Dynamic modeling of planar parallel robots considering passive joint sensor data

Asier Zubizarreta, Itziar Cabanes, Marga Marcos and Charles Pinto

Robotica / Volume 28 / Issue 05 / September 2010, pp 649 - 661
DOI: 10.1017/S0263574709990300, Published online: 11 August 2009

Robotica (2010) volume 28, pp. 649–661. © Cambridge University Press 2009
doi:10.1017/S0263574709990300

Dynamic modeling of planar parallel robots considering passive joint sensor data

Asier Zubizarreta†, Itziar Cabanes†*, Marga Marcos† and Charles Pinto‡

†*Department of Automatic Control and System Engineering, University of the Basque Country, Spain*
‡*Department of Mechanical Engineering, University of the Basque Country, Spain*

(Received in Final Form: June 24, 2009. First published online: August 11, 2009)

SUMMARY

Model-based advanced control approaches are needed to achieve high speed and acceleration and precision in robotic operations. These control schemes need a proper dynamic model. Many approaches have been proposed by different authors in order to obtain the dynamic model of these structures. However, most of them do not consider the possibility to introduce redundant sensor data. In this paper, a methodology for obtaining a compact dynamic model considering passive joint sensor data is proposed. The dynamic model is defined in compact and structured form, which makes it appropriate to be used in advanced control techniques.

1. Introduction

Parallel robots, mechanisms whose end-effector is joined to the base by more than one kinematic chain, have became an interesting alternative to classical serial robots. Higher payload and stiffness and higher operating speed and accuracy are some of the advantages of this structures. In order to take the most of these advantages, great effort has been made recently to design specific mechanisms for particular tasks. However, their complex structure presents some disadvantages, such as highly coupled kinematics and dynamics, small workspace, presence of internal singularities, and presence of unactuated or passive joints among others.[24]

Extensive research has been made to correct these disadvantages. Most of the authors have focused their efforts in analyzing the complex kinematics of these mechanisms. However, dynamics has been addressed by fewer authors due to the complexity of its formulation.

Advanced control techniques can be used to fully exploit the advantages of parallel robots in terms of high speed and accuracy. These control approaches can increase significantly the performance of parallel robots if an accurate dynamic model is calculated. This makes dynamics an important area of study when considering control performance of parallel robots. In this paper, direct and inverse dynamic modelling of planar parallel robots is formulated and then this formulation is applied to the 3RRR parallel robot.

* Corresponding author. E-mail: itziar.cabanes@ehu.es

In serial robotics literature, Lagrangian and Newton–Euler formulations have been widely used to calculate the dynamic model of robots,[3] as Eq. (1) illustrates:

$$\boldsymbol{\tau} = D(\boldsymbol{q}) \cdot \ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}}) \cdot \dot{\boldsymbol{q}} + G(\boldsymbol{q}). \tag{1}$$

where $\boldsymbol{\tau}$ is the vector of the actuator torque/forces, $\boldsymbol{q}$ is the vector of the n articular coordinates, D is the symmetric and positive-defined inertia matrix, C is the Coriolis matrix, where $(\dot{D} - 2C)$ is skew-symmetric, and G is the gravity terms vector. Friction terms have been neglected in this model, although they can be introduced in it as an additional term. This model has been used in the literature to implement multi-articular model-based control techniques, such as computed torque control and its properties ensure global stability of the control system.[7]

However, obtaining a model with similar characteristics in parallel robots is not an easy task due to the additional kinematic constraints that relate the joint variables. Opposite to serial robots, in parallel robots the numbers of degrees of freedom (dof) is less than the number of joint variables, which means that some of the joints are passive or nonactuated. This increases the complexity of the kinematic and dynamic problems. Some authors have proposed the use of sensors in passive joints to simplify kinematic problems: Merlet[23] determined the location of the minimum number of extra sensors that have to be introduced in a Gough–Stewart platfrom in order to calculate its direct kinematics in a closed form; using the same platform, Baron and Angeles, in ref. [2], describe a method that allows to decouple the translation and orientation kinematics using sensor redundancy. Other authors have tried to increase control performance: Marquet *et al.*[22] introduced an extra sensor in the pick and place H4 robot, and with the use of a redundant kinematic model, they implemented a proportional-integral derivative (PID) based control which reduces the errors in a 45 % respect to its nonredundant counterpart. Using the same idea, Bauma *et al.*, in ref. [4], use the redundant sensors in order to increase the position accuracy for various platforms. However, as far as the authors know, no work has dealt with introducing extra sensor data in the inverse dynamic model (IDM).

The introduction of extra passive sensors in the structure of the mechanism can be difficult and increases its cost. However, as it is shown in the previously cited works, its

advantages can compensate these drawbacks. This work shows the increase of control performance of a novel computed torque control (CTC) approach using a redundant dynamic model. In this novel CTC scheme, the control signal is computed using both active and extra sensorized passive joint data. This scheme proves to be more robust than the classical nonredundant CTC approach and can significantly reduce the positioning error of the parallel robot.

Therefore, redundant model-based control schemes proves to be an interesting approach for the control of parallel robots. This approach, however, requires the definition of the dynamic model in terms of both active and some extra sensorized passive joints. In this paper, the formulation of the dynamic model of parallel robots for the implementation of redundant control approaches is presented.

Three main classical approaches have been proposed to obtain the dynamic model of parallel robots: Lagrangian formulation, the use of Newton–Euler laws, and the use of the principle of virtual work. The next sections analyze the most interesting works in each approach.

1.1. Newton–Euler formulation

Traditional Newton–Euler (N-E) formulation requires the definition of the equations of each body of the mechanical system. This way, all the internal and external forces acting in each body of the mechanism are calculated, leading to a high-dimension equation system. In the literature, most of the authors have applied this method to particular mechanisms, especially to the Gough platform. That is the case of Do and Yang[10] that obtain a system of 36 linear equations that represent the platform dynamics. Dasgupta and Mrythyunjaya[9] also calculate the dynamic model of this robot, but propose a method to avoid calculating the internal reactions of the mechanism, in order to reduce the computational cost. However, the model is not obtained in a structured form. Later, Riebe and Ulbrich[28] use the Jacobian matrices of the mechanism bodies to project all the equations of motion into the cartesian space. The authors calculate the IDM defined in the end-effector cartesian coordinates, with the structure of Eq. (1), and use it to implement a CTC technique.

On the other hand, other authors have tried to deduce generic algorithms to calculate the dynamic model of general parallel mechanisms. As most of parallel mechanisms are composed by a mobile platform and a group of serial chains connecting it to the base, a popular strategy consists on splitting the mechanism in these two subsystems. This way, Dasgupta and Choudhurry[8] calculate the robot dynamics in two steps: first, the dynamics of the legs are considered in order to calculate the reaction force on the mobile platform using N-E laws. Then, the dynamics of the full mechanism is obtained applying N-E formulation to the mobile platform. A similar strategy is proposed by Khalil and Guegan[17] and later extended by Khalil and Ibrahim[18]. In this approach, reaction forces between the legs and the mobile platform are first calculated using the dynamic model of the serial subsystem. Then, these forces are projected via the legs Jacobian matrices to the mobile platform coordinate system. Finally, applying the N-E formulation to the platform, the full dynamic model is obtained. A similar strategy is applied by Guo and Li[15] to the Gough–Stewart platform. The same idea is used by Angeles in refs. [1, 33], where the coupling of the models of the serial chain subsystem and the mobile platform is made via the natural ortogonal complement that maps the twist of the mobile platform in the actuated articular coordinate space.

1.2. Principle of virtual work

One of the most popular approach to obtain the dynamic model of parallel robots is the application of the principle of virtual work. In this formulation, an energetic approach is used.

Some authors have used directly this principle to calculate the dynamic model of specific parallel robots. That is the case of Zhu *et al.*[35] that obtained the dynamic model of Tau parallel robot. Similarly, Staicu *et al.*[29] calculated the dynamic model of a spatial 3 d.o.f. parallel manipulator. Zhiyong *et al.*[34] applied it to the 5R parallel robot. Honegger *et al.*[16] used it to model the Hexaglide parallel robot. Wang and Gosselin[32] and later Tsai[31] obtained the dynamic model for the Gough platform using this principle. This approach showed an increase of 30 % in the efficiency of the algorithm compared to the N-E approach.

On the other hand, there is another important trend based on the use of the reduced model concept and the principle of virtual work. This method considers parallel robots as the combination of a set of free elements, called the *free system*, and a number of constraints. The works in this area focus on obtaining the relation between the dynamic model of the free system without the restrictions and the dynamic model of the constrained system. The formulations proposed by Nakamura and Godoussi[26], Murray and Lovell[25] and Ghorbel *et al.*[11–14] define this relationship in a similar way, defining a transformation matrix T that relates the free system model $\boldsymbol{\tau_r}$ and the constrained model $\boldsymbol{\tau}$, $\boldsymbol{\tau} = T^T \cdot \boldsymbol{\tau_r}$. Thus, the constraints are implicitly introduced in T.

1.3. Lagrange formulation

The Lagrangian Formulation has been widely used in serial robotics to calculate the dynamic model. It is an energy-based approach and, with a proper choice of the generalized coordinates, it avoids the calculation of the internal forces of the mechanism. This way, the reaction forces between elements introduced in the N-E approach are not calculated and the computational efficiency can be increased. The Lagrangian method requires the definition of the Lagrangian, which is calculated by subtracting the potential energy of the system from the kinetic energy. Then the Lagrangian is differentiated respect to the independent set of coordinates of the system, and the dynamic model is obtained. This, obviously, requires that the model is defined only in terms of these independent coordinates. However, in parallel robots not all the joints are actuated, and the relation between passive or non-actuated joints and active ones is, in general, difficult to formulate explicitly. Therefore, a set of Lagrange multipliers and geometric or kinematic constraints are introduced in order to simplify the resolution of the problem. Thus, the number of the dynamic equations increases but the relations between independent and non independent coordinates are implicitly calculated.

Using this formulation, Lee and Shah[20] calculated the dynamic model of a 3-dof tripod. Later, Lebret *et al.*[19] obtained the dynamic model of the Gough platform in terms of the end-effector cartesian coordinates without using Lagrange multipliers. An interesting approach was proposed by Bhattacharya *et al.*[5], and later extended in ref. [6], where, as previously explained, the parallel robot is split in two subsystems: mobile platform and serial chain subsystems. In this approach, two sets of Lagrangian equations are constructed, one for each subsystem. The two sets are then joined using the Jacobian matrix of the robot.

The three main approaches proposed in the literature provide a wide variety of ways to solve the dynamic problem. However, most of them do not consider the possibility of including passive joint sensor data explicitly in the dynamic model, in order to apply advanced control techniques based on sensor redundancy as the one presented in ref. [36]. The dynamic model explicitly defined in terms of active and some extra sensorized passive joints is believed to be more robust against model parameter uncertainties. Additionally, as passive joints are directly measured, the computational cost of calculating the model is reduced compared with the model defined only in terms of actuated joints. In this paper, a Lagrangian formulation approach is proposed. This formulation allows to define the dynamic model in terms of whatever generalized coordinates the designer selects, which allows to calculate easily the dynamic model in terms of redundant coordinates. Additionally, the approach avoids the calculation of internal forces and, although it requires complex derivations, these can be easily automatized with the help of symbolic computation. The technique proposed in this paper provides a guide to obtain the dynamic model of planar parallel robots to be used in model-based control approaches, considering either actuated or active joints and some sensorized passive joints.

The rest of the paper is organized as follows: in Section 2, the proposed formulation is introduced and its basis described. Section 3 introduces the guidelines for the dynamic modelling of planar parallel robots considering either active and sensorized passive joints. Finally, Section 4 illustrates the methodology by applying it to the 3RRR parallel robot. The calculated model is used to implement a model-based control approach, which is validated by a set of simulations. The most important ideas of the proposed approach are summarized in Section 5.

2. Dynamic Modeling Based on the Lagrangian Formulation

The proposed approach for the dynamic modelling of planar parallel robots is based on three previous contributions, explained in Section 1:

- Use of the transformation matrix T defined in Section 1.2, as proposed in refs. [11–14, 25, 26].
- Consideration of the mechanism as a two subsystem set: moving platform and serial chain subsystem, as presented in refs. [1, 17, 18].
- Application of the Lagrangian formulation with Lagrange multipliers to each subsystem independently, as considered in refs. [5, 6].

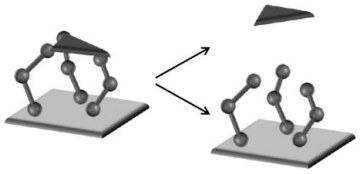

Fig. 1. Serial chain and platform subsystems.

The use of reduced models is an easy way to obtain the dynamic model of parallel robots when there is not a moving platform, as in the case of the 5R planar parallel robot. However, in the general case, the moving platform must be attached to one of the serial subchains, increasing the complexity of the problem. To avoid this, splitting the mechanism into a platform and a serial chain subsystems is proposed (Fig. 1). Besides, the use of the Lagrangian Formulation allows the designer to choose the most suitable generalized coordinates for the design of the dynamic model. Thus, it is possible to consider passive joints directly in the formulation.

Based on these three main ideas, a methodology for obtaining the dynamic model of planar parallel robots is proposed, where it is possible to consider only active joints, or both active and some passive sensorized joints. The resulting model is intended to be used to implement advanced control techniques in order to improve the accuracy and trajectory-tracking of the robot as it is demonstrated in ref. [36].

Next, the guidelines for easily obtaining the dynamic model of a generic planar parallel robot are given.

3. Dynamic Model Considering Passive and Active Joints

Suppose a mechanism with $n = n_a \leq 3$ dof, whose mobile platform is joined to the base by n serial chains.

Let $\boldsymbol{q_a}$ be the set of *active joints* of the mechanism, of dimension n, $\boldsymbol{q_p}$ the set of the n_p *passive joints* of the mechanism and $\boldsymbol{q_s}$ the set of the n_s *passive sensorized* joints. Let the set of *control coordinates* be $\boldsymbol{q} = [\boldsymbol{q_a} \;\; \boldsymbol{q_s}]^T$ and the set of $n_r = n + n_p + n_s$ *generalized coordinates* $\boldsymbol{q_r} = [\boldsymbol{q} \;\; \boldsymbol{q_p}]^T$. Let $\boldsymbol{x}$ be the n dimensional independent *task coordinates* of the end-effector.

The choice of the sensorized coordinates is determined by the specific parallel robot to be modelled. Although it is difficult to determine precise rules to make this choice, some guidelines are described next:

- In general, parallel robots are composed by the end-effector mobile platform and some serial chains joining it to the base. Sensorizing the same passive joints on each serial chain provides, in general, a symmetric dynamic model with high robustness.
- Sensors have to be placed in each kinematic chain so that the combined information of the active and passive sensors result in an increase of the accuracy of the Tool Center Point (TCP) positioning.
- In general, only revolute (R) and prismatic (P) passive joints will be sensorized, as there exist multiple and relatively low-cost sensor solutions for the measurement of the position and speed of these joints.

Next, the guidelines for calculating the dynamic model are introduced. As the dynamic model will be used for real-time control purposes, a simplified approach will be considered, neglecting the effects of friction and element elasticity.

3.1. Closure loop equations of the mechanism

Initially, a set of independent closure loop equations have to be defined. These equations characterize the constraints of the mechanism and the relation between the different coordinates. Thus, their definition is an important step in the dynamic modelling of a parallel robot.

In order to relate the different sets of coordinates of the robot, three different sets of closure loop equations have to be defined:

- The n independent closure loop equations that relate the task coordinates $\boldsymbol{x}$ and the active articular coordinates $\boldsymbol{q_a}$

$$\phi_i(\boldsymbol{x}, \boldsymbol{q_a}) = 0, i = 1, \ldots n, \tag{2}$$

- The closure loop equations that relate $\boldsymbol{x}$ and the generalized coordinates $\boldsymbol{q_r}$

$$\Gamma(\boldsymbol{x}, \boldsymbol{q_r}) = \boldsymbol{0} \in \mathbb{R}^{n_r} \rightarrow \boldsymbol{x} = f_i(\boldsymbol{q_r}), i = 1, \ldots n \tag{3}$$

  $\Gamma(\boldsymbol{x}, \boldsymbol{q_r})$ can be easily calculated formulating the vector equations that link the TCP with the origin of the cartesian frame.
- The closure loop equations relating active joints $\boldsymbol{q_a}$ and passive $\boldsymbol{q_p}$ and sensorized joints $\boldsymbol{q_s}$.
  In general, it is easier to define these parameters in terms of active joints and the independent task coordinates:

$$\begin{matrix} f_s(\boldsymbol{q_a}, \boldsymbol{x}, \boldsymbol{q_s}) = \boldsymbol{0} \\ f_p(\boldsymbol{q_a}, \boldsymbol{x}, \boldsymbol{q_p}) = \boldsymbol{0} \end{matrix}. \tag{4}$$

3.2. Position problem

Making use of Eqs. (2) and (3) the direct and inverse position problem can be calculated. In the general case, a closed form solution to the direct kinematics problem will not be available. In that case, numerical methods such as Newton–Raphson method, interval analysis techniques or iterative geometric approaches like Geometrical-Iterative Method GIM[27] can be used to solve the direct kinematic problem. Making use of Eq. (4), the positions of passive parameters can be obtained.

3.3. Velocity problem

The velocity problem can be calculated directly taking the derivative of the closure loop equations, which is an easy task in simple planar robots. Alternatively, it is possible to directly calculate the vectorial velocity equations from the mechanism, which is easier in more complex robots.

From the velocity equations of the mechanism, Jacobian matrices can be obtained. These include implicitly the constraints of the mechanism and will be a valuable asset to map the dynamic model into different coordinate systems. Thus, the objective of this step is to calculate the following relations.

3.3.1. Jacobian matrix of the robot J_R.

It is defined as

$$\dot{\boldsymbol{x}} = \underbrace{J_R}_{n \times n}(\boldsymbol{x}, \boldsymbol{q_a}) \cdot \dot{\boldsymbol{q}}_{\boldsymbol{a}}. \tag{5}$$

This expression is usually obtained by differentiating Eq. (2):

$$\begin{matrix} \frac{\partial \phi(\boldsymbol{x}, \boldsymbol{q_a})}{\partial \boldsymbol{x}} \cdot \dot{\boldsymbol{x}} + \frac{\partial \phi(\boldsymbol{x}, \boldsymbol{q_a})}{\partial \boldsymbol{q_a}} \cdot \dot{\boldsymbol{q}}_{\boldsymbol{a}} = 0 \\ \underbrace{J_{xa}}_{n \times n}(\boldsymbol{x}, \boldsymbol{q_a}) \cdot \dot{\boldsymbol{x}} + \underbrace{J_{q_a}}_{n \times n}(\boldsymbol{x}, \boldsymbol{q_a}) \cdot \dot{\boldsymbol{q}}_{\boldsymbol{a}} = 0 \\ \dot{\boldsymbol{x}} = -J_{xa}^{-1} \cdot J_{q_a} \cdot \dot{\boldsymbol{q}}_{\boldsymbol{a}} = J_R \cdot \dot{\boldsymbol{q}}_{\boldsymbol{a}} \end{matrix}. \tag{6}$$

3.3.2. Constraint Jacobian matrix J_C.

It relates the velocities of the task coordinates of the mobile platform $\dot{\boldsymbol{x}}$ and the velocities of the generalized coordinates of the serial subsystem $\dot{\boldsymbol{q}}_{\boldsymbol{r}}$. It is obtained by differentiating Eq. (3) or by defining directly the vectorial velocity equations of each kinematic loop. The objective is to obtain an expression:

$$\boldsymbol{v} = J_C(\boldsymbol{x}, \boldsymbol{q_r}) \cdot \dot{\boldsymbol{q}}_{\boldsymbol{r}}, \tag{7}$$

where $J_C(\boldsymbol{x}, \boldsymbol{q_r}) \in \mathbb{R}^{n \times (n + n_s + n_p)}$.

A general procedure to calculate J_C follows: let $\boldsymbol{q_i}$ be the coordinates of the ith serial subchain $i = 1, \ldots n$, composed by active, passive, and sensorized joints, then, it is easy to obtain a relation such as:

$$\dot{\boldsymbol{q}}_{\boldsymbol{i}} = J_i \cdot \boldsymbol{v}, i = 1, \ldots n, \tag{8}$$

where J_i is the Jacobian of i-th the serial subchain. If Eq. (8) is calculated for each chain and its terms are reordered,

$$\begin{matrix} \dot{\boldsymbol{q}}_{\boldsymbol{a}} = \underbrace{J_{q_a}}_{n \times n} \cdot \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{q}}_{\boldsymbol{s}} = \underbrace{J_{q_s}}_{n_s \times n} \cdot \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{q}}_{\boldsymbol{p}} = \underbrace{J_{q_p}}_{n_p \times n} \cdot \dot{\boldsymbol{x}} \end{matrix}. \tag{9}$$

So,

$$\dot{\boldsymbol{q}}_{\boldsymbol{r}} = [\, J_{q_a} \; J_{q_s} \; J_{q_p} \,]^T_{n_r \times n} \cdot \dot{\boldsymbol{x}} = \underbrace{J_C^{inv}}_{n_r \times n} \cdot \dot{\boldsymbol{x}}. \tag{10}$$

In order to calculate J_C, J_C^{inv} has to be inverted. For that purpose the Moore–Penrose generalized inverse (11) is used. This inverse is generally implemented using an orthogonalization algorithm such as QR decomposition:

$$A^{\dagger} = (A^T \cdot A)^{-1} \cdot A^T. \tag{11}$$

So,

$$\dot{\boldsymbol{x}} = (J_C^{inv})^{\dagger} \cdot \dot{\boldsymbol{q}}_{\boldsymbol{r}} = \underbrace{J_C}_{n \times n_r} \cdot \dot{\boldsymbol{q}}_{\boldsymbol{r}}. \tag{12}$$

Alternatively, it can be obtained on a similar way from that proposed for the Jacobian matrix of the robot J_R. However, as the partial Jacobians are not square, the Moore–Penrose

generalized inverse will have to be used in order to calculate J_C.

3.3.3. Transformation matrix T. The transformation matrix T, relates the active joints $\dot{\boldsymbol{q}}_{\boldsymbol{a}}$ and the generalized coordinates of the serial subsystem $\dot{\boldsymbol{q}}_{\boldsymbol{r}}$:

$$\dot{\boldsymbol{q}}_{\boldsymbol{r}} = \begin{bmatrix} \dot{\boldsymbol{q}}_{\boldsymbol{a}} \\ \dot{\boldsymbol{q}}_{\boldsymbol{s}} \\ \dot{\boldsymbol{q}}_{\boldsymbol{p}} \end{bmatrix} = [\, I_{n\times n} \;\; J_s \;\; J_p \,]^T \dot{\boldsymbol{q}}_{\boldsymbol{a}} = T \cdot \dot{\boldsymbol{q}}_{\boldsymbol{a}}, \tag{13}$$

where $J_s \in \mathbb{R}^{n_s \times n}$ and $J_p \in \mathbb{R}^{n_p \times n}$ are the Jacobian matrices relating the sensorized $\boldsymbol{q}_{\boldsymbol{s}}$ and passive $\boldsymbol{q}_{\boldsymbol{p}}$ joints of the robot and the active ones $\boldsymbol{q}_{\boldsymbol{a}}$. These matrices can be obtained by differentiating the closure loop equations (4) or calculating the vectorial velocity equations of each kinematic loop.

For planar robots, these Jacobians are, in general, easy to obtain. However, a more general procedure is proposed next.

Using the previously calculated Eq. (9) and the Jacobian matrix of the robot J_R (5):

$$\begin{aligned} \dot{\boldsymbol{q}}_{\boldsymbol{s}} &= J_{q_s} \cdot \dot{\boldsymbol{x}} = J_{q_s} \cdot J_R \dot{\boldsymbol{q}}_{\boldsymbol{a}} = \underbrace{J_s}_{n_s \times n} \cdot \dot{\boldsymbol{q}}_{\boldsymbol{a}} \\ \dot{\boldsymbol{q}}_{\boldsymbol{p}} &= J_{q_p} \cdot \dot{\boldsymbol{x}} = J_{q_p} \cdot J_R \dot{\boldsymbol{q}}_{\boldsymbol{a}} = \underbrace{J_p}_{n_p \times n} \cdot \dot{\boldsymbol{q}}_{\boldsymbol{a}} \cdot \end{aligned} \tag{14}$$

3.3.4. Transformation matrix T_q. This transformation matrix relates the control coordinates $\dot{\boldsymbol{q}}$ and the generalized coordinates of the serial subsystem $\dot{\boldsymbol{q}}_{\boldsymbol{r}}$. This way, the serial subsystem model defined in terms of $\dot{\boldsymbol{q}}_{\boldsymbol{r}}$ can be projected to the space defined by the control coordinates:

$$\begin{aligned} \dot{\boldsymbol{q}}_{\boldsymbol{r}} &= \begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{q}}_{\boldsymbol{p}} \end{bmatrix} = \left[\, I_{(n+n_s)\times(n+n_s)} \;\; J_{q_p/q} \,\right]^T \dot{\boldsymbol{q}} \\ &= T_q \cdot \dot{\boldsymbol{q}}, \end{aligned} \tag{15}$$

where $J_{q_p/q} = \frac{\partial \boldsymbol{q}_{\boldsymbol{p}}}{\partial \boldsymbol{q}}$ is the Jacobian matrix relating the passive joints and the control coordinates. These expressions are easy to calculate directly defining a closure loop equation that relates all these coordinates. However, in more complex mechanisms, the previously defined velocity equations can be used. Reordering Eq. (8):

$$\begin{aligned} \dot{\boldsymbol{q}} &= \underbrace{J_q}_{(n_s+n)\times n} \cdot \boldsymbol{v} \\ \dot{\boldsymbol{q}}_{\boldsymbol{p}} &= \underbrace{J_{q_p}}_{(n_p)\times n} \cdot \boldsymbol{v} \end{aligned} \quad . \tag{16}$$

In order to eliminate $\dot{\boldsymbol{x}}$ from Eq. (16) the Moore–Penrose generalized inverse (11) is used. So $J_{q_p/q}$ is defined as

$$J_{q_p/q} = J_{q_p} \cdot J_q^{\dagger}. \tag{17}$$

3.4. Acceleration problem

The acceleration equations are obtained by differentiating the velocity Eqs. (7), (13) and (15). The main interest in this section is to determine the three following relations:

- Relation between the cartesian coordinate acceleration $\ddot{\boldsymbol{x}}$ and the generalized coordinate acceleration $\ddot{\boldsymbol{q}}_{\boldsymbol{r}}$

$$\ddot{\boldsymbol{x}} = J_C \cdot \ddot{\boldsymbol{q}}_{\boldsymbol{r}} + \dot{J}_C \cdot \dot{\boldsymbol{q}}_{\boldsymbol{r}} \tag{18}$$

- Relation between the generalized coordinate acceleration $\ddot{\boldsymbol{q}}_{\boldsymbol{r}}$ and the active joint acceleration $\ddot{\boldsymbol{q}}_{\boldsymbol{a}}$

$$\ddot{\boldsymbol{q}}_{\boldsymbol{r}} = T \ddot{\boldsymbol{q}}_{\boldsymbol{a}} + \dot{T} \dot{\boldsymbol{q}}_{\boldsymbol{a}} \tag{19}$$

  where $\dot{T} = [\, 0_{n\times n} \;\; \dot{J}_s \;\; \dot{J}_p \,]^T$.
- Relation between the generalized coordinate acceleration $\ddot{\boldsymbol{q}}_{\boldsymbol{r}}$ and the control coordinate acceleration $\ddot{\boldsymbol{q}}$

$$\ddot{\boldsymbol{q}}_{\boldsymbol{r}} = T_q \ddot{\boldsymbol{q}} + \dot{T}_q \dot{\boldsymbol{q}} \tag{20}$$

  where $\dot{T}_q = [0_{(n+n_s)\times(n+n_s)} \;\; \dot{J}_{q_p/q} \,]^T$.

3.5. Lagrangian equation

Once the necessary kinematic relations have been obtained (Sections 3.1–3.4), the dynamic model of the parallel robot can be calculated.

The Lagrangian formulation is used to obtain the dynamic model of the parallel robot. For that purpose, the mechanism will be split in two subsystems as explained before. That way, the Lagrangian equation of the mobile platform L_p and the serial chain system L_s will be calculated and the dynamic model of each subsystem deduced. Note that while the mobile platform equations will be defined in terms of the task coordinates $\boldsymbol{x}$, the n serial chain subsystem equations will be defined in terms of the generalized articular coordinates $\boldsymbol{q}_{\boldsymbol{r}}$:

$$\begin{aligned} L &= \sum_{i=1}^{n} L_{s_i}(\boldsymbol{q}_{\boldsymbol{r}}, \dot{\boldsymbol{q}}_{\boldsymbol{r}}) + L_p(\boldsymbol{x}, \dot{\boldsymbol{x}}) \\ &= L_s(\boldsymbol{q}_{\boldsymbol{r}}, \dot{\boldsymbol{q}}_{\boldsymbol{r}}) + L_p(\boldsymbol{x}, \dot{\boldsymbol{x}}) \end{aligned} . \tag{21}$$

3.6. Differential equations of the mobile platform and serial chain subsystems

Once the Lagrangian has been defined, the differential equations associated to each subsystem can be calculated. As the Lagrangian of the mobile platform L_p and the serial chain subsystem L_s are defined in different coordinates, two sets of differential equations can be defined, one for each subsystem. Thus, the model will be defined in terms of $n + n_r$ coordinates, so, as only n coordinates are independent, n_r Lagrange multipliers will have to be introduced in order to calculate the model[30]. This way, in order to model the constraints that join the serial chains to the platform, the closure loop equation relating the coordinates defining each subsystem (3) and n_r Lagrange multipliers λ_i are introduced:

- Dynamic equations associated to the serial chain subsystem $j = 1, \ldots n_r$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L_s}{\partial \dot{q}_{r_j}}\right) - \frac{\partial L_s}{\partial q_{r_j}} = \sum_{i=1}^{n_r} \lambda_i \cdot \frac{\partial \Gamma_i(\boldsymbol{x}, \boldsymbol{q}_{\boldsymbol{r}})}{\partial q_{r_j}} + \tau_{r_j}, \tag{22}$$

where τ_{r_j} represent the virtual forces/torques of the subsystem formed by the generalized articular coordinates $\boldsymbol{q_r}$.

- Dynamic equations associated to the platform $k = 1, \ldots n$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L_p}{\partial \dot{x}_k}\right) - \frac{\partial L_p}{\partial x_k} = \sum_{i=1}^{n_r} \lambda_i \cdot \frac{\partial \Gamma_i(\boldsymbol{x}, \boldsymbol{q_r})}{\partial x_k} + Q_{x_k}, \quad (23)$$

where Q_{x_k} are the external known forces/torques applied to the mobile platform. This vector is defined in the fixed frame

Using the kinematic relations previously calculated and considering that the Lagrange multipliers, $\boldsymbol{\lambda}$, are associated to the internal constraints forces that join the platform and the serial chain subsystem, the equation system can be rewritten as[5]

$$\begin{aligned} D_r(\boldsymbol{q_r})\ddot{\boldsymbol{q}}_r + C_r(\boldsymbol{q_r}, \dot{\boldsymbol{q}}_r)\dot{\boldsymbol{q}} + \boldsymbol{G_r}(\boldsymbol{q_r}) &= \underbrace{\left[\frac{\partial \Gamma(\boldsymbol{x}, \boldsymbol{q_r})}{\partial \boldsymbol{q_r}}\right]^T}_{n_r \times n_r} \boldsymbol{\lambda} + \boldsymbol{\tau_r}, \\ D_x(\boldsymbol{x})\ddot{\boldsymbol{x}} + C_x(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \boldsymbol{G_x}(\boldsymbol{x}) &= \underbrace{\left[\frac{\partial \Gamma(\boldsymbol{x}, \boldsymbol{q_r})}{\partial \boldsymbol{x}}\right]^T}_{n \times n_r} \boldsymbol{\lambda} + \boldsymbol{Q_x}. \end{aligned} \quad (24)$$

Additionally it can be demonstrated that (see Appendix),

$$-\left(\frac{\partial \boldsymbol{\Gamma}(\boldsymbol{x}, \boldsymbol{q_r})}{\partial \boldsymbol{q}}\right)^T \left[\left(\frac{\partial \boldsymbol{\Gamma}(\boldsymbol{x}, \boldsymbol{q_r})}{\partial \boldsymbol{x}}\right)^T\right]^{\dagger} = \boldsymbol{J}_C^T, \quad (25)$$

where D_r, D_x represent the Inertia matrices, C_r the Coriolis terms, $\boldsymbol{G_r}$, $\boldsymbol{G_x}$ the Gravity terms of the dynamic model of each subsystem and J_C the Jacobian matrix of the constraints. $\boldsymbol{\tau_r}$ is the vector of the virtual forces τ_{r_j} associated to the generalized articular coordinates $\boldsymbol{q_r}$. $\boldsymbol{Q_x}$ is the vector of the external forces and torques applied to the mobile platform Q_{x_k}. Note that J_C is needed to project this vector to the generalized coordinate space in the dynamic equation corresponding to the serial chain subsystem.

3.7. Dynamic model

The dynamic model can be obtained in terms of the active joints and the sensorized passive joints. For that purpose, first the internal forces are eliminated from the equation system (24); then, the expressions (15) and (20) are introduced to replace $\boldsymbol{q_r}$ with $\boldsymbol{q}$:

$$\begin{aligned} \boldsymbol{\tau} = {} & D(\boldsymbol{x}, \boldsymbol{q_r})\ddot{\boldsymbol{q}}_r + C(\boldsymbol{x}, \boldsymbol{q_r}, \dot{\boldsymbol{x}}, \dot{\boldsymbol{q}}_r)\dot{\boldsymbol{q}}_r \\ & + \boldsymbol{G}(\boldsymbol{x}, \boldsymbol{q_r}) + \boldsymbol{F}_{\text{ext}} \end{aligned} \quad (26)$$

with,

$$\begin{aligned} D &= T^T \left[D_r + J_C^T D_x J_C\right] \cdot T_q \\ C &= T^T \left[\left(C_r + J_C^T \left(D_x \dot{J}_C + C_x J_C\right)\right) \cdot T_q \right. \\ &\quad \left. + \left(D_r + J_C^T D_x J_C\right) \cdot \dot{T}_q\right] \\ \boldsymbol{G} &= T^T \left[\boldsymbol{G_r} + J_C^T \boldsymbol{G_x}\right] \\ \boldsymbol{F}_{\text{ext}} &= -T^T J_C^T \cdot \boldsymbol{Q_x} \end{aligned}$$

where $D, C \in \mathbb{R}^{n \times n_r}$ and $\boldsymbol{\tau}, \boldsymbol{G}, \boldsymbol{F}_{\text{ext}} \in \mathbb{R}^n$.

Equation (26) determines the dynamic model of the robot considering active joints and passive sensorized joints as the control variables. This model is only valid if the robot is not in a singular position. Besides, this model cannot be inverted in order to define the direct dynamic model.

For a better understanding, the process of obtaining the inverse dynamic model is summarized in Algorithm 1.

Algorithm 1 Methodology for the calculation of the inverse dynamic model

1: Calculation of the closure-loop equations relating $\boldsymbol{q_a}$ and $\boldsymbol{x}$ (2); $\boldsymbol{q_r}$ and $\boldsymbol{x}$ (3); $\boldsymbol{q_p}$ and $\boldsymbol{q}$ (4).
2: Resolution of the direct and inverse position problems as stated in 4.2.
3: Resolution of the velocity problem: J_R (5), J_C (7), T (13) and T_q (15).
4: Resolution of the acceleration problem: $\dot{J}_C$ (18), $\dot{T}$ (19) and $\dot{T}_q$ (20).
5: Calculation of the Lagrangian equation of the mechanism (21).
6: Resolution of the differential equations related to the serial chain subsystem (22) and the platform subsystem (23).
7: Calculation of $\boldsymbol{\tau}$ considering both active $\boldsymbol{q_a}$ and extra sensorized passive joint data $\boldsymbol{q_s}$ (26).

It is important to note that the formulation proposed in this section focuses in planar parallel robots whose end-effector is attached to a mobile platform. However, there are some mechanisms, as the 5R planar parallel manipulator, that do not have a mobile platform. In these cases, the mobile platform is considered to be a point without mass nor inertia. The proposed methodology is then also valid for these kind of mechanisms.[36]

Obtaining the dynamic model in terms of only the active joints $\boldsymbol{q_a}$ is a particular case of this formulation. In this case, $\mathbf{q_s}$ is an empty vector, $\mathbf{q} = \boldsymbol{q_a}$ and $T_q = T$. Using these considerations and following the same procedure as the one described in this section, the dynamic model in terms of only active joints can be obtained.

4. Case of Study: 3RRR Planar Parallel Robot

The 3RRR is a 3-dof planar parallel manipulator. It is composed by a moving platform and three RRR serial subchains that join it to the base (Fig. 2).

Let $P(x, y)$ be the end-effector position in the plane and θ_z its orientation. Let O be the origin of the fixed reference frame and A_i, B_i, C_i, $i = 1, 2, 3$ define the rotational

Table I. Parameters of 3RRR parallel robot.

Body	Length	Mass	Inertia	Center
$\overline{A_iB_i}$	L_i	m_{L_i}	$[I_{zz}]_{L_i}$	$l_{c_{L_i}}$
$\overline{B_iC_i}$	l_i	m_{l_i}	$[I_{zz}]_{l_i}$	$l_{c_{l_i}}$
$C_1C_2C_3$	$\frac{h_{i(i+1)} = \overline{C_iC_{i+1}}}{d_i = \overline{C_iP}}$	m_p	$[I_{zz}]_p$	$P(x, y)$
Load	–	m_c	$[I_{zz}]_c$	$P(x, y)$
Sensor	–	m_{s_i}	$[I_{zz}]_s$	$B_i(x, y)$

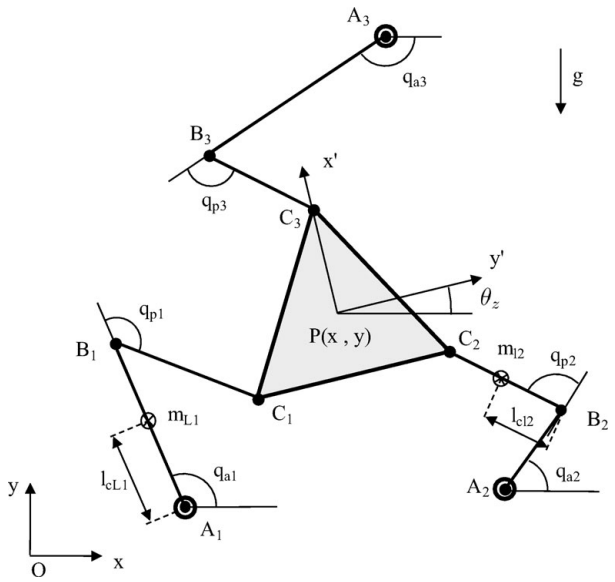

Fig. 2. 3RRR parallel manipulator.

articulations of each leg. The parameters of the mechanism are defined in Table I.

Let $\boldsymbol{q_a} = [q_{a_1} \;\; q_{a_2} \;\; q_{a_3}]^T$ be the actuated joint vector, $\boldsymbol{q_p} = [q_{p_1} \;\; q_{p_2} \;\; q_{p_3}]^T$ the passive joint vector and $\boldsymbol{x} = [x \;\; y \;\; \theta_z]^T$ the cartesian coordinate vector.

In this particular case, the choice of the sensorized joints is straightforward. The passive rotational joints q_{p_i}, with $i = 1, 2, 3$, will be sensorized. This way, the direct kinematic problem of the robot will have an unique solution, as each of the 2-dof serial subchains will be fully sensorized. This configuration will provide symmetry to the dynamic model of the mechanism, simplifying its parametric formulation. Besides, the practical implementation of these sensors can be easily done using rotational encoders.

The masses will be considered to be concentrated in the gravity center of each body and friction is neglected.

If q_{p_i} are considered sensorized, then $\boldsymbol{q_s} = [q_{p_1} \;\; q_{p_2} \;\; q_{p_3}]^T$ and $\boldsymbol{q_r} = \boldsymbol{q} = [\boldsymbol{q_a} \;\; \boldsymbol{q_p}]^T$. The sensors will be located in B_i points.

To obtain the dynamic model, the guidelines proposed in Section 2 are followed.

4.1. Closure loop equations of the mechanism

Based on Fig. 2, the three vectorial closure loop equations can be expressed as follows.

$$\overrightarrow{OP} = \overrightarrow{OA_i} + \overrightarrow{A_iB_i} + \overrightarrow{B_iC_i} + \overrightarrow{C_iP}, \qquad i = 1, 2, 3 \tag{27}$$

Using these equations the closure loop equations relating $\boldsymbol{q_a}$ and $\boldsymbol{x}$ can be defined:

$$\begin{aligned} \Phi_i(\boldsymbol{x}, \boldsymbol{q_a}) &= 0 \\ 0 = -l_i^2 &+ \big(x - OA_{xi} - L_i \cos q_{a_i} \\ &- d_i \cos(\theta_z + \phi_i)\big)^2 + \big(y - OA_{yi} \\ &- L_i \sin q_{a_i} - d_i \sin(\theta_z + \phi_i)\big)^2, \\ i &= 1, 2, 3 \end{aligned} \tag{28}$$

where ϕ_i is a constant value, whose value depends of the geometry of the platform. In this case $\phi_i = \frac{(4(i-1)+1)\pi}{6}$.

Similarly, the closure loop equations relating $\boldsymbol{q_r} = \boldsymbol{q}$ and $\boldsymbol{x}$ can be obtained from the vector closure loop Eq. (27). In order to obtain the closure loop that relates $\boldsymbol{q_s} = \boldsymbol{q_p}$ and $\boldsymbol{q_a}$ another vectorial closure loop equations must be defined:

$$\begin{aligned} 0 &= \overrightarrow{OA_1} + \overrightarrow{A_1B_1} + \overrightarrow{B_1C_1} + \overrightarrow{C_1C_2} - \overrightarrow{B_2C_2} - \overrightarrow{A_2B_2} - \overrightarrow{OA_2} \\ 0 &= \overrightarrow{OA_2} + \overrightarrow{A_2B_2} + \overrightarrow{B_2C_2} + \overrightarrow{C_2C_3} - \overrightarrow{B_3C_3} - \overrightarrow{A_3B_3} - \overrightarrow{OA_3} \\ 0 &= \overrightarrow{OA_3} + \overrightarrow{A_3B_3} + \overrightarrow{B_3C_3} + \overrightarrow{C_3C_1} - \overrightarrow{B_2C_2} - \overrightarrow{A_1B_1} - \overrightarrow{OA_1} \end{aligned} \tag{29}$$

From Eq. (29) and imposing the constraint $|\overrightarrow{C_iC_{i+1}}|^2 = h_{i(i+1)}^2$, the three independent closure loop equations that relate $\boldsymbol{q_s} = \boldsymbol{q_p}$ and $\boldsymbol{q_a}$ can be obtained:

$$\begin{aligned} f_i(\boldsymbol{q_a}, \boldsymbol{q_s}) &= 0 \\ 0 = -h_{i(i+1)}^2 &+ \big[l_i \cos(q_{a_i} + q_{p_i}) \\ &+ L_i \cos q_{a_i} + OA_{xi} - OA_{x(i+1)} \\ &- L_{i+1} \cos q_{a_{i+1}} - l_i \cos(q_{a_{i+1}} + q_{p_{i+1}})\big]^2 \\ &+ \big[l_i \sin(q_{a_i} + q_{p_i}) + L_i \sin q_{a_i} \\ &+ OA_{yi} - OA_{y(i+1)} - L_{i+1} \sin q_{a_{i+1}} \\ &- l_i \sin(q_{a_{i+1}} + q_{p_{i+1}})\big]^2, \; i = 1, 2, 3. \end{aligned} \tag{30}$$

4.2. Position problem

As the serial subchains are fully sensorized, the redundant direct position problem of the robot, which calculates $\boldsymbol{x} = [x \;\; y \;\; \theta_z]^T$ using $\boldsymbol{q_a}$ and $\boldsymbol{q_p}$ can be calculated using Eq. (27). First, the vertexes of the triangular platform are calculated:

$$\begin{aligned} \overrightarrow{OC_i} &= \overrightarrow{OA_i} + \overrightarrow{A_iB_i} + \overrightarrow{B_iC_i}, \; i = 1, 2, 3 \\ &\begin{cases} OC_{ix} = OA_{xi} + L_i \cos q_{a_i} + l_i \cos(q_{a_i} + q_{p_i}) \\ OC_{iy} = OA_{yi} + L_i \sin q_{a_i} + l_i \sin(q_{a_i} + q_{p_i}) \end{cases}. \end{aligned} \tag{31}$$

In order to increase the robustness against parameter uncertainty, the end-effector position is defined using all the data available. The center of mass of the triangle, where the tool center point $\overrightarrow{OP} = [x, y]^T$ is located, is calculated as follows:

$$\begin{aligned} x &= \tfrac{1}{3}(OC_{1x} + OC_{2x} + OC_{3x}) \\ y &= \tfrac{1}{3}(OC_{1y} + OC_{2y} + OC_{3y}) \end{aligned}. \tag{32}$$

Similarly, in order to calculate the orientation of the platform, the data from all sensors is used. For that purpose, first the vectors that join the different vertexes are defined:

$$\overrightarrow{C_i P} = \overrightarrow{OP} - \overrightarrow{OC_i}, \quad i = 1, 2, 3. \tag{33}$$

Then, using these vectors, the orientation angle can be obtained:

$$\theta_z = \frac{1}{3}\left[\arctan\left(\frac{C_1P_y}{C_1P_x}\right) + \arctan\left(\frac{C_2P_y}{C_2P_x}\right) + \arctan\left(\frac{C_2P_y}{C_2P_x}\right) - \frac{5\pi}{2}\right]. \tag{34}$$

The inverse position problem can be calculated using Eq. (28):

$$q_{a_i} = 2 \cdot \arctan\left(\frac{-b_i \pm \sqrt{b_i^2 - 4a_i c_i}}{2a_i}\right), \quad i = 1, 2, 3, \tag{35}$$

where

$$\begin{cases} a_i = -K_{xi} - K_i \\ b_i = 2K_{yi} \\ c_i = K_{xi} - K_i \end{cases},$$

$$\text{and} \begin{cases} K_{xi} = x - OA_{xi} - d_i \cos(\theta_z + \phi_i) \\ K_{yi} = x - OA_{yi} - d_i \sin(\theta_z + \phi_i) \\ K_i = \frac{l_i^2 - K_{xi}^2 - K_{yi}^2 - L_i^2}{-2L_i} \end{cases}.$$

After calculating the active joints value, passive joints can be derived:

$$q_{p_i} = 2 \cdot \arctan\left(\frac{-b_{pi} \pm \sqrt{b_{pi}^2 - 4a_{pi}c_{pi}}}{2a_{pi}}\right) - q_{a_i}, \quad i = 1, 2, 3, \tag{36}$$

where

$$\begin{cases} a_{pi} = -K_{pxi} - K_{pi} \\ b_{pi} = 2K_{pyi} \\ c_{pi} = K_{pxi} - K_{pi} \end{cases},$$

$$\text{and} \begin{cases} K_{pxi} = x - L_i \cos q_{a_i} - d_i \cos(\theta_z + \phi_i) \\ K_{pyi} = x - L_i \sin q_{a_i} - d_i \sin(\theta_z + \phi_i) \\ K_{pi} = \frac{OA_{xi}^2 + OA_{yi}^2 - K_{pxi}^2 - K_{pyi}^2 - l_i^2}{-2l_i} \end{cases}.$$

Note that the sign in Eqs. (35) and (36) determine the working modes[21] of the mechanism.

4.3. Velocity problem

In order to determine the velocity equations and the Jacobian Matrices, the previously calculated closure loop Eqs. (3), (28), and (30) are differentiated.

Jacobian matrices relating $\dot{\boldsymbol{x}}$ and $\dot{\boldsymbol{q}}_{\boldsymbol{a}}$:

$$\frac{\partial \phi(\boldsymbol{x}, \boldsymbol{q_a})}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} + \frac{\partial \phi(\boldsymbol{x}, \boldsymbol{q_a})}{\partial \boldsymbol{q_a}}\dot{\boldsymbol{q}}_{\boldsymbol{a}} = \underbrace{J_{xa}}_{3\times3} \cdot \dot{\boldsymbol{x}} + \underbrace{J_{q_a}}_{3\times3} \cdot \dot{\boldsymbol{q}}_{\boldsymbol{a}} = 0. \tag{37}$$

In order to calculate the constraint matrix, the Jacobian matrices relating $\dot{\boldsymbol{x}}$ and $\dot{\boldsymbol{q}}$ have to be calculated. For that purpose the alternative method proposed in Section 3.3.2 is used:

$$\frac{\partial \Gamma(\boldsymbol{x}, \boldsymbol{q})}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} + \frac{\partial \Gamma(\boldsymbol{x}, \boldsymbol{q})}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} = \underbrace{J_x}_{6\times3} \cdot \dot{\boldsymbol{x}} + \underbrace{J_q}_{6\times6} \cdot \dot{\boldsymbol{q}} = 0. \tag{38}$$

Using Eq. (38), and with the use of the Moore–Penrose generalized inverse (11) the constraint matrix can be defined as:

$$J_C = -J_x^{\dagger} \cdot J_q. \tag{39}$$

The transformation matrix T is required to calculate the model. For that purpose, first the Jacobian matrices relating $\dot{\boldsymbol{q}}_{\boldsymbol{a}}$ and $\dot{\boldsymbol{q}}_{\boldsymbol{s}}$ have to be found:

$$\frac{\partial f(\boldsymbol{q_a}, \boldsymbol{q_s})}{\partial \boldsymbol{q_a}}\dot{\boldsymbol{q}}_{\boldsymbol{a}} + \frac{\partial f(\boldsymbol{q_a}, \boldsymbol{q_s})}{\partial \boldsymbol{q_s}}\dot{\boldsymbol{q}}_{\boldsymbol{s}} = \underbrace{J_{q_{as}}}_{3\times3} \cdot \dot{\boldsymbol{q}}_{\boldsymbol{a}} + \underbrace{J_{q_s}}_{3\times3} \cdot \dot{\boldsymbol{q}}_{\boldsymbol{s}} = 0. \tag{40}$$

Using Eq. (40) the transformation matrix can be defined:

$$\dot{\boldsymbol{q}} = \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{q_a}}\dot{\boldsymbol{q}}_{\boldsymbol{a}} = \frac{\partial \begin{bmatrix} \boldsymbol{q_a} \\ \boldsymbol{q_s} \end{bmatrix}}{\partial \boldsymbol{q_a}}\dot{\boldsymbol{q}}_{\boldsymbol{a}} = [I_{3\times3} - J_{q_s}^{-1} J_{q_{as}}]^T \dot{\boldsymbol{q}}_{\boldsymbol{a}} = T \cdot \dot{\boldsymbol{q}}_{\boldsymbol{a}}. \tag{41}$$

4.4. Acceleration problem

To obtain the acceleration expresions, the velocity equations (37), (38), (40), and (41) must be differentiated:

$$J_{xa}\ddot{\boldsymbol{x}} + J_{q_a}\ddot{\boldsymbol{q}}_{\boldsymbol{a}} + \dot{J}_{q_a}\dot{\boldsymbol{q}}_{\boldsymbol{a}} + \dot{J}_{xa}\dot{\boldsymbol{x}} = 0, \tag{42}$$

$$J_x\ddot{\boldsymbol{x}} + J_q\ddot{\boldsymbol{q}} + \dot{J}_q\dot{\boldsymbol{q}} + \dot{J}_x\dot{\boldsymbol{x}} = 0, \tag{43}$$

$$J_{q_{as}}\ddot{\boldsymbol{q}}_{\boldsymbol{a}} + J_{q_s}\ddot{\boldsymbol{q}}_{\boldsymbol{s}} + \dot{J}_{q_{as}}\dot{\boldsymbol{q}}_{\boldsymbol{a}} + \dot{J}_{q_s}\dot{\boldsymbol{q}}_{\boldsymbol{s}} = 0, \tag{44}$$

$$\ddot{\boldsymbol{q}} = T\ddot{\boldsymbol{q}}_{\boldsymbol{a}} + \dot{T}\dot{\boldsymbol{q}}_{\boldsymbol{a}}. \tag{45}$$

4.5. Lagrangian equation

As stated in Section 2, two subsystems are considered in order to formulate the Lagrangian equation. This way, the kinetic K_p and potential energy U_p of the platform is defined as:

$$K_p = \tfrac{1}{2}\dot{\boldsymbol{x}}^T D_x \dot{\boldsymbol{x}}, \tag{46}$$

where

$$D_x = \begin{bmatrix} m_p + m_c & 0 & 0 \\ 0 & m_p + m_c & 0 \\ 0 & 0 & [I_{zz}]_p^c + [I_{zz}]_c^c \end{bmatrix},$$

$$U_p = \begin{bmatrix} 0 \\ m_p \cdot g \\ 0 \end{bmatrix} \cdot y = G_x \cdot y,$$

and g is the gravity.

Thus, the platform Lagrangian is

$$L_p = K_p - U_p. \tag{47}$$

For the serial subchain subsystem, the dynamic model of each leg correspond to a 2-dof. serial planar manipulator, whose Lagrangian equation is

$$\begin{aligned} L_s = {} & \tfrac{1}{2}\sum_{i=1}^{3}\Big[m_{L_i} l_{c_{L_i}}^2 + m_{s_i} L_i^2 \\ & + m_{l_i}\left(L_i^2 + l_{c_{l_i}}^2 + 2L_i l_{c_{l_i}} \cos q_{p_i}\right) \\ & + [I_{zz}]_{L_i}^c + [I_{zz}]_{l_i}^c + [I_{zz}]_{s_i}^c\Big] \cdot \dot{q}_{a_i}^2 \\ & + \sum_{i=1}^{3} m_{l_i}\left(l_{c_{l_i}}^2 + L_i l_{c_{l_i}} \cos q_{p_i}\right)\dot{q}_{a_i}\dot{q}_{p_i} \\ & + \tfrac{1}{2}\sum_{i=1}^{3}\left[m_{l_i} l_{c_{l_i}}^2 + [I_{zz}]_{l_i}^c\right]\dot{q}_{p_i}^2 \\ & - \sum_{i=1}^{3}\left[\left(m_{L_i} l_{c_{L_i}} + m_{s_i} L_i + m_{l_i} L_i\right) g \sin q_{a_i} + \right. \\ & \left. - m_{l_i} l_{c_{l_i}} g \sin\left(q_{a_i} + q_{p_i}\right)\right]. \end{aligned} \tag{48}$$

Combining Eqs. (46) and (48) the Lagrangian equation of the whole system is obtained:

$$L = L_p + L_s. \tag{49}$$

4.6. Diferential equations of the mobile platform and serial chain subsystems

Using Eq. (49), the Lagrange multiplier formulation is applied to obtain the differential equations associated to the platform and the serial chain subsystem.

4.6.1. Differential equations associated to the platform (k = 1, . . . 3).

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L_p}{\partial \dot{x}_k}\right) - \frac{\partial L_p}{\partial x_k} = \sum_{i=1}^{6}\lambda_i \cdot \frac{\partial \Gamma_i(x,q)}{\partial x_k} + Q_{xk}, \tag{50}$$

where $[Q_{xk}] = \boldsymbol{Q_x}$ are the external forces and torques applied to the platform. The partial derivatives that define Eq. (50) can be obtained as follows:

$$\begin{aligned} \tfrac{\partial L_p}{\partial \dot{x}} &= D_x \dot{x} \\ \tfrac{d}{dt}\tfrac{\partial L_p}{\partial \dot{x}} &= D_x \ddot{x}\,. \\ \tfrac{\partial L_p}{\partial x} &= -G_x \end{aligned} \tag{51}$$

Making use of Eqs. (50) and (51) the differential equation is defined. Identifying the inertia, Coriolis, and gravity matrices, the following expression is obtained:

$$D_x \ddot{\boldsymbol{x}} + G_x - Q_x = J_x^T \boldsymbol{\lambda}. \tag{52}$$

4.6.2. Differential equations associated to the serial subchain subsystem (j = 1, . . . 6).

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L_s}{\partial \dot{q}_{r_j}}\right) - \frac{\partial L_s}{\partial q_{r_j}} = \sum_{i=1}^{6}\lambda_i \cdot \frac{\partial \Gamma_i(x,q)}{\partial q_{r_j}} + \tau_{r_j} \tag{53}$$

Following the same procedure used for the platform equations, the inertia, Coriolis, and gravity matrices can be identified:

$$D_q \ddot{\boldsymbol{q}} + C_q \dot{\boldsymbol{q}} + G_q = \tau_r + J_q^T \boldsymbol{\lambda}, \tag{54}$$

$$D_q = \begin{bmatrix} d_{a_{11}} & 0 & 0 & d_{14} & 0 & 0 \\ 0 & d_{a_{22}} & 0 & 0 & d_{25} & 0 \\ 0 & 0 & d_{a_{33}} & 0 & 0 & d_{36} \\ d_{31} & 0 & 0 & d_{p_{11}} & 0 & 0 \\ 0 & d_{52} & 0 & 0 & d_{p_{22}} & 0 \\ 0 & 0 & d_{63} & 0 & 0 & d_{p_{33}} \end{bmatrix}, \tag{55}$$

where:

- $d_{a_{ii}} = m_{L_i} \cdot l_{c_{L_i}}^2 + m_{s_i} L_i^2 + m_{l_i}\left(L_i^2 + l_{c_{l_i}}^2 + 2L_i l_{c_{l_i}} \cos q_{p_i}\right) + [I_{zz}]_{L_i}^c + [I_{zz}]_{l_i}^c + [I_{zz}]_{s_i}^c$, $i = 1, 2, 3$
- $d_{p_{ii}} = m_{l_i} l_{c_{l_i}}^2 + [I_{zz}]_{l_i}^c$, $i = 1, 2, 3$
- $d_{ij} = d_{ji} = m_{l_i}\left(l_{c_{l_i}}^2 + L_i l_{c_{l_i}} \cos q_{p_i}\right) + [I_{zz}]_{l_i}^c$, $(ij) = (14), (25), (36)$

$$C_q = \begin{bmatrix} c_{11} & 0 & 0 & c_{14} & 0 & 0 \\ 0 & c_{22} & 0 & 0 & c_{25} & 0 \\ 0 & 0 & c_{33} & 0 & 0 & c_{36} \\ c_{41} & 0 & 0 & 0 & 0 & 0 \\ 0 & c_{52} & 0 & 0 & 0 & 0 \\ 0 & 0 & c_{63} & 0 & 0 & 0 \end{bmatrix}, \tag{56}$$

where:

- $h_i = -m_{l_i} L_i l_{c_{l_i}} \sin q_{p_i}$, $i = 1, 2, 3$
- $c_{ii} = h_i \dot{q}_{p_i}$, $i = 1, 2, 3$
- $c_{ij} = h_i(\dot{q}_{a_i} + \dot{q}_{p_i})$, $ij = (14), (25), (36)$
- $c_{ij} = h_i \dot{q}_{a_i}$, $ij = (41), (52), (63)$

$$\boldsymbol{G_q} = [g_{a_1} \;\; g_{a_2} \;\; g_{a_3} \;\; g_{p_1} \;\; g_{p_2} \;\; g_{p_3}]^T, \tag{57}$$

where:

- $g_{a_i} = (m_{L_i} l_{c_{L_i}} + m_{l_i} L_i + m_{s_i} L_i) g \cos q_{a_i}$ $+ m_{l_i} l_{c_{l_i}} g \cos(q_{a_i} + q_{p_i})$, $i = 1, 2, 3$
- $g_{p_i} = m_{l_i} l_{c_{l_i}} g(\cos q_{a_i} + q_{p_i})$, $i = 1, 2, 3$.

4.7. Dynamic model

As it has been described in Section 3, in order to obtain the dynamic model in terms of the control coordinates $\boldsymbol{q} = [\boldsymbol{q_a}\ \boldsymbol{q_s}]^T$ a series of transformations have to be applied to the

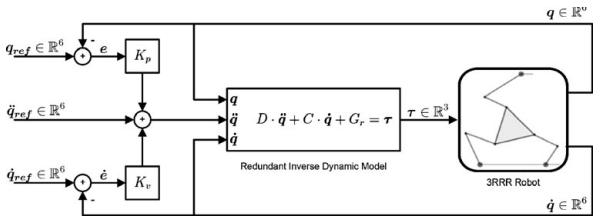

Fig. 3. Extended CTC approach.

differential equations (52) and (54). Using the robot Jacobian matrix, defined in Eq. (39) and the relations (41) and (43) the full dynamic model can be obtained considering both active and sensorized joints:

$$D\ddot{\boldsymbol{q}} + C\dot{\boldsymbol{q}} + \boldsymbol{G} - \boldsymbol{F_{ext}} = \boldsymbol{\tau_{q_a}},$$

$$\text{with } \begin{cases} D = T^T(D_q + J_C^T D_x J_C) \\ C = T^T(C_q + J_C^T D_x \dot{J}_C) \\ \boldsymbol{G} = T^T(\boldsymbol{G_q} + J_C^T \boldsymbol{G_x}) \\ \boldsymbol{F_{ext}} = T^T J_C^T \boldsymbol{Q_x} \end{cases}. \tag{58}$$

Following this methodology, passive joint sensor data is directly and explicitly considered in the model. This model can be calculated in a systematic, matrix-based way. Although its computational cost is higher due to this structure, this form is very useful for control engineers to implement novel control laws and demonstrate their stability. This way, as the authors demonstrated in ref. [36], model-based control techniques using this redundant dynamic model will be more robust against parameter uncertainty due to the redundant data, and their performance will increase significantly. This leads to better positioning accuracy and less trajectory tracking error.

4.8. Simulation results

The developed redundant dynamic model can be used to implement advanced control approaches. In [36] a novel computed torque control scheme considering a redundant dynamic model was proposed. This scheme, called Extended CTC approach (Fig. 3), is more robust than classical CTC approach when parameter uncertainties are high. The control law is shown in Eq. (59):

$$\boldsymbol{\tau} = D \cdot (K_p \cdot \boldsymbol{e} + K_v \cdot \dot{\boldsymbol{e}}) + (D \cdot \ddot{\boldsymbol{q}}_{\boldsymbol{ref}} + C \cdot \dot{\boldsymbol{q}} + G_r). \tag{59}$$

In order to validate the proposed dynamic model, the Extended CTC approach is compared to the classical CTC approach. For that purpose, a set of simulations have been conducted using the 3RRR robot. The classical CTC scheme and the Extended CTC scheme are implemented using the previously defined dynamic model. A circular nonsingular trajectory with center $(x, y) = (0, 0)$, radius of 0.1 m and $\theta_z = 0^\circ$ has been defined on the task space. Using the inverse kinematic problem defined in Section 3.2, the trajectories for both active and passive joints have been derived, considering a fixed working mode[21]. The model has been implemented in Matlab/Simulink environment, with a computational cost of 5 ms running on a Pentium D 2.4 Ghz, which allows its implementation in real-time.

Table II. Performance indexes.

ISE	IAE	ITAE
$\int_\infty^0 e^2(t)\,\mathrm{d}t$	$\int_\infty^0 \lvert e(t)\rvert\,(t)\,\mathrm{d}t$	$\int_\infty^0 t \cdot \lvert e(t)\rvert\ \mathrm{d}t$

The comparative analysis of control schemes consists of studying the ISE, IAE, and ITAE performance indexes of the end-effector positioning error $e(t)$ (Table II), for the predefined trajectory.

Model parameters have been defined as follows. Base point coordinates: $OA_1 = [-0.15, -0.84]$ (m), $OA_2 = [0.69, -0.17]$ (m), $OA_3 = [-0.66, 0.21]$ (m). Link lengths: $L_i = 0.5$(m), $l_i = 0.4$(m), $i = 1, 2, 3$. Link center of mass position: $l_{c_{L_i}} = 0.25$ (m), $l_{c_{l_i}} = 0.2$ (m), $i = 1, 2, 3$. Link masses: $m_{L_i} = 0.4239$ (kg), $m_{l_i} = 0.3391$ (kg), $i = 1, 2, 3$. Link inertia moments: $I_{L_i} = 0.0088$ (kg $\cdot$ m^2), $I_{l_i} = 0.0045$ (kg $\cdot$ m^2), $i = 1, 2, 3$. Platform geometry constants: $d_i = 0.1732$ (m), $i = 1, 2, 3$. Platform masses: $m_p = 1.3576$ (kg), $m_c = 0.5$ (kg). Platform inertia moments: $I_p = 0.0085$ (kg $\cdot$ m^2), $I_c = 8.3333 \cdot 10^{-4}$ (kg $\cdot$ m^2) Passive sensor masses: $m_{s_i} = 0.0656$ (kg), $i = 1, 2, 3$. Sensor inertia moments: $I_{s_i} = 3.687 \cdot 10^{-6}$ (kg $\cdot$ m^2), $i = 1, 2, 3$.

The K_p and K_v gains on both extended and classical CTC scheme have been tuned to obtain a maximum overshoot of 10 % and a peak time of 0.1 s.

To show the effect of parameter uncertainty in the model identification, the real robot parameters have been randomly modified from 1 %–5 % of their nominal values. In order to obtain statistical data, each simulation iteration is repeated 10 times for each case. For that purpose in a first step the parameter modification templates are created, that is, each model parameter (length, inertia, mass) is randomly assigned a 0, -1 or 1 value, meaning no value change, increment in of its nominal value or decrement of its nominal value, respectively. Ten different parameter templates are generated randomly, one for each simulation iteration. When the simulations are launched, for each parameter variation percentage and simulation iteration the modified parameters are calculated using the templates. For a given variation percentage, the average behavior of the controller in each set of iterations is considered.

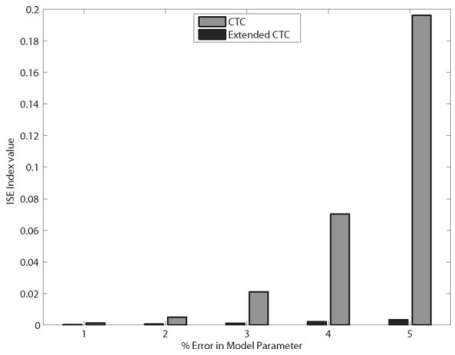

Fig. 4. Integral of the square error vs. % of parameter variation.

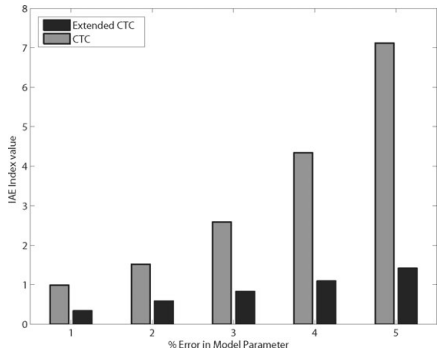

Fig. 5. Integral of the absolute error vs. % of parameter variation.

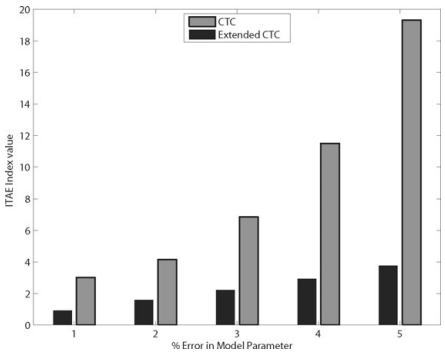

Fig. 6. Integral of the time absolute error vs. % of parameter variation.

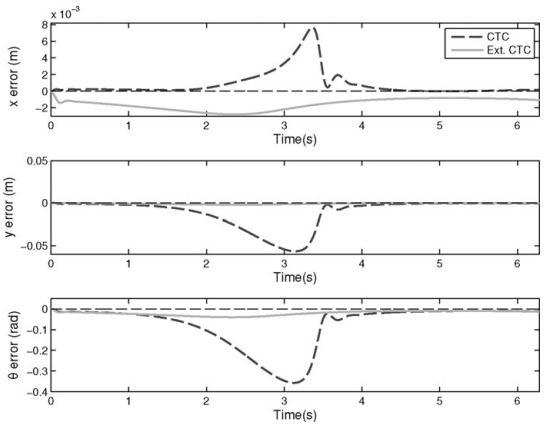

Fig. 7. Evolution of the TCP positioning error.

Statistical results are summarized in Figs. 4–6. As it could be expected, the performance indexes increase with the parameter variation. However, the extended CTC scheme that implements the proposed redundant dynamic model has a lower increase ratio than the traditional CTC scheme in all the performance indexes. This demonstrates that the proposed dynamic model is more robust against model parameter uncertainties.

In Figs. 7–9 the trajectories of the error signals, the TCP coordinates and the joint coordinates for a case in which parameters have been modified by 5 % of their nominal values are shown. In solid line the evolution of the extended CTC approach is shown, and in dashed line the classical CTC. The reference trajectory is plotted using triangles. As it can be seen, positioning errors are decreased with the

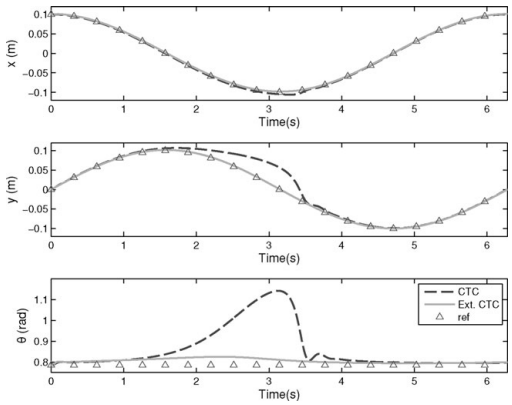

Fig. 8. Evolution of the TCP position.

use of extended CTC scheme, which validates the proposed redundant approach.

5. Conclusions

A useful formulation for the calculation of the dynamic model of planar parallel robot is proposed. For that purpose, first the closure loop equations of the mechanism that relate the active and passive joint coordinates and the task coordinates are calculated. Based on these equations, the direct and inverse position problems can be formulated. The velocity Jacobian matrices are obtained differentiating the closure loop equations. A second derivation leads to the solution of the acceleration problem, which completes the kinematic analysis of the mechanism. For the calculation of the dynamic model the Lagrangian Formulation is applied. First, the mechanism is divided in two subsystems: end-effector platform and serial chain subsystem. Then, the Lagrangian equation of each subsystem is obtained, and the differential equations associated to each subsystem derived. Finally, the two differential equation system are combined using the kinematic relations defined previously.

The proposed methodology is a systematic, matrix-based, powerful method that can be easily followed. The main advantage of the formulation is that the model can be defined in terms of only active joints, or in terms of active and some extra sensorized passive joints making it redundant. The consideration of extra sensors explicitly in the model can be used in model-based control techniques in order to obtain a better control performance, improving robustness against parameter uncertainty and reducing tracking and positioning errors as demonstrated in previous works.

An example based on the 3RRR parallel robot is developed in order to illustrate the proposed approach. This robot can be considered as the benchmark platform in planar parallel robots. The calculated redundant model is validated by the implementation of the model in an extended CTC control scheme. It is demonstrated that this approach, based on the redundant model is more robust and more accurate than the traditional CTC approach, in which the model is defined in terms of the active joints. The computational cost guarantee its implementation in Real-Time.

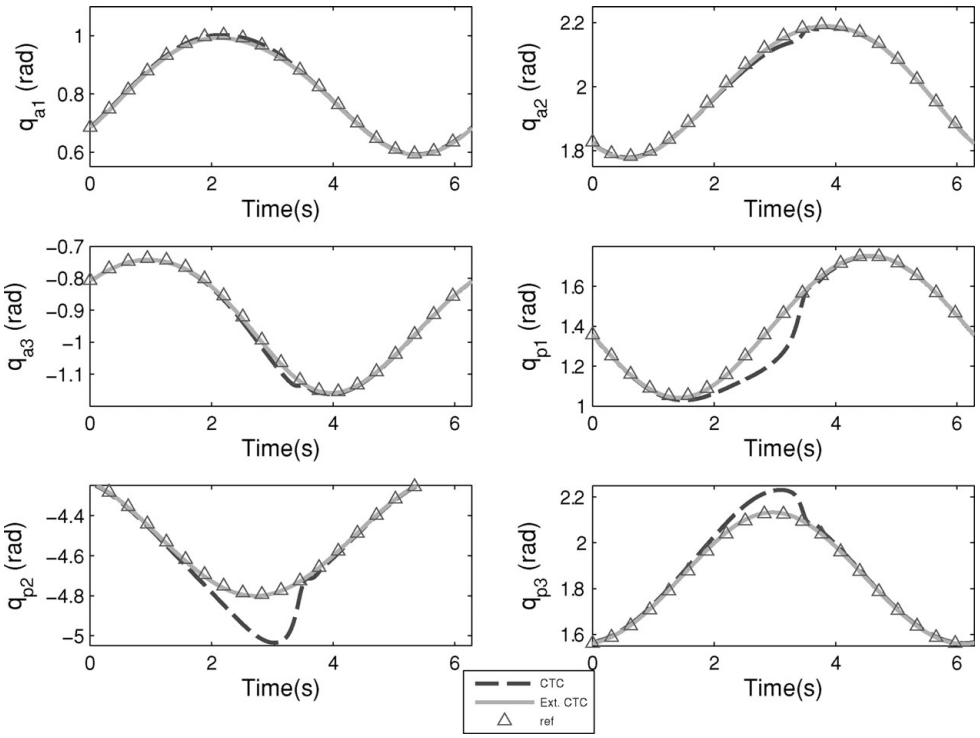

Fig. 9. Evolution of the joint coordinates position.

Acknowledgments

This work was financed in part by the GV/EJ BF106.2R129 research fellowship and GIC07/78, the MCYT&FEDER under project DPI-2006-4003 and DIPE 06-16 and by UPV/EHU under grant GIU07/36.

References

1. J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms* (Springer, Secaucus, NJ, 2007).
2. L. Baron and J. Angeles, "Kinematic decoupling of parallel manipulators using joint-sensor data," *IEEE Trans. Robot. Autom.* **16**(6), 644–651 (2000).
3. A. Barrientos, L. F. Peñín, C. Balaguer and R. Aracil, *Fundamentos de Robótica*, 2nd ed. (in spanish) (McGraw Hill, Madrid, Spain, 2007).
4. V. Bauma, M. Valášek and Z. Sika, "Increase of pkm Positioning Accuracy by Redundant Measurement," *Proceedings of the 5th Chemnitz Parallel Kinematics Seminar*, Chemnitz, Germany, (2006) pp. 547–563.
5. S. Bhattacharya, H. Hatwal and A. Ghosh, "An on-line parameter estimation scheme for generalized stewart platform type parallel manipulators," *Mech. Mach. Theory* **32**(1), 79–89 (1997).
6. S. Bhattacharya, N. Nenchev and M. Uchiyama, "A recursive formula for the inverse of the inertia matrix of a parallel manipulator," *Mech. Mach. Theory* **33**(7), 957–964 (1998).
7. J. Craig, *Introduction to Robotics: Mechanics and Control*. Number 0201543613 (Prentice Hall, Englewood Cliffs, NJ, 2005).
8. B. Dasgupta and P. Choudhurry, "General strategy based on the Newton–Euler approach for the dynamic formulation of parallel manipulators," *Mech. Mach. Theory*, **34**, 801–824 (1999).
9. B. Dasgupta and T. S. Mruthyunjaya, "A Newton–Euler formulation for the inverse dynamics of the Stewart platform," *Mech. Mach. Theory* **33**(8), 1135–1152 (1998).
10. W. Q. D. Do and D. C. H. Yang, "Inverse dynamic analysis and simulation of a platform type of robot," *J. Robot. Syst.* **5**(3), 209–227 (1988).
11. F. Ghorbel, "Modeling and pd Control of Closed-Chain Mechanical Systems," *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, LA (1995) pp. 540–542.
12. F. Ghorbel, O. Chételat, R. Gunawardana and R. Longchamp, "Modeling and set point control of closed-chain mechanisms: Theory and experiment," *IEEE Trans. Control Syst. Technol.* **8**(5), 801–815 (2000).
13. F. Ghorbel, O. Chételat and R. Longchamp, "A Reduced Model for Constrained Rigid Bodies with Application to Parallel Robots," *Proceedings of the IFAC Symposium on Robot Control*, Capri, Italy (1994) pp. 57–62.
14. R. Gunawardana and F. Ghorbel, "Pd Control of Closed-Chain Mechanical Systems: An Experimental Study," *Proceedings of the 5th IFAC Symposium on Robot Control (SYROCO'97)*, Nantes, France (1947) pp. 79–84.
15. H. B. Guo and H. R. Li, "Dynamic analysis and simulation of a six degree of freedom Stewart platform manipulator," *Proc. Inst. Mech. Eng. Part C: J. Mech. Eng. Sci.* **220**, 61–72 (2006).
16. M. Honegger, A. Codourey and E. Burdet, "Adaptive Control of the Hexaglide, a 6 dof Parallel Manipulator," *IEEE International Conference in Robotics and Automation*, Albuquerque, NM (1997) pp. 543–548.
17. W. Khalil and S. Guegan, "Inverse and direct dynamic modeling of Gough–Stewart robots," *IEEE Trans. Robot.* **20**(4), 754–762 (2004).
18. W. Khalil and O. Ibrahim, "General solution for the dynamic modeling of parallel robots," *J. Int. Robot Syst.* 19–37 (2007).
19. G. Lebret, K. Liu and F. L. Lewis, "Dynamic analysis and control of a stewart platform manipulators," *J. Robot. Syst.* **10**(5), 629–655 (1993).
20. K.-M. Lee and D. K. Shah, "Dynamic analysis of a three-degree-of-freedom in-parallel actuated manipulator," *IEEE J. Robot. Autom.* **4**(40), 361–367 (1988).

21. E. Macho, O. Altuzarra, C. Pinto and A. Hernández, "Workspaces associated to assembly modes of the 5r planar parallel manipulator," *Robotica* **26**, 395–403 (2008).
22. F. Marquet, O. Company, S. Krut and F. Pierrot, "Enhancing Parallel Robots Accuracy with Redundant Sensors," *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, DC (2002) pp. 4114–4119.
23. J. P. Merlet, "Closed Form Resolution of the Direct Kinematics of Parallel Manipulators Using Extra Sensors Data," *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, Atlanta, GA (1993) pp. 200–204.
24. J. P. Merlet, *Parallel Robots*, 2nd ed. (Kluwer, Dordrecht, 2006).
25. J. Murray and G. Lovell, "Dynamic modeling of closed-chain robotic manipulators and implications for trajectory control," *IEEE Trans. Robot. Autom.* **5**(4), 522–528 (1989).
26. Y. Nakamura and M. Godoussi, "Dynamics computation of closed-link robot mechanisms with nonredundant and redundant actuators," *IEEE Trans. Robot. Autom.* **5**(3), 294–302 (1989).
27. V. Petuya, A. Alonso, O. Altuzarra and A. Hernández, "Resolution of the Direct Position Problem of Parallel Kinematic Platforms Using the Geometrical-Iterative Method," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain (2005) pp. 3244–3249.
28. S. Riebe and H. Ulbrich, "Modelling and online computation of the dynamics of a parallel kinematic with six degrees-of-freedom," *Arch. Appl. Mech.* **72**, 813–829 (2003).
29. S. Staicu, D. Zhang and R. Rugescu, "Dynamic modelling of a 3 dof parallel manipulator using recursive matrix relations," *Robotica* **24**, 125–130 (2006).
30. L.-W. Tsai, *Robot Analysis*. Number ISBN: 0-471-32593-7 (Wiley-Interscience, New York, 1999).
31. L.-W. Tsai, "Solving the inverse dynamics of a Stewart–Gough manipulator by the principle of virtual work," *ASME J. Mech. Des.* **122**, 3–9 (2000).
32. J. Wang and C. Gosselin, "A new approach for the dynamic analysis of parallel manipulator," *Multibody Sys. Dyn.* **2**, 317–334 (1998).
33. K. E. Zanganeh, R. Sinatra and J. Angeles, "Kinematics and dynamics of a six-degree-of-freedom parallel manipulator with revolute legs," *Robotica* **15**, 385–394 (1997).
34. Y. Zhiyong, W. Jiang and M. Jiangping, "Motor-mechanism dynamic model based neural network optimized ctc control of a high speed parallel manipulator," *Mechatronics* **17**, 381–390 (2007). doi:10.1016/j.mechatronics.2007.04.009.
35. Z. Zhu, J. Li, Z. Gan and W. Zhang, "Kinematic and dynamic modelling for real-time control of tau parallel robot," *Mech. Mach. Theory* **40**, 1051–1067 (2005).
36. A. Zubizarreta, I. Cabanes, M. Marcos and Ch. Pinto, "Control of parallel robots using passive sensor data," *Proceedings of the IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems*, Nice, France (2008) pp. 2398–2403.

Appendix: Constraint matrix J_C

In this section further explanations regarding the relation between the constraint Jacobian (7) and the Jacobian of the constraint equations (3) are introduced:

The n_r constraints of the mechanism can be calculated formulating the vector equations that link the TCP with the origin of the cartesian frame:

$$\Gamma(\boldsymbol{x}, \boldsymbol{q}) = \mathbf{0} \in \mathbb{R}^{n_r \times 1}, \tag{A 1}$$

which defines a set with n_r equations. Differentiating it and using the proposed coordinate partitioning, the velocity equation can be rewritten as

$$\underbrace{\frac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{x}}}_{n_r \times n} \dot{\boldsymbol{x}} + \underbrace{\frac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{q}}}_{n_r \times n_r} \dot{\boldsymbol{q}} = 0. \tag{A 2}$$

So, clearing $\dot{\boldsymbol{x}}$,

$$\dot{\boldsymbol{x}} = -\underbrace{\left[\frac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{x}}\right]^{\dagger}}_{n \times n_r} \underbrace{\frac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{q}}}_{n_r \times n_r} \dot{\boldsymbol{q}}. \tag{A 3}$$

Thus, by direct comparison with Eq. (24), it is demonstrated that

$$-\left(\frac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{q}}\right)^{T} \left[\left(\frac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{x}}\right)^{T}\right]^{\dagger} = \boldsymbol{J}_C^{T}, \tag{A 4}$$

which demonstrates that the relation in Eq. (25) arises naturally because of the applied formulation, based on the coordinate partitioning.