

Shaoping Bai, Lelai Zhou, and Guanglei Wu

Contents

Introduction	1856
Recursive Newton-Euler Method	1856
Outward Recursion to Calculate Velocities and Accelerations	1857
Inward Recursion to Calculate Forces and Torques	1859
Lagrange's Equation	1859
Examples	1861
Example I: A 5-dof Lightweight Robotic Arm	1861
Example II: A Spherical Parallel Manipulator	1864
Summary	1871
References	1872

Abstract

This book chapter is about fundamentals of manipulator dynamics and their applications. Two approaches of manipulator dynamics, namely, *recursive Newton-Euler approach* and the *Lagrange equations*, are introduced and discussed. Examples are included to demonstrate their application in manipulator dynamics simulations and analysis. This book chapter can provide basic understanding on manipulator dynamics, which is applicable to manipulators, including serial and parallel manipulators.

S. Bai (✉) • L. Zhou • G. Wu

Department of Mechanical and Manufacturing Engineering, Aalborg University, Aalborg, Denmark

e-mail: shb@m-tech.aau.dk; lzh@m-tech.aau.dk; gwu@m-tech.aau.dk

© Springer-Verlag London 2015

A.Y.C. Nee (ed.), *Handbook of Manufacturing Engineering and Technology*,
DOI 10.1007/978-1-4471-4670-4_91

1855

Introduction

Manipulator dynamics concerns with the actuation forces required to generate motion of manipulators. The manipulator dynamics is expressed in terms of equations of motion, which describe the relationships between the actuation, reaction forces, and the accelerations and motion trajectories of manipulators. In robotics, dynamics is a fundamental for the manipulator design, analysis, and control.

Two types of problems are involved in the dynamics study, namely, the inverse and forward dynamics. The inverse dynamics calculates the required actuating forces at joints for a prescribed trajectories or known kinematic information (position, velocity, and acceleration). The forward dynamics, on the other hand, determines the joints' and end-effector accelerations with applied actuating forces at joints.

The equations of motion can be established with several approaches. The most used approaches include *recursive Newton-Euler approach* and the *Lagrange equations*, which can be found in literatures (Murray et al. 1994; Angeles 1997; Featherstone and Orin 2000). In this chapter, the dynamics equation formulation with the above two approaches is introduced. The formulations are confined within the rigid-body dynamics. Readers with interests in flexible body manipulators are directed to literatures (Shabana 2013; Low 1987; Dwivedy and Eberhard 2006). Two examples are included to demonstrate their applications, one applicable to a serial manipulator and the other to a parallel robot.

Recursive Newton-Euler Method

The motion of a rigid body can be described by the Newton and Euler's equations, the former for the translation and the latter for the rotation. Basically, the equations of motion of a manipulator can be established for each body of the manipulator on the basis of free-body diagrams. Alternatively, the equations of motion can be formulated through the recursive Newton-Euler method for serial manipulators, with which the free-body diagrams can be omitted. The formulation presented is based on the method introduced by Luh et al. (1980). This formulation is well suited for inverse dynamics problems, i.e., to find the driving torques with known kinematics information.

The recursive Newton-Euler method consists of two steps. The first step is an outward recursion, from the base to the end link, to calculate velocity and acceleration. The second step is an inward recursion, from the end link to the base, to calculate the forces and torques at joints. In the following formulation, the recursive method is applied to a manipulator with n rigid links. Coordinate systems are established by following Denavit-Hartenberg's convention (D-H convention) (Denavit and Hartenberg 1955).

Outward Recursion to Calculate Velocities and Accelerations

In the inverse dynamics problems, the positions, velocities, and accelerations ($\theta_i, \dot{\theta}_i, \ddot{\theta}_i$ for a revolute joint and $d_i, \dot{d}_i, \ddot{d}_i$ for a prismatic joint) of the joints are known. The outward recursion propagates velocities, and accelerations of each link from the base to the end link.

Revolute joints

Referring to Fig. 1, the angular velocity of a revolute joint, propagated from link $i-1$ to link i , is given by

$$\omega_i = \omega_{i-1} + \dot{\theta}_i z_i \tag{1}$$

or in local components

$$\mathbf{R}_i \omega'_i = \mathbf{R}_{i-1} \omega'_{i-1} + \dot{\theta}_i \mathbf{R}_i z'_i \tag{2}$$

where \mathbf{R}_i stands for the orientation matrix of link i and ω'_i is the angular velocity of link i with respect to the i th frame (local coordinate system). Premultiplying both sides of Eq. 2 with \mathbf{R}_i^T and simplifying yield

$$\omega'_i = \mathbf{R}_{i-1}^i \omega'_{i-1} + \dot{\theta}_i z'_i \tag{3}$$

where \mathbf{R}_{i-1}^i is the rotation matrix relating link $i-1$ and link i . z'_i is the unit vector $[0,0,1]^T$ parallel to the z_i axis. Similarly, the propagation of linear velocity from link $i-1$ to link i can be derived as

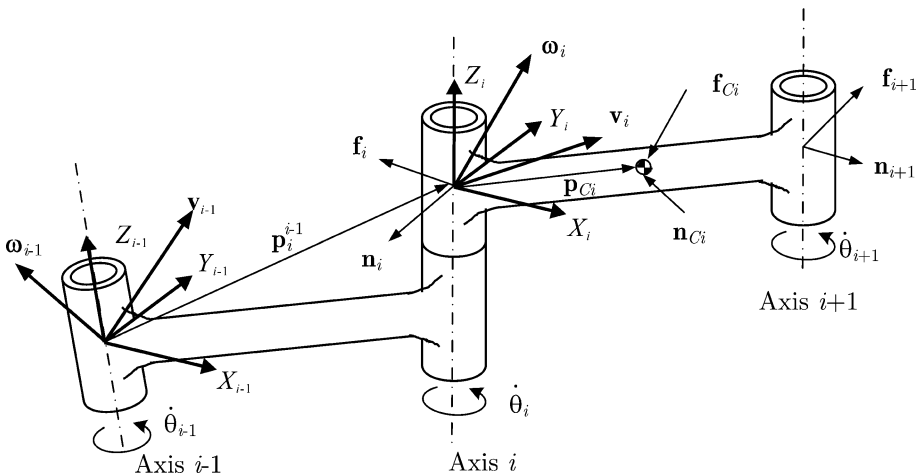


Fig. 1 The illustration of the symbols

$$\mathbf{v}'_i = \mathbf{R}_{i-1}^i (\mathbf{v}'_{i-1} + \boldsymbol{\omega}'_{i-1} \times \mathbf{p}_i^{i-1}) \quad (4)$$

where \mathbf{p}_i^{i-1} denotes the position vector of the frame of link i in the $(i-1)$ th frame.

The angular acceleration is calculated through

$$\dot{\boldsymbol{\omega}}'_i = \mathbf{R}_{i-1}^i \dot{\boldsymbol{\omega}}'_{i-1} + \ddot{\theta}_i \mathbf{z}'_i + \mathbf{R}_{i-1}^i \boldsymbol{\omega}'_{i-1} \times \dot{\theta}_i \mathbf{z}'_i \quad (5)$$

The linear acceleration with respect to the link-attached coordinate system is given by

$$\dot{\mathbf{v}}'_i = \mathbf{R}_{i-1}^i (\dot{\mathbf{v}}'_{i-1} + \dot{\boldsymbol{\omega}}'_{i-1} \times \mathbf{p}_i^{i-1} + \boldsymbol{\omega}'_{i-1} \times (\boldsymbol{\omega}'_{i-1} \times \mathbf{p}_i^{i-1})) \quad (6)$$

where $\dot{\mathbf{v}}'_i$ is the linear acceleration of link i in the i th frame.

The linear acceleration of the center of mass of link i is further calculated from the velocity and acceleration of the link

$$\dot{\mathbf{v}}'_{Ci} = \dot{\mathbf{v}}'_i + \dot{\boldsymbol{\omega}}'_i \times \mathbf{p}'_{Ci} + \boldsymbol{\omega}'_i \times (\boldsymbol{\omega}'_i \times \mathbf{p}'_{Ci}) \quad (7)$$

where \mathbf{p}'_{Ci} denotes the position vector of the center of mass of link i in its own frame.

Prismatic Joints

If the joints in question are prismatic, the angular velocities are identical for link i and link $i-1$, which means $\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1}$. Expressing with local components yields

$$\mathbf{R}_i \boldsymbol{\omega}'_i = \mathbf{R}_{i-1} \boldsymbol{\omega}'_{i-1} \quad (8)$$

that is,

$$\boldsymbol{\omega}'_i = \mathbf{R}_{i-1}^i \boldsymbol{\omega}'_{i-1} \quad (9)$$

where matrix \mathbf{R}_{i-1}^i is independent to rotations.

Likewise, the local angular acceleration of link i is

$$\dot{\boldsymbol{\omega}}'_i = \mathbf{R}_{i-1}^i \dot{\boldsymbol{\omega}}'_{i-1} \quad (10)$$

Furthermore, the local linear velocity and acceleration become

$$\mathbf{v}'_i = \mathbf{R}_{i-1}^i (\mathbf{v}'_{i-1} + \boldsymbol{\omega}'_{i-1} \times \mathbf{p}_i^{i-1}) + \dot{d}_i \mathbf{z}'_i \quad (11)$$

$$\dot{\mathbf{v}}'_i = \mathbf{R}_{i-1}^i (\dot{\mathbf{v}}'_{i-1} + \dot{\boldsymbol{\omega}}'_{i-1} \times \mathbf{p}_i^{i-1} + \boldsymbol{\omega}'_{i-1} \times (\boldsymbol{\omega}'_{i-1} \times \mathbf{p}_i^{i-1})) + 2\boldsymbol{\omega}'_i \times \dot{d}_i \mathbf{z}'_i + \ddot{d}_i \mathbf{z}'_i \quad (12)$$

where d_i is the displacement from axis x_{i-1} to x_i along the z_i axis.

Inward Recursion to Calculate Forces and Torques

With the velocities and accelerations found, the inertial force and torque applied to link i at its center of mass can be calculated by applying the Newton-Euler equations:

$$\mathbf{f}'_{C_i} = m_i \dot{\mathbf{v}}'_{C_i} \quad (13)$$

$$\mathbf{n}'_{C_i} = \mathbf{I}_i \dot{\boldsymbol{\omega}}'_i + \boldsymbol{\omega}'_i \times \mathbf{I}_i \boldsymbol{\omega}'_i \quad (14)$$

where m_i is the mass of the link and \mathbf{I}_i denotes the inertia tensor of link i about its center of mass, calculated in a frame identical to the link's coordinate system except the origin. In the equations above, all items are expressed in the local coordinate systems.

The forces and torques acting on the links are calculated through the outward recursion. The joint torques that generate the forces and torques will be calculated through the inward recursion.

Referring to Fig. 1, the inward recursion propagates the joint forces and torques from the end link to the base as

$$\mathbf{f}'_i = \mathbf{f}'_{C_i} + \mathbf{R}^i_{i+1} \mathbf{f}'_{i+1} \quad (15)$$

$$\mathbf{n}'_i = \mathbf{n}'_{C_i} + \mathbf{R}^i_{i+1} \mathbf{n}'_{i+1} + \mathbf{p}'_{C_i} \times \mathbf{f}'_{C_i} + \mathbf{p}'_{i+1} \times \mathbf{R}^i_{i+1} \mathbf{f}'_{i+1} \quad (16)$$

where \mathbf{f}'_i and \mathbf{n}'_i are the force and torque exerted on link i with respect to the i th frame.

The actuating torque on a revolute joint can be calculated by

$$\tau_i = \mathbf{n}'_i{}^T \mathbf{z}'_i \quad (17)$$

For prismatic joints, the driving force is

$$f_i = \mathbf{f}'_i{}^T \mathbf{z}'_i \quad (18)$$

Forward Dynamics Problem

In the forward dynamics problem, the joint accelerations are to be calculated with known joint torques. One method called articulated-body algorithm (ABA) (Featherstone 1983) is applicable to this type of problem with recursive approach. More dynamics modeling methods were reviewed and discussed in (Featherstone and Orin 2000).

Lagrange's Equation

The equations of motion for a manipulator can also be formulated with Lagrangian dynamics method. Two types of Lagrange's equations can be formulated, one with dependent coordinates (1st kind) and the other with independent coordinates

(2nd kind). This section starts with the Lagrange's equation of the 1st kind. Hereafter, the vector \mathbf{q} represents a set of n unknown dependent coordinates, m is the total number of independent kinematically constraint equations, and $f = n - m$ is the number of dynamic degrees of freedom (de Jalón and Bayo 1994). The constraint conditions for the entire system are written in the following general form:

$$\Phi(\mathbf{q}) = 0 \quad (19)$$

Differentiating Eq. 19 with respect to time leads to

$$\frac{d\Phi(\mathbf{q})}{dt} = \frac{\partial\Phi}{\partial\mathbf{q}} \frac{d\mathbf{q}}{dt} = \Phi_{\mathbf{q}}\dot{\mathbf{q}} = 0 \quad (20)$$

where matrix $\Phi_{\mathbf{q}}$ is the Jacobian matrix of the constraint Eq. 19.

Let $L = T - V$ be the Lagrangian of the system, where $T = T(\mathbf{q}, \dot{\mathbf{q}})$ and $V = V(\mathbf{q})$ are the kinetic and potential energies, respectively, and \mathbf{Q}_{ex} is the vector of generalized external forces acting along the dependent coordinates \mathbf{q} . Considering a manipulator as a constrained multibody system, the Lagrange's equation can be derived by generalizing the Hamilton's principle by means of Lagrange multiplier technique (de Jalón and Bayo 1994). Skipping details of derivation, the equations of motion become

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L}{\partial \mathbf{q}} + \Phi_{\mathbf{q}}^T \boldsymbol{\lambda} = \mathbf{Q}_{ex} \quad (21)$$

which is the Lagrange's equation of the 1st kind. In the equation, $\boldsymbol{\lambda}$ is a m -dimensional vector of Lagrange multipliers. Item $\Phi_{\mathbf{q}}^T \boldsymbol{\lambda}$ represents the generalized forces due to reaction forces at joints.

With independent coordinates, the constraint Jacobian vanishes, and the term with Lagrange multipliers is dropped out. The Lagrange's equation of the 2nd kind is thus obtained:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L}{\partial \mathbf{q}} = \mathbf{Q}_{ex} \quad (22)$$

The Lagrange's equation of the 1st kind stands for a system of n equations for n dependent co-ordinates and m unknown Lagrange multipliers. To solve the equations, additional acceleration equations from the kinematic constraints are required, as presented below.

The kinetic energy of a robotic system can be written as follows:

$$T = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} \quad (23)$$

where $\mathbf{M}(\mathbf{q}) = \mathbf{M}$ is the mass matrix of the system. The first and second terms in Eq. 21 can be expressed as below:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{q}}} \right) = \mathbf{M}\ddot{\mathbf{q}} + \dot{\mathbf{M}}\dot{\mathbf{q}}; \quad \frac{\partial L}{\partial \mathbf{q}} = L_{\mathbf{q}} = T_{\mathbf{q}} - V_{\mathbf{q}} \quad (24)$$

For the general case in which the kinetic energy depends on \mathbf{q} , Eq. 21 becomes

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \lambda = \mathbf{Q}_{ex} + L_{\mathbf{q}} - \dot{\mathbf{M}}\dot{\mathbf{q}} \quad (25)$$

On the other hand, differentiating the constraint Eq. 19 twice with respect to time yields

$$\Phi_{\mathbf{q}}\ddot{\mathbf{q}} + \dot{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} = \mathbf{0} \quad (26)$$

Let $\mathbf{c} = -\dot{\Phi}_{\mathbf{q}}\dot{\mathbf{q}}$ and $\mathbf{n} = \mathbf{Q}_{ex} + L_{\mathbf{q}} - \dot{\mathbf{M}}\dot{\mathbf{q}}$. Combining Eqs. 25 and 26 leads to:

$$\begin{bmatrix} \mathbf{M} & \Phi_{\mathbf{q}}^T \\ \Phi_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{n} \\ \mathbf{c} \end{bmatrix} \quad (27)$$

Equation 27 can solve simultaneously the accelerations and the Lagrange multipliers.

Examples

Two examples are included to apply the equations of motion to robotic manipulators. In the first example, a lightweight robotic arm consisting of five revolute joints developed at Aalborg University (AAU), Denmark, is considered. In the second example, the dynamics of a spherical parallel manipulator is presented.

Example I: A 5-dof Lightweight Robotic Arm

The lightweight robotic arm is demonstrated in Fig. 2a. Following the D-H convention, Cartesian coordinate systems are attached to each link of the manipulator, as shown in Fig. 2b. The D-H parameters of the manipulator are listed in Table 1.

Dynamics Modeling

The kinematics of the robotic arm can be found in (Zhou et al. 2011). This chapter includes here only the dynamics formulation.

Jacobian matrix The joint angular velocity can be calculated with the Jacobian matrix

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^{-1} \mathbf{v}_{ef} \quad (28)$$

where $\dot{\boldsymbol{\theta}} = [\dot{\theta}_1, \dot{\theta}_2, \dots, \dot{\theta}_n]^T$ denotes an n -dimensional (n denotes the number of dof) vector of the joint angular velocities, \mathbf{J} is the Jacobian of the robotic arm, and \mathbf{v}_{ef} the velocity of the end-effector.

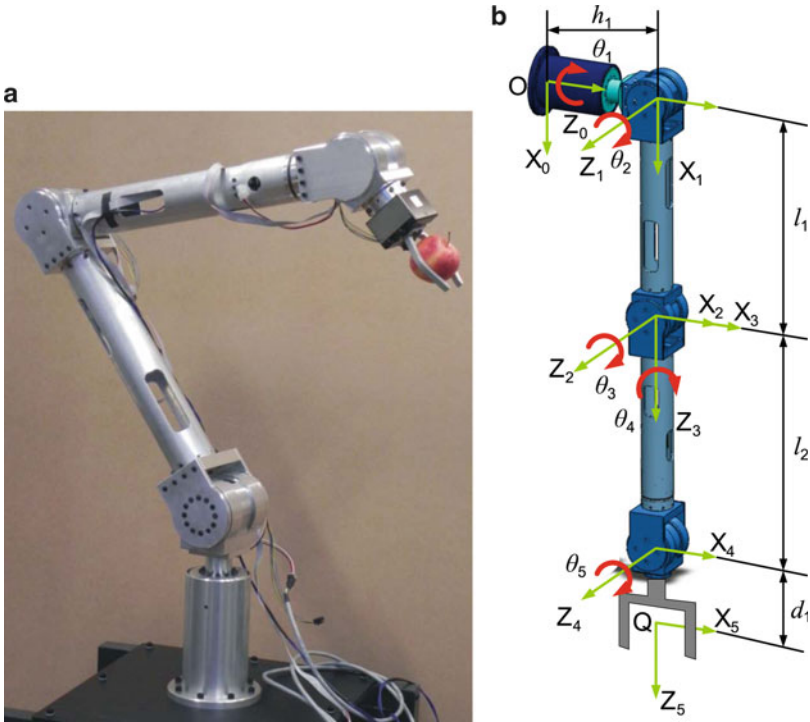


Fig. 2 AAU 5-dof robotic arm: (a) a prototype and (b) coordinate systems

Table 1 D-H parameters of the 5-dof robotic arm

Joint i	α_i	a_i	d_i	θ_i
1	$\pi/2$	0	h_1	θ_1
2	0	l_1	0	θ_2
3	$\pi/2$	0	0	θ_3
4	$-\pi/2$	0	l_2	θ_4
5	$\pi/2$	0	d_1	θ_5

For a revolute joint, the Jacobian matrix can be calculated by (Tsai 1999)

$$\mathbf{J} = [\mathbf{j}_1, \mathbf{j}_2, \dots, \mathbf{j}_n], \mathbf{j}_i = \begin{bmatrix} \mathbf{z}_{i-1} \times \mathbf{p}_{i-1} \\ \mathbf{z}_{i-1} \end{bmatrix} \tag{29}$$

where \mathbf{z}_{i-1} and \mathbf{p}_{i-1} are given by

$$\mathbf{z}_{i-1} = \mathbf{R}_{i-1}^i [0 \ 0 \ 1]^T, \mathbf{p}_{i-1} = \mathbf{R}_{i-1}^i \mathbf{q}_{i-1} + \mathbf{p}_i \tag{30}$$

where $\mathbf{q}_{i-1} = [a_i \cos \theta_i, a_i \sin \theta_i, d_i]^T$. The local coordinates of the end-effector are defined as $\mathbf{p}_n = [0, 0, 0]^T$. When the desired end-effector velocity \mathbf{v}_{ef} is given, the joint angular velocity can be solved by Eq. 28.

The joint angular acceleration $\ddot{\boldsymbol{\theta}}$ can be calculated through time derivative of the velocity $\dot{\boldsymbol{\theta}}$.

$$\ddot{\boldsymbol{\theta}} = \mathbf{J}^{-1} \dot{\mathbf{v}}_{ef} + \mathbf{J}^{-1} \dot{\mathbf{v}}_{ef} \quad (31)$$

Simulation Results

The dynamics of the 5-dof robotic arm was simulated with a program implemented in MATLAB. The trajectory of the end-effector in the global coordinate system is defined as $x_{ef}(t) = 5 + 400(1 - \cos(t))$, $y_{ef}(t) = -990 + 800(1 - \cos(t/2))$, and $z_{ef}(t) = 280 + 250(\cos(t/2) - 1)$, all with unit of *mm*, which leads to

$$\dot{\mathbf{x}}_{ef} = \begin{bmatrix} 400 \sin(t) \\ 400 \sin(t/2) \\ -125 \sin(t/2) \end{bmatrix}$$

The Euler angles for the end-effector are given as $\boldsymbol{\psi} = [0, \cos(t/180), 0]^T$, following ZYZ convention. The rotation implies the end-effector remains horizontal during the prescribed motion. For the prescribed motion, the rotation matrix \mathbf{R} and angular velocity $\boldsymbol{\omega}$ can be readily found. The velocities of the end-effector are $\mathbf{v}_{ef} = [\dot{\mathbf{x}}_{ef}^T, \boldsymbol{\omega}^T]^T$, from which the joint angular velocities and accelerations are calculated.

To implement the Newton-Euler method to solve the inverse dynamics problem, the equations in section “[Recursive Newton-Euler Method](#)” are summarized in a recursive manner:

Outward recursion: $i : 1 \rightarrow 5$

$$\boldsymbol{\omega}'_i = \mathbf{R}_{i-1}^i \boldsymbol{\omega}'_{i-1} + \dot{\boldsymbol{\theta}}_i \mathbf{z}'_i \quad (32a)$$

$$\boldsymbol{\omega}'_i = \mathbf{R}_{i-1}^i \boldsymbol{\omega}'_{i-1} + \ddot{\boldsymbol{\theta}}_i \mathbf{z}'_i + \mathbf{R}_{i-1}^i \boldsymbol{\omega}'_{i-1} \times \dot{\boldsymbol{\theta}}_i \mathbf{z}'_i \quad (32b)$$

$$\dot{\mathbf{v}}'_i = \mathbf{R}_{i-1}^i (\dot{\mathbf{v}}'_{i-1} + \boldsymbol{\omega}'_{i-1} \times \mathbf{p}_i^{i-1} + \boldsymbol{\omega}'_{i-1} \times (\boldsymbol{\omega}'_{i-1} \times \mathbf{p}_i^{i-1})) \quad (32c)$$

$$\dot{\mathbf{v}}'_{C_i} = \dot{\mathbf{v}}'_{i-1} + \boldsymbol{\omega}'_{i-1} \times \mathbf{p}'_{C_i} + \boldsymbol{\omega}'_{i-1} \times (\boldsymbol{\omega}'_{i-1} \times \mathbf{p}'_{C_i}) \quad (32d)$$

$$\mathbf{f}'_{C_i} = m_i \dot{\mathbf{v}}'_{C_i} \quad (32e)$$

$$\mathbf{n}'_{C_i} = \mathbf{I}_i \dot{\boldsymbol{\omega}}'_i + \boldsymbol{\omega}'_i \times \mathbf{I}_i \boldsymbol{\omega}'_i \quad (32f)$$

Inward recursion: $i : 5 \rightarrow 1$

$$\mathbf{f}'_i = \mathbf{f}'_{C_i} + \mathbf{R}_{i+1}^i \mathbf{f}'_{i+1} \quad (32g)$$

$$\mathbf{n}'_i = \mathbf{n}'_{C_i} + \mathbf{R}^i_{i+1} \mathbf{n}'_{i+1} + \mathbf{p}'_{C_i} \times \mathbf{f}'_{C_i} + \mathbf{p}'_{i+1} \times \mathbf{R}^i_{i+1} \mathbf{f}'_{i+1} \tag{32h}$$

$$\tau_i = \mathbf{n}'_i{}^T \mathbf{z}'_i \tag{32i}$$

The joint angular velocities and accelerations are solved through Eq. 28, with results displayed in Fig. 3. Let the payload at the end-effector of the robot be 5 kg, which implies $\mathbf{f}_6 = 5\mathbf{g}N$, $\mathbf{n}_6 = \mathbf{0}$, where \mathbf{g} is the vector of gravity acceleration. Through the modeling of inverse dynamics of Eqs. 32g–32i, the joint torques are solved, as shown in Fig. 4.

Example II: A Spherical Parallel Manipulator

In this example, the Lagrange’s equation is illustrated with a spherical parallel manipulator (SPM) (Gosselin and Angeles 1989). The SPMs are of closed-kinematic chain. The Lagrange’s equation of the 1st kind is adopted for the dynamics modeling of this manipulator.

A general SPM is shown in Fig. 5a with the parameterized *i*th leg in Fig. 5b. The *i*th limb consists of three revolute joints, whose axes are parallel to the unit vectors \mathbf{u}_i , \mathbf{v}_i , and \mathbf{w}_i . All three limbs have identical architectures, defined by angles α_1 and α_2 . Moreover, β and γ define the geometry of two triangular pyramids on the base and the mobile platforms, respectively. The origin *O* of the base coordinate system *xyz* is located at point *O*. The *z* axis is normal to the bottom surface of the base pyramid and points upward, while the *y* axis is located in the plane made by the *z* axis and \mathbf{u}_1 .

Kinematic Modeling

Under the prescribed coordinate system, unit vector \mathbf{u}_i is derived as

$$\mathbf{u}_i = [-\sin \eta_i \sin \gamma \quad \cos \eta_i \sin \gamma \quad -\cos \gamma]^T \tag{33}$$

where $\eta_i = 2(i - 1)\pi/3$, $i = 1, 2, 3$.

The unit vector \mathbf{v}_i of the axis of the intermediate revolute joint in the *i*th leg is expressed as:

$$\mathbf{v}_i = \begin{bmatrix} -s\eta_i s\gamma \alpha_1 + (c\eta_i s\theta_i - s\eta_i c\gamma c\theta_i) s\alpha_1 \\ c\eta_i s\gamma \alpha_1 + (s\eta_i s\theta_i + c\eta_i c\gamma c\theta_i) s\alpha_1 \\ -c\gamma \alpha_1 + s\gamma c\theta_i s\alpha_1 \end{bmatrix} \tag{34}$$

where $s(\cdot) = \sin(\cdot)$ and $c(\cdot) = \cos(\cdot)$.

The unit vector \mathbf{w}_i of the top revolute joint in the *i*th leg is a function of the orientation of the mobile platform (MP), namely,

$$\mathbf{w}_i = [w_{ix} \quad w_{iy} \quad w_{iz}]^T = \mathbf{Q}\mathbf{w}_i^* \tag{35}$$

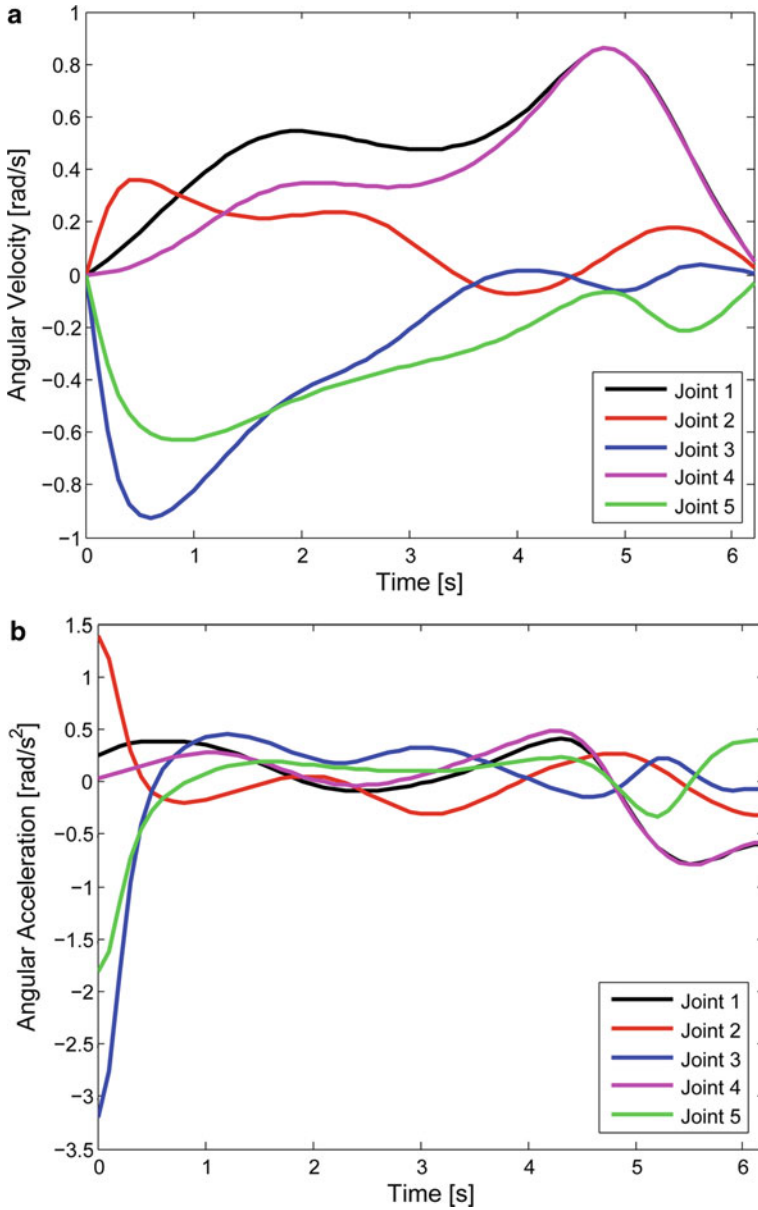


Fig. 3 Angular velocities and accelerations of the robotic arm joints

where \mathbf{Q} is the rotation matrix that carries the MP from its reference orientation to the current one and \mathbf{w}_i^* is the unit vector of the axis of the top revolute joint in the i th leg when the mobile platform is in its reference orientation, which is given as

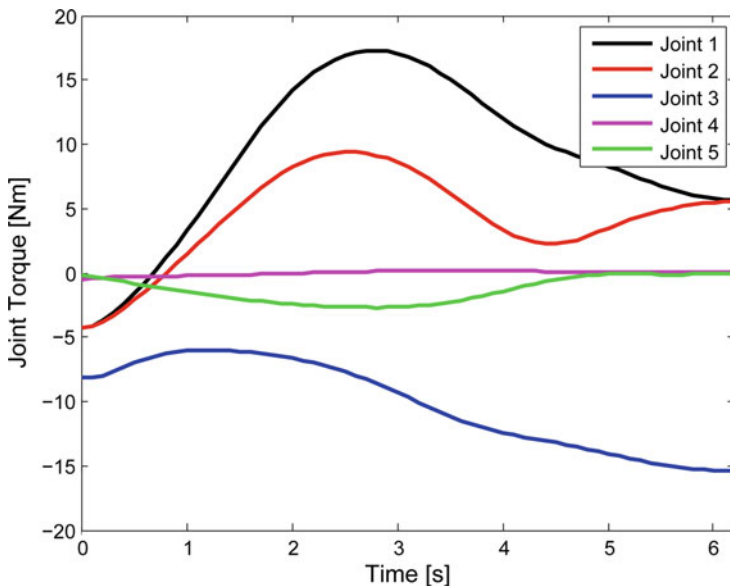


Fig. 4 Joint torques of the robotic arm

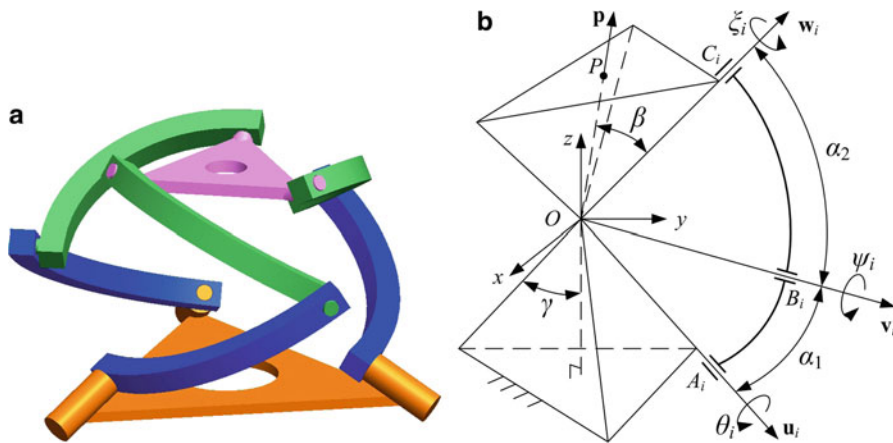


Fig. 5 A general SPM and the coordinate system, where only one leg is shown for clarity

$$\mathbf{w}_i^* = [-\sin \eta_i \sin \beta \quad \cos \eta_i \sin \beta \quad \cos \beta]^T \tag{36}$$

Kinematic Jacobian Matrix

Let $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$ denote the angular velocity of the mobile platform; the velocity equation via the i th leg can be stated as

$$\boldsymbol{\omega} = \dot{\theta}_i \mathbf{u}_i + \dot{\psi}_i \mathbf{v}_i + \dot{\xi}_i \mathbf{w}_i \quad (37)$$

To eliminate $\dot{\psi}_i$ and $\dot{\xi}_i$, dot-multiplying Eq. 37 on both sides with $\mathbf{v}_i \times \mathbf{w}_i$ yields

$$(\mathbf{v}_i \times \mathbf{w}_i) \cdot \boldsymbol{\omega} = (\mathbf{u}_i \times \mathbf{v}_i) \cdot \mathbf{w}_i \dot{\theta}_i \quad (38)$$

from which the SPM velocity equation is obtained as

$$\mathbf{A}\boldsymbol{\omega} = \mathbf{B}\dot{\boldsymbol{\theta}} \quad (39)$$

with

$$\mathbf{A} = [\mathbf{v}_1 \times \mathbf{w}_1 \quad \mathbf{v}_2 \times \mathbf{w}_2 \quad \mathbf{v}_3 \times \mathbf{w}_3]^T \quad (40a)$$

$$\mathbf{B} = \text{diag}[(\mathbf{u}_1 \times \mathbf{v}_1) \cdot \mathbf{w}_1 \quad (\mathbf{u}_2 \times \mathbf{v}_2) \cdot \mathbf{w}_2 \quad (\mathbf{u}_3 \times \mathbf{v}_3) \cdot \mathbf{w}_3] \quad (40b)$$

where $\dot{\boldsymbol{\theta}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]^T$. Matrices \mathbf{A} and \mathbf{B} are the forward and inverse Jacobian matrices of the manipulator, respectively. The kinematic Jacobian matrix \mathbf{J} of the manipulator can be expressed as follows as long as matrix \mathbf{A} is not singular:

$$\mathbf{J} = \mathbf{B}^{-1}\mathbf{A} = [\mathbf{j}_1 \quad \mathbf{j}_2 \quad \mathbf{j}_3]^T; \mathbf{j}_i = \frac{\mathbf{v}_i \times \mathbf{w}_i}{\mathbf{u}_i \times \mathbf{v}_i \cdot \mathbf{w}_i} \quad (41)$$

Equation 38 is thus rewritten for the single joint velocity as

$$\dot{\theta}_i = \mathbf{j}_i^T \boldsymbol{\omega} \quad (42)$$

Velocity and Acceleration Analysis

The motions of the link and mobile platform are shown in Fig. 6. The angle rates $\dot{\boldsymbol{\phi}} = [\dot{\phi}, \dot{\theta}, \dot{\sigma}]^T$ and the angular velocity $\boldsymbol{\omega}$ are linearly dependent, namely, $\boldsymbol{\omega} = \boldsymbol{\Psi}\dot{\boldsymbol{\phi}}$. Differentiating the equation with respect to time yields

$$\dot{\boldsymbol{\omega}} = \boldsymbol{\Psi}\ddot{\boldsymbol{\phi}} + \dot{\boldsymbol{\Psi}}\dot{\boldsymbol{\phi}} \quad (43)$$

Matrix $\boldsymbol{\Psi}$ is dependent on the rotations. For example, a rotation with the Euler convention $ZY Z$ of $\mathbf{Q} = \mathbf{R}_z(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\sigma - \phi)$, matrix $\boldsymbol{\Psi}$ is

$$\boldsymbol{\Psi} = \begin{bmatrix} -s\theta c\phi & -s\phi & s\theta c\phi \\ -s\theta s\phi & c\phi & s\theta s\phi \\ 1 - c\theta & 0 & c\phi \end{bmatrix} \quad (44)$$

The velocity $\dot{\psi}$ of the intermediate joint of i th leg is found by making use of Eq. 37 to eliminate $\dot{\theta}_i$ and $\dot{\xi}_i$. Dot-multiplying Eq. 37 on both sides with $\mathbf{u}_i \times \mathbf{w}_i$ leads to

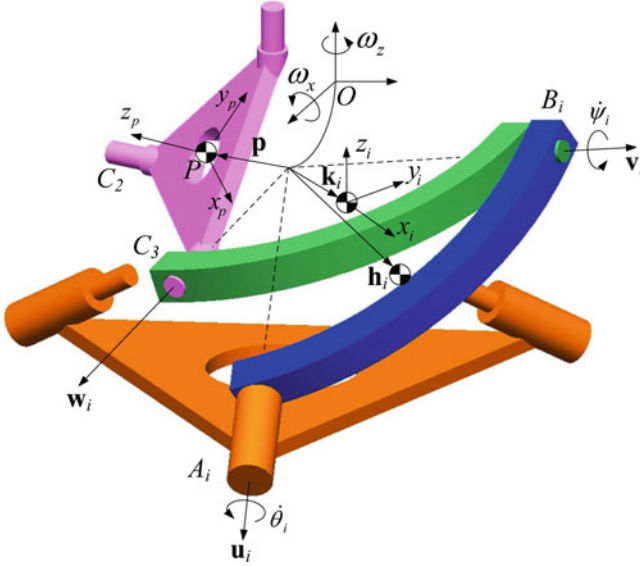


Fig. 6 The movement of the mobile platform and links

$$(\mathbf{u}_i \times \mathbf{w}_i) \cdot \boldsymbol{\omega} = \dot{\psi}_i (\mathbf{u}_i \times \mathbf{w}_i) \cdot \mathbf{v}_i \text{ or } \dot{\psi}_i = \frac{(\mathbf{u}_i \times \mathbf{w}_i)^T}{(\mathbf{u}_i \times \mathbf{w}_i) \cdot \mathbf{v}_i} \boldsymbol{\omega} \equiv \mathbf{J}_{\psi_i}^T \boldsymbol{\omega} \quad (45)$$

The angular velocity of the distal link in the i th leg in the reference frame xyz is found as $\boldsymbol{\rho}_i = \dot{\theta}_i \mathbf{u}_i + \dot{\psi}_i \mathbf{v}_i$. Let $\boldsymbol{\rho}_{li}$ denote the corresponding angular velocity in the local frame $x_i y_i z_i$, which means

$$\boldsymbol{\rho}_i = [\mathbf{e}_{ix} \quad \mathbf{e}_{iy} \quad \mathbf{e}_{iz}] \boldsymbol{\rho}_{li} \text{ or } \boldsymbol{\rho}_i = \mathbf{E}_i \boldsymbol{\rho}_{li} \quad (46)$$

with

$$\mathbf{e}_{ix} = \frac{\mathbf{v}_i + \mathbf{w}_i}{\|\mathbf{v}_i + \mathbf{w}_i\|}, \mathbf{e}_{iy} = \frac{\mathbf{v}_i + \mathbf{w}_i}{\|\mathbf{v}_i + \mathbf{w}_i\|}, \mathbf{e}_{iz} = \frac{\mathbf{v}_i + \mathbf{w}_i}{\|\mathbf{v}_i + \mathbf{w}_i\|} \quad (47)$$

which yields

$$\boldsymbol{\rho}_{li} = \mathbf{E}_i^T \boldsymbol{\rho}_i \quad (48)$$

Dynamic Modeling

The Lagrange’s equation for the SPM is expressed as

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L}{\partial \mathbf{q}} + \mathbf{C}_q^T \boldsymbol{\lambda} = \mathbf{Q}_{ex} \quad (49)$$

where $\mathbf{q} = [\theta_1, \theta_2, \theta_3, \phi, \theta, \sigma]^T$, $\mathbf{Q}_{ex} = [\boldsymbol{\tau}^T, \mathbf{0}]^T \in \mathbb{R}^6$ is the vector of external forces, and vector $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3]^T$ characterizes the actuator torques. Moreover, $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]^T$ is a vector of Lagrange multipliers. Matrix \mathbf{C}_q is the system's constraint Jacobian, which can be found from the velocity Eq. 39, namely,

$$\mathbf{B}\dot{\boldsymbol{\theta}} - \mathbf{A}\boldsymbol{\omega} = [\mathbf{B} \quad -\mathbf{A}\boldsymbol{\Psi}] \begin{bmatrix} \dot{\boldsymbol{\theta}}^T \\ \dot{\boldsymbol{\phi}}^T \end{bmatrix} = \mathbf{0} \quad (50)$$

that is,

$$\mathbf{C}_q \dot{\mathbf{q}} = \mathbf{0} \quad (51)$$

with $\mathbf{C}_q = [\mathbf{B} - \mathbf{A}\boldsymbol{\Psi}]$.

Lagrangian of the Mobile Platform

The local frame $x_p y_p z_p$ of the MP is established with the origin located at point P , i.e., the center of mass of the MP of position vector \mathbf{p} . Henceforth, the Lagrangian of the mobile platform is obtained as

$$L_p = T_p - V_p = \frac{1}{2} \boldsymbol{\omega}'^T \mathbf{I}_p \boldsymbol{\omega}' - m_p \mathbf{g}^T \mathbf{p} \quad (52)$$

where \mathbf{I}_p denotes the local inertia tensor of the mobile platform and $\boldsymbol{\omega}'$ its local angular velocity. Moreover, $\mathbf{g} = [0, 0, 9.81]^T$.

Lagrangian of a Single Leg

The Lagrangian of the i th leg is derived as

$$L_i = T_i - V_i = \frac{1}{2} I_{l1} \dot{\theta}_i^2 + \frac{1}{2} \boldsymbol{\rho}_{li}^T \mathbf{I}_{l2} \boldsymbol{\rho}_{li} - m_{l1} \mathbf{g}^T \mathbf{h}_i - m_{l2} \mathbf{g}^T \mathbf{k}_i \quad i = 1, 2, 3 \quad (53)$$

where I_{l1} is the proximal link's mass moment of inertia about \mathbf{u}_i and \mathbf{I}_{l2} is the distal link's local inertia tensor about point O . Moreover, \mathbf{h}_i and \mathbf{k}_i indicate the position vectors for the centers of the mass of the proximal and distal links, respectively.

Substituting the Lagrangian L_p and L_i , $i = 1, 2, 3$, into Eq. 49, the inertia matrix can be derived as

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} \mathbf{M}_1 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{M}_2 \end{bmatrix} \quad (54)$$

with

$$\mathbf{M}_1 = \text{diag}[I_{l1} \quad I_{l1} \quad I_{l1}] \quad (55a)$$

$$\mathbf{M}_2 = \boldsymbol{\Psi}^T \left[\mathbf{Q} \mathbf{I}_p \mathbf{Q}^T + \sum_{i=1}^3 \left(\mathbf{j}_i \mathbf{u}_i^T + \mathbf{j}_{\psi_i} \mathbf{v}_i^T \right) \mathbf{E} \mathbf{I}_{l2} \mathbf{E}^T \left(\mathbf{u}_i \mathbf{j}_i^T + \mathbf{v}_i \mathbf{j}_{\psi_i}^T \right) \right] \boldsymbol{\Psi} \quad (55b)$$

Table 2 Parameters of the SPM and the initial simulation condition

α_1 [deg]	α_2 [deg]	β [deg]	γ [deg]	$[\phi, \theta, \sigma]$ [rad]	$[\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]$ [rad/s]	\mathbf{m} [Nm]
60	80	75	0	$[0, \pi/6, 0]$	$[-8, -7, -9]$	$[0.1, 0.1, 0.1]$

Table 3 Mass and inertia properties of the SPM

Mobile platform		Proximal link		Distal link	
m_p [kg]	\mathbf{I}_p [10^{-4} kg m ²]	m_{l1} [kg]	I_{l1} [10^{-3} kg m ²]	m_{l2} [kg]	\mathbf{I}_{l2} [10^{-4} kg m ²]
0.332	[3.855 3.855 7.688]	0.123	1.169	0.114	[1.852 0.089 1.921]

and the other terms also can be derived. Moreover, differentiating Eq. 51 with respect to time yields

$$\mathbf{C}_q \ddot{\mathbf{q}} = -\dot{\mathbf{C}}_q \dot{\mathbf{q}} \quad (56)$$

with

$$\mathbf{C}_q \ddot{\mathbf{q}} = \mathbf{C}_q \begin{bmatrix} \ddot{\theta}^T & \ddot{\phi}^T \end{bmatrix}^T; \quad \dot{\mathbf{C}}_q \dot{\mathbf{q}} = [\dot{\mathbf{B}} - \dot{\mathbf{A}}\Psi - \mathbf{A}\dot{\Psi}] \dot{\mathbf{q}} \quad (57)$$

where

$$\dot{\mathbf{A}} = [\mathbf{j}_{A1} \quad \mathbf{j}_{A2} \quad \mathbf{j}_{A3}]^T \quad (58a)$$

$$\dot{\mathbf{B}} = \text{diag}[j_{B1} \quad j_{B2} \quad j_{B3}] \quad (58b)$$

and

$$\mathbf{j}_{Ai} = \dot{\theta}_i (\mathbf{u}_i \times \mathbf{v}_i) \times \mathbf{w}_i + \mathbf{v}_i \times (\boldsymbol{\omega} \times \mathbf{w}_i) \quad (59a)$$

$$j_{Bi} = \dot{\theta}_i [\mathbf{u}_i \times (\mathbf{u}_i \times \mathbf{v}_i)] \cdot \mathbf{w}_i + (\mathbf{u}_i \times \mathbf{v}_i) \cdot (\boldsymbol{\omega} \times \mathbf{w}_i) \quad (59b)$$

Combining Eqs. 49 and 56 leads to

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & \mathbf{C}_q^T \\ \mathbf{C}_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau} - \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}} + T_q - V_q \\ -\dot{\mathbf{C}}_q \dot{\mathbf{q}} \end{bmatrix} \quad (60)$$

With an external moment vector \mathbf{m} , the actuator torques are expressed as

$$\boldsymbol{\tau}_a = \boldsymbol{\tau} - \mathbf{J}^{-T} \mathbf{m} \quad (61)$$

Simulation Results

The previously developed dynamic model is applied to a co-axial SPM (Bai 2010), whose properties are given in Tables 2 and 3, where inertia tensors \mathbf{I}_p and \mathbf{I}_{l2} contain

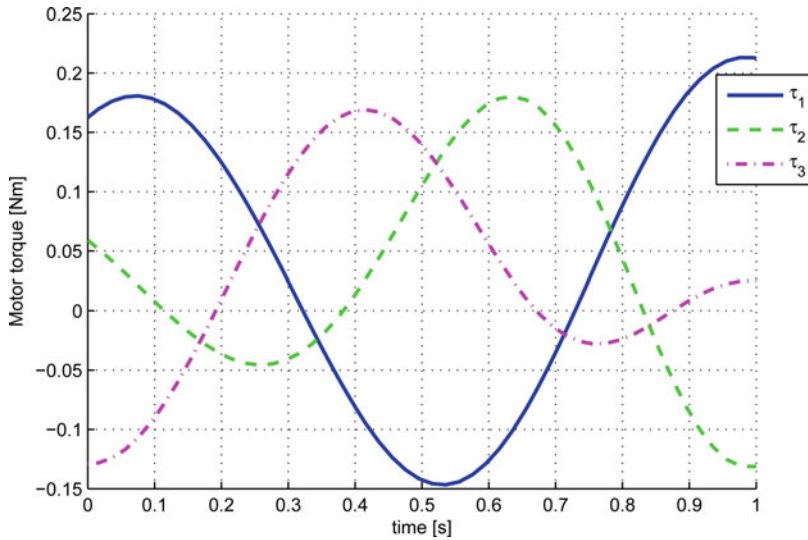


Fig. 7 The simulation of inverse dynamics: actuator torques

only diagonal entries. Simulations were conducted for motion of constant angular accelerations at actuating joints, their initial conditions being specified in Table 2. The results of the actuating torques are shown in Fig. 7. The results agree well with the simulation results obtained from CAE software, namely, the MSC Adams.

Summary

This book chapter introduces the fundamental of manipulator dynamics, with focus on the *recursive Newton-Euler approach* and the *Lagrange equations*, which are approaches mostly common used in robot dynamics analysis and control. Both inverse and forward dynamic problems are discussed. Applications to manipulators are demonstrated.

In solving the dynamics equation, numerics problems may arise. One is computational efficiency. The recursive approach presented in this chapter is quite efficient, while the Lagrange's equations, as a closed-form equation, implies high computational cost in finding solutions, which uses numerical methods, for example, Newton-Raphson method. Other problems include the singular Jacobian and the instability of the solution, etc., among others. The Jacobian matrix can become singular when a singular configuration is reached or redundant constraints are included, which may lead to a crash of simulation or a large error. Instability takes place in numerical integration of the equation of motion, when the roundoff errors increase with time, which can lead to the constraint equations not satisfied. A method to improve the stability is the Baumgarte stabilization method, which is introduced in (de Jalón and Bayo 1994).

The approaches presented in this chapter are closely relevant to multibody dynamics, which was extensively studied in the last century. With the advances of robotics in the twenty-first century, manipulator dynamics has significantly been expanded, not limited to multibody dynamics, but get involved with other subsets of the dynamics. A new development among others is the continuum manipulators, where continuum mechanics is the major theory applicable (Walker 2013). The new development of manipulator dynamics, which is not discussed in this chapter, presents new opportunity and challenges for robotics.

References

- Angeles J (1997) Fundamentals of robotic mechanical systems: theory, methods, and algorithms. Springer, New York
- Bai S (2010) Optimum design of spherical parallel manipulators for a prescribed workspace. *Mech Mach Theory* 45(2):200–211
- de Jalón JG, Bayo E (1994) Kinematic and dynamic simulation of multibody systems: the real-time challenge. Springer, New York
- Denavit J, Hartenberg RS (1955) A kinematic notation for lower pair mechanisms based on matrices. *ASME J Appl Mech* 77:215–221
- Dwivedy SK, Eberhard P (2006) Dynamic analysis of flexible manipulators, a literature review. *Mech Mach Theory* 41(7):749–777
- Featherstone R (1983) The calculation of robot dynamics using articulated-body inertias. *Int J Robot Res* 2(1):13–30
- Featherstone R, Orin D (2000) Robot dynamics: equations and algorithms. In: Proceedings of the 2000 I.E. International Conference on Robotics & Automation, San Francisco, pp 826–834, Apr 2000
- Gosselin CM, Angeles J (1989) The optimum kinematic design of a spherical three-degree-of-freedom parallel manipulator. *ASME J Mech Des* 111(2):202–207
- Low KH (1987) A systematic formulation of dynamic equations for robot manipulators with elastic links. *J Robot Syst* 4(3):435–456
- Luh JYS, Walker MW, Paul RPC (1980) On-line computational scheme for mechanical manipulators. *J Dyn Sys Meas Control* 102(2):69–76
- Murray RM, Li Z, Sastry SS (1994) A mathematical introduction to robotic manipulation. CRC Press, Boca Raton
- Shabana AA (2013) Dynamics of multibody systems. Cambridge University Press, New York, USA
- Tsai LW (1999) Robot analysis: the mechanics of serial and parallel manipulators. Wiley, New York
- Walker ID (2013) Continuous backbone continuum robot manipulators. ISRN Robot Article ID: 726506, 19pp
- Zhou L, Bai S, Hansen MR (2011) Design optimization on the drive train of a light-weight robotic arm. *Mechatronics* 21(3):560–569