TABLE OF CONTENTS

Abstract

This study examined the control of a planar two-link robot arm. We first derive the equations of motion for the two-link robot manipulator using the Lagrangian approach. These equations are described by nonlinear system of ordinary differential equations. Because of the nonlinear behaviour, it is a challenging task to control the motion of the two-link robot manipulator at accurate position defined by the user. For this, we focus mainly on control of the robot manipulator to get the desired position using the computed torque control method. After deriving the equation of motion, control simulation is represented using MATLAB.

The control approach design was based on the dynamic model of the robot. The mathematical model of the system was nonlinear, and thus a feedback linearization control was first proposed to obtain a linear system. With the linearized model, a state feedback controller is designed to move the robotic arms to a desired position. We have also presented a PID controller to simulate how we would balance the two-links on a moving robot to any specific angle including upside-down.

Introduction

Robotic manipulators are a major component in the manufacturing industry. In recent years, the control of manipulator robots has been the subject of much research, due to the robots' increasingly frequent use in dangerous or inaccessible environments, where human beings can hardly intervene. They are used for many reasons including speed, accuracy, and repeatability. Increasingly, robotic manipulators are finding their way into our everyday life. In fact, in almost every product we encounter, a robotic manipulator has played a part in its production. These robot models are highly nonlinear which makes the control strategy very difficult. Two approaches to controlling manipulator robots are proposed in this project work.

The double pendulum system is a common and typical model to investigate nonlinear dynamics due to its various and complex dynamical phenomenon. The double pendulum is an example of a simple dynamical system that exhibits complex behavior, including chaos. It consists of two-point masses at the end of light rods. Each mass plus rod is a regular simple pendulum. The two pendula are joined together, and the system is free to oscillate in a plane that represents the 2-arm robotic manipulator dynamics and it also forms 2 degree of freedom.

DYNAMICS OF THE PLANT MODEL

We can begin to understand the complex movement of human arms after defining the movement of the two-link manipulator. Double pendula are an example of a simple physical system which can exhibit chaotic behavior. Understanding the two-link manipulator is key to learning robotic manipulators. The total potential and kinetic energies of the two-link system are defined and used to form the Lagrangian. The physical system is shown in figure (1).

The system consists of two masses connected by weightless bars. The bars have lengths L_1 and L_2. The masses are denoted by M_1 and M_2 respectively. Let θ_1 and θ_2 denote the angle in which the first bar rotates about the origin and the second bar rotates about the endpoint of the first bar, respectively. This system has two degrees of freedom θ_1 and θ_2.

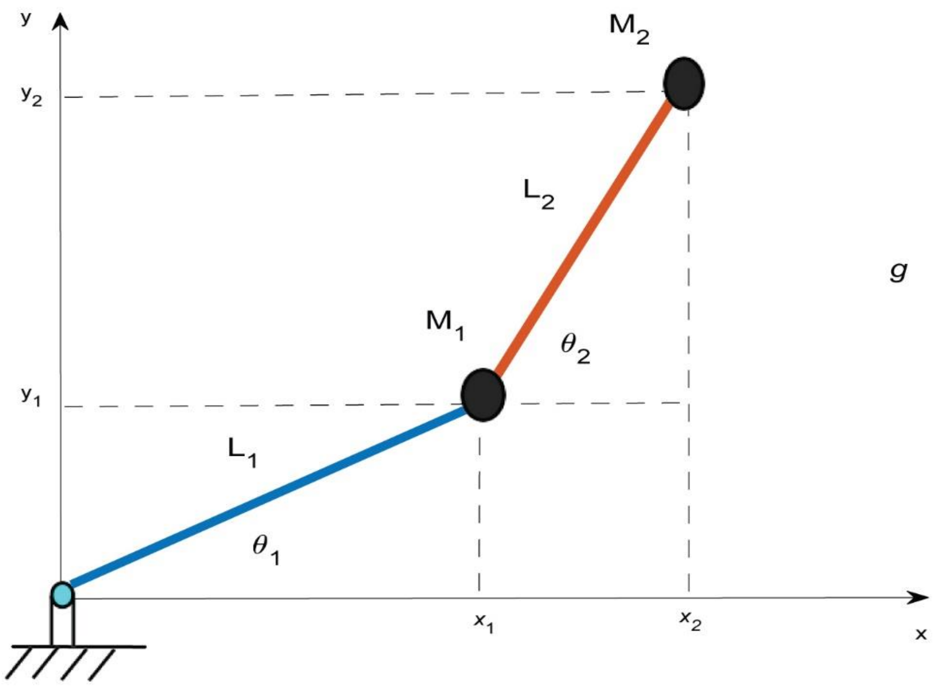

Figure (1): Freebody diagram of 2 arm robotic manipulator.

Position of M_1 is given by

$$x_1 = L_1 \cos(\theta_1), \qquad y_1 = L_1 \sin(\theta_1)$$

Simillarly, Position of M_2 is given by

$x_2 = L_1 \cos(\theta_1) + L_2 \cos(\theta_2)$, $y_2 = L_1 \sin(\theta_1) + L_2 \sin(\theta_2)$

Velocity of M_1 and M_2

$$v_1 = \sqrt{\dot{x}_1^2 + \dot{y}_1^2}\ , v_2 = \sqrt{\dot{x}_2^2 + \dot{y}_2^2} \longrightarrow \text{(i)}$$

Kinetic energy of the system is

$$\text{KE}=\frac{1}{2} M_1 v_1^2 + \frac{1}{2} M_2 v_2^2$$

Substituting v_1 and v_2 from equation (i),

$$\text{K.E}= \frac{1}{2}(M_1 + M_2) L_1^2 \dot{\theta}_1^2 + \frac{1}{2} M_2 L_2^2 \dot{\theta}_2^2 + M_2 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2) \longrightarrow \text{(ii)}$$

Potential Energy,

P.E= M_1g y_1+ M_2g y_2

or $(M_1 + M_2)$g $L_1 \sin(\theta_1)$ + M_2g $L_2 \sin(\theta_2)$

By Lagrange Dynamics, we form the Lagrangian $\mathscr{L}$ which is defined as, $\mathscr{L}$=K.E –P.E

The Euler-Lagrange equation is given by the equation

$$\frac{d}{dt}\left[\frac{\partial \text{L}}{\partial \dot{\theta}_i}\right] - \frac{\partial \text{L}}{\partial \theta_i} = \tau_i \quad \text{i=1,2...} \longrightarrow \text{(iii)}$$

Where τ_i is the torque applied to the i$^{\text{th}}$ link.

By solving we get

$$(M_1 + M_2) L_2 \ddot{\theta}_1 + M_2 L_1 L_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + M_2 L_1 L_2 \theta_2 \sin(\theta_1 - \theta_2) + (M_1 + M_2) g L_1 \cos(\theta_1) = \tau_1$$

And

$$M_2 L_2^2 \ddot{\theta}_2 + M_2 L_1 L_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - M_2 L_1 L_2 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + M_2 g L_2 \cos(\theta_2) = \tau_2$$

These equations can be written as-

$$M(\theta)\ddot{\theta} + c(\theta, \dot{\theta}) + G(\theta) = F$$

$$Y = \theta \longrightarrow \text{(iv)}$$

Where,

- Y is the output vector.

- $M(\theta) = \begin{bmatrix} (M_1 + M_2)L_1^2 & M_2L_1L_2\cos(\theta_1 - \theta_2) \\ M_2L_1L_2\cos(\theta_1 - \theta_2) & M_2L_2^2 \end{bmatrix}$

- $c(\theta, \dot{\theta}) = \begin{bmatrix} M_2\ L_1\ L_2\ \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) \\ -M_2\ L_1\ L_2\ \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) \end{bmatrix}$

- $G(\theta) = \begin{bmatrix} (M_1 + M_2)gL_1\cos(\theta_1) \\ M_2gL_2cos\theta_2 \end{bmatrix}$
- $F = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$

PID Controller Design

Robotic manipulators are generally difficult to control. In particular, it is a challenging task to stabilize a robot manipulator at a fixed, accurate position. In this section, we focus mainly on control of the robot manipulator to get the desired position using computed torque control method. After deriving the equation of motion, control simulation is represented using MATLAB. We define control as the ability to hold the system of two links in a particular position on the xy-plane. Having control gives us the ability to hold each link at a particular angle θi with respect to the positive x-axis. The proportional-integral-derivative (PID) controller is a common control algorithm. The "P" in PID stands for Proportional control, the "I" stands for Integral control, and the "D" stands for Derivative control. This algorithm works by defining an error variable, Verror = Vset − Vsensor. That takes the position we want to go (Vset) minus the position we are actually at (Vsensor). We get the proportional part of the PID control by taking a constant defined as KP and multiplying it by the error. The I comes from taking a constant KI and multiplying it by the integral of the error with respect to time. Derivative control is defined as a constant KD multiplied by the derivative of the error with respect to time. Many industrial processes are controlled using PID controllers.

The equations of motion can be written from previous equation (iv) as

$$M(\theta)\ddot{\theta} + c(\theta, \dot{\theta}) + G(\theta) = F$$

Where

- $\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$,
- $M(\theta) = \begin{bmatrix} (M_1 + M_2)L_1^2 & M_2\ L_1\ L_2 \cos(\theta_1 - \theta_2) \\ M_2\ L_1\ L_2 \cos(\theta_1 - \theta_2) & M_2 L_2^2 \end{bmatrix}$
- $C(\theta, \dot{\theta}) = \begin{bmatrix} M_2\ L_1\ L_2\ \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) \\ -M_2\ L_1\ L_2\ \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) \end{bmatrix}$,
- $G(\theta) = \begin{bmatrix} (M_1 + M_2) g L_1 \cos(\theta_1) \\ M_2 g L_2 cos\theta_2 \end{bmatrix}$,
- $F = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$

We can solve for some theoretical values of forces given certain initial inputs solving for $\ddot{\theta}$ we get

$$\ddot{\theta} = -M^{-1}(\theta)\left[c(\theta, \dot{\theta}) + G(\theta)\right] + \hat{F},$$

Where, $\hat{F} = M^{-1}(\theta)F$.

Thus, we decoupled the system to have the new input $\hat{F} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$.

However, the physical torque inputs to the system are $F = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = M(\theta) \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$.

Let us denote the error signals by $e(\theta_1) = \theta_{1f} - \theta_1$, $e(\theta_2) = \theta_{2f} - \theta_2$, where the target positions of M_1 and M_2 are given by the angels θ_{1f} and θ_{2f} respectively.

We assume that the system has initial positions $\theta_0 = \begin{bmatrix} \theta_1(0) \\ \theta_2(0) \end{bmatrix}$.

A common technique for controlling a system with input is to use the following general structure of PID controller $f = K_P e + K_D \dot{e} + K_I \int e\, dt$.

In our situation we have inputs f_1 and f_2 is to employ two independent controllers, one for each link, as follows

- $f_1 = K_{P_1} e(\theta_1) + K_{D_1} \dot{e}(\theta_1) + K_{I_1} \int e\,(\theta_1) dt = K_{P_1}(\theta_{1f} - \theta_1) - K_{D_1} \dot{\theta}_1 + K_{I_1} \int (\theta_{1f} - \theta_1) dt$
- $f_2 = K_{P_2} e(\theta_2) + K_{D_2} \dot{e}(\theta_2) + K_{I_2} \int e\,(\theta_2) dt = K_{P_2}(\theta_{2f} - \theta_2) - K_{D_2} \dot{\theta}_2 + K_{I_2} \int (\theta_{2f} - \theta_2) dt$,

  Where θ_{1f} and θ_{2f} are given constants.

The complete system of equations with control is then

$$\ddot{\theta} = -M^{-1}(\theta)\left[c(\theta, \dot{\theta}) + G(\theta)\right] + \hat{F}$$

Where, $\hat{F} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} K_{P_1}(\theta_{1f} - \theta_1) - K_{D_1} \dot{\theta}_1 + K_{I_1} \int (\theta_{1f} - \theta_1) dt \\ K_{P_2}(\theta_{2f} - \theta_2) - K_{D_2} \dot{\theta}_2 + K_{I_2} \int (\theta_{2f} - \theta_2) dt \end{bmatrix}$.

We would like to emphasize that the actual physical torques are

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = M(\theta) \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}.$$

To implement the PID controller, we introduce the following new states
$x_1 = \int e\,(\theta_1) dt,\ x_2 = \int e\,(\theta_2) dt$

Differentiating with respect to t gives, $\dot{x}_1 = e(\theta_1) = \theta_{1f} - \theta_1$,
$\dot{x}_2 = e(\theta_2) = \theta_{2f} - \theta_2$.

The complete equations are $\dot{x}_1 = \theta_{1f} - \theta_1,\ \dot{x}_2 = \theta_{2f} - \theta_2$

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = -M^{-1}(\theta)\left[c(\theta, \dot{\theta}) + G(\theta)\right] + \begin{bmatrix} K_{P_1}(\theta_{1f} - \theta_1) - K_{D_1} \dot{\theta}_1 + K_{I_1} x_1 \\ K_{P_2}(\theta_{2f} - \theta_2) - K_{D_2} \dot{\theta}_2 + K_{I_2} x_1 \end{bmatrix}$$

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = M(\theta) \begin{bmatrix} K_{P_1}(\theta_{1f} - \theta_1) - K_{D_1} \dot{\theta}_1 + K_{I_1} x_1 \\ K_{P_2}(\theta_{2f} - \theta_2) - K_{D_2} \dot{\theta}_2 + K_{I_2} x_1 \end{bmatrix}.$$

To discretize the system of differential equations in time, we transform them into a system of first-order ordinary differential equations. To do this, we define six new variables as follows

$u_1 = x_1,\ u_2 = x_2,\ u_3 = \theta_1,\ u_4 = \theta_2,\ u_5 = \dot{\theta}_1, u_6 = \dot{\theta}_2,$

After differentiating, we have, $\dot{u}_1 = \dot{x}_1 = \theta_{1f} - u_3$, $\quad \dot{u}_2 = \dot{x}_2 = \theta_{2f} - u_4$,
$\dot{u}_3 = \dot{\theta}_1 = u_5$, $\dot{u}_4 = \dot{\theta}_2 = u_6$, $\quad \dot{u}_5 = \ddot{\theta}_1 = \phi(t, u_1, u_2, u_3, u_4, u_5, u_6)$,

$\dot{u}_6 = \ddot{\theta}_2 = \psi(t, u_1, u_2, u_3, u_4, u_5, u_6)$,

Where ϕ and ψ are expressed in terms of u_k, $k = 1 - 6$, as

$$\begin{bmatrix} \phi \\ \psi \end{bmatrix} = -M^{-1}(\theta)\left[c(\theta, \dot{\theta}) + G(\theta)\right] + \begin{bmatrix} K_{P_1}(\theta_{1f} - u_3) - K_{D_1}u_5 + K_{I_1}u_1 \\ K_{P_2}(\theta_{2f} - u_4) - K_{D_2}u_6 + K_{I_2}u_2 \end{bmatrix},$$

and $\theta = \begin{bmatrix} u_3 \\ u_4 \end{bmatrix}$.

A simple calculation shows that

$$\phi = \frac{-M_2 L_2 u_6^2 \sin(u_3 - u_4) - (M_1 + M_2) g \cos(u_3) - M_2 \cos(u_3 - u_4)(L_1 u_5^2 \sin(u_3 - u_4) - g\cos(u_4))}{L_1(M_1 + M_2 - M_2 cos^2(u_3 - u_4))} + K_{P_1}(\theta_{1f} - u_3) - K_{D_1}u_5 + K_{I_1}u_1$$

$$\psi = \frac{\cos(u_3 - u_4)(M_2 L_2 u_6^2 \sin(u_3 - u_4) + (M_1 + M_2) g \cos(u_3)) + (M_1 + M_2)(L_1 u_5^2 \sin(u_3 - u_4) - g\cos(u_4))}{L_2(M_1 + M_2 - M_2 cos^2(u_3 - u_4))} + K_{P_2}(\theta_{2f} - u_4) - K_{D_2}u_6 + K_{I_2}u_2$$

Thus, we obtain a system of first-order nonlinear differential equations of the form

$$\frac{dU}{dt} = H(t, U), \; U(0) = U_0$$

Where, $\mathrm{U} = [u_1, u_2, u_3, u_4, u_5, u_6]^t$ and $\mathrm{H} = [h_1, h_2, h_3, h_4, h_5, h_6]^t$ with
$h_1 = \theta_{1f} - \mathrm{u}_3$, $\quad h_2 = \theta_{2f} - \mathrm{u}_4$, $\quad h_3 = \mathrm{u}_5$, $\quad h_4 = \mathrm{u}_6$,

$h_5 = \phi(t, u_1, u_2, u_3, u_4, u_5, u_6)$, $\quad h_6 = \psi(t, u_1, u_2, u_3, u_4, u_5, u_6)$.

The initials conditions are given by

$U_0 = [u_1(0), u_2(0), u_3(0), u_4(0), u_5(0), u_6(0)]^t$

Where

$u_1(0) = x_1(0)$, $u_2(0) = x_2(0)$, $u_3(0) = \theta_1(0)$, $u_4(0) = \theta_2(0)$,

$u_5(0) = \dot{\theta}_1(0)$, $u_6(0) = \dot{\theta}_2(0)$.

We have to solve the differential equation for unknown U. Once we solve for, we obtain to torques using U=$(u_1, u_2, u_3, u_4, u_5, u_6)^t$,

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = \mathrm{M}(\theta)\begin{bmatrix}K_{P_1}(\theta_{1f}-u_3)-K_{D_1}u_5+K_{I_1}u_1\\ K_{P_2}(\theta_{2f}-u_4)-K_{D_2}u_6+K_{I_2}u_2\end{bmatrix},$$

or equivalently

τ_1=$(M_1+M_2)L_1^2(K_{P_1}(\theta_{1f}-u_3)$- $K_{D_1}u_5+K_{I_1}u_1)$+ M_2 L_1 L_2 cos(u_3 - u_4)$(K_{P_2}(\theta_{2f}-u_4)-K_{D_2}u_6+K_{I_2}u_2)$

$\tau_2 = M_2$ L_1 L_2 cos(u_3 - u_4) $(K_{P_1}(\theta_{1f}-u_3)$- $K_{D_1}u_5+K_{I_1}u_1)$+ M_2 $L_2^2(K_{P_2}(\theta_{2f}-u_4)-K_{D_2}u_6+K_{I_2}u_2)$

Let us consider a simplified model of a two-link manipulator. We take the following parameters

$M_1 = M_2 = 1, \quad L_1 = 2, \quad L_2 = 1$

The target positions (final positions) are $\begin{bmatrix}\theta_{1f}\\ \theta_{2f}\end{bmatrix} = \begin{bmatrix}\pi/4\\ -\pi/6\end{bmatrix}$

The initial positions, initial angle velocities, and initial states are, respectively taken as

$$\begin{bmatrix}\theta_1(0)\\ \theta_2(0)\end{bmatrix} = \begin{bmatrix}\pi/2\\ \pi/2\end{bmatrix}, \quad \begin{bmatrix}\dot\theta_1(0)\\ \dot\theta_2(0)\end{bmatrix} = \begin{bmatrix}0\\ 0\end{bmatrix}, \quad \begin{bmatrix}x_1(0)\\ x_2(0)\end{bmatrix} = \begin{bmatrix}0\\ 0\end{bmatrix}.$$

- The PID parameters for θ_1 and θ_2, are taken as
- $\mathrm{KP}_1 = 30, \quad \mathrm{K}D_1 = 15, \quad \mathrm{K}I_1 = 20$
- $\mathrm{KP}_2 = 30, \quad \mathrm{K}D_2 = 10, \quad \mathrm{K}I_2 = 20$

State Feedback Linearization of the plant

The main idea of this technique is to transform the nonlinear dynamics of the system to a completely or partially linear one. The equations of motion of the two-link manipulator can be written from equation (iv) are

$M(\theta)\ddot\theta + c(\theta,\dot\theta) + G(\theta) = F$

$Y = \theta$

where Y is the output vector.

Differentiate the output Y until the control input F appeared. In this case, the control input F appeared in the second derivative of the output Y.

$$\ddot{Y} = \ddot{\theta} = M(\theta)^{-1}\left(-C(\theta,\dot{\theta}) - G(\theta) + F\right) = v \longrightarrow \text{(v)}$$

Here det(M) $\neq$0.

where v is the control vector to this linear double-integrator system. The relative degree is equal to two. This mean that by using this control law, we can do complete linearization of the nonlinear system of the plant.

The relation between the control vector and the actual control torque is given by

$$F = M(\theta)v + C(\theta,\dot{\theta}) + G(\theta)$$

Applying the above control law on equation (v), we have the dynamic model of the manipulator robot with two DOF, becomes a linear double-integrator system.

$$\frac{\theta_1(s)}{v_1(s)} = \frac{1}{s^2} \text{ and } \frac{\theta_2(s)}{v_2(s)} = \frac{1}{s^2}$$

where s denotes the independent variable of the Laplace transform.

- Based on this linearized model we have feedback control law in our simulation.
- Reference θ_{1f}= pi/4 and θ_{2f}=-pi/6

Simulation Results

i) PID

The PID controller helps get the outputs, which are the final positions of M_1 and M_2 determined by the angles θ_1 and θ_2, where we want it, in a short time, with minimal overshoot, and with little error. This is implementation is done using MATLAB and its response are provided below.

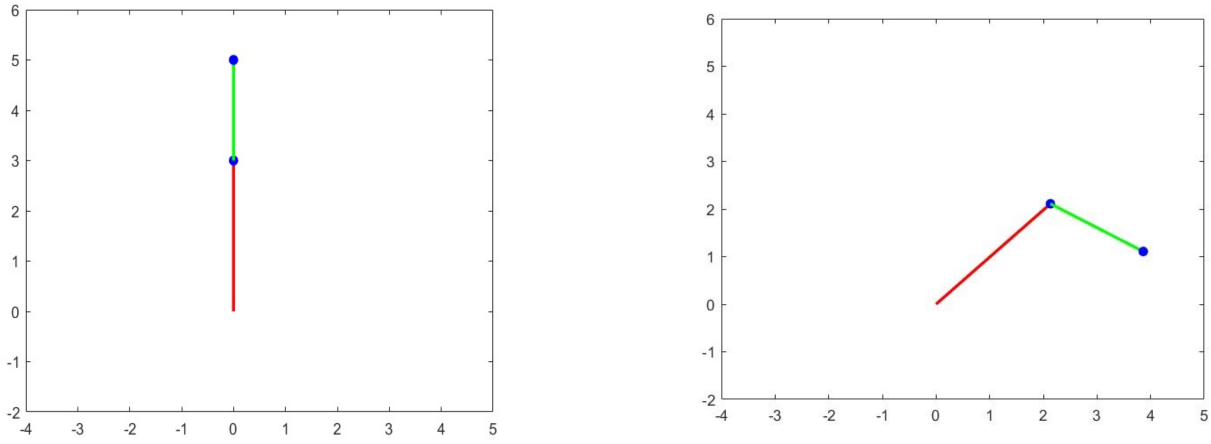

Figure (2): The initial positions (left) and the final position (right) of the 2-link robotic arm

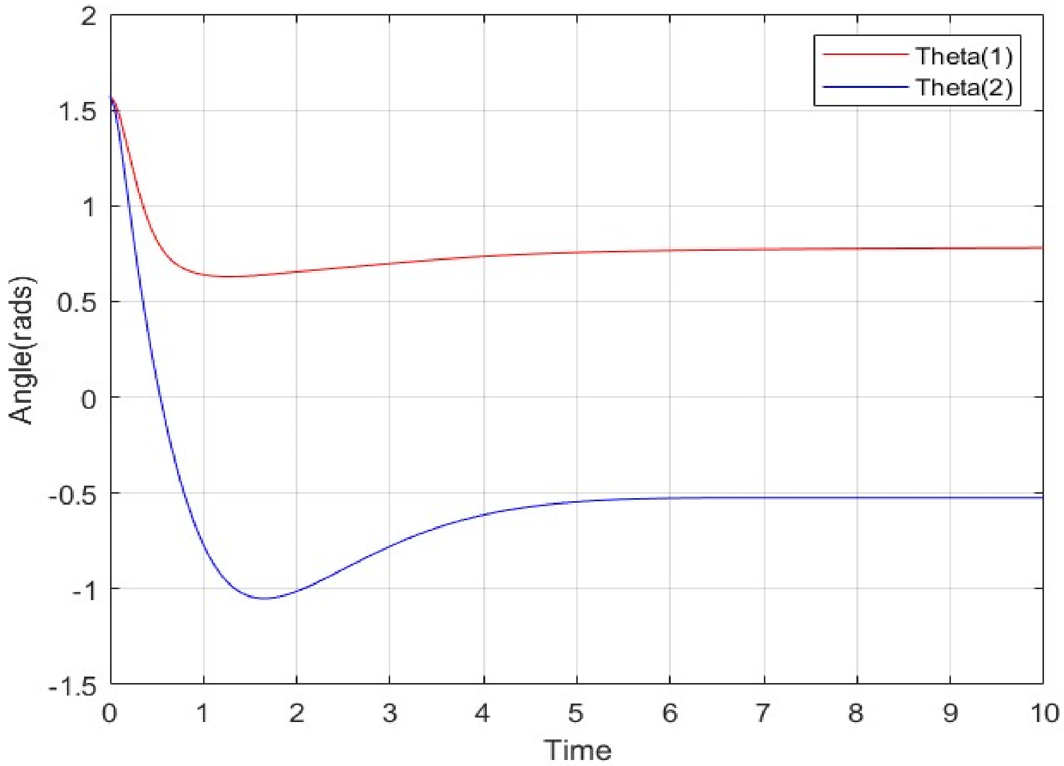

Figure (3): The positions of M_1 and M_2 versus time.

ii) <u>State feedback linearization</u>

After linearizing the plant model, state feedback controller is used to get the desired position of the robotic arm accurately, with minimum overshoot. This implementation is done in MATLAB Simulink, the results are depicted in figure (4) and figure (5).

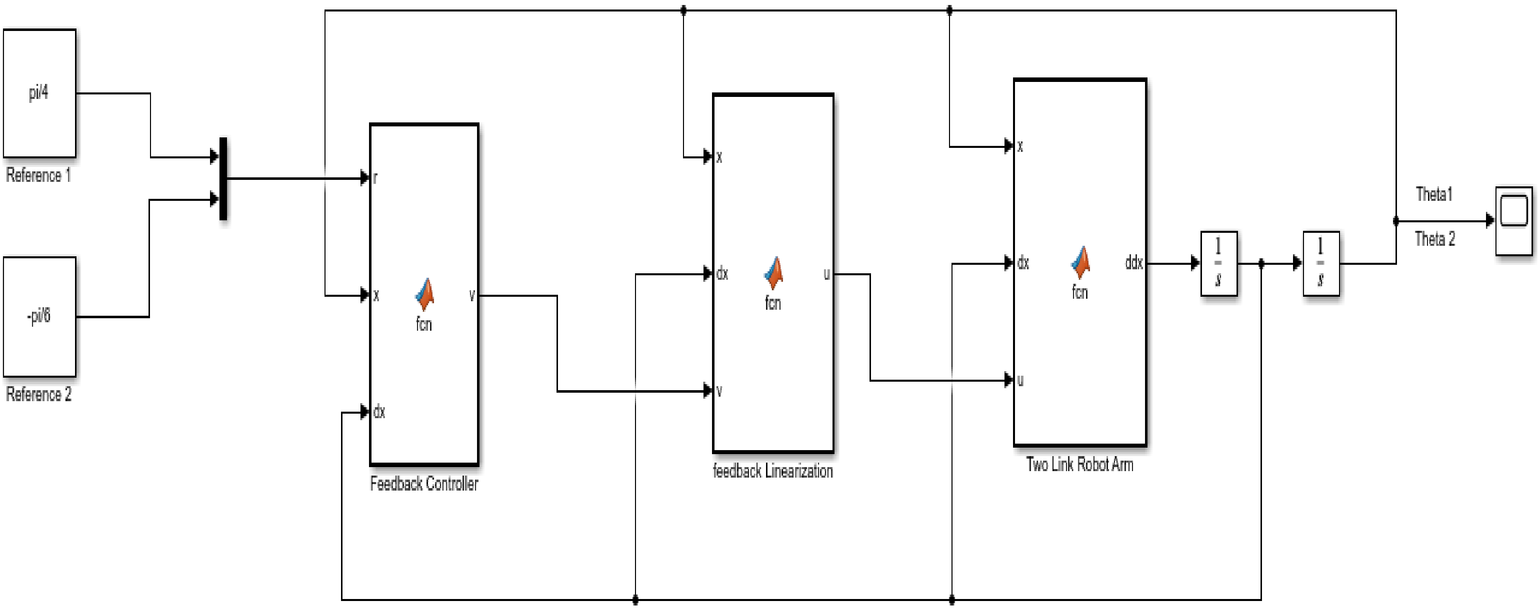

Figure (4): Closed loop system (State Feedback linearization)

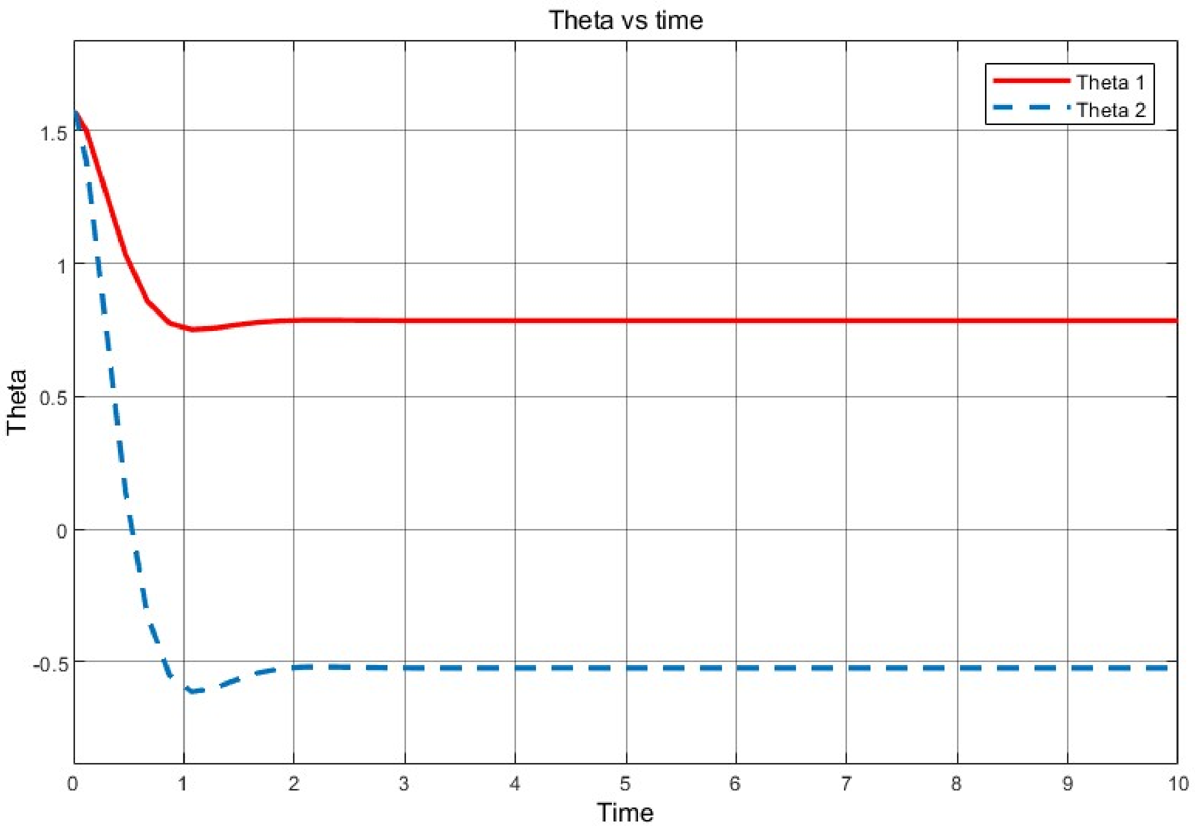

Figure (5): Initial and the desired orientations vs time of the 2-link of the robot.

Conclusion

In this project, a two-link robotic manipulator was studied, and its dynamics were modeled using La-grange mechanics. First, we derived the dynamical equation of the two-link robot manipulators using Euler-Lagrange method and then a simple and efficient control scheme was developed. More specifically, a robust Proportional-Integral-Derivative (PID) controller was introduced to control the motion of the robot at a specific position. In particular, we described how a PID controller can be used to keep the links in a desired position. The efficiency of the PID controller is verified by the simulation results. We have also linearized the plant using the feedback linearization technique and state feedback controller is implemented to the linearized plant model. The results of the simulations were also verified, and it provided less settling time, overshoot and rise time as compared to the PID based controller. But the drawback of feedback linearization is that if the internal dynamics of the system is unstable, than the system cannot be linearized with by this method.

Reference

1. Applied Nonlinear control by J.E. Slotine and W.Li.
2. https://en.wikipedia.org/wiki/Robotic_arm
3. https://vibgyorpublishers.org/content/ijre/fulltext.php?aid=ijre-5-028
4. https://en.wikipedia.org/wiki/Feedback_linearization
5. http://www.scholarpedia.org/article/Feedback_linearization