

Inverse dynamics of a 3-prismatic–revolute–revolute planar parallel manipulator using natural orthogonal complement

H Kordjazi and A Akbarzadeh*

Mechanical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

The manuscript was received on 6 May 2010 and was accepted after revision for publication on 26 July 2010.

DOI: 10.1243/09596518JSCE1098

Abstract: The performance of robotic systems with parallel kinematics can be evaluated by their kinematic, static, and dynamic properties. These properties are directly used in model-based controllers which potentially offer higher accuracy for robotic systems. Inverse dynamic solution is an essential part of these controllers. In the present work, the inverse dynamics model of a 3-PRR (prismatic–revolute–revolute) planar parallel manipulator based on the natural orthogonal complement (NOC) method is developed. To drive the NOC for the 3-PRR closed-loop systems, the explicit expressions of the loop constraints equations and the associated Jacobian matrices are first obtained. Next the NOC matrix, which is a velocity transformation matrix relating the Cartesian angular/translational velocities of various bodies to the motor joint rates, is calculated. Finally results of the NOC method are compared with simulation of a 3-PRR planar parallel manipulator using two commercial softwares: SimMechanics toolbox of Matlab and COSMOSMotion of SolidWorks. In order to verify the theoretical results, two different configurations for the robot are considered: a horizontal and a vertical. Results of the NOC method as well as the two simulations are compared for the two robot configurations.

Keywords: inverse dynamics, natural orthogonal complement, planar parallel manipulator

1 INTRODUCTION

Parallel robots are widely used for industrial application ranging from packaging to ultra-high precision assembly tasks and assistant for surgery. These manipulators are closed-loop mechanisms that consist of separate serial chains connecting the moving platform to the fixed base. Compared with the serial robots, parallel mechanisms have some potential advantages such as greater rigidity, greater accuracy, better stiffness, larger dynamic charge capacity, and greater load bearing. But two major drawbacks of parallel manipulators are small workspace and complicated singularities. The workspace of parallel manipulators is small because of the constraints created by the closed-loop kinematic chains. Parallel robots can be equipped with

revolute or prismatic, hydraulic, or pneumatic actuators.

The two principal problems associated with the dynamics of mechanical systems are inverse and direct dynamics. The inverse dynamics model of a robot manipulator provides the input actuator forces or torques required to follow a desired end-effector trajectory. Inverse dynamics is used in a wide range of controllers, in optimum trajectory planning, and also in manipulator design (selection of the actuators, structure design, and so on) [1]. A schematic of how inverse dynamic is used in an on-line control scheme is shown in Fig. 1. Unlike serial manipulators, the application of inverse dynamic for parallel manipulators in control requires the additional solution of direct kinematics. This is because as input, inverse dynamics needs information on both actuated and passive joints. However, in general only the actuated joints are measured. As shown in Fig. 1, to obtain the additional information on the passive joints, direct kinematics is used.

*Corresponding author: Mechanical Engineering, Ferdowsi University of Mashhad, VakilAbad Blvd, Mashhad, Iran.
email: ali_akbarzadeh_t@yahoo.com

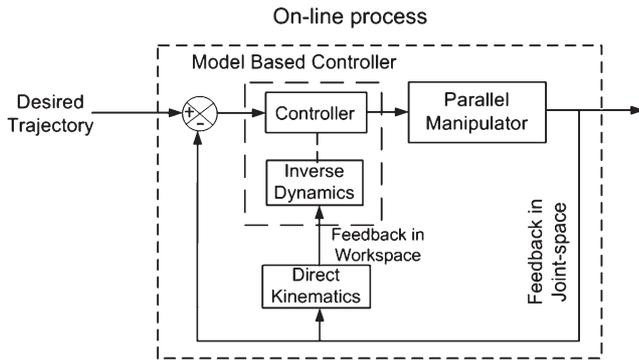


Fig. 1 Inverse dynamics in a model based controller

In the inverse dynamics problem, time histories of all the system degrees of freedom are supplied and the actuated joint torques or forces are computed. Because the position, velocities, and accelerations of the system are known, the solution process is primarily an algebraic one and typically does not require the use of numerical integration methods. Conversely, in the direct or forward dynamics problem, time histories of the actuated joint torques or forces are supplied and time histories of the joint coordinates, velocities, and accelerations are calculated. In the first step of solving for the direct dynamics problem, the equations of motion are solved algebraically to determine the accelerations. Next the underlying ordinary differential equations are integrated to obtain all the joint coordinate time histories. Because of the non-closed form solutions to such systems of non-linear ordinary differential equations, numerical integration methods is needed.

Most studies on methods for calculation of dynamic model of a closed-loop systems, such as parallel manipulator, use Lagrangian formulation [1, 2], the Newton–Euler formulation [3, 4] and the principle of virtual work [5, 6]. Most recently, Staicu [7] used recursive modelling for the kinematics and dynamics of a 3-PRR (prismatic–revolute–revolute) planar parallel manipulator. He used the principle of virtual work in the inverse dynamics problem for 3-PRR manipulator. Newton–Euler approaches typically use Cartesian variables as configuration–space variables. They admit recursive formulations by first developing equations of motion for each single body; these equations are then assembled to obtain the model of the entire system. The commonly known Newton–Euler method, which takes into account the free-body-diagrams of the mechanism, leads to a large number of equations with unknowns among which are also the connecting forces in the joints [5].

Most of the dynamical models based on the Lagrange formalism neglect the weight of intermedi-

ate bodies and take into consideration only the active forces or moments and the wrench of forces applied on the moving platform [5]. Many hypotheses may be assumed for simplifying the dynamic equations of parallel manipulators. For example, Pierrot *et al.* [8] and Codourey and Burdet [9] neglected the inertia of links and assumed point masses placed at the two distal ends of each link. Do and Yang [10] used the assumption that mass of each link is placed in the middle of each link and assumed the movable platform to act like a disc. These assumptions help simplify the inertia matrix.

The concept of natural orthogonal complement (NOC) was introduced by Angeles and Lee [11]. NOC is defined as the linear transformation that maps the independent joint velocities into the generalized twist of the system. The matrix is an orthogonal complement of the velocity constraint matrix arising out of the joints present in the system [11]. The dynamic modelling based on the NOC was found advantageous in references [12] and [13]. Xi and Sinatra [14] presented the inverse dynamics of hexapods using the NOC matrix. Saha [15, 16] introduced a representation of this NOC matrix as the product of two matrices, a lower block triangular matrix and a block diagonal matrix, termed the decoupled NOC matrices. Also Saha and Schiehlen [17] showed that the NOC of a closed loop parallel manipulator can be split into three matrices, namely, the lower-block triangular, the full-block, and the block diagonal.

In this work the NOC method is used for dynamic analysis of a 3-PRR planar parallel manipulator. Using this method, dynamic equations of robot are expressed in terms of the actuated joints. The resulting dynamic model is in the form of Euler–Lagrange without including constraints forces, or torques or the Lagrangian coefficients.

To the best of the authors' knowledge, the application of NOC to dynamic analysis of the 3-PRR robot is new and has not been presented before.

2 JOINT CONSTRAINTS

A joint coordinate is defined as the translational or rotational displacement of a one degree-of-freedom joint, which is used to describe the relative configuration of two successive bodies connected by this joint. Unlike serial manipulators, in parallel manipulators joint coordinates are not all independent. The 3-PRR requires nine generalized coordinates. The joint-position vector \mathbf{q} can be represented by

$$\mathbf{q} = \begin{Bmatrix} q_1 \\ q_2 \\ \vdots \\ q_m \end{Bmatrix} = \begin{Bmatrix} \mathbf{q}^a \\ \mathbf{q}^u \end{Bmatrix} \quad (1)$$

Since coordinate of some joints are independent, q^a and others are dependent, q^u , the joint coordinates are subjected to kinematic constraints. Utilizing the geometrical structure of the robot, constraints can be defined and expressed as

$$\varphi(\mathbf{q}) = 0 \quad (2)$$

The constrain equation, equation (2), can then be used to solve dependent equations with respect to the independent joints. Because of the non-linearity, these equations must be solved numerically. If $\Phi_{(m-n) \times n}$ is considered as the Jacobian matrix of the joint constraints then

$$\Phi = \frac{\partial \varphi(\mathbf{q})}{\partial \mathbf{q}} \quad (3)$$

The matrix $\Phi_{(m-n) \times n}$ can be partitioned into two parts. One consisting of columns of $\Phi_{(m-n) \times n}$ associated with the independent joint velocities and the other consisting of the columns of $\Phi_{(m-n) \times n}$ associated with the dependent joint velocities. Φ^a and Φ^u are defined as

$$\Phi^a = \frac{\partial \varphi(\mathbf{q})}{\partial \mathbf{q}^a}, \quad \Phi^u = \frac{\partial \varphi(\mathbf{q})}{\partial \mathbf{q}^u} \quad (4)$$

It is shown by Ma [18] that the dependent joint velocities can be expressed using the independent joint velocities

$$\dot{\mathbf{q}}^u = -(\Phi^u)^{-1} \Phi^a \dot{\mathbf{q}}^a \quad (5)$$

where $\dot{\mathbf{q}}^u$ is vector of dependent joint velocities and $\dot{\mathbf{q}}^a$ is vector of independent, actuated joint velocities. To continue the dynamic analysis, the concept of twist of rigid body and orthogonal complements is first defined.

3 ORTHOGONAL COMPLEMENT

Displacement of a rigid body can be expressed by a position vector and a rotation matrix. To describe the velocity field, the concept of twist is used. The twist of a rigid body is a six-dimensional vector

defined as

$$\mathbf{t}_i = \begin{bmatrix} \omega_i \\ \mathbf{v}_i \end{bmatrix} \quad (6)$$

The twist of the i th body can be expressed as a linear transformation of the joint velocities [18]

$$\mathbf{t}_i = \mathbf{K}_i \dot{\mathbf{q}} \quad (7)$$

Two types of orthogonal complements are used for dynamic analysis, joint orthogonal complement, and NOC.

3.1 Joint orthogonal complement

The relation between vector of all joint velocities, $\dot{\mathbf{q}}$, and independent joint velocities, $\dot{\mathbf{q}}^a$, can be expressed by joint orthogonal complement, matrix \mathbf{L} , by

$$\dot{\mathbf{q}} = \mathbf{L} \dot{\mathbf{q}}^a \quad (8)$$

where

$$\mathbf{L} = \begin{bmatrix} \mathbf{I}_{n \times n} \\ -(\Phi^u)^{-1} \Phi^a \end{bmatrix}_{m \times n} \quad (9)$$

For planar parallel manipulator n is calculated by equation $n = 3r - 2m$.

3.2 Natural orthogonal complement (NOC)

The NOC is defined as the linear transformation which relates the independent joint velocities to the generalized twist of the system, such as

$$\mathbf{t} = \mathbf{T} \dot{\mathbf{q}}^a \quad (10)$$

In the above equation, \mathbf{T} is the NOC matrix. The relation between \mathbf{T} and \mathbf{L} is given by

$$\mathbf{T} = \mathbf{K} \mathbf{L} \quad (11)$$

where \mathbf{K} is a $6r \times m$ matrix which consist of two parts; \mathbf{K}^a consisting of n columns of \mathbf{K} associated with the actuated joints and \mathbf{K}^u consisting of $(n - m)$ columns of \mathbf{K} associated with the passive joints.

4 DYNAMIC MODEL OF A SYSTEM WITH NOC

As mentioned earlier, dynamics consist of two parts: inverse and direct dynamics. In this section, the

inverse dynamic model of a system is derived using NOC. It is assumed that the bodies are rigid and the mass of each body is placed at its centre of mass. Inverse dynamics equation of a manipulator in terms of the independent joint coordinates can be expressed as [18]

$$\mathbf{M}\ddot{\mathbf{q}}^a + \mathbf{C}\dot{\mathbf{q}}^a + \mathbf{G} = \boldsymbol{\tau}^a \quad (12)$$

where parameters \mathbf{M} , \mathbf{C} , \mathbf{G} , and $\boldsymbol{\tau}^a$ are defined as

$$\mathbf{M} = \mathbf{M}(\mathbf{q}) = \mathbf{T}^T M_{\text{total}} \mathbf{T} \quad (13)$$

$$\mathbf{C} = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{T}^T M_{\text{total}} \dot{\mathbf{T}} + \mathbf{T}^T \boldsymbol{\Omega} M_{\text{total}} \mathbf{T} \quad (14)$$

$$\mathbf{G} = \mathbf{G}(\mathbf{q}) = -\mathbf{T}^T \mathbf{w}^g \quad (15)$$

$$\boldsymbol{\tau}^a = [\tau_1^a \quad \tau_2^a \quad \dots \quad \tau_n^a] \quad (16)$$

It is noted that $\boldsymbol{\tau}^a$ is the generalized actuating force of the robot. In the above relations M_{total} and $\boldsymbol{\Omega}$ are diagonal matrices and defined as

$$M_{\text{total}} = \text{diag}(M_1, M_2, \dots, M_r) \quad (17)$$

$$\boldsymbol{\Omega} = \text{diag}(\Omega_1, \Omega_2, \dots, \Omega_r) \quad (18)$$

where M_i and Ω_i are 6×6 matrices given by

$$M_i = \begin{bmatrix} \mathbf{I}_i & \mathbf{0} \\ \mathbf{0} & m_i \times \mathbf{1} \end{bmatrix}, \quad \Omega_i = \begin{bmatrix} \omega_i \times \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (19)$$

In the above equations, $\mathbf{1}$ is 3×3 identity matrix and $\mathbf{0}$ is a 3×3 zero matrix. Likewise \mathbf{w}^g is defined as

$$\mathbf{w}^g = [0 \quad m_1 \mathbf{g} \quad 0 \quad m_2 \mathbf{g} \quad \dots \quad 0 \quad m_r \mathbf{g}]^T \quad (20)$$

where $\mathbf{0}$ is three-dimensional (3D) zero vector.

5 DESCRIPTION OF THE MANIPULATOR

This section describes the 3-PRR planar parallel manipulator. The manipulator consists of a base plate, a movable platform, and three links. Each link has an actuated prismatic joint and two passive consecutive revolute joints. Therefore, the robot has a total of three active and six passive joints (see Figs 2 and 3). Three degrees-of-freedom (DOF) of the 3-PRR manipulator are the translations along the X and Y axis and the rotation about the Z axis.

6 KINEMATIC CONSTRAINTS

The 3-PRR robot has a total of nine joints and therefore nine generalized coordinates are considered. The relationship between the generalized coordinates defines the kinematics constraint equation, equation (2). The number of independent kinematics constraint equations, l , is given by

$$l = m - n = 9 - 3 = 6 \quad (21)$$

Using Denavit–Hartenberg frame assignment convention [19], the general form of link transformation matrix is as follows

$${}^A_B T = \begin{bmatrix} {}^A_B R & {}^A P_{BORG} \\ 0 & 1 \end{bmatrix} \quad (22)$$

where ${}^A_B R$ is the corresponding 3×3 rotation matrix and ${}^A P_{BORG}$ is the related transformation vector.

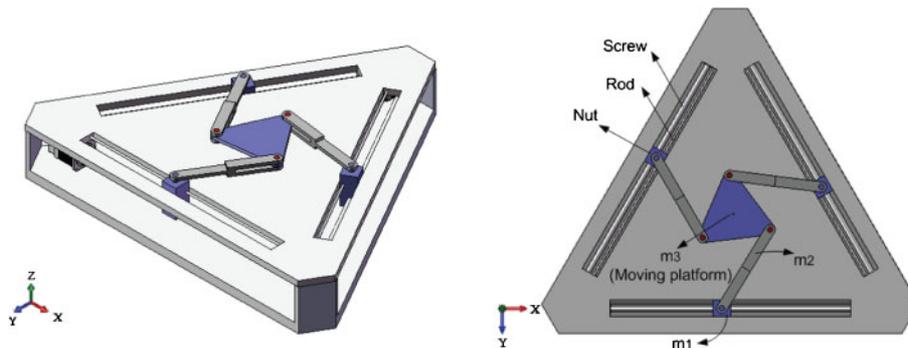


Fig. 2 A 3-PRR planar parallel manipulator

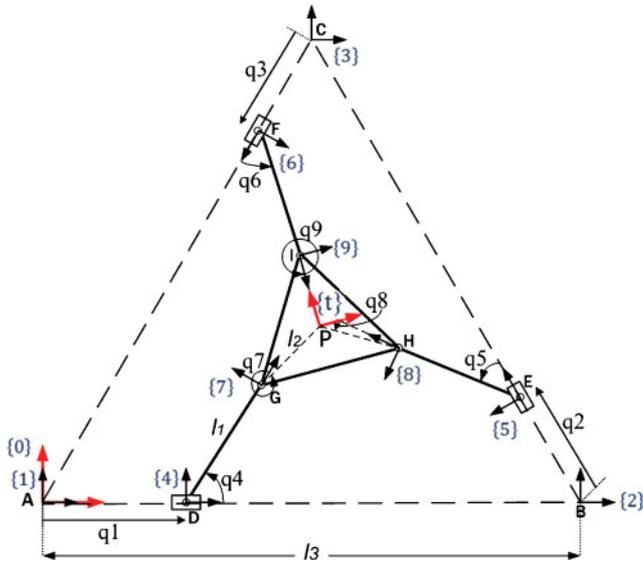


Fig. 3 Generalized coordinates for the 3-PRR manipulator

The individual transformation matrix for each link is

$${}^1_4\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & q_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^4_7\mathbf{T} = \begin{bmatrix} \cos(q_4) & -\sin(q_4) & 0 & l_1 \cos(q_4) \\ \sin(q_4) & \cos(q_4) & 0 & l_1 \sin(q_4) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^7_t\mathbf{T} = \begin{bmatrix} \cos(q_7) & -\sin(q_7) & 0 & l_2 \cos(q_7) \\ \sin(q_7) & \cos(q_7) & 0 & l_2 \sin(q_7) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^5_2\mathbf{T} = \begin{bmatrix} \cos(2\pi/3) & -\sin(2\pi/3) & 0 & q_2 \cos(2\pi/3) \\ \sin(2\pi/3) & \cos(2\pi/3) & 0 & q_2 \sin(2\pi/3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^8_5\mathbf{T} = \begin{bmatrix} \cos(q_5) & -\sin(q_5) & 0 & l_1 \cos(q_5) \\ \sin(q_5) & \cos(q_5) & 0 & l_1 \sin(q_5) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^8_t\mathbf{T} = \begin{bmatrix} \cos(q_8) & -\sin(q_8) & 0 & l_2 \cos(q_8) \\ \sin(q_8) & \cos(q_8) & 0 & l_2 \sin(q_8) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3_6\mathbf{T} = \begin{bmatrix} \cos(4\pi/3) & -\sin(4\pi/3) & 0 & q_3 \cos(4\pi/3) \\ \sin(4\pi/3) & \cos(4\pi/3) & 0 & q_3 \sin(4\pi/3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^6_9\mathbf{T} = \begin{bmatrix} \cos(q_6) & -\sin(q_6) & 0 & l_1 \cos(q_6) \\ \sin(q_6) & \cos(q_6) & 0 & l_1 \sin(q_6) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^9_t\mathbf{T} = \begin{bmatrix} \cos(q_9) & -\sin(q_9) & 0 & l_2 \cos(q_9) \\ \sin(q_9) & \cos(q_9) & 0 & l_2 \sin(q_9) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2_1\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1_3\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & l_3 \cos(\pi/3) \\ 0 & 1 & 0 & l_3 \sin(\pi/3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(23)

Referring to Fig. 3, it is possible to write

$${}^1_t\mathbf{T} = {}^1_4\mathbf{T} \times {}^4_7\mathbf{T} \times {}^7_t\mathbf{T}$$

$${}^2_t\mathbf{T} = {}^2_5\mathbf{T} \times {}^5_8\mathbf{T} \times {}^8_t\mathbf{T}$$

$${}^3_t\mathbf{T} = {}^3_6\mathbf{T} \times {}^6_9\mathbf{T} \times {}^9_t\mathbf{T}$$

(24)

A 3-PRR robot has a total of three kinematics chains: DGPHEBD, EHPIFCE, and ADGPIFA (see Fig. 3). However, only two of them are independent. Therefore two of the chains can be used for obtaining kinematics constraints. In this paper chains DGPHEBD and EHPIFCE are selected for calculating the kinematics constraints. Position of point P, centre of end-effector, with respect to the base reference frame can be calculated in three ways. This position can be

shown by three vectors: \mathbf{Q}_1 , \mathbf{Q}_2 , and \mathbf{Q}_3

$$\begin{aligned}\mathbf{Q}_1 &= \mathbf{AD} + \mathbf{DG} + \mathbf{GP} \\ \mathbf{Q}_2 &= \mathbf{AB} + \mathbf{BE} + \mathbf{EH} + \mathbf{HP} \\ \mathbf{Q}_3 &= \mathbf{AC} + \mathbf{CF} + \mathbf{FI} + \mathbf{IP}\end{aligned}\quad (25)$$

Note that the three vectors must be equal. Therefore

$$\mathbf{Q}_1 = \mathbf{Q}_2 = \mathbf{Q}_3 \quad (26)$$

Additionally, \mathbf{Q}_1 , \mathbf{Q}_2 , and \mathbf{Q}_3 can be obtained from the position vector of the following transformations

$$\begin{aligned}\mathbf{Q}_1 &= \text{Transformation vector of } ({}^1_4\mathbf{T} \times {}^4_7\mathbf{T} \times {}^7_t\mathbf{T}) \\ \mathbf{Q}_2 &= \text{Transformation vector of } ({}^1_2\mathbf{T} \times {}^2_5\mathbf{T} \times {}^5_8\mathbf{T} \times {}^8_t\mathbf{T}) \\ \mathbf{Q}_3 &= \text{Transformation vector of } ({}^1_3\mathbf{T} \times {}^3_6\mathbf{T} \times {}^6_9\mathbf{T} \times {}^9_t\mathbf{T})\end{aligned}\quad (27)$$

Using these position vectors, the constraint equation (2) can be formed. As stated earlier, there are a total of six kinematics constraints. Four of these constraints may be obtained by noting

$$\begin{aligned}\mathbf{Q}_{1x} &= \mathbf{Q}_{2x} \\ \mathbf{Q}_{1y} &= \mathbf{Q}_{2y} \\ \mathbf{Q}_{2x} &= \mathbf{Q}_{3x} \\ \mathbf{Q}_{2y} &= \mathbf{Q}_{3y}\end{aligned}\quad (28)$$

Using Matlab software the four constraint equations, equation (28), can be simplified as

$$\begin{aligned}\varphi_1 &= l_2 \cos(q_4 + q_7) + l_1 \cos(q_4) + q_1 + \frac{l_2}{2} \cos(q_8 + q_5) + \\ &\frac{\sqrt{3}}{2} l_2 \sin(q_8 + q_5) + \frac{l_1}{2} \cos(q_5) + \frac{\sqrt{3}}{2} l_1 \sin(q_5) + \frac{q_2}{2} - l_3 \\ \varphi_2 &= l_2 \sin(q_4 + q_7) + l_1 \sin(q_4) + \frac{l_2}{2} \sin(q_8 + q_5) - \\ &\frac{\sqrt{3}}{2} l_2 \cos(q_8 + q_5) - \frac{\sqrt{3}}{2} l_1 \cos(q_5) + \frac{l_1}{2} \sin(q_5) - \frac{\sqrt{3}}{2} q_2 \\ \varphi_3 &= l_2 \cos(q_4 + q_7) + l_1 \cos(q_4) + q_1 + \frac{l_2}{2} \cos(q_9 + q_6) - \\ &\frac{\sqrt{3}}{2} l_2 \sin(q_9 + q_6) + \frac{l_1}{2} \cos(q_6) - \frac{\sqrt{3}}{2} l_1 \sin(q_6) + \frac{q_3}{2} - \frac{l_3}{2} \\ \varphi_4 &= l_2 \sin(q_4 + q_7) + l_1 \sin(q_4) + \frac{l_2}{2} \sin(q_9 + q_6) + \frac{\sqrt{3}}{2} q_3 \\ &+ \frac{\sqrt{3}}{2} l_2 \cos(q_9 + q_6) + \frac{\sqrt{3}}{2} l_1 \cos(q_6) + \frac{l_1}{2} \sin(q_6) - \frac{\sqrt{3}}{2} l_3\end{aligned}\quad (29)$$

As mentioned in equation (7), it can be deduced that

$$\mathbf{t}_i = \begin{bmatrix} \omega_i \\ \mathbf{v}_i \end{bmatrix} = \mathbf{K}_i \dot{\mathbf{q}} \quad (30)$$

To obtaining \mathbf{K} the twist of each body is written and $k_i (i=1:7)$ is calculated. For example, to calculate k_1 , k_4 , and k_7 , we have

$$\begin{aligned}\omega_1 &= 0, \mathbf{v}_1 = \dot{\mathbf{q}}_1 \\ \omega_4 &= \dot{\mathbf{q}}_4, \mathbf{v}_4 = \dot{\mathbf{q}}_1 + \dot{\mathbf{q}}_4 \times \mathbf{r}_{44} \\ \omega_7 &= \dot{\mathbf{q}}_7, \mathbf{v}_7 = \dot{\mathbf{q}}_1 + \dot{\mathbf{q}}_4 \times \mathbf{r}_{47} + \dot{\mathbf{q}}_7 \times \mathbf{r}_{7t}\end{aligned}\quad (31)$$

Therefore, define matrices $k_i (i=1:7)$ and \mathbf{K} such as:

$$\begin{aligned}k_1 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{e}_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{6 \times 9} \\ k_2 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{e}_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{6 \times 9} \\ k_3 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{e}_3 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{6 \times 9} \\ k_4 &= \begin{bmatrix} 0 & 0 & 0 & \mathbf{e}_4 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{e}_1 & 0 & 0 & \mathbf{e}_4 \times \mathbf{r}_{44} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{6 \times 9} \\ k_5 &= \begin{bmatrix} 0 & 0 & 0 & 0 & \mathbf{e}_5 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{e}_2 & 0 & 0 & \mathbf{e}_5 \times \mathbf{r}_{55} & 0 & 0 & 0 & 0 \end{bmatrix}_{6 \times 9} \\ k_6 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \mathbf{e}_6 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{e}_3 & 0 & 0 & \mathbf{e}_6 \times \mathbf{r}_{66} & 0 & 0 & 0 \end{bmatrix}_{6 \times 9} \\ k_7 &= \begin{bmatrix} 0 & 0 & 0 & \mathbf{e}_4 & 0 & 0 & \mathbf{e}_7 & 0 & 0 \\ \mathbf{e}_1 & 0 & 0 & \mathbf{e}_4 \times \mathbf{r}_{47} & 0 & 0 & \mathbf{e}_7 \times \mathbf{r}_{7t} & 0 & 0 \end{bmatrix}_{6 \times 9} \\ \mathbf{K} &= \begin{bmatrix} \mathbf{k}_1 \\ \mathbf{k}_2 \\ \cdot \\ \cdot \\ \mathbf{k}_7 \end{bmatrix}_{42 \times 9}\end{aligned}\quad (32)$$

where $\mathbf{0}$ represents a three-dimensional zero vector and vector \mathbf{e}_i is

$$\begin{aligned}\mathbf{e}_i &= \begin{bmatrix} \cos\left(2(i-1)\frac{\pi}{3}\right) & \sin\left(2(i-1)\frac{\pi}{3}\right) & 0 \end{bmatrix} & i=1:3 \\ \mathbf{e}_i &= [0 \quad 0 \quad 1] & i=4:9\end{aligned}\quad (33)$$

Additionally, vectors \mathbf{r}_{44} , \mathbf{r}_{47} , \mathbf{r}_{7t} , \mathbf{r}_{55} , and \mathbf{r}_{66} are defined as

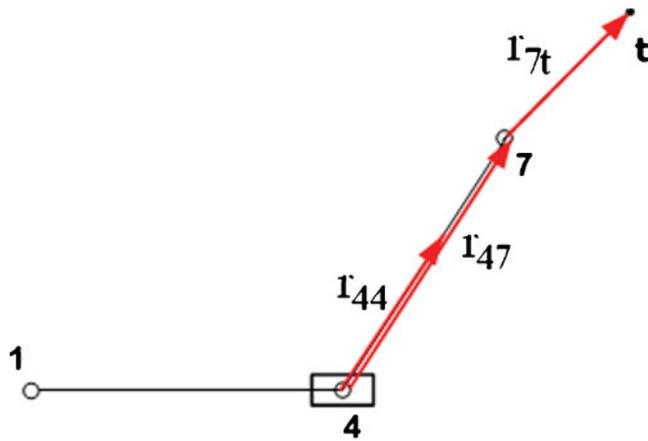


Fig. 4 Vectors r_{ij} for first leg of 3-PRR robot

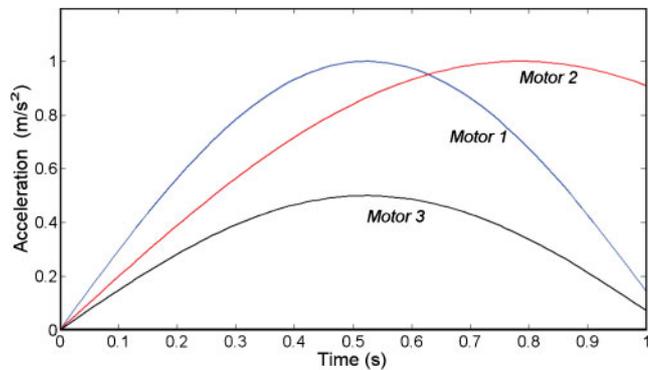
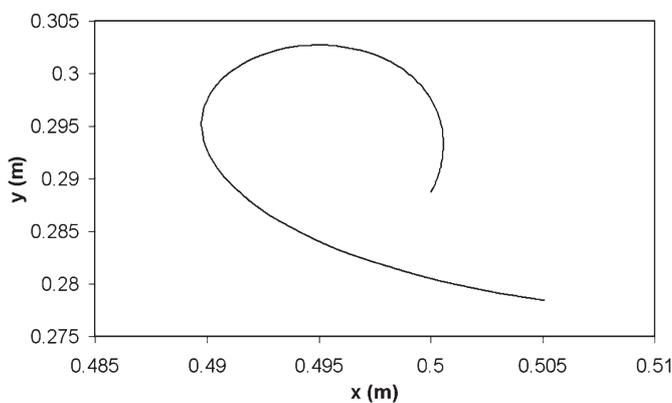


Fig. 5 Motor accelerations

Table 1 Properties of 3-PRR robot

Mass number	m_i (gr)	I_i (gr mm ²)
1	137.2	0.017
2	520.1	0.606
3	1419.5	5.076



$$\begin{aligned}
 r_{44} &= \frac{1}{2} {}^4_7 \mathbf{T}(1:3,4) \\
 r_{47} &= {}^4_7 \mathbf{T}(1:3,4) \quad \text{and} \quad r_{7t} = {}^7_t \mathbf{T}(1:3,4) \\
 r_{55} &= \frac{1}{2} {}^5_8 \mathbf{T}(1:3,4) \quad \text{and} \quad r_{66} = \frac{1}{2} {}^6_9 \mathbf{T}(1:3,4)
 \end{aligned} \tag{34}$$

Note that $\mathbf{T}(1:3,4)$, first three rows of fourth column of \mathbf{T} matrix, denotes the translation vector of the \mathbf{T} matrix. The position vectors used for the first leg are shown in Fig. 4.

It should be noted that selecting of constraints is arbitrary. However, the selected constraints must be independent of each others. Using the Jacobian matrix of constraints, equation (3), is a convenient way to insure independence of constraints. If the determinant of the Jacobian matrix is not equal to zero, then constraints are independent.

In the previous section, four constraint equations were obtained. According to equation (21), two more constraints are needed. As stated earlier, there are a total of three kinematics loops, where any two selected loops are independent. Considering Fig. 3, the two loops DGPHEBD and EHPIFCE can be selected. These two loops each make out a hexagon (not necessary regular). For a hexagon the total internal angles is 4π radians. Therefore, two additional constraints can be defined by writing the summation of internal angles for these two loops

$$\begin{aligned}
 \varphi_5 &= q_4 + q_7 - q_8 - q_5 \\
 \varphi_6 &= q_8 + q_5 - q_9 - q_6
 \end{aligned} \tag{35}$$

The six independent kinematics constraint equations, $\varphi_1 \sim \varphi_6$, can be used to obtain the Jacobian matrix, $\Phi(\mathbf{q})$. Upon obtaining $\Phi(\mathbf{q})$, Simulink toolbox of Matlab is used to calculate matrices \mathbf{M} , \mathbf{C} , and \mathbf{G} .

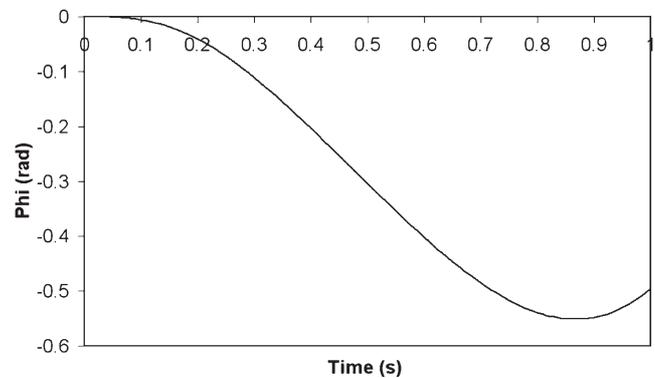


Fig. 6 (Left) Trajectory of point P, (Right) Orientation of moving platform

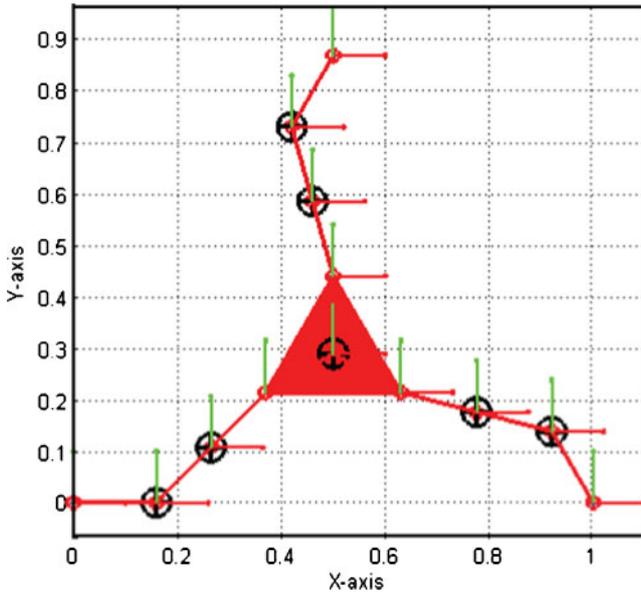


Fig. 7 Initial configuration of the robot

7 NUMERICAL EXAMPLE

Two case studies are presented to verify the theoretical dynamical results obtained in previous sections. In the first case, the robot is assumed to be in the horizontal plane. In the second case, the robot is assumed to be in the vertical plane. The NOC results are compared with two commercial dynamical simulation softwares: SimMechanics toolbox of Matlab and COSMOSMotion of Solidworks. Mass and inertia properties for a 3-PRR robot are defined in Table 1, also

$$l_1 = 0.3 \text{ m}, l_2 = 0.15 \text{ m}, l_3 = 1 \text{ m} \tag{36}$$

7.1 SimMechanics simulation

The robot is simulated in SimMechanics toolbox of Matlab software. Total simulation time is one second. To perform the simulation, physical model of the 3-

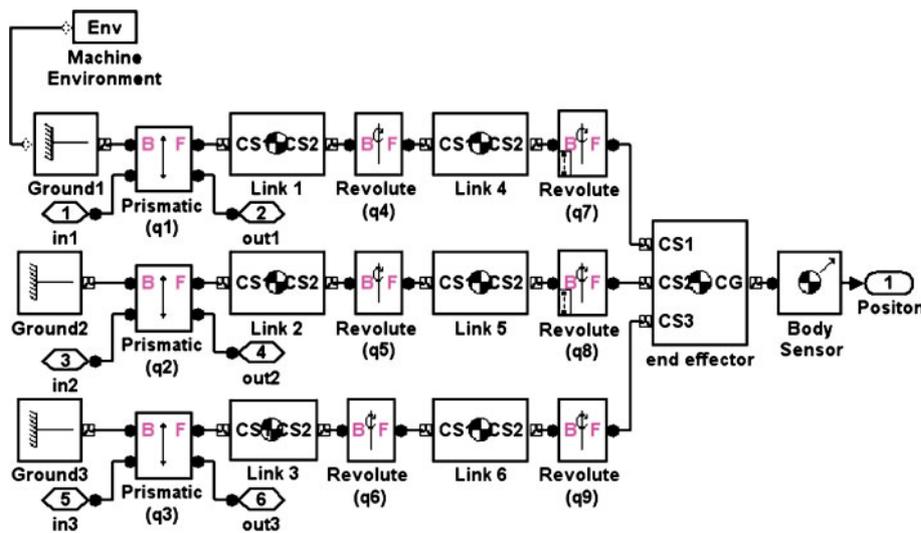


Fig. 8 The 3-PRR robot model in SimMechanics

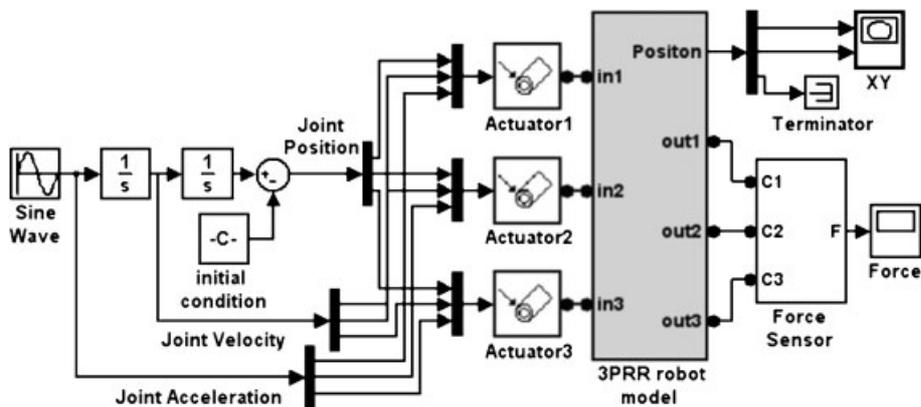


Fig. 9 Inverse dynamic model of 3-PRR robot in SimMechanics

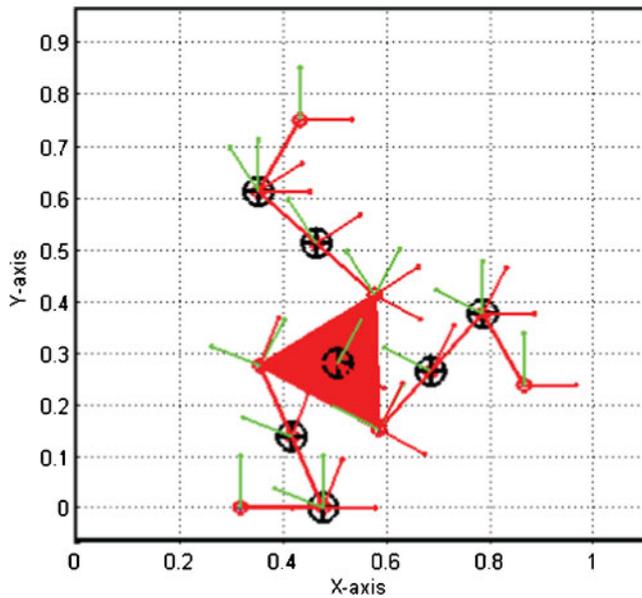


Fig. 10 Final configuration of the robot (after one second)

PRR robot is defined as a subsystem in SimMechanics. The inputs for all three motors are joint trajectories. Accordingly, as shown in Fig. 5, we have

$$a_1 = \sin 3t, \quad a_2 = \sin 2t, \quad a_3 = 0.5 \sin 3t \quad (37)$$

The resulting trajectory for point P, centre of end-effector and orientation of moving platform are shown in Fig. 6.

All three actuators have equal initial condition with zero velocity and 0.1595 m displacement. Then,

$$\begin{aligned} x_1(t=0) &= 0.1595 \text{ m}, & v_1(t=0) &= 0 \\ x_2(t=0) &= 0.1595 \text{ m}, & v_2(t=0) &= 0 \\ x_3(t=0) &= 0.1595 \text{ m}, & v_3(t=0) &= 0 \end{aligned} \quad (38)$$

The initial configuration of the 3-PRR robot is shown in Fig. 7.

The subsystem is shown in Fig. 8. The complete simulation model of the system is shown in Fig. 9. This model represents the inverse dynamics model where the inputs are joint trajectories and the outputs are motor forces. As shown in Fig. 9, three force sensors record the required linear force exerted by the three leadscrews. The final configuration of robot is shown in Fig. 10.

7.2 COSMOSMotion simulation

A second dynamical analysis package is used to verify the theoretical results. Using link properties defined in Table 1, a SolidWorks model is developed, see Fig. 11. The same acceleration input trajectories and COSMOSMotion simulator are used to simulate the motion. The robot is configured in two configurations: horizontal and vertical. Motor forces required to maintain motion are recorded. A snapshot of COSMOSMotion while recording motor

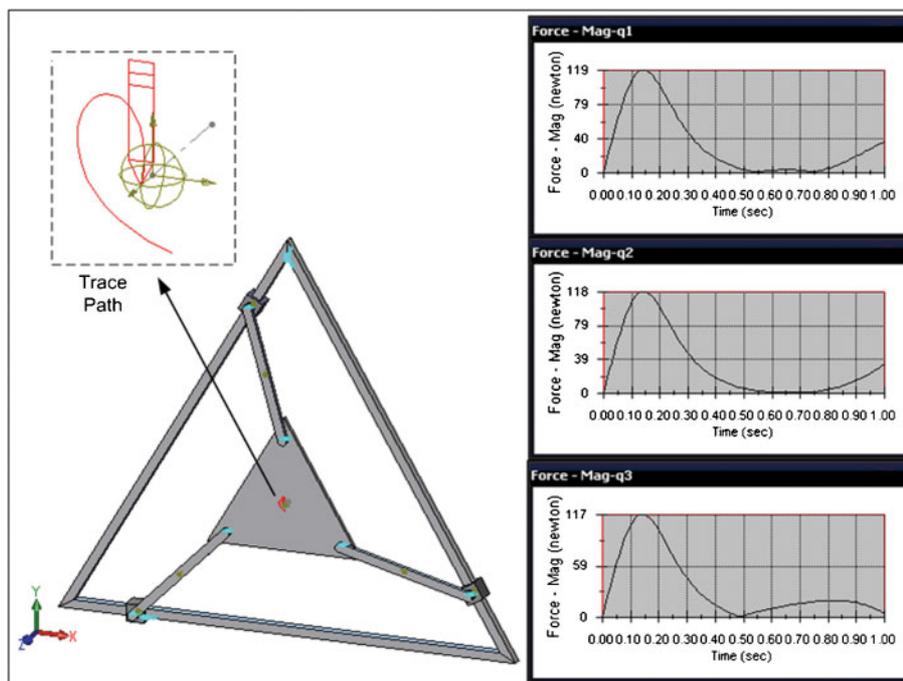


Fig. 11 COSMOSMotion simulation for horizontal configuration

forces (magnitude of each prismatic force) in the horizontal configuration is shown in Fig. 11. Simulation results are presented in the next section.

7.3 Results comparison

The results of the theoretical and the two simulation methods are shown in Fig. 12 for the robot in the

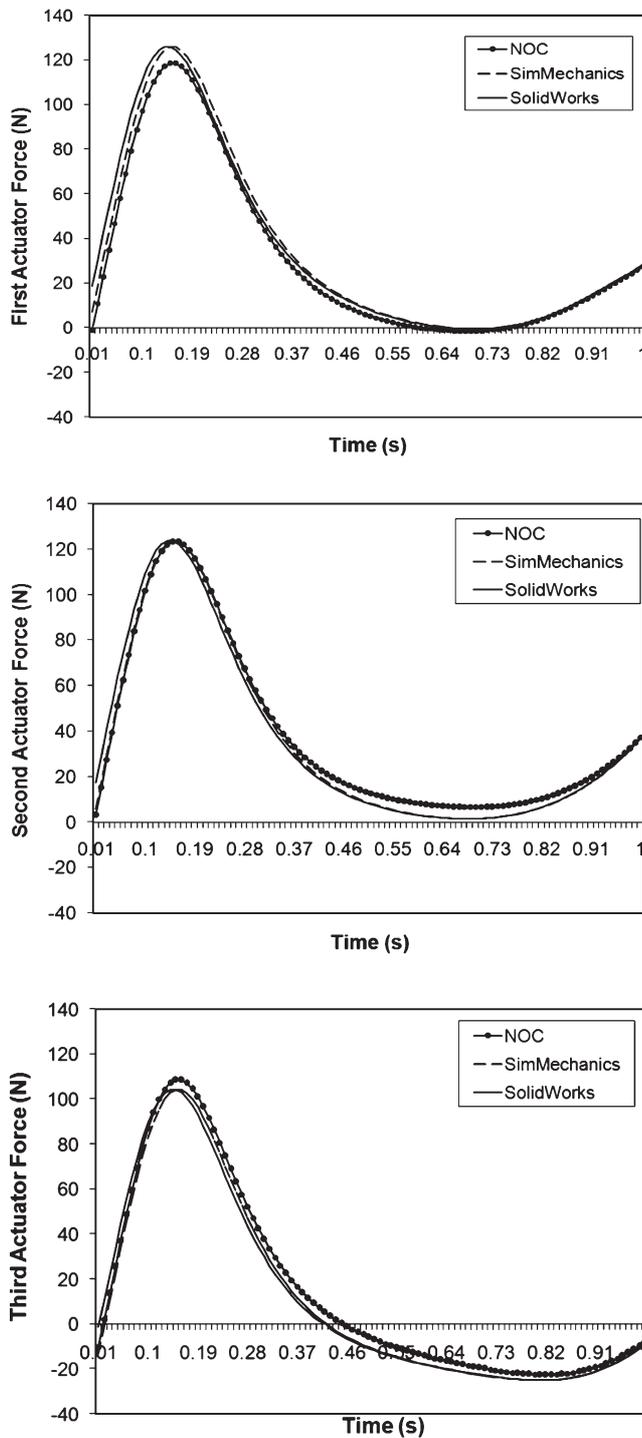


Fig. 12 Forces for vertical configuration

vertical configuration. The next robot is configured in the horizontal plane and simulations are repeated (Fig. 13).

As shown, all results closely follow each other. Therefore, the NOC method is verified.

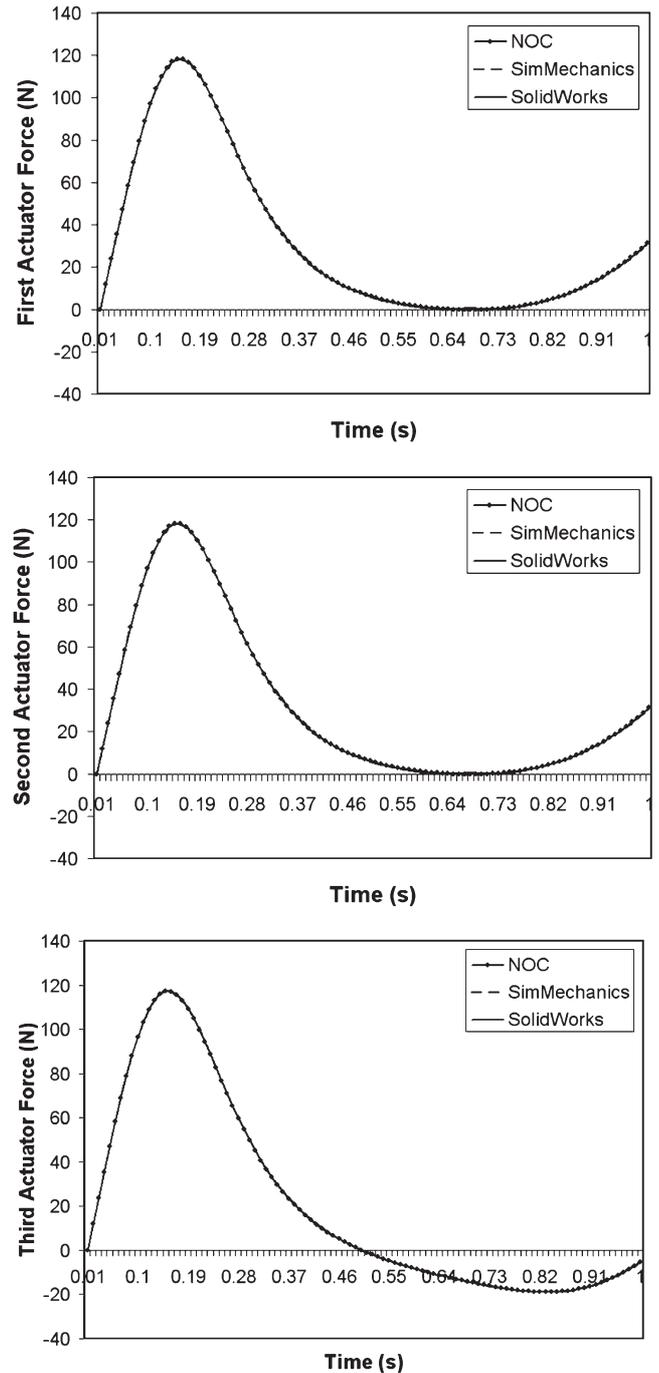


Fig. 13 Forces for horizontal configuration

8 CONCLUSION

The application of NOC for the inverse dynamic analysis of 3-PRR planar parallel manipulators is, for the first time, presented. The method uses joint orthogonal complement which indirectly incorporates the kinematics. Several other advantages of NOC are pointed out. It is shown that using NOC, the Lagrange multipliers and passive joint coordinates are dismissed in the equations. Also the NOC method does not need velocity and acceleration inversions for deriving the dynamics equations. To demonstrate the method, a 3-PRR robot is simulated in both horizontal as well as the vertical configuration. A desired joint trajectory is supplied and required motor forces are calculated using NOC. Additionally a simple model of the 3-PRR is created and simulated in both SimMechanics as well as COSMOSMotion of Solidworks software. Result of the two simulation packages closely agree with results obtained with the NOC formulation.

© Authors 2011

REFERENCES

- 1 **Notash, L.** and **Kamalzadeh, A.** Inverse dynamics of wire-actuated parallel manipulators with a constraining linkage. *Mechanism and Machine Theory*, 2007, **42**, 1103–1118.
- 2 **Liu, M-J., Li, C-X., and Li, C-N.** Dynamics analysis of the Gough–Stewart platform manipulator. *IEEE Trans. Robotics Autom.*, 2000, **16**(1), 94–98.
- 3 **Dasgupta, B.** and **Choudhury, P.** A general strategy based on the Newton–Euler approach for the dynamic formulation of parallel manipulators. *Mechanism and Machine Theory*, 1999, **34**(6), 801–824.
- 4 **Harib, K.** and **Srinivasan, K.** Kinematic and dynamic analysis of Stewart platform-based machine tool structures. *Robotica*, 2003, **21**(5), 541–554.
- 5 **Enferadi, J.** and **Akbarzadeh, A.** Inverse dynamics analysis of a general spherical star-triangle parallel manipulator using principle of virtual work. *Nonlinear dynamics*, 2010, **61**(3), 419–434. DOI: 10.1007/s11071-010-9659-9.
- 6 **Staicu, S.** Power requirement comparison in the 3-RPR planar parallel robot dynamics. *Mechanism and Machine Theory*, 2009, **44**, 1045–1057.
- 7 **Staicu, S.** Inverse dynamics of the 3-PRR planar parallel robot. *Robotics Autonomous Systems*, 2009, **57**, 556–563.
- 8 **Pierrot, F., Dauchez, P., and Fournier, A.** Fast parallel robots. *J. Robotic Systems*, 1991, **8**(6), 829–840.
- 9 **Codourey, A.** and **Burdet, E.** A body oriented method for finding a linear form of the dynamic equations of fully parallel robot. *IEEE International Conference on Robotics and automation*, 1997, Albuquerque, pp. 1612–1618.
- 10 **Do, W. Q. D.** and **Yang, D. C. H.** Inverse dynamic analysis and simulation of a platform type of robot. *J. Robotic Systems*, 1988, **5**(3), 209–227.
- 11 **Angeles, J.** and **Lee, S.** The formulation of dynamical equations of holonomic mechanical systems using a natural orthogonal complement. *ASME J. Appl. Mech.*, 1988, **55**, 243–244.
- 12 **Angeles, J.** and **Ma, O.** Dynamic simulation of n-axis serial robotic manipulators using a natural orthogonal complement. *Int. J. Robotics Res.*, 1988, **7**(5), 32–47.
- 13 **Saha, S. K.** and **Angeles, J.** Dynamics of nonholonomic mechanical systems using a natural orthogonal complement. *ASME Trans. J. Appl. Mech.*, 1991, **58**, 238–243.
- 14 **Xi, F.** and **Sinatra, R.** Inverse dynamics of hexapods using the natural orthogonal complement method. *J. Mfg Systems.*, 2002, **21**(2), 73–82.
- 15 **Saha, S. K.** A decomposition of the manipulator inertia matrix. *IEEE Trans. Robotics and Autom.*, 1997, **13**(2), 301–304.
- 16 **Saha, S. K.** Analytical expression for the inverted inertia matrix of serial robots. *Int. J. Robotic Res.*, 1999, **18**(1), 20–36.
- 17 **Saha, S. K.** and **Schiehlen, W. O.** Recursive kinematics and dynamics for parallel structured closed-loop multibody systems. *Mech. Structs Mach.*, 2001, **29**(2), 143–175.
- 18 **Ma, O.** *Mechanical analysis of parallel manipulators with simulation, design, and control applications*, PhD Thesis, Department of Mechanics, McGill University, Montreal, Quebec, 1991.
- 19 **Craig, J.** *Introduction to robotics, mechanics and control*, third edition, 2004 (Prentice Hall).

APPENDIX

Notation

\mathbf{g}	gravity acceleration vector
\mathbf{I}_i	inertia tensor about mass centre of i th body
$\mathbf{I}_{n \times n}$	$n \times n$ identity matrix
\mathbf{L}	joint orthogonal complement matrix
m	number of one degree of freedom joint
m_i	mass of i th body
n	number of degree of freedom
\mathbf{q}^a	vector of independent joint coordinates
$\dot{\mathbf{q}}^a$	vector of actuated joint velocities
\mathbf{q}^u	vector of dependent joint coordinates

$\dot{\mathbf{q}}^u$	vector of dependent joint velocities	τ^a	generalize actuating force
r	number of rigid body	Φ	Jacobian matrix of the joint constraints
\mathbf{t}_i	twist vector of i th body	ω_i	angular velocity vector of i th body
\mathbf{T}	natural orthogonal complement matrix		
\mathbf{v}_i	translational velocity vector of i th body		