DOI:10.13434/j.cnki.1007-4546.2019.0101

Dynamics and Proportional-derivative Control of 3-link Planar Manipulator

MUHAMMAD Adeel

(State Key Laboratory of Mechanics and Control of Mechanical Structures, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: In this paper, an effective dynamic equation of a three link manipulator for a control purpose has been dealt with by the Euler-Lagrange method. The structural properties of the derived dynamic equation were proved so that the vast control strategies developed for the serial counterparts can be easily extended for controlling the three link manipulator. In addition, it is illustrated how to design a PD controller for the robot manipulator by making use of computed torque method strategy to develop our controller. Simulation results are included in order to depict the performance of the controller.

Keywords: robotic manipulator; dynamic modeling; PD controller; euler-lagrange equation

1 Introduction

In robotics, a manipulator is a device used to manipulate materials without direct contact. The applications were originally for dealing with radioactive or bio hazardous materials, using robotic arms, or they were used in accessible places. In more recent developments, they have been used in applications such as robotically-assisted surgery and in space. The Euler-Lagrange equations give an expression of the dynamic equations of motion corresponding to those which may be obtained using Newton's second law. But the Lagrangian formalism is helpful for more difficult systems, such as multi-link robots.

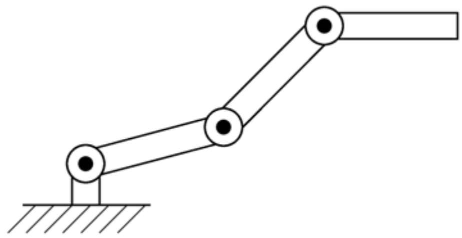

Figure 1 Three-link planar manipulator

To understand the complexity, we may take example of three degree of freedom robot manipulator movement. The total energy which is sum of kinetic energy and potential energy of the three link system is defined and used to form Lagrangian equations. Finally, to define the torque applied on each link these equations are used[1].

With the increase in performance demands for robotic manipulators during the last several years, a major need has arisen to design a new generation of manipulators that are lighter, faster, more controllable, and easier to program[2]. The reasonable dynamic modeling is a necessary premise to design efficient control strategies[3]. The Euler-Lagrange equations give an expression of the dynamic equations of motion corresponding to those which may be obtained using Newton's second law. But the Lagrangian formalism is helpful for more difficult systems, such as multi-link robots. Modeling of 3-link planar manipulator involves the linear and rotational dynamics of the links[4]. Several control strategies have been reported for control of these manipulators which include a proportional derivative (PD) controller, optimal control, adaptive control, fuzzy logic control and decomposed dynamic control. Several robust controllers have also been proposed for control of these manipulator systems[5]. Computed-torque control allows us to conveniently derive very effective robot controllers while providing a frame

Received October 22, 2018

work to bring together classical independent joint control and some modem design techniques. The author follows this strategy to develop our controllers[6]. The PD controller is for point to point motion control, while the PID controller is for vibration suppression[7]. PD control method is widely utilized for the dynamic characteristics controlling in industrial robot manipulator area[8]. For industrial robot manipulator system, PD control theory is extensively used in the dynamic characteristics controlling. A PD robust controller is introduced to optimize the stability and convergence of traditional PD controller and avoid excess initial driving torque for two-link industrial manipulator system[9].

The author has computed the inertia tensor for each link to synthesize PD laws of joint torques by developing the Matlab codes using the OriginTM software to plot and depict the simulation results.

2 Methodology

2.1 Inertia tensor computation

Let us consider the dynamics of robot manipulators. Since the dynamic equations describe the relationship between force and motion thoroughly, therefore the equations of motion are important to consider in the design of robots, in simulation and animation of robot motion, and also in the design of control algorithms.

In order to determine the Euler-Lagrange equations in a specific situation, the Lagrangian of the system is formed, which is in fact the difference between the kinetic energy and potential energy.

The Euler-Lagrange equations have properties that can be employed to design and analyze the feedback control algorithms, that is

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{y}} - \frac{\partial L}{\partial y} = f \qquad (1)$$

Where, L is a function of system which is the difference between the kinetic energy (K.E) and potential energy (P.E).

Generally, for any system, an application of the Euler-Lagrange equations governs a system of n coupled second order non-linear differential equations of the form, that is

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial \dot{q}_i} = \tau_i \qquad i = 1, \cdots, n \qquad (2)$$

The order n of the system is found by the number of generalized coordinates that are necessary to depict the development of the system.

A commonly used convention for selecting frames of reference in robotic applications is the Denavit-Hartenberg, or DH convention. In this practice, each homogeneous transformation is considered as a product of four basic transformations.

$$A_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \qquad (3)$$

Where: the four quantities are parameters which are generally given the names link length, link twist, link offset and joint angle respectively and are derived from specific aspects of the geometric relationship between two coordinate frames.

The matrix A_i is the homogeneous transformation matrix that expresses the position and orientation of $o_i x_i y_i z_i$ with respect to $o_{i-1} x_{i-1} y_{i-1} z_{i-1}$. It is not a constant matrix but varies as the configuration of the robot is changed.

To consider the assumption that all joints are either revolute or prismatic means that A_i is a function of only a single joint variable, namely q_i is

$$A_i = A_i(q_i) \tag{4}$$

Since the matrix A_i is a function of a single variable, so the three of the above four quantities are constant for a given line, while the fourth parameter, θ_i for a revolute joint and d_i for a prismatic joint, is the joint variable.

The n Denavit-Hartenberg joint variables serve as a set of generalized coordinates for an n-link rigid robot. For the case of an n-link robotic manipulator, we can express the K.E and P.E of the system in terms of a set of generalized coordinates considering the Euler-Lagrange equations which can be used to derive the dynamic equations.

As the Kinetic energy of a rigid object is the sum of two terms: the translational energy obtained by concentrating the entire mass of the object at the center of mass and the rotational K.E of the body about the center of the mass.

Considering the Figure 2

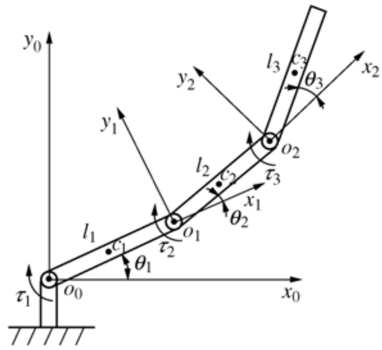

Figure 2 Three link planar robot manipulator

The K.E of the rigid body is given as

$$K = \frac{1}{2} m v^{\mathrm{T}} v + \frac{1}{2} \boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{I} \boldsymbol{\omega} \tag{5}$$

and

$$\| r_1 - r_2 \| = l \qquad \text{or} \qquad (r_1 - r_2)^{\mathrm{T}} (r_1 - r_2) = l^2 \tag{6}$$

Where: m represents the total mass of the object; $\boldsymbol{v}$ is the linear velocity vectors; $\boldsymbol{\omega}$ is the angular velocity vectors; $\boldsymbol{I}$ is symmetric matrix called Inertia Tensor.

Here, the linear and angular velocity vectors $\boldsymbol{v}$ and $\boldsymbol{\omega}$ respectively, are expressed in the inertial frame.

Also, ω is found from the skew symmetric matrix

$$S(\boldsymbol{\omega}) = \dot{R} R^{\mathrm{T}} \tag{7}$$

Where, R is the orientation transformation between the body attached frame and the inertial frame. A matrix $\boldsymbol{S}_{n\times n}$ is said to be Skew Symmetric if and only if $\boldsymbol{S}^{\mathrm{T}} + \boldsymbol{S} = 0$. It is necessary to express the inertia tensor, I also in the inertial frame in order to compute the triple product. The inertia tensor relative to the inertial reference frame will depend on the configuration of the object. To denote as $\boldsymbol{I}$ the inertia tensor expressed instead in the body attached frame, the two matrices are related via a similarity transformation.

Here, the inertia matrix expressed in the body attached frame is a constant matrix independent of the motion of the object and can be easily computed.

$$\boldsymbol{I} = R \boldsymbol{I} R^{\mathrm{T}} \tag{8}$$

Suppose the matrix density of the object be represented as a function of position $\rho(x, y, z)$. Then the inertia tensor in the body attached frame is computed as

$$\boldsymbol{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{9}$$

These integrals are computed over the region of space occupied by the rigid body. The diagonal elements of the inertia tensor I_{xx}, I_{yy}, I_{zz} are called the principal moments of inertia about the x, y, z axes, respectively. The off diagonal terms $I_{xy}, I_{xz}, I_{yz}, I_{yx}, I_{zx}, I_{zy}$ are called the cross products of inertia. If the mass distribution of the body is symmetric with respect to the body attached frame, then the cross products of inertia are identically zero. The z-axes all point out of the page, and are not shown in the figure. As we know that in 3D case here, target frame is frame 1 and reference frame is frame 0.

$$\boldsymbol{R}_1^0 = [x_1^0 \quad y_1^0 \quad z_1^0] \tag{10}$$

$$\boldsymbol{x}_1^0 = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix}, \boldsymbol{y}_1^0 = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix}, \boldsymbol{z}_1^0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{11}$$

In this case, the rotation matrix $\boldsymbol{R}$ is the orientation transformation between the body attached frame and the inertial frame.

$$\boldsymbol{R}_1^0 = \begin{bmatrix} x_1 . x_0 & y_1 . x_0 & z_1 . x_0 \\ x_1 . y_0 & y_1 . y_0 & z_1 . y_0 \\ x_1 . z_0 & y_1 . z_0 & z_1 . z_0 \end{bmatrix} \tag{12}$$

From Equation (8) and Equation (9), we see that $\boldsymbol{I}_{ij}$ is the inertia tensor expressed in the inertial frame, $\boldsymbol{I}_{ii}$ is the inertia tensor expressed in body attached frame.

Since the mass distribution of the body is symmetric with respect to the body attached frame, then the cross products of inertia are identically zero.

2.2 Deriving the equations of motion in matrix form

The dynamic model is expressed as

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{u} \tag{13}$$

Where: the inertia matrix $\boldsymbol{D}$ is a symmetric positive definite matrix that is in general configuration dependent matrix; $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is 3×3 vector of Coriolis and centrifugal torques; $\boldsymbol{g}(\boldsymbol{q})$ is N-vector of gravitational torques; $\boldsymbol{u}$ is 3×1 vector of the actuating torques; $\ddot{\boldsymbol{q}}$ is 3×1 vector of the joint accelerations; $\dot{\boldsymbol{q}}$ is 3×1 vector of the joint velocity; $\boldsymbol{q}$ is a vector of the joint position.

$$q_i = \begin{cases} \theta_i & \text{If joint } i \text{ is revolute} \\ d_i & \text{If joint } i \text{ is prismatic} \end{cases} \tag{14}$$

In the case of joints, with the i-th joint, i associate a joint variable, denoted by q_i. A constraint on the k coordinates $(r_1, \cdots, r_k)$ is called holonomic if it is an equality constraint of the form $g_i(r_1, \cdots, r_k) = 0$, $i = 1, \cdots, l$, and non-holonomic otherwise.

The forward kinematic equations define a function between the space of Cartesian positions and orientations and the space of joint positions. The velocity relationships are then determined by the Jacobian of this function. The Jacobian is a matrix that can be thought of as the vector version of the ordinary derivative of a scalar function.

For an n-link manipulator, we first derived the Jacobian representing the instantaneous transformation between the n-vector of joint velocities and the 6-vector consisting of the linear and angular velocities of the end-effector. This Jacobian is then a 6×n matrix. The same approach is used to determine the transformation between the joint velocities and the linear and angular velocity of any point on the manipulator.

A moving coordinate frame has both a linear and angular velocity. Linear velocity describes velocity of its origin (moving point).

If

$$R(t) \in SO(3) \tag{15}$$

Then

$$\dot{R}(t) = S(\omega(t))R(t) \tag{16}$$

Where, $\omega(t)$ is the instantaneous angular velocity of the frame. The operator S gives a skew symmetric matrix.

$$\boldsymbol{S}(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_z \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{17}$$

The manipulator Jacobian relates the vector of joint velocities to the body velocity $\boldsymbol{\xi} = (\boldsymbol{v}, \omega)^{\mathrm{T}}$ of the end effector, shown as

$$\boldsymbol{\xi} = \boldsymbol{J}\dot{q} \tag{18}$$

This relationship can be written as two equations, one for linear velocity and one for angular velocity, shown as

$$\boldsymbol{v} = J_v \dot{q}, \quad \omega = J_\omega \dot{q} \tag{19}$$

The i-th column of the Jacobian matrix corresponds to the i-th joint of the robot manipulator and takes one of the two forms depending on whether the i-th joint is prismatic or revolute.

$$J_i = \begin{cases} \begin{bmatrix} z_{i-1} \times (o_n - o_{i-1}) \\ z_{i-1} \end{bmatrix} & \text{If joint } i \text{ is revolute} \\ \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} & \text{If joint } i \text{ is prismatic} \end{cases} \tag{20}$$

As the body rotates, a perpendicular from any point of the body to the axis sweeps out an angle θ, and this angle is the same for every point of the body. If $\boldsymbol{\kappa}$ is a unit vector in the direction of the axis of rotation, then the angular velocity is given by

$$\boldsymbol{\omega} = \dot{\theta}\boldsymbol{\kappa} \tag{21}$$

Where, $\dot{\theta}$ is the time derivative of θ. The linear velocity of any point on the body is

$$\boldsymbol{v} = \boldsymbol{\omega} \times \boldsymbol{r} \tag{22}$$

Where, r is a vector from origin.

Consider a manipulator consisting of n-links. The linear and angular velocities of any point on any link can be expressed in terms of the Jacobian matrix and the derivative of the joint variables. For this case, the joint variables are indeed the generalized coordinates.

For appropriate Jacobian matrices $J\boldsymbol{v}_i$ and $J\boldsymbol{\omega}_i$

$$\boldsymbol{v}_i = J\boldsymbol{v}_i(q)\dot{q} \tag{23}$$

$$\omega_i = J\omega_i(q)\dot{q} \tag{24}$$

Suppose mass of link i is m_i and inertia matrix of link i is I_i(inertia matrix of link i, evaluated around a coordinate frame parallel to frame i whose origin is at the center of mass, equals I_i). We can derive the formula expressions for the kinetic energy and potential energy of a rigid robot using the Denavit-Hartenberg (DH) joint variables as generalized coordinates.

From Equation (6) and Equation (8), it follows that the overall kinetic energy of the manipulator equals

$$K = \sum_{i=1}^{n} [\frac{1}{2} m_i \boldsymbol{v}_i^{\mathrm{T}} \boldsymbol{v}_i + \frac{1}{2} \boldsymbol{\omega}_i^{\mathrm{T}} I_i \boldsymbol{\omega}_i] \tag{25}$$

Using Equation (14) and Equation (15), we have

$$K = \frac{1}{2} \dot{q}^{\mathrm{T}} \sum_{i=1}^{n} [m_i J\boldsymbol{v}_i(q)^{\mathrm{T}} J\boldsymbol{v}_i(q) + J\boldsymbol{\omega}_i(q)^{\mathrm{T}} R_i(q) I_i R_i(q)^{\mathrm{T}} J\boldsymbol{\omega}_i(q)] \dot{q} = \frac{1}{2} \dot{q}^{\mathrm{T}} D(q) \dot{q}$$

From Equation (18) after substituting the results, we can get

$$D(q) = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \tag{26}$$

Now, we will find the matrix $C(q, \dot{q})$ i.e. $C(\theta, \dot{\theta})$.

The k, j-thelement of the matrix $C(q, \dot{q})$ is defined as

$$c_{kj} = \sum_{i=1}^{n} c_{ijk}(q) \dot{q}_i \tag{27}$$

$$c_{ijk} = \frac{1}{2} \left\{ \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\} \tag{28}$$

The terms in Equation (28) are called as Christoffel symbols (of the first kind) for our three-link manipulator case,

$$c_{kj} = \sum_{i=1}^{3} c_{ijk}(q) \dot{q}_i = \sum_{i=1}^{3} \frac{1}{2} \left\{ \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\} \dot{q}_i \tag{29}$$

Sinceis concerned with potential energy, so we will not consider this factor in our case. Also, τ_i is the control input.

As we know that the complete dynamic model of an n-degrees-of-freedom manipulator is described by the matrix form of Euler-Lagrange equations

$$\boldsymbol{D}(\boldsymbol{q}) \ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{30}$$

Where: $\boldsymbol{D}(\boldsymbol{q})$ is 3×3 position dependent manipulator inertia matrix; $\boldsymbol{g}(\boldsymbol{q})$ is n-vector of gravitational torques, which we have not considered in our case.

Therefore, the author get

$$\boldsymbol{D}(\boldsymbol{q}) \ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}} = \boldsymbol{\tau} \tag{31}$$

In matrix form

$$\begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{32}$$

2.3 Synthesize PD law

Now we will synthesize PD or PID laws of joint torques so as to stabilize the three-link around a reference configuration that all the joints angles are of zero value.

Proportional-integral-derivative controller which is commonly known as PID controller is described as a control loop technique. It is mostly used for industry related applications. An important function of PID controller is to constantly determine an error value $e(t)$ as the difference between a required set point and a computed process variable and applies a correction depending on proportional, integral, and derivative terms which are denoted by P, I and D, respectively. That is why, it is commonly known as a PID controller.

In frequency domain, PID controller is expressed as

$$U(s) = K_p(\Theta^d(s) - \Theta(s)) - K_d s\Theta(s) + K_i \frac{\Theta^d(s) - \Theta(s)}{s} \tag{33}$$

In time domain using Inverse Laplace transformation, we have

$$u(t) = K_p(\Theta^d(t) - \Theta(t)) - K_d\dot{\Theta}(t) + K_i\int_0^t [\Theta^d(t) - \Theta(t)]\mathrm{d}t \tag{34}$$

Proportional-derivative (PD) control is useful for fast response controllers that do not need a steady-state error of 0. Proportional controllers are fast. Derivative controllers are fast. The two together is very fast. Proportional-Derivative or PD control combines proportional control and derivative control in parallel.

In order to reject a constant disturbance using PD control, large gains are often required. The input $U(s)$ is given in frequency domain as

$$U(s) = K_p(\Theta^d(s) - \Theta(s)) - K_d s\Theta(s) \tag{35}$$

Here, if we consider $\Theta^d(s) = 0$. So the above equation takes the form

$$U(s) = -K_p\Theta(s) - K_d s\Theta(s) \tag{36}$$

By using Inverse Laplace transformation, it can be expressed in the time domain as

$$u(t) = -K_p\Theta(t) - K_d\dot{\Theta}(t) \tag{37}$$

Now, for our case, since we have

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{\tau} \tag{38}$$

Since we are considering PD controller here, so

$$\boldsymbol{\tau} = -\boldsymbol{K}_p\boldsymbol{q} - \boldsymbol{K}_d\dot{\boldsymbol{q}} \tag{39}$$

Now, we want to convert the second order system into the first order by substitution method as follows:

$$\dot{\boldsymbol{q}} = \boldsymbol{p} \tag{40}$$

From Equation (38) and Equation (40), we have the expression as

$$\begin{cases}\dot{\boldsymbol{q}} = \boldsymbol{p} \\ \boldsymbol{D}(\boldsymbol{q})\dot{\boldsymbol{p}} + \boldsymbol{C}(\boldsymbol{q},\boldsymbol{p})\boldsymbol{p} = \boldsymbol{\tau}\end{cases} \tag{41}$$

From Equation (39)

$$\boldsymbol{\tau} = -\boldsymbol{K}_p\boldsymbol{q} - \boldsymbol{K}_d\boldsymbol{p} \tag{42}$$

Now

$$\boldsymbol{X} = \begin{bmatrix} q_{3\times1} \\ p_{3\times1} \end{bmatrix}_{6\times1} \tag{43}$$

$$K_p = K_d = I_{3\times3} \tag{44}$$

Equation (28) can be rewritten as follows.

$$\begin{bmatrix} \boldsymbol{E}_{3\times3} & 0 \\ 0 & \boldsymbol{D}(q)_{3\times3} \end{bmatrix}_{6\times6} \dot{\boldsymbol{X}}_{6\times1} = \begin{bmatrix} \boldsymbol{p}_{3\times1} \\ -\boldsymbol{C}(q,p)_{3\times3}\boldsymbol{p}_{3\times1} + \boldsymbol{\tau}_{3\times1} \end{bmatrix}_{6\times1} \tag{45}$$

Where:

$$\dot{\boldsymbol{X}}_{6\times1} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{p}} \end{bmatrix}, \quad \dot{\boldsymbol{q}} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}, \quad \dot{\boldsymbol{p}} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix}; \quad \boldsymbol{E}_{3\times3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Now, we will compute the matrix $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ i.e. the matrix $\boldsymbol{C}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})$ in symbolic form as

$$M\dot{X} = N, \qquad \dot{X} = M^{-1}N, \qquad \mathrm{d}X = M^{-1}N \tag{46}$$

So, we have

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} I & O \\ O & D(\theta) \end{bmatrix}^{-1} \begin{bmatrix} \dot{\theta} \\ -C(\theta,\dot{\theta})\dot{\theta} + \tau \end{bmatrix} \tag{47}$$

Using an additional line spacing of 12 points before the beginning of the double column section, as shown above.

3 Simulation results and analysis

The author has developed MATLAB codes for simulating the system dynamics in the presence of the toque control inputs, and use OriginTM software to plot and depict the simulation results.

Figure 3 ~ Figure 8 describe the simulation results. Here, we have considered the units of angle and time as radian and second respectively. From Figure 3 ~ Figure 5, the angles represent the position of respective links with respect to their corresponding x-axis. The author can see a significant fall and then a small rise in graph 0 ~ 10 s considerably smooth behavior in these three graphs 10 ~ 30 s. Whereas Figure 6 ~ Figure 8 depict the fashion of angular velocities with respect to time. It can be observed in these graphs that there is a sinusoidal sort of decrease in negative values in the beginning but soon there is a gradual increase till further minor decrease in positive values and then from 10 ~ 30 s, the same streamlined style is experienced.

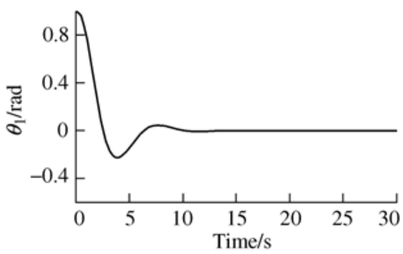

Figure 3 The time history of θ_1

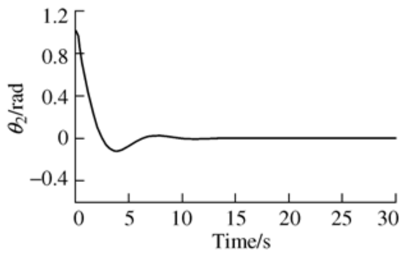

Figure 4 The time history of θ_2

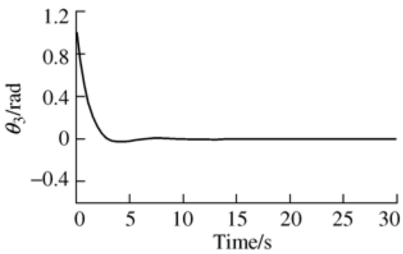

Figure 5 The time history of θ_3

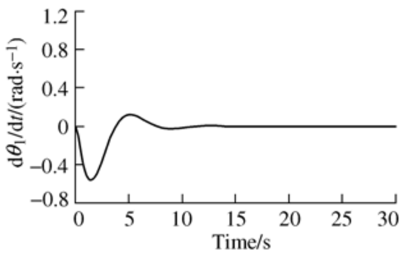

Figure 6 The time history of $\dot{\theta}_1$

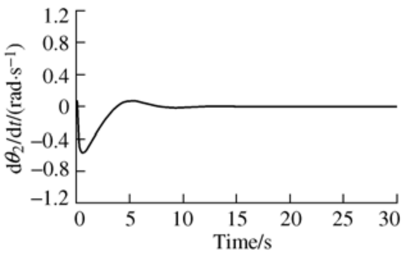

Figure 7 The time history of $\dot{\theta}_2$

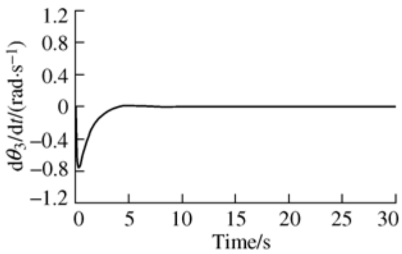

Figure 8 The time history of $\dot{\theta}_3$

4 Conclusions

In this paper, the author aimed to apply computed torque controller system for 3-link robot manipulator and stimulate the applied controller performance using MATLAB. The simulation results show the validity of the proposed method and give the possibility of a computed torque control. Dynamic modeling is the basic element for controller design of mechanisms. The structural properties of the derived dynamic equation were proved so that the vast control strategies developed for the serial counterparts can be easily extended for controlling the three link manipulator. The simulation results indicate the effectiveness of the proposed approach and demonstrate the satisfactory performance compared to the conventional controller in the presence of the parameter uncertainties and un-modelled dynamics for the motion control of manipulators. The developed control scheme can be further extended to make use in arm control of manipulators for target capturing applications i.e., in space robotics. This study is useful for the future real time implementation of motion control of space manipulators in complex dynamic scenario. The author has tried to show the design of a PD controller for the robot manipulator by following computed torque method strategy to develop our controllers. in order to try to find advantages of this control law by simulations.

References

[1] Shah J A, Rattan S S. Dynamic analysis of two link robot manipulator for control design using pid computed torque control[J]. International Journal of Research in Computer Applications and Robotics, 2016, 5(4):277-283

[2] Nye T W, Leblanc D J, Cipra R J. Design and modeling of a computer-controlled planar manipulator[J]. International Journal of Robotics Research, 1987, 6(1):85-95

[3] Cui M Y, Wu Z J, Xie X J. Stochastic modeling and tracking control for a two-link planar rigid robot manipulator[J]. International Journal of Innovative Computing Information and Control Ijicic, 2013, 9(4):1769-1780

[4] Sutar M K, Pathak P M, Mehta N K, et al. Trajectory control of a 3-link planar manipulator using virtual link based controller[C]// International and, National Conference on Machines and Mechanisms, 2013

[5] Mohamed Z, Khairudin M, Husain A R, et al. Linear matrix inequality-based robust proportional derivative control of a two-link flexible manipulator[J]. Journal of Vibration and Control, 2013, 22(5):1244-1256

[6] Eroglu E, Cansever G, Kucukdemiral L B, et al. PD, PID and fuzzy logic control of three link planar arm[C]// International Workshop on Robot Motion and Control. IEEE, 2004:191-196

[7] Mahamood R M, Pedro J O. Hybrid PD/PID controller design for two-link flexible manipulators[C]// Control Conference. IEEE, 2011:1358-1363

[8] Wang J Y. The design and simulation for two-link manipulators pd robust controller under uncertain upper bound disturbance[J]. Advanced Materials Research, 2013, 694-697:1652-1655

[9] Wang J Y, Zhuang Y X. A PD robust controller design and simulation based on two-link manipulator under given upper bound disturbance[J]. Advanced Materials Research, 2012, 503-504:1540-1544

Brief Biographies

MUHAMMAD Adeel is a master, Nanjing University of Aeronautics and Astronautics. His research interests are space robotics vision, control and capture. muhammadadeel@nuaa.edu.cn