



Solving the dynamic equations of a 3-PRS Parallel Manipulator for efficient model-based designs

M. Díaz-Rodríguez¹, J. A. Carretero², and R. Bautista-Quintero^{2,3}

¹Departamento de Tecnología y Diseño, Facultad de Ingeniería, Universidad de los Andes, Mérida, 5101, Venezuela

²Department of Mechanical Engineering, University of New Brunswick, Fredericton, NB, E3A 5A3, Canada

³Departamento de Ingeniería Mecánica, Instituto Tecnológico de Culiacán, Sinaloa, 80220, Mexico

Correspondence to: M. Díaz-Rodríguez (dmiguel@ula.ve)

Received: 6 December 2014 – Revised: 14 October 2015 – Accepted: 14 December 2015 – Published: 18 January 2016

Abstract. Introduction of parallel manipulator systems for different applications areas has influenced many researchers to develop techniques for obtaining accurate and computational efficient inverse dynamic models. Some subject areas make use of these models, such as, optimal design, parameter identification, model based control and even actuation redundancy approaches. In this context, by revisiting some of the current computationally-efficient solutions for obtaining the inverse dynamic model of parallel manipulators, this paper compares three different methods for inverse dynamic modelling of a general, lower mobility, 3-PRS parallel manipulator. The first method obtains the inverse dynamic model by describing the manipulator as three open kinematic chains. Then, vector-loop closure constraints are introduced for obtaining the relationship between the dynamics of the open kinematic chains (such as a serial robot) and the closed chains (such as a parallel robot). The second method exploits certain characteristics of parallel manipulators such that the platform and the links are considered as independent subsystems. The proposed third method is similar to the second method but it uses a different Jacobian matrix formulation in order to reduce computational complexity. Analysis of these numerical formulations will provide fundamental software support for efficient model-based designs. In addition, computational cost reduction presented in this paper can also be an effective guideline for optimal design of this type of manipulator and for real-time embedded control.

1 Introduction

Seminal research in Parallel Manipulators (PMs) described architectures of 6 degrees of freedom (DOF) which are mainly used to perform industrial tasks. Nevertheless, not all applications (e.g., commercial, space exploration, entertainment or even industrial) require full 6-DOF capabilities, thus, cost-effective PMs with less than 6 DOF (i.e., lower-mobility) have been developed. One such architecture is the 3-PRS manipulator which has a platform and a fixed base connected through three identical sets of links and joints (i.e., legs). Each leg has a slider attached to the base by an actuated prismatic joint (P), a coupler connected to the slider by a passive rotational joint (R) and to the platform by a passive spherical joint (S). The 3-PRS manipulator was first de-

scribed in Carretero et al. (2000b) for a telescope application, and then it was proposed as a machining centre in Fan et al. (2003) and for a medical application in Merlet (2001). The kinematics and workspace analyses of the manipulator have been extensively studied in Carretero et al. (2000a), Carretero et al. (2000b), Tsai et al. (2002), Li and Xu (2007), to name a few. On the other hand, despite the fact that inverse dynamic modelling is essential for optimal design, parameter identification (Mata et al., 2008), model-based control (Díaz-Rodríguez et al., 2013), and internal redundancy (Parsa et al., 2013) amongst others, few papers have focused on revisiting and comparing computationally-inexpensive methods in order to obtain the dynamic models that are cost-effective for real-time applications.

Lagrangian formulations allowed to develop the inverse dynamics model of the 3-PRS manipulator (Li and Xu, 2004). The formulation uses the Lagrange multiplier to include the constraints forces that lead to a modelling approach not only intricate but also computationally complex. Li and Xu (2004) applied the Principle of Virtual Work (PVW), but they simplify the dynamics of the coupler link by dividing its mass into two portions located at its extremes. Tsai and Yuan (2010) solved the inverse dynamic model along with the reaction forces through a special decomposition of the reaction forces at the joints that connect the leg with the platform. A similar approach was used in Yuan and Tsai (2014) for solving direct dynamics including friction effect. However, the later approach considers the calculation of reaction forces which are may be needed for structural design of a manipulator but its computation increases computational complexity which is unnecessary for parameter identification or model-based control. Staicu (2012) analyses and compares the power consumption of the 3-PRS vs. the 3-PRS configuration using the PVW with recursive modelling. The method obtains the Jacobian by differentiating the vector loop equation. Carbonari et al. (2013) solved the inverse dynamics of a 3-DOF parallel manipulator via screw theory and the PVW. On the other hand, the 3-RPS manipulator presents similar characteristics to the 3-PRS manipulator, in this respect, Mata et al. (2008) implement recursive velocity equations used in serial manipulator analysis to find the Jacobian of the manipulator for the inverse dynamic modelling, and Ibrahim and Khalil (2007) exploit architectural characteristics of the 3-RPS to give a closed form solution for the inverse and direct dynamics modelling.

The inherent complexity of the dynamic models lies on the way the system is modelled and how the Jacobian matrix is put forward. In this context, this paper compares the computational number of operation of three formulations for inverse dynamic modelling of a 3-PRS. The first formulation applies the general solution of PMs dynamic modelling proposed in Khalil and Ibrahim (2007). The second method considers the manipulator as a set of open kinematic chains and finds the Jacobian in joint space coordinates by taking into account the vector loop constraints at the split joints (Mata et al., 2008). The third method relies on the modelling approach originally presented in Li and Xu (2004).

The ultimate goal of contrasting these numerical formulations is focused on supporting the implementation of emerging model-based designs which not only depends of the discrete inverse dynamic model but also the numerical finite realization in a given computational platform (Williamson, 1991). In fact, software architecture, for model based control (Díaz-Rodríguez et al., 2013), relies on minimizing computational task timing commonly constrained by fast sampling periods (Goodwin et al., 1992). Similarly, in other applications, such parameter identification (Mata et al., 2008), and internal redundancy (Parsa et al., 2013) few papers have focused on revisiting and comparing computationally-

inexpensive methods in order to obtain the dynamic model of the 3-PRS configuration for cost-effective real-time applications.

To this end, this paper is organized as follows: Sect. 2 presents in general terms how the dynamic model for a parallel manipulator is developed for the three approaches investigated in this paper. Section 3 presents the implementation of the approaches for solving the dynamics problem of a 3-PRS spatial parallel manipulator. Section 4 summarizes and discusses the complexity and the computational load of these three formulations. Finally, the conclusions are drawn.

2 Development of the dynamic models

The dynamic model of a closed chain mechanical system such as a parallel manipulator can be obtained by virtually cutting or splitting the manipulator at one or more of its joints until the complete dynamic model of a tree-like system with several open chains is obtained. Newton–Euler formulations are then used for solving the dynamics of each serial chain. Finally, constraint equations obtained by means of the Lagrange Multipliers are incorporated to include the necessary forces at the splitting points as to ensure the kinematic chains remain closed. On the other hand, the Lagrangian approach can be applied by using the Lagrange equation with respect to a minimum set of generalized coordinates. Yiu et al. (2001) showed that either application of Lagrange Equation or tree-like system analysis are equivalent to one another and lead to the same set of equations when applied to a parallel manipulator. Moreover, similar results were obtained by Murray and Lovell (1989) using the D’Alembert’s principle and the principle of virtual work.

Regardless, of the dynamics equation (Newton–Euler, Lagrange or Principle of Virtual Work) used for developing the model, the dynamics of closed chain system can be written as:

$$\boldsymbol{\tau} = \mathbf{G}^T \mathbf{h}. \quad (1)$$

This equation establishes that the generalized forces corresponding to an open chain system (\mathbf{h}) are related to the actuated forces of a closed chain system ($\boldsymbol{\tau}$) by a linear transformation (\mathbf{G}^T). This linear transformation is based on the Jacobian matrix. Thus, the dynamics equation of a parallel manipulator essentially relies on finding the Jacobian matrix that relate the passive generalized coordinates to the actuated ones.

The following subsections revisit three approaches for solving the dynamics equations of a 3-PRS parallel manipulator. The differences in the approaches are based on two facts: (1) how the parallel manipulator is split into open chains and (2) the type of generalized coordinates used for modelling the manipulator. Each approach leads to establishing different formulations of the Jacobian matrix. Since each Jacobian matrix holds different formulation there are

differences in the number of algebraic operations to compute each. The objective of this work is mostly revisiting these approaches to find the cost-effectiveness of each when solving the inverse dynamics through an example. This is of particular interest as inexpensive computer models are essential when adaptive control algorithms are used to control manipulator at fast update rates. For instance, in the fields of parameter identification and actuation redundancy, a fast estimation of the inverse model implies cost-effectiveness for real-time applications.

2.1 Dynamics considering the legs and platform as subsystems

This approach is based on the general formulation for modelling parallel manipulators presented by Khalil and Ibrahim (2007), which is based on the following aspects: (1) the manipulator is split open at the spherical joints so that the moving platform is separated from the legs and (2) the local joint coordinates systems \mathbf{q} can be used to develop the dynamic equations of each leg \mathbf{h}_i while the Cartesian coordinates \mathbf{x} are used to obtain the dynamic equations of the platform \mathbf{h}_p . Then, the dynamic equations are combined and projected onto the active joint space as follows:

$$\boldsymbol{\tau} = \mathbf{J}_p^T \mathbf{h}_p + \sum_{i=1}^m \begin{pmatrix} \delta \dot{q}_i \\ \delta \dot{q}_a \end{pmatrix}^T \mathbf{h}_i, \quad (2)$$

where \mathbf{J}_p is the Jacobian projecting the task space coordinates (6 in the general case) to the n active joint coordinates, while m is the number of joints for each leg. Likewise,

$$\mathbf{h}_p = \begin{bmatrix} m_p \mathbf{g} - m_p \mathbf{a}_p \\ -\mathbf{I}_p \dot{\boldsymbol{\omega}}_p - \boldsymbol{\omega}_p \times (\mathbf{I}_p \boldsymbol{\omega}_p) \end{bmatrix}, \quad (3)$$

where m_p is the mass of the platform, \mathbf{I}_p denotes the inertial matrix of the moving platform about its centre of mass, \mathbf{a}_p stands for the acceleration of the end effector, and $\boldsymbol{\omega}_p, \dot{\boldsymbol{\omega}}_p$ respectively denote the angular velocity and angular acceleration of the platform.

2.1.1 Method I

In order to develop the model in actuated joint space one has to project the passive joint variables to the active ones. That is:

$$\boldsymbol{\tau} = \mathbf{h}_i^a + \mathbf{J}_p^T \mathbf{h}_p + \mathbf{G}_I^T \mathbf{h}_i^p, \quad (4)$$

where indices a and p stand for the active and passive joints, respectively, while \mathbf{G}_I is a $l \times n$ matrix projecting the dynamics from the passive to the active joints. Here, l represents the number of passive joints while n is the number of active joints on each leg.

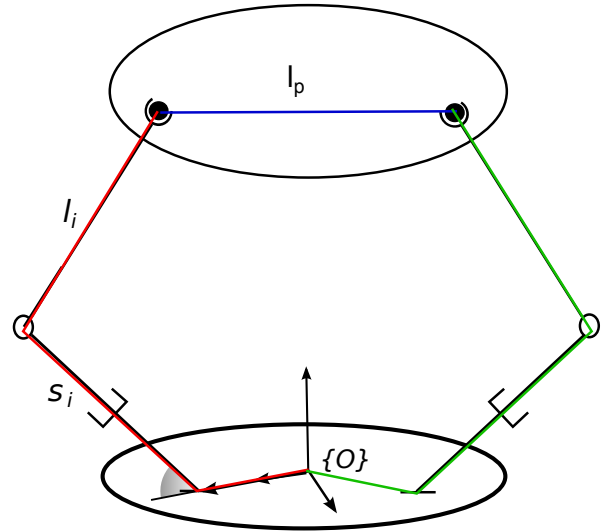


Figure 1. Closed chain equation for finding \mathbf{G}_I .

Equation (4) can be written in the form of Eq. (1). That is:

$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{I} & \mathbf{J}_p^T & \mathbf{G}_I^T \end{bmatrix} \begin{bmatrix} \mathbf{h}_i^a \\ \mathbf{h}_p \\ \mathbf{h}_i^p \end{bmatrix} = \mathbf{G}^T \mathbf{h}, \quad (5)$$

where \mathbf{I} is the identity matrix with dimension equal to the degree of freedom of the manipulator.

In Eq. (4), matrix \mathbf{G}_I can be obtained by considering the fact that the distance among spherical joints at the platform is constant due to the rigid body assumption. This distance is calculated based on the norm of the vector obtained by subtracting the position vector identifying the location of the spherical joints. The partial derivatives of each equation with respect to the joints coordinates yields matrix \mathbf{G}_I . Figure 1 shows how to establish the closed chain equations.

Note that in Eq. (4), \mathbf{h}_p is a 6×1 vector and, in the particular case of a 3-PRS parallel manipulator, $\boldsymbol{\tau}$ is a 3×1 vector. Therefore, matrix \mathbf{J}_p is not square and cannot be obtained using previous methods for solving the inverse kinematics of this kind of manipulator. For instance, when developing the inverse dynamics model using the Principle of Virtual Work, Li and Xu (2004) found a 3×3 square matrix \mathbf{J}_p . They did so by only considering three of the components of the platform inertial forces; they considered those associated with the desired degrees of freedom of the end effector. In Sect. 3 a method obtaining \mathbf{J}_p considering the 6 components of \mathbf{h}_p for the particular case of the 3-PRS manipulator is shown.

2.1.2 Method II

Another approach to formulate the dynamic model is to find the Jacobian matrices according to the approach presented by Khalil and Ibrahim (2007). The method is based on projecting the dynamics equation of the passive joint onto the task

space, and then, project them back to the active joint space so that:

$$\boldsymbol{\tau} = \mathbf{h}_i^a + \mathbf{J}_p^T [\mathbf{h}_p + \mathbf{G}_{II}^T \mathbf{h}_i^p], \quad (6)$$

where \mathbf{G}_{II} is a $l \times 6$ matrix that holds new definition that can be written as follow:

$$\mathbf{G}_{II}^T = \mathbf{J}_{v_i}^T \mathbf{J}_{q_i}^{-T}, \quad (7)$$

where \mathbf{J}_{v_i} and $\mathbf{J}_{q_i}^{-T}$ can be obtained respectively from the direct Jacobian \mathbf{J}_x and the inverse Jacobian \mathbf{J}_q of the manipulator.

Equation (6) can be written in the form of Eq. (1). That is:

$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{I} & \mathbf{J}_p^T & (\mathbf{G}_{II} \mathbf{J}_p)^T \end{bmatrix} \begin{bmatrix} \mathbf{h}_i^a \\ \mathbf{h}_p \\ \mathbf{h}_i^p \end{bmatrix} = \mathbf{G}^T \mathbf{h}. \quad (8)$$

2.2 Dynamics considering the manipulator as open kinematic chains

A parallel manipulator can be split open into $m - 1$ joints yielding m open chain systems. In this approach the platform is attached to one of the legs, see Fig. 2. Algorithms for obtaining the dynamics model of serial manipulators may now be applied to obtain the dynamics of each leg.

The cut joints introduce constraint forces, which can be included into the model by means of the Lagrange multipliers:

$$\boldsymbol{\tau} = \mathbf{h}_i + \mathbf{A} \boldsymbol{\lambda}_i, \quad (9)$$

where \mathbf{A} is the Jacobian that can be found by analysing the constraint equations. Mata et al. (2008) presented a method for the Jacobian matrix by considering the linear velocity at the split joints. The velocities can be computed through the Jacobian analysis of each leg. Then, the velocity obtained at the split joint following each leg are the same. In this approach, recursive modelling of the velocity analysis from conventional serial manipulator methods can be applied for each leg.

Once the Jacobian matrix is found, the Lagrange multipliers in Eq. (9) are eliminated by multiplying matrix \mathbf{C} , so that, $\mathbf{C}\mathbf{A} = 0$. One way to find \mathbf{C} is by obtaining the natural orthogonal complement of \mathbf{C} . On the other hand, the matrix can be found by separating matrix $\mathbf{A} = 0$ into the passive and the active joints. That is:

$$\boldsymbol{\tau} = \mathbf{h}^a + \mathbf{G}_{III}^T \mathbf{h}^p, \quad (10)$$

where \mathbf{G}_{III} has dimensions $l \times n$ and can be computed as follows:

$$\mathbf{G}_{III} = \mathbf{A}_p^{-1} \mathbf{A}_a, \quad (11)$$

where subscripts p and a respectively refer to the passive and active variable terms in Jacobian \mathbf{A} . Equation (10) can be

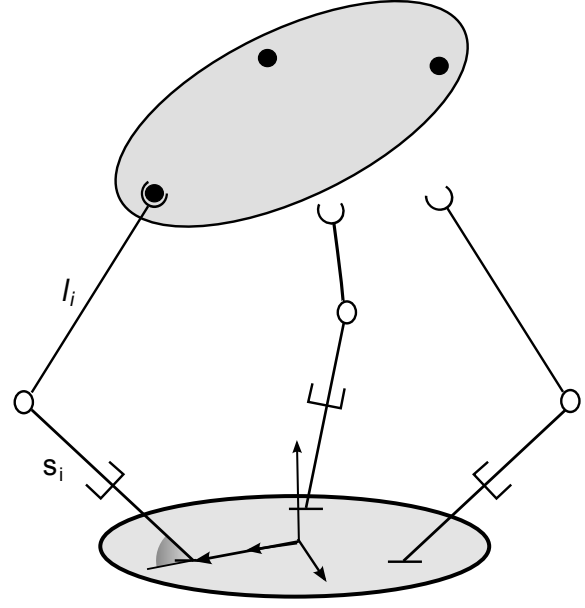


Figure 2. Several open chains obtained after cutting open the parallel manipulator.

written in the form of Eq. (1). That is:

$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{I} & \mathbf{G}_{III} \end{bmatrix} \begin{bmatrix} \mathbf{h}_i^a \\ \mathbf{h}_i^p \end{bmatrix} = \mathbf{G}^T \mathbf{h}. \quad (12)$$

As Eqs. (5), (8) and (12) show, the three methods establish a linear relation between the cut-open model (\mathbf{h}) to the original system. The different lies in how matrix \mathbf{G}^T and vector \mathbf{h} are solved in each model. This is illustrated in the next section for the particular case of a general 3-PRS parallel manipulator.

3 Inverse dynamics of the 3-PRS PM

A schematic representation of the 3-PRS PM is shown in Fig. 3 where the 7 moving rigid bodies are shown. There, links 0, 1, and 2 can be seen as 2-DOF serial manipulator with PR joints, also it applies to links 0, 4, and 5 as well as links 0, 6, and 7. The platform is indicated by the number 3. The manipulator is a lower mobility (i.e., less than 6-DOF) spacial PM with 3-DOF. This manipulator holds the characteristic of zero torsion at its platform because the three spherical joints move in vertical planes intersecting at a common line (Liu and Bonev, 2008). In addition, the topology of its legs provides 2-DOF of angular rotation (2R) in two axes (A/B axis, rolling and pitching) and 1-DOF translation (1T) motion (heave) at the end effector.

As Fig. 3 shows, reference frame $\{O\}$ defines the global coordinate system (x, y, z) while $\{p\}$ defines the local coordinate system attached to the moving platform. In the typical symmetrical configuration, the centre of the spherical joints

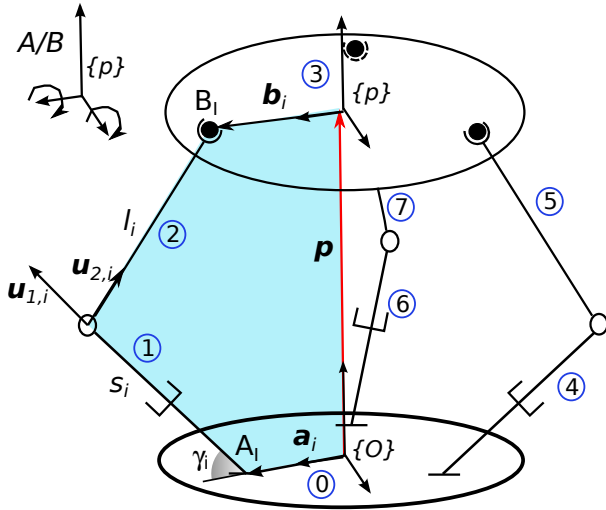


Figure 3. Sketch of a general 3-PRS manipulator.

at the platform form an equilateral triangle circumscribed in a circle with centre p and radius r_p . The line of action of the prismatic joints intersects the base Oxy plane at A_i also forming an equilateral triangle. The distance from O to A_i is the base platform radius r_b .

In order to take advantage of the dynamic equation algorithms developed for serial manipulators, when modelling the manipulator with leg and platform as subsystem, the joint coordinates \mathbf{q} can be used to develop the dynamics equations of each leg (Mata et al., 2002). To this end, a local coordinate system $\{O_{i,0}\}$ is defined at the bottom of leg i where the leg meets the based plane. Table 1 lists the Denavit–Hartenberg (D–H) parameters, according to Craig’s notation (Craig, 2005), of each leg i from $\{O_{i,0}\}$ up to the location of the axis $\{O_{i,2}\}$ at the revolute joint.

The rotation matrix ${}^{O_{i,0}}\mathbf{R}_O$ can be found as:

$${}^{O_{i,0}}\mathbf{R}_O = \begin{bmatrix} -\sin(\xi_i) & 0 & \cos(\xi_i) \\ \cos(\xi_i) & 0 & \sin(\xi_i) \\ 0 & 1 & 0 \end{bmatrix}, \quad (13)$$

where $\xi = [0 \quad 2/3\pi \quad 4/3\pi]^T$.

In addition, the roll-pitch-yaw (α , β and ϕ) Euler angles represent the orientation of the moving frame $\{p\}$ with respect to the global coordinate system $\{O\}$. The rotation matrix ${}^O\mathbf{R}_p$ is defined as:

$${}^O\mathbf{R}_p = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\phi - s_\alpha c_\phi & c_\alpha s_\beta c_\phi + s_\alpha s_\phi \\ s_\alpha c_\beta & s_\alpha s_\beta s_\phi + c_\alpha c_\phi & s_\alpha s_\beta c_\phi - c_\alpha s_\phi \\ -s_\beta & c_\beta s_\phi & c_\beta c_\phi \end{bmatrix}, \quad (14)$$

where $c_* = \cos(*)$ and $s_* = \sin(*)$.

Table 1. D–H parameters for a 3-PRS PM when modelling the manipulator with leg and platform as subsystem.

i	$\theta_{i,1}$	$d_{i,1}$	$a_{i,1}$	$\alpha_{i,1}$	$\theta_{i,2}$	$d_{i,2}$	$a_{i,2}$	$\alpha_{i,2}$
1	$\pi/2$	s_1	0	γ_1	θ_1	0	0	$-\pi/2$
2	$\pi/2$	s_2	0	γ_2	θ_2	0	0	$-\pi/2$
3	$\pi/2$	s_3	0	γ_3	θ_3	0	0	$-\pi/2$

The task space (\mathbf{x}) and joint space (\mathbf{q}_i) coordinates are given by:

$$\mathbf{x} = [x_p \quad y_p \quad z_p \quad \phi \quad \beta \quad \alpha]^T, \quad (15)$$

$$\mathbf{q} = [\mathbf{q}_1^T \quad \mathbf{q}_2^T \quad \mathbf{q}_3^T]^T, \quad (16)$$

$$\mathbf{q}_1 = \begin{bmatrix} s_1 \\ \theta_1 \end{bmatrix}, \quad \mathbf{q}_2 = \begin{bmatrix} s_2 \\ \theta_2 \end{bmatrix}, \quad \text{and} \quad \mathbf{q}_3 = \begin{bmatrix} s_3 \\ \theta_3 \end{bmatrix}, \quad (17)$$

where s_i represents the displacement along the axis of the prismatic joint i , and θ_i the angle of the link 2 and the axis of the prismatic joint i in the plane of movement of the leg i .

The components of \mathbf{b}_i with respect to the local coordinate system $\{p\}$ are given by:

$${}^p\mathbf{b}_1 = \begin{bmatrix} r_p \\ 0 \\ 0 \end{bmatrix}, \quad {}^p\mathbf{b}_2 = \begin{bmatrix} -\frac{1}{2}r_p \\ \frac{\sqrt{3}}{2}r_p \\ 0 \end{bmatrix}, \quad \text{and} \quad {}^p\mathbf{b}_3 = \begin{bmatrix} -\frac{1}{2}r_p \\ -\frac{\sqrt{3}}{2}r_p \\ 0 \end{bmatrix} \quad (18)$$

while the components of \mathbf{a}_i with respect to the global coordinate system $\{O\}$ are given by:

$$\mathbf{a}_1 = \begin{bmatrix} r_b \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{a}_2 = \begin{bmatrix} -\frac{1}{2}r_b \\ \frac{\sqrt{3}}{2}r_b \\ 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{a}_3 = \begin{bmatrix} -\frac{1}{2}r_b \\ -\frac{\sqrt{3}}{2}r_b \\ 0 \end{bmatrix}. \quad (19)$$

The position problem for the considered PM is not included in this paper since it can be found in Carretero et al. (2000b), Tsai et al. (2002), Li and Xu (2004), Mata et al. (2008). The following subsections focus on the computation of the Jacobian matrix for the aforementioned approaches.

3.1 Jacobian matrix for Model I

Matrices \mathbf{J}_p and \mathbf{G} for computing Eq. (4) are found following the approach presented in Li and Xu (2004). The vector loop equation of the i th leg can be written as:

$$\mathbf{p} + \mathbf{b}_i = \mathbf{a}_i + s_i \mathbf{u}_{1,i} + l_i \mathbf{u}_{2,i}, \quad (20)$$

where $\mathbf{u}_{1,i}$ and $\mathbf{u}_{2,i}$ are unit vectors, l_i is the distance between the rotational joint and the spherical joint, \mathbf{b}_i is the position vector from the origin O to the centre of the spherical joint. Vector \mathbf{a}_i is the position vector between O and local frame of

each leg. Differentiating Eq. (20) with respect to time and after some algebraic manipulation, the following equation can be obtained:

$$\mathbf{J}_q \dot{\mathbf{q}} = \mathbf{J}_x \dot{\mathbf{x}} \quad (21)$$

where:

$$\mathbf{J}_q = \begin{bmatrix} \mathbf{u}_{21}^T \mathbf{u}_{11} & 0 & 0 \\ 0 & \mathbf{u}_{22}^T \mathbf{u}_{12} & 0 \\ 0 & 0 & \mathbf{u}_{23}^T \mathbf{u}_{13} \end{bmatrix} \quad (22)$$

and

$$\mathbf{J}_x = \begin{bmatrix} \mathbf{u}_{21}^T & (\mathbf{b}_1 \times \mathbf{u}_{21})^T \\ \mathbf{u}_{22}^T & (\mathbf{b}_2 \times \mathbf{u}_{22})^T \\ \mathbf{u}_{23}^T & (\mathbf{b}_3 \times \mathbf{u}_{23})^T \end{bmatrix}. \quad (23)$$

Due to the constraints imposed by the fact that each legs moves on a plane, the following set of equations holds:

$$\begin{aligned} x_p &= -r_p s_\alpha c_\beta, \\ y_p &= -\frac{1}{2} r_p (c_\alpha c_\beta - s_\alpha s_\beta s_\phi - c_\alpha c_\phi), \\ \tan(\alpha) &= (s_\beta s_\phi) / (c_\phi + c_\beta). \end{aligned} \quad (24)$$

From these equations, a 6×3 Jacobian matrix mapping the dependent task space coordinates to the independent ones can be found such that:

$$\begin{bmatrix} \dot{x}_p & \dot{y}_p & \dot{z}_p & \dot{\phi} & \dot{\beta} & \dot{\alpha} \end{bmatrix}^T = \mathbf{J}_r^* \begin{bmatrix} \dot{z}_p & \dot{\phi} & \dot{\beta} \end{bmatrix}^T \quad (25)$$

It is important to note that $\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_p & \dot{y}_p & \dot{z}_p & \omega_{p_x} & \omega_{p_y} & \omega_{p_z} \end{bmatrix}^T$. In order to apply Eq. (25), one has to find the angular velocity of the platform through the rate of change of the generalised coordinates $\begin{bmatrix} \dot{\alpha} & \dot{\beta} & \dot{\phi} \end{bmatrix}^T$. That is:

$$\begin{bmatrix} \omega_{p_x} \\ \omega_{p_y} \\ \omega_{p_z} \end{bmatrix} = \begin{bmatrix} c_\alpha c_\beta & -s_\alpha & 0 \\ s_\alpha c_\beta & c_\alpha & 0 \\ -s_\beta & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix}. \quad (26)$$

After some algebraic manipulation the following equation can be obtained:

$$\mathbf{J}_p = \mathbf{J}_r \mathbf{J}_c^{-1} \quad (27)$$

where $\mathbf{J}_c = \begin{bmatrix} \mathbf{J}_q^{-1} \mathbf{J}_x \mathbf{J}_r \end{bmatrix}$.

The 3×3 matrix \mathbf{G}_1 is found by considering the fact that the distance among spherical joints at the platform is constant due to the rigid body assumption. That is:

$$\begin{aligned} l_p^2 - \|\mathbf{a}_i + s_i \mathbf{u}_{1i} + l_i \mathbf{u}_{2i} - \mathbf{a}_{i+1} - s_{i+1} \mathbf{u}_{1i+1} \\ - l_{i+1} \mathbf{u}_{2i+1}\|^2 = 0 \end{aligned} \quad (28)$$

with $i = 1, 2, 3$ and when $i = 3, i + 1 = 1$.

Thus, by obtaining the partial derivatives of the above set of equation matrix \mathbf{G}_1 can be written as:

$$\begin{bmatrix} \frac{\delta \dot{\theta}_1}{\delta \dot{s}_1} & \frac{\delta \dot{\theta}_2}{\delta \dot{s}_1} & \frac{\delta \dot{\theta}_3}{\delta \dot{s}_1} \\ \frac{\delta \dot{\theta}_1}{\delta \dot{s}_2} & \frac{\delta \dot{\theta}_2}{\delta \dot{s}_2} & \frac{\delta \dot{\theta}_3}{\delta \dot{s}_2} \\ \frac{\delta \dot{\theta}_1}{\delta \dot{s}_3} & \frac{\delta \dot{\theta}_2}{\delta \dot{s}_3} & \frac{\delta \dot{\theta}_3}{\delta \dot{s}_3} \end{bmatrix} = \mathbf{X}_p^{-1} \mathbf{X}_a = \mathbf{G}_1. \quad (29)$$

3.2 Jacobian matrix for Model II

Another approach to compute matrices \mathbf{J}_p and \mathbf{G} is to consider that the spherical joints in each leg is constrained to move on a plane normal to the revolute joint. The motion of each leg at point B_i can be found in terms of the joint coordinates. Moreover, it can also be expressed with respect to $O_{i,2}$. Due to the constraints provided by the P-R pair, the third row of the Jacobian matrix consist of zero entries. That is:

$${}^{i,2} \mathbf{v}_{B_i} = {}^i \mathbf{J}_q \mathbf{q}_i, \quad (30)$$

where

$${}^i \mathbf{J}_q = \begin{bmatrix} -\sin(\theta_i) & 0 \\ -\cos(\theta_i) & l_i \\ 0 & 0 \end{bmatrix}. \quad (31)$$

The linear velocity of the end effector can be related to the linear velocity of points B_i as follows:

$${}^{i,2} \mathbf{v}_{B_i} = {}^i \mathbf{J}_v \mathbf{v} = \begin{bmatrix} {}^{i,2} \mathbf{R}_O & -{}^{i,2} \mathbf{R}_O {}^O \mathbf{R}_p {}^p \tilde{\mathbf{b}}_i \end{bmatrix} \mathbf{v}, \quad (32)$$

where

$\mathbf{v} = \begin{bmatrix} v_p^T & \omega_p^T \end{bmatrix}^T = \begin{bmatrix} v_{p_x} & v_{p_y} & v_{p_z} & \omega_x & \omega_y & \omega_z \end{bmatrix}^T$, $\tilde{\mathbf{b}}$ stands for the skew symmetric matrix substituting the cross product $\mathbf{b}_i \times$, and

$${}^{i,2} \mathbf{R}_p = \begin{bmatrix} -c_{\sigma_i} s_{\gamma_i + \theta_i} & -c_{\sigma_i} c_{\gamma_i + \theta_i} & s_{\sigma_i} \\ -s_{\sigma_i} s_{\gamma_i + \theta_i} & -s_{\sigma_i} c_{\gamma_i + \theta_i} & -c_{\sigma_i} \\ c_{\gamma_i + \theta_i} & -s_{\gamma_i + \theta_i} & 0 \end{bmatrix}. \quad (33)$$

The first two rows of Eqs. (31) and (32) relate the task space to the joint spaces coordinates. That is:

$$\mathbf{J}_q \dot{\mathbf{q}} = \mathbf{J}_x \mathbf{v}, \quad (34)$$

where

$$\mathbf{J}_q = \begin{bmatrix} {}^1 \mathbf{J}_q & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}^2 \mathbf{J}_q & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & {}^3 \mathbf{J}_q \end{bmatrix} \quad (35)$$

Table 2. D–H Parameters for the first leg of the 3-PRS PM when modelling as a three open chains.

j	$\theta_{1,j}$	$d_{1,1}$	$a_{1,j}$	$\alpha_{1,j}$
1	$\pi/2$	s_1	0	$-\gamma_1$
2	θ_1	0	0	$-\pi/2$
3	θ_4	l_1	0	0
4	θ_5	0	0	$\pi/2$
5	θ_6	0	0	$\pi/2$

and,

$$\mathbf{J}_x = \begin{bmatrix} \mathbf{x}^{T1,2} \mathbf{R}_O & \mathbf{x}^T (-{}^{1,2}\mathbf{R}_O^O \mathbf{R}_p^p \tilde{b}_1) \\ \mathbf{y}^{T1,2} \mathbf{R}_O & \mathbf{y}^T (-{}^{1,2}\mathbf{R}_O^O \mathbf{R}_p^p \tilde{b}_1) \\ \mathbf{x}^{T2,2} \mathbf{R}_O & \mathbf{x}^T (-{}^{2,2}\mathbf{R}_O^O \mathbf{R}_p^p \tilde{b}_1) \\ \mathbf{y}^{T2,2} \mathbf{R}_O & \mathbf{y}^T (-{}^{2,2}\mathbf{R}_O^O \mathbf{R}_p^p \tilde{b}_1) \\ \mathbf{x}^{T3,2} \mathbf{R}_O & \mathbf{x}^T (-{}^{3,2}\mathbf{R}_O^O \mathbf{R}_p^p \tilde{b}_1) \\ \mathbf{y}^{T3,2} \mathbf{R}_O & \mathbf{y}^T (-{}^{3,2}\mathbf{R}_O^O \mathbf{R}_p^p \tilde{b}_1) \end{bmatrix}. \quad (36)$$

In the above equations $\mathbf{x} = [1 \ 0 \ 0]^T$, $\mathbf{y} = [0 \ 1 \ 0]^T$, and $\mathbf{0}$ is a 2×2 zero matrix.

From the above equation, matrix \mathbf{G}_{II} can be obtained as follows:

$$\mathbf{J}_q^{-1} \mathbf{J}_x = \begin{bmatrix} \mathbf{J}_{q1}^{-1} \mathbf{J}_{v1} \\ \mathbf{J}_{q2}^{-1} \mathbf{J}_{v2} \\ \mathbf{J}_{q3}^{-1} \mathbf{J}_{v2} \end{bmatrix} \quad (37)$$

$$\mathbf{G}_{II}^T = \begin{bmatrix} \mathbf{J}_{v1}^T \mathbf{J}_{q1}^{-T}(:,2) & \mathbf{J}_{v2}^T \mathbf{J}_{q1}^{-T}(:,2) & \mathbf{J}_{v3}^T \mathbf{J}_{q3}^{-T}(:,2) \end{bmatrix} \quad (38)$$

where $\mathbf{A}(:,2)$ denotes the 2nd column of matrix \mathbf{A} .

The Jacobian matrix, relating the task space coordinates \mathbf{J}_r , is found by considering the third column of velocity equation following each leg and the platform. The Jacobian matrix is obtained by following method 2 which is graphically represented in Fig. 4.

3.3 Jacobian matrix for Model III

The inverse dynamic is computed as a function of three open chains which are obtained after disassembling two of the three spherical joints. The platform is attached to one of the legs, and the spherical joint is modelled as three intersecting revolute joints with the three axes mutually perpendicular to one another. Therefore, the chain with the end effector platform is modelled by using the set of D–H parameters presented in Table 2. The remaining legs have only sets of two variables which are the same as those presented in Table 1.

One of the advantages of considering the manipulator as tree-like serial chains is that the Jacobian for each leg can

be computed by using well-known recursive modelling from serial manipulator. In this respect, the velocities at the cut joints can be computed through recursive formulation (Angeles, 2002). That is:

$$\mathbf{A}_i \dot{\mathbf{q}}_i = V_{B_{i,i+1}} = \mathbf{A}_{i+1} \dot{\mathbf{q}}_{i+1}, \quad (39)$$

where $i = 1, 2, 3$ and when $i = 3, i+1 = 1$, \mathbf{A}_i is the Jacobian matrix for the i th leg, and $V_{B_{i,i+1}}$ is the velocity at the cut joint connecting leg i and $i+1$.

From Eq. (39) the following equation can be obtained:

$$\mathbf{A}_i \dot{\mathbf{q}}_i - \mathbf{A}_{i+1} \dot{\mathbf{q}}_{i+1} = 0. \quad (40)$$

This equation provides a set of three linear systems relating the joint coordinates following each loop. If the set of linear equations is appended together the following equation holds:

$$\mathbf{X} = \begin{bmatrix} \mathbf{A}_1 & -\mathbf{A}_2 & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{A}_2 & -\mathbf{A}_3 \\ \mathbf{A}_1 & \mathbf{0}_{3 \times 3} & -\mathbf{A}_3 \end{bmatrix} \dot{\mathbf{q}} = 0. \quad (41)$$

The relationship between the active and passive coordinates can be obtained from Eq. (41) as follows:

$$\mathbf{G}_{III} = \mathbf{X}_p^{-1} \mathbf{X}_a. \quad (42)$$

4 Results and discussion

In order to solve the inverse dynamics of the 3-PRS parallel manipulator, each term of Eqs. (4), (6), and (10) are found in closed form by using a Computer Algebra Symbolic (CAS) program, such as Maple. One of the advantages of developing the model in a CAS program relies on the fact that the mathematical operations can be performed symbolically and simplified. In the present case, built-in functions such as `simplify` and `combine` (with `option=trig`) of Maple programming environment were used to reduce the number of operations for solving each model.

A second advantage of obtaining the model in closed form is that the code can be written automatically for Matlab by using the code generation capabilities of the software. The Matlab procedure with `optimize=tryhard` option of the package `CodeGeneration` were used in this case to develop Matlab code. The number of algebraic operations (i.e., additions/subtractions and divisions/multiplications) necessary for solving the dynamic problem was obtained through the `cost` function of the `codegen` package. Without any loss of generality, the number of operations for matrix inversion (i.e., when obtaining \mathbf{J}_p) were computed by considering the number of operations conventional LU decomposition takes to solve a linear system: about $+$, $-n^3/3 - n^2/2 + 5n/6$ and \times $n^3/3 + n^2 - n/3$ (Chapra and Canale, 2006). The objective of this paper is to evaluate the complexity and the computational load of these three formulations which is presented in Table 3.

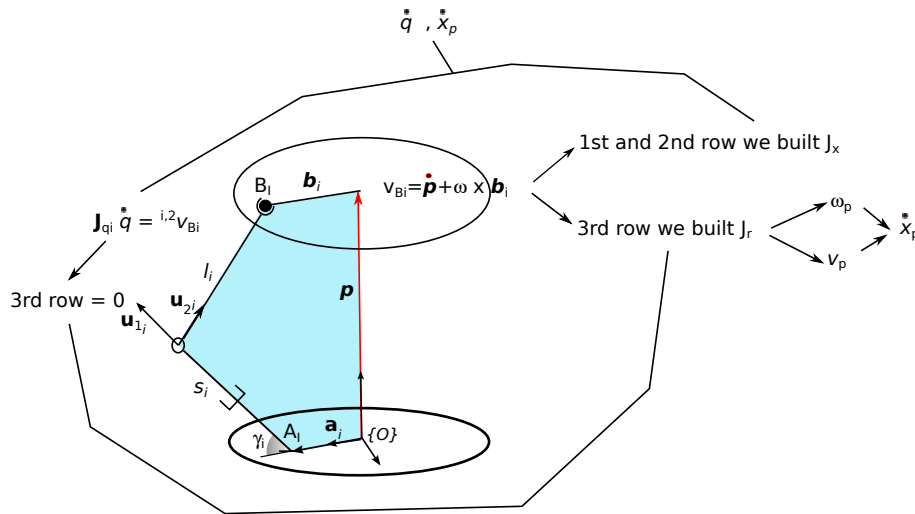


Figure 4. Formulation of the Jacobian matrix using Method II.

Table 3. Computational effort for solving the inverse dynamic problem.

Model Eq. (6)			Model Eq. (4)			Model Eq. (10)		
Term	×/÷	+/-	Term	×/÷	+/-	Term	×/÷	+/-
h^a	21	15	h^a	21	15	h^a	97	67
h^p	27	18	h^p	27	18	h^p	600	380
h_p	404	253	h_p	404	253	h_p	150	56
X_p	–	–	X_p	6	12	X_p	0	0
X_a	–	–	X_a	31	27	X_a	478	308
G_{II}	76	17	G_I	109	72	G_{III}	–	–
$J_q^{-1}J_x$	90	50	$J_q^{-1}J_x$	70	28			
J_r	52	26	J_r	52	26			
J_c	196	121	J_c	196	113			
J_p	265	226	J_p	245	204			
$G^T h$	832	568	$G^T h$	833	574	$G^T h$	1193	773

As Table 3 shows, the models in Eqs. (4) and (6), which are based on splitting the platform from the legs, hold fewer number of operation than those of the model considering the platform attached to one of the legs. This fact is due to the inversion of a 6×6 matrix when finding the matrix G_{III} . On the other hand, by having the platform attached to one leg, the projection of the platform generalised forces onto the actuated joints is cumbersome. The result shows that either using Eqs. (4) or (6) a reduction of about 30 % in the number of multiplication and about 25 % in additions is obtained when comparing to the model given in Eq. (10). These results indicate that considering the platform and legs as subsystem can improve speed in the calculation of dynamics for applications such as model-based control, see for example Díaz-Rodríguez et al. (2013).

5 Conclusions

Three strategies for dynamic modelling of parallel manipulators were revisited and applied for developing the inverse dynamic model of a general 3-PRS parallel manipulator. The first method exploited parallel manipulators particularities such that the platform and the links are considered as subsystems, namely: the legs and the platform, thus, joint coordinates were used for modelling the legs dynamics and task space coordinates for the platform. The core concept of the second method is based on solving the dynamic model through this approach reduces to how the Jacobian matrix relating the task and the joint space coordinates is computed and it was explained in this paper. A third method was presented which is similar to the second method but uses a different formulation of the Jacobian matrix. In the third approach the dynamic model was obtained by describing the manipulator as three open kinematic chains. The

vector-loop closure constraints introduced the relationship between the dynamics of the open kinematic chains and the original closed chains. A Computer Algebraic Software allowed to find each term of the dynamic in symbolic form and to compute the computational burden of each model.

The results showed that the approaches based on splitting the manipulator in two sub-systems (platform and legs) require about 30 % in the number of multiplication and about 25 % in additions are obtained when comparing to the model given by Eq. (10). These results indicate that in problems when the model is needed to be computed on-line at high rates of speed, method 1 and 2 can be useful. This work has provided numerical guidelines for implementing computationally efficient models for use in numerically intensive optimal mechanical synthesis problems or in resource-constraint embedded computers, particularly for control and model identification. The software support presented for solving the inverse dynamic problem efficiently also provides some insight on some of the advantages and/or disadvantages on these revisited methods.

Acknowledgements. The authors acknowledge the financial support from the Natural Science and Engineering Research Council of Canada (NSERC), the New Brunswick Innovation Foundation (NBIF), Fondo Nacional de Ciencia, Tecnología e Innovación (FONACIT-Venezuela), CONACYT scholarship 326912/381134 and also the SNI-México.

Disclaimer. Conflict of interests – the authors declare that the research was conducted in the absence of any commercial, financial, or personal relationships that could be construed as a potential conflict of interests.

Edited by: D. Pisla

Reviewed by: two anonymous referees

References

- Angeles, J.: Fundamentals of Robotic Mechanical Systems, Chapter 4, 2nd Edn., Springer, 2002.
- Carbonari, L., Battistelli, M., Callegari, M., and Palpacelli, M.-C.: Dynamic modelling of a 3-CPU parallel robot via screw theory, *Mech. Sci.*, 4, 185–197, doi:10.5194/ms-4-185-2013, 2013.
- Carretero, J. A., Nahon, M. A., and Podhorodeski, R. P.: Workspace analysis and optimization of a novel 3-DOF parallel manipulator, *Int. J. Robot. Autom.*, 14, 178–188, 2000a.
- Carretero, J. A., Podhorodeski, R. P., Nahon, M. A., and Gosselin, C. M.: Kinematic analysis and optimization of a new three degree-of-freedom spatial parallel manipulator, *J. Mech. Design*, 122, 17–24, 2000b.
- Chapra, S. C. and Canale, R.: Numerical Methods for Engineers, 5th Edn., McGraw-Hill, Inc., New York, NY, USA, 2006.
- Craig, J. J.: Introduction to Robotics: Mechanics and Control, 3rd Edn., Pearson Education, Upper Saddle River, NJ, USA, 2005.
- Díaz-Rodríguez, M., Valera, A., Mata, V., and Valles, M.: Model-Based Control of a 3-DOF Parallel Robot Based on Identified Relevant Parameters, *IEEE-ASME T. Mech.*, 18, 1737–1744, 2013.
- Fan, K. C., Wang, H., and Chang, T. H.: Sensitivity analysis of the 3-PRS parallel kinematic spindle platform of a serial-parallel machine tool, *Int. J. Mach. Tool. Manu.*, 43, 1561–1569, 2003.
- Goodwin, G. C., Middleton, R. H., and Poor, H. V.: High-speed digital signal processing and control, *Proceedings of the IEEE*, 80, 240–259, 1992.
- Ibrahim, O. and Khalil, W.: Kinematic and Dynamic Modeling of the 3-PRS Parallel Manipulator, in: *Proceedings of the 12th IFToMM World congress, France, 18–21 June 20017*, 1–6, 2007.
- Khalil, W. and Ibrahim, O.: General Solution for the Dynamic Modeling of Parallel Robots, *J. Intell. Robot. Syst.*, 49, 19–37, 2007.
- Li, Y. M. and Xu, Q. S.: Kinematics and inverse dynamics analysis for a general 3-PRS spatial parallel manipulator, *Robotica*, 22, 219–229, 2004.
- Li, Y. M. and Xu, Q. S.: Kinematic analysis of a 3-PRS parallel manipulator, *Robot. CIM. Int. Manuf.*, 23, 395–408, 2007.
- Liu, X. J. and Bonev, I. A.: Orientation Capability, Error Analysis, and Dimensional Optimization of Two Articulated Tool Heads With Parallel Kinematics, *J. Manuf. Sci. Eng.*, 130, 1–9, 2008.
- Mata, V., Provenzano, S., Valero, F., and Cuadrado, J. I.: Serial-robot dynamics algorithms for moderately large numbers of joints, *Mech. Mach. Theory*, 37, 739–755, 2002.
- Mata, V., Farhat, N., Díaz-Rodríguez, M., Valera, A., and Page, A.: Dynamic parameters identification for parallel manipulator, *Tech Education and Publishing, Vienna, Austria*, 21–44, 2008.
- Merlet, J. P.: Micro parallel robot MIPS for medical applications, in: *Proceedings of the 8th international conference on emerging technologies and factory automation (ETFA 2001), France, 15–18 October 2001*, 611–619, 2001.
- Murray, J. and Lovell, G.: Dynamic modeling of closed-chain robotic manipulators and implications for trajectory control, *IEEE T. Robot. Autom.*, 5, 522–528, 1989.
- Parsa, S. S., Carretero, J. A., and Boudreau, R.: Internal redundancy: an approach to improve the dynamic parameters around sharp corners, *Mech. Sci.*, 4, 233–242, doi:10.5194/ms-4-233-2013, 2013.
- Staicu, S.: Matrix modeling of inverse dynamics of spatial and planar parallel robots, *Multibody Syst. Dyn.*, 27, 239–265, 2012.
- Tsai, M. S. and Yuan, W. H.: Inverse dynamics analysis for a 3-PRS parallel mechanism based on a special decomposition of the reaction forces, *Mech. Mach. Theory*, 45, 1491–1508, 2010.
- Tsai, M. S., Shiau, T. N., Tsai, Y. J., and Chang, T. H.: Direct kinematic analysis of a 3-PRS parallel manipulator, *Mech. Mach. Theory*, 38, 71–83, 2002.
- Williamson, D.: Digital Control and Implementation: Finite Wordlength considerations, Prentice Hall, 1991.
- Yiu, Y., Cheng, H., Xiong, Z. H., Liu, G., and Li, Z.: On the dynamics of parallel manipulators, in: *Robotics and Automation, 2001, Proceedings 2001 ICRA, IEEE International Conference on*, 4, 3766–3771, 2001.
- Yuan, W. H. and Tsai, M. S.: A novel approach for forward dynamic analysis of 3-PRS parallel manipulator with consideration of friction effect, *Robot. CIM. Int. Manuf.*, 30, 215–325, 2014.