International Journal of Scientific Research in Mechanical and Materials Engineering
© 2018 IJSRMME | Volume 2 | Issue 5 | ISSN : 2457-0435

Modeling and Controlling of a 3 DOF Robot Manipulator with Artificial Neural Networks

Bekir Cirak

Karamanoglu Mehmetbey University, Engineering Faculty, Mechanical Engineering Department,Yunus Emre Campuses, Karaman, Turkey

ABSTRACT

In this study, 3 DOF (Degree Of Free) robot manipulator were inspected. kinematic and inverse kinematic calculations were made for robot joints and Artificial Neural Network (ANN) method was applied for control. the coordinate of the end point of the robot is assumed to be input. Robot kinematic and inverse kinematic calculations and simulations were done with Matlab 14 a version.

Keywords : Robot Manipulator, Artificial Neural Networks, Kinematic, İnverse

I. INTRODUCTION

Robots have the potential to play a large role in our world. They are currently widely used in industrial applications for labor-intensive operations that require a high level of precision and repetition. In addition, robots can be found in the entertainment industry in the form of toys and animatronics. The function of robots in society is constantly evolving and current research endeavors to bring them further into the realm of domestic assistance, medicine, military, search and rescue, and exploration. In many of these applications, the robot must perform only one specific task and thus can be designed to handle a single operation. However, as the potential use for robots grows, so does their need to interact with objects in their environment (Ayman A. Aly,2010). Robots are defined as systems that perform location and direction change operations by programmed transport. Moving is a movement at a distance far from the body dimensions. Manipulator consists of multiple joints connected to each other. Movements of the joints are provided by the engine. The robot consists of mechanical parts, actuators and control units. The mechanical parts of the robot are classified as structural parts, power transmitting parts, bearings and coupling parts. The motors can also act as pneumatic or hydraulic as well as electrically. In modern robots, the control units are PC based and have advanced structures (Dhaval, V., Ohri, J., Ankit, P.2013).

Today, the usage areas of robots are increased. Especially in industrial environments, a large number of robots are used for production assembly and similar works. Because the separation media properties change rapidly, in the real world robots are required to reach the desired target without hitting the obstacles they encounter. The robots, the control technique used and the joint they contain according to their species. In terms of control technique it is possible to classify as adaptive robots, non-adaptive and intelligent robots. According to the joint types in the robots are grouped as rotary, prismatic, cylindrical, spherical and planar. In recent years, it is possible to find intelligent system applications on different stages of production. These intelligent systems are an important part of industry, healthcare and automation sector. In this study, the dynamic behavior of the 3-axis industrial robot manipulator and the change of the robot arm configuration between the joints with time are examined and the articulated neural network model of the joints given mathematical kinematic models has been inspected (Hala, B.N.D., Adel, M.A.,2002).

II. ROBOT KINEMATICS

Kinematics can be calculated in two parts; a. Forward kinematics, b. Inverse kinematics. In robot manipulators, the forward kinematic robot is used to determine the angle of the joints taking advantage of the position of the arm.

In Fig. 1, three kinematic robot manipulator simulation models, forward kinematics and robot dynamics are used (Kirby, R.,2010).

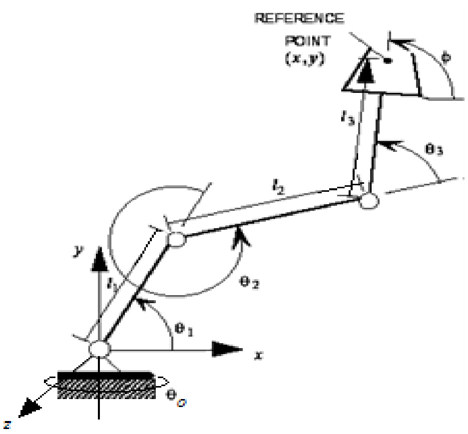


Figure 1. Three joint robots

The forward and inverse kinematic equations of the three Degree Of Freedom (DOF) robot manipulators given in Fig. 1 are given below.

2.1. Forward Kinematics

Forward kinematics equations calculate unknown values for x, y, z, and Ø from known values for the length of each of the link, l_1, l_2, and l_3, and θ_1, θ_2, and θ_3. Very simple equations are derived for the position coordinates x, y, z, and the angle of attack, Ø as follows:

$$x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \quad (1)$$

$$y = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \quad (2)$$

$$z = x \sin\theta_0 \quad (3)$$

$$Ø = \theta_1 + \theta_2 + \theta_3 \quad (4)$$

$$\theta_0 = \sin^{-1} \frac{z}{x} \quad (5)$$

2.2. Inverse kinematics

Equations of inverse kinematics are used to calculate unknown polar coordinates, θ_1, θ_2, and θ_3 values, given known values for l_1, l_2, l_3, x, y, z, and Ø . Very simple formula for θ_0 is presented below. Equations for θ_1, θ_2, and θ_3 are not so easy to be derived in a single step and need to be performed through a sequence of derivation steps as shown below:

From (1) and (2),

$$x - l_3 \cos Ø = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \quad (6)$$

$$y - l_3 \sin Ø = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \quad (7)$$

x' and y' defining from (6) and (7)

$$x' = x - l_3 \cos(Ø) \quad (8)$$

$$y' = y - l_3 \sin(Ø) \quad (9)$$

instead of (6) and (7) in (8) ve (9)

$$x' - l_1 \cos\theta_1 = l_2 \cos(\theta_1 + \theta_2) \quad (10)$$

$$y' - l_1 \sin\theta_1 = l_2 \sin(\theta_1 + \theta_2) \quad (11)$$

From where

$$(-2 l_1 x') \cos\theta_1 + (-2 l_1 y') \sin\theta_1$$

$+(x'^2 + y'^2 + l_1^2 + l_2^2) = 0$ (12)

Solving (12) to get θ_1 defining variables *P*, *Q*, and *R*

$P = -2l_1 x'$ $\quad Q = -2l_1 y'$ *and*
$R = x'^2 + y'^2 + l_1^2 + l_2^2$ (13)

Form of (12) is simplified

$$P\cos\theta_1 + Q\sin\theta_1 + R = 0 \quad (14)$$

(14) in θ_1, γ is defined

$$\gamma = atan\,2\left[\frac{Q}{\sqrt{P^2+Q^2}}, \frac{P}{\sqrt{P^2+Q^2}}\right] \quad (15)$$

Using (15), (14) can be rewritten as

$$\text{Cos}\,\gamma\,\text{Cos}\,\theta_1 + \text{Sin}\,\gamma\,\text{Sin}\,\theta_1 + \frac{R}{\sqrt{P^2+Q^2}} = 0 \quad (16)$$

Using triangle relation gives:

$$\text{Cos}\,(\theta_1 - \gamma) = \frac{-R}{\sqrt{P^2+Q^2}} \quad (17)$$

And thus formula for θ_1 is obtained as:

$$\theta_1 = \gamma + \sigma\cos^{-1}\left[\frac{-R}{\sqrt{P^2+Q^2}}\right] \quad (18)$$

where σ is the value range, ∓ 1

θ_1 therefore has two solutions and thus θ_2 should also has a corresponding couple of solutions so that the summation of θ_1 and θ_2 gives give the same (*x*, *y*) coordinates for the reference point. This result is shown in Fig. 2 below.

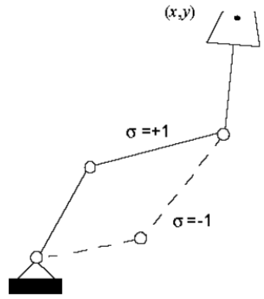


Figure 2. Coordinates for the reference point

Step 4: Deriving equations for θ_2 and θ_3. Using (10) and (11), it is easy to get θ_2 formula as:

$$\theta_2 = \text{atan2}\left[\frac{y' - l_1\sin\theta_1}{l_2}, \frac{x' - l_1\cos\theta_1}{l_2}\right] - \theta_1 \quad (19)$$

θ_3 is simply calculated using θ_1, θ_2, and $\emptyset$ values using the relation:

$$\theta_3 = \emptyset - (\theta_1 + \theta_2) \quad (20)$$

III. ROBOT DYNAMICS

Robot manipulator movement is strongly determined on the joints. Therefore, speed and acceleration information can be obtained from time-dependent force and moment changes on joints. These data are determined using the dynamic analysis of the robot manipulator it may be. For this purpose, some methods are used to obtain the dynamic equations of the robot manipulator.This methods are Lagrange-Euler (L-E), Newton-Euler (N-E), Regursive Lagrange (R-L) and generalized D'Alembert (Kosmatopoulos E.B.2005).

3.1. Lagrange-euler method

This method is defined by the total energy and work in the system. In this process, the potential and kinetic energy are utilized. The dynamic modeling of the three joint robot manipulators was done by the Lagrange-Euler method. M is momentum, *L* is Lagrange function, q_i is the coordinate of the generalized joints, $\dot{q}_i$ is the generalized velocity (Liu, J., Wang, X.2011).

Lagrange function;

$$L = K - V \quad (21)$$

$$L = \left[\frac{1}{2} \sum_{i,j} d_{ij}(q) \dot{q}_i \dot{q}_j - V(q) \right] \quad (22)$$

In this equation (4), K is kinetic energy and V is potential energy.

$$M_i = \left[\sum_j dk_j(q)\ddot{q}_j + \sum_{i,j} \frac{\partial dk_j}{\partial q_i} \dot{q}_i \dot{q}_j \right] - \left[\frac{1}{2} \sum \frac{\partial d_{ij}}{\partial q_k} \dot{q}_i \dot{q}_k - \frac{\partial V}{\partial q_k} \right] \quad (23)$$

$$M_k = \sum_j d_{kj}(q)\ddot{q}_j + \sum_{i,j} \left[\frac{1}{2} \frac{\partial k_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_i} - \frac{\partial d_{ij}}{\partial q_k} \right] \dot{q}_i \dot{q}_j + \frac{\partial}{\partial q_k} V \quad (24)$$

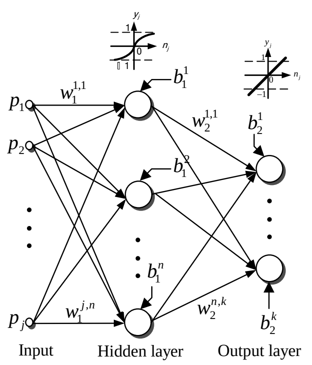
$$M = D(q)\ddot{q} + h(q, \dot{q}) + G(q) \quad (25)$$

$$\begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & K \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} \quad (26)$$

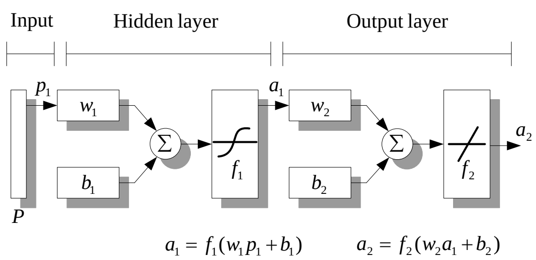
IV. ARTIFICIAL NEURAL NETWORKS (ANN)

Artificial neural networks are a method in which the learning of the human brain is tried to be applied. An artificial neural network is made up of information processing elements called neurons. A unique point of contact neurons. A unique point of contact there is weight. These calculations, which are computable, convey information directly. It is not possible to determine in advance the points of connection since the information is often disseminated to the network. For this reason, a learning algorithm in which weights are changed is needed. The purpose of the study is to calculate the point weights for untrained and untrained learning. In many cases, the network is trained using input / output pairs. The performance of this learning process is measured by using the training set to achieve the desired result and by generalizing the trained network. Educational learning is a two-layered forward feeder with the simplest network structure and an output layer. Each neuron in the output layer is signaled by all input neurons with recalculable weights. Fig. 5 shows the schematic structure of a multi-layer feed-forward network. In network input layer hidden three layers, the layer and the output layer located (Meza, J.L., Santibanez, V., Soto, R.,2012).

The BPNN (Backpropagation Neural Network) is a method for categorization and prediction **Error! Reference source not found.-Error! Reference source not found.**. The structure mainly contains input layer, output layer, and hidden layer, as shown in Fig. 3 (a). To achieve the target value, the Gradient Steepest Descent Method is applied to renew the weights, w, and the biases, b, by conveying the error gradient repeatedly. The procedure of BPNN can be divided into two phases, the feed-forward phase and the propagation phase, as shown in Fig. 3(b). The procedures are explicitly explained as below.



(a) Structure of BPNN



(b) Flowchart of BPNN.

Figure 3. Multilayer ANN

Once the input vector is identified in artificial neural networks, the weights are adjusted according to the learning process. BPNN algorithm is used in this study. BPNN algorithm is the most used algorithm in artificial neural networks. During the back propagation learning, the network passes each input pattern through the neurons in the hidden layers to produce the output neurons. It then compares the result obtained with the expected result to find the errors in the output layer. So, the output errors are passed back to the hidden layers directly from the derivative output layer. Once the error values are found, the neurons adjust their weights to reduce their faults. The weight change equations are arranged in the smallest shape of the average error margin in the network. The learning algorithm is specified by the notation given below (Rasit, K., Abdullah, F.2004). In n. iteration, j. Error mark at exit of neuron, j. On the neuron output layer,

$$ej\ (n) = dj\ (n) - yj\ (n) \qquad (27)$$

In this work, sigmoid function is used as an activation function in the calculation of local gradients. The YSA algorithm was implemented using Matlab 6.0 Neural Network Toolbox. The input and output data are very important in terms of convergence and learning process. Due to the nature of the sigmoid function, the output of the aging is between 0 and 1. For this reason, it is necessary to normalize the input / output data without any network training. In this study, input and output data were normalized to remain between 0-1 (Soltanpour, M.R., Jafaar, K.2012). (Zhu, C., Zhang, H.2011)

IV. SIMULATION

The Table 1 shows the data required for the simulation of three joint robot manipulators.

Table 1. Experimental Data

m_1 (kg)	m_2 (kg)	m_3 (kg)	L_1 (m)	L_2 (m)	L_3 (m)	I_1 (kgm^2)	I_2 (kgm^2)	I_3 (kgm^2)	I_{01} (kgm^2)	I_{02} (kgm^2)	I_{03} (kgm^2)	g (m/s^2)
0.61	2.11	2.11	0.18	0.18	0.18	0.116	0.2368	0.2368	0.2368	0.05	0.05	9.81

Where mn: m.link weight, Ln: n.link length, In: n. link inertia moment, I n n-1: link to n-1 of n. link inertia moment.The simulation was performed at the discrete time.The total simulation time is 40 seconds. The angular set values of the manipülatör of the robot manipulator are entered differently for each joint. New control techniques have been developed for the inspection of robot manipulators. The most important element in robot manipulator control is to follow the desired trajectory. The robot control with ANN in simulations is given in Fig.3. As a ANN entry used in the control algorithm, the robot manipulator has been given the normalized opening ratios of the joints. ANN output is the control voltages applied to motors mounted for robot manipülatör. A 7 x 8 x 1 triple layer was used for the back propagation algorithm. The robot control outputs are shown in Fig.4 and 5.

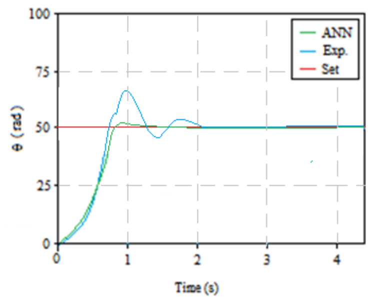


Figure 4. Simulation of ANN, Experimental and Reference Values

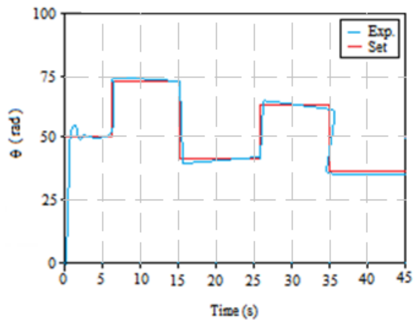


Figure 5. Simulation of Experimental and Reference Values

V. REFERENCES

[1]. Ayman A. Aly, "Intelligent Fuzzy Control for Antilock Brake System with Road- Surfaces Identifier," 2010 IEEE International Conference on Mechatronics and Automation (ICMA 2010) China, August 4-7, 2010.

[2]. Dhaval, V., Ohri, J., Ankit, P.: Comparison of Conventional & Fuzzy based Sliding Mode PID Controller for Robot Manipulator. In: IEEE International Conference on Individual andCollective Behaviors in Robotics, pp. 115-119 (2013)

[3]. Hala, B.N.D., Adel, M.A.: Fuzzy control of robot manipulators: some issues on design and rule base size reduction. Eng. Appl. Artif. Intell. 15, 401-416 (2002)

[4]. Kirby, R., Forlizzi, J., Simmons, R.: Affective social robots. Robot. Auton. Syst. 58, 322-332 (2010)

[5]. Kosmatopoulos E.B., Chassiakos A. And Christodoulo M.A., 'Robot Identification using dynamical neural networks."Engineering Systems with Intelligence. Kluwer Academic Publishers. 2005. pp.187-195.

[6]. Liu, J., Wang, X.: Sliding Mode Control for Robot. Springer, Berlin (2011) Meza, J.L., Santibanez, V., Soto, R., Llama, M.A.: Fuzzy self-tuning PID semiglobal regulator for robot manipulators. IEEE Trans. Ind. Electron. 59(6), 2709-2717 (2012)

[7]. Rasit, K., Abdullah, F.: Model based intelligent control of a 3-joint robotic manipulator: a simulation study using artificial neural networks. ISCIS, pp. 31-40. Springer, Berlin (2004)

[8]. Soltanpour, M.R., Jafaar, K., Soltani, M.: Robust nonlinear control of robot manipulator with uncertainties in kinematics, dynamics and actuator models. Int. J. Innovative Comput. Inf.Control 8, 5487-5498 (2012)