Chapter7

Controller Design and Response

7.1 Introduction

There are many controllers that are good to make the control of the project, like PID, PID with tracker controller and other controllers.

In this project we use tracker controller to control the project because this controller can deal with more than 2-dof. See Figure 7.1

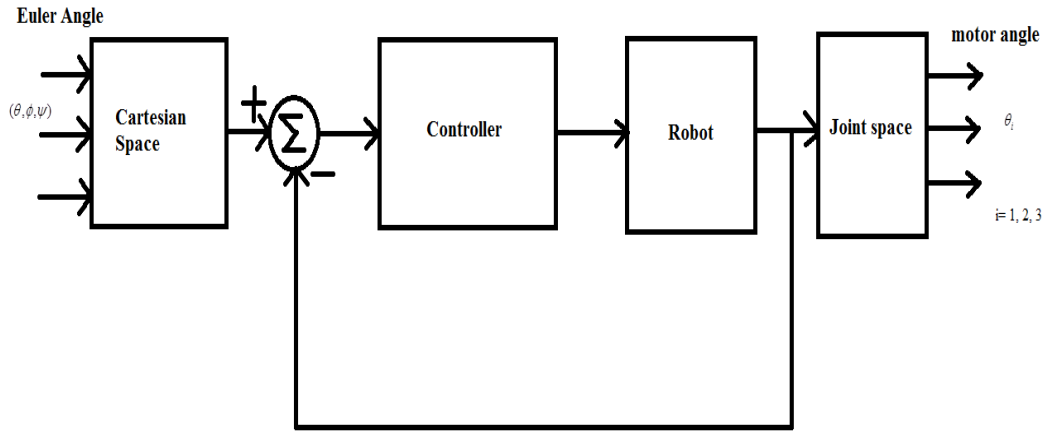

Figure 7.1 General block diagram for the control

7.2 Controller Design

Before taking any reading from the encoder, the Kpm must found for each motor.

Where Kpm is the number of pulses per revolution

The first step in this work is how to find the parameter of the controller of each motor, using MATLAB software.

The steps for finding the parameters:

1) Finding the transfer function of each motor

To find the transfer function for the motor; an encoder or a sensor is needed to give the position of the motor, then the arduino microcontroller is used to obtain the result.

In this stage we connect the motors and encoders with Arduino microcontroller, then the code is installed to microcontroller, we give the motors a simple rotation. The Arduino is give a reading of pulses (number of pulses) .We take these number to matlab and plot with time .The graph is first order , relation of voltage with speed .We take the integral of the relation that .we have to get another relation of voltage with position

$$G = \frac{14.95}{s(s+4.27)} = \frac{\theta}{v}$$
$$(s^2 + 4.27s)\theta = 14.95v \qquad (7.1)$$

2) Find the state space model of each motor

Take the Laplace inverse of equation (1.1) that give:

$$\ddot{x} + 4.27\dot{x} = 14.95v$$
$$x_1 = x$$
$$x_2 = \dot{x} = \dot{x}_1 \qquad (7.2)$$
Then
$$\dot{x}_2 = \ddot{x} = 14.95v - 4.27x_2$$

From equation (1.2) the state space representation can be calculate:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$
$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & -4.27 \end{bmatrix} x + \begin{bmatrix} 0 \\ 14.95 \end{bmatrix} v$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

The input u is the voltage (v), and the feed forward matrix D is equal to zero.

Now convert the matrix above to discrete matrix, A to Ad, B to Bd and C to Cd, Ts= 0.006 sec, then by using this relation to convert the matrix the result is:

$$A_d = I + T_s * \mathrm{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + 0.006 \begin{bmatrix} 0 & 1 \\ 0 & -4.27 \end{bmatrix} = \begin{bmatrix} 1 & 0.006 \\ 0 & 0.02562 \end{bmatrix} \qquad (7.3)$$

$$B_d = T_s * B = 0.006 \begin{bmatrix} 0 \\ 14.95 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.0897 \end{bmatrix} \qquad (7.4)$$

.(Then the extended system matrix was calculated as in equation (7.5
The matlab file is in appendix A.

$$A_e = [A_d \quad \text{zeros}(2,1); -C \quad 0] = \begin{bmatrix} 1 & 0.006 & 0 \\ 0 & 0.0256 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$
$$B_e = [B_d; 0] = \begin{bmatrix} 0 \\ 0.0897 \\ 0 \end{bmatrix} \qquad (7.5)$$

3) Now the optimal method (Quadratic optimal regulator) used to find the values of the gain.

After applying these steps in the matlab, the controller gain of each motor is:

For first motor: Kp= 4.2071 , Kd = 0.024254 , KI= 10

For second motor: Kp= 4.2071 , Kd= 0.024254 , KI= 10

For third motor: Kp= 5.5 , Kd=0.06, KI= 7

Kp= proportional gain, Kd= derivative gain

Then this value of gain is used in the code of microcontroller.

4) Experimental result

The experiment starts by giving a specific angle for each motor then the angle was checked

To check this angle the code is install on the arduino and the result angle will appear on the screen.

7.3 controller used

The digital tracker controller is used to control the robot. Figure 7.2 show .the simulation block of the tracker

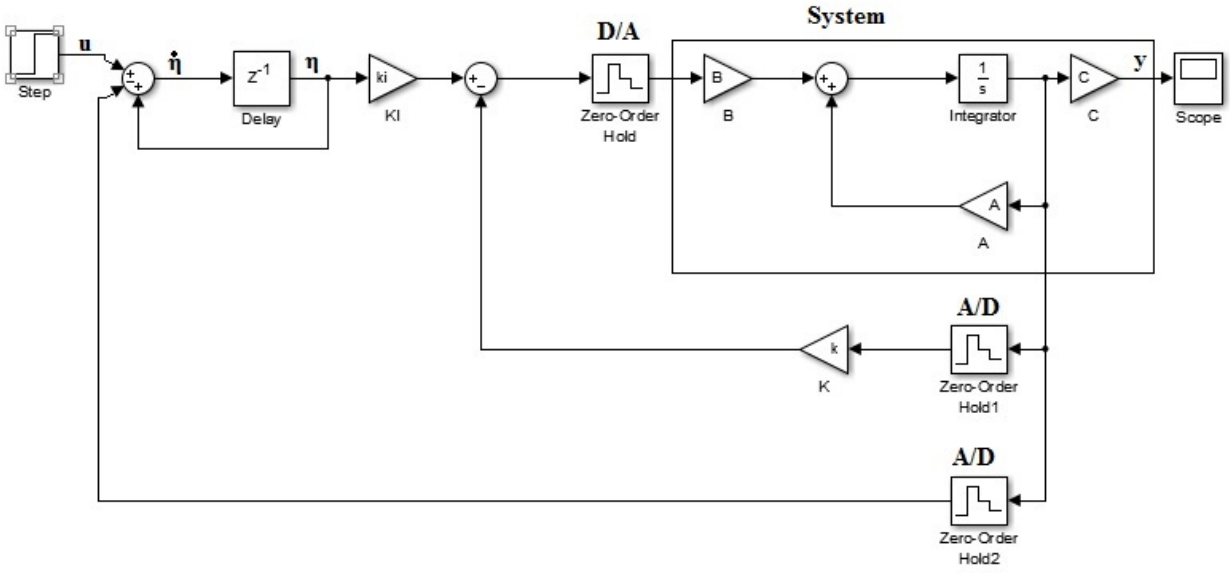

Figure 7.2: Tracker Simulink model

$$\dot{\eta} = (\mathrm{r} - \mathrm{y})\,\mathrm{T} + \eta \qquad (0.6)$$

$$\eta_{(\mathrm{k}+1)} = (\mathrm{r} - \mathrm{y})\,\mathrm{T}_{ss} + \eta \qquad (0.7)$$

Where T_{ss} settling time

7.4Expermental Result

This chapter discuss the response of the motor angles in terms of peak time, overshoot and steady state error .The plotting is exucuted using MATLAB.

The control of the robot is in the joint space and kinemtaic level. The forces and the dynamics of the robot were not insidered in this work. The motion in the joint space is generated based on the otienation of the eneffector through the inverse kinematics model.

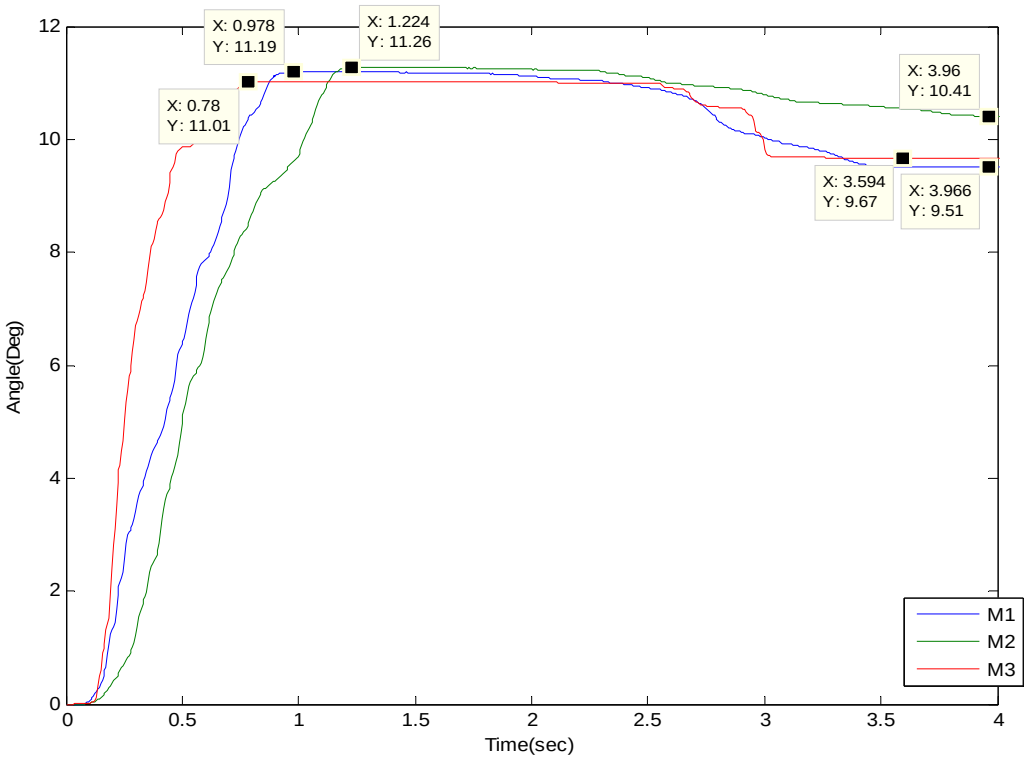

Figure 7.3 Response of 3 motors @ angle 10^0

For motor 1:

$$\%OS = 15.01\%$$

$$Tp = 0.978s$$

For motor 2:

$$\%OS = 10.41\%$$
$$Tp = 1.224s$$

For motor 3:

$$\%OS = 12.17\%$$
$$Tp = 0.78s$$

.

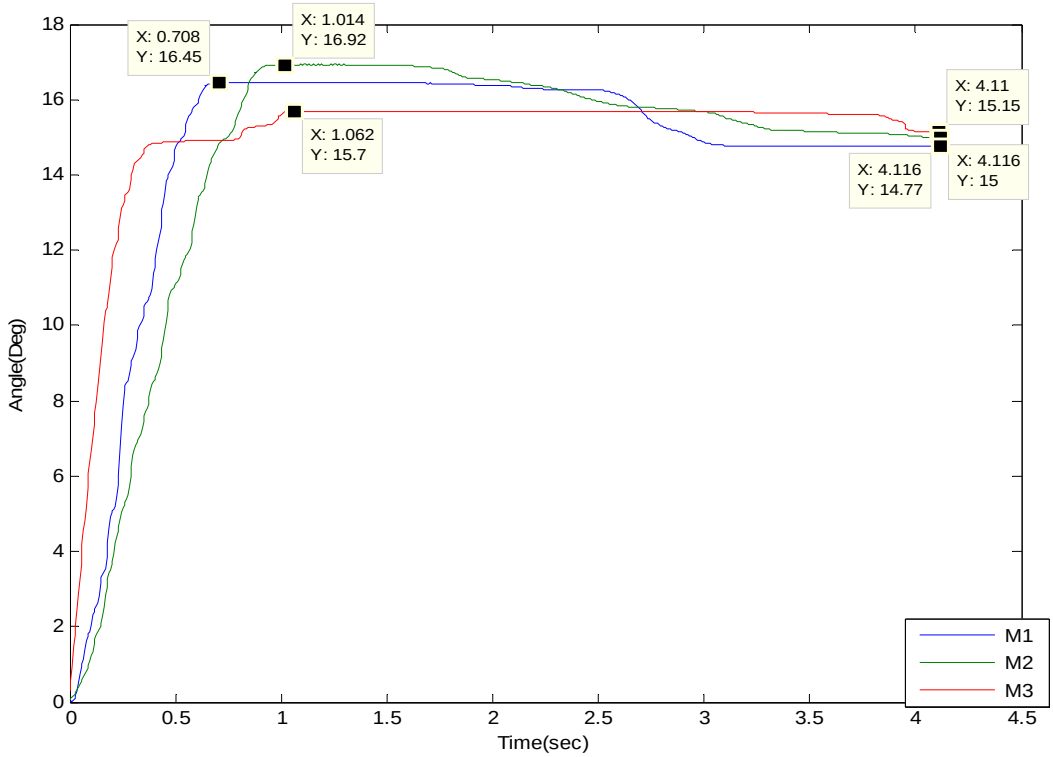

Figure7.4 Response of 3 motors @ angle 15^0

For motor 1:

$$\%OS = 11.62\%$$
$$Tp = 0.708s$$

For motor 2:

$$\%OS = 11.34\%$$
$$Tp = 1.014s$$

For motor 3:

$$\%OS = 3.5\%$$
$$Tp = 1.062s$$

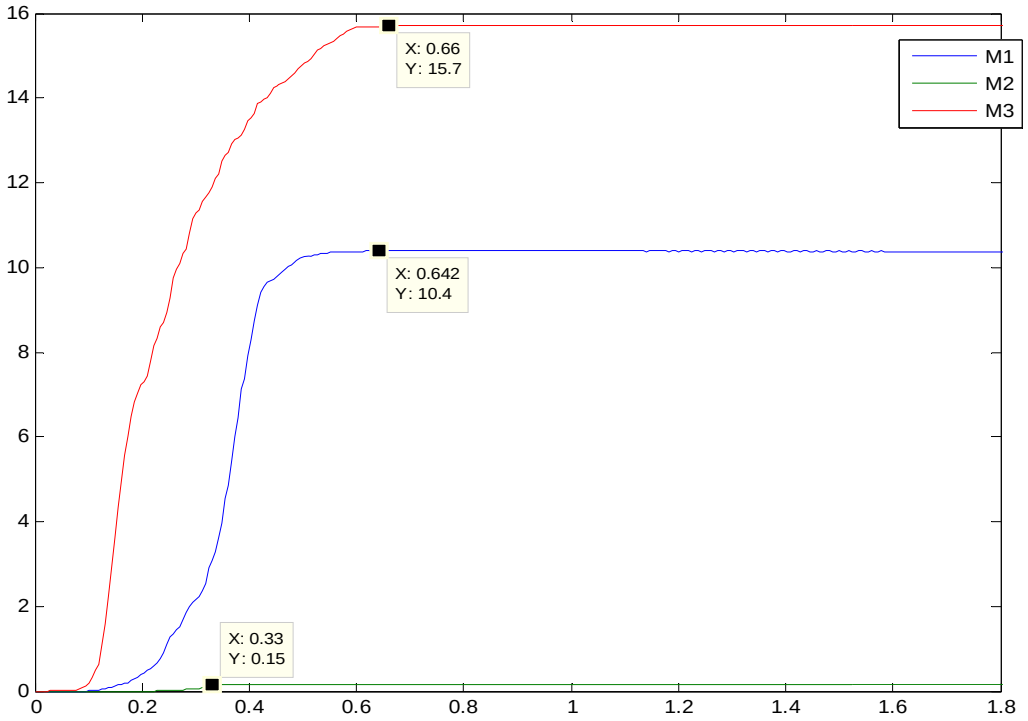

Figure 7.5 Response of the 3 motors @ different angles

For motor 1:

$$\%error = 3.86\%$$
$$Tp = 0.662s$$

For motor 2:

$$\%error = 33.33\%$$
$$Tp = 0.33s$$

For motor 3:

$$\%error = 4.45\%$$
$$Tp = 0.642s$$

From figures above its notice that the inverse kinematics was used to decide the angle that we must entered to the robot. For example if we need end effecter to reach an angles: α, β, γ

By invers kinematics calculations the input angles: $\theta_1, \theta_2, \theta_3$

Since the same controller was used in different motors we have two different responses, however to solve this problem the parameters must be tuned for the two motors to get the same responses. Backlash between gears and deflection on motor shaft affect to response on motor 3.

On the other hand the overall real response represent a high overshoot compared to the simulated one because of sampling time was too high.