

Dynamics analysis of a 3-RRP spherical parallel manipulator using the natural orthogonal complement

Alireza Akbarzadeh · Javad Enferadi · Mahdi Sharifnia

Received: 10 September 2011 / Accepted: 19 June 2012 / Published online: 21 July 2012
© Springer Science+Business Media B.V. 2012

Abstract In the present research, application of the Natural Orthogonal Complement (NOC) for the dynamic analysis of a spherical parallel manipulator, referred to as SST, is presented. Both inverse and direct dynamics are considered. The NOC and the SST fully parallel robot are explained. To drive the NOC for the SST manipulator, constraints between joint variables are written using the transformation matrices obtained from three different branches of the robot. The Newton–Euler formulation is used to model the dynamics of each individual body, including moving platform and legs of the manipulator. D’Alembert’s principle is applied and Newton–Euler dynamical equations free from non-working generalized constraint forces are obtained. Finally two examples, one for direct and one for inverse dynamics are presented. The correctness and accuracy of the obtained solution are verified by comparing with the solution of the virtual work method as well as commercial multi-body dynamics software.

Keywords Spherical parallel manipulator · Natural orthogonal complement · Inverse dynamics · Direct dynamics

1 Introduction

Dynamics modeling of mechanical systems has been applied to different engineering branches and has attracted attention of many researchers. Many solutions for solving the

A. Akbarzadeh · M. Sharifnia
Center of Excellence on Soft Computing and Intelligent Information Processing (SCIIP),
M.E. Department, Ferdowsi University of Mashhad, Mashhad, Iran

A. Akbarzadeh
e-mail: ali_akbarzadeh_t@yahoo.com

M. Sharifnia
e-mail: sharifnia.mehdi@gmail.com

J. Enferadi (✉)
M.E. Department, Islamic Azad University, Mashhad Branch, Mashhad, Iran
e-mail: Javadenferadi@gmail.com

dynamical equations of motion have been proposed. When multi-body systems are considered, the quality of system modeling and solution of its related equations are important. Parallel and serial robots are some of the multi-body systems which require knowledge of their dynamic models. This information, for example, can be used for design and selection of the robot components, control and simulation.

Generally, solving kinematics problem of a multi-body system is necessary for the dynamics analysis. In [1–11], the common dynamics methods such as Newton–Euler method, Lagrangian formulation and the principle of virtual work are used. Many researchers have considered kinematics and dynamics analysis of parallel robots. In [12], the direct kinematics of parallel robots with revolute joints is presented. The direct kinematics of parallel robots is usually more difficult than its serial counterpart. For parallel robots, obtaining solution to the direct kinematics is commonly challenging and most of the times results in a high degree polynomial with multiple answers [13–15]. Numerical methods are commonly employed to obtain the multiple solutions. Still additional methods are required to identify the one correct solution among the multiple solutions. These make obtaining the direct kinematics problem challenging [16]. Direct dynamics problem requires solution of the direct kinematics. This requirement further introduces challenges for direct dynamic applications of parallel robots. Therefore, dynamical solution methods that are independent of directly obtaining kinematics solution are certainly more desirable. One method with such characteristic is the natural orthogonal complement (NOC) method. The NOC was first introduced by Angeles and Lee [17]. The NOC dynamic modeling was found to have certain advantages in [18] and others. This method defines natural orthogonal complement for Cartesian velocity-constraint matrix. The velocity constraint matrix may also be defined in joint space in which case the form of natural orthogonal complement will be different. The NOC generally leads to elimination of non-working constraint wrenches. Furthermore, using the NOC, the number of driven dynamic equations becomes minimum and separated. The NOC uses the Newton–Euler dynamic equations and incorporates the kinematics constraint equations. This eliminates the need to directly solve the kinematics.

In [1], kinematics and dynamics of a parallel robot with a 3UPS-PU (where U, P and S, represent universal, prismatic and spherical joints, respectively) architecture is investigated. To obtain the dynamical equations of motion for this robot, the Newton–Euler method and the NOC approach are used. In [2], a formulation of inverse dynamics for Stewart platform using Newton–Euler method is presented. Gravity effects and viscous friction in joints are considered. Additionally, it is shown that to obtain actuated forces a proper elimination method can lead to a fast and economic algorithm which is useful for real time control.

The decoupled natural orthogonal complement (DeNOC) method was first introduced in [5]. Unlike the NOC, the DeNOC method allows one to write expressions of the elements of the matrices and vectors associated with the dynamic equations of motion in analytical and recursive form. In this reference, DeNOC method is used to obtain the dynamics analysis of a hexaglide parallel robot. The concept of DeNOC is applied to a 3-RRR planar parallel manipulator to develop dynamics formulation that is both recursive and modular. An algorithmic approach to the development of both forward and inverse dynamics is demonstrated [6]. Recently, application of DeNOC was extended to obtain a recursive forward dynamics algorithm for serial robots with flexible links. The algorithm is applied to a two flexible link arm and is shown to be computationally more efficient and numerically more stable than other algorithms present in the literature [19]. Additional studies on DeNOC for both serial and parallel mechanism as well as other dynamic systems may be found in [5, 20–24]. In [4], the dynamics of non-holonomic mechanical system is derived using the NOC.

References [7–9] used the principle of virtual work and introduced a new recursive matrix method which is useful for dynamic modeling of parallel robots. Recursive matrix method is used to obtain the dynamics analysis of the Agile Wrist [25]. Additionally, actuated powers in two states, full and simplified dynamic models are compared.

In [14], a new spherical star-triangle parallel robot, called SST, is introduced. In [10] and [11], inverse and direct dynamics of this robot is presented using virtual work method. However, the virtual work method requires solution of the direct kinematics problem for obtaining the direct dynamics. Therefore, an algorithm for selecting the admissible solution among eight possible solutions for the SST robot is also presented [11]. However, time to obtain direct kinematics solution can negatively affect the performance of the system. With the NOC, the system kinematics is integrated with its dynamics as a set of differential equations. This potentially can aid in lowering the computation time.

In this paper, first geometry of the SST parallel manipulator and the NOC are briefly described. Next, the constraint equations and other necessary equations for dynamics analysis of the SST parallel robot using the NOC are formulated. Finally, two examples, direct and inverse dynamics, are presented. In the first example, a robot trajectory is given and inverse dynamics using the NOC is solved to obtain required motor torques. Results are next verified with inverse dynamics using the virtual work method [10]. In the second example, motor torques, obtained from the inverse dynamics of the first example, are used as input for the direct dynamics solution using the NOC. The output of the direct dynamics is compared with the input of the inverse dynamics.

2 Introducing the SST parallel robot

The SST parallel manipulator [14] consists of a fixed virtual spherical triangular base, \mathbf{P} , and a moving platform which is shaped like a spherical star, \mathbf{S} . The fixed base and the moving platform are connected via three legs. Each of the three moving legs is made of RRP (revolute-revolute-spherical slider [26]) joints. The isotropic model of this manipulator is shown in Fig. 1. To develop the mathematical model of the manipulator, first a sphere with center at O and a fixed spherical triangle, $P_1P_2P_3$, on its surface is considered. The unit vector \mathbf{v}_k is defined along OP_k . The revolution axis of each actuated joint is along unit vector \mathbf{e}_k ($k = 1, 2, 3$). This unit vector is perpendicular to plane OP_kP_{k+1} . Rotation about axis of the unit vector \mathbf{e}_k is defined by angle γ_k . See Fig. 2.

The moving spherical star (MSS), \mathbf{S} , is next considered. The MSS is made of three arcs which are located on surface of a second sphere. The three arcs of MSS intersect at point E . The angle between these arcs (α_1, α_2 and α_3) can be manually selected by the robot designer to obtain the desired performance. Position of point E defines end-effector position. Direction OE can be defined by unit vector \mathbf{s} . See Figs. 1 and 2. The arcs of the moveable star platform ER_k intersect the line which is along the actuator links at the point R_k . Angular position of the actuators are defined by the unit vector \mathbf{e}_{k+3} ($k = 1, 2, 3$). Direction of this unit vector is defined along OR_k . Additionally, there exist two more joints. The second joint is also a revolute joint with its axis along \mathbf{e}_{k+3} . The third joint is referred to as spherical slider joint with its axis along \mathbf{e}_{k+6} . The axes of all three joints pass through center of the sphere, point O . See Fig. 2. Rotation about axis of the unit vector \mathbf{e}_{k+3} is defined by angle μ_k .

Finally, the motion of the spherical slider joint can also be viewed as a revolute joint with an axis that passes through the origin of the sphere. This axis is defined by a unit

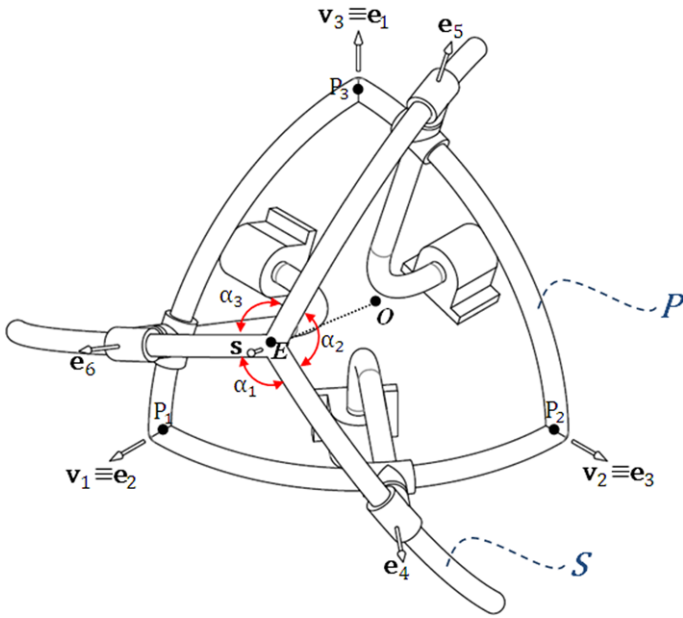


Fig. 1 Isotropic model of the SST manipulator with motors

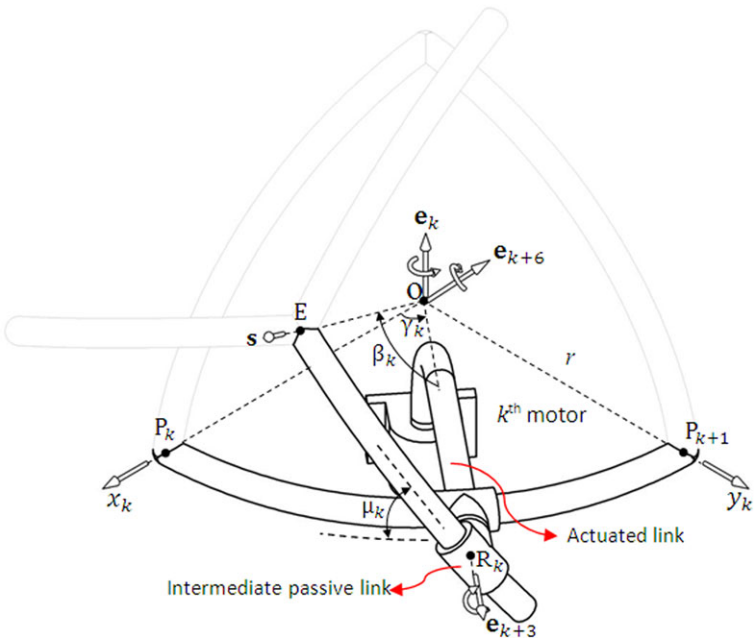


Fig. 2 Parameters description of k th leg

vector, \mathbf{e}_{k+6} ($k = 1, 2, 3$). See Fig. 2. This unit vector is perpendicular to the plane OER_k and passes through origin. Therefore

$$\mathbf{e}_{k+6} = \frac{\mathbf{s} \times \mathbf{e}_{k+3}}{\|\mathbf{s} \times \mathbf{e}_{k+3}\|} \quad \text{for } k = 1, 2, 3 \tag{1}$$

Finally, rotation about axis of the unit vector \mathbf{e}_{k+6} is defined by angle β_k .

3 The Natural Orthogonal Complement (NOC)

Dynamic models developed using the NOC offer many advantages such as being structurally algorithmic, computationally efficient and numerically robust. The method leads to the elimination of the non-working kinematic-constraint wrenches and also to the derivation of the minimum number of equations. Additionally, the dynamics model is in the form of the Euler–Lagrange equations without involving constraint forces, moments or Lagrange multipliers. The resulting equations of motion for the manipulator are in terms of the actuated joint coordinates [24].

Any formulation of equations of motion requires characterization of the role of the physical restrictions that are imposed on a system’s movement. These restrictions lead to constraint equations between joint variables (generalized coordinates), which include both passive and active joints. In dynamics analysis of parallel manipulator using the NOC, obtaining the constraint equation is an important stage. If we describe the joint variables with q_i ($i = 1, 2, \dots, m$), then we can write an independent constraint equations in general form as

$$\mathbf{f}(\mathbf{q}) = \mathbf{0} \tag{2}$$

If the degrees of freedom of the robot number n , then the number of dimensions of the constraint vector equation $\mathbf{f}(\mathbf{q})$ will be equal to $(m - n)$. The time derivative of the constraint equations can be written as

$$\Phi \dot{\mathbf{q}} = \mathbf{0}_{(m-n) \times 1} \tag{3}$$

in which Φ is $(m - n) \times m$ matrix and is called Jacobian matrix of both active and passive joints. Additionally, $\dot{\mathbf{q}}$ is the time derivative of the joint variables and is called joint-velocity vector. Therefore, we can write

$$\Phi = \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} \tag{4}$$

We can separate Φ and \mathbf{q} with respect to actuated and unactuated parts as

$$\Phi = \left[\begin{array}{cc} [\Phi^a]_{(m-n) \times n} & [\Phi^u]_{(m-n) \times (m-n)} \end{array} \right] = \left[\begin{array}{cc} \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}^a} & \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}^u} \end{array} \right]_{(m-n) \times m}, \tag{5}$$

$$\dot{\mathbf{q}} = \left[\begin{array}{c} [\dot{\mathbf{q}}^a]_{n \times 1} \\ [\dot{\mathbf{q}}^u]_{(m-n) \times 1} \end{array} \right]_{m \times 1} \tag{6}$$

where the trailing superscript “a” and “u” designate “actuated joints” and “unactuated joints”, respectively. Dimensions of the matrices Φ^a and Φ^u are equal to $(m - n) \times n$ and $(m - n) \times (m - n)$, respectively. Additionally, the $\dot{\mathbf{q}}^a$ and $\dot{\mathbf{q}}^u$ are vectors of the actuated and unactuated joint rates, respectively. By placing Eqs. (5) and (6) into Eq. (3), we will have

$$\dot{\mathbf{q}}^u = -(\Phi^u)^{-1} \Phi^a \dot{\mathbf{q}}^a \tag{7}$$

The above equation can be rewritten as

$$\Phi^u \dot{\mathbf{q}}^u = -\Phi^a \dot{\mathbf{q}}^a \tag{7a}$$

By taking time derivative of the above equation, we can write

$$\dot{\Phi}^u \dot{q}^u + \Phi^u \ddot{q}^u = -\dot{\Phi}^a \dot{q}^a - \Phi^a \ddot{q}^a \tag{7b}$$

or

$$\Phi^u \ddot{q}^u = -[\dot{\Phi}^a \quad \Phi^u] \begin{bmatrix} \dot{q}^a \\ \dot{q}^u \end{bmatrix} - \Phi^a \ddot{q}^a \tag{7c}$$

Using Eqs. (5) and (6), we can write

$$\ddot{q}^u = -(\Phi^u)^{-1}(\dot{\Phi} \dot{q} + \Phi^a \ddot{q}^a) \tag{8}$$

where $\dot{\Phi}^a = \frac{d\Phi^a}{dt}$ and $\dot{\Phi}^u = \frac{d\Phi^u}{dt}$. In the NOC, the twist vector, t_i , and its derivative, \dot{t}_i , of i th rigid body can be defined as

$$t_i = \begin{bmatrix} \omega_i \\ v_i \end{bmatrix}_{6 \times 1}, \quad \dot{t}_i = \begin{bmatrix} \dot{\omega}_i \\ \dot{v}_i \end{bmatrix}_{6 \times 1} \quad \text{for } i = 1, 2, \dots, r \tag{9}$$

In which r is number of rigid bodies in a system. Therefore, the twist vector and its time derivative for all rigid bodies existing in the system can be written as

$$t = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_r \end{bmatrix}_{6r \times 1}, \quad \dot{t} = \begin{bmatrix} \dot{t}_1 \\ \dot{t}_2 \\ \vdots \\ \dot{t}_r \end{bmatrix}_{6r \times 1} \tag{10}$$

The twist vector of the i th rigid body can be written as a linear transformation of the joint-velocity vector as follows:

$$t_i = \begin{bmatrix} \omega_i \\ v_i \end{bmatrix} = K_i \dot{q} \tag{11}$$

where $K_i = \frac{\partial t_i}{\partial \dot{q}}$ is a $(6 \times m)$ matrix and is called Jacobian matrix of i th rigid body. Equation (11) is also used in the direct kinematics of a serial type multi-body system to express the twist of the i th rigid body as a linear transformation of the joint velocities. Equation (11), for $i = 1, 2, \dots, r$, can be assembled and expressed in compact form as

$$t = K \dot{q}, \quad K = \begin{bmatrix} K_1 \\ \vdots \\ K_r \end{bmatrix}_{6r \times m} \tag{12}$$

where K represents Jacobian matrix of all rigid bodies existing in a system. Additionally, the K matrix can be divided into two parts K^a and K^u as

$$K = [K^a]_{6r \times n} \quad [K^u]_{6r \times (m-n)} = \begin{bmatrix} \frac{\partial t}{\partial q^a} & \frac{\partial t}{\partial q^u} \end{bmatrix}_{6r \times m} \tag{13}$$

Therefore, Eq. (12) can be rewritten as

$$t = K^a \dot{q}^a + K^u \dot{q}^u \tag{14}$$

The time derivative of Eq. (12) leads to the twist vector differentiation as

$$\dot{t} = K \ddot{q} + \dot{K} \dot{q} \tag{15}$$

Using Eqs. (3), (5), (6), and (7), we can obtain the joint-velocity vector as a function of independent the joint-velocity vector as

$$\dot{q} = L \dot{q}^a \tag{16}$$

where the \mathbf{L} matrix is defined as

$$\mathbf{L} = \begin{bmatrix} \mathbf{I}_{n \times n} \\ -(\Phi^u)^{-1} \Phi^a \end{bmatrix}_{m \times n} \tag{17}$$

The \mathbf{L} matrix is called joint orthogonal complement and shows the relation between the independent and dependent joint velocities. Multiplying the two sides of Eq. (16) by the Φ matrix and using Eq. (3), we will have

$$\Phi \mathbf{L} \dot{\mathbf{q}}^a = 0 \tag{18}$$

Since the $\dot{\mathbf{q}}^a$ vector represents array of independent and arbitrary joint rates, we can write

$$\Phi \mathbf{L} = 0 \tag{19}$$

The time derivative of Eq. (16) can be written as

$$\dot{\mathbf{q}} = \mathbf{L} \ddot{\mathbf{q}}^a + \dot{\mathbf{L}} \dot{\mathbf{q}}^a \tag{20}$$

By substitution Eq. (16) into Eq. (12), we will have

$$\mathbf{t} = \mathbf{K} \dot{\mathbf{q}} = \mathbf{K} \mathbf{L} \dot{\mathbf{q}}^a = \mathbf{T} \dot{\mathbf{q}}^a \tag{21}$$

where

$$\mathbf{T} = \mathbf{K} \mathbf{L} \tag{22}$$

where \mathbf{T} is a $(6r \times n)$ matrix and is called the NOC matrix. The \mathbf{T} matrix is defined as the linear transformation which maps the independent joint velocities into the generalized twist of the system. Substituting the \mathbf{K} matrix from Eq. (13) and the \mathbf{L} matrix from Eq. (17) into above equation leads to

$$\mathbf{T} = \mathbf{K}^a - \mathbf{K}^u (\Phi^u)^{-1} \Phi^a \tag{23}$$

Additionally, the time derivative of Eq. (21) leads to

$$\dot{\mathbf{t}} = \mathbf{T} \ddot{\mathbf{q}}^a + \dot{\mathbf{T}} \dot{\mathbf{q}}^a \tag{24}$$

where

$$\dot{\mathbf{T}} = \dot{\mathbf{K}}^a - \dot{\mathbf{K}}^u (\Phi^u)^{-1} \Phi^a + \mathbf{K}^u (\Phi^u)^{-1} \dot{\Phi}^u (\Phi^u)^{-1} \Phi^a - \mathbf{K}^u (\Phi^u)^{-1} \dot{\Phi}^a \tag{25}$$

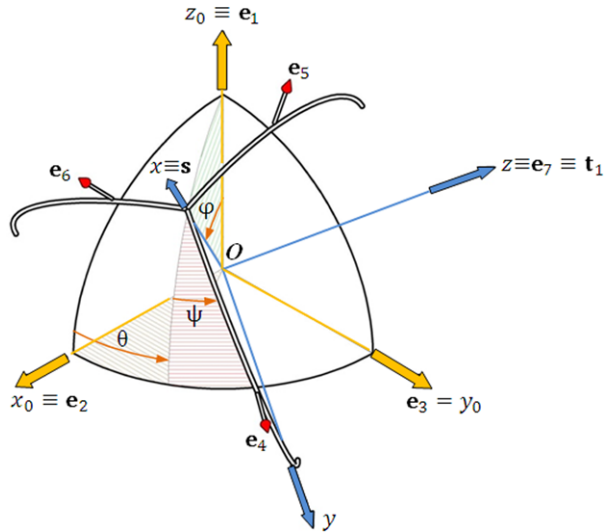
Equations (17), (23), and (25) allows obtaining the \mathbf{L} , \mathbf{T} and $\dot{\mathbf{T}}$ matrices when the constraint equations of a multi-body system are supplied. These matrices will be used in dynamics analysis of the multi-body system.

4 Constraints equations for the SST parallel manipulator

Consider the joint variables γ_k, μ_k and β_k , for $k = 1, 2, 3$, as shown in Fig. 2. Additionally, consider Fig. 3 in which the moving coordinate frame $\{E\}$, xyz , is attached to the MSS and the base coordinate frame $\{B\}$, $x_0y_0z_0$, is attached to the robot fixed base. To write the constraint equations between joint variables of the SST robot, for each branch, we must place the $x_0y_0z_0$ frame on the xyz frame. Another words, we must obtain a rotation matrix between these two frames. For this purpose, we need to define four additional coordinate frames for each branch as

- A fixed, non moving, coordinate frame for each leg, $\{0k\}$ or $x_{0k}y_{0k}z_{0k}$.

Fig. 3 The moving and base coordinate frames



- A moving coordinate frame for each leg, {1k} or $x_{1k}y_{1k}z_{1k}$, attached to the actuated link.
- A moving coordinate frame for each leg, {2k} or $x_{2k}y_{2k}z_{2k}$, attached to the intermediate passive link.
- A moving coordinate frame for each leg, {3k} or $x_{3k}y_{3k}z_{3k}$, attached to the MSS.

Next, consider the coordinate frames in Fig. 4 for k th branch as well as Figs. 5, 6 and 7 for each branch. Using rotation matrices, we can place the coordinate frame $x_{0k}y_{0k}z_{0k}$ on the moving coordinate frame $x_{3k}y_{3k}z_{3k}$ by

$${}^{0k}_{3k}\mathbf{R} = {}^{0k}_{1k}\mathbf{R}{}^{1k}_{2k}\mathbf{R}{}^{2k}_{3k}\mathbf{R} \quad \text{for } k = 1, 2, 3 \tag{26}$$

where

$${}^{0k}_{1k}\mathbf{R} = \text{Rot}(z_{0k}, \gamma_k) \quad \text{for } k = 1, 2, 3, \tag{27}$$

$${}^{1k}_{2k}\mathbf{R} = \text{Rot}(x_{1k}, \mu_k) \quad \text{for } k = 1, 2, 3, \tag{28}$$

$${}^{2k}_{3k}\mathbf{R} = \text{Rot}(z_{2k}, \beta_k) \quad \text{for } k = 1, 2, 3 \tag{29}$$

Consider the first branch in Fig. 5. The coordinate frame $x_{01}y_{01}z_{01}$ coincides with the base coordinate frame {B}. Additionally, the coordinate frame $x_{31}y_{31}z_{31}$ also coincides with the moving coordinate frame {E}. Therefore, we can write

$${}^B_E\mathbf{R} = {}^{01}_{31}\mathbf{R} \tag{30}$$

Consider the second branch in Fig. 6. We can place the coordinate frame $x_{32}y_{32}z_{32}$ on the moving coordinate frame {E} using rotation about the x_{32} axis by angle $(-\alpha_2)$. Therefore, we can write

$${}^{32}_E\mathbf{R} = {}^{32}_{31}\mathbf{R} = \text{Rot}(x_{32}, -\alpha_2) = \text{Rot}(s, -\alpha_2) \tag{31}$$

Additionally, the rotation matrix between the coordinate frames $x_{01}y_{01}z_{01}$ and $x_{02}y_{02}z_{02}$ can be written as

$${}^B_{02}\mathbf{R} = {}^{01}_{02}\mathbf{R} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{32}$$

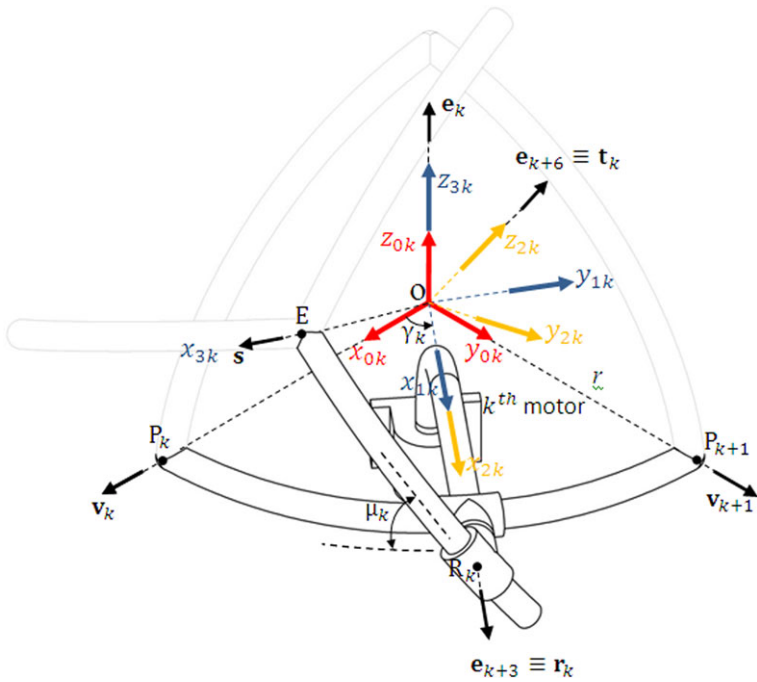
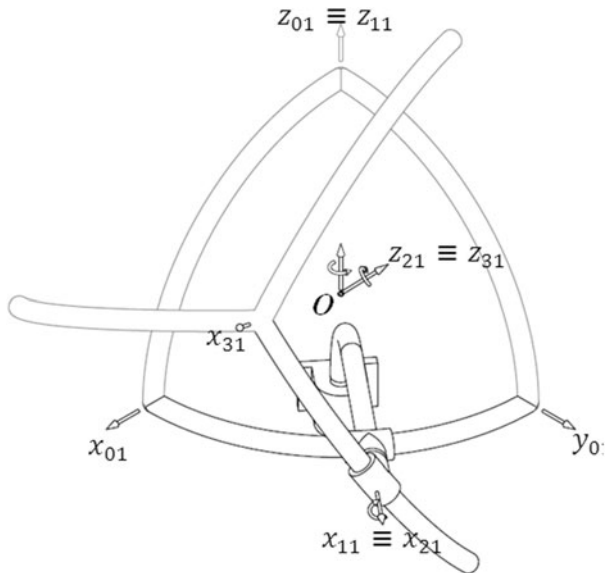


Fig. 4 Coordinate frame description of k th leg

Fig. 5 Coordinate frames for the first branch



Using Eqs. (26), (31), and (32), we can write

$${}^B \mathbf{R} = {}^0_1 \mathbf{R} {}^1_2 \mathbf{R} {}^2_3 \mathbf{R} {}^3_1 \mathbf{R} = {}^B_{02} \mathbf{R} {}^0_2 \mathbf{R} {}^2_3 \mathbf{R} {}^3_1 \mathbf{R} \quad (33)$$

Fig. 6 Coordinate frames for the second branch

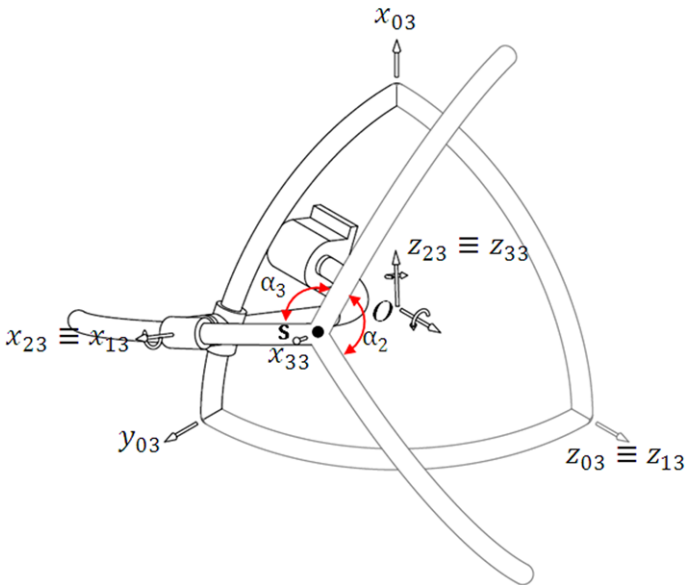
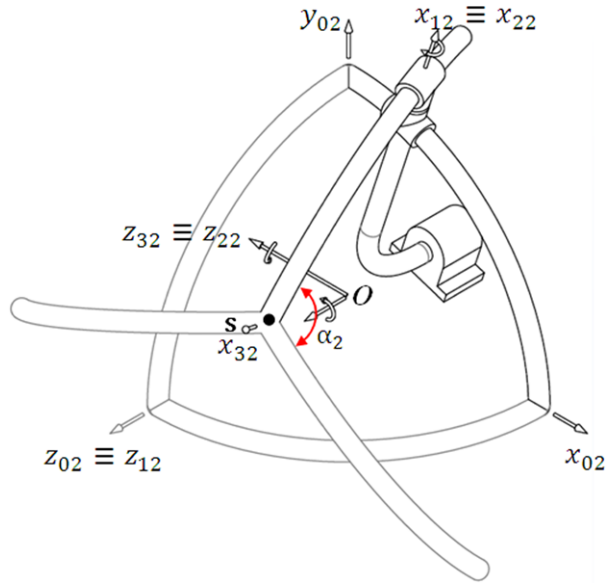


Fig. 7 Coordinate frames for the third branch

Similarly, for the third branch (see Fig. 7), we can obtain

$${}^B\mathbf{R} = {}^0_1\mathbf{R} {}^0_3\mathbf{R} {}^3_3\mathbf{R} {}^3_3\mathbf{R} = {}^B_0\mathbf{R} {}^0_3\mathbf{R} {}^3_3\mathbf{R} {}^3_3\mathbf{R} \tag{34}$$

where

$${}^3_3\mathbf{R} = {}^3_3\mathbf{R} = \text{Rot}(x_{31}, -(\alpha_2 + \alpha_3)) = \text{Rot}(\mathbf{s}, -(\alpha_2 + \alpha_3)), \tag{35}$$

$${}^B_{03}\mathbf{R} = {}^{01}_{03}\mathbf{R} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{36}$$

Finally, the three rotation matrices, obtained from each branch, are equated in order to obtain the constraint equations. Therefore, we can write

$${}^B_{\text{E}}\mathbf{R} = {}^{01}_{31}\mathbf{R} = {}^{01}_{02}\mathbf{R} {}^{02}_{32}\mathbf{R} {}^{32}_{31}\mathbf{R} = {}^{01}_{03}\mathbf{R} {}^{03}_{33}\mathbf{R} {}^{33}_{31}\mathbf{R} \tag{37}$$

By equating corresponding terms of the above equation, six independent constraint equations can be obtained. The six equations also represent the constraint Eq. (2), $\mathbf{f}(\mathbf{q}) = \mathbf{0}$, which is utilized to obtain the \mathbf{L} matrix.

5 Computing the \mathbf{K} matrix of the SST robot

To write the \mathbf{K} matrix, we must write the twist vector, \mathbf{t} , for every one of the system rigid bodies. Then, using Eq. (11), the matrix \mathbf{K}_i and consequently matrix \mathbf{K} can be determined. Consider Figs. 3 and 4. To obtain the twist vector, \mathbf{t} , several steps are taken. First, extension of the robot joints are defined by vectors \mathbf{e}_i as

$$\mathbf{e}_1 = [0 \ 0 \ 1]^T, \tag{38}$$

$$\mathbf{e}_2 = [1 \ 0 \ 0]^T, \tag{39}$$

$$\mathbf{e}_3 = [0 \ 1 \ 0]^T, \tag{40}$$

$$\mathbf{e}_4 = {}^{01}_{11}\mathbf{R} [1 \ 0 \ 0]^T, \tag{41}$$

$$\mathbf{e}_5 = {}^{01}_{02}\mathbf{R} {}^{02}_{12}\mathbf{R} [1 \ 0 \ 0]^T, \tag{42}$$

$$\mathbf{e}_6 = {}^{01}_{03}\mathbf{R} {}^{03}_{13}\mathbf{R} [1 \ 0 \ 0]^T, \tag{43}$$

$$\mathbf{e}_7 = {}^{01}_{11}\mathbf{R} {}^{11}_{21}\mathbf{R} [0 \ 0 \ 1]^T \tag{44}$$

These unit vectors were earlier defined in Sect. 2. Additionally, consider Figs. 1 and 2. The actuated links of the branches 1, 2, and 3 are called rigid bodies 1, 2, and 3, respectively. The intermediate passive links of the branches 1, 2, and 3 are called rigid bodies 4, 5, and 6, respectively. The MSS is called rigid body 7.

In the next step, position vectors \mathbf{r}_{ij} are defined. Here indices i and j represent the joint number and rigid body number, respectively. The vector \mathbf{r}_{ij} connects joint number i to center of mass of the rigid body j . Consider Fig. 8. Using the simple assumption of having mass center of the actuated link being located in the middle of the rigid body, we can write

$$\mathbf{r}_{11} = 0.5R\mathbf{e}_4, \tag{45}$$

$$\mathbf{r}_{22} = 0.5R\mathbf{e}_5, \tag{46}$$

$$\mathbf{r}_{33} = 0.5R\mathbf{e}_6 \tag{47}$$

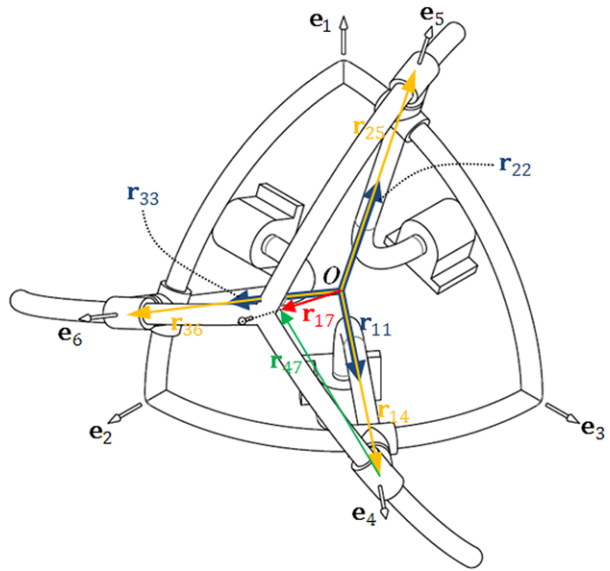
Additionally, we assume that the mass centers of the intermediate passive links are at a distance of R , radius of sphere, from center of the sphere.

$$\mathbf{r}_{14} = R\mathbf{e}_4, \tag{48}$$

$$\mathbf{r}_{25} = R\mathbf{e}_5, \tag{49}$$

$$\mathbf{r}_{36} = R\mathbf{e}_6 \tag{50}$$

Fig. 8 The position vectors \mathbf{r}_{ij}



The mass centers of the passive revolute joints are located on their respective axis of rotation. Therefore

$$\mathbf{r}_{44} = \mathbf{0}, \tag{51}$$

$$\mathbf{r}_{55} = \mathbf{0}, \tag{52}$$

$$\mathbf{r}_{66} = \mathbf{0} \tag{53}$$

The mass center of the MSS may help define the other vectors as

$$\mathbf{r}_{17} = (2R/\pi) {}_{02}^0 \mathbf{R} {}_{32}^0 \mathbf{R} {}_{31}^3 \mathbf{R} [1 \ 0 \ 0]^T \tag{54}$$

We can then write

$$\mathbf{r}_{47} = -\mathbf{r}_{14} + \mathbf{r}_{17} \tag{55}$$

The twist vectors of the rigid bodies 1 to 7 can now be written as

$$\mathbf{t}_1 = \begin{bmatrix} \boldsymbol{\omega}_1 \\ \mathbf{v}_1 \end{bmatrix} = \begin{bmatrix} \dot{\gamma}_1 \mathbf{e}_1 \\ \dot{\gamma}_1 \mathbf{e}_1 \times \mathbf{r}_{11} \end{bmatrix}, \tag{56}$$

$$\mathbf{t}_2 = \begin{bmatrix} \boldsymbol{\omega}_2 \\ \mathbf{v}_2 \end{bmatrix} = \begin{bmatrix} \dot{\gamma}_2 \mathbf{e}_2 \\ \dot{\gamma}_2 \mathbf{e}_2 \times \mathbf{r}_{22} \end{bmatrix}, \tag{57}$$

$$\mathbf{t}_3 = \begin{bmatrix} \boldsymbol{\omega}_3 \\ \mathbf{v}_3 \end{bmatrix} = \begin{bmatrix} \dot{\gamma}_3 \mathbf{e}_3 \\ \dot{\gamma}_3 \mathbf{e}_3 \times \mathbf{r}_{33} \end{bmatrix}, \tag{58}$$

$$\mathbf{t}_4 = \begin{bmatrix} \boldsymbol{\omega}_4 \\ \mathbf{v}_4 \end{bmatrix} = \begin{bmatrix} \dot{\gamma}_1 \mathbf{e}_1 - \dot{\mu}_1 \mathbf{e}_4 \\ \dot{\gamma}_1 \mathbf{e}_1 \times \mathbf{r}_{14} - \dot{\mu}_1 \mathbf{e}_4 \times \mathbf{r}_{44} \end{bmatrix}, \tag{59}$$

$$\mathbf{t}_5 = \begin{bmatrix} \boldsymbol{\omega}_5 \\ \mathbf{v}_5 \end{bmatrix} = \begin{bmatrix} \dot{\gamma}_2 \mathbf{e}_2 - \dot{\mu}_2 \mathbf{e}_5 \\ \dot{\gamma}_2 \mathbf{e}_2 \times \mathbf{r}_{25} - \dot{\mu}_2 \mathbf{e}_5 \times \mathbf{r}_{55} \end{bmatrix}, \tag{60}$$

$$\mathbf{t}_6 = \begin{bmatrix} \boldsymbol{\omega}_6 \\ \mathbf{v}_6 \end{bmatrix} = \begin{bmatrix} \dot{\gamma}_3 \mathbf{e}_3 - \dot{\mu}_3 \mathbf{e}_6 \\ \dot{\gamma}_3 \mathbf{e}_3 \times \mathbf{r}_{36} - \dot{\mu}_3 \mathbf{e}_6 \times \mathbf{r}_{66} \end{bmatrix}, \tag{61}$$

$$\mathbf{t}_7 = \begin{bmatrix} \boldsymbol{\omega}_7 \\ \mathbf{v}_7 \end{bmatrix} = \begin{bmatrix} \dot{\gamma}_1 \mathbf{e}_1 - \dot{\mu}_1 \mathbf{e}_4 - \dot{\beta}_1 \mathbf{e}_7 \\ \dot{\gamma}_1 \mathbf{e}_1 \times \mathbf{r}_{17} - \dot{\mu}_1 \mathbf{e}_4 \times \mathbf{r}_{47} - \dot{\beta}_1 \mathbf{e}_7 \times \mathbf{r}_{17} \end{bmatrix} \tag{62}$$

Therefore, using Eq. (12) the \mathbf{K} matrix representing Jacobian of all rigid bodies can be obtained. Upon obtaining the \mathbf{K} and \mathbf{L} matrices, Eq. (22) can be used to obtain the NOC matrix \mathbf{T} .

6 Dynamics modeling

Consider a multi-body system consisting of r rigid bodies. The wrench acting on the i th rigid body can be represented as

$$\mathbf{w}_i^* = [\mathbf{n}_i^* \quad \mathbf{f}_i^*]^T \quad \text{for } i = 1, 2, \dots, r \tag{63}$$

where \mathbf{n}_i^* and \mathbf{f}_i^* represent the D’Alembert’s resultant inertia force and torque exerted at the center of mass, respectively. Therefore, we can write

$$\mathbf{f}_i^* = -m_i \ddot{\mathbf{c}}_i \quad \text{for } i = 1, 2, \dots, r, \tag{64}$$

$$\mathbf{n}_i^* = -\mathbf{I}_i \dot{\boldsymbol{\omega}}_i - \boldsymbol{\omega}_i \times \mathbf{I}_i \boldsymbol{\omega}_i \quad \text{for } i = 1, 2, \dots, r \tag{65}$$

where m_i is the body mass, $\ddot{\mathbf{c}}_i$ is acceleration of the i th mass center, \mathbf{I}_i is moment of inertia of the i th body about i th mass center, $\boldsymbol{\omega}_i$ is angular velocity of the i th body and $\dot{\boldsymbol{\omega}}_i$ is angular acceleration of the i th body. The Newton–Euler equations for each individual body can be written as

$$\mathbf{w}_i^* = -\mathbf{M}_i \dot{\mathbf{t}}_i - \boldsymbol{\Omega}_i \mathbf{M}_i \mathbf{t}_i \quad \text{for } i = 1, 2, \dots, r \tag{66}$$

where \mathbf{M}_i and $\boldsymbol{\Omega}_i$ are defined as

$$\mathbf{M}_i = \begin{bmatrix} \mathbf{I}_i & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & m_i \mathbf{1}_{3 \times 3} \end{bmatrix}_{6 \times 6}, \quad \boldsymbol{\Omega}_i = \begin{bmatrix} \boldsymbol{\omega}_i \mathbf{1}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix}_{6 \times 6} \quad \text{for } i = 1, 2, \dots, r \tag{67}$$

When all r rigid bodies in a multi-body system are considered, Eq. (66) can be written as

$$\mathbf{w}^* = -\mathbf{M}_i \dot{\mathbf{t}}_i - \boldsymbol{\Omega}_i \mathbf{M}_i \mathbf{t}_i = -\mathbf{M} \dot{\mathbf{t}} - \boldsymbol{\Omega} \mathbf{M} \mathbf{t} \tag{68}$$

where

$$\mathbf{M} = \text{diag}(\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_r), \tag{69}$$

$$\boldsymbol{\Omega} = \text{diag}(\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \dots, \boldsymbol{\Omega}_r) \tag{70}$$

and \mathbf{w}^* is called the dissipate wrench vector and is defined as

$$\mathbf{w}^* = [(\mathbf{w}_1^*)^T \quad (\mathbf{w}_2^*)^T \quad \dots \quad (\mathbf{w}_r^*)^T]^T \tag{71}$$

The vector of generalized forces of the actuated joints, $\boldsymbol{\tau}^a$, that includes forces and moments of actuated joints is defined as

$$\boldsymbol{\tau}^a = [(\boldsymbol{\tau}_1^a)^T \quad (\boldsymbol{\tau}_2^a)^T \quad \dots \quad (\boldsymbol{\tau}_n^a)^T]^T \tag{72}$$

where n is the number of actuated joints. Therefore, we can obtain powers of the actuated joints as

$$\pi^a = (\dot{\mathbf{q}}^a)^T \boldsymbol{\tau}^a \tag{73}$$

Additionally, the powers related to the inertia (π^*), gravity (π^g) and friction (π^f) can be written as

$$\pi^* = \sum_{i=1}^r \mathbf{t}_i \cdot \mathbf{w}_i^* = \sum_{i=1}^r \mathbf{t}_i^T \mathbf{w}_i^* = \mathbf{t}^T \mathbf{w}^*, \tag{74}$$

$$\pi^g = \sum_{i=1}^r \mathbf{t}_i \cdot \mathbf{w}_i^g = \sum_{i=1}^r \mathbf{t}_i^T \mathbf{w}_i^g = \mathbf{t}^T \mathbf{w}^g, \tag{75}$$

$$\pi^f = \sum_{i=1}^r \mathbf{t}_i \cdot \mathbf{w}_i^f = \sum_{i=1}^r \mathbf{t}_i^T \mathbf{w}_i^f = \mathbf{t}^T \mathbf{w}^f \tag{76}$$

where \mathbf{w}^g and \mathbf{w}^f represent the gravity and friction wrenches, respectively. We can write

$$\mathbf{w}^g = [(\mathbf{w}_1^g)^T \quad (\mathbf{w}_2^g)^T \quad \dots \quad (\mathbf{w}_r^g)^T]^T, \tag{77}$$

$$\mathbf{w}^f = [(\mathbf{w}_1^f)^T \quad (\mathbf{w}_2^f)^T \quad \dots \quad (\mathbf{w}_r^f)^T]^T \tag{78}$$

Using the natural orthogonal complement \mathbf{T} , Eqs. (74), (75), and (76) can be written as

$$\pi^* = (\dot{\mathbf{q}}^a)^T \mathbf{T}^T \mathbf{w}^*, \tag{79}$$

$$\pi^g = (\dot{\mathbf{q}}^a)^T \mathbf{T}^T \mathbf{w}^g, \tag{80}$$

$$\pi^f = (\dot{\mathbf{q}}^a)^T \mathbf{T}^T \mathbf{w}^f \tag{81}$$

In a dynamics system, if we apply the D'Alembert's resultant inertia forces and torques, we can say that the sum of the above powers is zero. Therefore, we have

$$\pi^a + \pi^* + \pi^g + \pi^f = 0 \tag{82}$$

Substituting Eqs. (79), (80), and (81) into the above equation will yield

$$(\dot{\mathbf{q}}^a)^T \boldsymbol{\tau}^a + (\dot{\mathbf{q}}^a)^T \mathbf{T}^T \mathbf{w}^* + (\dot{\mathbf{q}}^a)^T \mathbf{T}^T \mathbf{w}^g + (\dot{\mathbf{q}}^a)^T \mathbf{T}^T \mathbf{w}^f = 0 \tag{83}$$

Factoring actuated velocity vector will yield

$$(\dot{\mathbf{q}}^a)^T (\boldsymbol{\tau}^a + \mathbf{T}^T \mathbf{w}^* + \mathbf{T}^T \mathbf{w}^g + \mathbf{T}^T \mathbf{w}^f) = 0 \tag{84}$$

Since the above equation is valid for any $\dot{\mathbf{q}}^a$, it follows that

$$\boldsymbol{\tau}^a + \mathbf{T}^T \mathbf{w}^* + \mathbf{T}^T \mathbf{w}^g + \mathbf{T}^T \mathbf{w}^f = 0 \tag{85}$$

or

$$\boldsymbol{\tau}^a = -\mathbf{T}^T (\mathbf{w}^* + \mathbf{w}^g + \mathbf{w}^f) \tag{86}$$

Furthermore, we can write the power dissipated by friction in another form as

$$\pi^f = \dot{\mathbf{q}}^T \boldsymbol{\tau}^f \tag{87}$$

Substituting Eq. (16) into the above equation will yield

$$\pi^f = (\dot{\mathbf{q}}^a)^T \mathbf{L}^T \boldsymbol{\tau}^f \tag{88}$$

Therefore, Eq. (86) can be rewritten as

$$\boldsymbol{\tau}^a = -\mathbf{T}^T (\mathbf{w}^* + \mathbf{w}^g) - \mathbf{L}^T \boldsymbol{\tau}^f \tag{89}$$

If we substitute \mathbf{w}^* from Eq. (68) into the above equation, it will yield

$$\mathbf{T}^T \mathbf{M} \dot{\mathbf{t}} + \mathbf{T}^T \boldsymbol{\Omega} \mathbf{M} \mathbf{t} - \mathbf{T}^T \mathbf{w}^g - \mathbf{L}^T \boldsymbol{\tau}^f = \boldsymbol{\tau}^a \tag{90}$$

Substituting Eqs. (21) and (24) into the above equation will yield

$$\mathbf{T}^T \mathbf{M} \mathbf{T} \ddot{\mathbf{q}}^a + (\mathbf{T}^T \mathbf{M} \dot{\mathbf{T}} + \mathbf{T}^T \boldsymbol{\Omega} \mathbf{M} \mathbf{T}) \dot{\mathbf{q}}^a - \mathbf{T}^T \mathbf{w}^g - \mathbf{L}^T \boldsymbol{\tau}^f = \boldsymbol{\tau}^a \tag{91}$$

This equation represents system dynamical model in terms of the actuated joints. Finally, the dynamics equation of motion can be represented as

$$\mathbf{I} \ddot{\mathbf{q}}^a + \mathbf{C} \dot{\mathbf{q}}^a - \boldsymbol{\tau}^f - \boldsymbol{\tau}^g = \boldsymbol{\tau}^a \tag{92}$$

where

$$\mathbf{I} = \mathbf{I}(\mathbf{q}) = \mathbf{T}^T \mathbf{M} \mathbf{T} \tag{93}$$

$$\mathbf{C} = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{T}^T \mathbf{M} \dot{\mathbf{T}} + \mathbf{T}^T \boldsymbol{\Omega} \mathbf{M} \mathbf{T} \tag{94}$$

$$\boldsymbol{\tau}^g = \boldsymbol{\tau}^g(\mathbf{q}) = \mathbf{T}^T \mathbf{w}^g \tag{95}$$

$$\boldsymbol{\tau}^f = \boldsymbol{\tau}^f(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{L}^T \boldsymbol{\tau}^f \tag{96}$$

From the foregoing discussion, then, it becomes apparent that Eq. (92) represents the system’s Euler–Lagrange dynamical equations, free of non-working generalized constraint forces.

6.1 Inverse dynamics analysis

For the inverse dynamics problem, a desired trajectory of the MSS is given and the problem is to determine the input torques required to produce the desired motion. The solution process first requires solving the inverse kinematics problem, which calculates the required values of the actuated joint-trajectories, \mathbf{q}^a , $\dot{\mathbf{q}}^a$ and $\ddot{\mathbf{q}}^a$. Next, the \mathbf{K} , \mathbf{L} and \mathbf{T} matrices are used to evaluate Eqs. (93) to (96). The dynamic Eq. (92) can now be used to obtain the necessary actuated torques.

A possible advantage of the NOC for the inverse dynamics calculations may be its lower computation time. As will be shown in Sect. 7.2, inverse dynamics computation time using the NOC is approximately 37 % lower than the virtual work method. Clearly, this statement should be taken with great care as a true scientific comparison requires counting and comparing exact number of additions and multiplications for each method.

6.2 Direct dynamics analysis

The direct dynamics problem aims to find the response of a robot arm corresponding to given applied moments and/or forces. That is, given the vector of joint moments or forces, it computes the resulting motion of the manipulator as a function of time. In the direct dynamics problem, the vector of initial actuated joint positions and vector of initial actuated joint velocities are given. Therefore, we need to formulate the dynamics equations in terms of the generalized coordinates of joints and their time derivatives. Unlike other common dynamics methods such as principle of virtual work [10], solution of the direct dynamics problem does not require directly solving the direct kinematics problem. Additionally, the dynamics equations are formulated in terms of only the actuated joints [11]. However, using the NOC kinematics, the equations are integrated with its dynamics as a set of differential equations. Furthermore, the \mathbf{L} matrix allows obtaining the unactuated joint velocities given the actuated joint velocities. This eliminates the need to obtain the dynamics equations in terms of only the actuated joints. These advantages can potentially simplify the direct dynamics problem.

The direct dynamics problem may be solved using Eqs. (7) and (92), a state space for \mathbf{q} and $\dot{\mathbf{q}}$. Now having \mathbf{q} , the position at the middle of MSS (point E) can be computed at any time using the unit vector \mathbf{s} that is defined as

$$\mathbf{s} = {}^{01}\mathbf{R} {}^{02}\mathbf{R} {}^{32}\mathbf{R} {}^{31}\mathbf{R} [1 \ 0 \ 0]^T$$

7 Numerical simulation

The solution outlined in this paper is applied to an isotropic model of the SST robot [14]. See Fig. 1. Based on the previous sections, a computer program is developed using MATLAB software for both direct and inverse dynamics solutions. For inverse dynamics simulation, a trajectory for the MSS is supplied and required motor torques are calculated. Results are verified with another reference [10] using a different method. For direct dynamics simulation, output of the inverse dynamics simulation is used. Results are confirmed with input trajectory used in the inverse dynamics simulation.

7.1 Specification of the SST manipulator

(a) *Architecture parameters—fixed base*: Assume that the radius is 0.35 meters and that the planes OP_1P_2 , OP_2P_3 and OP_3P_1 of SST manipulator are located in x - y , y - z and z - x of base coordinate frame, respectively. See Fig. 1. Therefore

$$\mathbf{v}_1 = [1 \ 0 \ 0]^T, \quad \mathbf{v}_2 = [0 \ 1 \ 0]^T, \quad \mathbf{v}_3 = [0 \ 0 \ 1]^T$$

(b) *Architecture parameters—MSS*: Assume radius is 0.4 meters and that the angle between the planes OER_k and OER_{k+1} is 120 degree. See Figs. 1 and 3. Therefore

$$\alpha_1 = \alpha_2 = \alpha_3 = 120^\circ$$

(c) *Mass properties of MSS*:

$$m_s = 2.6 \text{ kg}$$

$${}^E\mathbf{I}_s = \begin{bmatrix} 0.224 & 0 & 0 \\ 0 & 0.224 & 0 \\ 0 & 0 & 0.150 \end{bmatrix} \text{ kg}\cdot\text{m}^2$$

(d) Equal mass properties for each actuator link, $k = 1, 2, 3$:

$$m_a = 0.3 \text{ kg}$$

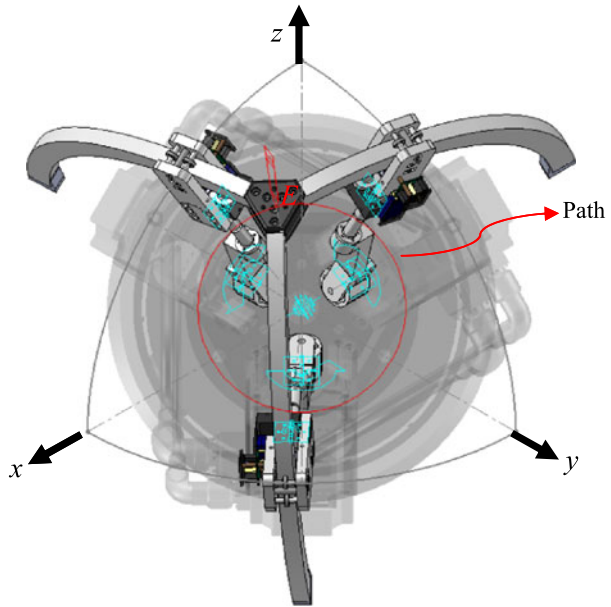
$${}^{1k}\mathbf{I}_a = \begin{bmatrix} 0.0001 & 0 & -0.0001 \\ 0 & 0.008 & 0 \\ -0.0001 & 0 & 0.008 \end{bmatrix} \text{ kg}\cdot\text{m}^2$$

(e) Equal mass properties for each intermediate passive link, $k = 1, 2, 3$:

$$m_{ipl} = 0.06 \text{ kg}$$

$${}^{2k}\mathbf{I}_{ipl} = \begin{bmatrix} 0.00001 & 0 & 0 \\ 0 & 0.005 & 0 \\ 0 & 0 & 0.005 \end{bmatrix} \text{ kg}\cdot\text{m}^2$$

Fig. 9 Inverse dynamics input: circular path followed by point E on MSS



7.2 Example-1: simulation of inverse dynamics

For the inverse dynamics simulation, the trajectory of MSS is specified as follow

$$\theta = \tan^{-1}(s_y/s_x)$$

$$\varphi = \cos^{-1} s_z$$

$$\psi = 0$$

where

$$s_x = \frac{1}{71} \left(4\sqrt{6} \cos(12t) - 12\sqrt{2} \sin(12t) + \frac{\sqrt{13395}}{3} \right)$$

$$s_y = \frac{1}{71} \left(4\sqrt{6} \cos(12t) + 12\sqrt{2} \sin(12t) + \frac{\sqrt{13395}}{3} \right)$$

$$s_z = \frac{1}{71} \left(-8\sqrt{6} \cos(12t) + \frac{\sqrt{13395}}{3} \right)$$

where $0 \leq t \leq \pi/6$. Angles θ , φ and ψ are shown in Fig. 4. This implies that position of point E, moves along a circular path on the surface of the sphere while MSS does not rotate about the unit vector s . See Fig. 9. Results of this simulation are calculated as function of time and are plotted in Fig. 10. The output of the inverse dynamics problem using virtual work [10] and commercial software for this example is also plotted in Fig. 10. As can be seen the required torques are nearly identical. These results verify the correctness of the mathematical model. Additionally, simulation time using the virtual work method is slightly lower than the NOC. The CPU time, for virtual work and the NOC methods, using a 2 GHz processor with 2 GB RAM memory, are 2.7612 and 1.7316 seconds, respectively.

Fig. 10 Required input torques for circular path using both the NOC and virtual work [10]

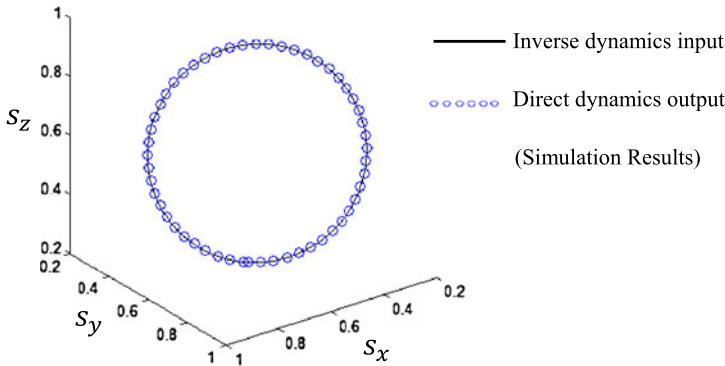
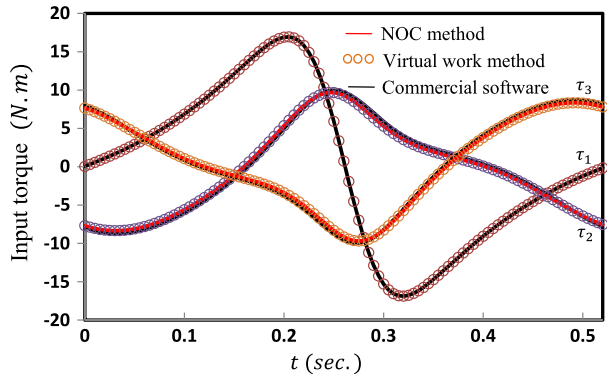


Fig. 11 Direct dynamics output: three-dimensional trajectory of point E

7.3 Example-2: simulation of direct dynamics

For the direct dynamics simulation, the output of the inverse dynamics problem given in previous subsection is used. The input torques, Fig. 10, are applied as a function of time. Additionally, the initial conditions of actuators are assumed as

$$\begin{aligned} \gamma_1 &= \frac{\pi}{4} \text{ rad}, & \gamma_2 &= 0.5549 \text{ rad}, & \gamma_3 &= 1.0159 \text{ rad} \\ \dot{\gamma}_1 &= 4.2096 \frac{\text{rad}}{\text{s}}, & \dot{\gamma}_2 &= -0.6918 \frac{\text{rad}}{\text{s}}, & \dot{\gamma}_3 &= -0.6918 \frac{\text{rad}}{\text{s}} \end{aligned}$$

Simulation result is plotted in Fig. 11. Results show that the trajectory of point E moves along a circular path. Figure 11 also shows that the simulated path and real path are very close. This result verifies the correctness of our mathematical model.

8 Conclusion

The application of the Natural Orthogonal Complement for the inverse and direct dynamic analysis of SST spherical parallel manipulators for the first time is presented. The method uses joint orthogonal complement which indirectly incorporates the kinematics model. Several other advantages of NOC are pointed out. Using this method, the direct and inverse

kinematics problems are indirectly solved. Furthermore, there is no need to obtain the velocity and acceleration inversions in order to solve the inverse and the direct dynamics problems. Using the NOC, the solution for both direct and inverse dynamics problems of the SST parallel robot is presented. The derived formulations can be used to compute the required actuator torques for a given trajectory of the moving platform or to obtain the resulting trajectory for a given actuator torques. The developed formulations are implemented and simulated using two examples. First, inverse kinematics solution is verified using another dynamical method as well as a commercial multi-body dynamical package. Next, direct dynamics results are verified using the inverse dynamics results. The study presented in this paper provides a framework for our future research in the areas of SST manipulator control.

Acknowledgement This work was supported by a grant # 13206 (on March 09, 2010), titled “Design, construction and dynamic analysis of a parallel spherical robot”, sponsored by the Ferdowsi University of Mashhad’s Research Council.

References

1. Fattah, A., Kasaei, G.: Kinematics and dynamics of a parallel manipulator with a new architecture. *Robotica* **18**, 535–543 (2000)
2. Dasgupta, B., Mruthyunjaya, T.S.: A Newton–Euler formulation for the inverse dynamics of the Stewart platform manipulator. *Mech. Mach. Theory* **33**(8), 1135–1152 (1998)
3. Koteswara Rao, A.B., Saha, S.K., Rao, P.V.M.: Dynamics modelling of hexaslides using the decoupled natural orthogonal complement matrices. *Multibody Syst. Dyn.* **15**, 159–180 (2006)
4. Saha, S.K., Angeles, J.: Dynamics of nonholonomic mechanical systems using a natural orthogonal complement. *Trans. Am. Soc. Mech. Eng.* **58**, 238–243 (1991)
5. Saha, S.K.: Dynamics of serial multibody systems using the decoupled natural orthogonal complement matrices. *J. Appl. Mech.* **66**, 986–996 (1999)
6. Khan, W.A., Krovi, V.N., Saha, S.K., Angeles, J.: Modular and recursive kinematics and dynamics for parallel manipulators. *Multibody Syst. Dyn.* **14**(3–4), 419–455 (2005)
7. Staicu, S., Zhang, D.: A novel dynamic modeling approach for parallel mechanisms analysis. *Robot. Comput.-Integr. Manuf.* **24**, 167–172 (2008)
8. Staicu, S.: Inverse dynamics of the 3-PRR planar parallel robot. *Robot. Auton. Syst.* **57**, 556–563 (2009)
9. Staicu, S.: Dynamics analysis of the star parallel manipulator. *Robot. Auton. Syst.* **57**, 1057–1064 (2009)
10. Enferadi, J., Akbarzadeh, A.: Inverse dynamics analysis of a general spherical star-triangle parallel manipulator using principle of virtual work. *Nonlinear Dyn.* **61**(3), 419–434 (2010)
11. Enferadi, J., Akbarzadeh, A.: A virtual work based algorithm for solving direct dynamics problem of a 3-RRP spherical parallel manipulator. *J. Intell. Robot. Syst.* **63**, 25–49 (2010)
12. Bai, S., Hansen, M.R.: Forward kinematics of spherical parallel manipulators with revolute joints. In: *Proceedings of the 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Xi’an, China, 2–5 July 2008*
13. Enferadi, J., Akbarzadeh, A.: A novel approach for forward position analysis of a double-triangle spherical parallel manipulator. *Eur. J. Mech. A, Solids* **29**, 348–355 (2010)
14. Enferadi, J., Akbarzadeh, A.: A novel spherical parallel manipulator: forward position problem, singularity analysis, and isotropy design. *Robotica* **27**, 663–676 (2009)
15. Daniali, H.R.M., Zsombor-Murray, P.J., Angeles, J.: The kinematics of 3-dof planar and spherical double-triangular parallel manipulators. In: Angeles, J., Hommel, G., Kovacs, P. (eds.) *Computational Kinematics*, pp. 153–164. Kluwer Academic, Dordrecht (1993)
16. Kamali, K., Akbarzadeh, A.: A novel method for direct kinematics solution of fully parallel manipulators using basic regions theory. *Proc. IMechE Part I, J. Syst. Control Eng.* **225**(5), 683–701 (2011)
17. Angeles, J., Lee, S.: The formulation of dynamical equations of holonomic mechanical systems using a natural orthogonal complement. *J. Appl. Mech.* **55**, 243–244 (1988)
18. Angeles, J., Ma, O.: Dynamic simulation of n-axis serial robotic manipulators using a natural orthogonal complement. *Int. J. Robot. Res.* **7**(5), 32–47 (1988)
19. Mohan, A., Saha, S.K.: A recursive, numerically stable, and efficient simulation algorithm for serial robots with flexible links. *Multibody Syst. Dyn.* **21**(1), 1–35 (2007)
20. Hanzaki, A.R., Saha, S.K., Rao, P.V.M.: An improved dynamic modeling of a multibody system with spherical joints. *Multibody Syst. Dyn.* **21**(4), 325–345 (2009)

21. Mohan, A., Saha, S.K.: A recursive, numerically stable, and efficient simulation algorithm for serial robots. *Multibody Syst. Dyn.* **17**, 291–319 (2007)
22. Mohan, A., Singh, S.P., Saha, S.K.: A cohesive modeling technique for theoretical and experimental estimation of damping in serial robots with rigid and flexible links. *Multibody Syst. Dyn.* **23**, 333–360 (2010)
23. Bhangale, P.P., Saha, S.K., Agrawal, V.P.: A dynamic model based robot arm selection criterion. *Multibody Syst. Dyn.* **12**, 95–115 (2004)
24. Sundarraman, P., Saha, S.K., Vasa, N.J., Baskaran, R., Sunilkumar, V., Raghavendra, K.: Modeling and analysis of a fuel-injection pump used in diesel engines. *Int. J. Automot. Technol.* **13**(2), 193–203 (2012)
25. Staicu, S.: Recursive modeling in dynamics of Agile Wrist spherical parallel robot. *Robot. Comput.-Integr. Manuf.* **25**, 409–416 (2009)
26. Chiang, C.H.: *Kinematics of Spherical Mechanisms*. Cambridge University Press, Cambridge (1988)