

Artificial Neural Network Based Position Control of Three-Degree of Freedom (3-DOF) Serial Robot

Tadele Tegege

Lecturer, Faculty of Electrical and Computer Engineering
Jimma Institute of Technology, Jimma University, Ethiopia

Abstract – Artificial neural network based PID controller for 3-DOF serial robot position control is designed and modelled for three-degree of freedom (3-dof) serial robot manipulator and to control a robot arm using PID and ANN controller to acquire the desired position. The necessary process where modelled before applying control techniques to guarantee the execution of any task according to a desired input with minimum error. Both forward and inverse kinematics where derived in robot modeling based on Denavit Hartenberg (DH) representation. PID controller robot taken consists of three rotational joints (3-RRR) driven by motors. The manipulator was forced to track accurately a prescribed trajectory. This is achieved through the application of modern control methods known as artificial intelligent controller. The problem of controlling the joints was more simplified by combining the ANN with PID controller methods and provides the network with more data about the structure and behaviour of the system.

Keywords – Robotics, Denavit-Hartenberg, Inverse kinematics, Artificial Neural Network.

I. INTRODUCTION

Robotic system is designed and developed to assist or replace a human in doing and accomplishing tasks that are boring, complex, too dangerous, and impossible for human. Kinematics of robot arm was mathematically modelled using a Denavit Hartenberg (D-H), [1], and [2] method. The forward kinematic model/equations, describe the functional relationship between the joint variables and the position and orientation of the end effectors.

To control any robot manipulator the core of the controller is a description of kinematic analysis, this is done by using a common method in industrial and academic research over the last several decades, research in robotic manipulators has focused mainly on designs that resemble the human arm.

A robot is a reprogrammable multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks [1]. Through this paper, the term robot manipulator is used.

Robot manipulator is one of the motivating disciplines in industrial and educational applications, and an essential branch to control sciences because of its intelligent aspects, nonlinear characteristics, and its real time implementation. It was developed to enhance human's work such as in the manufacturing or manipulation of heavy materials, and unpredictable environments. Whatever the kind of task robot manipulator may be

provided with, robot performance measures the high quality and large quantity of work that it can do in the desired time and place. Robot manipulator has immeasurable tasks, so it is designed to be flexible in general motions to move from one position to another with smooth movement to avoid sharp jolt in the robot arm. All commercial industrial robots have two physically separate basic elements the manipulator arm and the controller. There are three main subsystems in robot manipulators: mechanical system, electrical system, and control system [2].

Mechanical system comprises of all movable parts. It consists of a group of links (rigid bodies) connected together by joints which allow the motion for the desired link. The mechanical system is used to move the end effector (the top link) to xyz position with respect to the base. This movement depends on the electrical system (e.g. motors, power amplifiers, and other electronic circuits) and it is done by some rotations and translations to the other links.

II. KINEMATICS MODELING

To control any robot manipulator the core of the controller is a description of kinematic analysis, this is done by using a common method in industrial and academic research over the last several decades, research in robotic manipulators has focused mainly on designs that resemble the human arm. This kind of design has proven to be effective for many tasks; however, it may

present some limitations. Their structure is based on stiff links and large sections, which are passive supporting structures; the result is a heavy robot mechanism that is suitable for most industrial applications, where speed of operation and accuracy are essential. In spite of being very efficient for open environments, when constraints are added to the environment, the manipulator may fail to reach its desired end effectors position. This failure is due to the lack of degrees of freedom in the robot to meet both the environmental constraint conditions and the desired end effectors position requirements.

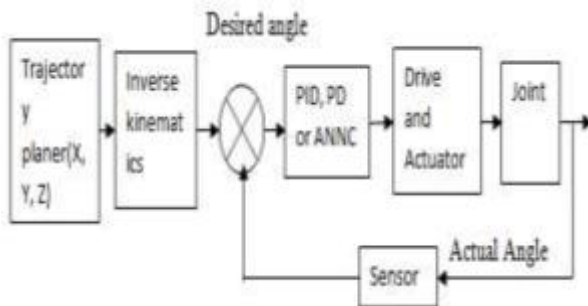


Fig .1. Block diagram of the system

2.1. Forward/Direct Kinematics

The forward kinematic equations, describe the functional relationship between the joint variables and the position and orientation of the end-effectors. Suppose the robot has i links, the joints and links numbered from 1 to i and 0 to I respectively [2].

The configuration space of the end effectors contains the transformation matrix T that relates the position and orientation of the end-effectors.

The following equation explains the forward kinematic problem. $F(\Theta_1, \Theta_2 \dots \Theta_n) = [x, y, z]$ (1)

Where Θ_1, Θ_2 and Θ_n are the input variables, $[x, y, z]$ are the desired position and R_d the desired rotation.

Consider a fixed frame $o_0 x_0 y_0 z_0$ and the rotation frame o_1, x_1, y_1, z_1 . The orientation is represented as a series of three revolute about a combination of the principal axes of the link frame. The rotations of the rotated frame about the fixed frame were represented by the three angles α, θ and ϕ . We consider to develop the rotation matrix representing a rotation of α angle about the Ox axis followed by a rotation of θ angle about the Oz axis followed by a rotation of ϕ angle about the Oy axis, by using rotational translation formula is;

$$\text{Rot}(x, \alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & C\alpha & -S\alpha \\ 0 & S\alpha & C\alpha \end{pmatrix} \quad (2)$$

$$\text{Rot}(y, \phi) = \begin{pmatrix} C\phi & 0 & S\phi \\ 0 & 1 & 0 \\ -S\phi & 0 & C\phi \end{pmatrix} \quad (3)$$

$$\text{Rot}(z, \theta) = \begin{pmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

$$\begin{pmatrix} C\phi & 0 & S\phi \\ 0 & 1 & 0 \\ S\phi & 0 & C\phi \end{pmatrix} \begin{pmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & C\alpha & -S\alpha \\ 0 & S\alpha & C\alpha \end{pmatrix} = \begin{pmatrix} C\phi C\theta & S\phi S\alpha - C\phi S\theta C\alpha & C\phi S\theta S\alpha + S\phi C\alpha \\ S\theta & C\theta C\alpha & -C\theta S\alpha \\ -S\phi C\theta & S\phi S\theta C\alpha + C\phi S\alpha & C\phi C\alpha - S\phi S\theta S\alpha \end{pmatrix} \quad (5)$$

Where $C\phi \equiv \cos\phi$; $S\phi \equiv \sin\phi$; $S\theta \equiv \sin\theta$; $C\theta \equiv \cos\theta$; $C\alpha \equiv \cos\alpha$; $S\alpha \equiv \sin\alpha$.

This 3 x 3 rotation matrix is used to describe the rotational operations of the body-attached frame with respect to the reference frame.

The homogeneous coordinates are then used represent position vectors in a three dimensional space, and the rotation matrices will be expanded to 4x4 homogeneous transformation matrices to include the translational operations of the body-attached coordinate frames. This matrix representation of a rigid mechanical link to describe the special geometry of a robot arm was first used by Denavit and Hertenberg [3].

The advantage of using the Denavit- Hertenberg representation of linkages is its algorithmic universality in deriving the kinematic equation of a robot arm.

1.The Denavit-Hartenberg principle

To describe the translational and rotational relationship between adjacent links, Denavit and Hartenberg proposed a matrix method of systematically establishing a coordinate system (body-attached frame) to each link of an articulated chain. An orthonormal Cartesian coordinate system (x_i, y_i, z_i) can be established for each link at its joint axis, where $i = 1, 2, 3, \dots, n$ (n = number of degrees of freedom) plus the base coordinate frame.

Since a rotary joint has only 1 degree of freedom, each coordinate frame of a robot arm corresponds to joint and is fixed in link. When the joint actuator activates joint, link will move with respect to link. Since the i th coordinate system is fixed in link, it moves together with

the link. Thus, the nth coordinate frame moves with the hand (link). The base coordinates are defined as the 0th coordinate frame () which is also the inertial coordinate frame (CF) of the robot arm. Thus, for a three axis robot arm, we have four CFs, namely, (x0, y0, z0). . . (x3 ,y3 , z3).

- a. The Zi-1 axis lies along the axis of motion of the ith joint.
- b. The Xi axis is normal to the Zi-1 axis, and pointing away from it.
- c. The Yi axis completes the right hand coordinate system as required.

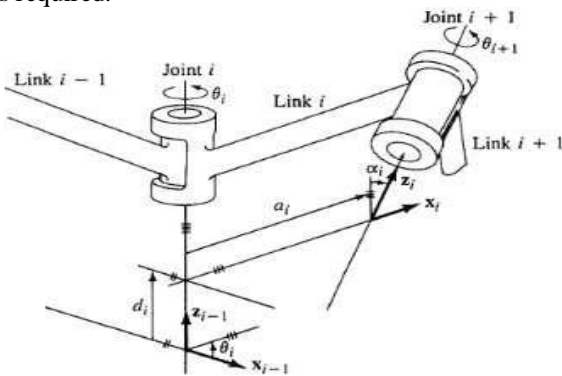


Fig. 2 . Link coordinates system and its parameters.

where Θ_i is the joint angle from the x_{i-1} axis to the x_i axis about the Z_{i-1} axis (using the right-hand rule) d_i is the distance from the origin of the (i -1)th coordinate frame to the intersection of the Z_{i-1} axis with x_i along the Z_{i-1} axis. a_i is the offset distance from the intersection of the Z_{i-1} axis with the x_i to the origin of the ith frame along the x_i axis (or the shortest distance between the Z_{i-1} and Z_i axis). α_i is the offset angle from the Z_{i-1} axis to the Z_i axis about the x_i axis (using the right-hand rule).

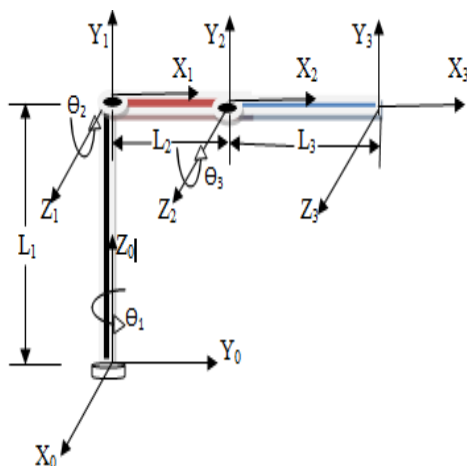


Fig.3 Kinematic diagram and frame assignment of 3-DOF of robot manipulator.

2. Obtaining the DH parameters

To describe the kinematics of any robot, four parameters are given for each link Θ_i , a_i , d_i , and α_i . Where two of them described the link, and the others describe the connection with other links.

Since the manipulator under consideration is a 3-dof with all the three joints are revolute, is the only variable and the rest variables are constant.

Table-I: DH parameter representation for

Joint i	Θ_i	A_i	$a_i(\text{cm})$	$d_i(\text{cm})$	Joint range
1	Θ_1	90°	$L_1=30$	$d_1=30$	-360° to 360°
2	Θ_2	0°	$L_2=17$	0	-90° to 90°
3	Θ_3	0°	$L_3=14$	0	0° to 150°

3.1 3-DOF serial manipulator

Obtain link transformation matrices A_i (A matrices)

After obtaining the table of DH convention, a series of homogeneous matrices can be derived depending on the number of the DOF. The transformation matrix for each joint from joint i to the joint can be calculated as:

$$A_i^{i-1} = Rot(z, \theta_i) Trans(z, d_i) Trans(x, a_i) Rot(x, \alpha_i) \quad (6)$$

3.2 Inverse Kinematics

Inverse kinematic problem may express mathematically as follows.

$$= \begin{bmatrix} C1C23 & -C1S23 & S1 & 14C1C23 + 17C1C2 \\ S1C23 & -S1S23 & -C1 & 14S1C23 + 17S1C2 \\ S23 & C23 & 0 & 14S23 + 17S2 + 30 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Where joint angles and represents the position and the orientation. Hence, we choose the geometric approach to find the inverse kinematics of 3-DOF manipulator.

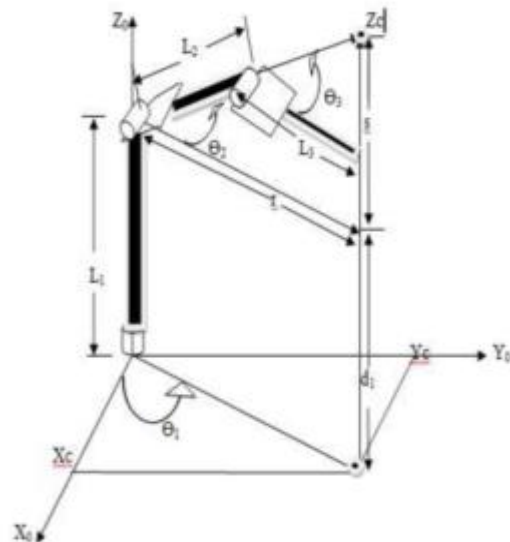


Fig. 4 3-DOF manipulator diagram.

Let the position of the end-effector be (Xc, Yc, Zc) meter as shown above. To find the joint variables $\Theta_1, \Theta_2, \Theta_3$

To obtain Θ_2 ;

$$V1 = \text{atan2}(s, r) \text{ and}$$

$$V2 = \text{atan2}(L3 * \sin(\Theta_3) * L2 + L3 * \cos(\Theta_3))$$

$$\Theta_2 = V_1 - V_2$$

$$= \text{atan2}(s, r) - \text{atan2}(L_3 \sin(\Theta_3), L_2 + L_3 \cos(\Theta_3))$$

(8)

This indicates, if the position and orientation of the end effectors are given then using inverse kinematics technique we can find the joint angle of the manipulator.

III. DC MOTOR MODELLING

The robots consist of joints and each one of these joints will have a motor to actuate the link to the desired position. There are two widespread types of joints on a manipulator. The first type is revolute (rotary), and it allows only relative rotation between two links. The second type of joints is called prismatic or sliding joint. This type of joints allows only linear relative motion between two links along its axis. From the equivalent circuit of controlled DC motor the motor shaft is coupled to a gear train to the load of the link and the total linear distance travelled on each gear is the same. That is,

$$dm = dL \quad \text{and} \quad rm \theta_m = rL \theta_L$$

(9)

Where r_m, r_L are respectively, the radii of the input gear and the output gear. Since the radius of the gear is proportional to the number of teeth it has, then

$$N_m \theta_m = N_L \theta_L$$

(10)

$$\frac{N_m}{N_L} = \frac{\theta_L}{\theta_m} = n$$

(11)

Where n is the gear ratio and it relates θ_L to θ_m by

$$\theta_L(t) = n\theta_m(t)$$

(12)

Taking the first two time derivatives, we have

$$\theta'_L(t) = n\theta'_m(t) \quad \text{where} \quad \theta'(t) = \frac{d\theta(t)}{dt}$$

(13)

$$\ddot{\theta}_L(t) = n\ddot{\theta}_m(t)$$

(14)

If a load is attached to the output gear, then the torque developed at the motor shaft is equal to the sum of the torques dissipated by the motor and its load. That is,

$$\tau(t) = \tau_m(t) + \tau_L^*(t)$$

(15)

The load torque referred to the load shaft I

$$\tau(t) = \tau_{LL}(t) + \tau_{fLL}(t)$$

(16)

And the motor torque referred to the motor shaft is $\tau_m(t) = J_m \ddot{\theta}_m(t) + f_m \dot{\theta}_m(t)$

(17)

recalling that conservation of work requires that the work done by the load referred to the load shaft, $\tau_L \theta_L$, be equal to the work done by the load referred to the motor shaft, $\tau_L^* \theta_m$, lead to

$$\tau_L^*(t) = \frac{\tau_L(t) \theta_L(t)}{\theta_m(t)} = n\tau_L(t)$$

(18)

Using the above equation, we have,

$$\tau_L^*(t) = n^2 [J_L \ddot{\theta}_L(t) + f_L \dot{\theta}_L(t)]$$

(19)

From eq(7) and (9), the torque developed at the motor shaft is

$$\tau(t) = \tau_m(t) + \tau_L^*(t) = (J_m + n^2 J_L) \ddot{\theta}_m(t) + (f_m + n^2 f_L) \dot{\theta}_m(t)$$

(20)

Where $J_{eff} = J_m + n^2 J_L$ is the effective moment of inertia of the combine motor and load referred to the motor shaft and $f_{eff} = f_m + n^2 f_L$ is the effective viscous friction coefficient of the combined motor and load referred to the motor shaft.

Based on the above results we can now derive the transfer function of this single joint manipulator system.

Since the torque developed at the motor shaft increase linearly with the armature current, independent of speed and angular position, we have

$$\tau(t) = K_a i_a(t)$$

(21)

Where K_a is known as the motor-torque proportional constant in N.m/A. applying the Kirchhoff's voltage law to the armature circuit, we have

$$V_a(t) = R_a i_a(t) + L_a \dot{i}_a(t) + e_b(t)$$

(22)

Where e_b is Back electromotive force (emf) which is proportional to the angular velocity of the motor,

$$e_b(t) = K_b \dot{\theta}_m(t)$$

(23)

And K_b is a proportional constant in V.s/rad. taking the Laplace transform of the above equation and solving for $I_a(s)$, we have

$$I_a(s) = \frac{V_a(s) - sK_b \theta(s)}{R_a + sL_a}$$

(24)

Taking the Laplace transform of eq (21)

$$T(s) = s^2 J_{eff} \theta_m(s) + s f_{eff} \theta_m(s)$$

(25)

Taking the Laplace transform of eq (22)

$$T(s) = K_a I_a(s) = K_a \left[\frac{V_a(s) - sK_b \theta_m(s)}{R_a + sL_a} \right]$$

(26)

Equating the above two Laplace transform equation and rearrange, we obtain the transfer function from the armature voltage to angular displacement of the motor shaft,

$$\frac{\theta_m(s)}{V_a(s)} = \frac{K_a}{s^2 J_{eff} L_a + (L_a f_{eff} + R_a J_{eff})s + R_a f_{eff} + K_a K_b}$$

(27)

Since the electrical time constant of the motor is much smaller than the mechanical time constant, we neglect the armature inductance effect, L_a . This allows us to simplify the above equation to

$$\frac{\theta_m(s)}{V_a(s)} = \frac{K_a}{s[s R_a J_{eff} + R_a f_{eff} + K_a K_b]}$$

(28)

Since the output of the control system is the angular displacement of the joint $\theta_L(s)$ using eq(12) and its Laplace transformed equivalence, we relate the angular position of the joint $\theta_L(s)$ to $V_a(s)$,

$$\frac{\theta_L(s)}{V_a(s)} = \frac{n K_a}{s[s R_a J_{eff} + R_a f_{eff} + K_a K_b]}$$

(29)

This is the general transfer function of "single joint" manipulator relating the applied voltage to the angular displacement of the joint.

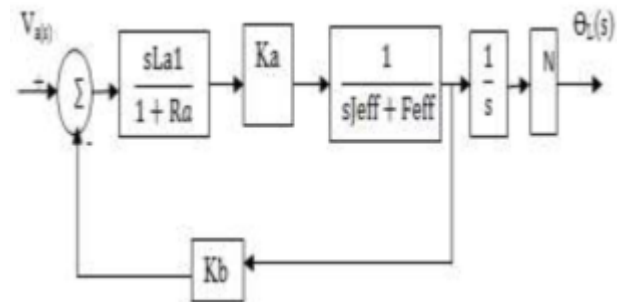


Fig .5 .Block diagram of a single joint robot actuator.

Table-II: Initial Motor Choice Parameters

Parameter	Desired value	Selected
Voltage	12V	12V
Torque	0.425oz-in	0.425oz-in
Speed	5000rpm	5187rpm

Given these parameters, it is unlikely that a standard catalogue motor will fulfil all parameters simultaneously since they are not independent. The selection process consists of finding the best fit. As a first step, determine the required output power as follows:

$$PO = \frac{N * M}{1350} \quad (30)$$

Where is a unit's conversion factor

The motor should be rated at least 1.5 to 2 times the desired output power in relation to its maximum output power (at nominal voltage). A motor with approximately 2.4 to 3.2 Watts maximum output power should suffice. Referring to the Micro MO catalogue, the smallest motor to achieve this power rating is the 1331 series (13mm diameter x 31mm long). The 16 Volt version is closest to the desired operating voltage (12volts). To find the no-load speed, a good first approximation is simply to ratio the voltages and speeds.

$$N_{12} = \frac{12}{16} * NO = \frac{12}{16} * 11400 = 8850 \text{ rpm} \quad (31)$$

Since our desired speed is 5,000 rpm, this represents only 58.5% of the no-load speed (at 16 Volts). Unless size is of paramount importance, this motor (1331T024S) is not a great choice even though it can provide the power required. The next selection from the catalogue which meets the power requirements is the series 2230 motor (22mm diameter and 30mm long) and the selection would again be the 16 Volt.

Table- III: DC motor parameter and values.

Parameter	Values
No load speed	9000rpm
Armature resistor	1 ohm
Inductance	0.05 Mh
Viscous friction	0.00003N.m .s ² /rad
Momentum inertia	0.01Kg.m
No load current	4.04A
No load power	2.88W
Motor constant(Ka,Kb)	0.023 N.m/A

The approximate no-load speed at 12 Volts would be

The current through the motor will be the sum of the load current and the no-load current, where:

$$I = I + I_{IO} \quad (32)$$

Since, $I = M/km$

$$I + I_{IO} = 4.04 = 4.16A$$

Where I_M is current through the motor, I is current due to load, M is desired torque, I_{IO} is no-load current.

To calculate the speed at the desired load torque, the following formula applies:

$$N = NO \text{ at } 12v - 1350 * \frac{I_M * R}{KM} \quad (33)$$

As this speed value is close to the desired value, the selection of this motor appears reasonable. Three identical motors are used to actuate each link independently and its specification is:

Three identical motors are used to actuate each link independently and its specification is:

$$\begin{aligned} V_a &= 12V, R_a = 1\text{ohm}, L_a \\ &= 0.05\text{mH}, J_m \\ &= 0.01\text{Kg.m}^2, \end{aligned}$$

$$\text{Full load speed } (N_f) = 5187\text{rps}, f_m = 0.00003\text{N.m.s}^2/\text{rad}, K_a = k_b = 0.023 \text{ N.m/A}$$

The required speed of the joint is taken to be 104 rps.

This is low speed which enables to use independent PID controller for each joint since the coupling effect of one link on the other can be neglected for slow motion of robots.

Hence, the gear ratio can be obtained from eq(13) which is

$$n = 104/5187 = 0.02$$

This gear ratio highly minimizes the effect of the load inertia on the motor.

Because of this and the motor specification is the same for each joint, the transfer function of each joint is the same. Substituting the constant values of the motor and joint parameter into eq (30), the transfer function for each joint becomes:

$$\begin{aligned} \frac{\Theta L(s)}{V_a(s)} &= \frac{nka}{S[sR_a J_{eff} + R_{a\text{eff}} + k_a k_b]} \\ &= \frac{0.02 * 0.023}{S[s * 1 * 0.01 + 1 * 0.00003 + 0.023 * 0.023]} \quad (34) \end{aligned}$$

Eq(33) is the general transfer function of a joint

IV. TRAJECTORY MODELLING

A trajectory is the path followed by the manipulator, plus the time profile along the path and it can be planned either in joint space (directly specifying the time evolution of the joint angles) or in Cartesian space (specifying the position and orientation of the end frame).

1. Trajectory modelling in Cartesian space

Cartesian-space trajectories relate to the motions of a robot relative to the Cartesian reference frame. In Cartesian-space, the joint values must be repeatedly calculated through the inverse kinematic equations of the

robot. In this thesis continues path particularly „circular trajectory“ is taken as sample and planned using Cartesian trajectory planning technique since the trajectory is continues and simple to visualize.

The robot needs to draw a circle with 0.07m radius and center at (0.2, 0, 0.3) point starting from a (0.2, 0, 0.37) as shown below. The position of the center of the circle and its radius is chosen to be within the workspace of the manipulator.

The circle is made to lie on the x-z plane.

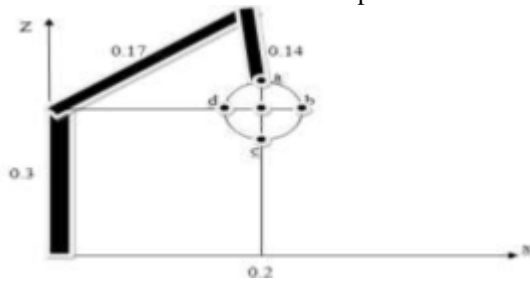


Fig.6 Manipulator home position.

The time required by the manipulator to finish drawing the circle is taken to be 10sec. Taking four sample points a, b, c and d on the circle, it is possible to get the time based relation of x and z components of any point on the circle.

Table –IV: Values of the end effector on the circle (x and z) with respect to time.

t(sec)	0	2.5	5	7.5	10
X in m	0.2	0.27	0.2	0.13	0.2
Z in m	0.37	0.23	0.23	0.3	0.37

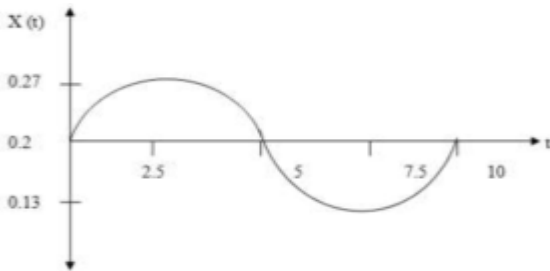


Fig.7 . Graph of x(t) versus t

This shows that

$$X(t) = 0.2 + 0.07 \sin(2 * \pi * t / 10) \quad (35)$$

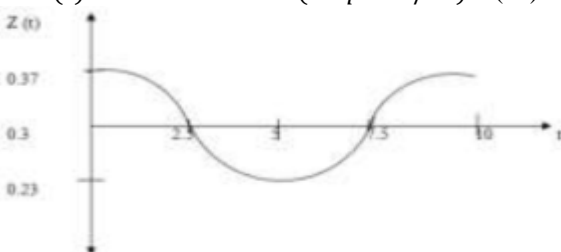


Fig .8. Graph of z (t)

This show that

$$Z(t) = 0.3 + 0.07 * \cos(2 * \pi * t / 10) \quad (36)$$

Now we can check back using the following Mat-lab code.

```
>>Plot(x, z)
```

Executing this Mat-lab code generates the figure shown below.

The code clearly shows that the trajectory given by equations is perfect circle.

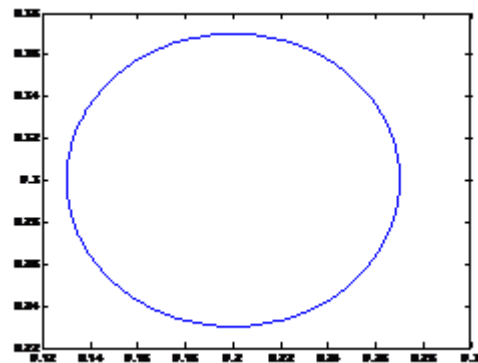


Fig. 9 Graph of x (t) versus z (t) using mat lab.

Therefore, the trajectory for a circle on x-z plane with r meter radius, (Xc, Yc, Zc) meter center and for T sec. to finish drawing the circle starting from the point p (Xc, 0, Zc+r) on the circle is given by;

V. CONTROLLER DESIGN

Design of controller is very important to control and adjust robot manipulator. For robotics PID controller has relatively better than the others because it combines the advantage of both PI and PD controller that is improve the steady – state part and improve the transient part respectively. Although PID is better performance than that of the above controller type, it has some drawback for non-linear system.

Obviously a robotic system has non linearity behaviour due to gear backlash, load variations and other parameters that have unpredictable nature. In order to overcome this problem using intelligent control in accordance with PID controller is the best approach. Thus, designing and implementing position controller for three degree of freedom (3-DOF) serial robot manipulator is the main task of this research.

1.PID controller design

PID algorithm, assumes the controlled plant is known, such as a robot manipulator. The error signal, e (t), is the difference between the set point, and the process output. If the error between the output and the input values is

large, then large input signal is applied to the physical system. If the error is small, a small input signal is used. As its name suggested, any change in the control signal, $u(t)$ is directly proportional to change in the error signal for a given proportional gain K_p .

Mathematically the output of the PID controller is given

$$U(t) = \frac{K_p e(t) + K_d \frac{de(t)}{dt} + K_i \int_0^t e(t)}{n} \quad (37)$$

Where n is the gear ratio.

The transfer function of the PID controller in parallel is

$$G_{PID}(s) = \frac{K_d s^2 + K_p s + K_i}{s} \quad (38)$$

Where T_i and T_d are the integral and derivative times constant respectively.

Therefore, adjustment of these parameters is a good solution to acquire the optimum values for the desired control response. As mentioned, the PID tuning is an important issue, and it is concerned with the best selection of the three parameters, so an acceptable performance of the control loop is established. Thus the tuned value of, using iterative method is tabulated as below. Thus the tuned value of using iterative method is tabulated in table .4 below which shows tuned values of PID gain for the system specification of.

Table-V: Tuned values of PID gain

Iteration	Kd	Ki	Kp	Ts	%ov
1	2.3	1	10	0.0200	5.4772e-004
2	80	1	15	0.0200	7.1079e-004
3	90	1	20	0.0150	7.9582e-004
4	100	1	20	0.0150	6.2196e-004
5	100	30	20	5	229.0168
6	20.32	2.3	58.8	0.43	0.0465
7	27.32	2.3	58.8	0.0279	0.028
8	58.32	21.3	58.8	0.025	0.0068
9	65.93	27.3	59.98	0.02	0.0055

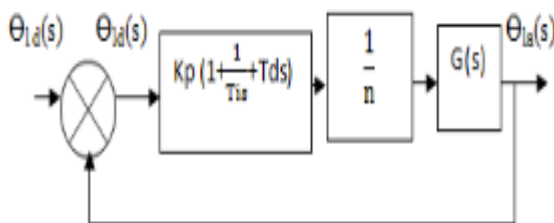


Fig.10. PID Controller Block Diagram.

From eqn (30) $G(s)$

The closed loop transfer function is

$$\frac{\theta_{LA}(s)}{\theta_{Ld}(s)} = \frac{K_p (1 + \frac{1}{T_i s} + T_d s) G(s)/n}{1 + K_p (1 + \frac{1}{T_i s} + T_d s) G(s)/n} \quad (39)$$

2. PD Controller Design

The basic aim of PD controller design is for training the Artificial Neural Network i.e. the input of ANNC are $e(t), de(t)/dt$ and bias form PD controller and also the output ANNC is connected with the output of PD controller. From equation (30), the closed loop transfer function of PD:

$$\frac{G_{pd}(S)}{1+G_{pd}(S)} = \frac{K_a K_v s + K_a K_p}{S^2 R_a J_{eff} + S(R_{a_{eff}} + K_a K_v + K_a K_b) + K_a K_p} \quad (40)$$

From the specification $T_s=0.5$ second; %OV5% this implies

$$\%S = \exp(-\pi \xi / (\sqrt{1 - \xi^2})), \quad \xi \leq 0.69 \quad (41)$$

$$\omega_n = \frac{4}{\xi T_s} \quad (42)$$

From this mathematical equation, $\omega_n = 11.5942$ rad/sec²

$$(11.5942 \text{ rad/sec})^2 = \frac{0.023 K_p}{0.01 * 1} \quad (43)$$

The velocity feedback gain K_v can be found as;

$$K_v = \frac{2\sqrt{K_a K_p J_{eff} R_a}}{K_a} - \frac{(R_{a_{eff}} + K_a K_b)}{K_a} \quad \text{and}$$

Thus PD controller transfer function is

$$\frac{U(s)}{E(s)} = 58.45 + 6.9329s \quad (44)$$

This PD controller is used to train the artificial neural network.

3. Intelligent Controller Design

The ANN controller enables the robot to act as human being in case of its learning, adaption, decision making behaviours and tackling the nonlinearity of the robotic system. Therefore, robots with this type of advanced controller are highly accurate, precise and cost effective in terms of power consumption and using one robot for multipurpose rather than constructing one automated system for each task.

Due to this, they are applicable in manufacturing industries for various machine loading or unloading, processing operations, and assembly and inspection. The ANN controller understands the system parameters by its self and the structure of controller can easily be changed to improve the performance of robot. Error between the desired trajectory path and the path of the robot converges to zero rapidly and as the robot performs its tasks the controller learns the robot parameters and generates better control signal. The performance of controller is tested in

simulation and on a real manipulator with satisfactory results.

Artificial Neural Network controller (ANNC) is used instead of PD or PID controller for position control of 3-dof manipulator. This is due to the fact that ANNC can be applied Very successfully in the identification and control of non-linear systems.

The universal approximation capabilities of the multilayer perception make it a popular choice for modelling nonlinear systems and for implementing general-purpose nonlinear controllers. In designing ANNC it is required to determine the number of input, hidden and output layers.

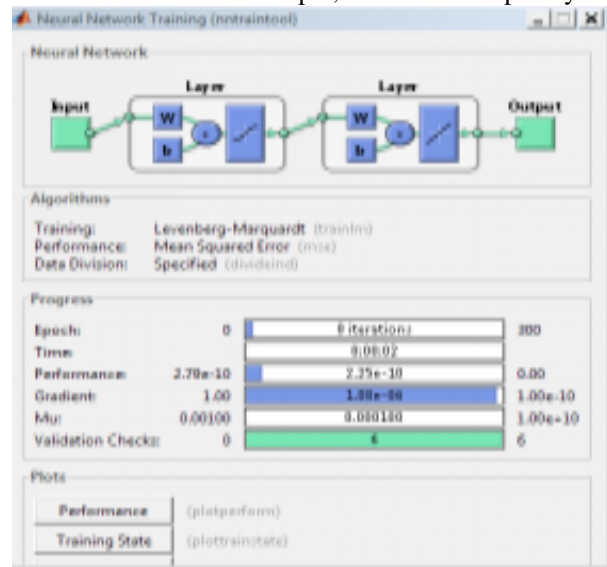


Fig. 11. Neural Network Training.

The number of these three layers is chosen as follows. Number of input layers =3.

These are Error, change in error and bias

Number of hidden layers=6. Number of output layers=1.

It is the output of the PD controller. The neural controller is trained from the input and output data of the PD controller.

VI. SIMULATION RESULT AND DISCUSSION

For studying the performance of the proposed intelligent position controller for 3-DOF serial robot using MATLAB/ Simulink has been done on the PID of Dc motor. MATLAB/Simulink is used to implement the developed mathematical model equations into suitable Simulink block and observation and testing of the system were implemented on it.

1. ANN based position control of 3-DOF serial robot Simulink Model

The overall Simulink model of ANN based position control of 3-DOF serial robot was implemented using MATLAB/Simulink software.

The Simulink block contains the PID model block which is given in (39), PD model block which is given in (40) and ANN model block, which consists of three layers.

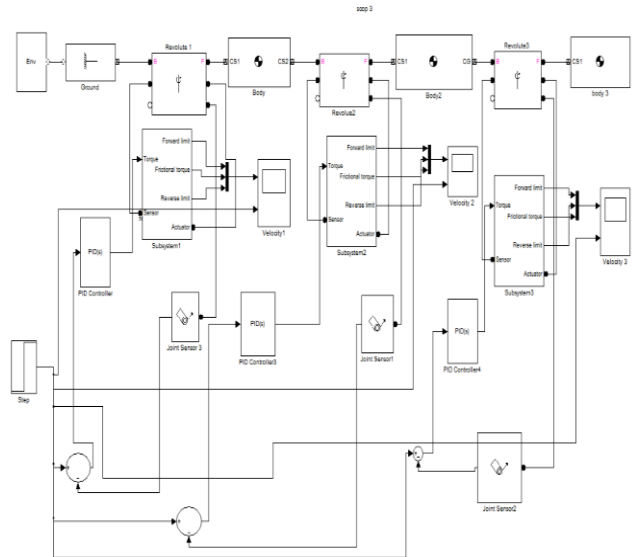


Fig.12 General Simulink model of the proposed system.

2. Simulation Result of PD, PID and ANN based position control of 3- DOF serial robot

Simulation has been carried out for different operating conditions parameter variation of the motor drive to study the dynamic performance of the ANN position control.

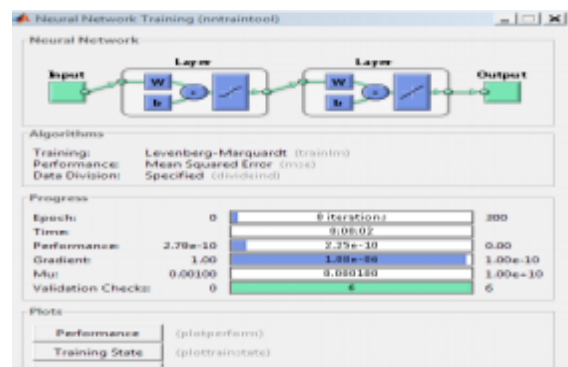


Fig .13. step response of the system with PD
From the output observed it is calculated that;

From this, $\xi=0.37$ and hence $T_s=4.6/\xi\omega_n=1.07s$ with high steady state error.

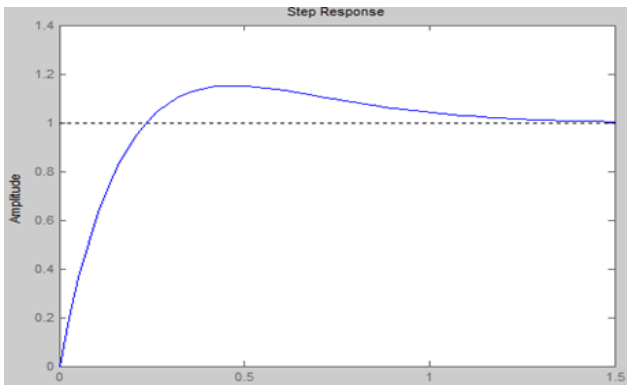


Fig. 14. PID simulation result From the figure,

From the graph= 0.46 , $T_s=4.6/0.86s$ with small steady error. To see the simulation result of ANN it should be trained first with number given layers.

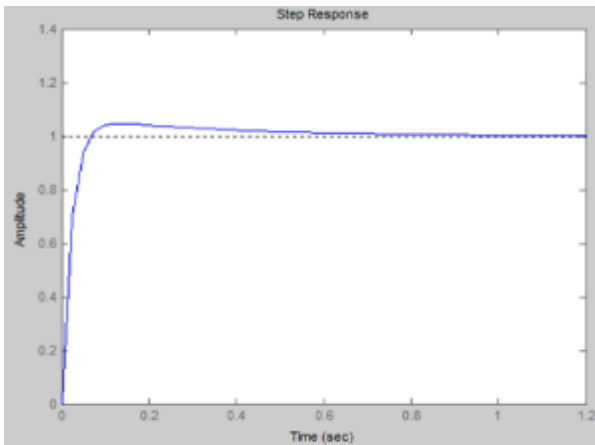


Fig. 15. Reference model input and output of ANNC.

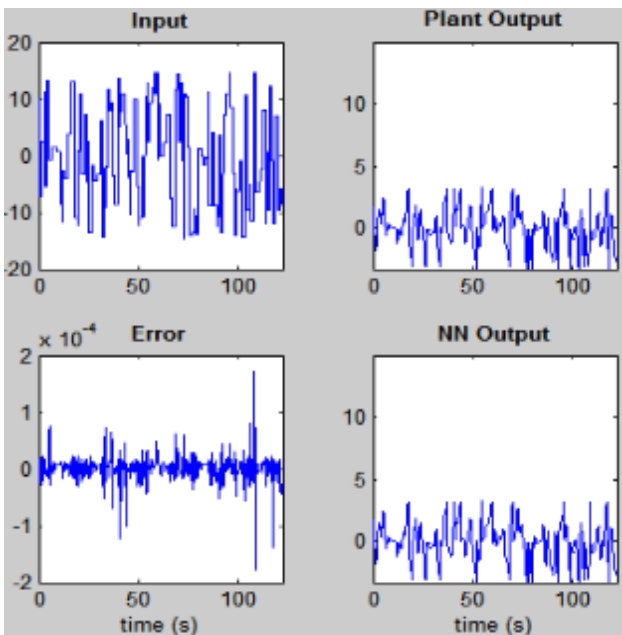


Fig. 16. ANNC simulation result From the output obtained, %s is approximately 0.0513.

From this $=0.68, =0.51s$ with high steady state error.

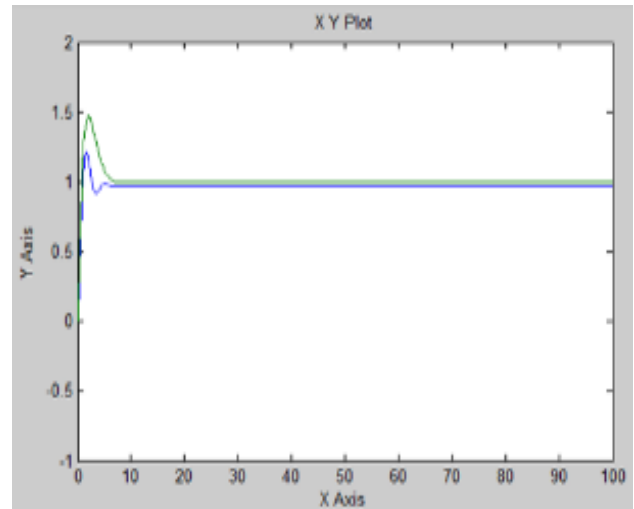


Fig.17. Simulation result of ANNC and PID on the X, Y graph.

From the graph the green one shows PID and the black one shows the ANN controllers. From the result implementing PID with ANNC increased the accuracy or the performance of the system because of the combined effect of their advantage by overcoming their individual drawbacks.

Table-VI Comparison of PD, PID and the ANNC in terms of overshoot transient response and steady state error.

Controller	Rise time (sec)	Peak overshoot (%)	Settling time (sec)	Steady state error
PD	0.42	20	1	0.06
PID	0.35	5.8	0.86	0.051
ANNC	0.31	0.0513	0.51	0.02

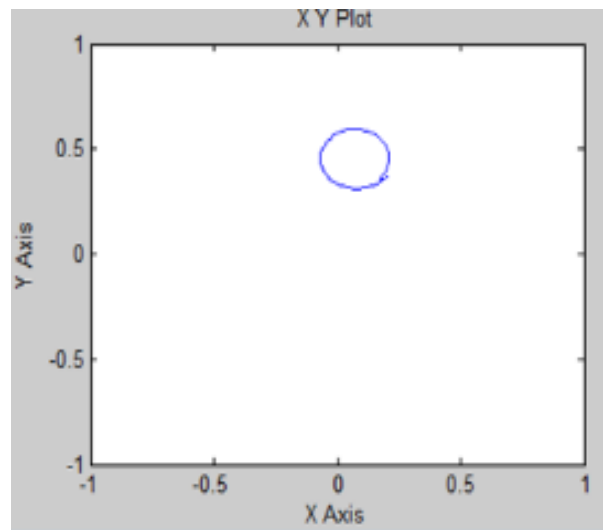


Fig.18 .Simulink result of the manipulator.

The manipulator result shows the expected idea that the robot needs to draw a circle with a given position and orientation of the robot manipulator within specified reference frame. The position of the center of the circle and its radius is chosen to be within the workspace of the manipulator. The circle is made to lie on the x-z plane.

VII. CONCLUSION AND FUTURE WORK

Robotics has become recently an interesting area of research. In this study of the robot Manipulator was done from two sides: modelling and controlling. Modelling process includes kinematic analysis and DC motor modelling. This process is important before controlling the robot to save the robot from being damaged. Applying a control technique is important to guarantee high efficiency and lower error for the motion of the robot. The desired tasks were accomplished using three stages: the first stage was to provide systematic rules for analysing forward and inverse kinematics solutions for the robot manipulator with revolute joints using DH parameters, secondly PID controller was designed. Thirdly PD with ANNC were designed and considered as a second choice to control the robot.

The comparison of the PD, PID and the ANNC techniques in terms of overshoot, transient response and steady state error were done. All simulations were presented using MATLAB and SIMULINK, which are used widely in control applications. The objective of this research was to control 3-DOF robot arm to reach the specified location with minimum error while meeting certain specification. The tracking path from the initial position to the final position was considered in this thesis; we set the continuous position for each motor used independent joint control method.

This research is used the PID controller to compare its results with ANNC. The system was model of a 3-DOF, which means it has 3 motors to control their positions independently. Tuning of PID parameters was accomplished by iterative or manual method. The simulations and numerical results of the controllers are presented in this paper. It is proved that ANNC is more efficient in the time response behaviour than the PID and PD controller. As a future work it is possible to extend to the controller for more than 3-OF serial robot and it will be designed and implemented for high DOF serial robot and will be built on the PCB.

REFERENCES

- [1]. J.J. Craig, Introduction to Robotics Mechanics and Control, 3rd Edition, Prentice Hall, 2005.
- [2]. K. H. Kalil, Nonlinear Systems, 3rd Edition, Prentice-Hall, 2002.
- [3]. R. Gorez, "A Survey of PID Auto-Tuning Methods," Journal A, Vol. 38, No. 1, pp. 3–10, 1997
- [4]. J.G. Ziegler and N.B. Nichols, "Optimum Settings for Automatic Controllers," Transaction American Society of Mechanical Engineering., Vol. 64, pp. 759–768, 1942
- [5]. S.N. Sivanandam, S. Sumathi and S.N. Deepa, Introduction to ANNC Logic Using MATLAB, Springer, pp. 97–107, 2007.
- [6]. G.K.I. Mann, B.G. Hu and R.G. Gosine, "A systematic study of PID
- [7]. Controllers function-based evaluation approach," IEEE Transactions on PID System., Vol. 9, No. 5, pp. 699–712, Oct. 2001
- [8]. R. Ketata, D. De Geest, and A. Titli, "PID controllers: Design, evaluation, Parallel, and hierarchical combination with a PID controllers," Fuzzy Sets System. Vol. 71, No. 1, pp. 113–129, Apr. 1995.