



Dynamic modelling of a 3-CPU parallel robot via screw theory

L. Carbonari, M. Battistelli, M. Callegari, and M.-C. Palpacelli

Università Politecnica delle Marche, Via Brecce Bianche, 60131, Ancona, Italy

Correspondence to: L. Carbonari (l.carbonari@univpm.it)

Received: 15 November 2012 – Revised: 15 March 2013 – Accepted: 5 April 2013 – Published: 24 April 2013

Abstract. The article describes the dynamic modelling of I.Ca.Ro., a novel Cartesian parallel robot recently designed and prototyped by the robotics research group of the Polytechnic University of Marche. By means of screw theory and virtual work principle, a computationally efficient model has been built, with the final aim of realising advanced model based controllers. Then a dynamic analysis has been performed in order to point out possible model simplifications that could lead to a more efficient run time implementation.

1 Introduction

Many approaches are available for the dynamic modelling of multi-body mechanical systems (Kovacs et al., 2003; Moon, 2008; Papastavridis, 2012) and in the last years, many most of them have been investigated by robotics researchers to achieve efficient models of robots dynamics. Indeed, the efficiency in computation of inverse dynamics of robotic manipulators has a fundamental importance if such tools are involved in the implementation of model based control algorithms whose effectiveness is strongly affected by the computational efficiency of the mathematical model (Lin and Song, 1990; Wang et al., 2007). Thus, it is interesting to investigate the possibility to build simplified dynamics models, especially for parallel kinematic machines that are characterized by an inherent toughness due to the closed kinematic structure. Such peculiarity often complicates the computation of the dynamic model and sometimes prevents the use of model based controls. This inherent complexity is the main reason why only few dynamic models of parallel robots are presently available in scientific literature in symbolic form (Dasgupta and Mruthyunjaya, 1998; Tsai, 2000; Caccavale et al., 2003).

The traditional Newton-Euler formulation, which has been widely used in the past (Do and Yang, 1988; Dasgupta and Mruthyunjaya, 1998) and is still used for specific tasks by some researchers (Kunquan and Rui, 2011; Khalil and Ibrahim, 2007), hardly adapts to the particular case of parallel kinematics machines.

As a matter of fact, all mechanical principles have been used to carry on dynamic analysis of robotic systems, such as the generalized momentum approach (Lopes, 2009), the Hamilton's principle (Miller, 2004), the Lagrange formulation (Wronka and Dunnigan, 2011; Di Gregorio and Parenti-Castelli, 2004) and the virtual work principle (Zhang and Song, 1993).

This last method was proposed in 1993 by Zhang and Song, who used it for the inverse dynamic modelling of open-loop manipulators; later Wang and Gosselin in 1998 expanded the approach to the study of closed kinematics mechanical chains and it is still much used for the modelling of PKMs (Daun et al., 2010). Due to its computational efficiency, such approach is often used when the dynamic modelling aims at the realization of model-based control algorithms. In fact, even if all methods lead to equivalent dynamic equations, these equations present different levels of complexity and associated computational loads; minimizing the number of operations involved in the computation of the manipulator dynamics model has been the main goal of recently proposed techniques (Abdellatif and Heimann, 2009; Yang et al., 2012): since by the use of the virtual work principle constraint forces and moments do not need to be computed, this approach leads to faster computational algorithms, which is a very important advantage for the purpose of robot control. Furthermore, the vector approach specific of the virtual work principle is particularly feasible for computer implementation.

In order to formulate the dynamic model of a mechanical system, the knowledge of its position kinematics is strictly necessary. As a matter of fact, the solution of the forward kinematics problem (FKP) of a parallel platform represents a challenging issue that not necessarily yields to a closed form solution, especially when the robot end-effector is allowed to perform motions of rotation.

As argued by authors in past works (Carbonari, 2012; Carbonari and Callegari, 2012) the 3-CPU parallel architecture can provide the end-effector with different kinds of mobility, depending on the mutual configuration of the joints that compose the leg's kinematic chain. Carbonari et al. (2013) also demonstrated that a reconfiguration of the universal joint allows to modify the kinematic behaviour of a 3-CPU parallel robot, switching from a pure rotational to a pure translational kinematic behaviour.

This paper focuses on the dynamic modelling of a pure translational 3-CPU architecture, called I.Ca.Ro. by Callegari and Palpacelli (2008), aimed at the realization of a non-linear model based control scheme. The main object of the present work is to produce a numerically efficient dynamic model of the machine, suitable to be used for the realization of a control algorithm. To this aim, the differential kinematics of the manipulator has been tackled taking advantage of a screw based approach (Gallardo et al., 2003). For the seek of completeness, the position kinematics is also presented here in order to improve the reader's understanding of the problem.

2 Robot kinematics

The I.Ca.Ro. parallel robot is a pure translational Cartesian tripod whose limbs are built of a C-P-U (cylindrical-prismatic-universal) joints chain. The first body of each leg is connected to the robot chassis by means of a cylindrical joint, realized through a prismatic actuated pair and a revolute passive joint coaxial to the first one (refer to Fig. 1). The second body is linked to the first one through a prismatic joint, perpendicular to the axis of the cylindrical joint. At last, the second body is connected to the moving platform through a universal joint composed of two revolutes: the first revolute joint is parallel to the second link and the second is perpendicular to the first one. The last revolute of each leg connects the manipulator with the respective limb; the axes of such joints are coplanar.

Due to the robot kinematic architecture, the I.Ca.Ro. parallel manipulator is only able to provide the end-effector with pure translations. In fact, by means of screw theory it can be observed that each leg exerts a constraint wrench made of a pure torque along the direction of its passive prismatic pair. The connection of the three legs to the mobile platform produces a wrench system of three orthogonal torques, whose dual space is spanned by a basis of three linearly independent pure translations. Thus, the orientation of a reference frame

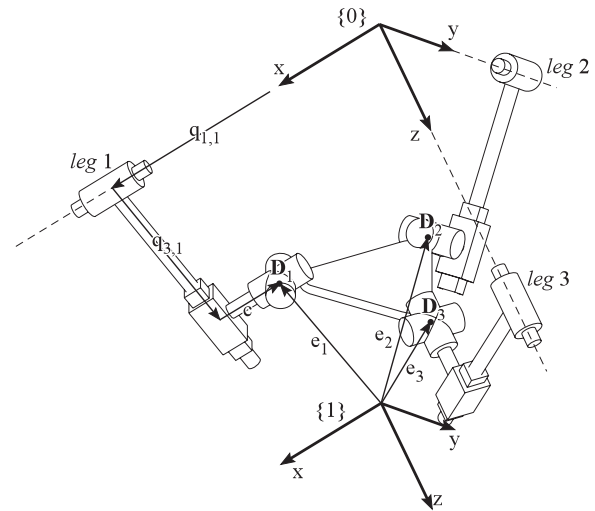


Figure 1. Kinematics of the 3-CPU pure translational parallel robot.

that solidly moves with it remains constant notwithstanding the displacement that the actuated joints perform.

With respect to the notation introduced in Fig. 1, the homogeneous transformation matrix that describes the configuration of reference frame {1} with respect to reference frame {0} can be expressed as:

$${}^0\mathbf{T}_1 = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where p_x , p_y and p_z denote the position of the centre of the moving frame {1} and the 3 by 3 identity rotation matrix suggests that the orientation of the manipulator remains constant.

The forward and inverse kinematics problems of the robot can be easily solved taking advantage of three loop closure equations. The positions of the three attachment points D_i between the end-effector and the three limbs can be simply reached through the transformation matrix ${}^0\mathbf{T}_1$, being their position fixed with respect to reference frame {1}: $D_i = {}^0\mathbf{T}_1 [e_i^T \ 1]^T$, where $e_1 = -e [0 \ 1/\sqrt{2} \ 1/\sqrt{2}]^T$, $e_2 = -e [1/\sqrt{2} \ 0 \ 1/\sqrt{2}]^T$, $e_3 = -e [1/\sqrt{2} \ 1/\sqrt{2} \ 0]^T$. As it is shown in the following, the coordinates of such points can be also reached through the use of legs' joints displacements. The comparison between the different expressions of these three points provides the solution of the problem. The reference frames shown in Fig. 2 are used to define legs kinematics.

Starting with frame {0}, which is the frame attached to manipulator chassis, a displacement along its x-axis and a rotation about the same axis are needed to describe the position of the second frame {A}. Transformation matrix will look

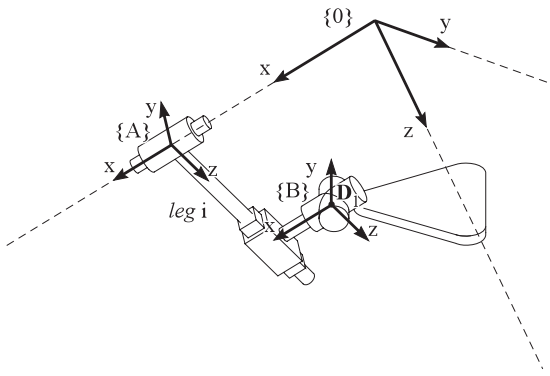


Figure 2. Reference frames along leg structure.

like:

$${}^0\mathbf{T}_A = \begin{bmatrix} 1 & 0 & 0 & q_{1,1} \\ 0 & cq_{2,1} & -sq_{2,1} & 0 \\ 0 & sq_{2,1} & cq_{2,1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where $q_{2,1}$ denotes the rotation of the cylindrical pair of the leg 1 and shorthand notation is used for trigonometric functions. To reach the configuration of the third reference frame {B}, thus the position of attachment point D_1 , a translational transformation matrix is sufficient:

$${}^A\mathbf{T}_B = \begin{bmatrix} 1 & 0 & 0 & -c \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_{3,1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

with $q_{3,1}$ denoting the displacement performed by prismatic pair of the first leg. Coordinates of point D_1 can now be achieved as:

$$D_1 = {}^0\mathbf{T}_A {}^A\mathbf{T}_B \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \quad (4)$$

Since the kinematic chain is identical for each leg, it is not necessary to explicit their homogeneous transformations as specific cases. Indeed, Eq. (4) can be exploited if pre multiplied by a transformation that rotates the starting reference frame. In particular, it is possible to define the matrices:

$${}^0\mathbf{T}_{leg2} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^0\mathbf{T}_{leg3} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

such that the coordinates of the remaining attachment points can be expressed as:

$$D_2 = {}^0\mathbf{T}_{leg2} {}^{leg2}\mathbf{T}_A {}^A\mathbf{T}_B \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \quad (6)$$

$$D_3 = {}^0\mathbf{T}_{leg3} {}^{leg3}\mathbf{T}_A {}^A\mathbf{T}_B \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$$

Even if it is not specified, it should be evident that transformations ${}^{leg2}\mathbf{T}_A$, ${}^{leg3}\mathbf{T}_A$ and ${}^A\mathbf{T}_B$ of the 2nd and 3rd equations

in (6) are expressed in terms of the respective joints variables $q_{1,i}$, $q_{2,i}$ and $q_{3,i}$.

In order to give a general expression of legs kinematics, it is also introduced the matrix ${}^0\mathbf{T}_{leg1}$, an identity 4 by 4 matrix that allows to write the Eq. (4) as:

$$D_1 = {}^0\mathbf{T}_{leg1} {}^{leg1}\mathbf{T}_A {}^A\mathbf{T}_B \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \quad (7)$$

where ${}^{leg1}\mathbf{T}_A = {}^0\mathbf{T}_A$.

Expansion of (6) and (7) yields to the expression of such points coordinates in terms of joints variables, that are:

$$D_1 = \begin{bmatrix} q_{1,1} - c \\ q_{3,1}sq_{2,1} \\ -q_{3,1}cq_{2,1} \\ 1 \end{bmatrix} \quad D_2 = \begin{bmatrix} -q_{3,2}cq_{2,2} \\ q_{1,2} - c \\ q_{3,2}sq_{2,2} \\ 1 \end{bmatrix} \quad D_3 = \begin{bmatrix} q_{3,3}sq_{2,3} \\ -q_{3,3}cq_{2,3} \\ q_{1,3} - c \\ 1 \end{bmatrix} \quad (8)$$

Equations (8) can be inverted to achieve the expression of legs joints variables as functions of coordinates of the three attachment points D_i :

$$q_{1,1} = D_{1,x} + c \quad q_{2,1} = \tan^{-1} \frac{D_{1,y}}{-D_{1,z}} \quad q_{3,1} = \frac{D_{1,y}}{\sin q_{2,1}}$$

$$q_{1,2} = D_{2,y} + c \quad q_{2,2} = \tan^{-1} \frac{D_{2,z}}{-D_{2,x}} \quad q_{3,2} = \frac{D_{2,z}}{\sin q_{2,2}}$$

$$q_{1,3} = D_{3,z} + c \quad q_{2,3} = \tan^{-1} \frac{D_{3,x}}{-D_{3,y}} \quad q_{3,3} = \frac{D_{3,x}}{\sin q_{2,3}}$$

It is worth to remark that such coordinates are uniquely determined if the pose of the manipulator is known.

3 Velocity kinematics

In order to build the Jacobian matrices of a PKM, both position and orientation of the joints axes are needed. Firstly it is necessary to define an appropriate number of reference frames preferably attached in convenient points of the kinematic chain. It is worth to remember that the screws must be expressed with respect to a reference frame attached to robot manipulator and whose orientation is constant and coincident to that of robot absolute reference frame {0}. Furthermore, in order to define the dynamical model of the machine, the moving frame should be centred at the c.o.m. of the body object of the velocity analysis.

In the case of the pure translational robot I.Ca.Ro. the moving platform reference frame {1} represents a feasible choice for expressing the screw coordinates. In fact, it is solid with the end-effector and it does not rotate; moreover, due to the pure translational mobility of the end-effector, the origin of frame {1} moves with the same velocity and acceleration of the end-effector c.o.m. even if it is not centred on it.

The screw coordinates of every kinematic pair involved in leg kinematic chain must be expressed: to this aim, convenient local frames must be arranged along the legs as shown in Fig. 3.

Two new frames {C} and {D} must be attached to the revolute joints that compose the universal joints. The homogeneous transformations denoting configuration of such frames

are defined as:

$$\begin{aligned}
 {}^B\mathbf{T}_C &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cq_{4,1} & -sq_{4,1} & 0 \\ 0 & sq_{4,1} & cq_{4,1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^C\mathbf{T}_D &= \begin{bmatrix} cq_{5,1} & 0 & sq_{5,1} & 0 \\ 0 & 1 & 0 & 0 \\ -sq_{5,1} & 0 & cq_{5,1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{10}$$

The pose of frames {A}, {B}, {C} and {D} is described by the following homogeneous transformations:

$$\begin{aligned}
 {}^1\mathbf{T}_{A,1} &= {}^1\mathbf{T}_0^0 \mathbf{T}_{leg1}^{leg1} {}^1\mathbf{T}_A \\
 {}^1\mathbf{T}_{B,1} &= {}^1\mathbf{T}_0^0 \mathbf{T}_{leg1}^{leg1} {}^1\mathbf{T}_A {}^A\mathbf{T}_B \\
 {}^1\mathbf{T}_{C,1} &= {}^1\mathbf{T}_0^0 \mathbf{T}_{leg1}^{leg1} {}^1\mathbf{T}_A {}^A\mathbf{T}_B {}^B\mathbf{T}_C \\
 {}^1\mathbf{T}_{D,1} &= {}^1\mathbf{T}_0^0 \mathbf{T}_{leg1}^{leg1} {}^1\mathbf{T}_A {}^A\mathbf{T}_B {}^B\mathbf{T}_C {}^C\mathbf{T}_D
 \end{aligned} \tag{11}$$

The unit vectors of joints' axes can be easily expressed in the global frame by means of the proper mapping between the local frames and the global one:

– **Joint 1**, prismatic:

$$\begin{bmatrix} s_{1,i} \\ 1 \end{bmatrix} = {}^1\mathbf{T}_{A,i} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \rightarrow S_{1,i} = \begin{bmatrix} \mathbf{0} \\ s_{1,i} \end{bmatrix} \tag{12}$$

– **Joint 2**, revolute:

$$\begin{bmatrix} s_{2,i} \\ 1 \end{bmatrix} = {}^1\mathbf{T}_{A,i} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} r_{2,i} \\ 1 \end{bmatrix} = {}^1\mathbf{T}_{A,i} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \rightarrow S_{2,i} = \begin{bmatrix} s_{2,i} \\ r_{2,i} \times s_{2,i} \end{bmatrix} \tag{13}$$

– **Joint 3**, prismatic:

$$\begin{bmatrix} s_{3,i} \\ 1 \end{bmatrix} = {}^1\mathbf{T}_{A,i} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \rightarrow S_{3,i} = \begin{bmatrix} \mathbf{0} \\ s_{3,i} \end{bmatrix} \tag{14}$$

– **Joint 4**, first revolute of the universal joint:

$$\begin{bmatrix} s_{4,i} \\ 1 \end{bmatrix} = {}^1\mathbf{T}_{B,i} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} r_{4,i} \\ 1 \end{bmatrix} = {}^1\mathbf{T}_{B,i} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \rightarrow S_{4,i} = \begin{bmatrix} s_{4,i} \\ r_{4,i} \times s_{4,i} \end{bmatrix} \tag{15}$$

– **Joint 5**, second revolute of the universal joint:

$$\begin{bmatrix} s_{5,i} \\ 1 \end{bmatrix} = {}^1\mathbf{T}_{C,i} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} r_{5,i} \\ 1 \end{bmatrix} = {}^1\mathbf{T}_{C,i} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \rightarrow S_{5,i} = \begin{bmatrix} s_{5,i} \\ r_{5,i} \times s_{5,i} \end{bmatrix} \tag{16}$$

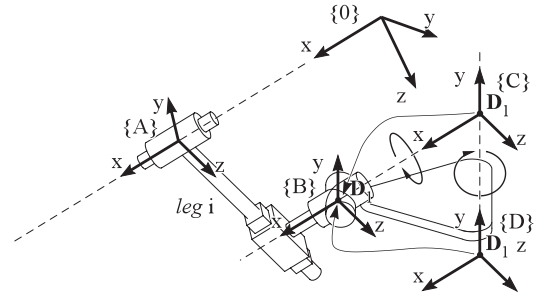


Figure 3. Local frames used for definition of joints unit screws.

In order to simplify expressions of manipulator Jacobian matrices it is possible to use three screws which have the main characteristic of being reciprocal to all unit screws of the leg, with the exception of the actuated joints screws. Such screws are here called $S_{r,i}$.

To this aim, it is possible to make use of the unit screw $S_{4,i}$ shown in Fig. 4, which turns reciprocal to each non-actuated screw present in the leg kinematic chain due to the fact that it is coplanar to both $S_{2,i}$ and $S_{5,i}$, it intersects the axis of the prismatic joint described by $S_{3,i}$ and, by definition, it is reciprocal to itself. Furthermore, it is not reciprocal to $S_{1,i}$ being parallel but not aligned to the axis of the actuated prismatic joint.

At this point vector expressions have been given for joints screws and for legs reciprocal screws. The Jacobian matrices can be formulated in order to achieve an expression for the velocity problem which has the well known form $\mathbf{J}_X \dot{\mathbf{x}} = \mathbf{J}_Q \dot{\mathbf{q}}$, where $\dot{\mathbf{x}}$ is the velocity vector of the platform that, in a general way, can be expressed as $\dot{\mathbf{x}} = [\omega_x \ \omega_y \ \omega_z \ \dot{q}_x \ \dot{q}_y \ \dot{q}_z]^T$.

Firstly it is introduced the Jacobian matrix \mathbf{J}_X , whose expression can be formulated as a function of reciprocal screws:

$$\begin{aligned}
 \mathbf{J}_X &= \begin{bmatrix} S_{r,1}^T \\ S_{r,2}^T \\ S_{r,3}^T \end{bmatrix} = \begin{bmatrix} S_{4,1}^T \\ S_{4,2}^T \\ S_{4,3}^T \end{bmatrix} \\
 &= \begin{bmatrix} 0 & q_{3,1}cq_{2,1}-p_z & q_{3,1}sq_{2,1}+p_y & 1 & 0 & 0 \\ q_{3,2}sq_{2,2}+p_z & 0 & q_{3,2}cq_{2,2}-p_x & 0 & 1 & 0 \\ q_{3,3}cq_{2,3}-p_y & q_{3,3}sq_{2,3}+p_x & 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{17}$$

The moving platform of I.Ca.Ro. PKM is only allowed to perform pure translations; this implies that the first three components of the vector $\dot{\mathbf{x}}$, i.e. ω_x , ω_y and ω_z , are identically null. As a consequence, such components and the first three columns of matrix \mathbf{J}_X can be eliminated due to the fact that they do not give any contribution to equation $\mathbf{J}_X \dot{\mathbf{x}} = \mathbf{J}_Q \dot{\mathbf{q}}$. Thus, the Jacobian \mathbf{J}_X is a three by three identity matrix which multiplies the end-effector velocity vector $\dot{\mathbf{x}} = [\dot{q}_x \ \dot{q}_y \ \dot{q}_z]^T$.

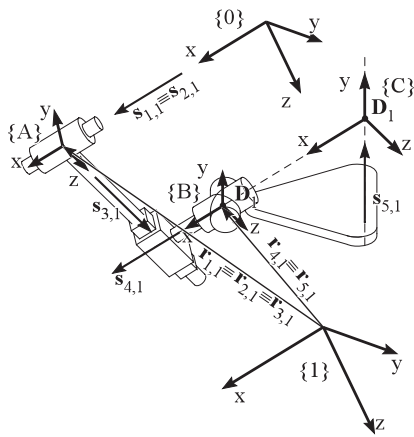


Figure 4. Unit screws of each kinematic joints.

Finally matrix \mathbf{J}_Q is introduced:

$$\mathbf{J}_Q = \begin{bmatrix} \mathbf{S}_{r,1}^T \mathbf{S}_{1,1} & 0 & 0 \\ 0 & \mathbf{S}_{r,2}^T \mathbf{S}_{1,2} & 0 \\ 0 & 0 & \mathbf{S}_{r,3}^T \mathbf{S}_{1,3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (18)$$

Due to pure translational robot kinematic behaviour, the end-effector velocity problem can be simply expressed as:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix} \quad (19)$$

Thanks to the screw based approach, the velocities of passive joints have been eliminated from the formulation of end-effector velocity kinematics. Nevertheless this information is needed to perform other types of analysis such as the study of robot dynamics.

Thus, the knowledge of the velocity vectors of each member composing the legs is necessary and it can be achieved through the computation of passive joints velocity as functions of active joints rates. To this aim, robot architecture constraint equations are exploited to build a matrix \mathbf{A} relating prismatic actuated joints velocities to all other rates:

$$\dot{\mathbf{q}}_p = \mathbf{A} \dot{\mathbf{q}}_a \quad (20)$$

where $\dot{\mathbf{q}}_p$ is a vector collecting velocities of passive joints and $\dot{\mathbf{q}}_a$ is the vector of actuated joints rates.

The main aim of this computation is to provide the needed tools for the dynamic modelling of the robotic system. Therefore, a simplification based on influence of each body is introduced yielding a relevant computational simplification. The mass of the elements that compose the revolute joints is negligible if compared with masses of legs linkages and translating parts of prismatic actuators. Hence, it is supposed here that they only marginally affect the whole dynamic behaviour of the robot: thus, their contribution is not considered. This simplification is reasonably acceptable and enormously simplifies robot model because of the complexity introduced by the velocity expressions of these elements.

Such simplification allows us to reduce the dimension of matrix \mathbf{A} . If the actual number of active and passive joints is considered, Eq. (20) can be expanded to:

$$\begin{bmatrix} \dot{q}_{2,1} \\ \dot{q}_{2,2} \\ \dot{q}_{2,3} \\ \dot{q}_{3,1} \\ \dot{q}_{3,2} \\ \dot{q}_{3,3} \end{bmatrix} = \mathbf{A} \begin{bmatrix} \dot{q}_{1,1} \\ \dot{q}_{1,2} \\ \dot{q}_{1,3} \end{bmatrix} \quad (21)$$

where $\dot{q}_{1,i}$ is the translation rate of cylindrical pair of i -th leg, $\dot{q}_{2,i}$ is the rotation rate of the same pair and $\dot{q}_{3,i}$ is the translation rate of the passive prismatic joint.

The constraint matrix \mathbf{A} can be built considering the mobility of each attachment point between legs and manipulator; indeed, the velocity of such points is known and equal to the velocity of the moving platform due to the fact that they solidly move with the end-effector which only performs pure translational motions. The velocities of passive joints can be related to the components of the velocity vectors as visible in Fig. 5 for a general mobility.

The component of $\mathbf{v}_{D,i}$ along the direction perpendicular to leg plane is expressed by:

$$v_{D,i}^\perp = \mathbf{v}_{D,i}^T (\mathbf{s}_{2,i} \times \mathbf{s}_{3,i}) \quad (22)$$

The velocity along this direction is fully due to the rotation of the cylindrical joint, so that:

$$\dot{q}_{2,i} = \frac{\mathbf{v}_{D,i}^T (\mathbf{s}_{2,i} \times \mathbf{s}_{3,i})}{q_{3,i}} \quad (23)$$

The component of velocity that lies on the leg plane is due to both the actuated and non actuated prismatic joints:

$$\begin{aligned} \dot{q}_{1,i} &= \mathbf{v}_{D,i}^T \mathbf{s}_{1,i} \\ \dot{q}_{3,i} &= \mathbf{v}_{D,i}^T \mathbf{s}_{3,i} \end{aligned} \quad (24)$$

The first equation in (24) simply relates the velocity along the axis of the cylindrical joint to the actuation rate; thus, it is not useful for the construction of the constraint matrix. On the other hand, the second equation can be used for the scope.

Equation (23) and the second equation in (24) can be expanded and written in the matrix form (20). In the case of a pure translational robot the velocities of legs attachment points correspond to the velocity of the origin of end effector reference frame. Exploiting Eqs. (22) and the second of (24), expressions of non actuated joints rates are achievable. In this case, the simplification introduced by end-effector mobility allows to show which is the actual shape of such expressions.

For the revolute joints it is:

$$\begin{aligned} \dot{q}_{2,1} &= \frac{-\dot{q}_{1,2} c q_{2,1} - \dot{q}_{1,3} s q_{2,1}}{q_{3,1}} \\ \dot{q}_{2,2} &= \frac{-\dot{q}_{1,1} s q_{2,2} - \dot{q}_{1,3} c q_{2,2}}{q_{3,2}} \\ \dot{q}_{2,3} &= \frac{-\dot{q}_{1,1} c q_{2,3} - \dot{q}_{1,2} s q_{2,3}}{q_{3,3}} \end{aligned} \quad (25)$$

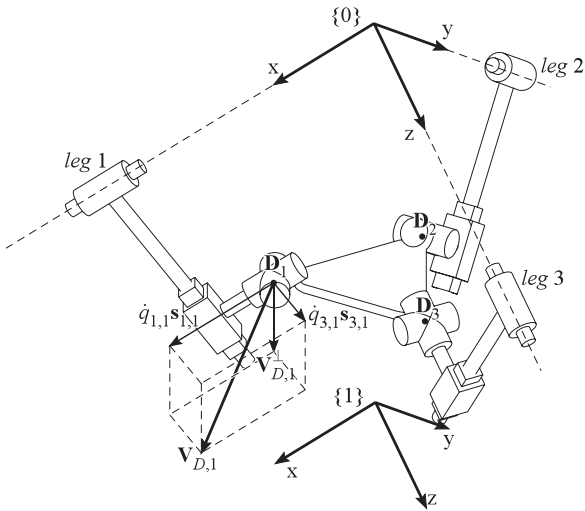


Figure 5. Velocity of attachment points between legs and moving platform.

while prismatic joints rates are:

$$\begin{aligned} \dot{q}_{3,1} &= -\dot{q}_{1,2}sq_{2,1} + \dot{q}_{1,3}cq_{2,1} \\ \dot{q}_{3,2} &= \dot{q}_{1,1}cq_{2,2} - \dot{q}_{1,3}sq_{2,2} \\ \dot{q}_{3,3} &= -\dot{q}_{1,1}sq_{2,3} + \dot{q}_{1,2}cq_{2,3} \end{aligned} \quad (26)$$

Hence, the matrix formulation of the 6 passive velocities is:

$$\begin{bmatrix} \dot{q}_{2,1} \\ \dot{q}_{2,2} \\ \dot{q}_{2,3} \\ \dot{q}_{3,1} \\ \dot{q}_{3,2} \\ \dot{q}_{3,3} \end{bmatrix} = \begin{bmatrix} 0 & \frac{-cq_{2,1}}{q_{3,1}} & \frac{-sq_{2,1}}{q_{3,1}} \\ \frac{-sq_{2,2}}{q_{3,2}} & 0 & \frac{-cq_{2,2}}{q_{3,2}} \\ \frac{cq_{2,3}}{q_{3,3}} & \frac{-sq_{2,3}}{q_{3,3}} & 0 \\ 0 & -sq_{2,1} & cq_{2,1} \\ cq_{2,2} & 0 & -sq_{2,2} \\ -sq_{2,3} & cq_{2,3} & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_{1,1} \\ \dot{q}_{1,2} \\ \dot{q}_{1,3} \end{bmatrix} \quad (27)$$

In the remainder of this work, the constraint matrix is used to express the velocity of the reference frames attached to the robot bodies. To do that, further Jacobian matrices are introduced. In particular, the velocity of the c.o.m. of each body is written according to the general formulation:

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}_a \quad (28)$$

where $\dot{\mathbf{q}}_a$ is the vector of actuated joints rates.

It is important to remark that the target of this section is the definition of legs bodies velocities, whose serial kinematics chain does not allow the simplification of passive joints rates. Thus, the Jacobian formulation $\dot{\mathbf{x}} = \tilde{\mathbf{J}} \begin{bmatrix} \dot{\mathbf{q}}_a^T \\ \dot{\mathbf{q}}_p^T \end{bmatrix}^T$ involves also the velocity of non actuated joints. Nevertheless, the influence of such joints can be explicated by means of the constraint matrix (27). Equation (28) becomes:

$$\dot{\mathbf{x}} = \tilde{\mathbf{J}} \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \end{bmatrix} \dot{\mathbf{q}}_a \quad (29)$$

where \mathbf{I} is a 3×3 identity matrix; the matrices $\tilde{\mathbf{J}}$ can be very quickly expressed taking advance of joints screws; thus the

formulation of legs velocities turns out to be an immediate iterative process based on collection of already introduced vectors.

Firstly, the velocities of the three sliders are achieved: for the sake of conciseness, these bodies are denoted as sl_1 , sl_2 and sl_3 with reference to the leg which they are part of. The serial chain that allows reaching their screw is composed only by the actuated prismatic pair. In this case the body is not allowed to rotate, so that the position of the screw axis does not influence the screw expression:

$$\dot{\mathbf{x}}_{sl,i} = \mathbf{S}_{1,i}\dot{q}_{i,1} = \begin{bmatrix} \mathbf{0} \\ s_{1,i} \end{bmatrix} \dot{q}_{i,1} \quad (30)$$

Equations (30) can be written according to the generic formulation (29):

$$\begin{aligned} \dot{\mathbf{x}}_{sl,1} &= \begin{bmatrix} \mathbf{S}_{1,1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \end{bmatrix} \dot{\mathbf{q}}_a = \mathbf{J}_{sl1} \dot{\mathbf{q}}_a \\ \dot{\mathbf{x}}_{sl,2} &= \begin{bmatrix} \mathbf{0} & \mathbf{S}_{1,2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \end{bmatrix} \dot{\mathbf{q}}_a = \mathbf{J}_{sl2} \dot{\mathbf{q}}_a \\ \dot{\mathbf{x}}_{sl,3} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{S}_{1,3} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \end{bmatrix} \dot{\mathbf{q}}_a = \mathbf{J}_{sl3} \dot{\mathbf{q}}_a \end{aligned} \quad (31)$$

In a very similar way, velocities of the first links (called here $l_{1,i}$) can be achieved. The serial kinematic chain characteristic of such bodies is composed by the actuated prismatic pair and the non actuated revolute joint:

$$\dot{\mathbf{x}}_{l1,i} = \mathbf{S}_{1,i}\dot{q}_{i,1} + \mathbf{S}_{2,i}\dot{q}_{i,2} = \begin{bmatrix} \mathbf{0} \\ s_{1,i} \end{bmatrix} \dot{q}_{i,1} + \begin{bmatrix} s_{2,i} \\ \mathbf{r}_{2,i} \times s_{2,i} \end{bmatrix} \dot{q}_{i,2} \quad (32)$$

The position of the screw axis is relevant for the computation of the revolute joint unit screw. Even though its expression does not coincide with the previously found value, axis position is quickly computable using homogeneous transformation matrix and the position vector $\mathbf{p}_{l1,i}$ of the center of mass of body $l_{1,i}$ with respect to reference frame $\{A\}$. The position of the screw axis is computable as the difference between absolute position $\mathbf{O}_{A,i}$ of frame $\{A\}$ and absolute position of center of mass:

$$\begin{aligned} \mathbf{p}_{l1,i} &= {}^0\mathbf{T}_A \mathbf{p}_{l1,i} \\ \mathbf{O}_{A,i} &= {}^0\mathbf{T}_A \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \rightarrow \mathbf{r}_{2,i} = \mathbf{O}_{A,i} - \mathbf{p}_{l1,i} \end{aligned} \quad (33)$$

Expansion of (32) for all robot legs yields:

$$\begin{aligned} \dot{\mathbf{x}}_{l1,1} &= \begin{bmatrix} \mathbf{S}_{1,1} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{2,1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \end{bmatrix} \dot{\mathbf{q}}_a = \mathbf{J}_{l1,1} \dot{\mathbf{q}}_a \\ \dot{\mathbf{x}}_{l1,2} &= \begin{bmatrix} \mathbf{0} & \mathbf{S}_{1,2} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{2,2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \end{bmatrix} \dot{\mathbf{q}}_a = \mathbf{J}_{l1,2} \dot{\mathbf{q}}_a \\ \dot{\mathbf{x}}_{l1,3} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{S}_{1,3} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{2,3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \end{bmatrix} \dot{\mathbf{q}}_a = \mathbf{J}_{l1,3} \dot{\mathbf{q}}_a \end{aligned} \quad (34)$$

Finally, the velocity of the last link that composes the leg (here called body $l_{2,i}$) is a linear combination of the elementary screws of the actuated prismatic joint, the first passive revolute joint and the non actuated prismatic joint of each leg:

$$\begin{aligned} \dot{\mathbf{x}}_{l2,i} &= \mathbf{S}_{1,i}\dot{q}_{i,1} + \mathbf{S}_{2,i}\dot{q}_{i,2} + \mathbf{S}_{3,i}\dot{q}_{i,3} \\ &= \begin{bmatrix} \mathbf{0} \\ s_{1,i} \end{bmatrix} \dot{q}_{i,1} + \begin{bmatrix} s_{2,i} \\ \mathbf{r}_{2,i} \times s_{2,i} \end{bmatrix} \dot{q}_{i,2} + \begin{bmatrix} \mathbf{0} \\ s_{3,i} \end{bmatrix} \dot{q}_{i,3} \end{aligned} \quad (35)$$

The position $\mathbf{r}_{2,i}$ of the screw axis is once again different from the previously exposed case. Nevertheless, its expression is achievable by the definition of a position vector $\mathbf{p}_{l2,i}$ of the center of mass of bodies $l2_i$ with respect to their attached frame $\{B\}$. The distance from this point to the axis of the revolute joint is given by:

$$\begin{aligned} \mathbf{p}_{l2,i} &= {}^0\mathbf{T}_B \mathbf{p}_{l2,i} \\ \mathbf{O}_{B,i} &= {}^0\mathbf{T}_B \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \rightarrow \mathbf{r}_{2,i} = \mathbf{O}_{B,i} - \mathbf{p}_{l2,i} \end{aligned} \quad (36)$$

As done in previous case, the Jacobian formulation can be plainly reached also for velocities of bodies $l2_i$:

$$\begin{aligned} \dot{\mathbf{x}}_{l2,1} &= \begin{bmatrix} S_{1,1} & 0 & 0 & S_{2,1} & 0 & 0 & S_{3,1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I}_A \\ \mathbf{I}_A \\ \mathbf{I}_A \end{bmatrix} \dot{\mathbf{q}}_a = \mathbf{J}_{l2,1} \dot{\mathbf{q}}_a \\ \dot{\mathbf{x}}_{l2,2} &= \begin{bmatrix} 0 & S_{1,2} & 0 & 0 & S_{2,2} & 0 & 0 & S_{3,2} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I}_A \\ \mathbf{I}_A \\ \mathbf{I}_A \end{bmatrix} \dot{\mathbf{q}}_a = \mathbf{J}_{l2,2} \dot{\mathbf{q}}_a \\ \dot{\mathbf{x}}_{l2,3} &= \begin{bmatrix} 0 & 0 & S_{1,3} & 0 & 0 & S_{2,3} & 0 & 0 & S_{3,3} \end{bmatrix} \begin{bmatrix} \mathbf{I}_A \\ \mathbf{I}_A \\ \mathbf{I}_A \end{bmatrix} \dot{\mathbf{q}}_a = \mathbf{J}_{l2,3} \dot{\mathbf{q}}_a \end{aligned} \quad (37)$$

4 Acceleration kinematics

The dynamic modelling of a mechanical system requires a complete knowledge of machine kinematics. Therefore, acceleration of each body must be studied.

Manipulator velocity kinematics has been formulated through the well known Jacobian formulation $\dot{\mathbf{q}}_a = \mathbf{J}\dot{\mathbf{x}}$, where $\mathbf{J} = \mathbf{J}_Q^{-1}\mathbf{J}_X$ is in this case a 3 by 3 identity matrix. Direct differentiation of such expression yields:

$$\ddot{\mathbf{q}}_a = \mathbf{J}\dot{\mathbf{x}} + \mathbf{J}\ddot{\mathbf{x}} \quad (38)$$

where \mathbf{J}_X is the derivative of the jacobian matrix, which is a constant matrix. Thus, in the case of a pure translational machine, the derivation of \mathbf{J}_X yields $\mathbf{J}_X = \mathbf{0}$.

The acceleration kinematics of other robot members is easily achievable by direct differentiation of the velocity kinematics previously defined:

$$\ddot{\mathbf{x}}_{j,i} = \dot{\mathbf{J}}_{j,i} \dot{\mathbf{q}}_a + \mathbf{J}_{j,i} \ddot{\mathbf{q}}_a \quad (39)$$

where $\dot{\mathbf{J}}_{j,i}$ is the time derivative of the respective Jacobian matrix. Expansion of (39) yields to very a long formulation that, for sake of conciseness, is not shown here.

5 Virtual work principle

The virtual work principle approach for dynamic modelling requires the definition of the 6 dimensional vector $\mathbf{F}_{j,i}$, whose components collect resultants of both active and inertial forces and torques acting on the j -th body of i -th leg, computed with respect to the center of mass of the member:

$$\mathbf{F}_{j,i} = \begin{bmatrix} \mathbf{n}_{j,i} \\ \mathbf{F}_{j,i} \end{bmatrix} = \begin{bmatrix} -{}^0\mathbf{I}_{j,i} \dot{\boldsymbol{\omega}}_{j,i} - \boldsymbol{\omega}_{j,i} \times ({}^0\mathbf{I}_{j,i} \boldsymbol{\omega}_{j,i}) \\ m_{j,i} (\mathbf{g} - \dot{\mathbf{v}}_{j,i}) \end{bmatrix} \quad (40)$$

In a similar way, it is introduced the vector \mathbf{F}_{EE} that collects forces and torques acting on robot's end-effector and computed with respect to manipulator centre of mass:

$$\mathbf{F}_{EE} = \begin{bmatrix} \mathbf{n}_{EE} \\ \mathbf{F}_{EE} \end{bmatrix} = \begin{bmatrix} -{}^0\mathbf{I}_{EE} \dot{\boldsymbol{\omega}}_{EE} - \boldsymbol{\omega}_{EE} \times ({}^0\mathbf{I}_{EE} \boldsymbol{\omega}_{EE}) \\ m_{EE} (\mathbf{g} - \dot{\mathbf{v}}_{EE}) \end{bmatrix} \quad (41)$$

Virtual works principle allows writing:

$$\delta \mathbf{q}_a^T \boldsymbol{\tau} + \delta \mathbf{x}^T \mathbf{F}_{EE} + \sum_{i,j} \delta \mathbf{x}_{j,i}^T \mathbf{F}_{j,i} = 0 \quad (42)$$

where vector $\delta \mathbf{q}_a$ represents the virtual displacements of actuated joints, $\boldsymbol{\tau}$ is the vector of actuation torques and $\delta \mathbf{x}$ is the virtual displacement of rotation/displacement of respective body.

The differential kinematics of the manipulator, whose formulation has been introduced in previous sections, is usefully exploited to relate actuated joints displacements to other bodies twists. Indeed, end-effector differential kinematics expression allows writing:

$$\delta \mathbf{q}_a = \mathbf{J}_X \delta \mathbf{x} \quad (43)$$

In a similar way, the twist of other robot members can be expressed through the respective Jacobian matrices:

$$\delta \mathbf{x}_{j,i} = \mathbf{J}_{j,i} \delta \mathbf{q}_a \quad (44)$$

When Eq. (43) is invertible, i.e. when determinant of the Jacobian matrix is not null, the differential of end effector twist can be expressed in terms of actuated joints translations, being $\delta \mathbf{x} = \mathbf{J}_X^{-1} \delta \mathbf{q}_a$. Dynamics Eq. (42) becomes:

$$\delta \mathbf{q}_a^T \boldsymbol{\tau}_A + \delta \mathbf{q}_a^T \mathbf{J}_X^{-T} \mathbf{F}_{EE} + \delta \mathbf{q}_a^T \sum_{i,j} \mathbf{J}_{j,i}^T \mathbf{F}_{j,i} = 0 \quad (45)$$

For non null virtual displacements $\delta \mathbf{q}_a$, such term can be collected and eliminated, yielding:

$$\boldsymbol{\tau} + \mathbf{J}_X^{-T} \mathbf{F}_{EE} + \sum_{i,j} \mathbf{J}_{j,i}^T \mathbf{F}_{j,i} = 0 \quad (46)$$

Equation (46) can be collected in the canonical form:

$$\boldsymbol{\tau} + \mathbf{M}(\mathbf{q}_a) \ddot{\mathbf{q}}_a + \mathbf{V}(\mathbf{q}_a, \dot{\mathbf{q}}_a) + \mathbf{G}(\mathbf{q}_a) = \mathbf{0} \quad (47)$$

As known, each component of this equation includes different contributions to the dynamics of the manipulator: $\mathbf{M}(\mathbf{q}_a) \ddot{\mathbf{q}}_a$ called later $\boldsymbol{\tau}_M$ is a contribution due to inertial effects of bodies masses, $\mathbf{V}(\mathbf{q}_a, \dot{\mathbf{q}}_a)$, hereby called $\boldsymbol{\tau}_V$, is due to Coriolis and centripetal accelerations and, at last, $\mathbf{G}(\mathbf{q}_a)$ are the forces deriving from gravitational action on robot members, called here $\boldsymbol{\tau}_G$.

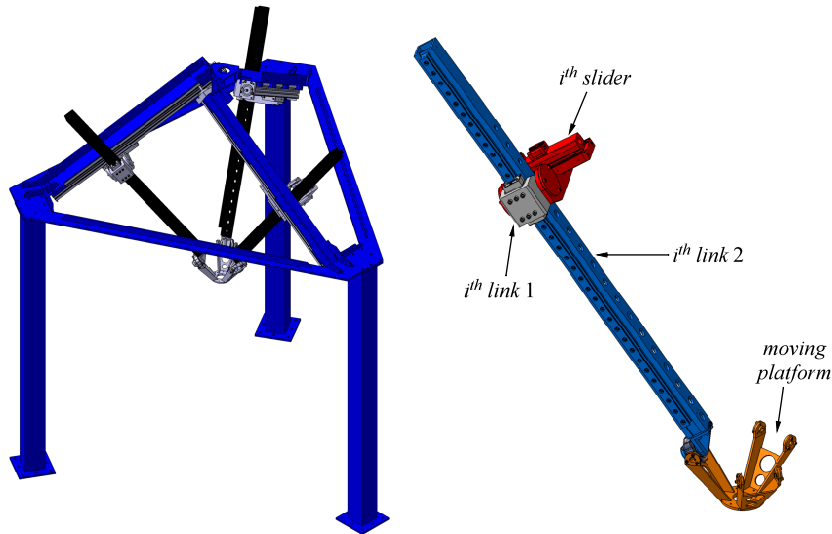


Figure 6. Multibody model of I.Ca.Ro. parallel manipulator.

Table 1. Physical characteristics of the I.Ca.Ro. robot members.

body	c.o.m. [m]	mass [kg]	inertia matrix [kg m ²]
slider	not relevant	5.19	not relevant
link 1	$1 \times 10^{-3} [-0.04 \quad -43.69 \quad 0]^T$	2.62	$\begin{bmatrix} 0.003 & \sim 0 & \sim 0 \\ \sim 0 & 0.004 & \sim 0 \\ \sim 0 & \sim 0 & 0.003 \end{bmatrix}$
link 2	$1 \times 10^{-3} [32.25 \quad 13.16 \quad -552.57]^T$	11.12	$\begin{bmatrix} 1.405 & -4.388 \times 10^{-4} & -0.018 \\ -4.388 \times 10^{-4} & 1.405 & -0.008 \\ -0.018 & -0.008 & 0.008 \end{bmatrix}$
moving platform	not relevant	1.60	not relevant

6 Model verification

In this section a verification of the inverse dynamics model is proposed. With this aim, a multibody model of the 3-CPU pure translational parallel platform has been settled up. Under hypothesis of coherence between the two models in terms of geometrical and mass parameters, a perfect correspondence on actuation forces should be noticed when an identical motion law is used.

The multibody model (see Fig. 6) of the parallel platform is based on a graphical CAD representation of robot members. Definition of joints between the bodies allows the software to reproduce machine kinematic and dynamic behaviour. Each member composing the robot has been measured through mass geometry instruments provided by the CAD environment. In order to improve readers' understanding on the correspondence between multibody model and mathematical model of the platform, Fig. 6 also shows the members of each leg with different colors. A characterization of interesting physical properties of each member is given in Table 1; it is worth to remark that the multibody model has

been built with the maximum respect of the actual I.Ca.Ro. prototype, in order to give a description as much as possible reliable of the mechanical system. For the sake of conciseness, magnitudes that are not useful for dynamic modelling of the manipulator are not reported.

It should be remarked that the inertia matrices expressed for each body refer to a reference frame centred in the centre of mass of and attached to the respective body. Since the model needs these matrices to be expressed with respect to the fixed reference frame $\{0\}$, a coordinates change must be made. Then, for those bodies that are allowed to rotate it is ${}^0\mathbf{I}_i = {}^0\mathbf{R}_i {}^i\mathbf{I}_i {}^0\mathbf{R}_i^T$ where ${}^0\mathbf{R}_i$ denote the orientation of the i -th body with respect to reference frame $\{0\}$.

The physical properties described in Table 1 have been used also for the mathematical model in order to perform a direct comparison with results provided by the multibody environment. As an example, results are reported deriving from a particular set of actuation displacements profiles: a harmonic time history has been chosen for each prismatic joint displacement; each slider is moved with a different frequency. Details on the used functions are shown in Fig. 7.

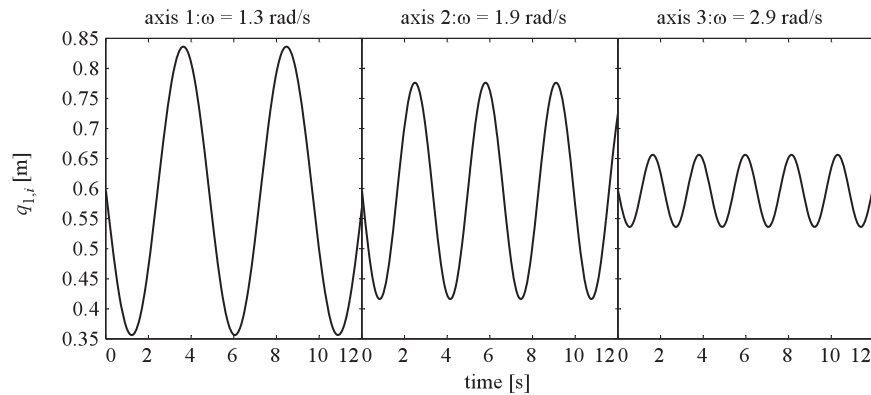


Figure 7. Harmonic displacement profiles used for model verification.

Such motion has been chosen in order to investigate a significant part of the robot workspace, pushing the machine to the physical limits given by the maximum velocity that the three motors are able to perform, which is 0.6 m s^{-1} .

Results of both virtual and mathematical models are used for computation of the relative difference subsisting between the two sets of forces. In particular, named the maximum absolute value of force recorded for motor i during multibody simulations, for the i -th axis it is defined the error ϵ_i as:

$$\epsilon_i = \frac{|\tau_{v,i} - \tau_{m,i}|}{|\tau_{v,i}|_{\max}} \quad (48)$$

where $\tau_{v,i}$ and $\tau_{m,i}$ are the instantaneous forces computed by multibody environment and mathematical model respectively. Equation (48) gives an idea of the deviation between the two methods and therefore it represents a sort of measure of the error introduced by the mathematical model. As visible in Fig. 8, this error never overcomes the 1.0% of the maximum force during simulation, while the average error is always lower than 1%.

According to Eq. (47), the actuation forces evaluated during the previous inverse dynamics simulation can be split in order to analyse the contribution of each part of the robot dynamics. Figure 9 shows the different contributions of the model on the total effort provided by each motor: as well visible, the most part of the force is due to the gravity acceleration acting on robot bodies while a negligible contribution is given by Coriolis and centripetal accelerations. This important information can be used during the realization of simplified mathematical models in which, the contribution of force vector τ_V can be ignored.

7 Simulations results

In this section, simulations are shown in order to test the reliability of the introduced model in actually reproducible conditions.

The first simulation approached is a linear trajectory inside the workspace. The robot, starting from its home posi-

tion, moves to a given point in the space. The trajectory has been planned in order to obtain continuity on platform accelerations. The maximum velocity reached by the manipulator is 0.6 m s^{-1} (which corresponds to the maximum linear velocity available for the actuated joints), while the maximum acceleration is 1.40 m s^{-2} . The starting and the ending points of the motions are 0.5 m apart; it should be remarked that the robot workspace is a cube with a 0.6 m edge. Thus, the trajectory spans a relevant distance with respect to the maximum displacements that the manipulator is able to perform.

For this motion, the forces computed thanks to Eq. (47) are shown in Fig. 10. Also in this case, the most important contribution to motors total effort is given by the gravity acceleration, while the part of force due to Coriolis and centripetal acceleration is negligible.

A second simulation has been performed with a different trajectory in the space. In this case, the end-effector has been moved from its home configuration to a point in the space. From that point, a horizontal circular trajectory (with diameter equal to 0.3 m) centred on robot vertical axis has been performed. Also in this case, the trajectory planning has been performed in order to obtain triangular profiles of acceleration. In this case the maximum velocity reached by the moving platform is 0.6 m s^{-1} , with a maximum acceleration of 0.75 m s^{-2} .

Also for this motion, forces profiles are shown (see Fig. 11). Again, the gravity contribute to the total effort is prevailing with respect to τ_M and τ_V . The influence of τ_V , in particular, represents a negligible contribution to total actuation effort. Nevertheless, Fig. 11 shows that τ_M considerably contributes to the total effort being $|\tau_{M,i}|_{\max} \approx 20\% |\tau_i|_{\max}$.

Given the results of the last simulation, it is interesting to investigate how much the term τ_M affects the dynamic behaviour of the manipulator when the motors are used at their maximum thrust. The robot I.Ca.Ro. is provided with brushless motors, which are able to feed the actuated joints with a maximum force of 420 N. To this aim a simulation has been performed with a circular trajectory, similar to the

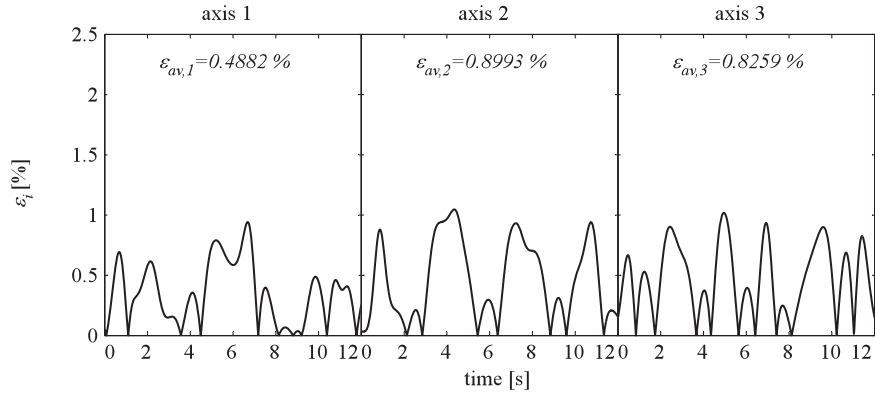


Figure 8. Difference between multibody model and mathematical model.

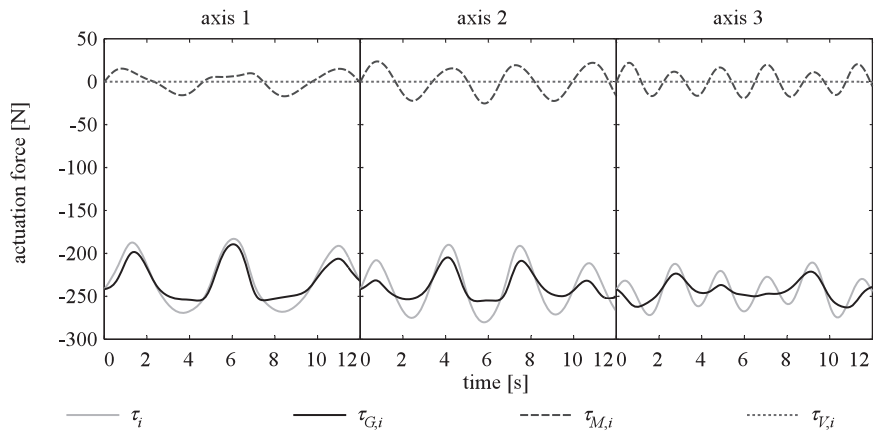


Figure 9. Different contributions to the total actuation efforts.

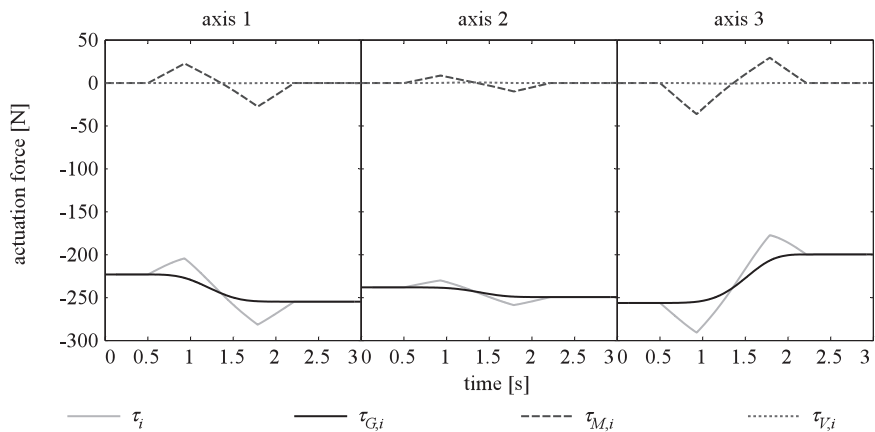


Figure 10. Different contributions to the total actuation efforts during a motion along a line.

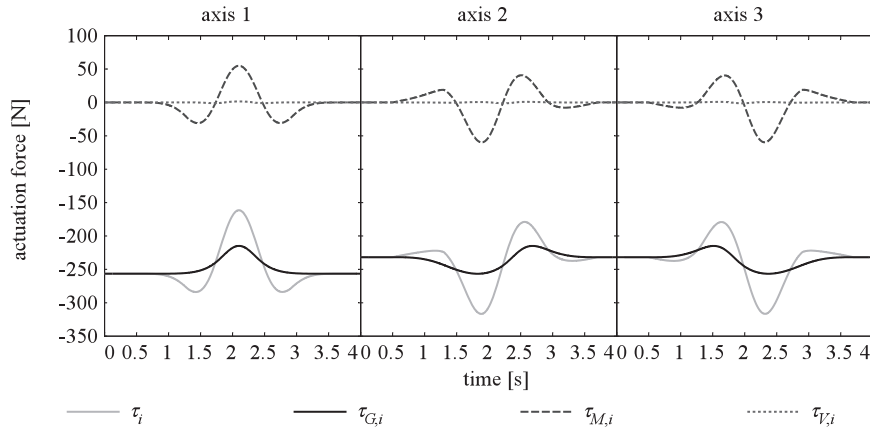


Figure 11. Different contributions to the total actuation efforts during a horizontal circular motion.

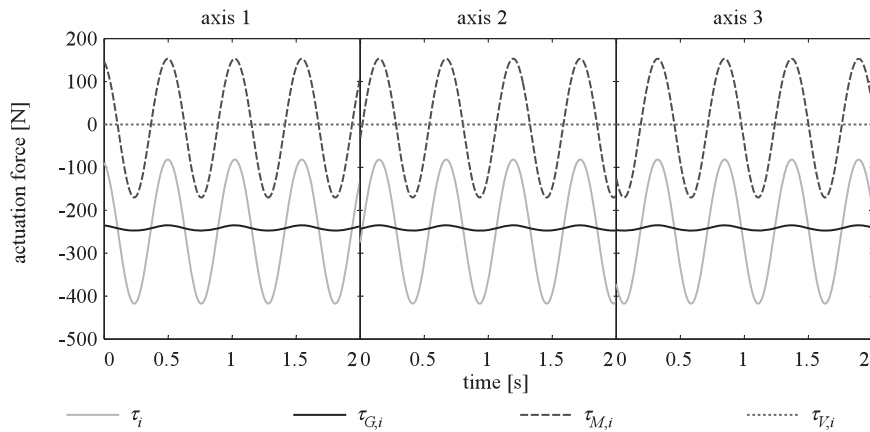


Figure 12. Different contributions to the total actuation efforts during a horizontal circular motion at maximum allowed thrust.

previous one. In this case the circle owns a diameter of 0.1 m and the manipulator is moved with a constant linear speed of 0.6 m s^{-1} . Also the initial velocity has been taken equal to 0.6 m s^{-1} in order to overlook effects due to acceleration ramps. Figure 12 shows that forces τ_M are in this simulation comparable with τ_G demonstrating that a control based on the dynamic model can actually improve the performances of the 3-CPU robot. Moreover, Fig. 12 also confirms that the Coriolis terms are negligible and so they may be omitted in a simplified model (from Eq. 47, $\tau \simeq -\mathbf{M}(\mathbf{q}_a)\ddot{\mathbf{q}}_a - \mathbf{G}(\mathbf{q}_a)$).

Even though the elimination of the Coriolis terms significantly lightens the dynamics formulation, further simplifications can be carried on the matrix \mathbf{M} itself. As an example, it is presented here the results obtained by the elimination of the terms out of the diagonal of such matrix. In particular, Fig. 13 shows the behaviour of τ_M and $\tau_{M'}$ for the circular motion just presented in Fig. 12; τ_M and $\tau_{M'}$ are computed as:

$$\tau_M = \mathbf{M}(\mathbf{q}_a)\ddot{\mathbf{q}}_a \quad \tau_{M'} = \mathbf{M}'(\mathbf{q}_a)\ddot{\mathbf{q}}_a \quad (49)$$

where the matrix \mathbf{M}' is the simplified \mathbf{M} matrix:

$$\mathbf{M}' = \begin{bmatrix} M_{1,1} & 0 & 0 \\ 0 & M_{2,2} & 0 \\ 0 & 0 & M_{3,3} \end{bmatrix} \quad (50)$$

Figure 13 shows that the use of matrix \mathbf{M}' overestimates the effect of the mass matrix on the robot dynamics of a maximum value of 30 N (see curve $\Delta\tau_{M,i}$). The error between τ_M and $\tau_{M'}$ is here estimated as:

$$\epsilon_{M,i} = \frac{|\tau_{M,i} - \tau_{M',i}|}{\max|\tau_{M,i}|} \quad (51)$$

Figure 14 shows the error ϵ_M for each motor: the graphic shows that the error does not overcome the value of 18%.

8 Conclusions

The dynamic modelling of a pure translational PKM has been tackled in this work. Authors proposed a screw based approach for modelling the robot's kinematics, allowing a fast writing of the Jacobian matrices.

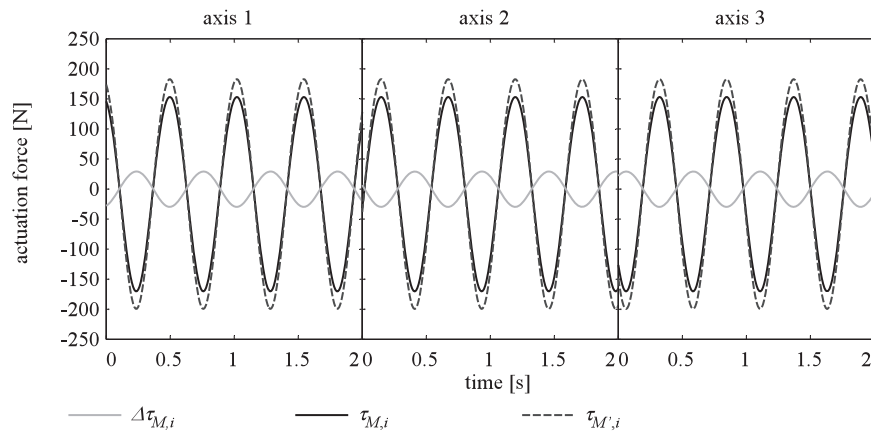


Figure 13. Mass matrix contribution to the dynamics in case of plain and simplified model.

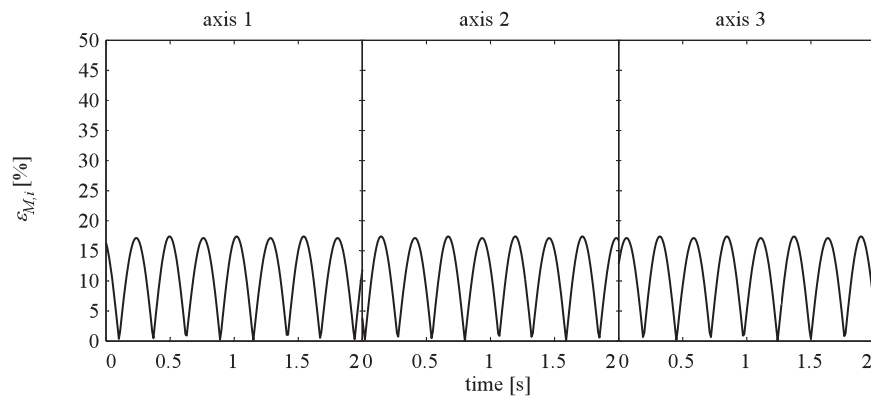


Figure 14. Difference between contributions of full mass matrix \mathbf{M} and diagonalized mass matrix \mathbf{M}' .

The dynamics of the I.Ca.Ro. manipulator was worked out by means of a virtual work principle approach. The resulting model has been verified through simple simulations, taking advance of a multibody model of the robot. Such verification pointed out that the error subsisting between the two virtual models never overcomes the 1.0% of the maximum value of torque involved into the motion.

At last, two simulations have been performed on two trajectories with main aim of investigating the different contributions to the dynamics model. Observation of the results of such simulations yielded a further investigation on the contribution to the whole motors efforts. Such study pointed out that the robot is poorly affected by Coriolis and centrifugal forces while the influence of the mass matrix is not negligible. It is author thought that the compensation of this effect by means of a model based control may improve the performances of robot I.Ca.Ro.

The implementation of the mass matrix on a control algorithm may represent a low efficiency step of a control algorithm because of the heavy mathematical formulation. Due to this, authors also presented a simple simplification of the model based on a simplification of the mass matrix.

Simulations demonstrated that such assumption yield an error that never overcomes the 18% in a situation of high motors stress.

Edited by: A. Tasora

Reviewed by: L. Bruzzone and one anonymous referee

References

- Abdellatif, H. and Heimann, B.: Computational efficient inverse dynamics of 6-DOF fully parallel manipulators by using the Lagrangian formalism, *Mech. Mach. Theory*, 44, 192–207, doi:10.1016/j.mechmachtheory.2008.02.003, 2009.
- Caccavale, F., Siciliano, B., and Villani, L.: The Tri-cept robot: dynamics and impedance control, *Mechatronics*, *IEEE/ASME Transactions on*, 8, 263–268, doi:10.1109/TMECH.2003.812839, 2003.
- Callegari, M. and Palpacelli, M.-C.: Prototype design of a translating parallel robot, *Meccanica*, 43, 133–151, doi:10.1007/s11012-008-9116-8, doi:10.1007/s11012-008-9116-8, 2008.
- Carbonari, L.: Extended analysis of the 3-cpu reconfigurable class of parallel robotic manipulators, Ph.D. thesis, Polytechnic University of Marche, Ancona, Italy, 2012.

- Carbonari, L. and Callegari, M.: The kinematotropic 3-CPU parallel robot: analysis of mobility and reconfigurability aspects, in: *Latest Advances in Robot Kinematics*, edited by: Lenarčič, J. and Husty, M., Springer, 2012.
- Carbonari, L., Callegari, M., Palmieri, G., and Palpacelli, M.: A new class of reconfigurable parallel kinematics machines, *Mech. Mach. Theory*, in review, 2013.
- Dasgupta, B. and Mruthyunjaya, T.: A Newton-Euler formulation for the inverse dynamics of the Stewart platform manipulator, *Mech. Mach. Theory*, 33, 1135–1152, doi:10.1016/S0094-114X(97)00118-3, 1998.
- Daun, Q., Daun, B., and Daun, X.: Dynamics modelling and hybrid control of the 6-UPS platform, in: *2010 International Conference on Mechatronics and Automation (ICMA)*, 434–439, doi:10.1109/ICMA.2010.5589105, 2010.
- Di Gregorio, R. and Parenti-Castelli, V.: Dynamics of a Class of Parallel Wrists, *J. Mech. Design*, 126, 436–441, doi:10.1115/1.1737382, 2004.
- Do, W. Q. D. and Yang, D. C. H.: Inverse dynamic analysis and simulation of a platform type of robot, *J. Robotic Syst.*, 5, 209–227, doi:10.1002/rob.4620050304, 1988.
- Gallardo, J., Rico, J., Frisoli, A., Checcacci, D., and Bergamasco, M.: Dynamics of parallel manipulators by means of screw theory, *Mech. Mach. Theory*, 38, 1113–1131, doi:10.1016/S0094-114X(03)00054-5, 2003.
- Khalil, W. and Ibrahim, O.: General Solution for the Dynamic Modeling of Parallel Robots, *J. Intell. Robotics Syst.*, 49, 19–37, doi:10.1007/s10846-007-9137-x, 2007.
- Kovecses, J., Piedboeuf, J. C., and Lange, C.: Dynamics modeling and simulation of constrained robotic systems, *Mechatronics*, *IEEE/ASME Transactions on*, 8, 165–177, doi:10.1109/TMECH.2003.812827, 2003.
- Kunquan, L. and Rui, W.: Closed-form Dynamic Equations of the 6-RSS Parallel Mechanism through the Newton-Euler Approach, in: *2011 Third International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 1, 712–715, doi:10.1109/ICMTMA.2011.180, 2011.
- Lin, Y.-J. and Song, S.-M.: A comparative study of inverse dynamics of manipulators with closed-chain geometry, *J. Robotic Syst.*, 7, 507–534, doi:10.1002/rob.4620070402, 1990.
- Lopes, A. M.: Dynamic modeling of a Stewart platform using the generalized momentum approach, *Commun. Nonlinear Sci.*, 14, 3389–3401, doi:10.1016/j.cnsns.2009.01.001, 2009.
- Miller, K.: Optimal Design and Modeling of Spatial Parallel Manipulators, *Int. J. Robot. Res.*, 23, 127–140, doi:10.1177/0278364904041322, 2004.
- Moon, F.: *Applied Dynamics: With Applications to Multibody and Mechatronic Systems*, John Wiley & Sons, 2008.
- Papastavridis, J.: *Analytical Mechanics: A Comprehensive Treatise on the Dynamics of Constrained Systems*, World Scientific, 2012.
- Tsai, L.-W.: Solving the Inverse Dynamics of a Stewart-Gough Manipulator by the Principle of Virtual Work, *J. Mech. Design*, 122, 3–9, doi:10.1115/1.533540, 2000.
- Wang, J. and Gosselin, C. M.: A New Approach for the Dynamic Analysis of Parallel Manipulators, *Multibody Syst. Dyn.*, 2, 317–334, doi:10.1023/A:1009740326195, 1998.
- Wang, J., Wu, J., Wang, L., and Li, T.: Simplified strategy of the dynamic model of a 6-UPS parallel kinematic machine for real-time control, *Mech. Mach. Theory*, 42, 1119–1140, doi:10.1016/j.mechmachtheory.2006.09.004, 2007.
- Wronka, C. and Dunnigan, M.: Derivation and analysis of a dynamic model of a robotic manipulator on a moving base, *Robot. Auton. Syst.*, 59, 758–769, doi:10.1016/j.robot.2011.05.010, 2011.
- Yang, C., Han, J., Zheng, S., and Ogbobe Peter, O.: Dynamic modeling and computational efficiency analysis for a spatial 6-DOF parallel motion system, *Nonlinear Dynam.*, 67, 1007–1022, doi:10.1007/s11071-011-0043-1, 2012.
- Zhang, C.-D. and Song, S.-M.: An efficient method for inverse dynamics of manipulators based on the virtual work principle, *J. Robotic Syst.*, 10, 605–627, doi:10.1002/rob.4620100505, 1993.