Trajectory tracking control of two-link industrial robot manipulator based on C++

Bose Selvam*

Department of Mechanical Engineering,
Mount Zion College of Engineering and Technology,
Pudukkottai, India
Email: selvammech.me@gmail.com
*Corresponding author

U. Natarajan

Department of Mechanical Engineering,
Alagappa Chettiar Government College of
Engineering and Technology,
Karaikudi, India
Email: nattu33@gmail.com

M. Balasubramonian

Department of Electrical and Electronics Engineering,
Alagappa Chettiar Government College of
Engineering and Technology,
Karaikudi, India
Email: balu94rec@gmail.com

S.K. Lakshmanaprabu

Department of Electronics and Instrumentation Engineering,
B.S. Abdur Rahman Crescent Institute of Science and Technology,
Chennai, India
Email: prabusk.leo@gmail.com

Abstract: In this paper, trajectory tracking of two-link planar rigid robot manipulator using proportional-integral-derivative (PID) control law based on open source, C++ software has been presented. Trajectory tracking control of industrial robot manipulator is an important task for a control engineer in order to increase the productivity in the manufacturing sector utilising manipulator. The significance of this paper is the demonstration of the importance of open software C++ for trajectory tracking control analysis of robot manipulator instead of utilising high-end commercial software like MATLAB which is costly. Fourth order Runge-Kutta (RK4) method has been utilised to solve the control problem represented as a simultaneous differential equation and the simulation algorithm is written as codes in C++ environment. Proportional-integral-derivative (PID) control law has been implemented for the trajectory control task. The proposed cost-free approach provides researchers and students a better platform for dynamic and control analysis of complex dynamic system such as robot manipulator. The simulation results obtained in both MATLAB and C++ environments are similar and hence the proposed C++ based trajectory control analysis has been validated.

Keywords: C++; trajectory control; rigid manipulator; two-link; gnuplot; Runge-Kutta; MATLAB; proportional-integral-derivative; PID; open source; Simulink.

Reference to this paper should be made as follows: Selvam, B., Natarajan, U., Balasubramonian, M. and Lakshmanaprabu, S.K. (2019) 'Trajectory tracking control of two-link industrial robot manipulator based on C++', *Int. J. Rapid Manufacturing*, Vol. 8, Nos. 1/2, pp.3–15.

Biographical notes: Bose Selvam received his Bachelor's of Technology in Mechanical Engineering from the SASTRA University and Master's of Engineering in Manufacturing Engineering from the ACGCET, India. He is currently pursuing his PhD program in the field of robotics.

U. Natarajan is currently working as an Assistant Professor and the Head of the Department of Mechanical Engineering, Alagappa Chettiar Government College of Engineering and Technology and received his PhD in Micro Machining from the Anna University, Chennai, India. His research interests include micro machining of components and algorithm design.

M. Balasubramonian is currently working as an Assistant Professor in the Department of Electrical and Electronics Engineering and received his BE in Electrical and Electronics Engineering in REC, Trichy, India. He received his PhD in Power Electronics from the Anna University, Chennai, India. His research interests include image processing and algorithm design.

S.K. Lakshmanaprabu is a Senior Research Fellow and he is currently pursuing his PhD in Multivariable Process Control in the Department of Electronics and Instrumentation Engineering of B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India. He received award of Rajiv Gandhi Nation Fellowship from University Grand Commission, Government of India, Delhi, for doing his PhD degree in the years 2013–2018. His areas of interest include multivariable control, evolutionary algorithm, fuzzy logic control, artificial intelligence, internet of things, image processing, model based development and hardware in the loop testing.

1 Introduction

Robot manipulators have been extensively used in industries for various tasks such as painting, assembly, pick and place, etc. The task of tracking desired trajectory of the manipulator is a challenging task for a control engineer for a system which involves complex dynamics and nonlinearities. The precise trajectory tracking control of robot manipulators in industrial applications such as assembly is a prerequisite (Raibert and Craig, 1981). Various control approaches for desired end point position control for two link robot manipulator have been presented and compared (Green and Sasiadek, 2004). Inverse dynamics robot model along with fuzzy logic control law provides greater endpoint accuracy. Pole placement technique has been applied to control the trajectory of two link manipulator (Mutoh et al., 2015). Sliding mode neural network control for precise tracking of position of two link manipulator has been proposed (Wai, 2003). Trajectory tracking of rigid robot manipulator considering disturbances and unknown parameters using iterative learning control schemes has been presented (Tayebi, 2004).

Proportional-integral-derivative (PID) control has been used in industries for numerous applications due to its simple nature of implementation and it eliminates steady state error through integral action and anticipates future through derivative action (Astrom and Hagglund, 1995; James, 2011). Authors have discussed in literatures about PID control, neural network and optimisation of process parameters in the field of manufacturing and control (Naiju et al., 2012; Godfrey and Nair, 2011). Adaptive neural network approach based on recurrent neural networks, lyapunov functions and PID control for trajectory tracking control of two-link robot manipulator has been presented (Perez et al., 2012). The PID controller parameters have been optimised (Ayala and dos Santos Coelho, 2012) using multi objective genetic algorithm for trajectory control of two-link rigid planar robot manipulator. PID controller with compensator to drastically improve the performance of the conventional industrial standard PID control has been proposed for trajectory tracking of six degree of freedom (d.o.f.) parallel manipulator (Shukla and Karki, 2014). Kelly (1995) suggested new PID controller tuning rules to improve trajectory tracking performance of the controller for two d.o.f. manipulator. Nonlinear PID control algorithm for precise tracking of six d.o.f. parallel manipulator (Su et al., 2004) has been proposed, where nonlinear combination of proportional, integral and derivative action of control error has been presented. Semiglobal stability with small output tracking error on application of PID control has been presented (Alvarez Ramirez et al., 2003; Parra Vega et al., 2003) for trajectory tracking of robot manipulator. PD-PID controller for motion tracking and vibration control of two link flexible manipulator has been proposed and the performance is investigated considering payload variations (Mahamood and Pedro, 2011). Trajectory tracking performance by various control strategies such as PID, sliding mode, computed torque and neural network on selective compliance assembly robot arm (SCARA) has been investigated (Visioli and Legnani, 2002). A new variable structure PID control consists of sliding mode PID with PID sliding surface has been proposed (Jafarov et al., 2005). Vibration control on various applications using complaint mechanism based on topology optimisation technique has been presented (Vijayan and Karthikeyan, 2014).

Modelling accuracy is not the only requirement in case of real-time control of a system as it imposes constraints such as timing, memory allocation and robustness. These constraints may be satisfied by realisation of manual C++ coding representing the system (Hopler and Otter, 2001). An object oriented C++ library has been designed to compute kinematics and dynamics of a system in real-time mode. Inverse dynamics of six d.o.f. manipulator model have been computed in 36 μs on 200 MHz in pentium PC. Toolboxes for MATLAB have been developed (Corke, 1996; Honey and Jamshidi, 1992; Dean-Leon et al., 2012) with efficient algorithms to compute kinematics and dynamics of robot manipulators. A Simulink library has been developed for modelling and control of robot manipulators (Saha and Shankar, 2012) which also provides C code for the developed model. The interpreted nature of MATLAB makes it less efficient in realisation of real-time implementation of a system (Gourdeau, 1997) whereas C++ programming language provides attractive features such as portability and efficient real-time implementation of a system without any use of any commercial software. Toolbox based on tools such as C++, Open GL and XML for dynamic modelling and real-time simulation of legged robots has been developed (Alexander et al., 2003). Time optimal path parameterisation for a given trajectory under kinodynamic constraints has been addressed (Pham, 2014) and implementation of the procedure in C++/Python is presented. C++ based three phase induction motor model has been developed (Kim et al., 2016) and compared the simulation results of the model for validation with high end commercial MATLAB/Simulink-based induction motor model. The mathematical model representing two-link planar rigid robot manipulator given in (Craig, 2005) has been considered in this paper. The contribution of this paper is development of C++ based solution for the trajectory tracking control of two-link planar rigid robot manipulator in order to replace the high end commercial MATLAB/Simulink-based approach for dynamic and control analysis of complex system like robot manipulator. PID control law has been implemented in the dynamic model for trajectory control of the manipulator. The simultaneous differential equations representing the dynamics and control are solved using fourth order Runge-Kutta (RK4) method. The entire procedure is written and executed in C++ platform and the control responses are generated using gnuplot plotting tool which is open software able to plot data executed as data file in C++ environment.

2 Description of two-link planar robot manipulator

The dynamic motion of two-link rigid planar robot manipulator is described as two coupled differential equations. The differential equations of motion characterising the dynamics consist of inertia, centrifugal, coriolis and gravity terms. The links of the manipulator are moved from one point to another in a desired trajectory by applying time varied torque at each joints of the manipulator. The equation of motion representing the dynamics of robot manipulator is generally described as

$$M(\theta)\ddot{\theta} + C\left(\theta, \dot{\theta}\right) + G(\theta) = \tau \tag{2.1}$$

where $M(\theta)$ is positive definite or inertia matrix of the manipulator, $C(\theta, \dot{\theta})$ is a vector consists of centrifugal and coriolis terms of the manipulator and $G(\theta)$ is a vector containing gravity terms of the manipulator, τ is a vector of applied torques at each joint of the manipulator. $\theta, \dot{\theta}, \ddot{\theta}$ are angular position, angular velocity and angular acceleration respectively. Figure 1 represents the two-link planar robot manipulator considered in this paper. The differential equations representing the dynamics of the two-link planar rigid robot manipulator are expressed as

$$\begin{aligned} \tau_1 = {} & m_2 l_2^2\left(\ddot{\theta}_1 + \ddot{\theta}_2\right) + m_2 l_1 l_2 c_2\left(2\ddot{\theta}_1 + \ddot{\theta}_2\right) + \left(m_1 + m_2\right) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 s_2 \dot{\theta}_2^2 \\ & -2 m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g c_{12} + \left(m_1 + m_2\right) l_1 g c_1 \end{aligned} \tag{2.2}$$

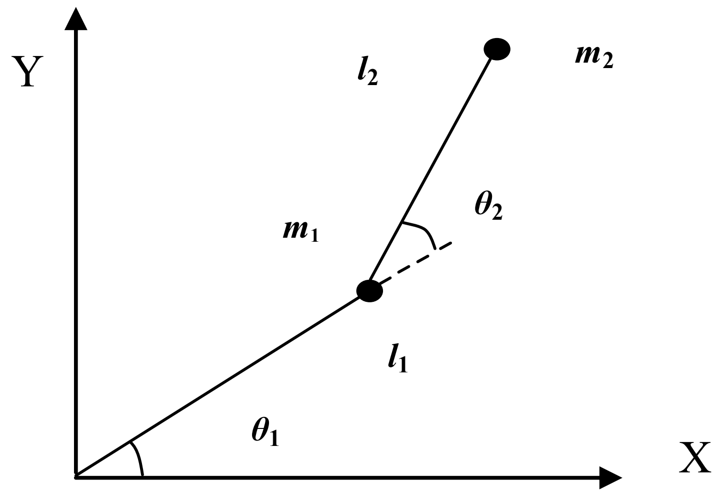
$$\tau_2 = m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_2 g c_{12} + m_2 l_2^2\left(\ddot{\theta}_1 + \ddot{\theta}_2\right) \tag{2.3}$$

where

$$s_{12} = \sin\left(\theta_1 + \theta_2\right); c_{12} = \cos\left(\theta_1 + \theta_2\right); c_1 = \cos\theta_1; c_2 = \cos\theta; s_1 \sin\theta_1; s_2 = \sin\theta_2$$

and subscripts 1 and 2 represents links 1 and 2 respectively.

Figure 1 Two-link planar robot manipulator



3 PID control law

The control of two-link robot manipulator in a desired trajectory is a complex task for a control engineer, as the dynamics involved is complex. PID control is a widespread technique for controlling industrial robot manipulators. In this paper, PID control is applied for trajectory control of the manipulator and is expressed in a time domain as

$$u(t) = k_p e(t) + k_i \int e(t)dt + k_d \frac{de(t)}{dt} \tag{3.1}$$

where $u(t)$ is a controller input or torque applied at the actuator of the manipulator, $e(t)$ is error or difference between actual and desired trajectory values over a period of time t, and k_p, k_i and k_d are proportional, integral and derivative gains respectively.

4 Implementation in C++

Trajectory tracking control of two-link robot manipulator in C++ environment is presented in this section. Before proceeding into the implementation of control law for tracking desired trajectory, solution for the differential equations of motion representing the dynamics of the two-link manipulator has to be derived. RK4 method has been utilised to solve the control problem of two-link robot manipulator and is illustrated as follows.

Runge-Kutta method of order 4 is given as (Jain, 1984)

$$
\begin{aligned}
f_0 &= f\left(t_n, x_n\right) \\
f_1 &= f\left(t_n + \frac{h}{2}, x_n + \frac{h}{2} f_0\right) \\
f_2 &= f\left(t_n + \frac{h}{2}, x_n + \frac{h}{2} f_1\right) \\
f_3 &= f\left(t_n + h, x_n + h f_2\right) \\
x_{n+1} &= x_n + h\left(\frac{1}{6} f_0 + \frac{1}{3} f_1 + \frac{1}{3} f_2 + \frac{1}{6} f_3\right)
\end{aligned}
\tag{4.1}
$$

where f_0 is function of time t and variable x to be determined, n is number of iterations, h is step interval, x_{n+1} is the solution for the variable at $(n + 1)^{\text{th}}$ iteration. Based on equation (4.1), the dynamic and trajectory control problem of two-link planar robot manipulator is solved. The above procedure is followed for solving the dynamics of the two-link robot manipulator, represented in equations (2.2) and (2.3). The two second order equations are converted into four first order equations and solved simultaneously utilising RK4 method. Before proceeding into the conversion, the equations (2.2) and (2.3) are written in the form as follows:

$$
\ddot{\theta}_1 = \frac{\begin{array}{c} \left(l_2^2 m_2\right) \times \left(\tau_1 + m_2 l_1 l_2 \sin\theta_2 \dot{\theta}_2^2 + 2 m_2 l_1 l_2 \sin\theta_2 \dot{\theta}_1 \dot{\theta}_2 - m_2 l_2 g \cos\left(\theta_1 + \theta_2\right)\right. \\ \left. - \left(m_1 + m_2\right) l_1 g \cos\theta_1\right) - \left(l_2^2 m_2 + l_1 l_2 m_2 \cos\theta_2\right) \\ \times \left(\tau_2 - m_2 l_1 l_2 \sin\theta_2 \dot{\theta}_1^2 - m_2 l_2 g \cos\left(\theta_1 + \theta_2\right)\right) \end{array}}{\begin{array}{c} \left(l_2^2 m_2 + 2 l_1 l_2 m_2 \cos\theta_2 + l_1^2 \left(m_1 + m_2\right)\right) \times \left(l_2^2 m_2\right) \\ - \left(l_2^2 m_2 + l_1 l_2 m_2 \cos\theta_2\right) \times \left(l_2^2 m_2 + l_1 l_2 m_2 \cos\theta_2\right) \end{array}}
\tag{4.2}
$$

$$
\ddot{\theta}_2 = \frac{\begin{array}{c} - \left(l_2^2 m_2 + l_1 l_2 m_2 \cos\theta_2\right) \times \left(\tau_1 + m_2 l_1 l_2 \sin\theta_2 \dot{\theta}_2^2 + 2 m_2 l_1 l_2 \sin\theta_2 \dot{\theta}_1 \dot{\theta}_2\right. \\ \left. - m_2 l_2 g \cos\left(\theta_1 + \theta_2\right) - \left(m_1 + m_2\right) l_1 g \cos\theta_1\right) + \left(l_2^2 m_2 + 2 l_1 l_2 m_2 \cos\theta_2\right. \\ \left. + l_1^2 \left(m_1 + m_2\right)\right) \times \left(\tau_2 - m_2 l_1 l_2 \sin\theta_2 \dot{\theta}_1^2 - m_2 l_2 g \cos\left(\theta_1 + \theta_2\right)\right) \end{array}}{\begin{array}{c} \left(l_2^2 m_2 + 2 l_1 l_2 m_2 \cos\theta_2 + l_1^2 \left(m_1 + m_2\right)\right) \times \left(l_2^2 m_2\right) \\ - \left(l_2^2 m_2 + l_1 l_2 m_2 \cos\theta_2\right) \times \left(l_2^2 m_2 + l_1 l_2 m_2 \cos\theta_2\right) \end{array}}
\tag{4.3}
$$

The new variables are introduced in order to convert the above mentioned second order equations into four first order equations.

$$
z = \begin{cases} z_1 = \theta_1 \\ z_2 = \theta_2 \\ z_3 = \dot{\theta}_1 \\ z_4 = \dot{\theta}_2 \end{cases}; \quad \dot{z} = \begin{cases} \dot{z}_1 = \dot{\theta}_1 \\ \dot{z}_2 = \dot{\theta}_2 \\ \dot{z}_3 = \ddot{\theta}_1 \\ \dot{z}_4 = \ddot{\theta}_2 \end{cases}
$$

With the above formulation, second order equations (4.2) and (4.3) are converted into first order are as follows:

$$
\dot{z}_1 = \dot{\theta}_1 \tag{4.4}
$$

$$\dot{z}_2 = \dot{\theta}_2 \tag{4.5}$$

$$\dot{z}_3 = \frac{\begin{array}{c}\left(l_2^2 m_2\right)\times\left(\tau_1 + m_2 l_1 l_2 \sin z_2 z_4^2 + 2m_2 l_1 l_2 \sin z_2 z_3 z_4 - m_2 l_2 g \cos\left(z_1 + z_2\right)\right. \\ \left. -\left(m_1 + m_2\right) l_1 g \cos z_1\right) - \left(l_2^2 m_2 + l_1 l_2 m_2 \cos z_2\right) \\ \times\left(\tau_2 - m_2 l_1 l_2 \sin z_2 z_3^2 - m_2 l_2 g \cos\left(z_1 + z_2\right)\right)\end{array}}{\begin{array}{c}\left(l_2^2 m_2 + 2 l_1 l_2 m_2 \cos z_2 + l_1^2\left(m_1 + m_2\right)\right)\times\left(l_2^2 m_2\right) \\ -\left(l_2^2 m_2 + l_1 l_2 m_2 \cos z_2\right)\times\left(l_2^2 m_2 + l_1 l_2 m_2 \cos z_2\right)\end{array}} \tag{4.6}$$

$$\dot{z}_4 = \frac{\begin{array}{c}-\left(l_2^2 m_2 + l_1 l_2 m_2 \cos z_2\right)\times\left(\tau_1 + m_2 l_1 l_2 \sin z_2 z_4^2 + 2m_2 l_1 l_2 \sin z_2 z_3 z_4\right. \\ \left. - m_2 l_2 g \cos\left(z_1 + z_2\right) - \left(m_1 + m_2\right) l_1 g \cos z_1\right) + \left(l_2^2 m_2 + 2 l_1 l_2 m_2 \cos z_2\right. \\ \left. + l_1^2\left(m_1 + m_2\right)\right)\times\left(\tau_2 - m_2 l_1 l_2 \sin z_2 z_3^2 - m_2 l_2 g \cos\left(z_1 + z_2\right)\right)\end{array}}{\begin{array}{c}\left(l_2^2 m_2 + 2 l_1 l_2 m_2 \cos z_2 + l_1^2\left(m_1 + m_2\right)\right)\times\left(l_2^2 m_2\right) \\ -\left(l_2^2 m_2 + l_1 l_2 m_2 \cos z_2\right)\times\left(l_2^2 m_2 + l_1 l_2 m_2 \cos z_2\right)\end{array}} \tag{4.7}$$

The above first order equations representing the dynamics of the system are solved using RK4 method is as follows:

$$z_1 = z_1 + h \times z_3 \tag{4.8}$$

$$z_2 = z_2 + h \times z_4 \tag{4.9}$$

$$z_3 = z_3 + h \times \dot{z}_3 \tag{4.10}$$

$$z_4 = z_4 + h \times \dot{z}_4 \tag{4.11}$$

where h is the step time. From the above procedures, angular position of each link of the manipulator is obtained. Next step is to implement the control law for tracking desired trajectory of the manipulator. The mathematical representation of control input from PID controller applied at each joint of the manipulator is described as follows:

$$u_1(t) = k_{p1} e_1(t) + k_{i1} \int e_1(t) dt + k_{d1} \frac{de_1(t)}{dt} \tag{4.12}$$

$$u_2(t) = k_{p2} e_2(t) + k_{i2} \int e_2(t) dt + k_{d2} \frac{de_2(t)}{dt} \tag{4.13}$$

where $u_1(t)$ and $u_2(t)$ are controlled torque applied at joints 1and 2 of the two-link manipulator respectively for tracking desired trajectory, where

$$e_1(t) = r_1(t) - z_1(t)$$

$$e_2(t) = r_2(t) - z_2(t)$$

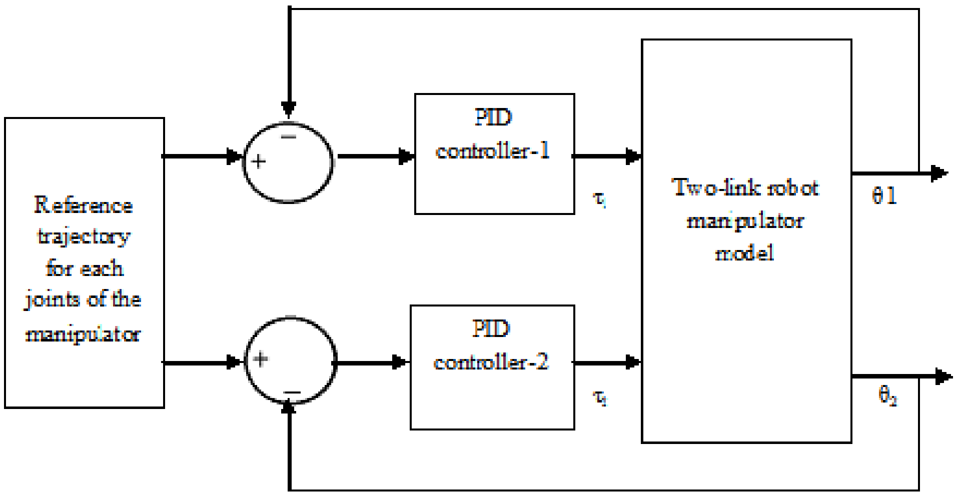
where $r_1(t)$ and $z_1(t)$ are reference and actual trajectory values respectively of joint 1, $r_2(t)$ and $z_2(t)$ are reference and actual trajectory values respectively of joint 2 with respect to time. Dynamic equations of motion of two-link manipulator with PID control are obtained by substituting equations (4.12) and (4.13) in equations (4.6) and (4.7) in place of τ_1 and τ_2 respectively and expressed as:

$$\ddot{\theta}_1 = \frac{\begin{array}{c}\left(l_2^2 m_2\right)\times\left(k_{p1}e_1(t)+k_{i1}\int e_1(t)dt+k_{d1}\dfrac{de_1(t)}{dt}+m_2 l_1 l_2 \sin z_2 z_4^2\right.\\ \left. +2m_2 l_1 l_2 \sin z_2 z_3 z_4 - m_2 l_2 g \cos\left(z_1+z_2\right)-\left(m_1+m_2\right)l_1 g \cos z_1\right)\\ -\left(l_2^2 m_2 + l_1 l_2 m_2 \cos z_2\right)\times\left(k_{p2}e_2(t)+k_{i2}\int e_2(t)dt+k_{d2}\dfrac{de_2(y)}{dt}\right.\\ \left. -m_2 l_1 l_2 \sin\theta_2\dot{\theta}_1^2 - m_2 l_2 g\cos\left(\theta_1+\theta_2\right)\right)\end{array}}{\begin{array}{c}\left(l_2^2 m_2 + 2l_1 l_2 m_2 \cos z_2 + l_1^2\left(m_1+m_2\right)\right)\times\left(l_2^2 m_2\right)\\ -\left(l_2^2 m_2 + l_1 l_2 m_2 \cos z_2\right)\times\left(l_2^2 m_2 + l_1 l_2 m_2 \cos z_2\right)\end{array}} \tag{4.14}$$

$$\ddot{\theta}_2 = \frac{\begin{array}{c}-\left(l_2^2 m_2 + l_1 l_2 m_2 \cos z_2\right)\times\left(k_{p1}e_1(t)+k_{i1}\int e_1(t)dt+k_{d1}\dfrac{de_1(t)}{dt}+m_2 l_1 l_2 \sin z_2 z_4^2\right.\\ \left. +2m_2 l_1 l_2 \sin z_2 z_3 z_4 - m_2 l_2 g \cos\left(z_1+z_2\right)-\left(m_1+m_2\right)l_1 g \cos z_1\right)\\ +\left(l_2^2 m_2 + 2l_1 l_2 m_2 \cos z_2 + l_1^2\left(m_1+m_2\right)\right)\times\left(k_{p2}e_2(t)+k_{i2}\int e_2(t)dt+k_{d2}\dfrac{de_2(t)}{dt}\right.\\ \left. -m_2 l_1 l_2 \sin z_2 z_3^2 - m_2 l_2 g\cos\left(z_1+z_2\right)\right)\end{array}}{\begin{array}{c}\left(l_2^2 m_2 + 2l_1 l_2 m_2 \cos z_2 + l_1^2\left(m_1+m_2\right)\right)\times\left(l_2^2 m_2\right)\\ -\left(l_2^2 m_2 + l_1 l_2 m_2 \cos z_2\right)\times\left(l_2^2 m_2 + l_1 l_2 m_2 \cos z_2\right)\end{array}} \tag{4.15}$$

The procedure for solving trajectory tracking control problem described above is implemented in C++ environment through manual coding. Gnuplot software has been utilised to plot the data file executed in C+ environment consisting of dynamic response of the manipulator model executed in C++. The dynamic model of two-link robot manipulator with PID control based on Simulink is given in Figure 2.

Figure 2 Simulink model of PID controlled two-link robot manipulator



5 Results and discussions

The reference trajectory to be followed by the joints of the two-link manipulator is a cubic polynomial and is given as

$$r_j(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3, \quad j = 1, 2 \tag{5.1}$$

where j is index of joint of the two-link manipulator. The first and second derivative of reference trajectory is given as

$$\dot{r}_j(t) = a_1 + 2a_2 t + 3a_3 t^2, \qquad j = 1, 2 \tag{5.2}$$

$$\ddot{r}_j(t) = 2a_2 + 6a_3 t, \qquad j = 1, 2 \tag{5.3}$$

The constraints of trajectory of two joints of the manipulator with respect to time (0 to 4 seconds) are given as follows

$$r_1(0) = 0; \quad r_1(2) = 1rad; \quad r_1(4) = 1rad$$

$$r_2(0) = 0; \quad r_2(2) = 2rad; \quad r_2(4) = 4rad$$

$$\dot{r}_1 = \dot{r}_2 = 0$$

where r_1 and r_2 are reference trajectory for joints 1 and 2 respectively. The physical parameters of the two-link robot manipulator considered in this paper are given in Table 1. The simulation is performed for the duration of four seconds. The scope of this article does not concentrate on optimisation, thus the optimised control parameters are taken from (Ayala and dos Santos Coelho, 2012) and given in Table 2. In order to validate the proposed method for analysing the trajectory control problem of two-link robot manipulator, the simulation results have been compared with that of MATLAB/Simulink-based model.

Table 1 Parameters of two-link robot manipulator model

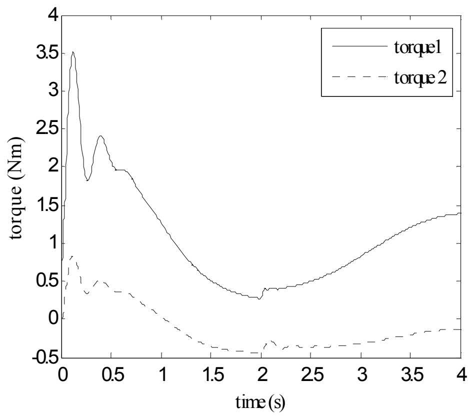
Parameters	*Link 1*	*Link 2*
Mass (kg)	0.1	0.1
Length (m)	0.8	0.4
Acceleration due to gravity (m/s^2)	9.81	

Table 2 Optimised values of PID controller parameters for each joint

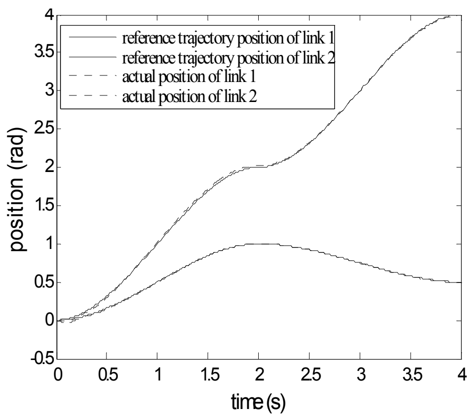
	Controller parameters					
	k_{p1}	k_{i1}	k_{d1}	k_{p2}	k_{i2}	k_{d2}
Values	184.76	49.68	8.94	11.46	16.54	0.20

The simulation results obtained using both C++ and MATLAB environments are shown through Figures 3–5. Figure 3 shows trajectory tracking control response using PID control based on MATLAB/Simulink model of two-link manipulator. In which Figure 3(a) shows the reference trajectory tracking performance of links 1 and 2 in radians and Figure 3(b) shows the time varied torques (output of PID controllers) provided at joints 1 and 2 for tracking the reference trajectory. The Simulink model is solved using ODE45 solver. Figure 4 shows trajectory tracking control response of two-link robot manipulator solved based on C++ environment, plotted through gnuplot software, in which Figure 4(a) and 4(b) represents reference trajectory tracking and time varied torque inputs respectively. Figure 5 shows the trajectory tracking error for joints 1 and 2, where Figures 5(a) and 5(b) shows tracking error obtained based on MATLAB and C++ platforms respectively. The simulation results shows that the trajectory control response for two-link robot manipulator model obtained based on C++ and MATLAB/Simulink environments is similar. The configuration of personal computer utilised for C++ based simulation is Intel Celeron processor with 1.70 GHz memory.

Figure 3 MATLAB-based response of robot manipulator, (a) controller input for joint 1 and joint 2 (b) trajectory tracking response

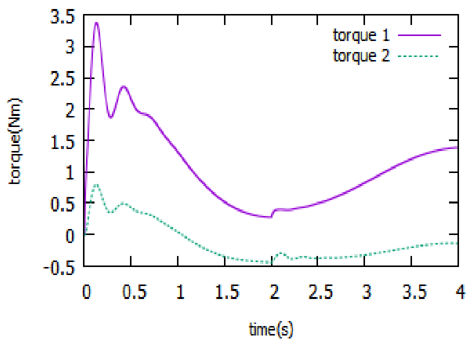


(a)

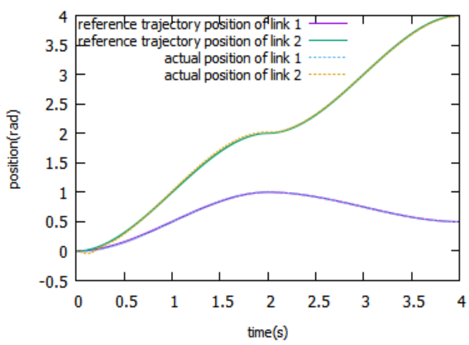


(b)

Figure 4 C++ based response of robot manipulator, (a) controller input for joint 1 and joint 2 (b) trajectory tracking response (see online version for colours)

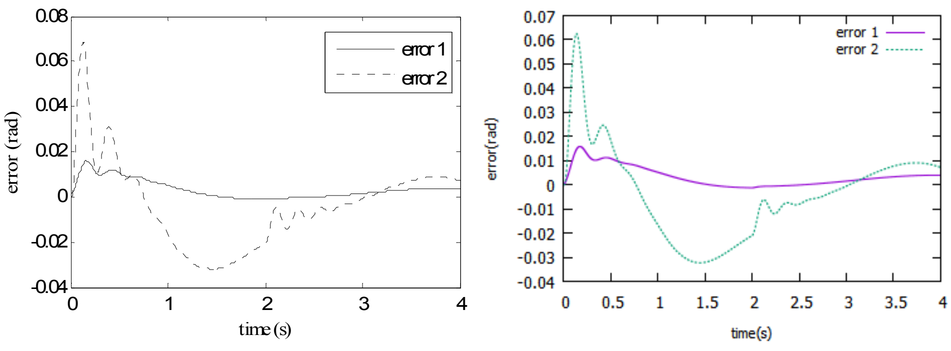


(a)



(b)

Figure 5 Trajectory tracking error, (a) trajectory tracking error (MATLAB) (b) trajectory tracking error (C++) (see online version for colours)



(a) (b)

6 Conclusions

The trajectory tracking control problem of two-link planar rigid robot manipulator with PID control law has been solved using RK4 method. The entire simulation algorithm is written in C++ environment. Gnuplot software is utilised for plotting the trajectory control response executed in C++ environment. Two-link robot planar rigid robot manipulator in Simulink has been developed. The simulation results obtained from both platforms have been compared for validation of the proposed C++ based trajectory tracking control method. The results demonstrate that the C++ based control analysis of complex dynamic system provides researchers a better option for dynamic analysis whereas the MATLAB-based analysis is an expensive approach in terms of acquiring copyright. The extension of this work will be performing real-time simulation of the control problem with the use of C++ codes developed in this paper, several digital processing units and software's.

References

Alexander, H., Hardt, M., von Stryk, O. and Hopler, R. (2003) 'Development of a toolbox for model based real time simulation and analysis of legged robots', *Proc. Appl. Math. Mech.*, Vol. 2, No. 1, pp.130–131.

Alvarez Ramirez, J., Kelly, R. and Cervantes, I. (2003) 'Semiglobal stability of saturated linear PID control for robot manipulators', *Automatica*, Vol.39, No. 6, pp.989–995.

Astrom, K.J. and Hagglund, T. (1995) *PID Controllers: Theory, Design and Tuning*, 2nd ed., Instrument Society of America, Research Triangle Park, North Carolina.

Ayala, H.V.H. and dos Santos Coelho, L. (2012) 'Tuning of PID controller based on a multi-objective genetic algorithm applied to a robotic manipulator', *Expert Syst. Appl.*, Vol. 39, No. 10, pp.8968–8974.

Corke, P.I. (1996) 'A robotics toolbox for MATLAB', *IEEE Robot Autom. Mag.*, Vol. 3, No. 1, pp.24–32.

Craig, J.J. (2005) *Introduction to Robotics: Mechanics and Control*, Vol. 3, pp.48–70, Pearson/Prentice Hall, Upper Saddle River, New Jersey, USA.

Dean-Leon, E, Nair, S. and Knoll, A. (2012) 'User friendly MATLAB-toolbox for symbolic robot dynamic modeling used for control design', in *Proceedings of the 2012 IEEE International Conference on Robotics and Biomimetics*, Guangzhou, China, pp.2181–2188.

Godfrey, W.W. and Nair, S.B. (2011) 'A bio-inspired technique for servicing networked robots', *Int. J. Rapid Manufacturing*, Vol. 2, No. 4, pp.258–279.

Gourdeau, R. (1997) 'Object-oriented programming for robotic manipulator simulation', *IEEE Robot Autom. Mag.*, Vol. 4, No. 3, pp.21–29.

Green, A. and Sasiadek, J.Z. (2004) 'Dynamics and trajectory tracking control of a two-link robot manipulator', *J. Vib. Control*, Vol. 10, No. 10, pp.1415–1440.

Honey, W.E. and Jamshidi, M. (1992) 'ROBO_SIM: a robotics simulation environment on personal computers', *Rob Auton. Syst.*, Vol. 9, No. 4, pp.305–317.

Hopler, R. and Otter, M. (2001) 'A versatile C++ toolbox for model based, real time control systems of robotic manipulators', in *Proceedings IEEE/RSJ International Conference Intelligent Robots and Systems*, Vol. 4, pp.2208–2214.

Jafarov, E.M., Parlakci, M.A. and Istefanopulos, Y. (2005) 'A new variable structure PID-controller design for robot manipulators', *IEEE Trans. Control Syst. Technol.*, Vol. 13, No. 1, pp.122–130.

Jain, M.K. (1984) *Numerical Solution of Differential Equations*, 2nd ed., John Wiley & Sons, New Delhi.

James, G. (2011*) Advanced Modern Engineering Mathematics*, 4th ed., Pearson Education Limited, England.

Kelly, R. (1995) 'A tuning procedure for stable PID control of robot manipulators', *Robotica*, Vol. 13, No. 2, pp.141–148.

Kim, J. et al. (2016) 'C++ based dynamic model of AC induction motor in discrete time domain', *IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC)*, Busan, Korea, pp.272–276.

Mahamood, R.M. and Pedro, J.O. (2011) 'Hybrid PD/PID controller design for two-link flexible manipulators', in *8th Asian Control Conference (ASCC)*, Kaohsiung, Taiwan, pp.1358–1363.

Mutoh, Y., Kemmotsu, M. and Awatsu, L. (2015) 'Trajectory tracking control of robot manipulators using discrete time-varying pole placement technique', *12th International Conference Informatics in Control, Automation and Robotics (ICINCO)*, Vol. 1, pp.373–379.

Naiju, C.D., Adithan, M. and Radhakrishnan, P. (2012) 'Multi-objective optimisation of DMLS process parameters using evolutionary algorithm', *Int. J. Rapid Manufacturing*, Vol. 3, No. 1, pp.23–35.

Parra Vega, V., Arimoto, S., Liu, Y.H., Hirzinger, G. and Akella, P. (2003) 'Dynamic sliding PID control for tracking of robot manipulators: theory and experiments', *IEEE Trans. Robot. Autom.*, Vol. 19, No. 6, pp.967–976.

Perez, J., Perez, J.P., Soto, R., Flores, A., Rodriguez, F. and Meza, J.L. (2012) 'Trajectory tracking error using PID control law for two-link robot manipulator via adaptive neural networks', *Procedia Technology*, Vol. 3, pp.139–146.

Pham, Q.C. (2014) 'A general, fast, and robust implementation of the time-optimal path parameterization algorithm', *IEEE Trans Robot*, Vol. 30, No. 6, pp.1533–1540.

Raibert, M.H. and Craig, J.J. (1981) 'Hybrid position/force control of manipulators', *J. Dyn. Syst. Meas. Control*, Vol. 103, No. 2, pp.126–133.

Saha, I. and Shankar, N. (2012) 'ModelRob: a Simulink library for model-based development of robot manipulators', *IEEE International Conference on Robotics and Automation*, RiverCentre, Saint Paul, Minnesota, USA, pp.2654–2659.

Shukla, A. and Karki, H. (2014) 'Modeling simulation & control of 6-DOF parallel manipulator using PID controller and compensator', *3rd International Conference on Advances in Control and Optimization of Dynamical Systems*, Vol. 47, pp.421–428.

Su, Y.X., Duan, B.Y. and Zheng, C.H. (2004) 'Nonlinear PID control of a six-DOF parallel manipulator', *IEE P-Contr. Theor. Ap.*, Vol. 151, No. 1, pp.95–102.

Tayebi, A. (2004) 'Adaptive iterative learning control for robot manipulators', *Automatica*, Vol. 40, No. 7, pp.1195–1203.

Vijayan, V. and Karthikeyan, T. (2014) 'Passive vibration isolation by compliant mechanism using topology optimization with building blocks', *Research Journal of Applied Sciences, Engineering and Technology*, Vol. 8, No. 13, pp.1522–1530.

Visioli, A. and Legnani, G. (2002) 'On the trajectory tracking control of industrial SCARA robot manipulators', *IEEE Trans. Ind. Electron*, Vol. 49, No. 1, pp.224–232.

Wai, R.J. (2003) 'Tracking control based on neural network strategy for robot manipulator', *Neurocomputing*, Vol. 51, pp.425–445.