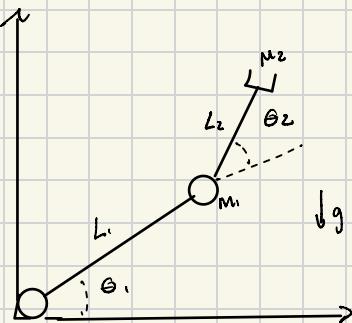


Point at $M_1 \rightarrow (x_1, y_1)$

$$x_1 = L_1 \cos \theta_1$$

$$y_1 = L_1 \sin \theta_1$$

Calculating Kinetic Energy, KE



Point at $M_2 \rightarrow (x_2, y_2)$

$$\dot{x}_2 = L_1 \cos(\theta_1) + L_2 \cos(\theta_2)$$

$$\dot{y}_2 = L_1 \sin(\theta_1) + L_2 \sin(\theta_2)$$

$$\dot{x}_1 = -L_1 \dot{\theta}_1 \sin(\theta_1)$$

$$\dot{x}_2 = -\dot{\theta}_1 L_1 \sin(\theta_1) - \dot{\theta}_2 L_2 \sin(\theta_2)$$

$$\dot{y}_1 = L_1 \dot{\theta}_1 \cos(\theta_1)$$

$$\dot{y}_2 = \dot{\theta}_1 L_1 \cos(\theta_1) + \dot{\theta}_2 L_2 \cos(\theta_2)$$

$$\text{Note } \dot{x} = \frac{dx}{dt}$$

$$\dot{\theta} = \frac{d\theta}{dt}$$

$v_1 \rightarrow$ velocity at M_1

$$v_1^2 = \dot{x}_1^2 + \dot{y}_1^2$$

$v_2 \rightarrow$ velocity at M_2

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2$$

$$KE = \frac{1}{2} mv^2 \rightarrow \text{Total KE at the system} = \underbrace{KE_{M_1} + KE_{M_2}}_{\text{Kinetic Energy of } M_1 \text{ & } M_2}$$

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

$$KE = \frac{1}{2} M_1 (\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2} M_2 (\dot{x}_2^2 + \dot{y}_2^2)$$

$$KE = \frac{1}{2} M_1 \left[(-L_1 \dot{\theta}_1 \sin(\theta_1))^2 + (L_1 \dot{\theta}_1 \cos(\theta_1))^2 \right] + \frac{1}{2} M_2 \left[(-L_1 \dot{\theta}_1 \sin(\theta_1) - L_2 \dot{\theta}_2 \sin(\theta_2))^2 + (\dot{\theta}_1 L_1 \cos(\theta_1) + \dot{\theta}_2 L_2 \cos(\theta_2))^2 \right]$$

\curvearrowright KE of M_1

\curvearrowright KE of M_2

Simplified we get the Kinetic Energy of the System.

$$KE = \frac{1}{2} (M_1 + M_2) L_1^2 \dot{\theta}_1^2 + \frac{1}{2} M_2 L_2^2 \dot{\theta}_2^2 + M_2 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2)$$

Calculating PE

$$PE = M_i g h_i(\theta) \quad i=1,2$$

$$PE(\theta) = \sum_{i=1}^2 PE_i(\theta) = M_i g h_i(\theta)$$

$$M_1 g L_1 \sin(\theta_1) + M_2 g [L_1 \sin(\theta_1) + L_2 \sin(\theta_2)]$$

$\underbrace{M_1 g L_1 \sin(\theta_1)}_{\text{PE of } M_1} \quad \underbrace{M_2 g [L_1 \sin(\theta_1) + L_2 \sin(\theta_2)]}_{\text{PE of } M_2}$

Introducing Lagrangian Dynamics

$$L = KE - PE$$

$$\left[\frac{1}{2}(M_1 + M_2) L_1^2 \dot{\theta}_1^2 + \frac{1}{2} M_2 L_2^2 \dot{\theta}_2^2 + M_2 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2) \right] - \left[g L_1 (M_1 + M_2) \sin(\theta_1) + M_2 g L_2 \sin(\theta_2) \right]$$

Simplifying we get

$$L = \frac{1}{2} (M_1 + M_2) L_1^2 \dot{\theta}_1^2 + \frac{1}{2} M_2 L_2^2 \dot{\theta}_2^2 + M_2 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2) - (M_1 + M_2)$$

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}_i} \right] - \frac{\partial L}{\partial \theta_i} = \tau \quad i=1,2$$

$$\begin{cases} \cos(\theta_1 - \theta_2) \\ -\sin(\theta_1 - \theta_2) \end{cases}$$

$$\frac{\partial L}{\partial \dot{\theta}_1} = (M_1 + M_2) L_1^2 \dot{\theta}_1 + M_2 L_1 L_2 \dot{\theta}_2 \cos(\theta_1 - \theta_2)$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = M_2 L_2^2 \dot{\theta}_2 + M_2 L_1 L_2 \dot{\theta}_1 \cos(\theta_1 - \theta_2) \quad \leftarrow \text{Trig Property}$$

$$\frac{\partial L}{\partial \theta_1} = -M_2 L_1 L_2 \dot{\theta}_1 \sin(\theta_1 - \theta_2) - (M_1 + M_2) g L_1 \cos(\theta_1)$$

$$\frac{\partial L}{\partial \theta_2} = M_2 L_1 L_2 \dot{\theta}_2 \sin(\theta_1 - \theta_2) - M_2 g L_2 \cos(\theta_2)$$

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{\theta}_1} \right] = (M_1 + M_2) L_1^2 \ddot{\theta}_1 + M_2 L_1 L_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) - M_2 L_1 L_2 \dot{\theta}_2 (\dot{\theta}_1 - \dot{\theta}_2) \sin(\theta_1 - \theta_2).$$

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{\theta}_2} \right] = M_2 L_2^2 \ddot{\theta}_2 + M_2 L_1 L_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - M_2 L_1 L_2 \dot{\theta}_1 (\dot{\theta}_1 - \dot{\theta}_2) \sin(\theta_1 - \theta_2).$$

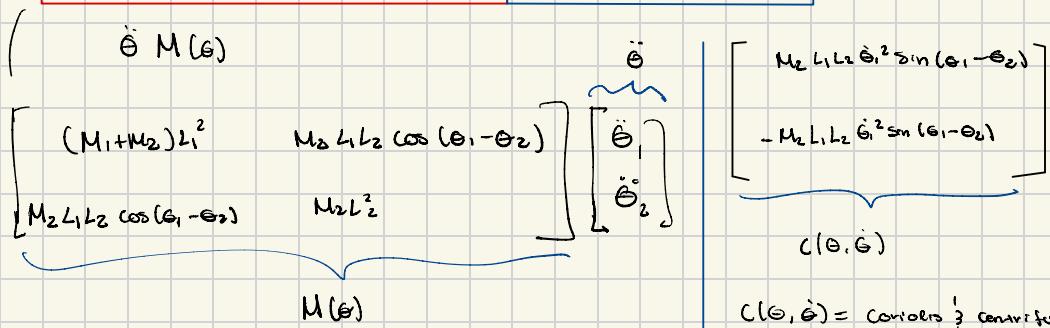
$$\Sigma m(\alpha - \beta) = \sin \alpha \cos \beta - \cos \alpha \sin \beta$$

$$(M_1 + M_2) L_1^2 \ddot{\theta}_1 + M_2 L_1 L_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) - M_2 L_1 L_2 \dot{\theta}_1 (\theta_1 - \theta_2) \sin(\theta_1 - \theta_2)$$

$$+ M_2 L_1 L_2 \dot{\theta}_1 \sin(\theta_1 - \theta_2) + (M_1 + M_2) g L_1 \cos(\theta_1) = T_1$$

$$(M_1 + M_2) L_1^2 \ddot{\theta}_1 + M_2 L_1 L_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + M_2 L_1 L_2 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + (M_1 + M_2) g L_1 \cos(\theta_1) = T_1$$

$$M_2 L_2^2 \ddot{\theta}_2 + M_2 L_1 L_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - M_2 L_1 L_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + M_2 g L_2 \cos(\theta_2) = T_2$$



$M(\theta) \rightarrow$ Mass or Inertia matrix

In the form $\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$

$M_{21} \rightarrow$ Inertia of 2nd Link

$M_{11} \rightarrow$ Inertia of 1st Link includes both masses of $M_1 \frac{1}{3} M_2$

$\begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = F$ Torques applied to joints

$C(\theta, \dot{\theta}) =$ Controls $\frac{1}{3}$ centrifugal forces

- Forces that act outwardly due to the rotation of the links

$$\begin{bmatrix} (M_1 + M_2)g L_1 \cos(\theta_1) \\ M_2 g L_2 \cos(\theta_2) \end{bmatrix}$$

$G(\theta)$

↳ gravitational Forces

$$M(\theta) \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} + C(\theta, \dot{\theta}) + G(\theta) = F$$

$$M(\theta) \ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) = F \rightarrow \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

Solving for $\ddot{\theta}$

$$\text{Note} \rightarrow M^{-1}(\theta) F = \hat{F}$$

$$\ddot{\theta} = -M^{-1}(\theta) [C(\theta, \dot{\theta}) + G(\theta)] + \hat{F} \rightarrow \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$

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

Calculating error

$$e(\theta_1) = \theta_{1,f} - \theta_1$$

\uparrow
actual position \uparrow theoretical position

$$e(\theta_2) = \theta_{2,f} - \theta_2$$

Introducing PID controllers

- Common technique for controlling a system with input

$$f = k_p e + k_i \int e dt + k_d \dot{e}$$

Initial positions

$$\theta_0 = \begin{bmatrix} \theta_{1,0} \\ \theta_{2,0} \end{bmatrix}$$

$$\begin{aligned} f_1 &= k_p e(\theta_1) + k_i \int e(\theta_1) dt + k_d \dot{e}(\theta_1) \\ f_1 &= k_p (\theta_{1,f} - \theta_1) - \underbrace{k_i (\theta_{1,f} - \theta_1)}_{-\dot{\theta}_1} + k_d \int (\theta_{1,f} - \theta_1) dt \end{aligned}$$

$$\begin{aligned} f_2 &= k_p e(\theta_2) + k_i \int e(\theta_2) dt + k_d \dot{e}(\theta_2) \\ f_2 &= k_p (\theta_{2,f} - \theta_2) - \underbrace{k_i (\theta_{2,f} - \theta_2)}_{-\dot{\theta}_2} + k_d \int (\theta_{2,f} - \theta_2) dt \end{aligned}$$

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} k_p (\theta_{1,f} - \theta_1) - k_i \dot{\theta}_1 + k_d \int (\theta_{1,f} - \theta_1) dt \\ k_p (\theta_{2,f} - \theta_2) - k_i \dot{\theta}_2 + k_d \int (\theta_{2,f} - \theta_2) dt \end{bmatrix} = \hat{F}$$

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = M(\theta_1) \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

To Implement PID controller \rightarrow we must implement time states

$$\begin{aligned} x_1 &= \int e(\theta_1) dt & \dot{x}_1 &= e(\theta_1) = \theta_{1f} - \theta_1, \quad x_2 = e(\theta_2) = \theta_{2f} - \theta_2 \\ x_2 &= \int e(\theta_2) dt & \dot{x}_2 &= e(\theta_2) = \theta_{2f} - \theta_2 \end{aligned}$$

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = -M^{-1}(\theta) \begin{bmatrix} C(\theta, \dot{\theta}) + G(\theta, 1) \\ C(\theta, \dot{\theta}) + G(\theta, 2) \end{bmatrix} + \begin{bmatrix} k_{p1}(\theta_{1f} - \theta_1) - k_{01}\theta_1 + k_{i1}x_1 \\ k_{p2}(\theta_{2f} - \theta_2) - k_{02}\theta_2 + k_{i2}x_2 \end{bmatrix}$$

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = M(\theta) \begin{bmatrix} k_{p1}(\theta_{1f} - \theta_1) - k_{01}\theta_1 + k_{i1}x_1 \\ k_{p2}(\theta_{2f} - \theta_2) - k_{02}\theta_2 + k_{i2}x_2 \end{bmatrix}$$

$$u_1 = \tau_1, \quad u_2 = x_2, \quad u_3 = \theta_1, \quad u_4 = \theta_2, \quad u_5 = \dot{\theta}_1, \quad u_6 = \dot{\theta}_2$$

$$\ddot{\theta}_1 = \dot{x}_1$$

$$\ddot{\theta}_2 = \dot{x}_2 \quad \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = M(\theta) \begin{bmatrix} k_{p1}(\theta_{1f} - u_3) - k_{01}u_3 + k_{i1}u_1 \\ k_{p2}(\theta_{2f} - u_4) - k_{02}u_4 + k_{i2}u_2 \end{bmatrix}$$

$$\ddot{\theta}_1 = \dot{\theta}_1 = \phi(t, u_1, u_2, u_3, u_4, u_5, u_6)$$

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

$$\begin{bmatrix} \phi \\ \psi \end{bmatrix} = -M^{-1}(\theta) \begin{bmatrix} C(\theta, \dot{\theta}) + G(\theta, 1) \\ C(\theta, \dot{\theta}) + G(\theta, 2) \end{bmatrix} + \begin{bmatrix} k_{p1}(\theta_{1f} - u_3) - k_{01}u_3 + k_{i1}u_1 \\ k_{p2}(\theta_{2f} - u_4) - k_{02}u_4 + k_{i2}u_2 \end{bmatrix}$$

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

$$(M_1 + M_2)L_1^2\ddot{\theta}_1 + M_2L_1L_2\ddot{\theta}_2 \cos(\theta_1 - \theta_2) + M_2L_1L_2\dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + (M_1 + M_2)gL_1 \cos(\theta_1) = \tau_1,$$

$$\rightarrow \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}_1} \right] - \frac{\partial L}{\partial \theta_1} = \tau_1$$

$$M_2L_2^2\ddot{\theta}_2 + M_2L_1L_2\ddot{\theta}_2 \cos(\theta_1 - \theta_2) - M_2L_1L_2\dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + M_2gL_2 \cos(\theta_2) = \tau_2, \quad \rightarrow \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}_2} \right] - \frac{\partial L}{\partial \theta_2} = \tau_2$$

$$\begin{bmatrix} (M_1+M_2)L_1^2 & M_2L_1L_2 \cos(\theta_1 - \theta_2) \\ M_2L_1L_2 \cos(\theta_1 - \theta_2) & M_2L_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$

$$S = \frac{M_2}{M_1 + M_2}$$

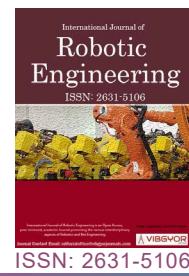
$$\ddot{\theta} = M(\theta) + C(\theta, \dot{\theta}) + G(\theta) = F$$

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

$$\hat{F} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$



A Two-Link Robot Manipulator: Simulation and Control Design



Mahboub Baccouch^{1*} and Stephen Dodds²

¹Professor, Department of Mathematics, University of Nebraska at Omaha, USA

²CEO, JUUK CO, Austin, USA

Abstract

In this paper, we design a robust, fast, and practical proportional-integral-derivative (PID) controller for the classical double pendulum system. 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 closed form solutions of the equations of motion are not available, we use the classical fourth-order Runge-Kutta method to approximate the solution of the initial-value problem. Because of the nonlinear behavior, 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. Several computer simulations are used to verify the performance of the controller. In particular, we present a PID controller to simulate how we would balance the two-links on a moving robot to any specific angle including upside-down.

Keywords

Two-link robot manipulator, Robotics, PID controller, Dynamic control, Simulation

Introduction

Robotic manipulators are a major component in the manufacturing industry. 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. In robotics, one of the most difficult tasks is to perform a precise and fast movement of a robotic arm. In this paper, one control model is proposed to solve

this problem.

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. Two-link manipulators are two-degree-

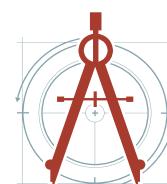
*Corresponding author: Mahboub Baccouch, Professor, Department of Mathematics, University of Nebraska at Omaha, USA, Tel: 402-554-4016, Fax: 402-554-2975

Accepted: December 08, 2020; Published: December 10, 2020

Copyright: © 2020 Baccouch M, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Baccouch and Dodds. Int J Robot Eng 2020, 5:028

Citation: Baccouch M, Dodds S (2020) A Two-Link Robot Manipulator: Simulation and Control Design. Int J Robot Eng 5:028



of-freedom robots. They are useful because they function like human arms. We can begin to understand the complex movement of human arms after defining the movement of the two-link manipulator [1]. The total potential and kinetic energies of the two-link system are defined and used to form the Lagrangian. The Euler-Lagrangian equation was used to define the torques applied to each link [2]. A proportional-integral-derivative (PID) controller is implemented to simulate the behavior of the system when torques are applied [3]. The PID controllers are the most common used controllers, due to their easy implementation and versatility. PID controllers remain the standard for industry and research given their flexibility and ease of use in both simulation and hardware systems [4-6].

Due to the nonlinear nature of the two-link manipulator, specialized control systems are often required to maintain a position or perform a task [7]. Researchers have found success using Sliding Mode Control compared to PID [8]. Others have used Artificial Neural Networks and Holographic Neural Networks as part of their PID feedback loop to perform the task of folding paper [9,10]. The proposed solution uses the command `ode45` in MATLAB to solve for the angle of each link relative to the positive X-axis. PID control is developed and considered to show how you can force the system to specific angles. We also show graphical results of the system after the PID control is applied.

In this paper, we propose a PID controller for a double pendulum system. We derive the equations of motion for a two-link robot manipulator based on Lagrangian Formulation. The equation of motion for two link robot is a nonlinear differential equation. As the closed form solutions are not available we have to use numerical solution. After deriving the equation of motion, control simulation is represented using MATLAB. We expand upon the standard two-link manipulator problem, to that of an upside-down pendulum with multiple links on top of a moving object. In this case we look to implement the control system into a mobile robot called the Computer and Electronics Engineering Robot (CEEnBot).

The structure of this paper is as follows. In section 4, we derive the equations of motion for a two-link robotic manipulator using the Lagrangian method. We also describe how to obtain the numerical solution using the `ode45` command defined in MATLAB to solve the system of ordinary differential equations numerically. We explain the design of the PID controller for the double pendulum system in section 5. In section 6, we present several numerical examples to test the PID controller. Finally, we provide some concluding remarks in section 7.

Dynamics of Two-Link Manipulators

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 as a whole. We consider the two-link robotic manipulator which is a classic example studied in introductory robotics courses. 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 angles 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 .

The first step in deriving the equations of motion using the Lagrangian approach is to find the kinetic energy KE and the potential energy PE of the system.

The equations for the x-position and the y-position of M_1 are given by

$$x_1 = L_1 \cos(\theta_1), \quad y_1 = L_1 \sin(\theta_1). \quad (4.1)$$

Similarly, the equations for the x-position and the y-position of M_2 are given by

$$x_2 = L_1 \cos(\theta_1) + L_2 \cos(\theta_2), \quad y_2 = L_1 \sin(\theta_1) + L_2 \sin(\theta_2) \quad (4.2)$$

Next, we calculate the velocities of M_1 and M_2 using the following formulas

$$v_1 = \sqrt{\dot{x}_1^2 + \dot{y}_1^2}, \quad v_2 = \sqrt{\dot{x}_2^2 + \dot{y}_2^2}, \quad (4.3)$$

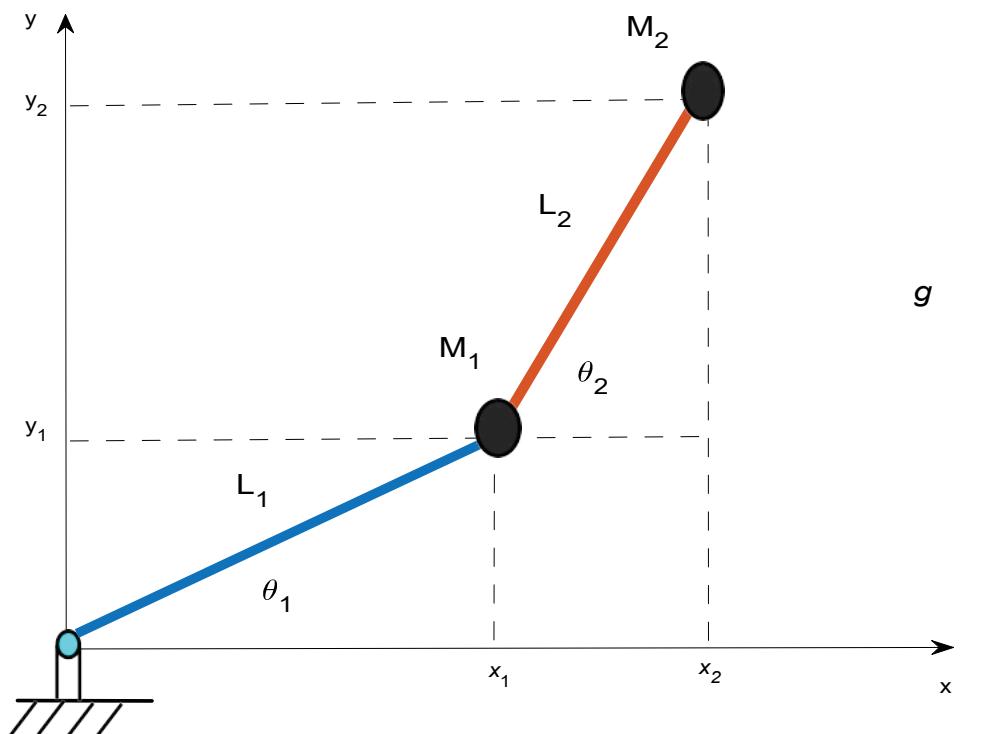


Figure 1: A simplified model of a two-link planar robot manipulator.

where,

$$\dot{x}_1 = -L_1 \dot{\theta}_1 \sin(\theta_1), \quad \dot{y}_1 = L_1 \dot{\theta}_1 \cos(\theta_1),$$

$$\dot{x}_2 = -L_1 \dot{\theta}_1 \sin(\theta_1) - L_2 \dot{\theta}_2 \sin(\theta_2), \quad \dot{y}_2 = L_1 \dot{\theta}_1 \cos(\theta_1) + L_2 \dot{\theta}_2 \cos(\theta_2)$$

Here and below the dot · is a derivative with respect to t , i.e., $\dot{\theta}_k = \frac{d\theta_k}{dt}$, $\dot{x}_k = \frac{dx_k}{dt}$, for $k = 1, 2$.

The kinematic energy can be calculated as follows

$$KE = \frac{1}{2}M_1v_1^2 + \frac{1}{2}M_2v_2^2. \quad (4.4)$$

The equation for the kinetic energy can be written as

$$KE = \frac{1}{2}M_1(\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2}M_2(\dot{x}_2^2 + \dot{y}_2^2). \quad (4.5)$$

Substituting (4.1) - (4.2) into (4.5), we get

$$KE = \frac{1}{2}M_1\left(\left(-L_1 \dot{\theta}_1 \sin(\theta_1)\right)^2 + \left(L_1 \dot{\theta}_1 \cos(\theta_1)\right)^2\right) + \frac{1}{2}M_2\left(\left(-L_1 \dot{\theta}_1 \sin(\theta_1) - L_2 \dot{\theta}_2 \sin(\theta_2)\right)^2 + \left(L_1 \dot{\theta}_1 \cos(\theta_1) + L_2 \dot{\theta}_2 \cos(\theta_2)\right)^2\right),$$

which can be simplified as

$$KE = \frac{1}{2}(M_1 + M_2)L_1^2\dot{\theta}_1^2 + \frac{1}{2}M_2L_2^2\dot{\theta}_2^2 + M_2L_1L_2\dot{\theta}_1\dot{\theta}_2 \cos(\theta_1 - \theta_2). \quad (4.6)$$

In order to calculate the Lagrangian, the potential energy PE has to be calculated. By definition the potential energy of the system due to gravity of the i^{th} pendulum is

$$PE_i(\theta) = M_i g h_i(\theta), \quad i=1, 2, \quad (4.7)$$

where h_i is the height of the center of mass of the i^{th} pendulum, g is the acceleration due to gravity

constant, and M_i is the mass of the i^{th} pendulum. Therefore, the total potential energy for both pendulums can be given by

$$\begin{aligned} PE = PE(\theta) &= \sum_{i=1}^2 PE_i(\theta) = \sum_{i=1}^2 M_i g h_i(\theta) = M_1 g L_1 \sin(\theta_1) + M_2 g (L_1 \sin(\theta_1) + L_2 \sin(\theta_2)) \\ &= (M_1 + M_2) g L_1 \sin(\theta_1) + M_2 g L_2 \sin(\theta_2) \end{aligned} \quad (4.8)$$

Next, by Lagrange Dynamics, we form the Lagrangian \mathcal{L} which is defined as

$$\mathcal{L} = KE - PE \quad (4.9)$$

Substituting the expressions for the kinetic energy (4.6) and potential energy (4.8) in for KE and PE we get

$$\begin{aligned} \mathcal{L} &= \frac{1}{2}(M_1 + M_2)L_1^2\dot{\theta}_1^2 + \frac{1}{2}M_2L_2^2\dot{\theta}_2^2 + M_2L_1L_2\dot{\theta}_1\dot{\theta}_2 \cos(\theta_1 - \theta_2) - (M_1 + M_2) \\ &\quad gL_1 \sin(\theta_1) - M_2gL_2 \sin(\theta_2). \end{aligned} \quad (4.10)$$

The Euler-Lagrange equation is given by the equation

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i} \right] - \frac{\partial \mathcal{L}}{\partial \theta_i} = \tau_i, \quad i = 1, 2, \quad (4.11)$$

where τ_i is the torque applied to the i^{th} link.

The derivations for the Lagrangian equation and Euler-Lagrange equation can be found in [1]. We were more concerned with using the formulas in this problem rather than going into too much in detail for these derivations.

From (4.10), we have

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i} = (M_1 + M_2)L_1^2\dot{\theta}_1 + M_2L_1L_2\dot{\theta}_2 \cos(\theta_1 - \theta_2),$$

$$\frac{\partial \mathcal{L}}{\partial \theta_1} = -M_2L_1L_2\dot{\theta}_1\dot{\theta}_2 \sin(\theta_1 - \theta_2) - (M_1 + M_2)gL_1 \cos(\theta_1),$$

and

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i} \right] = (M_1 + M_2)L_1^2\ddot{\theta}_1 + M_2L_1L_2\ddot{\theta}_2 \cos(\theta_1 - \theta_2) - M_2L_1L_2\dot{\theta}_2(\dot{\theta}_1 - \dot{\theta}_2)\sin(\theta_1 - \theta_2).$$

Similarly, we compute

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}_2} = M_2L_2^2\dot{\theta}_2 + M_2L_1L_2\dot{\theta}_1 \cos(\theta_1 - \theta_2),$$

$$\frac{\partial \mathcal{L}}{\partial \theta_2} = M_2L_1L_2\dot{\theta}_1\dot{\theta}_2 \sin(\theta_1 - \theta_2) - M_2gL_2 \cos(\theta_2),$$

and

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{\theta}_2} \right] = M_2L_2^2\ddot{\theta}_2 + M_2L_1L_2\ddot{\theta}_1 \cos(\theta_1 - \theta_2) - M_2L_1L_2\dot{\theta}_1(\dot{\theta}_1 - \dot{\theta}_2)\sin(\theta_1 - \theta_2).$$

Therefore, (4.11) gives the following two nonlinear equations of motion which are second-order system of ordinary differential equations

$$\begin{aligned} (M_1 + M_2)L_1^2\ddot{\theta}_1 + M_2L_1L_2\ddot{\theta}_2 \cos(\theta_1 - \theta_2) + \\ M_2L_1L_2\dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + (M_1 + M_2)gL_1 \cos(\theta_1) = \tau_1, \end{aligned} \quad (4.12a)$$

$$M_2 L_2^2 \ddot{\theta}_2 + M_2 L_1 L_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) - M_2 L_1 L_2 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + M_2 g L_2 \cos(\theta_2) = \tau_2, \quad (4.12b)$$

or equivalently,

$$L_1 \ddot{\theta}_1 + \delta L_1 L_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) = \frac{\delta \tau_1}{M_2 L_1} - \delta L_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) - g \cos(\theta_1), \quad (4.13a)$$

$$L_2 \ddot{\theta}_2 + L_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) = \frac{\tau_2}{M_2 L_2} + L_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) - g \cos(\theta_2), \quad (4.13b)$$

where, $\delta = \frac{M_2}{M_1 + M_2}$.

Solving for $\ddot{\theta}_1$ and $\ddot{\theta}_2$, we get the normal form of the dynamics equations

$$\ddot{\theta}_1 = g_1(t, \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2), \quad \ddot{\theta}_2 = g_2(t, \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2), \quad (4.14a)$$

where,

$$g_1 = \frac{\frac{\delta \tau_1}{M_2 L_1} - \delta L_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) - g \cos(\theta_1) - \delta \cos(\theta_1 - \theta_2) \left(\frac{\tau_2}{M_2 L_2} + L_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) - g \cos(\theta_2) \right)}{L_1 (1 - \delta \cos^2(\theta_1 - \theta_2))}, \quad (4.14b)$$

$$g_2 = \frac{\frac{\tau_2}{M_2 L_2} + L_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) - g \cos(\theta_2) - \cos(\theta_1 - \theta_2) \left(\frac{\delta \tau_1}{M_2 L_1} + \delta L_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) - g \cos(\theta_1) \right)}{L_2 (1 - \delta \cos^2(\theta_1 - \theta_2))}. \quad (4.14c)$$

In order to solve for the angles θ_1 and θ_2 , we need to solve the above second-order system of ordinary differential equations. To do this, we first reduce the system into an equivalent system of first-order ordinary differential equations.

Let us introduce four new variables

$$u_1 = \theta_1, u_2 = \theta_2, u_3 = \dot{\theta}_1, u_4 = \dot{\theta}_2. \quad (4.15)$$

After differentiating, we have

$$\dot{u}_1 = \dot{\theta}_1 = u_3, \quad \dot{u}_2 = \dot{\theta}_2 = u_4, \quad \dot{u}_3 = \ddot{\theta}_1 = g_1(t, u_1, u_2, u_3, u_4), \quad \dot{u}_4 = \ddot{\theta}_2 = g_2(t, u_1, u_2, u_3, u_4)$$

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

$$\frac{d\mathbf{U}}{dt} = \mathbf{S}(t, \mathbf{U}), \quad \mathbf{U}(0) = \mathbf{U}_0, \quad (4.16a)$$

Where $\mathbf{U} = [u_1, u_2, u_3, u_4]^T$ and $\mathbf{S} = [s_1, s_2, s_3, s_4]^T$ with

$$s_1 = u_3, \quad s_2 = u_4, \quad s_3 = g_1(t, u_1, u_2, u_3, u_4), \quad s_4 = g_2(t, u_1, u_2, u_3, u_4)$$

The initial conditions are given by

$$\mathbf{U}_0 = [u_1(0), u_2(0), u_3(0), u_4(0)]^T, \quad (4.16b)$$

Where,

$$u_1(0) = \theta_1(0), \quad u_2(0) = \theta_2(0), \quad u_3(0) = \dot{\theta}_1(0), \quad u_4(0) = \dot{\theta}_2(0).$$

The system (4.16a) subject to the initial condition (4.16b) can be solved for the unknown vector \mathbf{U} , using, for example, the **ode45** command defined in MATLAB to solve the system of ordinary differential equations numerically. This command is based on the fourth-order Runge-Kutta method. For more details consult [2,11].

Table 1: The PID terms and their effect on a control system.

Term	Math Function	Effect on Control System
P Proportional	$K_p V_{error}$	Typically the main drive in a control loop, K_p reduces a large part of the overall error.
I Integral	$K_I \int V_{error} dt$	Reduces the final error in a system. Summing even a small error over time produces a drive signal large enough to move the system toward a smaller error.
D Derivative	$k_D \frac{dV_{error}}{dt}$	Counteracts the K_p and K_I terms when the output changes quickly. This helps reduce overshoot and ringing. It has no effect on the final error.

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 θ_i 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 $V_{error} = V_{set} - V_{sensor}$ that takes the position we want to go (V_{set}) minus the position we are actually at (V_{sensor}). We get the proportional part of the PID control by taking a constant defined as K_p and multiplying it by the error. The I comes from taking a constant K_I and multiplying it by the integral of the error with respect to time. Derivative control is defined as a constant K_D multiplied by the derivative of the error with respect to time. Many industrial processes are controlled using PID controllers. Below is a table describing PID control **Table 1**.

The equations of motion (4.12) can be written compactly as

$$M(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + c(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + G(\boldsymbol{\theta}) = \mathbf{F}, \quad (5.1)$$

where,

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

$$c(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \begin{bmatrix} M_2L_1L_2\dot{\theta}_2^2 \sin(\theta_1 - \theta_2) \\ -M_2L_1L_2\dot{\theta}_1^2 \sin(\theta_1 - \theta_2) \end{bmatrix}, G(\boldsymbol{\theta}) = \begin{bmatrix} (M_1 + M_2)gL_1 \cos(\theta_1) \\ M_2gL_2 \cos(\theta_2) \end{bmatrix}, = \mathbf{F} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix},$$

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

$$\ddot{\boldsymbol{\theta}} = -M^{-1}(\boldsymbol{\theta})[c(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + G(\boldsymbol{\theta})] + \hat{\mathbf{F}}, \quad (5.2)$$

Where

$$\hat{\mathbf{F}} = M^{-1}(\boldsymbol{\theta})\mathbf{F}.$$

Thus, we decoupled the system to have the new input

$$\hat{\mathbf{F}} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}.$$

However, the physical torque inputs to the system are

$$\mathbf{F} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = M(\boldsymbol{\theta}) \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}. \quad (5.3)$$

Let us denote the error signals by

$$e(\theta_1) = \theta_{1f} - \theta_1, \quad e(\theta_2) = \theta_{2f} - \theta_2,$$

Where the target positions of M_1 and M_2 are given by the angles θ_{1f} and θ_{2f} , respectively.

We assume that the system has initial positions

$$\boldsymbol{\theta}_0 = \begin{bmatrix} \theta_1(0) \\ \theta_2(0) \end{bmatrix}.$$

A common technique for controlling a system with input is to use the following general structure of PID controller

$$f = K_P e + K_D \dot{e} + K_I \int e dt. \quad (5.4)$$

In our situation, the technique for controlling the double pendulum system with inputs f_1 and f_2 is to employ two independent controllers, one for each link, as follows

$$\begin{aligned} f_1 &= K_{P_1} e_1(\theta_1) + K_{D_1} \dot{e}_1(\theta_1) + K_{I_1} \int e(\theta_1) dt = K_{P_1} (\theta_{1f} - \theta_1) - K_{D_1} \dot{\theta}_1 + K_{I_1} \int (\theta_{1f} - \theta_1) dt, \\ f_2 &= K_{P_2} e_2(\theta_2) + K_{D_2} \dot{e}_2(\theta_2) + K_{I_2} \int e(\theta_2) dt = K_{P_2} (\theta_{2f} - \theta_2) - K_{D_2} \dot{\theta}_2 + K_{I_2} \int (\theta_{2f} - \theta_2) dt, \end{aligned}$$

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

The complete system of equations with control is then

$$\ddot{\boldsymbol{\theta}} = -M^{-1}(\boldsymbol{\theta}) [c(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{G}(\boldsymbol{\theta})] + \hat{\mathbf{F}}, \quad (5.5)$$

Where

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

We would like to emphasize that the actual physical torques are

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

To implement the PID controller, we introduce the following new states

$$x_1 = \int e(\theta_1) dt, \quad x_2 = \int e(\theta_2) dt.$$

Differentiating with respect to t gives

$$\dot{x}_1 = e(\theta_1) = \theta_{1f} - \theta_1, \quad \dot{x}_2 = e(\theta_2) = \theta_{2f} - \theta_2.$$

The complete equations are

$$\dot{\boldsymbol{\theta}} = \theta_{1f} - \theta_1, \quad (5.6)$$

$$\dot{x}_1 = \theta_{1f} - \theta_1, \quad (5.7)$$

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = -M^{-1}(\boldsymbol{\theta}) [c(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{G}(\boldsymbol{\theta})] + \begin{bmatrix} K_{P_1} (\theta_{1f} - \theta_1) - K_{D_1} \dot{\theta}_1 + K_{I_1} x_1 \\ K_{P_2} (\theta_{2f} - \theta_2) - K_{D_2} \dot{\theta}_2 + K_{I_2} x_2 \end{bmatrix}, \quad (5.8)$$

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = M(\boldsymbol{\theta}) \begin{bmatrix} K_{P_1} (\theta_{1f} - \theta_1) - K_{D_1} \dot{\theta}_1 + K_{I_1} x_1 \\ K_{P_2} (\theta_{2f} - \theta_2) - K_{D_2} \dot{\theta}_2 + K_{I_2} x_2 \end{bmatrix}. \quad (5.9)$$

To discretize the above 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

$$u_1 = x_1, u_2 = x_2, u_3 = \theta_1, u_4 = \theta_2, u_5 = \dot{\theta}_1, u_6 = \dot{\theta}_2, \quad (5.10)$$

After differentiating, we have

$$\begin{aligned} \dot{u}_1 &= \dot{x}_1 = \theta_{1f} - u_3, \quad \dot{u}_2 = \dot{x}_2 = \theta_{2f} - u_4, \\ \dot{u}_3 &= \dot{\theta}_1 = u_5, \quad \dot{u}_4 = \dot{\theta}_2 = u_6, \\ \ddot{u}_5 &= \ddot{\theta}_1 = \phi(t, u_1, u_2, u_3, u_4, u_5, u_6), \quad \ddot{u}_6 = \ddot{\theta}_2 = \psi(t, u_1, u_2, u_3, u_4, u_5, u_6), \end{aligned}$$

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

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

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

A simple calculation shows that

$$\begin{aligned} \phi &= \frac{-M_2 L_2 u_6^2 \sin(u_3 - u_4) - (M_1 + M_2) g \cos(u_3) - M_2 \cos(u_3 - u_4)(L_1 u_5^2 \sin(u_3 - u_4) - g \cos(u_4))}{L_1(M_1 + M_2 - M_2 \cos^2(u_3 - u_4))} \\ &\quad + K_{P_1}(\theta_{1f} - u_3) - K_{D_1}u_5 + K_{I_1}u_1, \\ \psi &= \frac{\cos(u_3 - u_4)(M_2 L_2 u_6^2 \sin(u_3 - u_4) + (M_1 + M_2) g \cos(u_3)) + (M_1 + M_2)(L_1 u_5^2 \sin(u_3 - u_4) - g \cos(u_4))}{L_2(M_1 + M_2 - M_2 \cos^2(u_3 - u_4))} \\ &\quad + K_{P_2}(\theta_{2f} - u_4) - K_{D_2}u_6 + K_{I_2}u_2 \end{aligned}$$

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

$$\frac{d\mathbf{U}}{dt} = \mathbf{H}(t, \mathbf{U}), \quad \mathbf{U}(0) = \mathbf{U}_0, \quad (5.10)$$

Where $\mathbf{U} = [u_1, u_2, u_3, u_4, u_5, u_6]^t$ and $\mathbf{H} = [h_1, h_2, h_3, h_4, h_5, h_6]^t$ with

$$h_1 = \theta_{1f} - u_3, \quad h_2 = \theta_{2f} - u_4, \quad h_3 = u_5, \quad h_4 = u_6,$$

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

The initial conditions are given by

$$\mathbf{U}_0 = [u_1(0), u_2(0), u_3(0), u_4(0), u_5(0), u_6(0)]^t,$$

Where,

$$u_1(0) = x_1(0), \quad x_2(0) = \theta_2(0), \quad u_3(0) = \theta_1(0), \quad u_4(0) = \theta_2(0), \quad u_5(0) = \dot{\theta}_1(0), \quad u_6(0) = \dot{\theta}_2(0).$$

The system (3.10) can be solved for the unknown vector \mathbf{U} , using, for example, the **ode45** command defined in MATLAB to solve the system of ordinary differential equations numerically. This command is based on the fourth-order Runge-Kutta method. For more details consult [2,11].

Once we solve for, we obtain to torques using $\mathbf{U} = (u_1, u_2, u_3, u_4, u_5, u_6)^t$,

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = M(\boldsymbol{\theta}) \begin{bmatrix} K_{P_1}(\theta_{1f} - u_3) - K_{D_1}u_5 + K_{I_1}u_1 \\ K_{P_2}(\theta_{2f} - u_4) - K_{D_2}u_6 + K_{I_2}u_2 \end{bmatrix}, \quad (5.11)$$

or equivalently,

$$\begin{aligned}\tau_1 &= (M_1 + M_2)L_1^2(K_{P_1}(\theta_{1f} - u_3) - K_{D_1}u_5 + K_{I_1}u_1) \\ &\quad + M_2L_1L_2 \cos(u_3 - u_4)(K_{P_2}(\theta_{2f} - u_4) - K_{D_2}u_6 + K_{I_2}u_2), \\ \tau_2 &= M_2L_1L_2 \cos(u_3 - u_4)(K_{P_1}(\theta_{1f} - u_3) - K_{D_1}u_5 + K_{I_1}u_1) + M_2L_2^2(K_{P_2}(\theta_{2f} - u_4) - K_{D_2}u_6 + K_{I_2}u_2).\end{aligned}$$

Simulations Results

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 will be demonstrated using the following examples.

Example 1. We consider the simplified model of a two-link manipulator shown in Figure 1. In this experiment, we take the following parameters

$$M_1 = M_2 = 1, L_1 = 2, L_2 = 1. \quad (6.1)$$

The target positions (final positions) are $\begin{bmatrix} \theta_{1f} \\ \theta_{2f} \end{bmatrix} = \begin{bmatrix} \pi/2 \\ 0 \end{bmatrix}$.

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

$$\begin{bmatrix} \theta_1(0) \\ \theta_2(0) \end{bmatrix} = \begin{bmatrix} \pi/2 \\ \pi/2 \end{bmatrix}, \begin{bmatrix} \dot{\theta}_1(0) \\ \dot{\theta}_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The PID parameters for θ_1 and θ_2 , are taken as

$$KP_1 = 30, \quad KD_1 = 15, \quad KI_1 = 20,$$

$$KP_2 = 30, \quad KD_2 = 10, \quad KI_2 = 20.$$

In Figure 2, we show the initial and target positions of the two-link manipulator. Next, we plot the positions θ_1 and θ_2 of M_1 and M_2 versus time over the interval [0,30] in Figure 3.

In Figure 4 we present the difference between where we want to go and where we are actually at *i.e.*, $e(\theta_1) = \theta_{1f} - \theta_1$, $e(\theta_2) = \theta_{2f} - \theta_2$,

where θ_{1f} and θ_{2f} are the target positions and $\theta_1(t)$ and $\theta_2(t)$ are the numerical approximations obtained by solving the system of ODEs. These results indicate that the PID controller gets the final positions

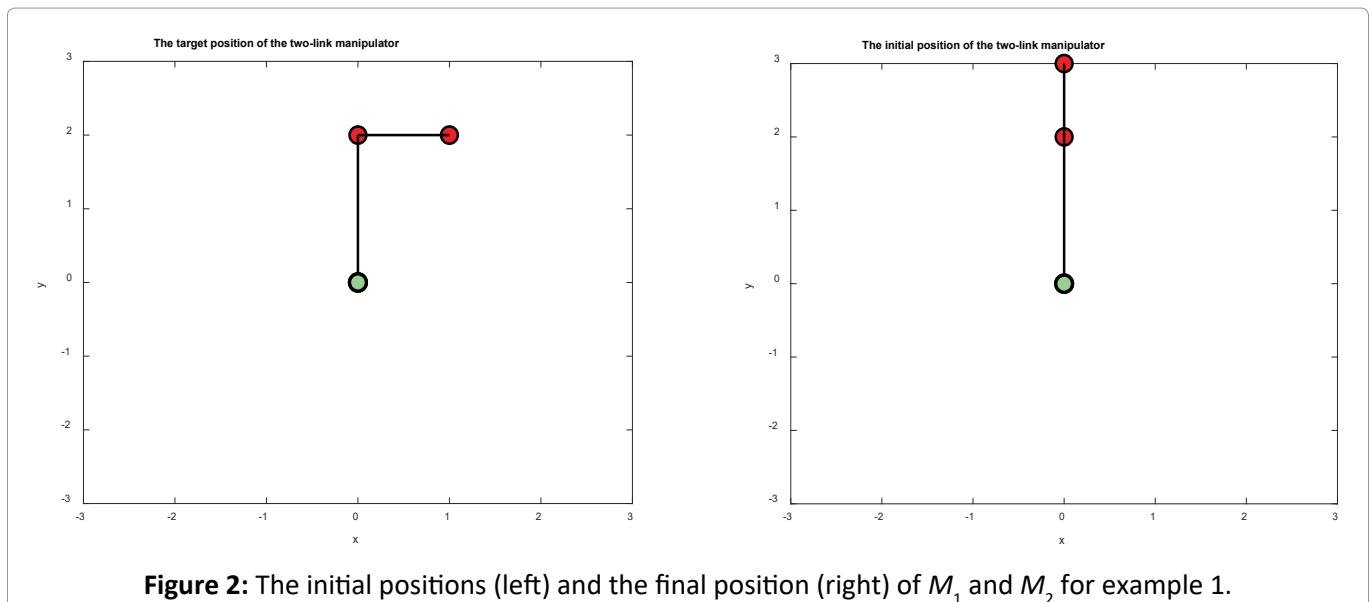


Figure 2: The initial positions (left) and the final position (right) of M_1 and M_2 for example 1.

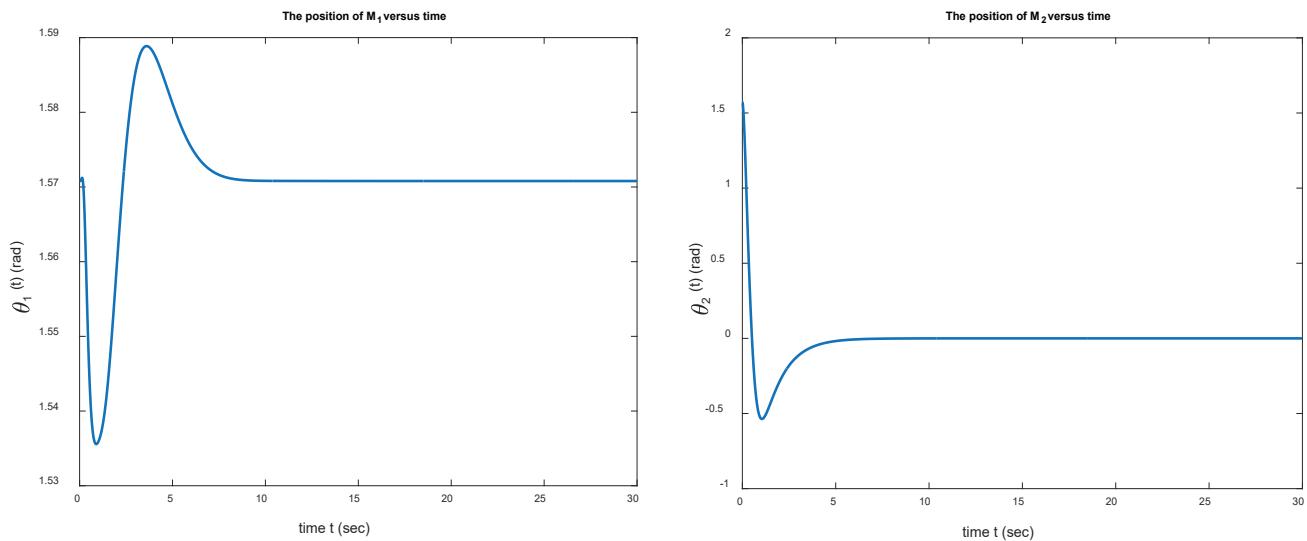


Figure 3: The positions of M_1 (left) and M_2 (right) versus time for example 1.

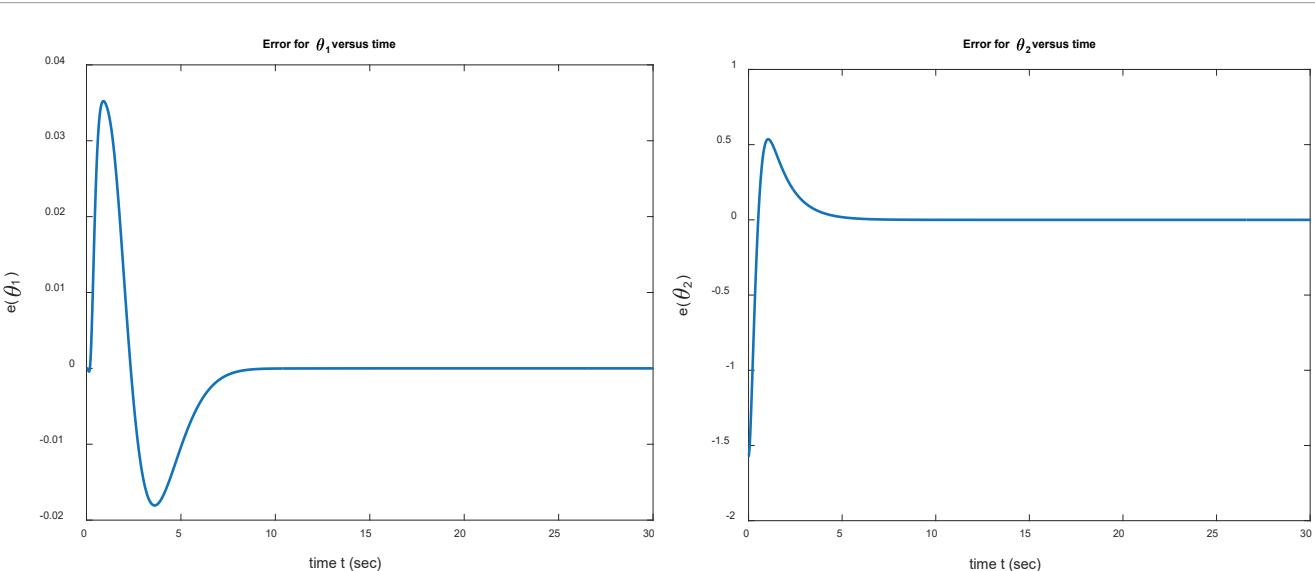


Figure 4: The errors $e(\theta_1) = \theta_1(t) - \theta_{1f}$ and $e(\theta_2) = \theta_2(t) - \theta_{2f}$ versus time for example 1.

of M_1 and M_2 determined by the angles θ_1 and θ_2 , in a short time (almost 7 seconds) with very small error.

Finally, in Figure 5, we show the torques f_1 and f_2 versus time

Note that the actual joints torques are

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

Example 2. We repeat the previous example with all parameters kept unchanged except for the final positions where we use

$$\begin{bmatrix} \theta_{1f} \\ \theta_{2f} \end{bmatrix} = \begin{bmatrix} \pi/4 \\ -\pi/4 \end{bmatrix}. \quad (6.2)$$

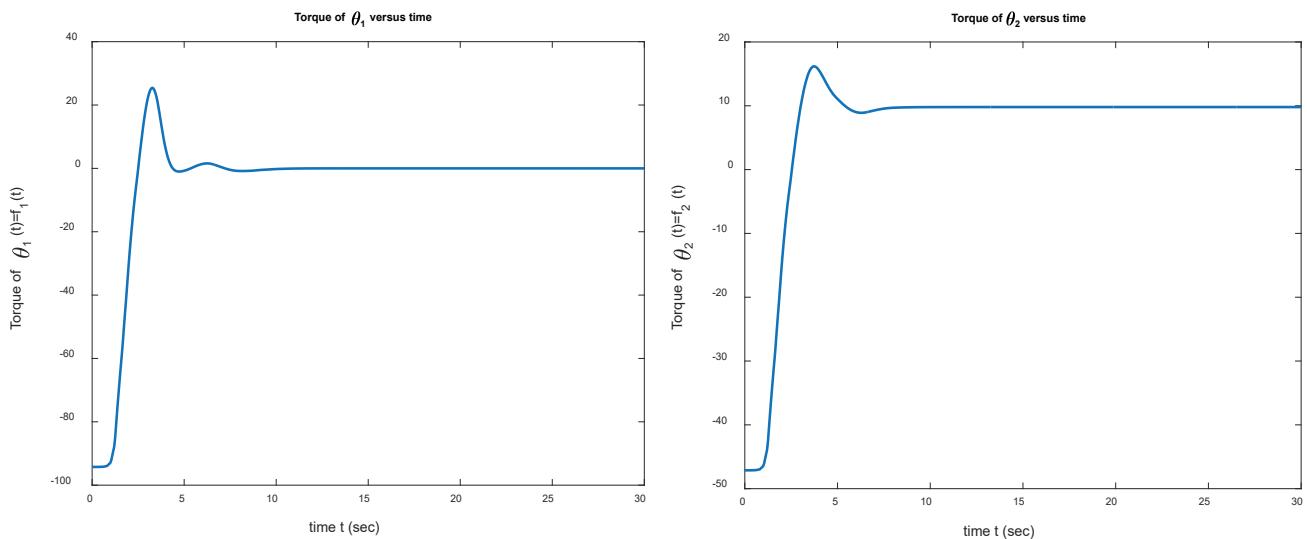


Figure 5: The torques of θ_1 (left) and θ_2 (right) versus time for example 1.

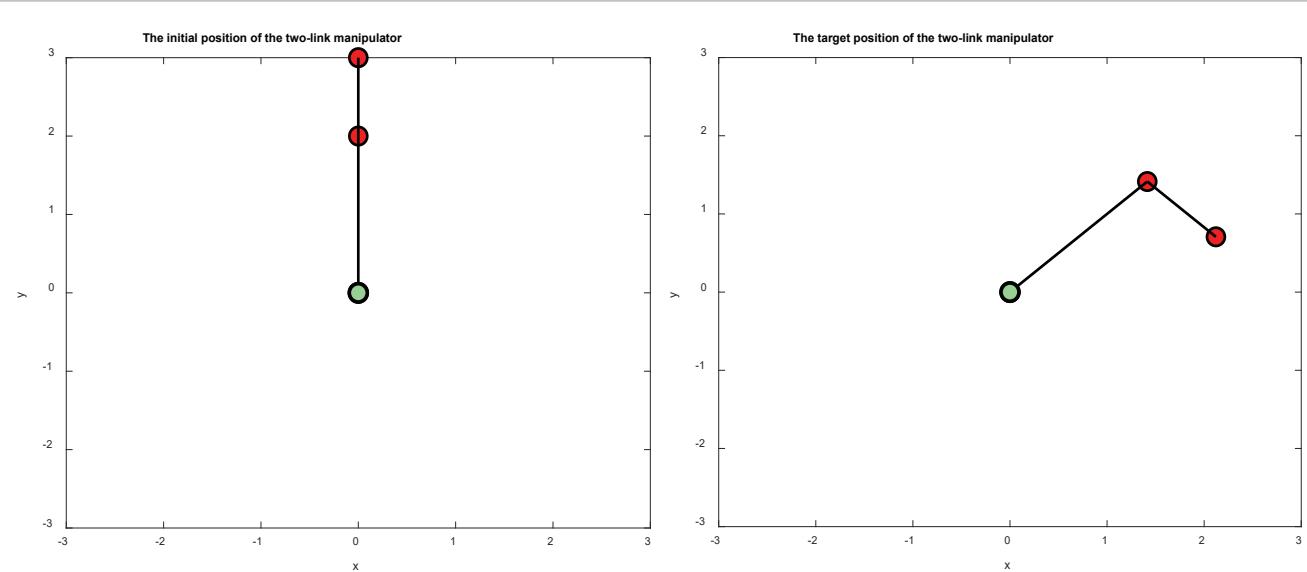


Figure 6: The initial positions (left) and the final position (right) of M_1 and M_2 for example 2.

The initial and target positions of M_1 and M_2 are shown in Figure 6. We present the positions θ_1 and θ_2 of M_1 and M_2 versus time in Figure 7. In Figure 8 we present the errors between the estimate positions and target positions. Finally, in Figure 9, we show the torques versus time. Once again, these results demonstrate that the PID controller gets the final positions of M_1 and M_2 in a short time (approximately 5 seconds).

Example 3. In this example, we consider the two-link manipulator shown in Figure 1 with the following parameters

$$M_1 = M_2 = 1, L_2 = 2, L_2 = 1. \quad (6.3)$$

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

$$\begin{bmatrix} \theta_1(0) \\ \theta_2(0) \end{bmatrix} = \begin{bmatrix} -\pi/2 \\ \pi/2 \end{bmatrix}, \begin{bmatrix} \dot{\theta}_1(0) \\ \dot{\theta}_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

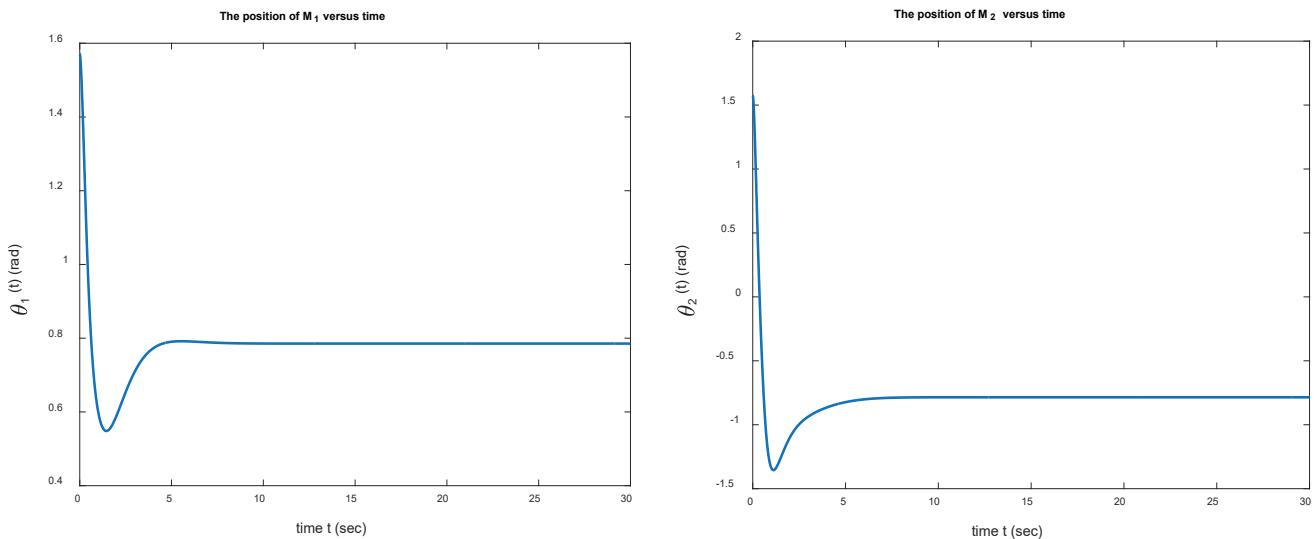


Figure 7: The positions of M_1 (left) and M_2 (right) versus time for example 2.

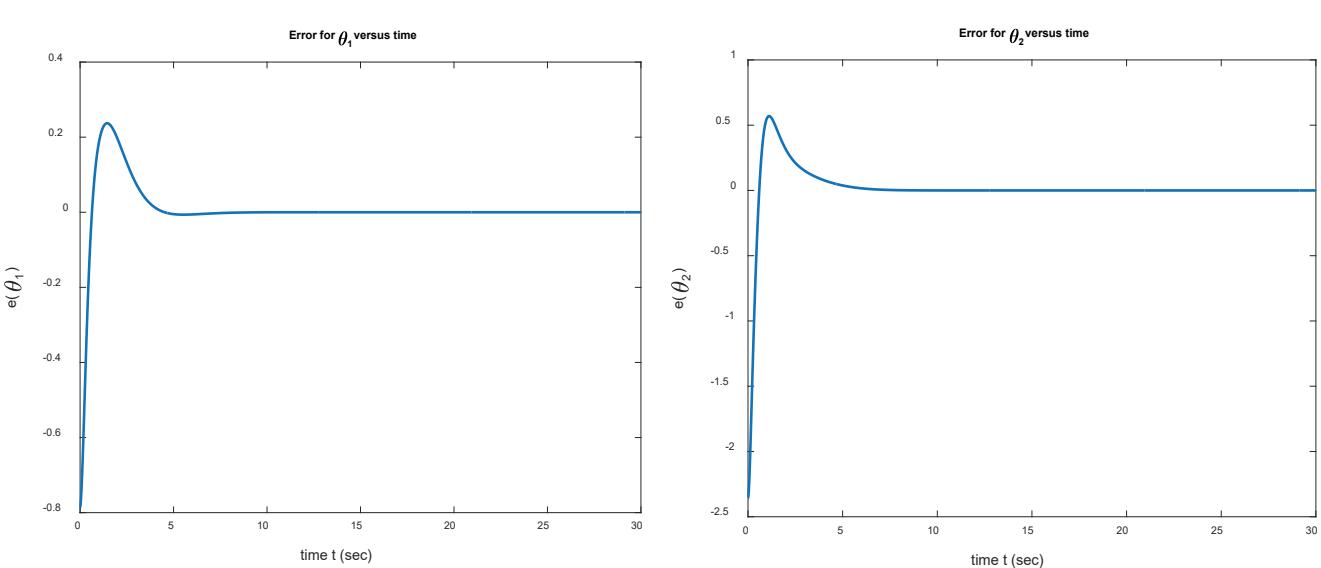


Figure 8: The errors $e(\theta_1) = \theta_1(t) - \theta_{1f}$ and $e(\theta_2) = \theta_2(t) - \theta_{2f}$ versus time for example 2.

The target positions (final positions) are $\begin{bmatrix} \theta_{1f} \\ \theta_{2f} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

The PID parameters for θ_1 and θ_2 , are taken as

$$\begin{aligned} KP_1 &= 30, & KD_1 &= 15, & KI_1 &= 20, \\ KP_2 &= 30, & KD_2 &= 20, & KI_2 &= 20. \end{aligned}$$

The initial and target positions of the two-link manipulator are shown in Figure 10. In Figure 11 we present the positions θ_1 and θ_2 of M_1 and M_2 versus time over the interval [0,30]. In Figure 12 we present the errors $e(\theta_1)$ and $e(\theta_2)$ between the target position and the estimate position found numerically. Again it takes only few seconds to get the target position. We would like to mention that the PID controller gets the final positions of M_1 and M_2 in a short time (approximately in seconds) with very small error. Finally, we show the torques f_1 and f_2 versus time in Figure 13.

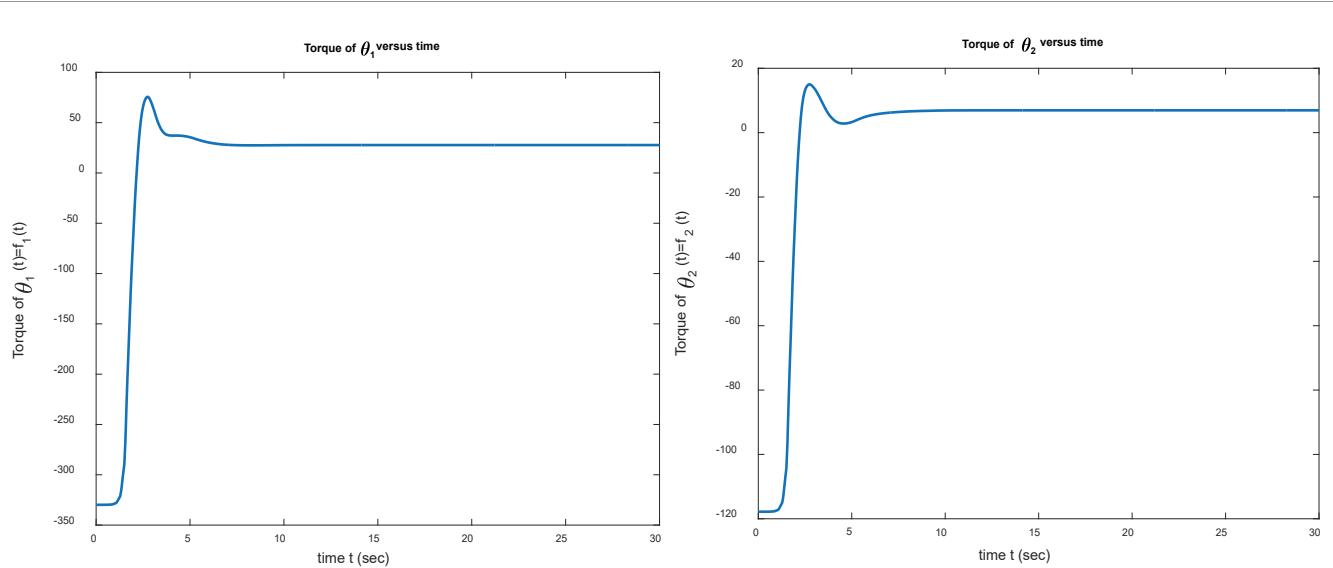


Figure 9: The torques of θ_1 (left) and θ_2 (right) versus time for example 2.

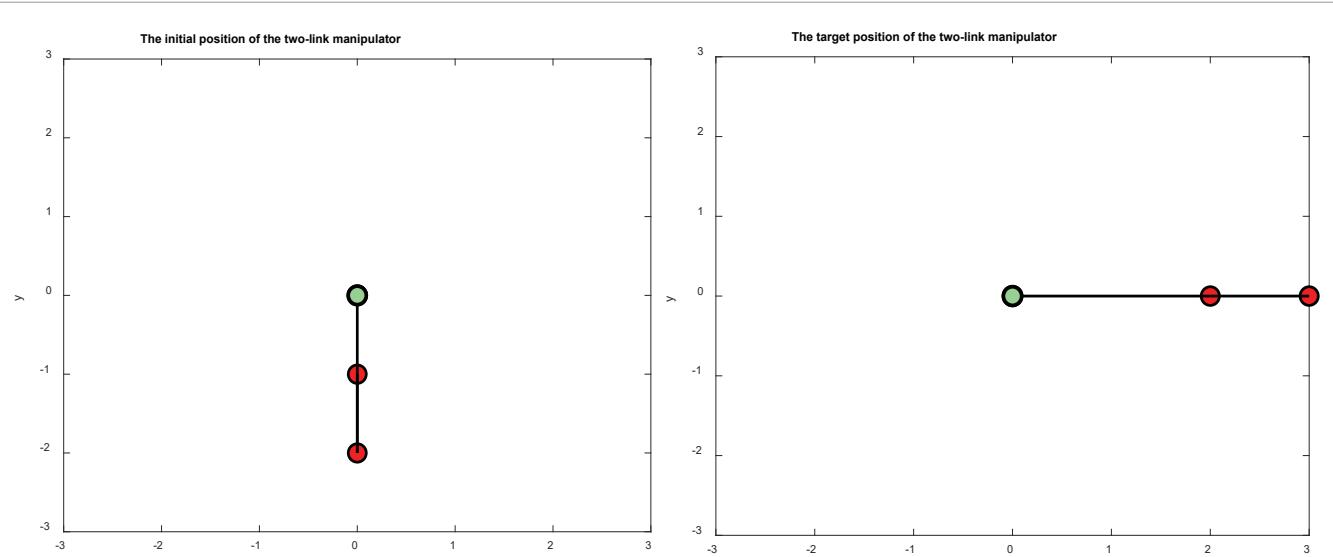


Figure 10: The initial positions (left) and the final position (right) of M_1 and M_2 for example 3.

Example 4. Finally, we consider the two-link manipulator shown in Figure 1 with the following parameters

$$M_1 = M_2 = 1, L_2 = 2, L_2 = 1. \quad (6.4)$$

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

$$\begin{bmatrix} \theta_1(0) \\ \theta_2(0) \end{bmatrix} = \begin{bmatrix} -\pi/2 \\ -\pi/2 \end{bmatrix}, \begin{bmatrix} \dot{\theta}_1(0) \\ \dot{\theta}_2(0) \end{bmatrix} = \begin{bmatrix} \pi/2 \\ \pi/2 \end{bmatrix}, \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The target positions (final positions) are

$$\begin{bmatrix} \theta_{1f} \\ \theta_{2f} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

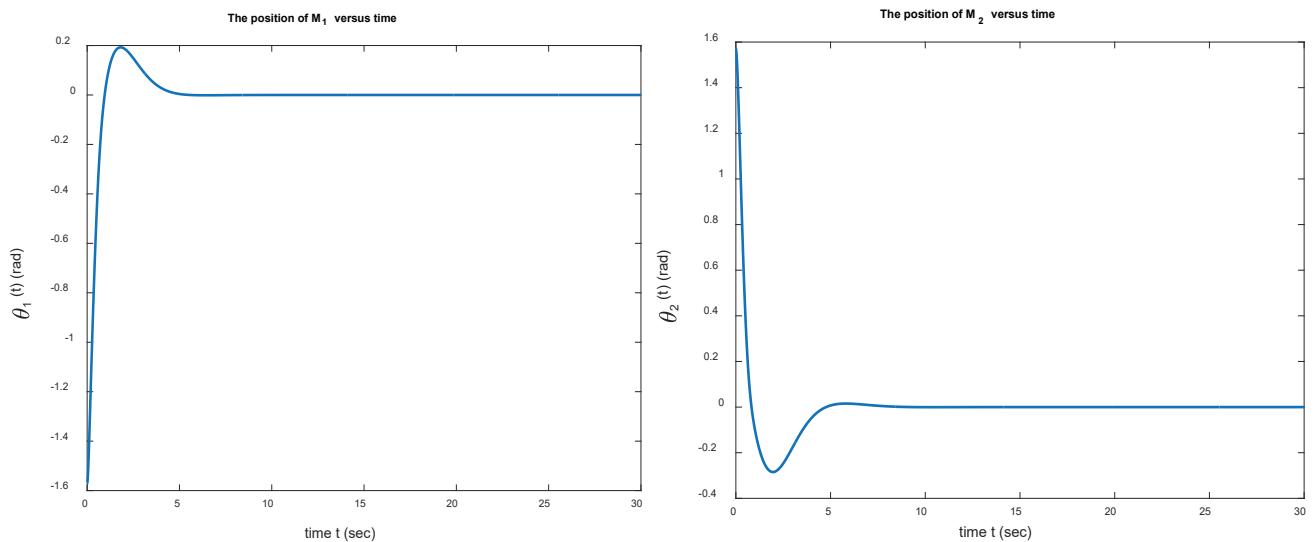


Figure 11: The positions of M_1 (left) and M_2 (right) versus time for example 3.

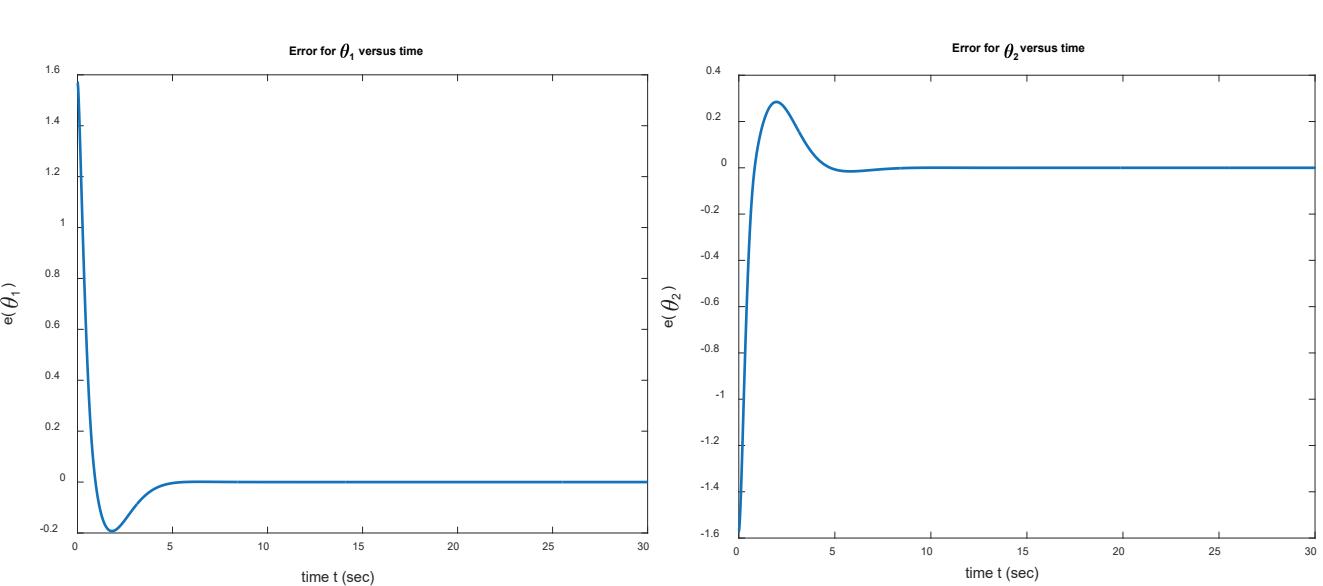


Figure 12: The errors $e(\theta_1) = \theta_1(t) - \theta_{1f}$ and $e(\theta_2) = \theta_2(t) - \theta_{2f}$ versus time for example 3.

The PID parameters for θ_1 and θ_2 are taken as

$$\begin{aligned} KP_1 &= 30, & KD_1 &= 15, & KI_1 &= 20, \\ KP_2 &= 30, & KD_2 &= 20, & KI_2 &= 20. \end{aligned}$$

The initial and target positions of the two-link manipulator are shown in Figure 14. The positions θ_1 and θ_2 of M_1 and M_2 versus time over the interval $[0, 30]$ are shown in Figure 15. In Figure 16 we plot the errors $e(\theta_1)$ and $e(\theta_2)$ between the target position and the approximate position found numerically. As in the previous examples, it takes only few seconds to get the target position. Finally, we present the torques f_1 and f_2 versus time in Figure 17.

Conclusion

In this paper, a two-link robotic manipulator was studied and its dynamics were modeled using Lagrange mechanics. First, we derived the dynamical equation of the two-link robot manipulators using

Citation: Baccouch M, Dodds S (2020) A Two-Link Robot Manipulator: Simulation and Control Design. Int J Robot Eng 5:028

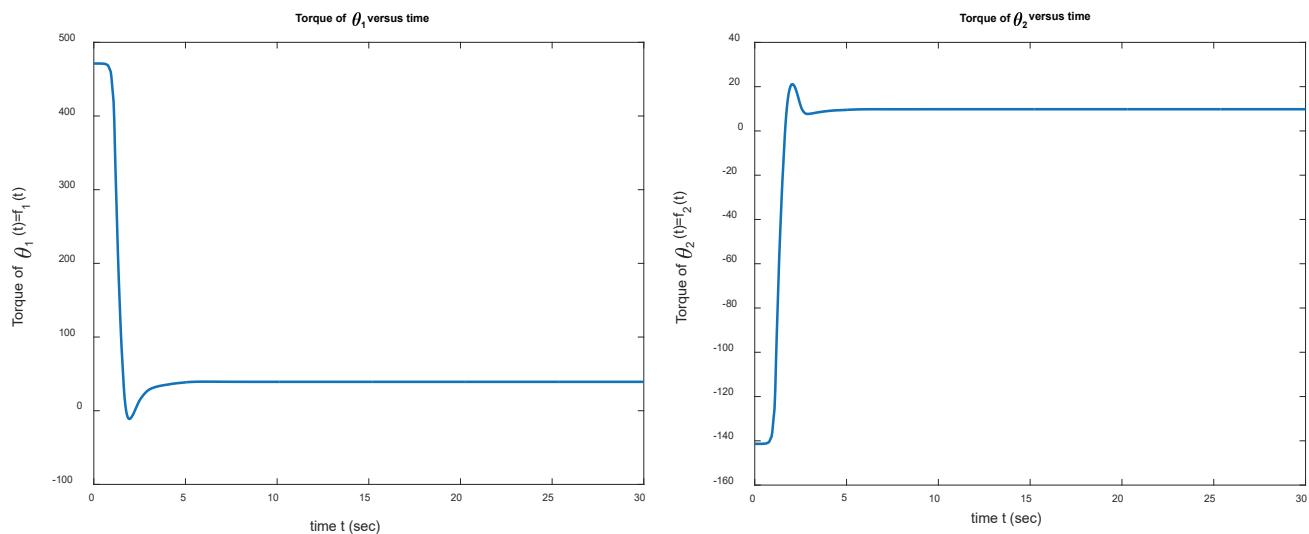


Figure 13: The torques of θ_1 (left) and θ_2 (right) versus time for example 3.

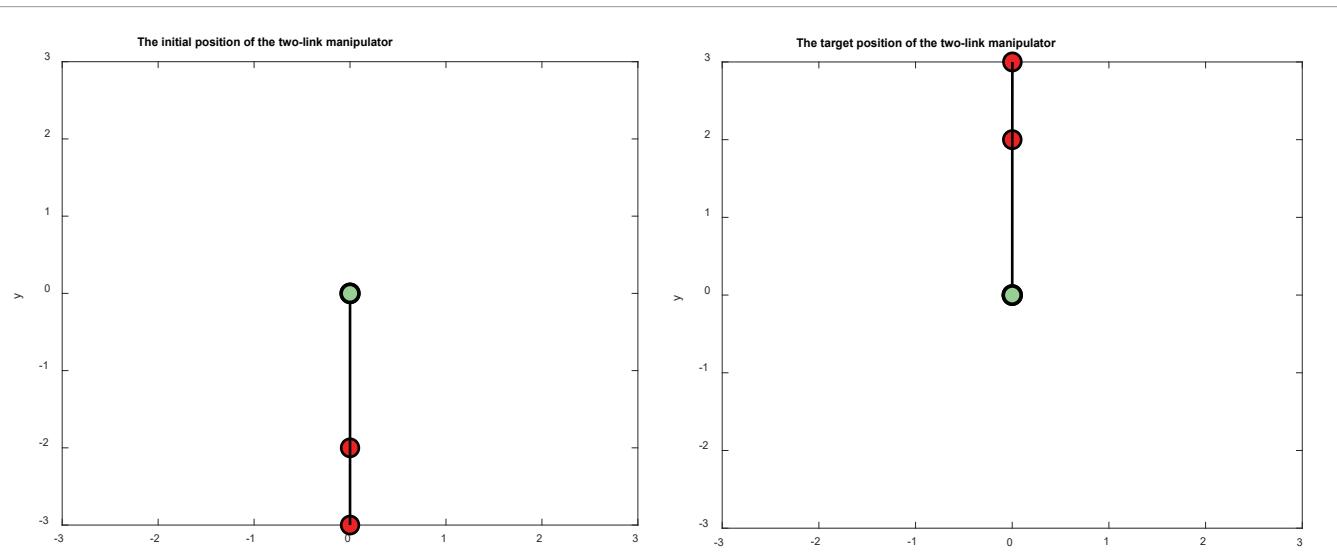


Figure 14: The initial positions (left) and the final position (right) of M_1 and M_2 for example 4.

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. The proposed research can be used to design and code control algorithms for a mobile robot. These algorithms allow us to balance upside-down pendulums on a mobile robot. There could be many solutions for coding this PID control depending on what sensors are used to find the angles of the links. In this study, we considered a planar robot having rotational joints. However, the PID controller presented in this paper can be extended for robots having prismatic joints. In the future, we are planning to consider complex joints such as a ball and socket joint.

Acknowledgement

This project was funded by the Kerrigan Research Minigrant Program. The authors would like to thank the anonymous reviewer for the valuable comments and suggestions which improved the quality of the paper.

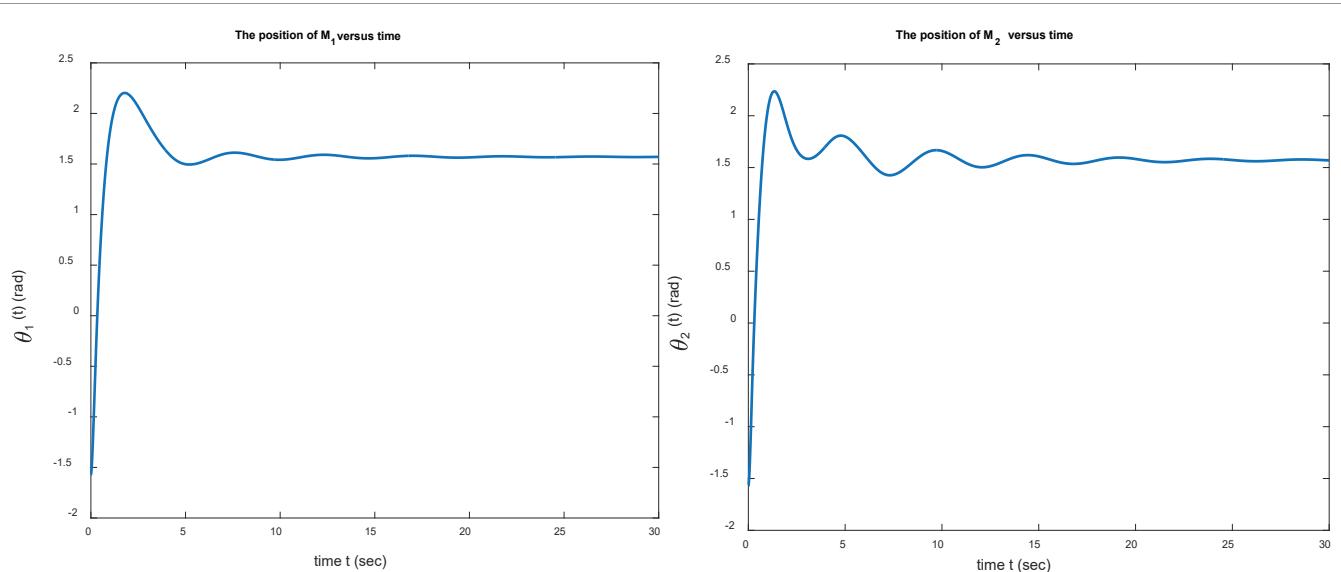


Figure 15: The positions of M_1 (left) and M_2 (right) versus time for example 4.

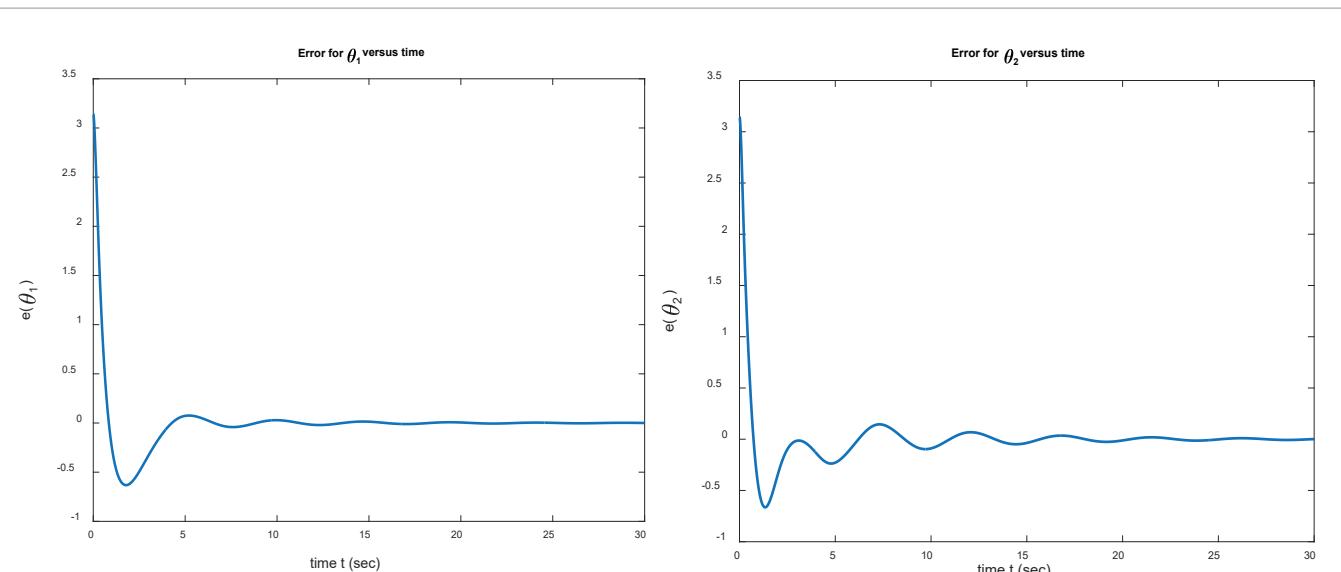


Figure 16: The errors $e(\theta_1) = \theta_1(t) - \theta_{1f}$ and $e(\theta_2) = \theta_2(t) - \theta_{2f}$ versus time for example 4.

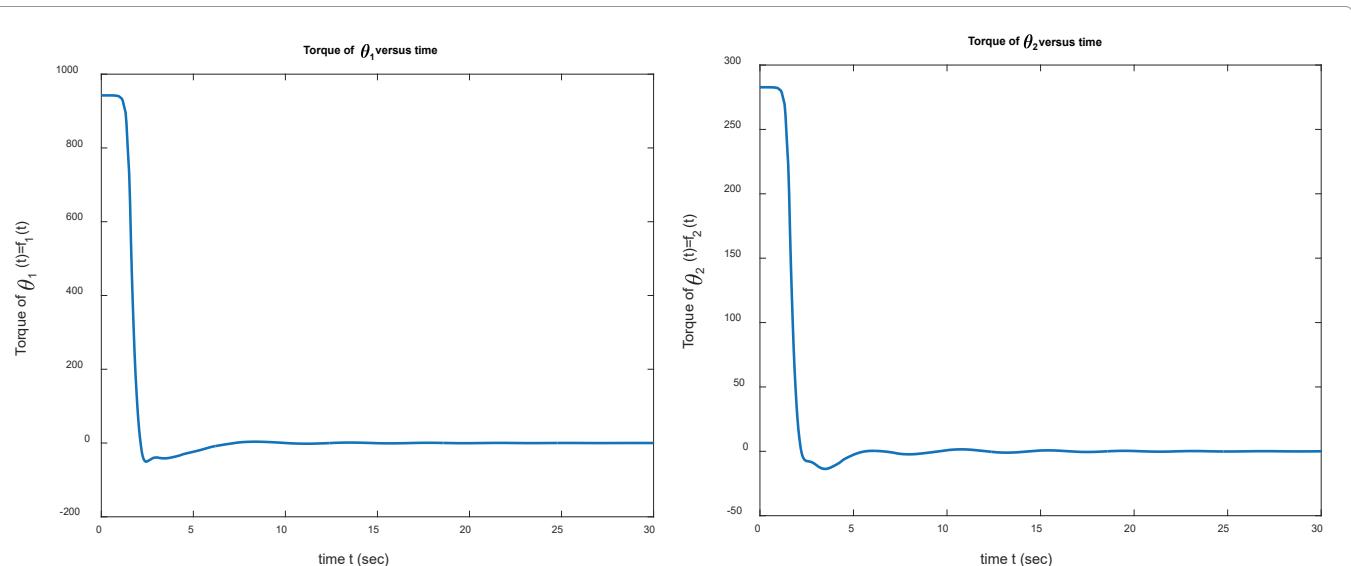


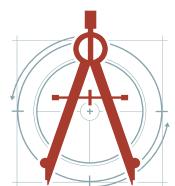
Figure 17: The torques of θ_1 (left) and θ_2 (right) versus time for example 4.

Conflict of Interest

The Authors declare that there is no conflict of interest.

References

1. RM Murray, Z Li, SS Sastry (1994) A mathematical introduction to robotic manipulation. CRC press.
2. RL Burden, JD Faires, AM Burden (2016) Numerical analysis. Cengage Learning, Boston, MA.
3. JYS Luh (1983) Conventional controller design for industrial robots a tutorial. IEEE Transactions on Systems, Man, and Cybernetics SMC-13: 298-316.
4. J Shah, S Rattan, B Nakra (2015) Dynamic analysis of two link robot manipulator for control design using computed torque control. International Journal of Research in Computer Applications and Robotics 3: 52-59.
5. G Nandy, B Chatterjee, A Mukherjee (2018) Dynamic analysis of two-link robot manipulator for control design. Advances in Communication, Devices and Networking, Springer, 767-775.
6. JS Kumar, EK Amutha (2014) Control and tracking of robotic manipulator using PID controller and hardware in loop simulation. International Conference on Communication and Network Technologies, IEEE.
7. A Mohammed, A Eltayeb (2018) Dynamics and control of a two-link manipulator using PID and sliding mode control. International Conference on Computer, Control, Electrical, and Electronics Engineering (ICC- CEEE), IEEE.
8. NK Chaturvedi, L Prasad (2018) A comparison of computed torque control and sliding mode control for a three link srobot manipulator. International Conference on Computing, Power and Communication Technologies (GU-COM), IEEE.
9. J Romero, L Diago, J Shinoda, I Hagiwara (2015) Evaluation of brain models to control a robotic origami arm using holographic neural networks. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 57137.
10. DM Wolpert, RC Miall, M Kawato (1998) Internal models in the cerebellum. Trends in cognitive sciences 2: 338-347.
11. CH Edwards, DE Penney, DT Calvis (2016) Differential equations and boundary value problems. Pearson Education Limited.



DOI: 10.35840/2631-5106/4128

Citation: Baccouch M, Dodds S (2020) A Two-Link Robot Manipulator: Simulation and Control Design. Int J Robot Eng 5:028