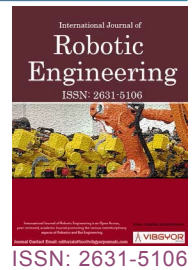


Proportional-Integral-Derivative Controller for an n -Link Revolute Robot Manipulator



Mahboub Baccouch* and Jerry Sun

Department of Mathematical and Statistical Sciences, University of Nebraska at Omaha, Omaha, 68182, NE, USA

Abstract

This paper aims to model and guide the motion of an n -link revolute robot arm manipulator by implementing an efficient proportional-integral-derivative (PID) control strategy. The n -link robot arm manipulator we examine shares similarities with a pendulum system comprising n arms and n masses. This system features a stationary point around which n distinct links, each carrying a mass at its end, are interconnected. We first derive the equations of motion using Lagrangian formulation. We then implement a PID controller for the robot manipulator, enabling it to attain designated target positions as required. These equations are represented by a second-order system of nonlinear ordinary differential equations. Due to the absence of closed-form solutions for the equations of motion, we employ the classical fourth-order Runge-Kutta method to approximate the solution of the initial-value problem. Due to the inherent nonlinear nature, achieving precise control over the motion of the n -link robot manipulator at user-defined positions poses a challenging endeavor. In light of this, our primary emphasis centers on controlling the robot manipulator to attain the desired position using the PID controller. Numerous computer simulations are conducted to validate the controller's performance. Notably, a PID controller is presented as part of the simulations, illustrating how we can achieve balance for the n links with $n = 3 - 6$ on a moving robot at various angles.

AMS subject classifications (2020): 65L05, 65L06, 70E60, 70B15, 70Q05, 93C85.

Keywords

n -Link robotic arm, Robot Manipulator, Dynamic equations, Lagrangian, PID controller

Introduction

The purpose of this paper is to design a nonlinear proportional-integral-derivative (PID) controller for trajectory tracking of a manipulator robot. Robotic arms play an essential role in various sectors, including manufacturing, transportation, and healthcare. They contribute significantly

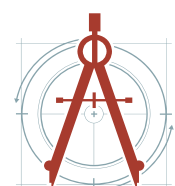
to enhancing human livelihoods, enabling their endeavors and innovations, ultimately leading to improved quality of life. An overview of commercially available robotic manipulators, sensors, and controllers can be found in [1]. The field of robotic manipulator control is a well-established and promising domain that encompasses research, development, and manufacturing. Industrial robots

*Corresponding author: Mahboub Baccouch, Department of Mathematical and Statistical Sciences, University of Nebraska at Omaha, Omaha, 68182, NE, USA

Accepted: June 13, 2024; Published: June 15, 2024

Copyright: © 2024 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 Sun. *Int J Robot Eng* 2024, 7:038



primarily function as positioning and handling devices. Hence, a valuable robot possesses the capability to control its motion along with the interactive forces and torques occurring between the robot and its surroundings.

Robot Manipulators are composed of links connected by joints into a kinematic chain. These joints predominantly consist of rotary components capable of achieving a diverse range of motions, thereby imparting flexibility and maneuvering capabilities to the robot. The n -link robot arm manipulator closely resembles an n -pendulum system with n arms and n masses, where there exists a fixed point about which n different links with a mass at the end of each are connected upon. Each adjacent mass and link combination forms its own simple pendulum such that the system contains n total simple pendulums, each conjoined with one another to form an n -pendulum system. When released from an arbitrary position without control, each pendulum is able to freely oscillate on the xy -plane. While the system itself is simple dynamically, the behavior exhibited by it is complex and nonlinear. In order to understand this behavior and define the motion of the system, we design a nonlinear PID controller for trajectory tracking of a manipulator robot.

Achieving precision and speed in the movements of a robot manipulator is critical in its efficiency. In this paper, we propose a proportional-integral-derivative (PID) control to achieve this. To define clearly designated parameters and to track the position of the system within those parameters we use a PID controller. The PID controller has three parts: a proportional (P) controller, an integral (I) controller, and a derivative (D) controller. This allows for the optimization of speed, stability, and precision. Subsequently, PID controllers are known to be reliable and efficient, considered a building block in industrial technology and the standard control in industrial robotics as well as research [2].

Other control systems have been examined as well for the n -link manipulator. An overview of various control theory concepts that are used in the control of robots can be found in [1]. In [3], the authors presented a numerical solution of the second-order robot arm control problem using the Runge-Kutta-Butcher algorithm. Research has also formulated using sets of stabilizing decentralized PID controls with Kharitonov's theorem for n -link manipulators to track positions and was successfully simulated on a two-link manipulator [4]. Recently, in [5], we designed a robust, fast, and practical PID controller for the classical double pendulum system. We first derived the equations of motion for the two-link robot manipulator using the Lagrangian approach. We used the classical fourth-order Runge-Kutta method to approximate the solution of the nonlinear system of second-order ordinary differential equations. We focused mainly on control of the robot manipulator to get the desired position using the PID control approach. In this paper, we extend the approach to an n -link robot manipulator for any n . More recently, an Adaptive Observer Based Neural Control for flexible-joint manipulators has been researched, utilizing the backstepping method with Lyapunov stability theory [6]. A Sliding Mode Control employed with the super-twisting method has also been successfully applied to robot manipulators, demonstrating fast error convergence and angular velocity approximation for each link [7]. Others have used the Lyapunov-based Control Scheme for linear n -link manipulators mounted on mobile sliders to navigate links [8].

In this paper, we propose a PID control for an n -link robot manipulator to guide its motion from an initial position to a target position on a plane in Figure 1. We first derive the total potential and kinetic energies for the n -link system. Then, we explicitly find the so-called Lagrangian. The Euler-Lagrangian equation [9] is then used to define the torques for each link. Our proposed PID controller uses the classical fourth order Runge-Kutta method [10] to approximate the angles and torques of the links relative to the x -axis from a system of nonlinear differential equations and initial conditions. The numerical approximations are performed by Java code. The system of differential equations our method relies on is based on the equations of motion for the n -link robot manipulator. These equations of motion are derived through Lagrangian dynamics, specifically from the Euler-Lagrange equation that is based on the total kinetic and potential energies of the n -link robot manipulator system. Through this control derivation, we are able to obtain the most precise and efficient guidance of the manipulator's motion to its designated target positions. This is demonstrated through subsequent simulations of the control that we perform.

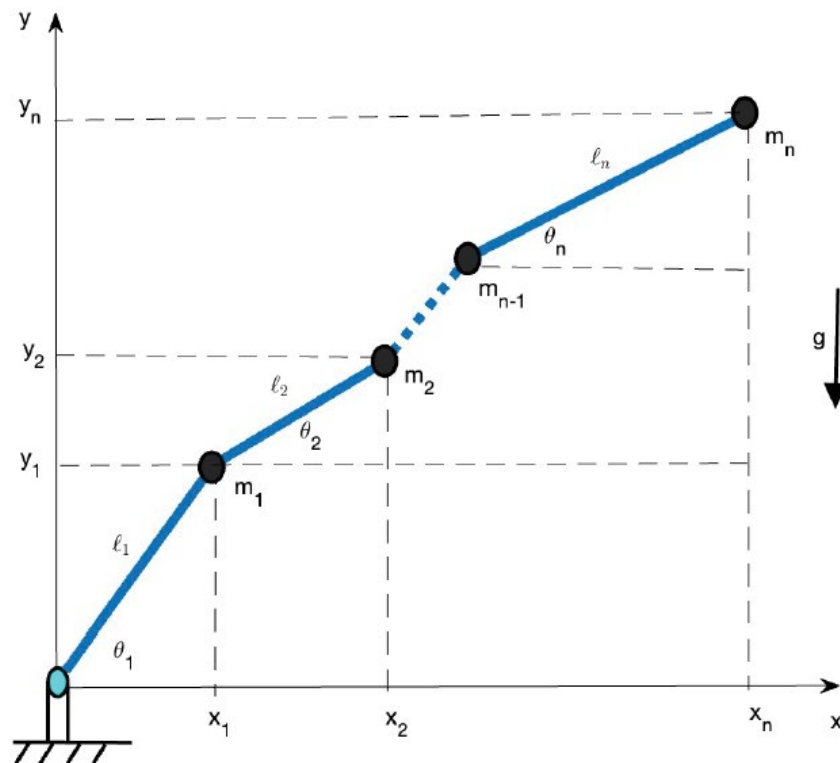


Figure 1: n -Link planar robot manipulator



Figure 2: The initial position (left) and the final position (right) for Example 6.1.

The paper is organized as follows. Section 2 presents the derivation of the equations of motion for the n -link robotic manipulator using the Lagrangian approach. Furthermore, the procedure for obtaining numerical solutions using the classical fourth-order Runge-Kutta method is detailed. The design of the PID controller for the n -pendulum system is expounded upon in Section 3. Section 4 provides a series of numerical examples intended to evaluate the performance of the PID controller. Concluding remarks are offered in Section 5.

Dynamics Equations

The robot manipulator is a device composed of multiple conjoined arms, joints, and links integrated with a control mechanism designed to manipulate objects and perform physical tasks without manual human intervention. Robot manipulators are consequently vital to the world of industry and everyday life in assembly, automation, manufacturing, logistics, and more.

A typical application involving an industrial manipulator with n revolute links is shown in Figure 2. This robot can be viewed as a pendulum made of n masses m_1, m_2, \dots, m_n and n rods of lengths l_1, l_2, \dots, l_n .

For our model problem, we consider a Cartesian coordinate system with origin placed at the base of the robot.

The equations of motion

We may understand the n -link robot manipulator as an n -link pendulum system on a two-dimensional plane. This system consists of n weightless bars, having lengths of $\ell_1, \ell_2, \dots, \ell_n$. There exists a mass at the end of each bar such that there are n masses in total connected by the bars. Denote these n masses as m_1, m_2, \dots, m_n . Each bar may rotate freely, where the first bar rotates about the origin and each subsequent bar rotates about the endpoint of its preceding bar, producing n degrees of freedom. The angles for each of these degrees of freedom produced by the bar relative to the x -axis may be denoted as $\vartheta_1, \vartheta_2, \dots, \vartheta_n$. Such a system exhibits chaotic motion that may be studied through Lagrangian dynamics. In order to derive the equations of motion through Lagrangian dynamics, we must first solve for the kinetic energy and the potential energy of the system.

Let (x_i, y_i) denote the position of the mass m_i for each $i = 1, 2, \dots, n$. The position (x_i, y_i) of m_i at time t is given by the equations:

$$x_i = \sum_{j=1}^i \ell_j \cos \theta_j, \quad y_i = \sum_{j=1}^i \ell_j \sin \theta_j, \quad i = 1, 2, \dots, n. \quad (4.1)$$

The velocity of mass m_i , denoted as v_i is given by

$$v_i = \sqrt{\dot{x}_i^2 + \dot{y}_i^2}, \quad i = 1, 2, \dots, n, \quad (4.2)$$

where \dot{x}_i and \dot{y}_i are, respectively, the derivative of x_i and y_i with respect to t . Since

$$\dot{x}_i = -\sum_{j=1}^i \ell_j \dot{\theta}_j \sin \theta_j, \quad \dot{y}_i = \sum_{j=1}^i \ell_j \dot{\theta}_j \cos \theta_j, \quad i = 1, 2, \dots, n,$$

We have

$$v_i = \sqrt{\left(-\sum_{j=1}^i \ell_j \dot{\theta}_j \sin \theta_j \right)^2 + \left(\sum_{j=1}^i \ell_j \dot{\theta}_j \cos \theta_j \right)^2}, \quad i = 1, 2, \dots, n. \quad (4.3)$$

Consequently, we can express the total kinetic energy of the system, KE, as

$$\begin{aligned} KE &= \sum_{i=1}^n \frac{1}{2} m_i v_i^2 \\ &= \sum_{i=1}^n \frac{1}{2} m_i \left(\left(-\sum_{j=1}^i \ell_j \dot{\theta}_j \sin \theta_j \right)^2 + \left(\sum_{j=1}^i \ell_j \dot{\theta}_j \cos \theta_j \right)^2 \right) \\ &= \sum_{i=1}^n \frac{1}{2} m_i \left(\sum_{j=1}^i \sum_{k=1}^i \ell_j \ell_k \dot{\theta}_j \dot{\theta}_k \sin \theta_j \sin \theta_k + \sum_{j=1}^i \sum_{k=1}^i \ell_j \ell_k \dot{\theta}_j \dot{\theta}_k \cos \theta_j \cos \theta_k \right) \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i m_i \ell_j \ell_k \dot{\theta}_j \dot{\theta}_k (\sin \theta_j \sin \theta_k + \cos \theta_j \cos \theta_k) \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i m_i \ell_j \ell_k \dot{\theta}_j \dot{\theta}_k \cos(\theta_j - \theta_k). \end{aligned} \quad (4.4)$$

We may also find the total potential energy of the system by summing the gravitational potential energies of each mass m_i to get

$$PE = \sum_{i=1}^n m_i g y_i = \sum_{i=1}^n \sum_{j=1}^i m_i g \ell_j \sin \theta_j. \quad (4.5)$$

Next, we define the Lagrangian function L , also called the Lagrangian, which is quantity that characterizes the state of a physical system. The Lagrangian function is just the kinetic energy (energy of

position) minus the potential energy (energy of position) *i.e.*,

$$L = KE - PE = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i m_i \ell_j \ell_k \dot{\theta}_j \dot{\theta}_k \cos(\theta_j - \theta_k) - \sum_{i=1}^n \sum_{j=1}^i m_i g \ell_j \sin \theta_j. \quad (4.6)$$

With these definitions, the Euler-Lagrange equations (or Lagrange's equations) of motion for a conservative system are given by

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}_q} \right] - \frac{\partial L}{\partial \theta_q} = \tau_q, \quad q = 1, 2, \dots, n, \quad (4.7)$$

Where τ_q is the torque applied to the q^{th} link. The derivations for the Lagrangian and the Euler-Lagrange equation may be found in [11].

Using (2.6), we have, for any $q = 1, 2, \dots, n$,

$$\begin{aligned} \frac{\partial L}{\partial \dot{\theta}_q} &= \sum_{i=q}^n \sum_{j=1}^i m_i \ell_j \ell_q \dot{\theta}_j \cos(\theta_q - \theta_j) \\ \frac{\partial L}{\partial \theta_q} &= - \sum_{i=q}^n \sum_{j=1}^i m_i \ell_j \ell_q \dot{\theta}_j \dot{\theta}_q \sin(\theta_q - \theta_j) - \sum_{i=q}^n m_i g \ell_q \cos \theta_q \\ \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}_q} \right] &= \sum_{i=q}^n \sum_{j=1}^i m_i \ell_j \ell_q \left[\ddot{\theta}_j \cos(\theta_q - \theta_j) - \dot{\theta}_j (\dot{\theta}_q - \dot{\theta}_j) \sin(\theta_q - \theta_j) \right]. \end{aligned}$$

Substituting these relations into (2.8), we get the equation of motion

$$\begin{aligned} \tau_q &= \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}_q} \right] - \frac{\partial L}{\partial \theta_q} \\ &= \sum_{i=q}^n \sum_{j=1}^i m_i \ell_j \ell_q \left[\ddot{\theta}_j \cos(\theta_q - \theta_j) - \dot{\theta}_j (\dot{\theta}_q - \dot{\theta}_j) \sin(\theta_q - \theta_j) \right] \\ &\quad + \sum_{i=q}^n \sum_{j=1}^i m_i \ell_j \ell_q \dot{\theta}_j \dot{\theta}_q \sin(\theta_q - \theta_j) + \sum_{i=q}^n m_i g \ell_q \cos \theta_q \\ &= \sum_{i=q}^n \sum_{j=1}^i m_i \ell_j \ell_q \left[\ddot{\theta}_q \cos(\theta_q - \theta_j) - \dot{\theta}_j (\dot{\theta}_q - \dot{\theta}_j) \sin(\theta_q - \theta_j) + \dot{\theta}_j \dot{\theta}_q \sin(\theta_q - \theta_j) \right] \\ &\quad + \sum_{i=q}^n m_i g \ell_q \cos \theta_q \\ &= \sum_{i=q}^n \sum_{j=1}^i m_i \ell_j \ell_q \left[\ddot{\theta}_q \cos(\theta_q - \theta_j) + \dot{\theta}_j^2 \sin(\theta_q - \theta_j) \right] + \sum_{i=q}^n m_i g \ell_q \cos \theta_q, \end{aligned}$$

for $q = 1, 2, \dots, n$. Thus, we get the following n nonlinear second-order system of ordinary differential equations

$$\sum_{i=q}^n \sum_{j=1}^i m_i \ell_j \ell_q \ddot{\theta}_j \cos(\theta_q - \theta_j) + \sum_{i=q}^n \sum_{j=1}^i m_i \ell_j \ell_q \dot{\theta}_j^2 \sin(\theta_q - \theta_j) + \sum_{i=q}^n m_i g \ell_q \cos \theta_q = \tau_q. \quad (4.8)$$

Define

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}, \quad \dot{\theta} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_n \end{bmatrix}, \quad \ddot{\theta} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \vdots \\ \ddot{\theta}_n \end{bmatrix}, \quad F = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_n \end{bmatrix}, \quad c(\theta, \dot{\theta}) = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}, \quad G(\theta) = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix},$$

$$M(\theta) = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix},$$

where

$$a_{qk} = \sum_{i=\min(k,q)}^n m_i \ell_k \ell_q \cos(\theta_q - \theta_k), \quad q, k = 1, 2, \dots, n,$$

$$c_q = \sum_{i=q}^n \sum_{j=1}^i m_i \ell_j \ell_q \dot{\theta}_j^2 \sin(\theta_q - \theta_j), \quad q = 1, 2, \dots, n,$$

$$g_q = \sum_{i=q}^n m_i g \ell_q \cos \theta_q, \quad q = 1, 2, \dots, n.$$

Then a more compact formulation of (2.8) is given by

$$M(\theta)\ddot{\vartheta} + c(\theta, \dot{\vartheta}) + G(\theta) = F, \quad (4.9)$$

where M is the $n \times n$ mass (or inertia) matrix, ϑ is the $n \times 1$ trajectory vector (the generalized coordinates), $\dot{\vartheta}$ and $\ddot{\vartheta}$ are the first and second derivative of ϑ , $c(\vartheta, \dot{\vartheta})$ is the $n \times 1$ Centrifugal/Coriolis force containing velocity-dependent torques, $G(\vartheta)$ is the $n \times 1$ gravitational torque, and F contains the input joint torques.

Since the kinetic energy is positive, vanishing only when the generalized velocity equals zero, the inertia matrix $M(\vartheta)$ is also positive definite. Thus, the matrix M is invertible. Multiplying (2.9) from the left by M^{-1} , we obtain the normal form of the dynamics equations

$$\ddot{\vartheta} = f(t, \vartheta, \dot{\vartheta}), \quad (4.10a)$$

Where

$$f(t, \vartheta, \dot{\vartheta}) = M^{-1}(\vartheta)F - M^{-1}(\vartheta)c(\vartheta, \dot{\vartheta}) - M^{-1}(\vartheta)G(\vartheta). \quad (4.10b)$$

A numerical scheme

In this subsection, we use the fourth-order Runge-Kutta (RK4) method to approximate the solution of the following initial-value problem (IVP)

$$\ddot{\vartheta} = f(t, \vartheta, \dot{\vartheta}), \quad t \in [0, T], \quad (4.11a)$$

$$\vartheta(0) = \alpha, \quad \dot{\vartheta}(0) = \beta, \quad (4.11b)$$

Where the initial conditions $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_n]^t$ and $\beta = [\beta_1, \beta_2, \dots, \beta_n]^t$ are given.

To apply the RK4 method, we first convert the second-order system of equations into a first order system of equations. To do this, we introduce the new variables

$$u_i = \vartheta_i, \quad i = 1, 2, \dots, n,$$

$$u_{n+i} = \dot{\vartheta}_i, \quad i = 1, 2, \dots, n.$$

Since $\dot{u}_i = \dot{\vartheta}_i = u_{n+i}$ and $\dot{u}_{n+i} = \ddot{\vartheta}_i = f_i, i = 1, 2, \dots, n$, we get the first-order system of equation

$$\dot{u} = s(t, u), \quad t \in [0, T], \quad (4.12a)$$

$$u(0) = u^{(0)}, \quad (4.12b)$$

where

$$\begin{aligned}
u &= [u_1, u_2, \dots, u_n, u_{n+1}, \dots, u_{2n}]^t, \\
s &= [s_1, s_2, \dots, s_n, s_{n+1}, \dots, s_{2n}]^t \\
&= [u_{n+1}, u_{n+2}, \dots, u_{2n}, f_1(t, u_1, u_2, \dots, u_n, u_{n+1}, \dots, u_{2n}), \dots, f_n(t, u_1, u_2, \dots, u_n, u_{n+1}, \dots, u_{2n})]^t, \\
u^{(0)} &= [\alpha^t, \beta^t] = [\alpha_1, \alpha_2, \dots, \alpha_n, \beta_1, \beta_2, \dots, \beta_n]^t.
\end{aligned}$$

Next, we use the fourth-order Runge-Kutta method [12] to approximate the solution u to the IVP (2.12). We first discretize the computational interval $[0, T]$ by non-overlapping intervals $I_l = (t_{i-1}, t_i)$, $l = 1, 2, \dots, m$ such that $0 = t_0 < t_1 < \dots < t_m = T$. For simplicity we assume that the partition is uniform. Let us define the length

of I_l as $h_i = x_i - x_{i-1} = \frac{T}{m}$. Then for each $l = 1, 2, \dots, m$ we compute k_1, k_2, k_3, k_4 using

$$\begin{aligned}
k_1 &= s(t_i, u^{(i)}), \\
k_2 &= s(t_i + \frac{h}{2}, u^{(i)} + \frac{h}{2}k_1), \\
k_3 &= s(t_i + \frac{h}{2}, u^{(i)} + \frac{h}{2}k_2), \\
k_4 &= s(t_i + h, u^{(i)} + hk_3),
\end{aligned}$$

And finally the RK4 approximation of u^{i+1} at t_{i+1} is given by

$$u^{(i+1)} = u^{(i)} + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad i = 1, 2, \dots, m,$$

For some arbitrary step-size $h > 0$. The initial condition is given by

$$u^{(0)} = [u_1, u_2, \dots, u_n, u_{n+1}, \dots, u_{2n}]^t = [\theta_1(0), \theta_2(0), \dots, \theta_n(0), \dot{\theta}_1(0), \dots, \dot{\theta}_n(0)]^t.$$

PID Controller

For the robot manipulator to reach target positions, we require the stabilization of its motion and the ability to fix it at a particular position. In order to achieve this, we must implement a control method. Our control method utilizes a computed torque technique based on our equations of motion to hold and move each link of the n -link robot manipulator at an angle ϑ_i with respect to the x-axis, such that it may obtain a specified position.

We utilize the proportional-integral-derivative (PID) control in this paper. The PID control is one of the most effective and popular control systems. There are three parts to the control system: the Proportional control (P), the Integral control (I), and the Derivative control (D). The PID control works through utilizing a continuously calculated error value $V_E = V_{SP} - V_{PV}$. The Proportional, Integral, and Derivative controls may now be expressed with constants K_p, K_I, K_D , as $K_p V_E$, $K_I \int V_E dt$, and $K_D \frac{dV_E}{dt}$, respectively.

In this paper, we define that setpoint as ϑ_f and the process variable as $\vartheta(t)$ such that our error function $e(t)$ is $e(t) = \vartheta_f - \vartheta(t)$. The three components are each responsible for their own purposes in the overall control. The Proportional control is the main drive in the system and acts proportional to the error value V_E . The Integral control is meant to account for residual error produced by the Proportional control over time, integrating it to eliminate it. The Derivative control estimates future error values based on its rate of change, reducing overshoot and ringing.

To formulate our PID control approach, we consider the dynamics of n -link robot manipulator described by the nonlinear equation

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

where $M(\vartheta) = (m_{qk})_{1 \leq q, k \leq n}$ is an $n \times n$ symmetric positive definite inertia matrix, $c(\theta, \dot{\theta}) = [c_1, c_2, \dots, c_n]^t$ is an $n \times 1$ Coriolis and centrifugal vector, $G(\vartheta) = [g_1, g_2, \dots, g_n]^t$ is an $n \times 1$ gravity vector of the manipulator, $\vartheta = [\vartheta_1, \vartheta_2, \dots, \vartheta_n]^t$ is the $n \times 1$ vector representing joint angular positions, and $\tau = [\tau_1, \tau_2, \dots, \tau_n]^t$ is the $n \times 1$ vector of applied joint torques. The entries of $M(\vartheta)$, $c(\theta, \dot{\theta})$, and $G(\vartheta)$ are, respectively,

$$m_{qk} = \sum_{i=\min(k,q)}^n m_i \ell_k \ell_q \cos(\theta_q - \theta_k), \quad q, k = 1, 2, \dots, n,$$

$$c_q = \sum_{i=q}^n \sum_{j=1}^i m_i \ell_j \ell_q \dot{\theta}_j^2 \sin(\theta_q - \theta_j), \quad q = 1, 2, \dots, n,$$

$$g_q = \sum_{i=q}^n m_i g \ell_q \cos \theta_q, \quad q = 1, 2, \dots, n.$$

Solving for $\ddot{\theta}$, we obtain

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

Where $\hat{\tau} = M^{-1}(\theta)\tau = [\hat{\tau}_1, \hat{\tau}_2, \dots, \hat{\tau}_n]^t$. We define the position tracking error of the q^{th} joint as

$$e_q(t) = \theta_{qf} - \theta_q(t), \quad (5.3)$$

For all $q = 1, 2, \dots, n$ where θ_{qf} denotes the q^{th} joint's desired constant position and ϑ_q is the actual joint position.

A PID controller for the q^{th} joint is described by

$$\hat{\tau}_q(t) = K_{P_q} e_q(t) + K_{D_q} \dot{e}_q(t) + K_{I_q} \int_0^t e_q(s) ds, \quad (5.4)$$

Where K_{P_q} , K_{D_q} and K_{I_q} are, respectively, the proportional, derivative, and integral gains of the q^{th} joint controller. The control problem is to provide a complete solution to the constant gain stabilizing control parameters K_{P_q} , K_{D_q} and K_{I_q} such that the position error $e_q(t)$ reduces to zero with time, *i.e.*, $\lim_{t \rightarrow \infty} e_q(t) = 0$.

The input to the system (3.2) with the PID controller is $\hat{\tau} = [\hat{\tau}_1, \hat{\tau}_2, \dots, \hat{\tau}_n]^t$, where for each $q = 1, 2, \dots, n$,

$$\begin{aligned} \hat{\tau}_q(t) &= K_{P_q} e_q(t) + K_{D_q} \dot{e}_q(t) + K_{I_q} \int_0^t e_q(s) ds \\ &= K_{P_q} (\theta_{qf} - \theta_q) - K_{D_q} \dot{\theta}_q + K_{I_q} \int_0^t (\theta_{qf} - \theta_q) ds, \end{aligned} \quad (5.5)$$

Where we used the fact that $\dot{e}_q = -\dot{\theta}_q$ since θ_{qf} is a constant denoting the desired target angle.

To implement the PID controller, we introduce new variables x_1, x_2, \dots, x_n , as follows

$$x_q(t) = \int_0^t e_q(s) ds = \int_0^t (\theta_{qf} - \theta_q) ds, \quad q = 1, 2, \dots, n,$$

So that

$$\dot{x}_q(t) = e_q(t) = \theta_{qf} - \theta_q, \quad q = 1, 2, \dots, n.$$

Thus, $\hat{\tau}_q(t)$ can be written as

$$\hat{\tau}_q(t) = K_{P_q} (\theta_{qf} - \theta_q) - K_{D_q} \dot{\theta}_q + K_{I_q} x_q. \quad (5.6)$$

With this notation, we rewrite (3.2) as

$$\ddot{\theta} = z(t, x, \theta, \dot{\theta}), \quad (5.7)$$

Where $z(t, x, \theta, \dot{\theta}) = -M^{-1}(\theta) [c(\theta, \dot{\theta}) + G(\theta)] + \hat{\tau}$. We note that

$$\hat{\tau} = \begin{bmatrix} \hat{\tau}_1 \\ \hat{\tau}_2 \\ \vdots \\ \hat{\tau}_n \end{bmatrix} = \begin{bmatrix} K_{P_1}(\theta_{1f} - \theta_1) - K_{D_1}\dot{\theta}_1 + K_{I_1}x_1 \\ K_{P_2}(\theta_{2f} - \theta_2) - K_{D_2}\dot{\theta}_2 + K_{I_2}x_2 \\ \vdots \\ K_{P_n}(\theta_{nf} - \theta_{2n}) - K_{D_n}\dot{\theta}_n + K_{I_n}x_n \end{bmatrix} = K_P(\theta_f - \theta) - K_D\dot{\theta} + K_Ix,$$

Where

$$K_S = \begin{bmatrix} K_{s_1} & 0 & 0 & \dots & 0 \\ 0 & K_{s_2} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & K_{s_n} \end{bmatrix}, \quad \theta_f = \begin{bmatrix} \theta_{1f} \\ \theta_{2f} \\ \vdots \\ \theta_{nf} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix},$$

where S stands for either P or D or I .

To numerically implement the PID controller, we convert the second-order system of differential equations (3.7) into a first-order system of differential equations by introducing the following variables:

$$u_i = x_i, \quad u_{n+i} = \theta_i, \quad u_{2n+i} = \dot{\theta}_i, \quad i = 1, 2, \dots, n.$$

Differentiating them with respect to t , we get

$$\dot{u}_i = \dot{x}_i = \dot{\theta}_{if} - \theta_i = \theta_{if} - u_{n+i},$$

$$\dot{u}_{n+i} = \dot{\theta}_i = u_{2n+i},$$

$$\dot{u}_{2n+i} = \ddot{\theta}_i = z_i(t, u_1, u_2, \dots, u_{3n}), \quad i = 1, 2, \dots, n,$$

where z_i are the entries of z i.e., we used the notation $z(t, x, \theta, \dot{\theta}) = [z_1, z_2, \dots, z_n]^t$.

Define

$$u = [u_1, \dots, u_n, u_{n+1}, \dots, u_{2n}, u_{2n+1}, \dots, u_{3n}]^t.$$

Then we obtain

$$\dot{u} = g(t, u), \quad u(0) = u^{(0)}, \quad (5.8)$$

where

$$g = [\theta_{1f} - u_{n+1}, \dots, \theta_{nf} - u_{2n}, u_{2n+1}, \dots, u_{3n}, z_1(t, u_1, u_2, \dots, u_{3n}), \dots, z_n(t, u_1, u_2, \dots, u_{3n})]^t.$$

Finally, the fourth-order Runge-Kutta method can be used to approximate u and thus we deduce approximations to the angles ϑ_r , the errors e_r and the torques τ_r . The fourth-order Runge-Kutta method consists of generating a sequence $\{u^{(i)}\}$ as follows: Given the initial condition $u(0) = u^{(0)}$, where

$$u^{(0)} = [x_1(0), x_2(0), \dots, x_n(0), \theta_1(0), \theta_2(0), \dots, \theta_n(0), \dot{\theta}_1(0), \dot{\theta}_2(0), \dots, \dot{\theta}_n(0)]^t,$$

we generate $u^{(i+1)}$ for $i = 0, 1, \dots, m$ using

$$u^{(i+1)} = u^{(i)} + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

where

$$k_1 = g(t_i, u^{(i)}),$$

$$k_2 = g(t_i + \frac{h}{2}, u^{(i)} + \frac{h}{2}k_1),$$

$$k_3 = g(t_i + \frac{h}{2}, u^{(i)} + \frac{h}{2}k_2),$$

$$k_4 = g(t_i + h, u^{(i)} + hk_3).$$

Here, $h = \frac{T}{m}$ is the step size and $t_i = ih$ for $i = 0, 1, \dots, m$.

Numerical Experiments

In this section, we present several numerical results to show the performance of the PID controller. We consider n -link robot manipulators with $n = 3, 4, 5, 6$. The case $n = 2$ is considered in [5].

Example 6.1. (simulation results for the three-link robot manipulator): In this example, we consider a 3-link robot manipulator. We take the following parameters: $m_1 = m_2 = m_3 = 1$, $l_1 = 2$, $l_2 = l_3 = 1$. We choose the final position as $\theta_{1f} = \frac{\pi}{2}$, $\theta_{2f} = 0$, $\theta_{3f} = 0$. The initial conditions are take as follows

$$\begin{bmatrix} \theta_1(0) \\ \theta_2(0) \\ \theta_3(0) \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} \\ \frac{\pi}{2} \\ \frac{\pi}{2} \end{bmatrix}, \quad \begin{bmatrix} \dot{\theta}_1(0) \\ \dot{\theta}_2(0) \\ \dot{\theta}_3(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

We choose the following PID parameters

$$K_{P_1} = K_{P_2} = K_{P_3} = 30, \quad K_{D_1} = 15, \quad K_{D_2} = K_{D_3} = 10, \quad K_{I_1} = K_{I_2} = K_{I_3} = 20.$$

The initial and final positions of the robot manipulator are shown in Figure 2. Figure 3, Figure 4, and Figure 5 demonstrate $\vartheta_i(t)$, $e_i(t)$ and $\tau_i(t)$ over time interval $[0, 30]$ for $i = 1, 2, 3$.

Example 6.2. (simulation results for the three-link robot manipulator): In this example, we repeat the previous example (Example 5.1) with the following values: $m_1 = m_2 = 1$, $m_3 = 3$, $l_1 = 2$, $l_2 = 4$, $l_3 = 1$. The final position is $(\theta_{1f}, \theta_{2f}, \theta_{3f}) = (\frac{\pi}{2}, \frac{\pi}{2}, 0)$. The initial conditions are

$$\begin{bmatrix} \theta_1(0) \\ \theta_2(0) \\ \theta_3(0) \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} \\ 0 \\ \frac{\pi}{2} \end{bmatrix}, \quad \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

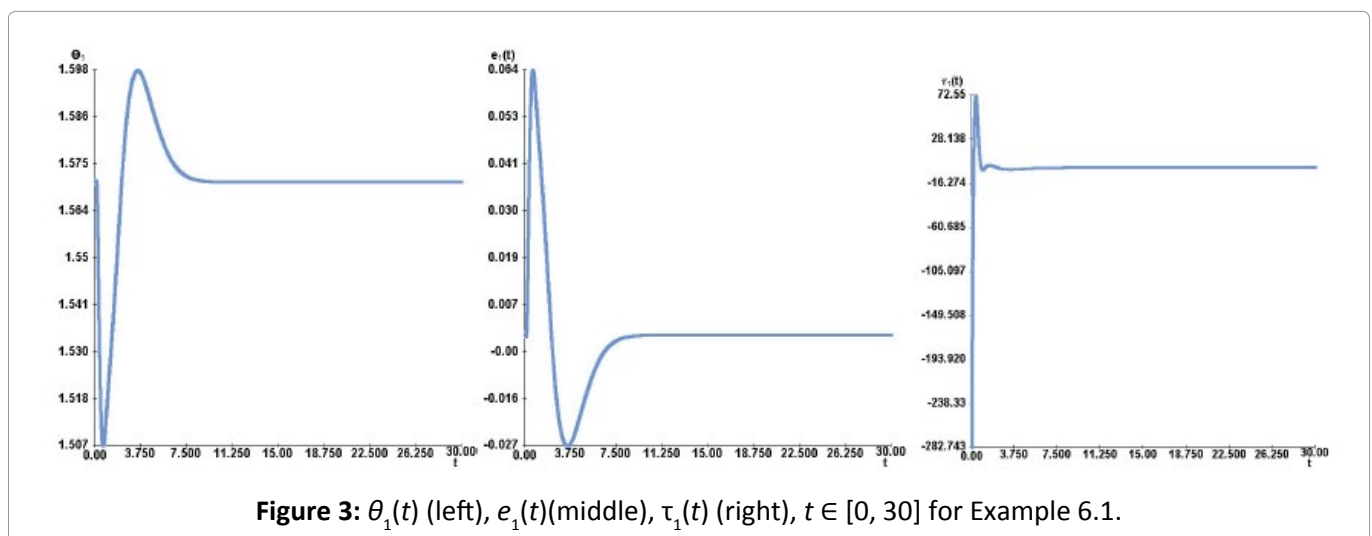
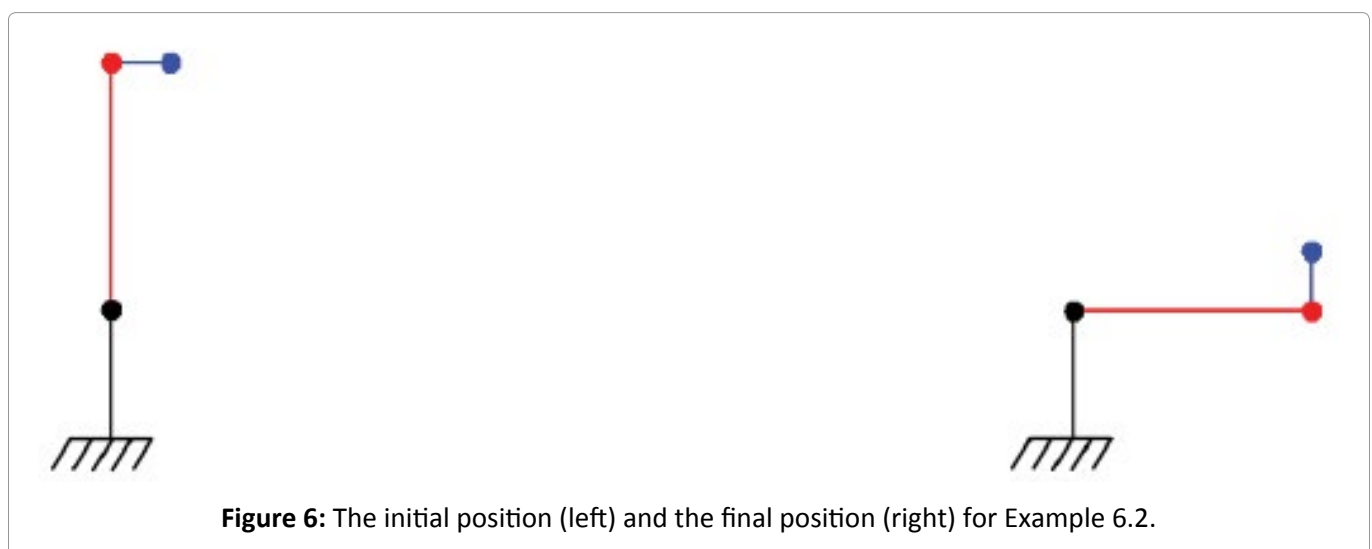
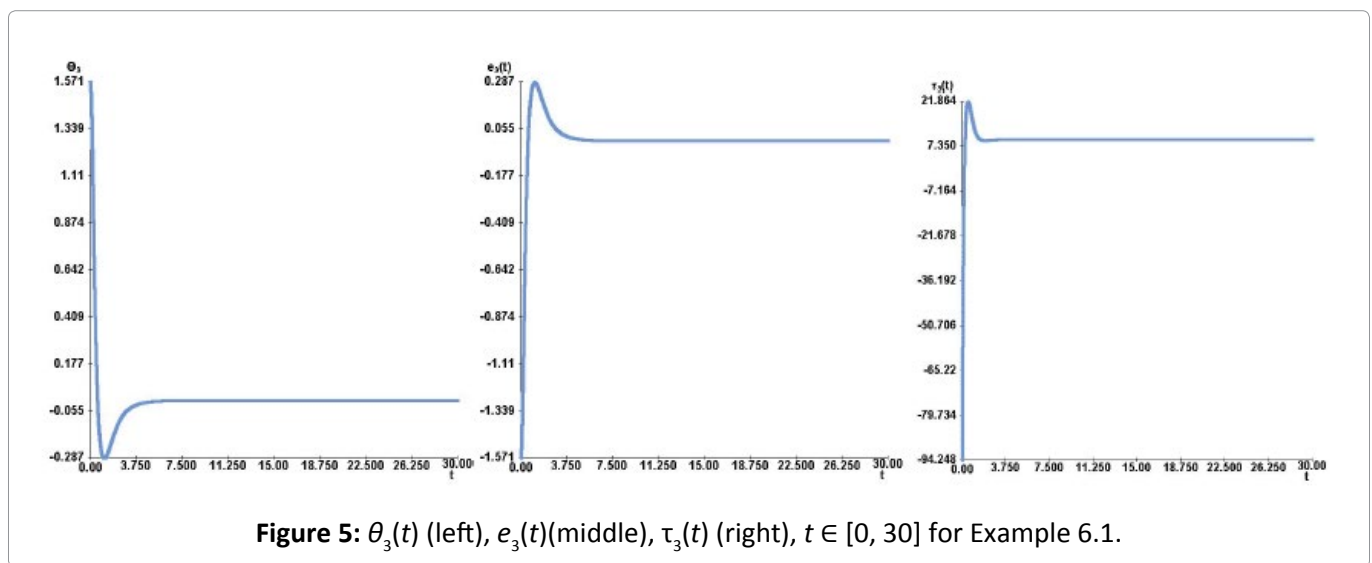
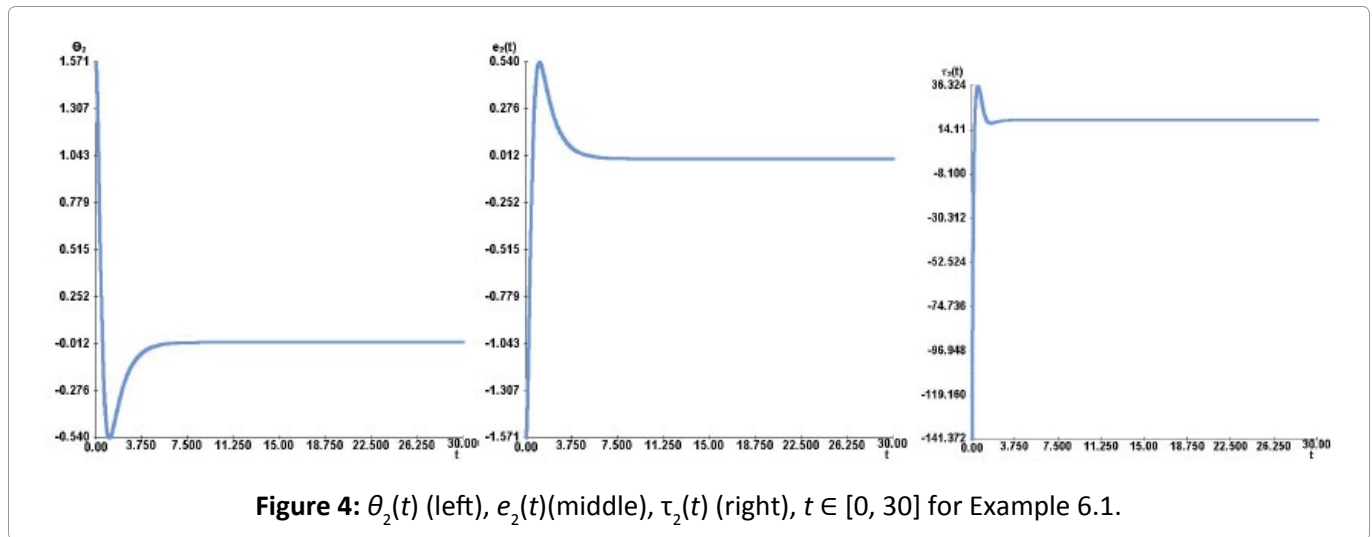


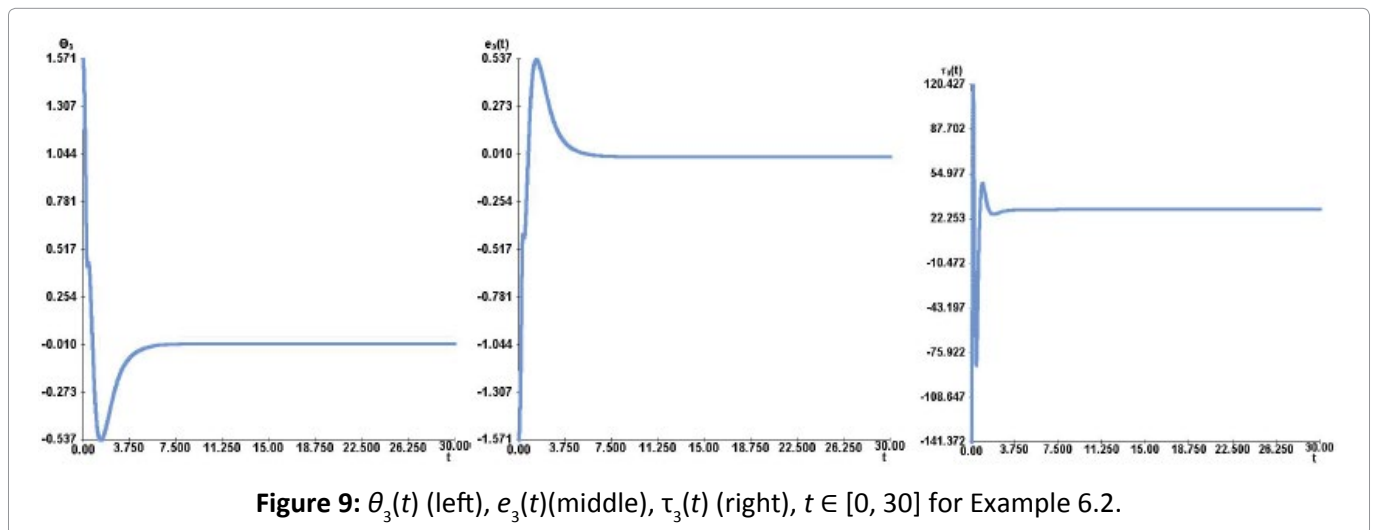
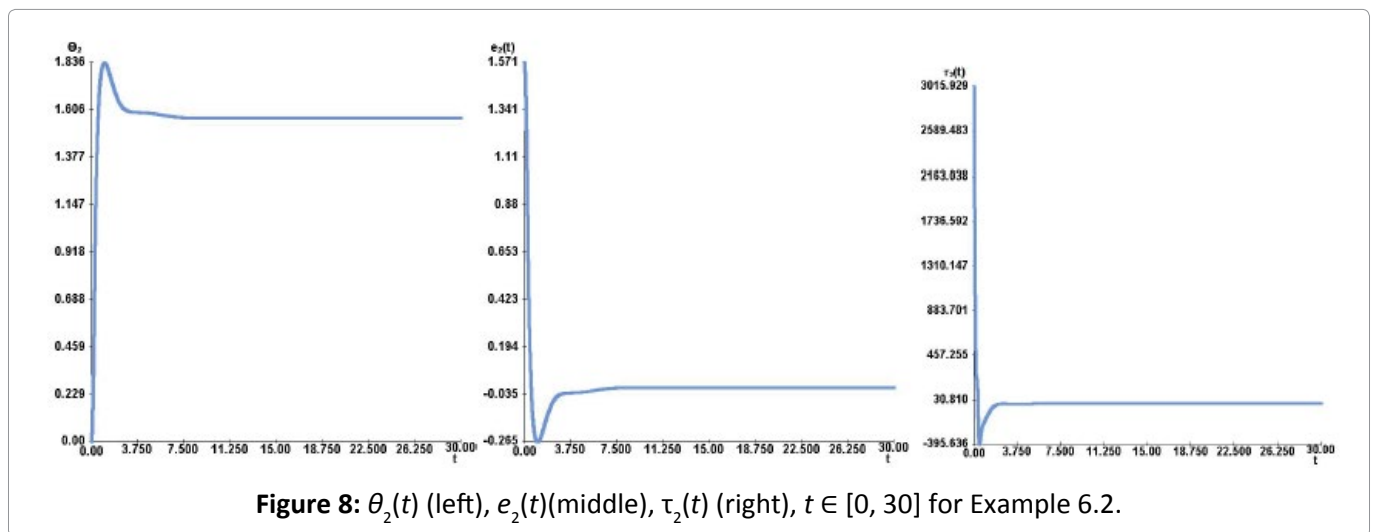
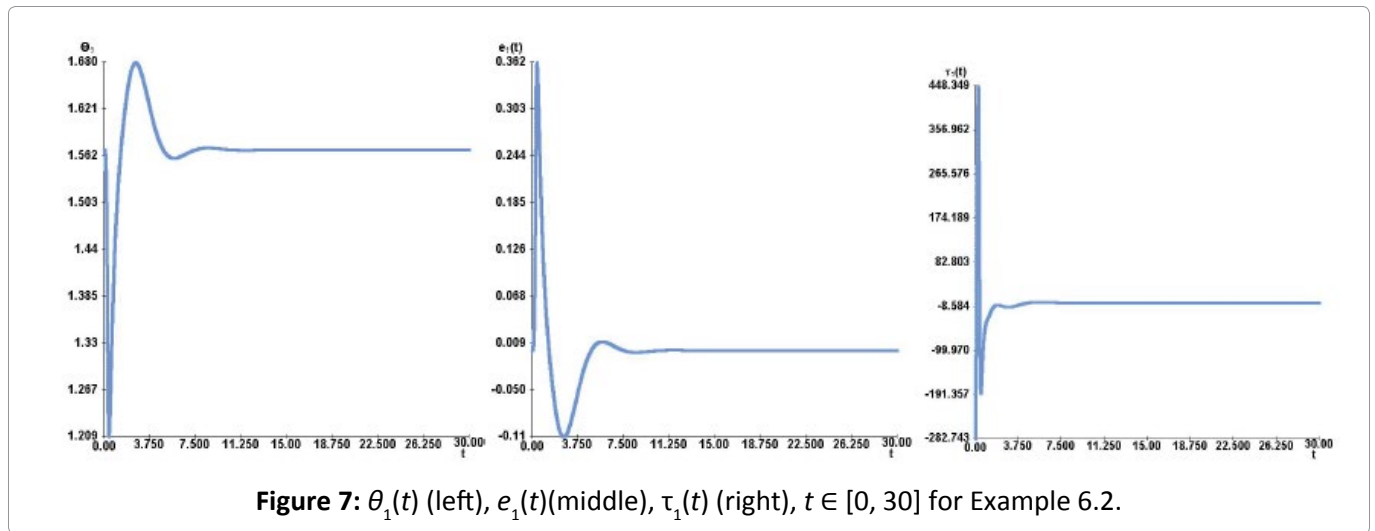
Figure 3: $\theta_1(t)$ (left), $e_1(t)$ (middle), $\tau_1(t)$ (right), $t \in [0, 30]$ for Example 6.1.



The PID parameters are

$$K_{P_1} = K_{P_2} = K_{P_3} = 30, \quad K_{D_1} = 15, \quad K_{D_2} = K_{D_3} = 10, \quad K_{I_1} = K_{I_2} = K_{I_3} = 10.$$

The initial and final positions of the robot manipulator can be observed in Figure 6. To further illustrate the dynamics, Figure 7, Figure 8, and Figure 9 provide insight into the variations of $\vartheta_i(t)$, $e_i(t)$ and $\tau_i(t)$ for i



= 1,2,3, spanning the time interval $[0, 30]$ (initial 30 seconds).

Example 6.3. (simulation results for the three-link robot manipulator): We repeat the previous example with the following values: $m_1 = 1, m_2 = m_3 = 2, l_1 = l_2 = l_3 = 1$. The final position is $(\theta_{1f}, \theta_{2f}, \theta_{3f}) = \left(\frac{\pi}{3}, 0, \frac{\pi}{4}\right)$. The initial conditions are



Figure 10: The initial position (left) and the final position (right) for Example 6.3.

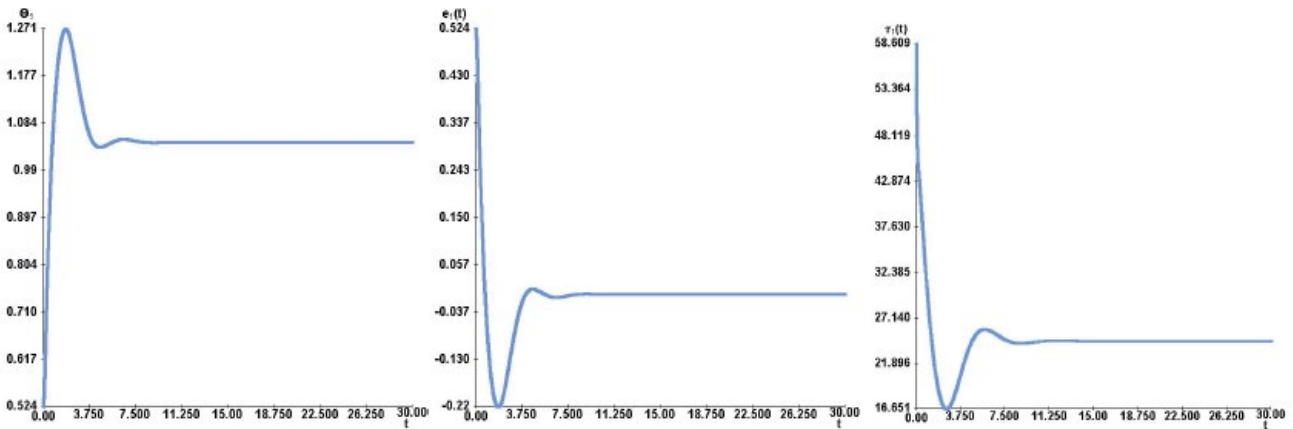


Figure 11: $\theta_1(t)$ (left), $e_1(t)$ (middle), $\tau_1(t)$ (right), $t \in [0, 30]$ for Example 6.3.

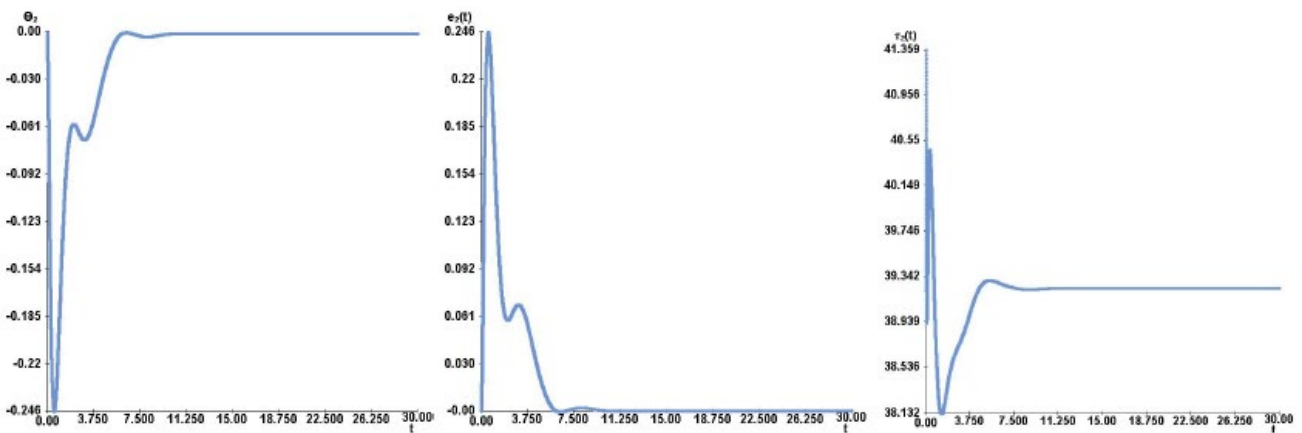


Figure 12: $\theta_2(t)$ (left), $e_2(t)$ (middle), $\tau_2(t)$ (right), $t \in [0, 30]$ for Example 6.3.

$$\begin{bmatrix} \theta_1(0) \\ \theta_2(0) \\ \theta_3(0) \end{bmatrix} = \begin{bmatrix} \frac{\pi}{6} \\ 0 \\ \frac{\pi}{5} \end{bmatrix}, \quad \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

The PID parameters are

$$K_{P_1} = K_{P_2} = K_{P_3} = 20, \quad K_{D_1} = 15, \quad K_{D_2} = K_{D_3} = 10, \quad K_{I_1} = K_{I_2} = K_{I_3} = 20.$$

Figure 10 displays the initial and final positions of the robot manipulator. Figure 11, Figure 12 and Figure 13 illustrate the temporal variations of $\vartheta_i(t)$, $e_i(t)$ and $\tau_i(t)$ for $i = 1,2,3$, over the time interval $[0, 30]$ (during the first 39 seconds).

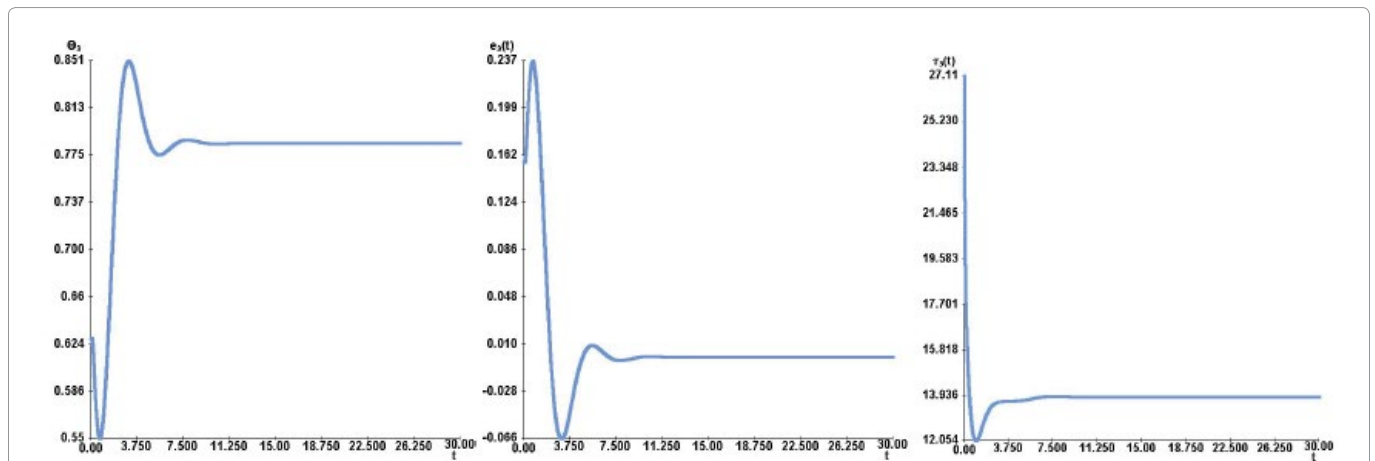


Figure 13: $\theta_3(t)$ (left), $e_3(t)$ (middle), $\tau_3(t)$ (right), $t \in [0, 30]$ for Example 6.3.

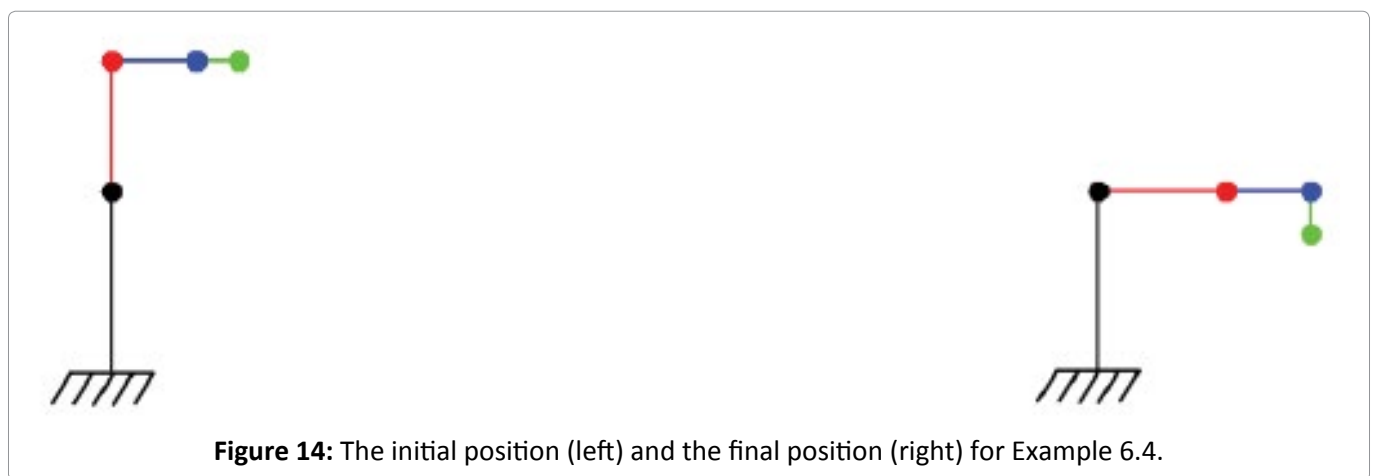


Figure 14: The initial position (left) and the final position (right) for Example 6.4.

Example 6.4. (simulation results for the four-link robot manipulator). In this experiment, we consider a 4-link robot manipulator. We utilize the following values: $m_1 = 4, m_2 = 3, m_3 = 2, m_4 = 1, l_1 = 4, l_2 = 3, l_3 = 2,$

$l_4 = 1$. The final position is $(\theta_{1f}, \theta_{2f}, \theta_{3f}, \theta_{4f}) = \left(\frac{\pi}{2}, 0, 0, \frac{3\pi}{2}\right)$.

The initial conditions are taken as follows

$$\begin{bmatrix} \theta_1(0) \\ \theta_2(0) \\ \theta_3(0) \\ \theta_4(0) \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} \\ \frac{\pi}{2} \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \\ x_4(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Our PID parameters are

$$K_{P_1} = K_{P_2} = K_{P_3} = K_{P_4} = 30, \quad K_{D_1} = 15, \quad K_{D_2} = K_{D_3} = K_{D_4} = 10, \quad K_{I_1} = K_{I_2} = 20, \quad K_{I_3} = K_{I_4} = 10.$$

Figure 14 presents the initial and final positions of the robot manipulator. Additionally, Figure 15, Figure 16, Figure 17 and Figure 18 depict the temporal variations of $\vartheta_i(t), e_i(t)$ and $\tau_i(t)$ for $i = 1, 2, 3, 4$, within the time interval $[0, 30]$ (during the initial 30 seconds).

Example 6.5. (Simulation results for the five-link robot manipulator): Here, we consider a 5-link robot manipulator. We take $m_1 = 25, m_2 = 20, m_3 = 15, m_4 = 10, m_5 = 5, l_1 = 5, l_2 = 4, l_3 = 3, l_4 = 2,$ and $l_5 = 5$. The

final position is $(\theta_{1f}, \theta_{2f}, \theta_{3f}, \theta_{4f}, \theta_{5f}) = \left(\frac{\pi}{4}, 0, \frac{\pi}{4}, 0, \frac{\pi}{4}\right)$. The initial conditions are

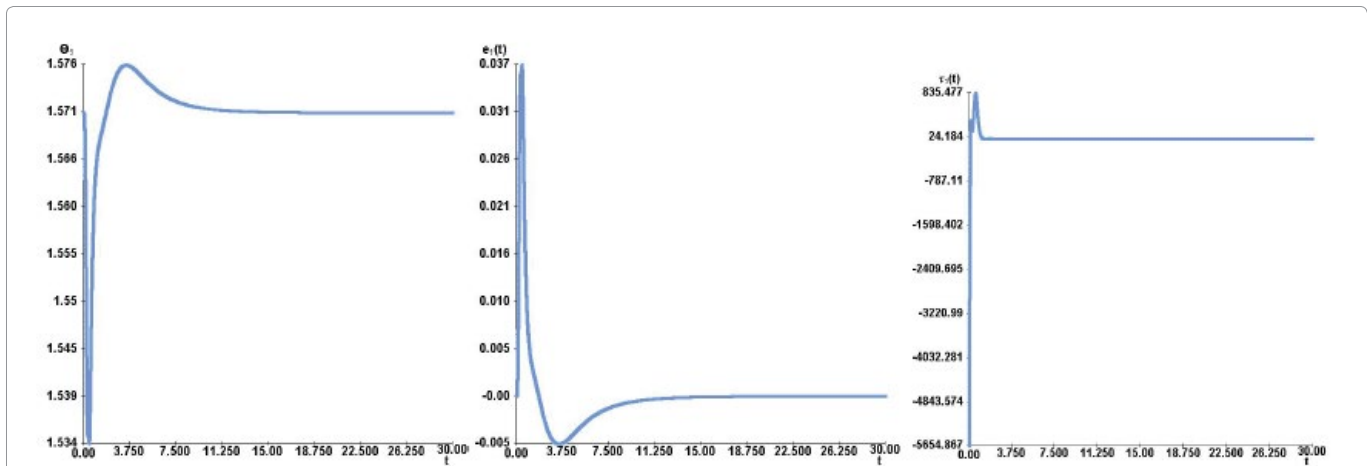


Figure 15: $\theta_1(t)$ (left), $e_1(t)$ (middle), $\tau_1(t)$ (right), $t \in [0, 30]$ for Example 6.4.

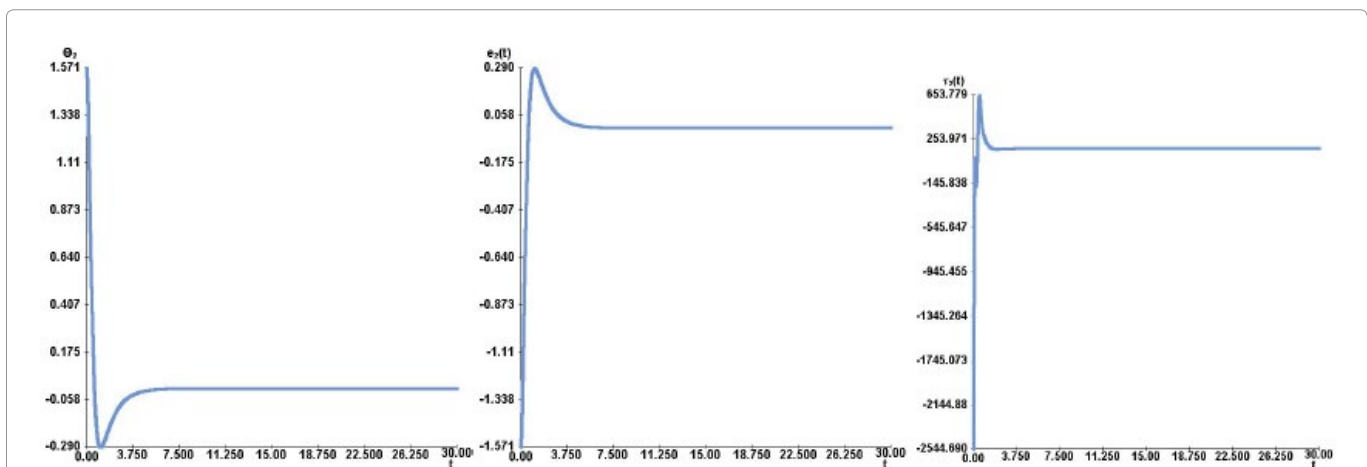


Figure 16: $\theta_2(t)$ (left), $e_2(t)$ (middle), $\tau_2(t)$ (right), $t \in [0, 30]$ for Example 6.4.

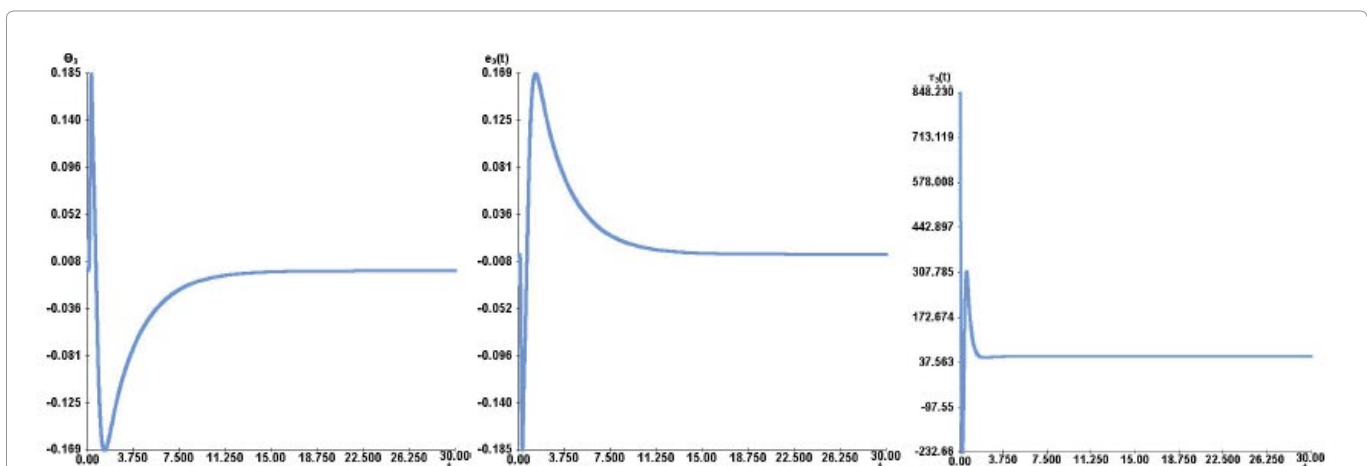


Figure 17: $\theta_3(t)$ (left), $e_3(t)$ (middle), $\tau_3(t)$ (right), $t \in [0, 30]$ for Example 6.4.

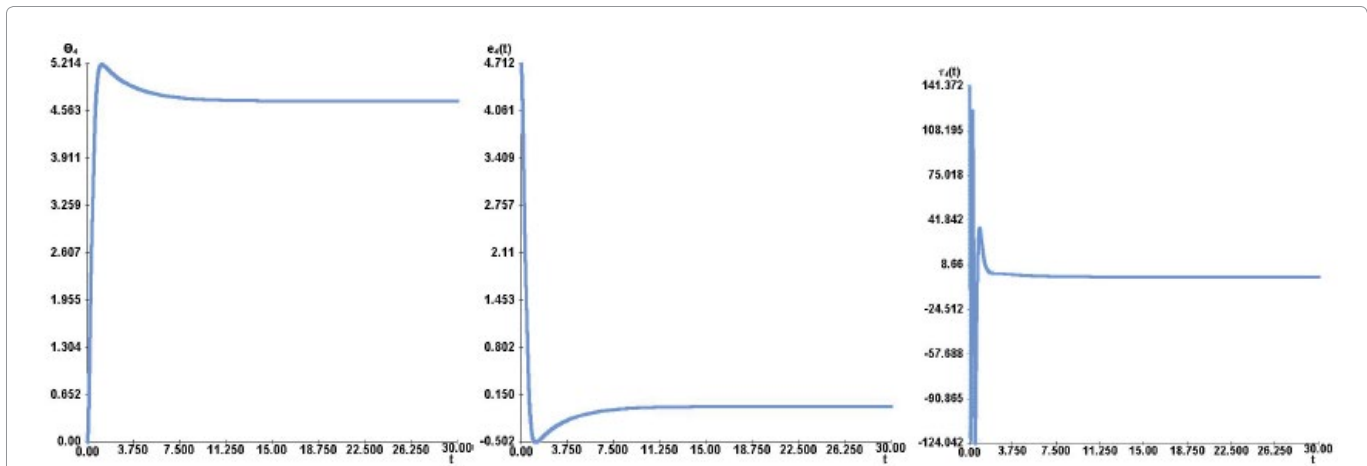


Figure 18: $\theta_4(t)$ (left), $e_4(t)$ (middle), $\tau_4(t)$ (right), $t \in [0, 30]$ for Example 6.4.

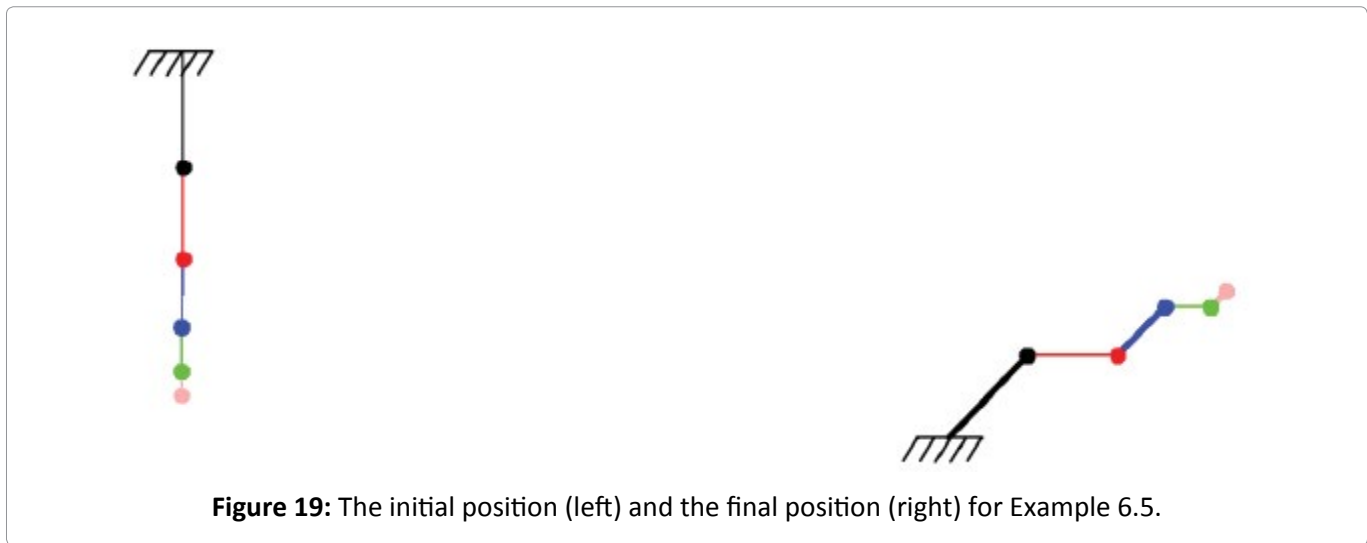


Figure 19: The initial position (left) and the final position (right) for Example 6.5.

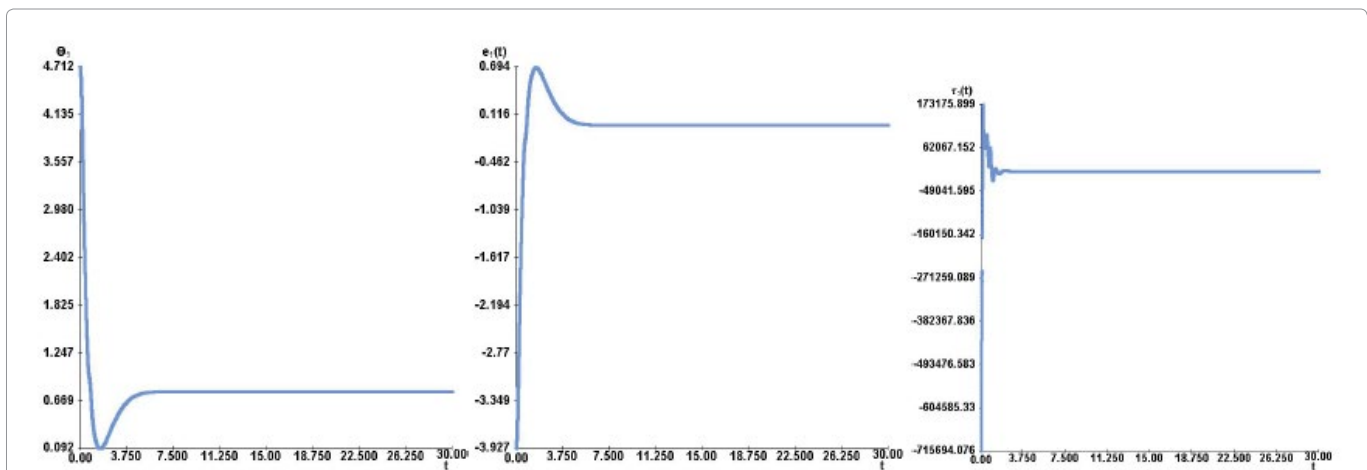


Figure 20: $\theta_1(t)$ (left), $e_1(t)$ (middle), $\tau_1(t)$ (right), $t \in [0, 30]$ for Example 6.5.

$$\begin{bmatrix} \theta_1(0) \\ \theta_2(0) \\ \theta_3(0) \\ \theta_4(0) \\ \theta_5(0) \end{bmatrix} = \begin{bmatrix} \frac{3\pi}{2} \\ \frac{3\pi}{2} \\ \frac{3\pi}{2} \\ \frac{3\pi}{2} \\ \frac{3\pi}{2} \end{bmatrix}, \quad \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \\ x_4(0) \\ x_5(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

For this experiment, we choose the following PID parameters $K_{P_1} = K_{P_2} = K_{P_3} = K_{P_4} = K_{P_5} = 30$, $K_{D_1} = 15$, $K_{D_2} = K_{D_3} = 10$, $K_{D_4} = K_{D_5} = 5$, $K_{I_1} = K_{I_2} = 20$, $K_{I_3} = K_{I_4} = 10$, and $K_{I_5} = 5$. The initial and final positions of the robot manipulator are shown in Figure 19, Figure 20, Figure 21, Figure 22, Figure 23, and Figure 24 illustrate the variations of $\vartheta_i(t)$, $e_i(t)$ and $\tau_i(t)$ for $i = 1, 2, \dots, 5$, across the time interval $[0, 30]$ (during the first 30 seconds).

Example 6.6. (Simulation results for the six-link robot manipulator): In this final experiment, we consider a 6-link robot manipulator. For this test we utilize the following values: $m_1 = 6$, $m_2 = 5$, $m_3 = 4$, $m_4 = 3$, $m_5 = 2$, $m_6 = 1$, $l_1 = 6$, $l_2 = 5$, $l_3 = 4$, $l_4 = 3$, $l_5 = 2$, and $l_6 = 1$. The target position is

$$(\theta_{1f}, \theta_{2f}, \theta_{3f}, \theta_{4f}, \theta_{5f}, \theta_{6f}) = \left(\frac{\pi}{2}, \frac{\pi}{3}, \frac{\pi}{4}, \frac{\pi}{5}, \frac{\pi}{6}, \frac{\pi}{7} \right).$$

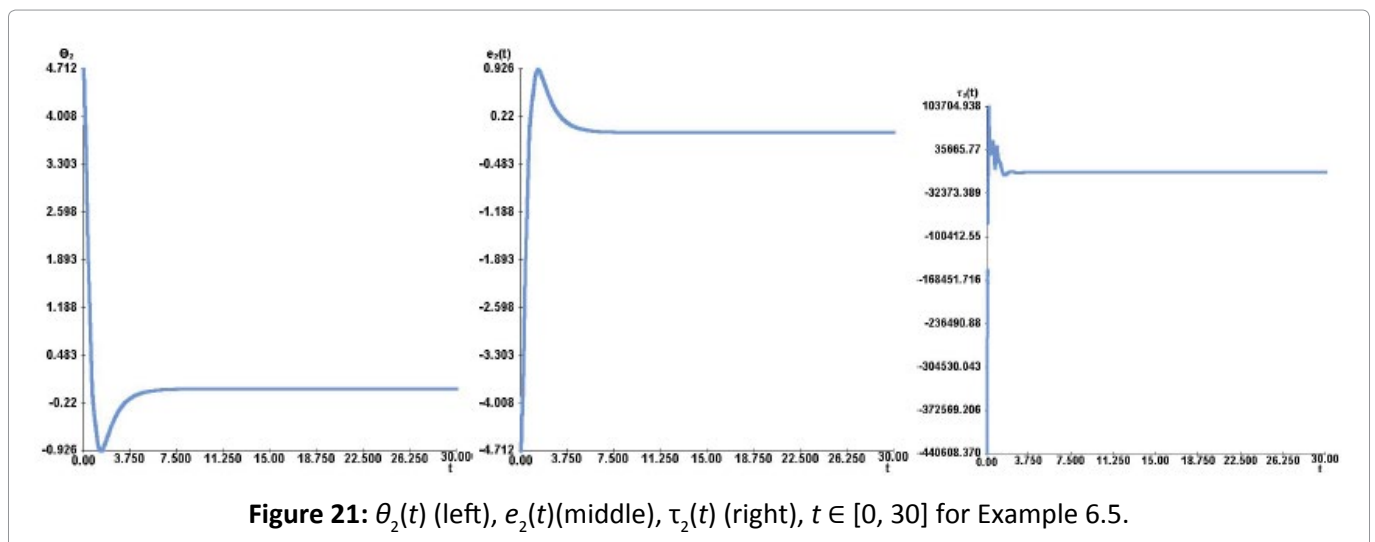


Figure 21: $\theta_2(t)$ (left), $e_2(t)$ (middle), $\tau_2(t)$ (right), $t \in [0, 30]$ for Example 6.5.

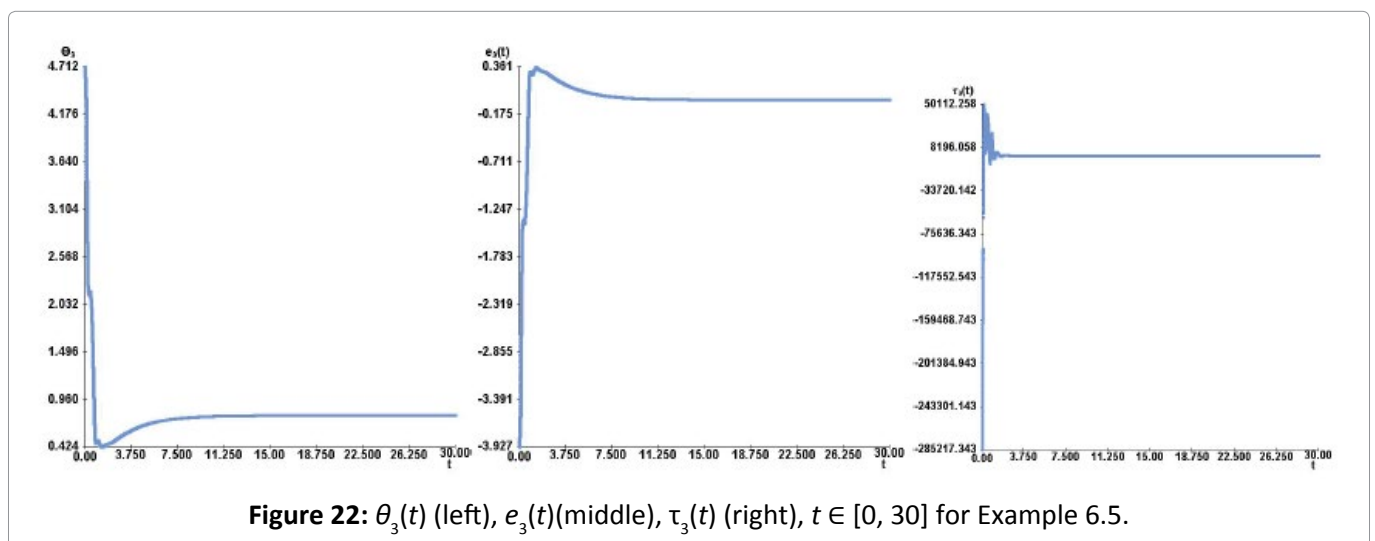
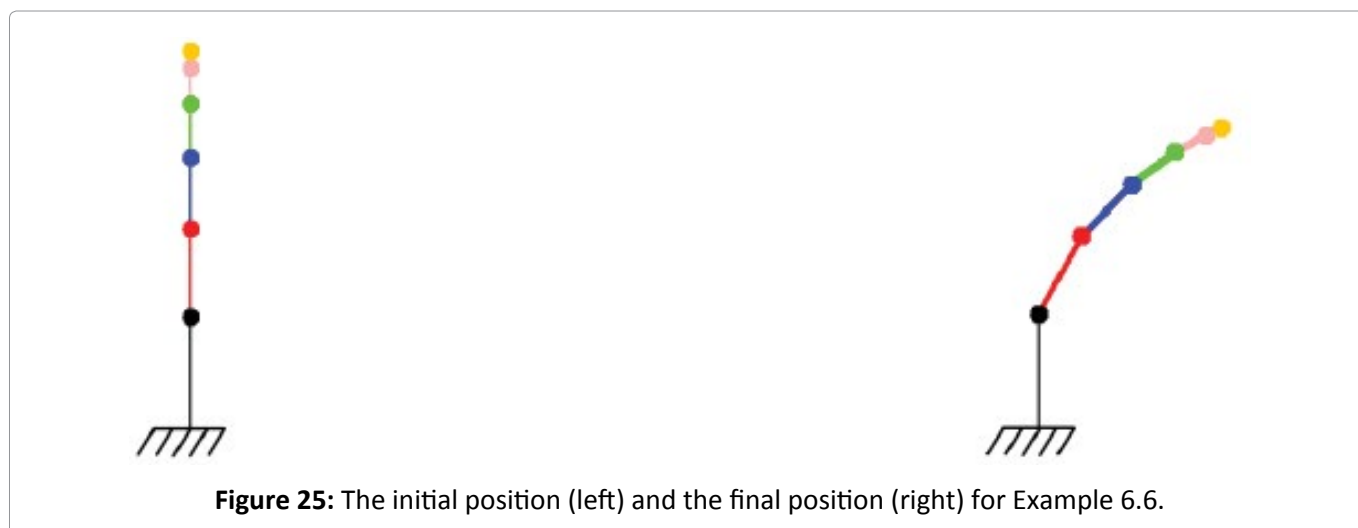
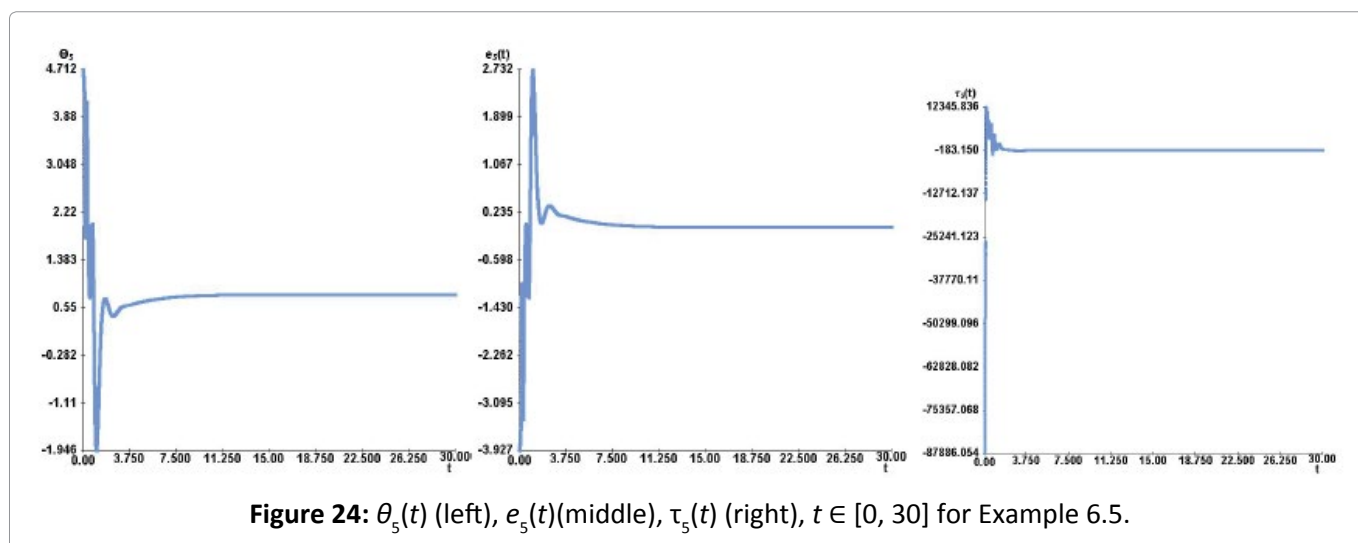
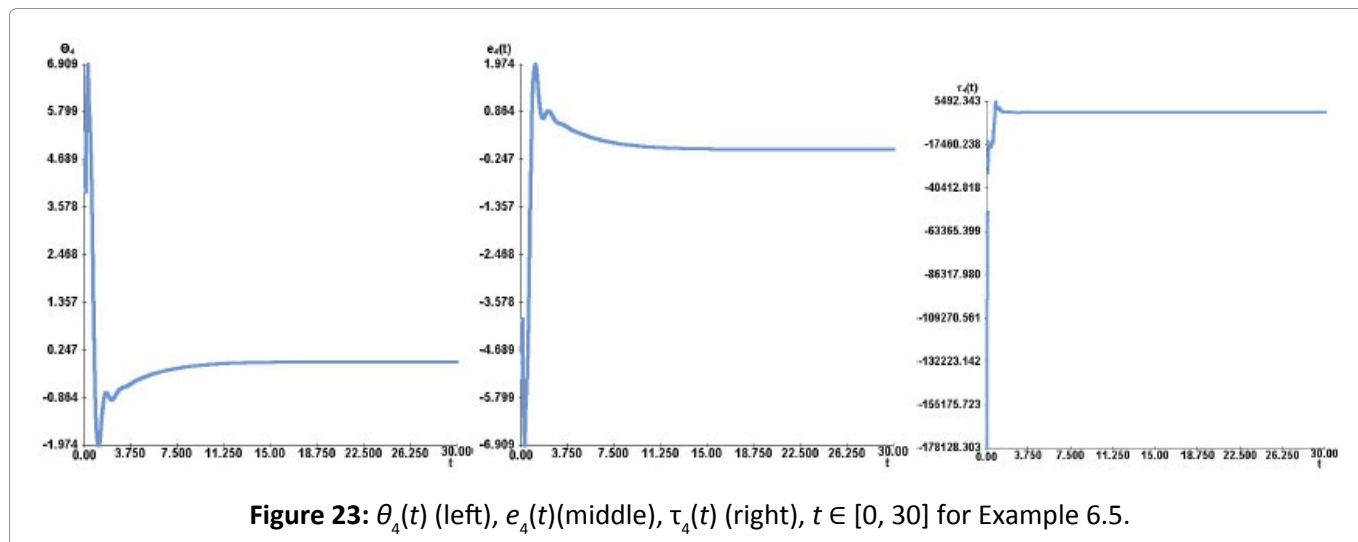


Figure 22: $\theta_3(t)$ (left), $e_3(t)$ (middle), $\tau_3(t)$ (right), $t \in [0, 30]$ for Example 6.5.



The initial conditions are

$$\begin{bmatrix} \theta_1(0) \\ \theta_2(0) \\ \theta_3(0) \\ \theta_4(0) \\ \theta_5(0) \\ \theta_6(0) \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} \\ \frac{\pi}{2} \\ \frac{\pi}{2} \\ \frac{\pi}{2} \\ \frac{\pi}{2} \\ \frac{\pi}{2} \end{bmatrix}, \quad \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \\ x_4(0) \\ x_5(0) \\ x_6(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The PID parameters are $K_{P_1} = K_{P_2} = K_{P_3} = K_{P_4} = K_{P_5} = K_{P_6} = 30$,

$K_{D_1} = 15$, $K_{D_2} = K_{D_3} = K_{D_4} = K_{D_5} = K_{D_6} = 10$, $K_{I_1} = K_{I_2} = 20$, $K_{I_3} = K_{I_4} = K_{I_5} = K_{I_6} = 10$. The initial and final positions of the robot manipulator are resented in Figure 25. Figure 26, Figure 27, Figure 28, Figure 29, Figure 30, and Figure 31 illustrate the behaviours of $\vartheta_i(t)$, $e_i(t)$ and $\tau_i(t)$ for $i = 1, 2, \dots, 6$ over the time interv30] (initial 30 seconds).

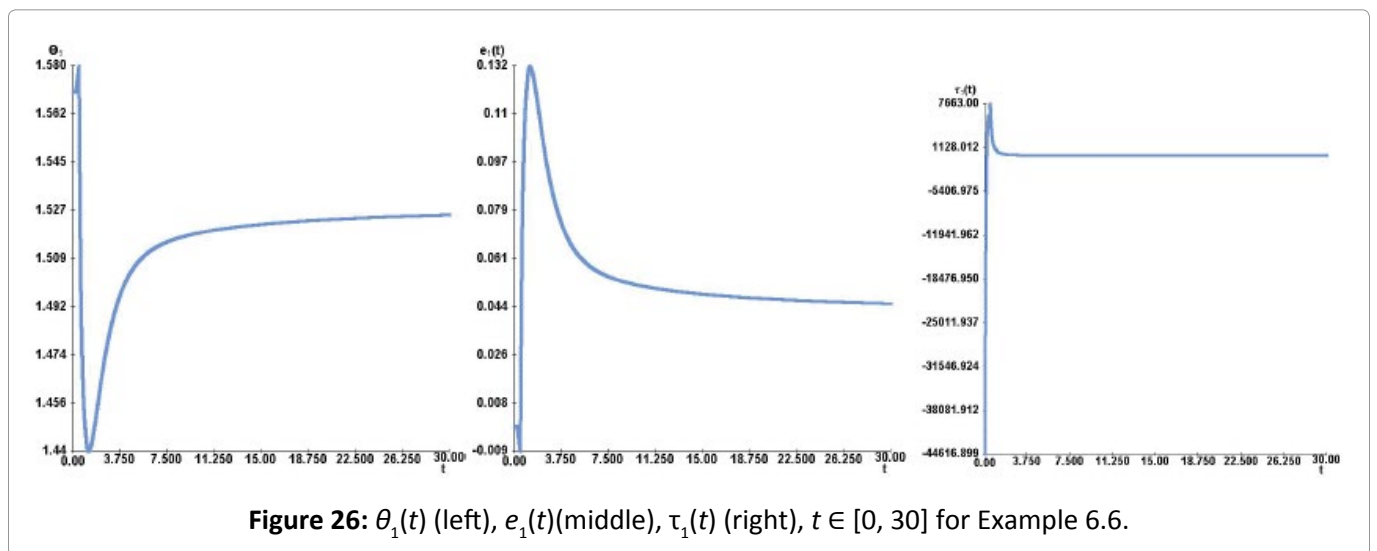


Figure 26: $\theta_1(t)$ (left), $e_1(t)$ (middle), $\tau_1(t)$ (right), $t \in [0, 30]$ for Example 6.6.

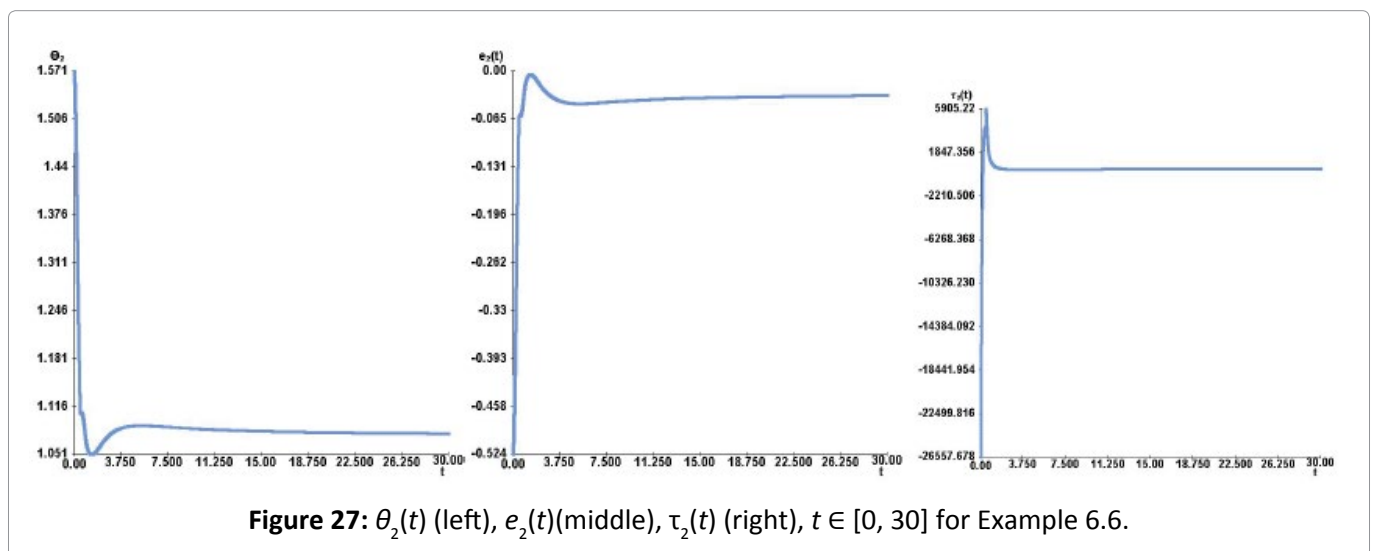
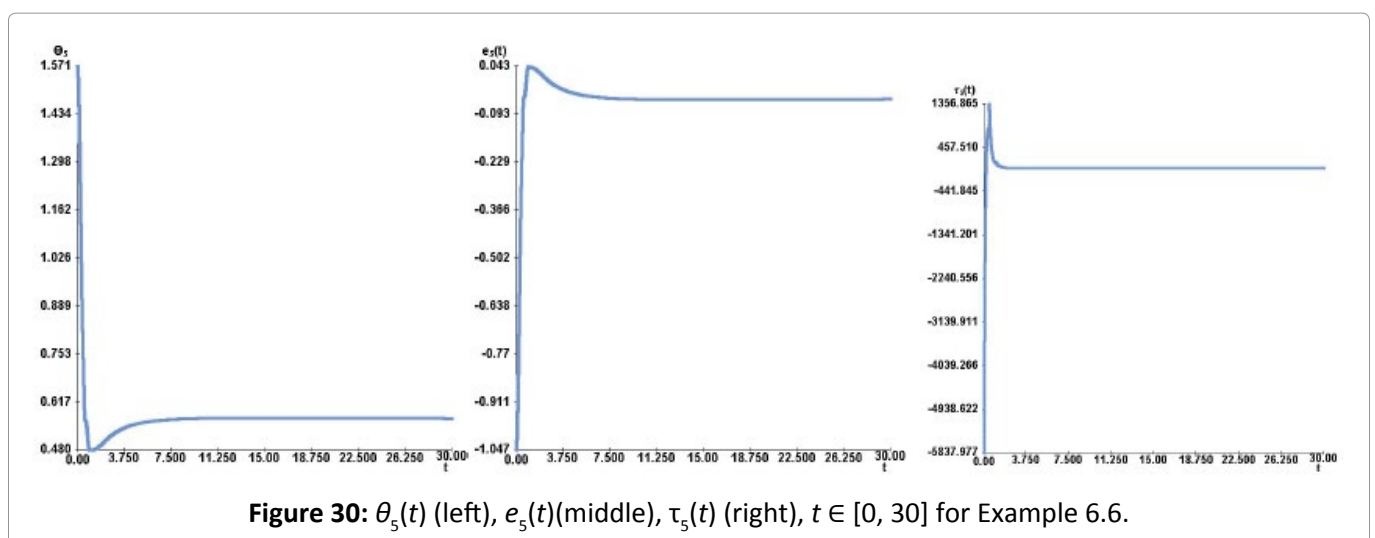
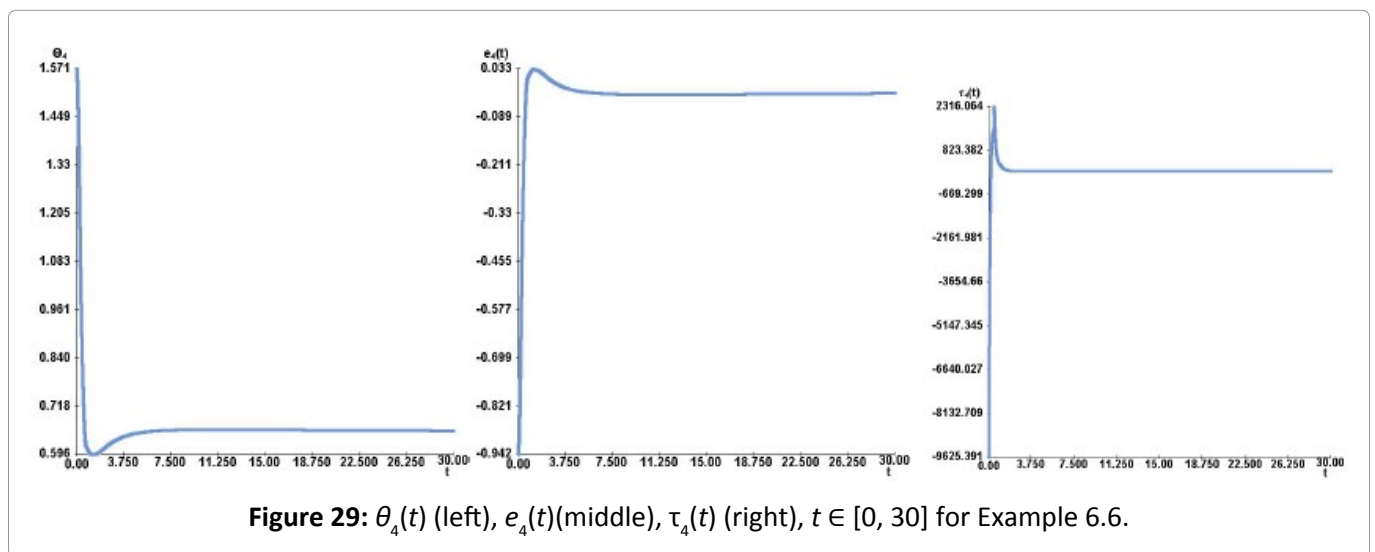
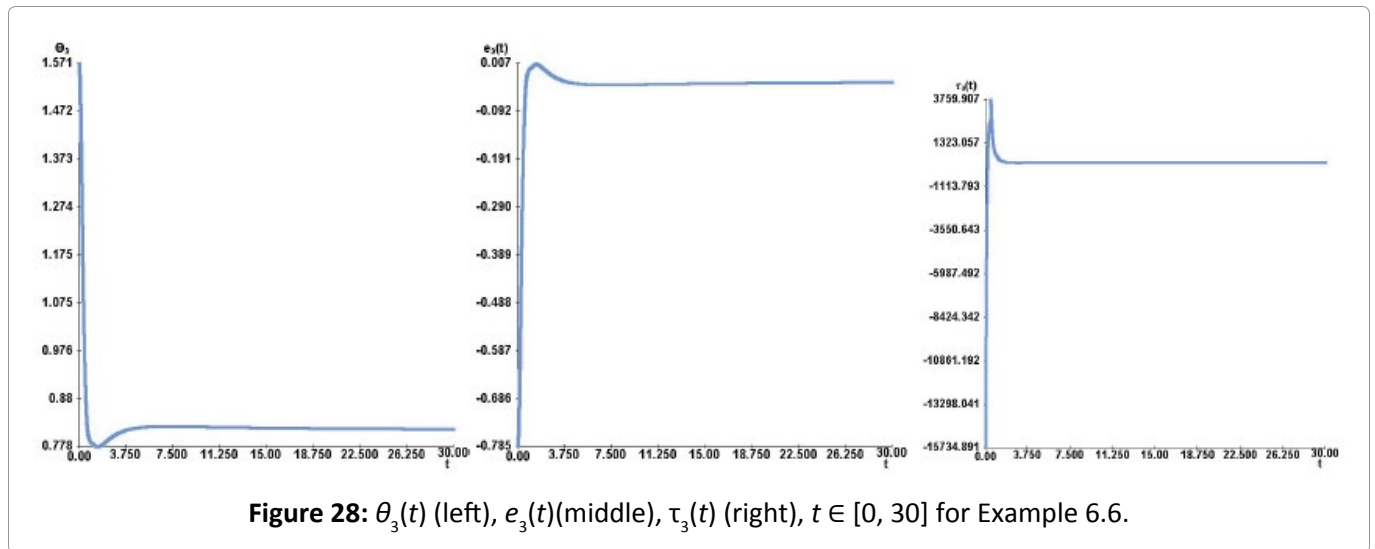


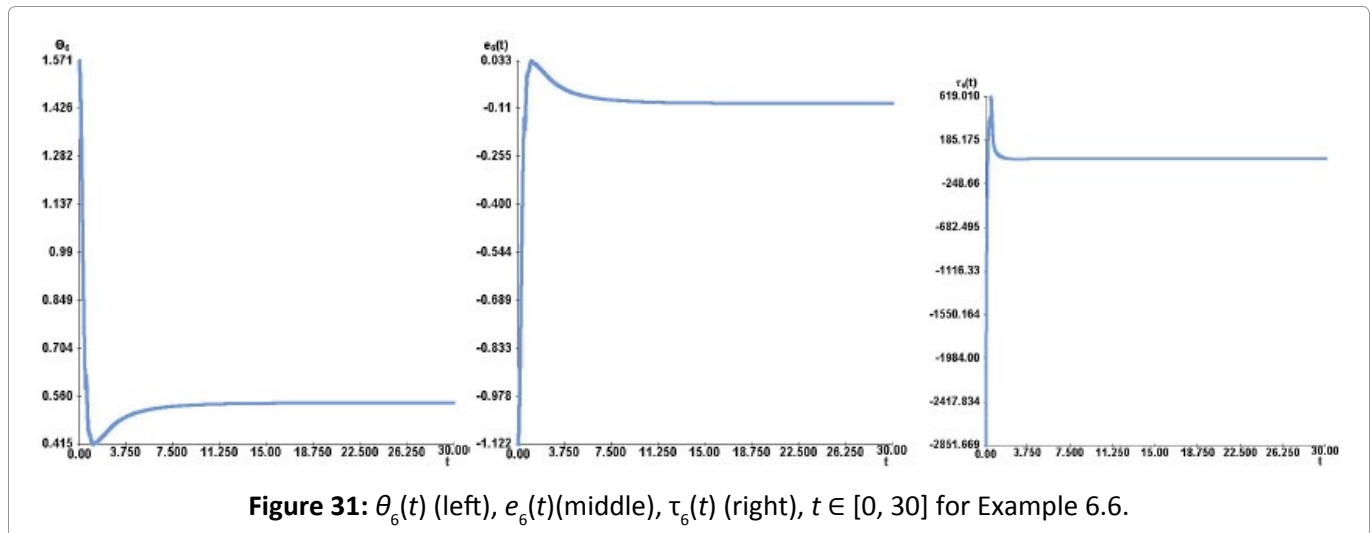
Figure 27: $\theta_2(t)$ (left), $e_2(t)$ (middle), $\tau_2(t)$ (right), $t \in [0, 30]$ for Example 6.6.



Conclusion

In this paper, we introduced a Proportional-Integral-Derivative (PID) control approach for an n -link robot manipulator, aimed at guiding its

motion to achieve target positions from given initial conditions. We first derived the system's equations of motion using the Lagrangian formulation. Subsequently, the PID is implemented, and the resulting second-order system of ordinary



differential equations is approximated using the classical fourth-order Runge-Kutta method for systems. The efficiency of the control strategy is demonstrated through a series of numerical experiments, involving the simulation of the manipulator's motion from an initial position to a desired target position. This paper focuses on the n -link robot manipulator in two-dimensional space. The PID controller introduced in this paper can be extended to accommodate robots with prismatic joints. In the future, we are planning to extend our approach to the n -link robot manipulator in three-dimensional space, where we will consider other typical joint types, including revolute (the attached links rotate about a common axis), prismatic (the attached links translate about a common axis) and spherical (the attached links rotate about a point).

Conflict of Interest

The authors declare that they have no conflicts of interest.

Ethical Statement

The authors declare that the material is their original work, which has not been previously published and is not currently being considered elsewhere.

Data Availability Statement

Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

References

- Lewis FL, Dawson DM, Abdallah CT (2003) Robot manipulator control: Theory and Practice. (2nd edn), CRC Press, Boca Raton.
- Johnson MA, Moradi MH (2005) PID control, Springer.
- Gopal D, Muruges V, Murugesan K (2006) Numerical solution of second-order robot arm control problem using Runge-Kutta-Butcher algorithm. International Journal of Computer Mathematics 83: 345-356.
- Leena G, Ray G (2012) A set of decentralized PID controllers for an n -link robot manipulator. Sadhana 37: 405-423.
- Baccouch M, Dodds S (2020) A two-link robot manipulator: Simulation and control design. Int J Robot Eng 5: 028.
- Ma H, Ren H, Zhou Q, Li H, Wang Z (2022) Observer-Based Neural Control of N -link Flexible-Joint Robots, IEEE Transactions on Neural Networks and Learning Systems 35: 5295-5305.
- Zhai J, Li Z (2021) Fast-exponential sliding mode control of robotic manipulator with super-twisting method. IEEE Transactions on Circuits and Systems II: Express Briefs 69: 489-493.
- Kumar SA, Chand R, Chand RP, Sharma B (2023) Linear manipulator: Motion control of an n -link robotic arm mounted on a mobile slider. Heliyon 9: e12867.
- Sa'nchez-Sa'nchez P, Arteaga-Pe'erez MA (2012) Simplified methodology for obtaining the dynamic model of robot manipulators. International Journal of Advanced Robotic Systems. 9.
- Islam MA (2015) Accurate solutions of initial value problems for ordinary differential equations with the fourth order Runge Kutta method. Journal of Mathematics Research 7: 41.
- Murray RM, Li Z, Sastry SS (2017) A mathematical introduction to robotic manipulation. CRC press.
- Musa H, Saidu I, Waziri MY (2010) A simplified derivation and analysis of fourth order Runge Kutta method. International Journal of Computer Applications 9: 51-55.