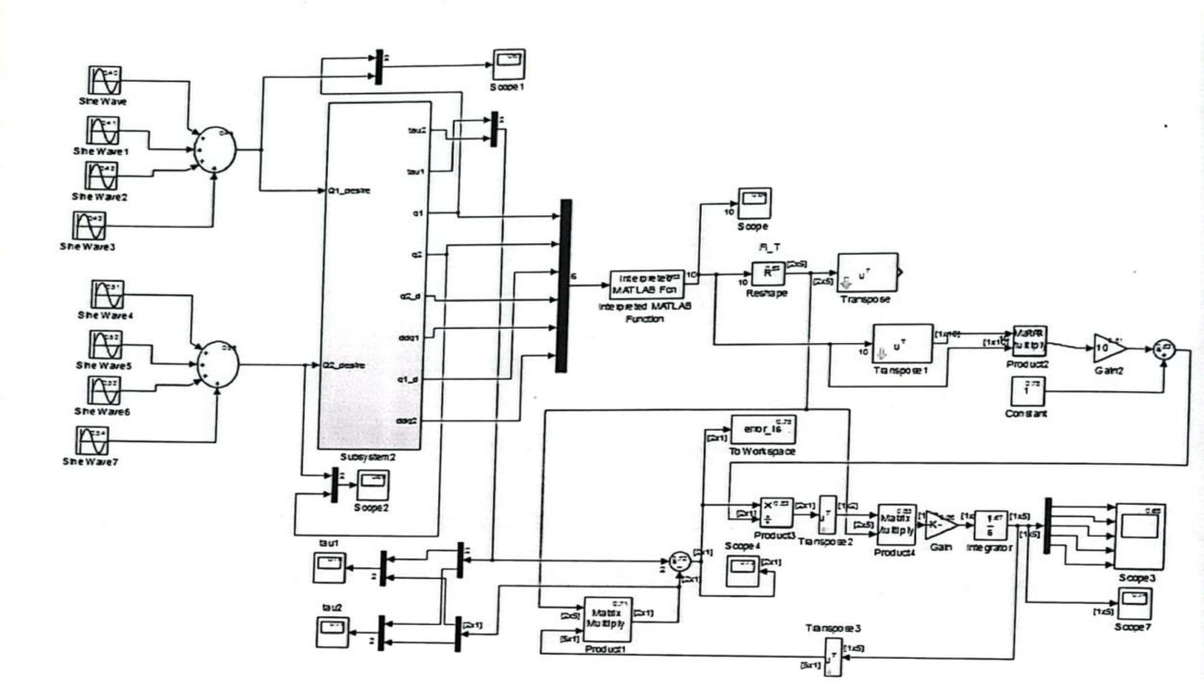


Figure 4. Estimation of uncertain parameters based on the LS method.

3.2. Estimation of Non-Deterministic Parameters with RLS Recursive Least Squares Method

Here, only the dynamic equations of the 2-link robot are shown in the form of mathematical relations.

$$\hat{\hat{\theta}}(k) = \hat{\theta}(k-1) + K(k)[y(k) - \phi^T(k)\hat{\theta}(k-1)] \quad (5)$$

$$K(k) = P(k-1)\phi(k)[I + \phi^T(k)P(k-1)\phi(k)]^{-1} \quad (6)$$

$$P(k) = [I - K(k)\phi^T(k)]P(k-1) \quad (7)$$

$y(K)$: the deterministic vector of the output of the system at every moment. Here the output of the system is the torque applied to the joints.

$\emptyset$(K): a normal matrix whose entries are a function of the input and derivatives of the system at any moment. Here, the system input is the path along which the robot moves.

θ: a vector of unknown dynamic parameters of the system whose domains are the values of fixed numbers that must be estimated.

$P(K_0)$: a symmetric and positive definite R×R matrix whose initial values must be determined. Here R is equal to the number of unknown parameters.

$\hat{\theta}(K_0)$: a vector with initial values, whose entries are selected based on trial and error.

$K(K_0)$: a matrix whose dimensions depend on the matrix ϕ(K).

Now we replace the equations related to the 2-link robot in the following relations:

$$K(k)_{5\times 2} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} \\ p_{31} & p_{32} & p_{33} & p_{34} & p_{35} \\ p_{41} & p_{42} & p_{43} & p_{44} & p_{45} \\ p_{51} & p_{52} & p_{53} & p_{54} & p_{55} \end{bmatrix} \times \begin{bmatrix} \phi_{11} & \phi_{21} \\ \phi_{12} & \phi_{22} \\ \phi_{13} & \phi_{23} \\ \phi_{14} & \phi_{24} \\ \phi_{15} & \phi_{25} \end{bmatrix} \quad (8)$$

$$\times \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} & \phi_{15} \\ \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} & \phi_{25} \end{bmatrix}\right. \quad (9)$$

$$\left.\times \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} \\ p_{31} & p_{32} & p_{33} & p_{34} & p_{35} \\ p_{41} & p_{42} & p_{43} & p_{44} & p_{45} \\ p_{51} & p_{52} & p_{53} & p_{54} & p_{55} \end{bmatrix} \times \begin{bmatrix} \phi_{11} & \phi_{21} \\ \phi_{12} & \phi_{22} \\ \phi_{13} & \phi_{23} \\ \phi_{14} & \phi_{24} \\ \phi_{15} & \phi_{25} \end{bmatrix}\right]^{-1} \quad (10)$$

$$P(k)_{5\times 5} = \left[\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \\ k_{31} & k_{32} \\ k_{41} & k_{42} \\ k_{51} & k_{52} \end{bmatrix}\right. \quad (11)$$

$$\left.\times \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} & \phi_{15} \\ \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} & \phi_{25} \end{bmatrix} \times \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} \\ p_{31} & p_{32} & p_{33} & p_{34} & p_{35} \\ p_{41} & p_{42} & p_{43} & p_{44} & p_{45} \\ p_{51} & p_{52} & p_{53} & p_{54} & p_{55} \end{bmatrix}\right] \quad (12)$$

$$\hat{\theta}(k)_{5\times 1} = \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \hat{\theta}_3 \\ \hat{\theta}_4 \\ \hat{\theta}_5 \end{bmatrix} + \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \\ k_{31} & k_{32} \\ k_{41} & k_{42} \\ k_{51} & k_{52} \end{bmatrix} \left[\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} - \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} & \phi_{15} \\ \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} & \phi_{25} \end{bmatrix} \times \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \hat{\theta}_3 \\ \hat{\theta}_4 \\ \hat{\theta}_5 \end{bmatrix}\right] \quad (13)$$

$\hat{\theta}(k)$ is estimated in such a way that the term $\tau(k) - \phi^T(k)\hat{\theta}(k)]$ reaches the lowest possible value. The term $\phi^T(k)\hat{\theta}(k)$ is equivalent to $\hat{\tau}(k)$. Fig. 5 shows how to simulate the recursive least squares method in Simulink software. According to Fig. 6, the plots converge to the desired values.

After estimating the non-deterministic parameters and comparing them with the estimated torque, the torque is applied to the modeled robot in MATLAB through the PID controller. According to Figs. 7 and 8, it can be concluded that the recursive least squares method has a better performance in estimating the values of unknown parameters than the conventional least squares method. As shown in the plots, based on the comparison of the estimated moments and the torque of the robot model, it is clear that the error between them is less than the previous method and the estimated torque diagram is more similar to the actual torque diagram of the robot model.

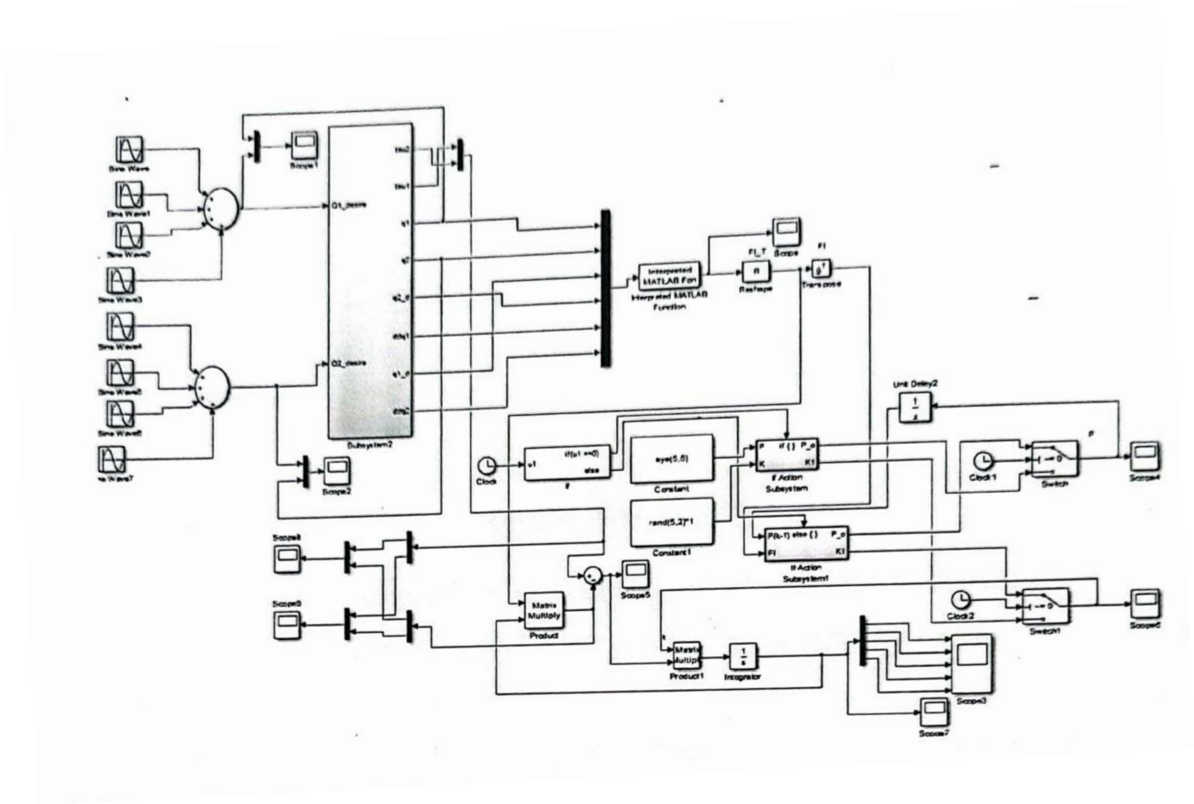


Figure 5. Estimation of non-deterministic parameters based on the method

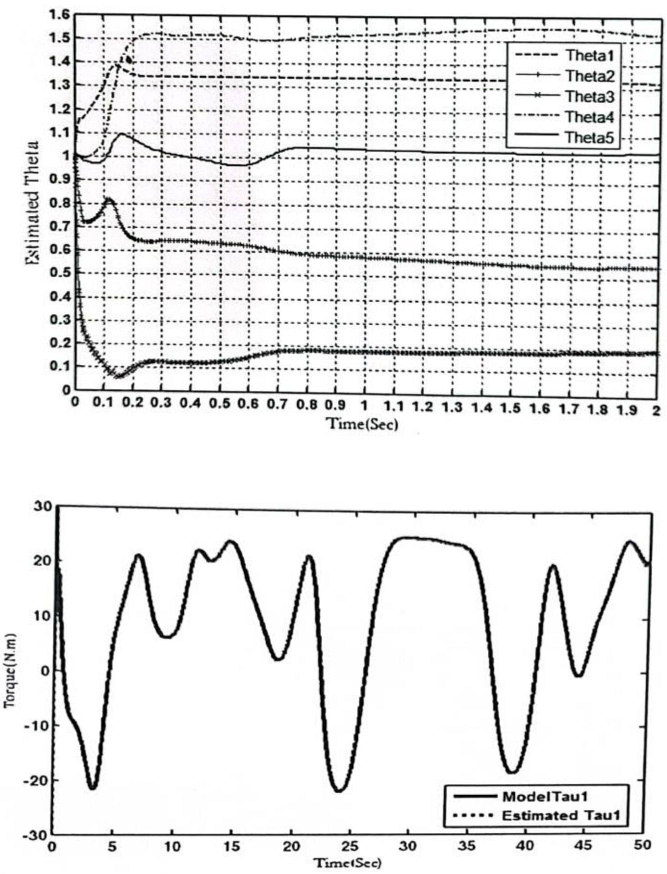


Figure 6. Parameter estimation diagram based on recursive least squares method.

Figure 7. Comparison of the estimated torque of joint 1 and the actual torque of the 2-link robot using the RIS method.

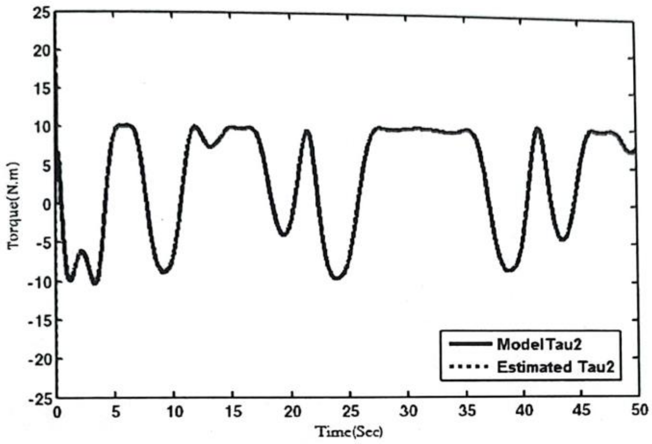


Figure 8. Comparison of the estimated torque of joint 2 and the actual moment of the 2-link robot using the RIS method.

3.3. Estimation of Non-Deterministic Parameters by Multilayer Neural Network Method

For the 2-link robot model simulated in MATLAB software, the input and output structure of neural networks is considered in the form of Fig. 9.

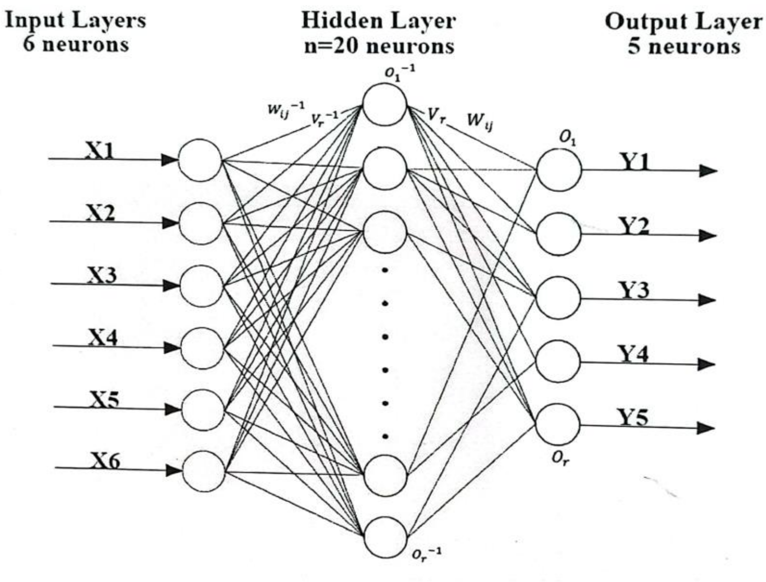


Figure 9. Input and output structure of multilayer neural network.

$$X = [q_1 \quad q_2 \quad \dot{q}_1 \quad \dot{q}_2 \quad \ddot{q}_1 \quad \ddot{q}_2 \;]^T \tag{14}$$

$$Y = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \;]^T \tag{15}$$

The position, speed, and acceleration of each joint are considered inputs of neural networks. The number of hidden layer neurons is considered equal to 20 by the trial and error method. The number of neurons in the output layer is also equal to the unknown parameters that must be estimated by the neural network. The activation function of hidden layer neurons is the Tan-sigmoid function and the transfer function of hidden layer neurons is the pure-line function. The algorithm should adjust the network parameters to reduce the mean square error.

$$F = \frac{1}{2} \sum_{k=1}^{n} (t_k - o_k)^2 \tag{16}$$

As you can see, the mean square error has been replaced by the squared error in the K^th^ iteration.

For the output layer

$$W_{ij}[n+1] = W_{ij}[n] - \alpha \frac{\partial F(\omega)}{\partial W_{ij}} \quad (17)$$

$$\Delta W_{ij} = -\alpha \frac{\partial F(\omega)}{\partial W_{ij}} \quad (18)$$

where α represents the learning rate.

Based on the chain rule, we will have:

$$\frac{\partial F(\omega)}{\partial W_{ij}} = \sum_{k=1}^{n} \frac{\partial F}{\partial O_r} \cdot \frac{\partial O_r}{\partial V_r} \cdot \frac{\partial V_r}{\partial W_{ij}} \quad (19)$$

Also for the hidden layer, we have:

$$W_{ij}^{-1} = W_{ij}[n]^{-1} - \alpha \frac{\partial F(\omega)}{\partial W_{ij}^{-1}} \quad (20)$$

$$\Delta W_{ij}^{-1} = -\alpha \frac{\partial F(\omega)}{\partial W_{ij}^{-1}} \quad (21)$$

According to the chain rule we have:

$$\frac{\partial F(\omega)}{\partial W_{ij}^{-1}} = \sum_{k=1}^{n} \frac{\partial F}{\partial O_r} \cdot \frac{\partial O_r}{\partial V_r} \cdot \frac{\partial V_r}{\partial o_i^{-1}} \cdot \frac{\partial o_i^{-1}}{\partial V_i^{-1}} \cdot \frac{\partial V_i^{-1}}{\partial W_{ij}^{-1}}. \quad (22)$$

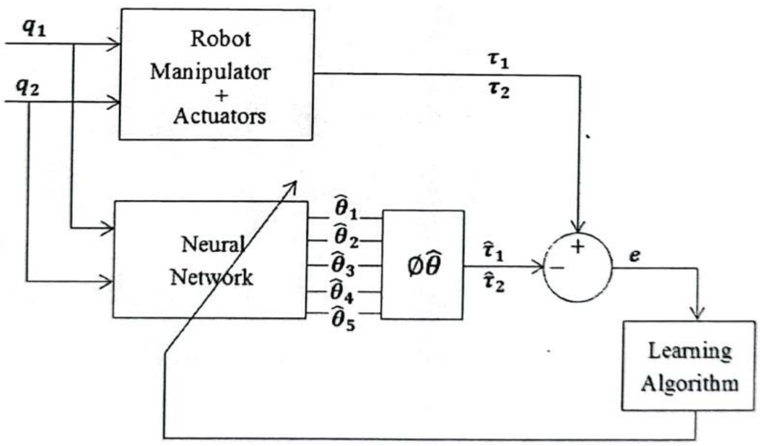


Figure 10. Neural network block diagram for estimating dynamic parameters of the 2-link robot.

The efficiency criterion is defined here as follows:

$$F = \frac{1}{2}\sum_{k=1}^{n} (\tau_{Actual} - \tau_{Extimated})^2 \quad (23)$$

Here, according to the block diagram of the neural network shown in Fig. 10, chain rules are formulated as follows.

$$V_r = \sum_{k=1}^{n} W_{ij} \times o_i^{-1}$$

$$O_r = \phi(V_r) = Purelin\,(V_r) \quad (24)$$

$$\tau_r = Y(q, \dot{q}, \ddot{q})\theta = \phi\theta$$

$$\Delta W_{ij} = -\alpha \frac{\partial F(\omega)}{\partial W_{ij}} = \sum_{k=1}^{n} \frac{\partial F}{\partial O_r} \cdot \frac{\partial O_r}{\partial V_r} \cdot \frac{\partial V_r}{\partial W_{ij}} = \sum_{k=1}^{n} \frac{\partial F}{\partial \tau_r} \cdot \frac{\partial \tau_r}{\partial \theta_m} \cdot \frac{\partial \theta_m}{\partial V_r} \cdot \frac{\partial V_r}{\partial W_{ij}}$$

Where *i,j=1,2,...,20* and *m=5*

Also, the Relation $Y(q, \dot{q}, \ddot{q})$ is the same as the regressor matrix.

$$\frac{\partial F(\omega)}{\partial W_{ij}^{-1}} = \sum_{k=1}^{n} \frac{\partial F}{\partial O_r} \cdot \frac{\partial O_r}{\partial V_r} \cdot \frac{\partial V_r}{\partial o_r^{-1}} \cdot \frac{\partial o_r^{-1}}{\partial V_r^{-1}} \cdot \frac{\partial V_r^{-1}}{\partial W_{ij}} \quad (28)$$

$$= \sum_{k=1}^{n} \frac{\partial F}{\partial \tau_r} \cdot \frac{\partial \tau_r}{\partial \theta_m} \cdot \frac{\partial \theta_m}{\partial V_r} \cdot \frac{\partial V_r}{\partial o_r^{-1}} \cdot \frac{\partial o_r^{-1}}{\partial V_r^{-1}} \cdot \frac{\partial V_r^{-1}}{\partial W_{ij}}$$

$$V_r^{-1} = \sum_{k=1}^{n} W_{ij}^{-1} \times x_l \tag{29}$$

$$O_r^{-1} = f^{-1}(V_r^{-1}) = Tansigmoid(V_r) \tag{30}$$

$$x_l = [q_1 \quad q_2 \quad \dot{q}_1 \quad \dot{q}_2 \quad \ddot{q}_1 \quad \ddot{q}_2\ ]^T \tag{31}$$

The data used to train the neural network are the same data related to the simulation robot model in MATLAB. However, in the next step, the real information of the robot itself is used for the guard robot. After coding in MATLAB software, the diagram shown in Fig. 11 is defined for the neural network, and then the results and related plots are checked.

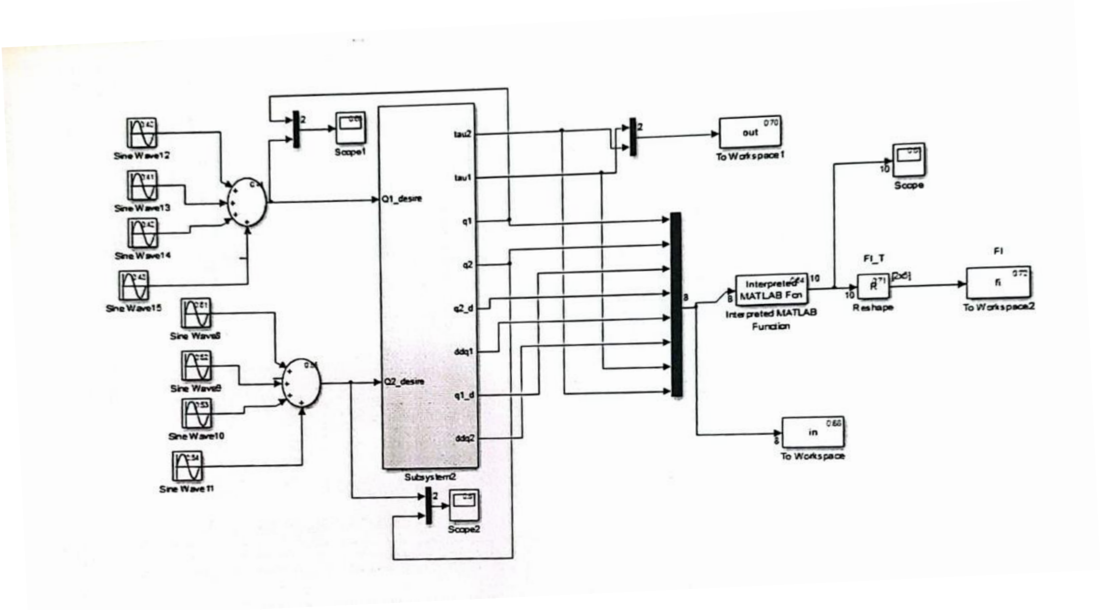


Figure 11. Estimation of dynamic parameters of the 2-link robot with multilayer neural network.

We set the training rate to 0.001 to implement the neural network. Also, the number of iterations is 200 and the target value is 0.001. Finally, the sampling rate is equal to 0.001 seconds.

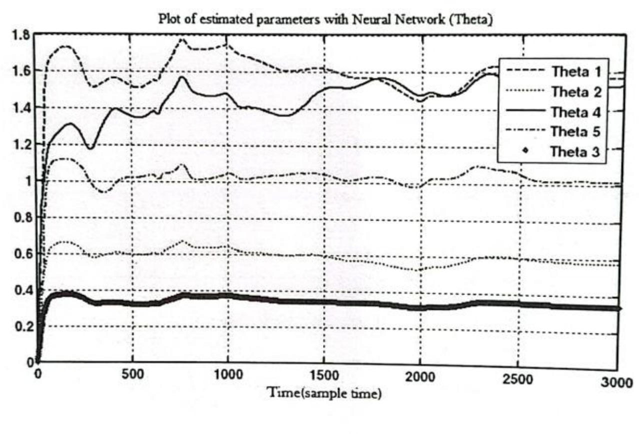


Figure 12. Diagram of estimation of parameter $\widehat{\theta}$ with MLP neural network.

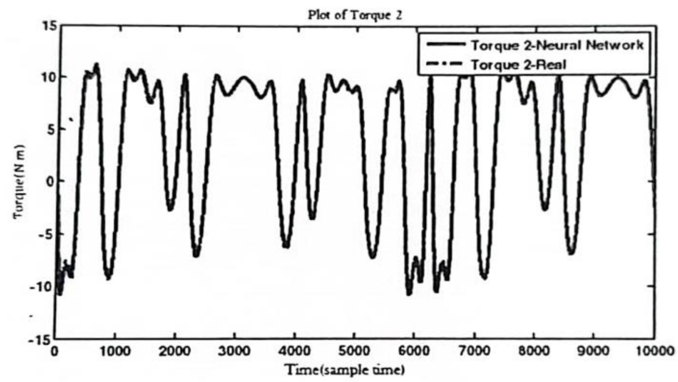


Figure 13. Comparison of estimated joint 1 torque and 2-link robot model.

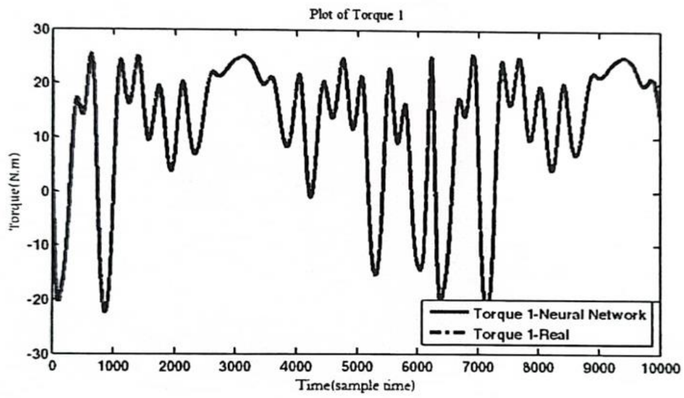


Figure 14. Comparison of estimated joint 2 torque and 2-link robot model.

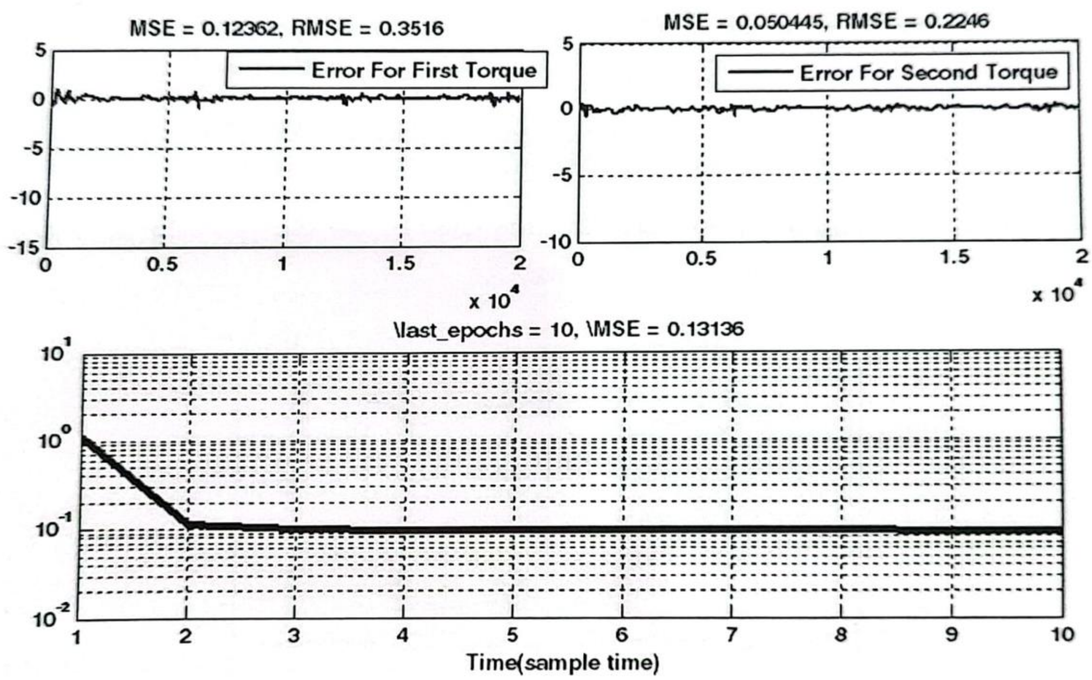
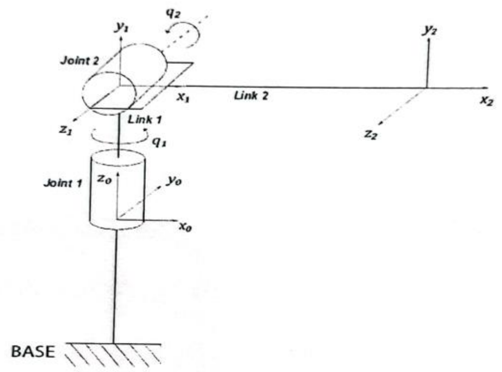


Figure 15. The results of neural network training.

According to the plots in Fig. 12, the speed and accuracy values are better than the previous two methods. It confirms the research hypothesis that the performance of the neural network in system recognition is better than the least square method. In addition, according to Fig. 13 showing the results of neural network training, we reached the desired answer after 10 rounds of training. It is also possible to train the developed model with proper accuracy even with one round of training. The simulation results can be compared using Table 1.

Table 1. Simulation results.

Title	**Actual Value**	**LS**	σ_i	**RLS**	σ_i	**NN**	σ_i
θ_5	**1.5**	**1.2264**	**0.2736**	**1.0998**	**0.4002**	**1.513**	**-0.013**
θ_2	**0.5**	**0.4401**	**0.0599**	**0.4675**	**0.0325**	**0.052432**	**-0.02432**
θ_3	**0.25**	**0.2286**	**0.0214**	**0.2662**	**-0.0162**	**0.3015**	**-0.0515**
θ_4	**1.5**	**1.5012**	**-0.0012**	**1.4769**	**0.0231**	**1.4221**	**0.0779**
θ_5	**1**	**1.0349**	**-0.0349**	**1.0163**	**-0.0163**	**-0.0489**	**-0.0489**



4. Implementation of Recognition Algorithms on the Guard Robot in a Functional Study

The dynamic equations of the 2-link plane robot, which was theoretically investigated in the previous section, are presented in Reference [9]. We used these equations to implement recognition algorithms. Now we repeat all the steps of the previous section for this robot. According to Fig. 14, the arm is connected to two rotational joints. Using the Denavit-Hartenberg rules, we write the coordinates and parameters of the

Figure 14. Symbolic representation of the guard robot.

Denavit-Hartenberg system for the guard robot as follows:

Table 2. Denavit-Hartenberg variables

Link	a_i	α_i	d_i	θ_i
1	0	90	d_i	θ_1^*
2	a_2	0	0	θ_2^*

$$d_1 = 45 \quad cm \tag{32}$$

$$d_2 = 10 \quad cm \tag{33}$$

the origin o_2 is placed at the end of link

5. Real Information Extraction of the

In this section, the function of the systen
First, the path of the robot is de
generating the stimulation path in MATl
Then, with the software designed by Vis

angle values are sent to the microcontroller at each time step through the serial port. Based on the existing program, the microcontroller provides the necessary commands to apply voltage to the motors connected to the robot's joints through the appropriate drive circuit.

Based on these commands, the robot goes to the desired position and direction. Again, this position is fed back to the microcontroller through the sensors installed on the robot so that the desired position and direction are accurately recorded. At any moment, the PID controller designed in the microcontroller minimizes the error between the desired path applied to the robot and the path traveled by the robot. Also, the information obtained from the robot such as position, speed, acceleration, and voltage and current related to the motor is stored in the computer at any moment with a sampling time of 15 milliseconds. This information is used in recognition algorithms after noise removal.

6. How to produce the excitation path

In identifying the system, determining a movement path is crucial. In the case of identifying the dynamic model of the robot, this movement path plays a significant role in stimulating all the dynamics of the robot. The excitation path should be such that the information obtained from the system is rich. Otherwise, the estimation of parameters may not converge to fixed numbers. To generate rich information that can be used in recognition algorithms, the excitation path must include different frequencies. Two common types of trajectories are conic and periodic polynomials. The former is desirable for most industrial arms that accept only simple speed control. While the latter is suitable for open controller architectures that allow the user to program his desired paths on the robot. In this experiment, we used the periodic path in the form of Fourier summation.

$$q_i(t) = q_{i0} + \sum_{l=1}^{N} a_{il}\, sin(\omega_f lt)\, b_{il} cos\,(\omega_f lt) \quad (34)$$

$$\dot{q}_i(t) = \sum_{l=1}^{N} a_{il}\omega_f l cos\,(\omega_f lt) + b_{il} cos \omega_f l\,(\omega_f lt) \quad (35)$$

$$\ddot{q}_i(t) = \sum_{l=1}^{N} -a_{il}(\omega_f l)^2\, sin(\quad lt) + b_{il}(\omega_f l)^2 cos\,(\omega_f lt) \quad (36)$$

Where ω_f is the frequency of the excitation paths, which must be chosen very carefully to avoid the excitation of the unmodeled dynamics of the robot.

The path optimization problem determines the Fourier series coefficients q_{i0}, a_k^i and b_k^i so that the following cost function is minimized:

$$f\big(q_i(t)\big) = \lambda_1 cond(W_c) + \lambda_2 \frac{1}{\sigma_0(W_c)} \quad (37)$$

Where λ_1 and λ_2 are scalar values that respectively provide the relative weights between the number of conditions of the observer matrix $cond(W_c)$ and the lowest eigenvalue of the matrix $\sigma_0(W_0)$.

Note that the above problem is a constrained optimization problem because the physical constraints of position, velocity, and acceleration must also be considered [14].

Therefore, the excitation path is obtained by solving the following optimization problem:

$$\min_{q,\dot{q},\ddot{q}} J_c = cond(W) \quad (38)$$

This optimization problem is easily solved by using the fmincon command in MATLAB software. Using the above algorithm, the optimal excitation path is created. Subsequently, the robot is commanded to follow this optimal path. Robot responses are recorded along the way. The collected data should be processed to improve the quality before being used in the estimation of dynamic parameters.

Conclusion

In this research, by properly modeling the robot and using real data, we were able to obtain optimal dynamic equations for model-based controllers. By estimating the unknown dynamic parameters of the robot, the torque obtained with the estimated parameters is close to the actual torque of the robot.

The proposed recognition method is based on multilayer neural networks. The efficiency of the recognition method was evaluated by the real data of the robot. The use of position measurement sensors with better accuracy is suggested to improve the accuracy of the system model.

Funding: Islamic Azad University, Najaf Abad branch

Conflict of interest: The authors declare that they have no conflict of interest.

References

[1] F.E. Serrano, Neural Networks Models for Control and Identification of a Two Links Robotic Manipulator, in *Proceedings CONCAPAN XXXI*, 2011.

[2] Y. Tan, X. Dang, F. Liang, & C.-Y. Su, Dynamic wavelet neural network for nonlinear dynamic system identification, in *IEEE International Conference on Control Applications. Conference Proceedings (Cat. No. 00CH37162)*, 2000, pp. 214-219.

[3] M. Gautier, & P. Poignet, Extended Kalman filtering and weighted least squares dynamic identification of robot, *IFAC Proceedings Volumes*, *33*(15), 2000, 935-940.

[4] S. Chen, S.A. Billings, & P. Grant, Non-linear system identification using neural networks, *International Journal of Control*, *51*(6), 1990, 1191-1214.

[5] Y. Pan, & J. Wang, Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks, *IEEE Transactions on Industrial Electronics*, *59*(8), 2001, 3089-3101.

[6] C.-H. Lu, Wavelet fuzzy neural networks for identification and predictive control of dynamic systems, *IEEE Transactions on Industrial Electronics*, *58*(7), 2010, 3046-3058.

[7] M. Moradi, & H. Malekizade, Neural network identification based multivariable feedback linearization robust control for a two-link manipulator, *Journal of Intelligent & Robotic Systems*, *72*(2), 2013, 167-178.

[8] B. Bhushan, M. Singh, & Y. Hage, Identification and control using MLP, Elman, NARXSP and radial basis function networks: a comparative analysis, *Artificial Intelligence Review*, *37*, 2012, 133-156.

[9] J. Velagic, M. Hebibovic, & B. Lacevic, On-Line Identification of a Robot Manipulator Using Neural Network with an Adaptive Learning Rate, *IFAC Proceedings Volumes*, *38*(1), 2005, 403-408.

[10] Z. Bingül, & O. Karahan, Dynamic identification of Staubli RX-60 robot using PSO and LS methods, *Expert Systems with Applications*, *38*(4), 2011, 4136-4149.

[11] M. Gautier, A. Janot, & P.-O. Vandanjon, A new closed-loop output error method for parameter identification of robot dynamics, *IEEE Transactions on Control Systems Technology*, *21*(2), 2012, 428-444.

[12] J. Jin, & N. Gans, Parameter identification for industrial robots with a fast and robust trajectory design approach, *Robotics and Computer-Integrated Manufacturing*, *31*, 2015, 21-29.

[13] J. Wu, J. Wang, & Z. You, An overview of dynamic parameter identification of robots, *Robotics and Computer-Integrated Manufacturing*, *26*(5), 2010, 414-419.

[14] S.-Z. Qin, H.-T. Su, & T. J. McAvoy, Comparison of four neural net learning methods for dynamic system identification, *IEEE Transactions on Neural Networks*, *3*(1), 1992, 122-130.