

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

**4,800**

Open access books available

**122,000**

International authors and editors

**135M**

Downloads

Our authors are among the

**154**

Countries delivered to

**TOP 1%**

most cited scientists

**12.2%**

Contributors from top 500 universities



**WEB OF SCIENCE™**

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.

For more information visit [www.intechopen.com](http://www.intechopen.com)



## Kinematic and Inverse Dynamic Analysis of a C5 Joint Parallel Robot

Georges Fried<sup>1</sup>, Karim Djouani<sup>1</sup> and Amir Fijany<sup>2</sup>

<sup>1</sup>University of Paris Est Créteil LISSI-SCTIC Laboratory

<sup>2</sup>The Italian Institute of Technology

<sup>1</sup>France

<sup>2</sup>Italy

### 1. Introduction

Parallel manipulators have been proposed to overcome accuracy problem in the end effector positioning, exhibited by serial manipulators (Stewart, 1965)(Reboulet, 1988)(Merlet, 2000). These parallel robots are primarily used in the applications for which the considered processes require a high degree of accuracy, high speeds or accelerations. Aircraft simulator (Stewart, 1965), machining tools (Neugebauer et al., 1998)(Poignet et al., 2002), and various other medical applications (Merlet, 2002)(Leroy et al., 2003)(Plitea et al., 2008) constitute some of the many possible applications of parallel robots.

The computation of the inverse dynamic model is essential for an effective robot control. In the field of parallel robots, many approaches have been developed for efficient computation of the inverse dynamics. The formalism of d'Alembert has been used to obtain an analytical expression of the dynamics model (Fichter, 1986)(Nakamura & Ghodoussi, 1989). The principle of virtual works has been applied in (Tsai, 2000) for solving the inverse dynamics of the Gough-Stewart platform and in (Zhu et al., 2005) for a Tau parallel robot. Lagrangian formalism is applied in (Leroy et al., 2003) for the dynamics modeling of a parallel robot used as a haptic interface for a surgical simulator. These various approaches do not seem effective for a robot dynamic control under the real time constraint. A better computational efficiency can be achieved by the development of approaches using recursive schemes, in particular, based on the Newton-Euler formulation. Gosselin (Gosselin, 1996) proposed an approach for the computation of the inverse dynamic model of planar and spatial parallel robots, in which all the masses and inertias are taken into account. This proposed method is difficult to generalize for all the parallel architectures. Dasgupta et al (Dasgupta & Choudhury, 1999) applied this method to several parallel manipulators. Khan (Khan, 2005) has developed a recursive algorithm for the inverse dynamics. This method is applied to a 3R planar parallel robot. Bi et al (Bi & Lang, 2006) use the Newton-Euler recursive scheme for the computation of the articular forces of a tripod system. Khalil et al (Khalil & Guegan, 2004) proposed a general method for the inverse and direct dynamic model computation of parallel robots, which is applied to several parallel manipulators (Khalil & Ibrahim, 2007).

Despite the large amount of contributions in this field, there is still a need for improving the computational efficiency of the inverse kinematic and dynamic model calculation for real-time control. In this paper, a parallel robot is considered as a multi robot system with

$k$  serial robots (the  $k$  parallel robot segments) moving a common load (the mobile platform). The proposed approach uses the methodology developed by Khalil et al (Khalil & Guegan, 2004)(Khalil & Ibrahim, 2007).

In this paper, we first review the proposed approaches for computation of kinematics and inverse dynamics of a C5 joint parallel robot. One of the interesting aspects of the parallel robots is that, unlike the serial robots, it is easy and efficient to directly compute the inverse of Jacobian matrix: However, there seems to be no proposed approach for direct computation of the Jacobian matrix. We propose a new method which allows the derivation of the Jacobian matrix in a factored form, i.e., as a product of two highly sparse matrices. This factored form enables a very fast computation and application of the Jacobian matrix, which is also needed for the inverse dynamics computation. Another issue for inverse dynamics computation is the determination of joint accelerations given the acceleration of the mobile platform. We propose a new scheme that, by using projection matrices, enable a fast and direct calculation of the joint accelerations. Since this calculation is needed in any inverse dynamics formalism, our proposed new method improves the efficiency of the computation. For calculation of the inverse dynamics, we consider the formalism developed by Khalil et al (Khalil & Guegan, 2004)(Khalil & Ibrahim, 2007). In this approach, since the inverse of Jacobian is used, the calculation of the joint forces would require a linear system solution. We show that, by using our factorized form of the Jacobian, our proposed scheme not only eliminates the need for linear system solution but it also does not require the explicit computation of either Jacobian or its inverse. As a result, a significantly better efficiency in the computation can be achieved. This paper is organized as follows. In Section 2, we present some preliminaries and the notation used in our approaches. The C5 parallel robot is presented in Section 3. The proposed methodologies for computation of the inverse kinematics and the inverse Jacobian matrix are reviewed in Sections 4 and 5. The new methodology for derivation of the Jacobian matrix in a factored form is presented in Section 6. In Section 7, we present a fast and direct scheme for calculation of joint accelerations, given the desired acceleration of the mobile platform. The formalism for computation of the inverse dynamics, developed by Khalil et al (Khalil & Guegan, 2004)(Khalil & Ibrahim, 2007), is discussed in Section 8 and it is shown how the new scheme for calculation of joint accelerations as well as the use of factored form of the Jacobian matrix can significantly improve the computational efficiency. A simulation of the proposed scheme for computation of the inverse dynamics is provided in section 9 validating the proposed approach. Finally, some concluding remarks are presented in Section 10.

## 2. Preliminaries

In this section, the required notation for a serial chain are presented (see also Fig. 1).

### 2.1 System model and notations

#### 2.1.1 Joint and link parameters

- $O_j$ : Origin of frame  $j$  which is taken to be the center of  $j^{th}$  joint
- $P_j$ : position vector from  $O_j$  to  $O_{j+1}$
- $N$ : number of bodies
- $Q_j, \dot{Q}_j$ : position and velocity of the  $j^{th}$  joint

### 2.1.2 Spatial quantities

- $H_j$ : spatial-axis (map matrix) of joint  $j$ . For a joint with one rotational degree of freedom (DOF) around z-axis,  $H_j$  is given by:

$$H_j = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- $\mathbf{V}_j = \begin{bmatrix} \omega_j \\ v_j \end{bmatrix} \in \mathbb{R}^6$ : spatial velocity of point  $O_j$
- $I_j \in \mathbb{R}^{6 \times 6}$ : spatial inertia of body  $j$

The spatial inertia of body  $j$  about its center of mass,  $G_j$ , is denoted by  $I_{G_j}$  and is given by:

$$I_{G_j} = \begin{bmatrix} J_{G_j} & 0 \\ 0 & m_j U \end{bmatrix} \in \mathbb{R}^{6 \times 6}$$

where  $J_{G_j}$  is the second moment of mass of link  $j$  about its center of mass and  $m_j$  is the mass of link  $j$ . The spatial inertia  $I_j$  can be calculated by:

$$I_j = \hat{S}_j I_{G_j} \hat{S}_j^t$$

Where  $s_j$  represents the position vector from  $O_j$  to  $G_j$

- $\mathbf{V}_{N+1} \in \mathbb{R}^6$ : spatial velocity of the end effector

### 2.1.3 Global quantities

The following global quantities are defined for  $j = N$  to 1

- $\dot{\mathbf{Q}} = \text{Col}(\dot{Q}_j)$ : vector of joint velocity
- $\mathcal{V} = \text{Col}(V_j) \in \mathbb{R}^{6N}$ : global vector of spatial velocities
- $\mathcal{H} = \text{Diag}(H_j) \in \mathbb{R}^{6N \times 6N}$ : global matrix of spatial axis.
- $\mathcal{M}$ : Symmetric positive definite (SPD) mass matrix

### 2.2 General notation

With any vector  $\mathbf{V} = [V_x \ V_y \ V_z]^t$ , a tensor  $\tilde{V}$  can be associated whose representation in any frame is a skew symmetric matrix given by:

$$\tilde{V} = \begin{bmatrix} 0 & -V_z & V_y \\ V_z & 0 & -V_x \\ -V_y & V_x & 0 \end{bmatrix}$$

The tensor  $\tilde{V}$  has the property that  $\tilde{V} = -\tilde{V}^t$  and  $\tilde{V}_1 \mathbf{V}_2 = \mathbf{V}_1 \times \mathbf{V}_2$  i.e., it is the vector cross-product operator. A matrix  $\hat{V}$  associated to the vector  $\mathbf{V}$  is defined as:

$$\hat{\mathbf{v}} = \begin{bmatrix} U \tilde{\mathbf{v}} \\ 0 \ U \end{bmatrix}$$

where  $U$  and  $0$  stand for unit and zero matrices of appropriate size.

In our derivation, we also make use of global matrices and vectors which lead to a compact representation of various factorizations. A bidiagonal block matrix  $\mathcal{P} \in \mathbb{R}^{6N \times 6N}$  is defined as:

$$\mathcal{P} = \begin{bmatrix} U & & & & & \\ -\hat{\mathbf{P}}_{N-1} & U & & & & \\ 0 & -\hat{\mathbf{P}}_{N-2} & U & & & \\ 0 & & 0 & & & \\ \vdots & & \vdots & & & \\ 0 & & 0 & & 0 & -\hat{\mathbf{P}}_1 \ U \end{bmatrix}$$

The inverse of matrix  $\mathcal{P}$  is a block triangular matrix given by:

$$\mathcal{P}^{-1} = \begin{bmatrix} U & 0 & \cdots & 0 \\ \hat{\mathbf{P}}_{N,N-1} & U & 0 & \cdots & 0 \\ \hat{\mathbf{P}}_{N,N-2} & \hat{\mathbf{P}}_{N-1,N-2} & U & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{\mathbf{P}}_{N,1} & \hat{\mathbf{P}}_{N-1,1} & \cdots & \hat{\mathbf{P}}_{2,1} & U \end{bmatrix}$$

### 2.3 Equations of motion

In this section, we briefly review the equations of motion: velocity, force and acceleration propagation, for a serial chain of interconnected bodies.

#### 2.3.1 Velocity propagation

The velocity propagation for a serial chain of interconnected bodies, shown in Fig. (1), is given by the following intrinsic equation:

$$\mathbf{V}_j - \hat{\mathbf{P}}_{j-1}^t \mathbf{V}_{j-1} = H_j \dot{\mathbf{Q}}_j \quad (1)$$

By using the matrix  $\mathcal{P}$ , Eq. (1) can be expressed in a global form as:

$$\mathcal{P}^t \mathbf{v} = \mathcal{H} \dot{\mathbf{Q}} \quad (2)$$

thus:

$$\mathbf{v} = (\mathcal{P}^t)^{-1} \mathcal{H} \dot{\mathbf{Q}} \quad (3)$$

The end effector spatial velocity  $\mathbf{V}_{N+1}$  is obtained by the following relation:

$$\mathbf{V}_{N+1} - \hat{\mathbf{P}}_N^t \mathbf{V}_N = \mathbf{0} \quad (4)$$

thus:

$$\mathbf{V}_{N+1} = \hat{\mathbf{P}}_N^t \mathbf{V}_N \quad (5)$$

Let  $\beta \in \mathbb{R}^{6 \times 6N}$  be the matrix defined by  $\beta = [\hat{\mathbf{P}}_N^t \ 0 \ \cdots \ 0]$ , Eq. (5) becomes:

$$\mathbf{V}_{N+1} = \beta \mathbf{v} \quad (6)$$

Thus, inserting the expression of  $\mathcal{V}$  from Eq. (3), we obtain:

$$\mathbf{V}_{N+1} = \beta (\mathcal{P}^t)^{-1} \mathcal{H} \dot{\mathbf{Q}} \quad (7)$$

Thus:

$$\mathcal{J} = \beta (\mathcal{P}^t)^{-1} \mathcal{H} \quad (8)$$

### 2.3.2 Acceleration and force propagation

The propagation of accelerations and forces among the links of serial chain are given by:

$$\dot{\mathbf{V}}_j = \hat{\mathbf{P}}_{j-1}^t \dot{\mathbf{V}}_{j-1} + H_j \ddot{\mathbf{Q}}_j \quad (9)$$

$$\mathbf{F}_j = I_j \dot{\mathbf{V}}_j + \hat{\mathbf{P}}_j \mathbf{F}_{j+1} \quad (10)$$

Eqs. (9)-(10) represent the simplified N-E algorithm (with nonlinear terms being excluded) for the serial chain (Luh et al., 1980).

The force  $\mathbf{F}_j$  can be written, by using a rather unconventional decomposition of inter body force of the form (see, for example (Fijany et al., 1995) (Fijany et al., 1997), as:

$$\mathbf{F}_j = H_j \mathbf{F}_{T_j} + W_j \mathbf{F}_{S_j} \quad (11)$$

Where  $\mathbf{F}_{S_j}$  represents the constraint force.

Complement to the Degrees Of Freedom (DOF), Degrees Of Constraint (DOC) are introduced ( $DOC = 6 - DOF$ ).

The projection matrices  $H_j$  and  $W_j$  are taken to satisfy the following orthogonality conditions:

$$H_j^t W_j = W_j^t H_j = 0 \quad (12)$$

$$H_j H_j^t + W_j W_j^t = U \quad (13)$$

$$H_j^t H_j = W_j^t W_j = U \quad (14)$$

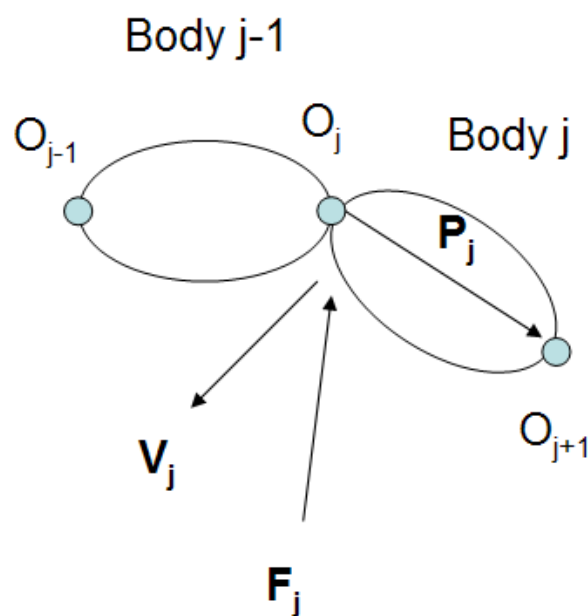


Fig. 1. Joint force and position vector of a serial chain.

### 3. C5 parallel robot

The C5 joint parallel robot (Dafaoui et al., 1998) consists of a static and a mobile part connected together by six actuated segments (Fig. 2 and 3). Each segment is connected to the static part at point  $A_i$  and linked to the mobile part through a C5 passive joint (3DOF in rotation and 2DOF in translation) at point  $B_i$ . Each C5 joint consists of a spherical joint tied to two crossed sliding plates (Fig 4). Each segment is equipped with a ball and a screw linear actuator driven by a DC motor.

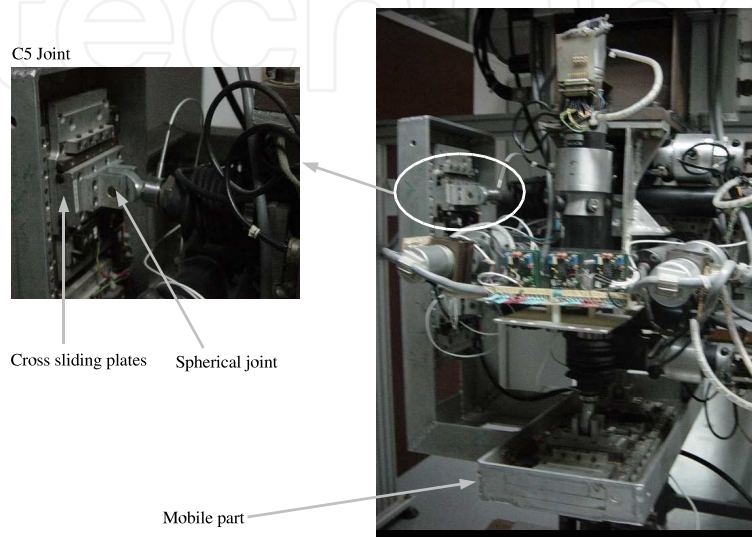


Fig. 2. C5 joint parallel robot

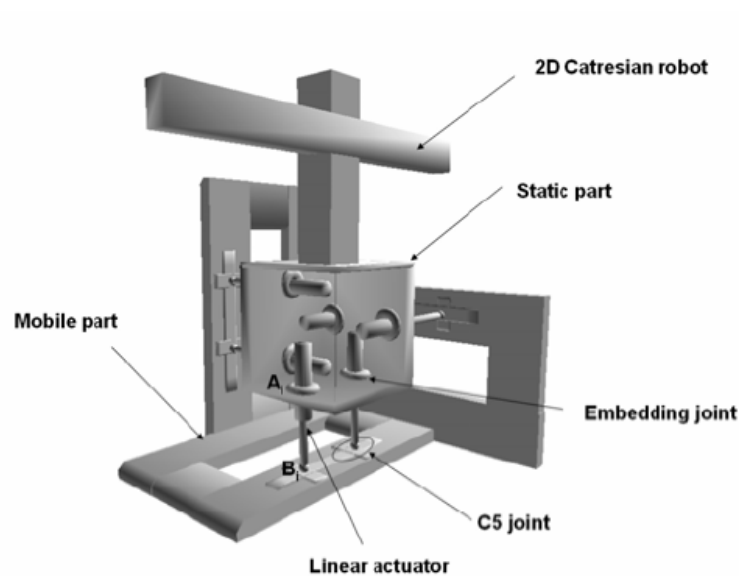


Fig. 3. C5 parallel robot representation.

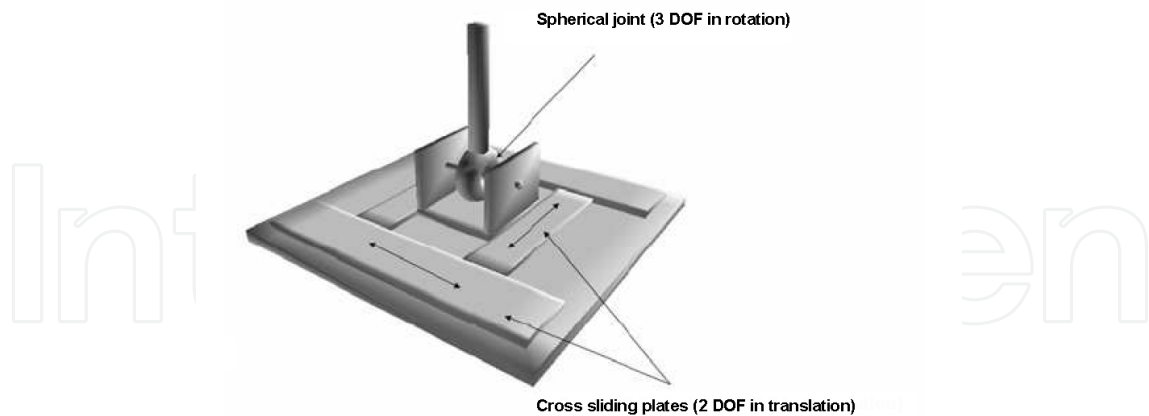


Fig. 4. Details of the C5 joint.

Following notations are used in the description of the parallel robot:

- $R_b$  is the absolute frame, attached to the fixed base:  $R_b = (0, x, y, z)$ .
- $R_p$  is the mobile frame, attached to the mobile part:  $R_p = (C, x_p, y_p, z_p)$ .
- $O$  is the origin of the absolute coordinate system.
- $C$  is the origin of the mobile coordinate system, whose coordinates in the absolute frame are given by:

$$OC_{/R_b} = [x_c \ y_c \ z_c]^t$$

- $A_i$  is the center of the joint between the segment  $i$  and the fixed base:

$$OA_{i/R_b} = [a_i^x \ a_i^y \ a_i^z]^t$$

- $B_i$  is the center of the rotational joint between the segment  $i$  and the mobile part:

$$CB_{i/R_p} = [b_i^x \ b_i^y \ b_i^z]^t$$

- $R$  is the rotation matrix, with elements  $r_{ij}$  (using the RPY formalism), expressing the orientation of the  $R_p$  coordinate system with respect to the  $R_b$  coordinate system. The expression for this matrix is given by:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (15)$$

where:

$$r_{11} = \cos \beta \cos \gamma$$

$$r_{12} = -\cos \beta \sin \gamma$$

$$r_{13} = \sin \beta$$

$$r_{21} = \sin \gamma \cos \alpha + \cos \gamma \sin \beta \sin \alpha$$

$$r_{22} = \cos \alpha \cos \gamma - \sin \alpha \sin \beta \sin \gamma$$



$$\begin{aligned}
 r_{23} &= -\cos \beta \sin \alpha \\
 r_{31} &= \sin \gamma \sin \alpha - \cos \gamma \sin \beta \cos \alpha \\
 r_{32} &= \sin \alpha \cos \gamma + \cos \alpha \sin \beta \sin \gamma \\
 r_{33} &= \cos \beta \cos \alpha
 \end{aligned}$$

- The rotation angles,  $\alpha$ ,  $\beta$  and  $\gamma$ , also called Roll Pitch and Yaw (RPY), describe the rotation of the mobile platform with respect to the fixed part.  $\alpha$  is the rotation around  $x$  axis,  $\beta$  around  $y$  axis and  $\gamma$  around  $z$  axis.
- $\mathbf{X}$  is the task coordinate vector.

$$\mathbf{X} = [\alpha \ \beta \ \gamma \ x_c \ y_c \ z_c]^t$$

In the following, the parallel robot is considered as six serial robots (the six segments) moving a common load (the mobile platform). According to our notation presented in the previous section, we define the following quantities:

- $i$  is the segment index
- $j$  is the body index
- $O_{ij}$  is the  $j^{\text{th}}$  joint center of the segment  $i$
- $P_{ij}$  is the position vector from  $O_{ij}$  to  $O_{i,j+1}$
- $Q_{ij}$  is the position of the  $j^{\text{th}}$  joint of the segment  $i$
- $Q_i = \begin{bmatrix} Q_{i2} \\ Q_{i1} \end{bmatrix} \in \mathbb{R}^6$  is the joint coordinate vector of the segment  $i$
- $\dot{Q}_i = \begin{bmatrix} \dot{Q}_{i2} \\ \dot{Q}_{i1} \end{bmatrix} \in \mathbb{R}^6$  is the joint velocity vector of the segment  $i$
- $\ddot{Q}_i = \begin{bmatrix} \ddot{Q}_{i2} \\ \ddot{Q}_{i1} \end{bmatrix} \in \mathbb{R}^6$  is the joint acceleration vector of the segment  $i$
- $H_{ij}$  is the spatial-axis of joint  $j^{\text{th}}$  joint of the segment  $i$ . For the C5 joint robot the projection matrices  $H_{ij}$  and  $W_{ij}$ , describe in the base frame, are given as:

$$\begin{aligned}
 H_{11} = H_{21} &= [0 \ 0 \ 0 \ 1 \ 0 \ 0]^t \\
 H_{31} = H_{41} &= [0 \ 0 \ 0 \ 0 \ 1 \ 0]^t \\
 H_{51} = H_{61} &= [0 \ 0 \ 0 \ 0 \ 0 \ 1]^t
 \end{aligned}$$

$$H_{12} = H_{22} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$H_{32} = H_{42} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$H_{52} = H_{62} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$W_{12} = W_{22} = [0 \ 0 \ 0 \ 1 \ 0 \ 0]^t$$

$$W_{32} = W_{42} = [0 \ 0 \ 0 \ 0 \ 1 \ 0]^t$$

$$W_{52} = W_{62} = [0 \ 0 \ 0 \ 0 \ 0 \ 1]^t$$

- $V_{ij} = \begin{bmatrix} \omega_{ij} \\ v_{ij} \end{bmatrix} \in \mathbb{R}^6$  is the spatial velocity of the link  $j$  for the segment  $i$
- $I_{ij} \in \mathbb{R}^{6 \times 6}$  : spatial inertia of body  $j$  for the segment  $i$
- $\mathcal{V}_i = Col (V_{ij}) \in \mathbb{R}^{12}$ : global vector of spatial velocities for the segment  $i$
- $\mathcal{H}_i = Diag (H_{ij}) \in \mathbb{R}^{12 \times 6}$ : global matrix of spatial axis for the leg  $i$
- $\mathcal{M}_i$  : Symmetric positive definite (SPD) mass matrix of the segment  $i$

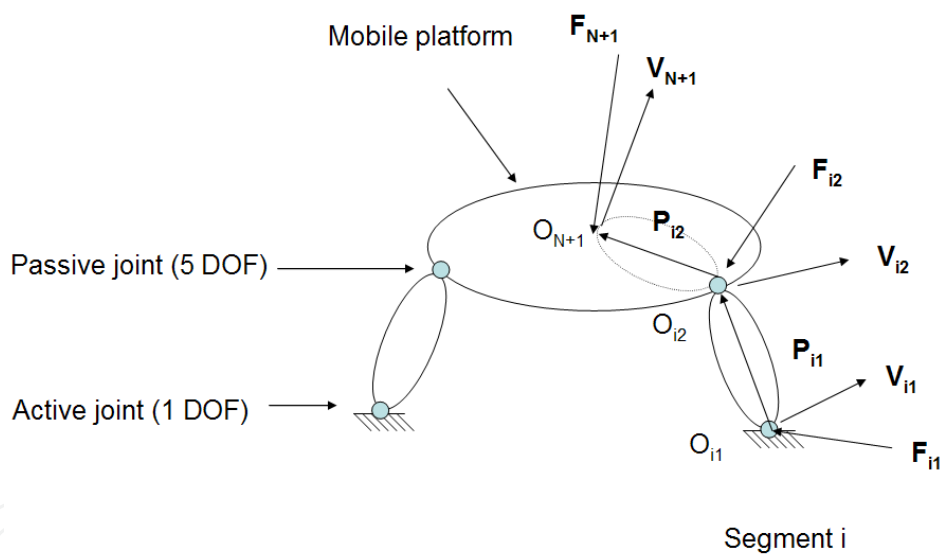


Fig. 5. Force and position vectors.

#### 4. Inverse kinematic model

In this section, we briefly review the methodology used for the inverse kinematic model computation. More details can be found in (Dafaoui et al., 1998).

The inverse kinematic model relates the active joint variables  $Q_a = [Q_{61} \ Q_{51} \ Q_{41} \ Q_{31} \ Q_{21} \ Q_{11}]^t$  to the operational variables which define the position and the orientation of the end effector ( $X$ ). This relation is given by the following equation

(Dafaoui et al., 1998):

$$\begin{aligned}
 Q_{11} &= x_c + \frac{r_{31}(z_c - L) + r_{21}y_c}{r_{11}} \\
 Q_{21} &= x_c + \frac{r_{31}(z_c + L) + r_{21}y_c}{r_{11}} \\
 Q_{31} &= y_c + \frac{r_{12}(x_c - L) + r_{32}z_c}{r_{22}} \\
 Q_{41} &= y_c + \frac{r_{12}(x_c + L) + r_{32}z_c}{r_{22}} \\
 Q_{51} &= z_c + \frac{r_{23}(y_c - L) + r_{13}x_c}{r_{33}} \\
 Q_{61} &= z_c + \frac{r_{23}(y_c + L) + r_{13}x_c}{r_{33}}
 \end{aligned} \tag{16}$$

where:

$$L = \frac{\|A_i A_{i+1}\|}{2} \text{ for } i = 1, 3 \text{ and } 5$$

For the C5 joint parallel robot, the actuators are equidistant from point  $O$  (Fig. 6).

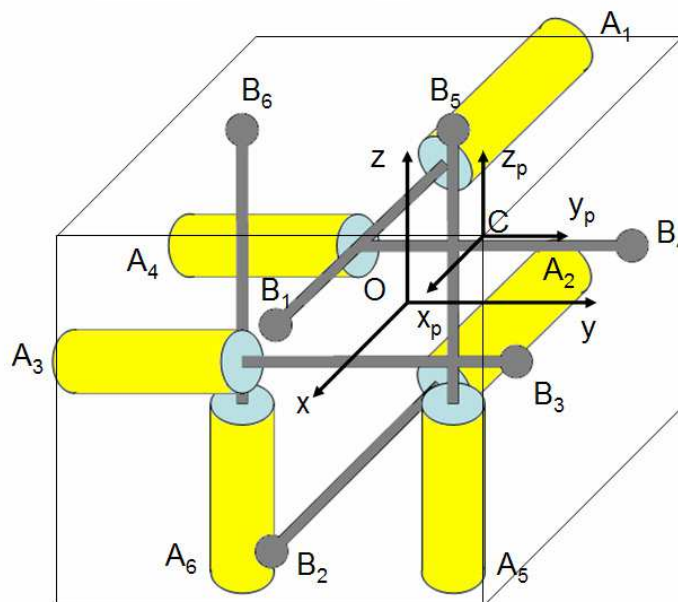


Fig. 6. The spatial arrangement of the C5 joint parallel robot segments.

### 5. Determination of the inverse Jacobian matrix

For parallel robots, the inverse Jacobian matrix computation ( $\mathcal{J}^{-1}$ ) is obtained by the determination of the velocity of point  $B_i$  (Merlet, 2000)(Gosselin, 1996):

$$\dot{O}B_i = v_{N+1} + B_i C \times \omega_{N+1} \tag{17}$$

By using the following:

$$\dot{Q}_{i1} = \dot{O}B_i n_i \tag{18}$$

where  $n_i$  is the unit vector of the segment  $i$ , defined by:

$$n_i = \frac{A_i B_i}{Q_{i1}} \tag{19}$$

and inserting Eq. (17) into Eq. (18), we obtain the following expression:

$$\dot{Q}_{i1} = \mathbf{n}_i \mathbf{v}_{N+1} + \boldsymbol{\omega}_{N+1} (\mathbf{n}_i \times \mathbf{B}_i \mathbf{C}) \quad (20)$$

The  $(6 - i)^{th}$  row of the inverse Jacobian matrix is given as:

$$\left[ (\mathbf{n}_i \times \mathbf{B}_i \mathbf{C})^t \mathbf{n}_i^t \right] \quad (21)$$

The inverse Jacobian matrix of the C5 parallel robot is then given by the following relation:

$$\mathcal{J}^{-1} = \begin{bmatrix} (\mathbf{n}_6 \times \mathbf{B}_6 \mathbf{C})^t \mathbf{n}_6^t \\ (\mathbf{n}_5 \times \mathbf{B}_5 \mathbf{C})^t \mathbf{n}_5^t \\ (\mathbf{n}_4 \times \mathbf{B}_4 \mathbf{C})^t \mathbf{n}_4^t \\ (\mathbf{n}_3 \times \mathbf{B}_3 \mathbf{C})^t \mathbf{n}_3^t \\ (\mathbf{n}_2 \times \mathbf{B}_2 \mathbf{C})^t \mathbf{n}_2^t \\ (\mathbf{n}_1 \times \mathbf{B}_1 \mathbf{C})^t \mathbf{n}_1^t \end{bmatrix} = \begin{bmatrix} (\mathbf{n}_6 \times \mathbf{P}_{62})^t \mathbf{n}_6^t \\ \vdots \\ (\mathbf{n}_1 \times \mathbf{P}_{12})^t \mathbf{n}_1^t \end{bmatrix} \quad (22)$$

As stated before, for parallel robots, unlike the serial robots, the inverse of Jacobian matrix can be directly and efficiently obtained. In fact, the cost of computation of  $\mathcal{J}^{-1}$  from Eq. (22) is  $(18m + 30a)$  where  $m$  and  $a$  denote the cost of multiplication and addition, respectively.

## 6. Factorized expression of the Jacobian matrix

### 6.1 General approach

The differential kinematic model of a manipulator is defined by the relationship between the spatial velocity of the end effector and the vector of generalized coordinate velocities of the robot:  $\mathbf{V}_{N+1} = \mathcal{J} \dot{\mathbf{Q}}_a$ , where  $\mathcal{J}$  is the Jacobian matrix.

For parallel robots, it seems more efficient to compute the Jacobian matrix  $\mathcal{J}$  by inverting the inverse Jacobian matrix  $\mathcal{J}^{-1}$  (see for example (Khalil & Ibrahim, 2007)). In deriving the forward kinematic model of the C5 parallel robot, an analytical expression of the Jacobian matrix is presented in (Dafaoui et al., 1998). From a computational efficiency point of view, such a classical method, which is only applicable to the C5 parallel robot, is not well suited for real-time control.

Here, we present our approach for direct and efficient computation of the Jacobian matrix (Fried et al., 2006). In this approach, an analytical expression of the Jacobian matrix is obtained in factorized form as a product of sparse matrices which achieves a much better computational efficiency.

In our approach, the parallel robot is considered as a multi-robot system, composed of serial robots (the segments) moving a common load (the mobile platform). A relationship between the Jacobian matrix of the parallel robot ( $\mathcal{J}$ ) to the Jacobian matrix of each segment ( $\mathcal{J}_i$ ) is first derived.

The principle of this approach consists of first computing the Jacobian matrix for each leg considered as an open serial chain. Secondly, the closing constraint is determined, allowing the computation of the parallel robot Jacobian matrix.

The Jacobian matrix  $\mathcal{J}$  of the parallel robot is obtained by the closing constraint determination of the kinematic chain. This determination can be obtained by expressing the actuated joint velocity  $\dot{\mathbf{Q}}_a$  of the parallel robot in function of vectors  $\dot{\mathbf{Q}}_i$  associated to each segment  $i$ . Let the matrix  $\Pi_i$  be defined as:

$$\dot{\mathbf{Q}}_i = \Pi_i \dot{\mathbf{Q}}_a \quad (23)$$

Inserting Eq. (23) into Eq. (7), we obtain:

$$\mathbf{V}_{N+1} = \beta_i (\mathcal{P}_i^t)^{-1} \mathcal{H}_i \Pi_i \dot{\mathbf{Q}}_a \quad (24)$$

Therefore, a factorized expression of the parallel robot Jacobian matrix is obtained as:

$$\mathcal{J} = \beta_i (\mathcal{P}_i^t)^{-1} \mathcal{H}_i \Pi_i \quad (25)$$

The matrices  $\mathcal{J}$  and  $\mathcal{J}_i$  are related by the following relationship:

$$\mathcal{J} = \mathcal{J}_i \Pi_i \quad (26)$$

The computation of matrix of  $\Pi_i$  depends on the considered parallel robot's structure. In the following, we present the computation of this matrix for the C5 parallel robot.

## 6.2 Application to the C5 parallel robot

Let  $\mathbf{P}_{i2} = [x_i \ y_i \ z_i]^t$  denote the position vector from  $B_i$  to C:

$$\mathbf{P}_{i2} = \mathbf{B}_i \mathbf{C} / R_b = -Q_{i1} \mathbf{n}_i + \mathbf{A}_i \mathbf{O} + \mathbf{OC} \quad (27)$$

The spatial arrangement of the segments (see Fig. 6) is as follows:

- The segments 1 and 2 are in the direction of the  $x$ -axis ( $\mathbf{n}_i = [1 \ 0 \ 0]^t$  for  $i = 1, 2$ ).
- The segments 3 and 4 are in the direction of the  $y$ -axis ( $\mathbf{n}_i = [0 \ 1 \ 0]^t$  for  $i = 3, 4$ ).
- The segments 5 and 6 are in the direction of the  $z$ -axis ( $\mathbf{n}_i = [0 \ 0 \ 1]^t$  for  $i = 5, 6$ ).

Thus, we deduce the following relations:

$$\begin{aligned} y_1 &= y_2 = y_c \\ z_3 &= z_4 = z_c \\ x_5 &= x_6 = x_c \end{aligned} \quad (28)$$

The global vector of articular coordinate velocity of the leg  $i$  is given by:

$$\dot{\mathbf{Q}}_i = [\dot{w}_{p_i} \ \dot{u}_{p_i} \ \dot{\gamma}_{p_i} \ \dot{\beta}_{p_i} \ \dot{\alpha}_{p_i} \ \dot{Q}_{i1}]^t \quad (29)$$

where  $\dot{u}_{p_i}$  and  $\dot{w}_{p_i}$  are translation velocities due to the crossed sliding plates.

### 6.2.1 Determination of matrix $\Pi_i$

The matrix  $\Pi_i$  given in Eq. (23) is obtained as follows. We have:

$$\begin{bmatrix} \dot{w}_{p_i} \\ \dot{u}_{p_i} \\ \dot{\gamma}_{p_i} \\ \dot{\beta}_{p_i} \\ \dot{\alpha}_{p_i} \\ \dot{Q}_{i1} \end{bmatrix} = \Pi_i \begin{bmatrix} \dot{Q}_{61} \\ \dot{Q}_{51} \\ \dot{Q}_{41} \\ \dot{Q}_{31} \\ \dot{Q}_{21} \\ \dot{Q}_{11} \end{bmatrix} \quad (30)$$

The elements  ${}^i\pi_{jk}$  of the matrix  $\Pi_i$  are computed by using Eq. (7). This equation is true for  $i = 1$  to 6, thus:

$$\beta_i (\mathcal{P}_i^t)^{-1} \mathcal{H}_i \dot{\mathbf{Q}}_i = \beta_j (\mathcal{P}_j^t)^{-1} \mathcal{H}_j \dot{\mathbf{Q}}_j \quad (31)$$

for  $i$  and  $j = 1$  to 6. From Eq. (31), we can show that for all  $i, j = 1, \dots, 6$ , we have the following relations:

$$\begin{cases} \dot{\alpha}_{p_i} = \dot{\alpha}_{p_j} \\ \dot{\beta}_{p_i} = \dot{\beta}_{p_j} \\ \dot{\gamma}_{p_i} = \dot{\gamma}_{p_j} \end{cases} \quad (32)$$

After some manipulations on relation (31), we obtain:

- For  $i = 1$  and  $j = 2$

$$\dot{Q}_{11} = (z_2 - z_1) \dot{\beta}_{p_i} + (y_1 - y_2) \dot{\gamma}_{p_i} + \dot{Q}_{21} \quad (33)$$

- For  $i = 3$  and  $j = 4$ :

$$\dot{Q}_{31} = (z_3 - z_4) \dot{\alpha}_{p_i} + (x_4 - x_3) \dot{\gamma}_{p_i} + \dot{Q}_{41} \quad (34)$$

- For  $i = 5$  and  $j = 6$ :

$$\dot{Q}_{51} = (y_6 - y_5) \dot{\alpha}_{p_i} + (x_5 - x_6) \dot{\beta}_{p_i} + \dot{Q}_{61} \quad (35)$$

- For  $i = 1$  and  $j = 3$ :

$$\dot{u}_{p_1} = (z_1 - z_3) \dot{\alpha}_{p_i} + (x_3 - x_1) \dot{\gamma}_{p_i} + \dot{Q}_{31} \quad (36)$$

- For  $i = 1$  and  $j = 5$ :

$$\dot{w}_{p_1} = (y_5 - y_1) \dot{\alpha}_{p_i} + (x_1 - x_5) \dot{\beta}_{p_i} + \dot{Q}_{51} \quad (37)$$

From Eq. (28), we have  $y_1 = y_2, z_3 = z_4$  and  $x_5 = x_6$ . Thus, the Eqs. (33, 34, and 35) can be written as:

$$\begin{aligned} \dot{Q}_{11} &= (z_2 - z_1) \dot{\beta}_{p_i} + \dot{Q}_{21} \\ \dot{Q}_{31} &= (x_4 - x_3) \dot{\gamma}_{p_i} + \dot{Q}_{41} \\ \dot{Q}_{51} &= (y_6 - y_5) \dot{\alpha}_{p_i} + \dot{Q}_{61} \end{aligned} \quad (38)$$

From Eqs. (30), (36), (37) and (38), the matrix  $\Pi_1$  is computed as:

$$\Pi_1 = \begin{bmatrix} \frac{y_5 - y_1}{y_5 - y_6} & \frac{y_6 - y_1}{y_6 - y_5} & 0 & 0 & \frac{x_1 - x_5}{z_1 - z_2} & \frac{x_1 - x_5}{z_2 - z_1} \\ \frac{z_1 - z_3}{y_5 - y_6} & \frac{z_1 - z_3}{y_6 - y_5} & \frac{x_3 - x_1}{x_3 - x_4} & \frac{x_4 - x_1}{x_4 - x_3} & 0 & 0 \\ 0 & 0 & \frac{1}{x_3 - x_4} & \frac{1}{x_4 - x_3} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{z_1 - z_2} & \frac{1}{z_2 - z_1} \\ \frac{1}{y_5 - y_6} & \frac{1}{y_6 - y_5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (39)$$

The computational cost of explicit construction of the matrix  $\Pi_1$  is  $(17m + 28a)$  wherein the cost of division has been taken to be the same as multiplication.

Considering the matrix  $\Pi_1$ , the expression of the jacobian matrix is given by:

$$\mathcal{J} = \mathcal{J}_1 \Pi_1 \quad (40)$$

With:

$$\mathcal{J}_1 = \beta_1 \mathcal{P}_1^{-t} \mathcal{H}_1 = \begin{bmatrix} \hat{P}_{12}^t & 0 \end{bmatrix} \begin{bmatrix} U & \hat{P}_{11}^t \\ 0 & U \end{bmatrix} \begin{bmatrix} H_{12} & 0 \\ 0 & H_{11} \end{bmatrix}$$

Thus:

$$\mathcal{J}_1 = \begin{bmatrix} \hat{P}_{12}^t & H_{12} & \hat{P}_{12}^t & \hat{P}_{11}^t & H_{11} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -y_1 & z_1 & 0 & 1 \\ 0 & 1 & x_1 & 0 & -z_1 & 0 \\ 1 & 0 & 0 & -x_1 & y_1 & 0 \end{bmatrix} \quad (41)$$

Note the highly sparse structure of the matrix  $\mathcal{J}_1$ . In fact, if the matrix  $\Pi_1$  is already computed then the computation of the matrix  $\mathcal{J}_1$  does not require any operation. However, if the explicit computation of  $\mathcal{J}$  is needed it can be then computed as  $\mathcal{J} = \mathcal{J}_1 \Pi_1$ . Exploiting the sparse structure of matrices  $\mathcal{J}_1$  and  $\Pi_1$ , this computation can be performed with a cost of  $29m + 37a$ .

## 7. Computation of joint accelerations of the segments

The conventional approach to calculate  $\ddot{\mathbf{Q}}_i$  is based on time derivation of Eq. (7) as:

$$\ddot{\mathbf{Q}}_i = \mathcal{J}_i^{-1} \dot{\mathbf{V}}_{N+1} + \frac{d}{dt} \mathcal{J}_i^{-1} \mathbf{V}_{N+1} \quad (42)$$

Eq. (42) represents the second-order inverse kinematic model of the segment  $i$ .

In the following, we propose a new and more efficient approach for computation of  $\ddot{\mathbf{Q}}_i$ . From the propagation of acceleration given in Eq. (9), we can derive the following relations:

$$\dot{\mathbf{V}}_{N+1} = \hat{\mathbf{P}}_{i2}^t \dot{\mathbf{V}}_{i2} \quad (43)$$

$$\dot{\mathbf{V}}_{i2} = H_{i2} \ddot{\mathbf{Q}}_{i2} + \hat{\mathbf{P}}_{i1}^t \dot{\mathbf{V}}_{i1} + \frac{d}{dt} \hat{\mathbf{P}}_{i1}^t \mathbf{V}_{i1} \quad (44)$$

$$\dot{\mathbf{V}}_{i1} = H_{i1} \ddot{\mathbf{Q}}_{i1} \quad (45)$$

Considering the orthogonality properties of the projection matrices  $H_{ij}$  and  $W_{ij}$  given in Eq. (12), by multiplying both sides of Eq. (44) by  $W_{i2}^t$  we get:

$$W_{i2}^t \dot{\mathbf{V}}_{i2} = \underbrace{W_{i2}^t H_{i2} \ddot{\mathbf{Q}}_{i2}}_0 + W_{i2}^t \hat{\mathbf{P}}_{i1}^t \dot{\mathbf{V}}_{i1} + \underbrace{W_{i2}^t \frac{d}{dt} \hat{\mathbf{P}}_{i1}^t H_{i1}}_0 \dot{\mathbf{Q}}_{i1} = W_{i2}^t \hat{\mathbf{P}}_{i1}^t \dot{\mathbf{V}}_{i1} \quad (46)$$

Note that the above projection indeed eliminates the term  $\ddot{\mathbf{Q}}_{i2}$  from the equation. From Eqs. (45) and (46), we then obtain:

$$\ddot{\mathbf{Q}}_{i1} = (W_{i2}^t \hat{\mathbf{P}}_{i1}^t H_{i1})^{-1} W_{i2}^t \dot{\mathbf{V}}_{i2} \quad (47)$$

Where the term defined by  $(W_{i2}^t \hat{\mathbf{P}}_{i1}^t H_{i1})^{-1}$  is a scalar.

Again, considering the properties given by Eq. (14), multiplying both sides of Eq. (44) by  $H_{i2}^t$  we get:

$$\ddot{\mathbf{Q}}_{i2} = H_{i2}^t \dot{\mathbf{V}}_{i2} - H_{i2}^t \hat{\mathbf{P}}_{i1}^t \dot{\mathbf{V}}_{i1} - \underbrace{H_{i2}^t \frac{d}{dt} \hat{\mathbf{P}}_{i1}^t H_{i1}}_0 \dot{\mathbf{Q}}_{i1} = H_{i2}^t \dot{\mathbf{V}}_{i2} - H_{i2}^t \hat{\mathbf{P}}_{i1}^t \dot{\mathbf{V}}_{i1} \quad (48)$$

Note that,  $H_{i2}^t \frac{d}{dt} \hat{\mathbf{P}}_{i1}^t H_{i1} = \mathbf{0}$  since  $\frac{d}{dt} \hat{\mathbf{P}}_{i1}^t$  is along  $H_{i1}$  and as can be seen from the description of  $H_{i1}$  and  $H_{i2}$  in section 3,  $H_{i2}^t H_{i1} = \mathbf{0}$ .

The joint accelerations of the segment  $i$  are then computed in four steps as follows:

1. Compute  $\dot{\mathbf{V}}_{i2}$  from Eq. (43)
2. Compute  $\ddot{\mathbf{Q}}_{i1}$  from Eq. (47)
3. Compute  $\dot{\mathbf{V}}_{i1}$  from Eq. (45)
4. Compute  $\ddot{\mathbf{Q}}_{i2}$  from Eq. (48)

Exploiting the sparse structure of matrices  $H_{i1}$ ,  $H_{i2}$ , and  $W_{i2}$  as well as an appropriate scheme for projection of the equations, the cost of computing  $\ddot{\mathbf{Q}}_i = \begin{bmatrix} \ddot{\mathbf{Q}}_{i2} \\ \ddot{\mathbf{Q}}_{i1} \end{bmatrix}$  for each segment is of  $(10m + 10a)$ .

## 8. Inverse dynamic model

The inverse dynamic computation for the parallel robot consists of determination of the required active joints torques to achieve a desired acceleration of the mobile platform, which is needed for accurate control of the robot. In this section, we review the approach proposed by Khalil et al in (Khalil & Guegan, 2004) (Khalil & Ibrahim, 2007).

Our contribution is to show that by using the factorized expression of the Jacobian matrix, and the new formulation of acceleration joints, a significantly better computational efficiency can be achieved for the inverse dynamic calculation.

### 8.1 Computation of inverse dynamic model

The dynamical equation of motion for each segment is given by:

$$\mathcal{M}_i \ddot{\mathbf{Q}}_i + \mathbf{C}_i + \mathbf{G}_i + \mathcal{J}_{B_i}^t \mathbf{F}_{i2} = \mathbf{\Gamma}_i \quad (49)$$

Where:

- $\mathbf{F}_{i2}$  is the force exerted to the mobile platform by the segment  $i$  (Fig. 5).
- $\mathcal{J}_{B_i}$  is the Jacobian matrix of the segment  $i$ , computed to the point  $B_i$ . The expression of  $\mathcal{J}_{B_i}$  is given by:

$$\mathcal{J}_{B_i} = \hat{\mathbf{P}}_{i2}^{-t} \mathcal{J}_i \quad (50)$$

- $\mathbf{C}_i + \mathbf{G}_i$  represents the contribution of the Coriolis, centrifugal, and gravitational terms.
- $\mathbf{\Gamma}_i$  represents the joint force vector of the segment  $i$ .

The contact forces exerted to the mobile platform by the segments, shown in Fig. 5, are computed from Eq. (49) as:

$$\mathbf{F}_{i2} = -\mathcal{J}_{B_i}^{-t} (\mathcal{M}_i \ddot{\mathbf{Q}}_i + \mathbf{C}_i + \mathbf{G}_i) + \mathcal{J}_{B_i}^{-t} \mathbf{\Gamma}_i \quad (51)$$

The dynamic behavior of the mobile platform is given by the following relation:

$$\mathbf{F}_{N+1} = \Lambda_C \dot{\mathbf{V}}_{N+1} + (\mathbf{G}_C + \mathbf{C}_C) \quad (52)$$

Where:

- $\mathbf{F}_{N+1}$  is the spatial force applied at the point  $C$ , representing the contribution of the contact forces  $\mathbf{F}_{iN}$  propagated to the point  $C$ :

$$\mathbf{F}_{N+1} = \begin{bmatrix} \mathbf{n}_{N+1} \\ \mathbf{f}_{N+1} \end{bmatrix} = \sum_{i=1}^6 \hat{\mathbf{P}}_{i2}^t \mathbf{F}_{i2} \quad (53)$$

- $\Lambda_C \in \mathbb{R}^{6 \times 6}$  is the spatial inertia matrix of the mobile platform:

$$\Lambda_C = \begin{bmatrix} I_C & m_C \tilde{\mathbf{G}}_C \\ -m_C \tilde{\mathbf{G}}_C & m_C U \end{bmatrix} \quad (54)$$



- $m_C$  is the platform mass
- $I_C \in \mathbb{R}^{3 \times 3}$  is the inertia tensor of the mobile platform expressed in the mobile platform center of mass and projected in the fixed frame  $R_b$ :

$$I_C = R I_{C/R_m} R^t \quad (55)$$

- $C_C \in \mathbb{R}^6$  is the vector of Coriolis and centrifugal forces:

$$C_C = \begin{bmatrix} \tilde{\omega}_{N+1} I_C \omega_{N+1} \\ -m_C \tilde{\omega}_{N+1} \tilde{G}C \omega_{N+1} \end{bmatrix} \quad (56)$$

- $G_C \in \mathbb{R}^6$  is the vector of gravitational forces:

$$G_C = \begin{bmatrix} -m_C \tilde{G}C \\ -m_C U \end{bmatrix} g \quad (57)$$

- $g$  being the acceleration vector of gravity

Substituting (51) in (53), we obtain:

$$F_{N+1} = \sum_{i=1}^6 \hat{P}_{i2}^t \left( \mathcal{J}_{B_i}^{-t} \Gamma_i - \mathcal{J}_{B_i}^{-t} (\mathcal{M}_i \ddot{Q}_i + C_i + G_i) \right) \quad (58)$$

The active joint forces vector is given by:

$$\Gamma = [\Gamma_{61} \ \Gamma_{51} \ \Gamma_{41} \ \Gamma_{31} \ \Gamma_{21} \ \Gamma_{11}]^t$$

We have  $\sum_{i=1}^6 \hat{P}_{i2}^t \mathcal{J}_{B_i}^{-t} \Gamma_i = \mathcal{J}^{-1} \Gamma$  where  $\mathcal{J}^{-1}$  is the inverse Jacobian matrix of the parallel robot. Eq. (58) can be rewritten as:

$$F_{N+1} = \mathcal{J}^{-1} \Gamma - \sum_{i=1}^6 \hat{P}_{i2}^t \mathcal{J}_{B_i}^{-t} (\mathcal{M}_i \ddot{Q}_i + C_i + G_i) \quad (59)$$

The inverse dynamic model, given by Kahlil et al in (Khalil & Ibrahim, 2007) is then expressed by:

$$\Gamma = \mathcal{J}^t \left[ F_{N+1} + \sum_{i=1}^6 \hat{P}_{i2}^t \mathcal{J}_{B_i}^{-t} (\mathcal{M}_i \ddot{Q}_i + C_i + G_i) \right] \quad (60)$$

## 8.2 Computational complexity analysis

The inverse dynamic model, given in Eq. (60), is computed in six steps:

1. Computation of joint accelerations from Eq. (42)
2. Computation of the vector defined as  $T_i = \mathcal{M}_i \ddot{Q}_i + C_i + G_i$  with the recursive Newton-Euler algorithm (Cork, 1996).
3. Computation of the vector resulting from the propagation of forces exerted on the mobile platform by all the segments as  $\Phi = \sum_{i=1}^6 \hat{P}_{i2}^t T_i$
4. Computation of  $F_{N+1}$  from Eq. (52)
5. Computation of the vector defined as  $K = F_{N+1} + \Phi$

### 6. Computation of the vector $\Gamma = \mathcal{J}^t \mathbf{K}$

The computation of the last step, as discussed by Khalil et al in (Khalil & Guegan, 2004) (Khalil & Ibrahim, 2007), is performed by first computing  $\mathcal{J}^{-t}$  from Eq. (22). As a result, the computation of  $\Gamma$  requires solving a linear system of dimension 6 as:

$$\mathcal{J}^{-t} \Gamma = \mathbf{K} \quad (61)$$

The computation cost for the derivation of  $\mathcal{J}^{-1}$ , from Eq. (22), is  $(18m + 30a)$ . The cost of solving the linear system of dimension 6 in Eq. (61) is of  $(116m + 95a)$  wherein the cost of division and multiplication are taken to be the same. Therefore, the total cost of Step 6 is of  $(134m + 125a)$ .

Our contribution concerns the improvement of the efficiency of the inverse dynamic computational by using the factorized expression of the Jacobian matrix and the new formulation of the joint accelerations. Using these formulations, the computations can be then performed as follows:

Step 1 is performed with a cost of  $(10m + 10a)$  for each segment by using the expression given in Eqs. (47) and (48). This represents a much better efficiency than that of using Eq. (42).

Steps 2, 3, 4 and 5 are the same as in Khalil et al in (Khalil & Guegan, 2004) (Khalil & Ibrahim, 2007).

For computation of Step 6, the factorized expression of  $\mathcal{J}$ , given by Eq. (40) is used. To this end, we have:

$$\Gamma = \Pi_1^t \mathcal{J}_1^t \mathbf{K} \quad (62)$$

Note that, the above equation not only eliminates the need for solving a linear system but it also does not require the explicit computation of neither  $\mathcal{J}^{-1}$  nor  $\mathcal{J}$ . By exploiting the sparse structure of matrices  $\Pi_1^t$  and  $\mathcal{J}_1$ , the cost of computation of the vector  $\Gamma$  from Eq. (62) is of  $(65m + 67a)$ , which represents a much better efficiency, almost half of the computations, compared by using Eq. (40).

## 9. Simulation of the inverse dynamic model

To validate our inverse dynamic model of the C5 joint parallel robot, a simulation under Matlab environment is presented in this section. The dynamic parameters used for the simulation are given in appendix.

The trajectory profile used for this study is given as follows:

- Fig. 7 shows cartesian trajectory of the mobile platform for a constant orientation ( $\alpha = 10^\circ$ ,  $\beta = 15^\circ$  and  $\gamma = 20^\circ$ ).
- Fig. 8 and 9 show respectively the linear velocity and linear acceleration of the mobile platform along the given trajectory.
- The active joint forces are computed using inverse dynamic model given by Eq. (60). Fig. 10 shows the active joint force evolution along the trajectory.

The simulation results show the feasibility of the proposed approach. The terms of Eq (60) have been computed, and especially, the joint accelerations and the Jacobian matrix in its factorized form, which represent our main contribution in this present paper.

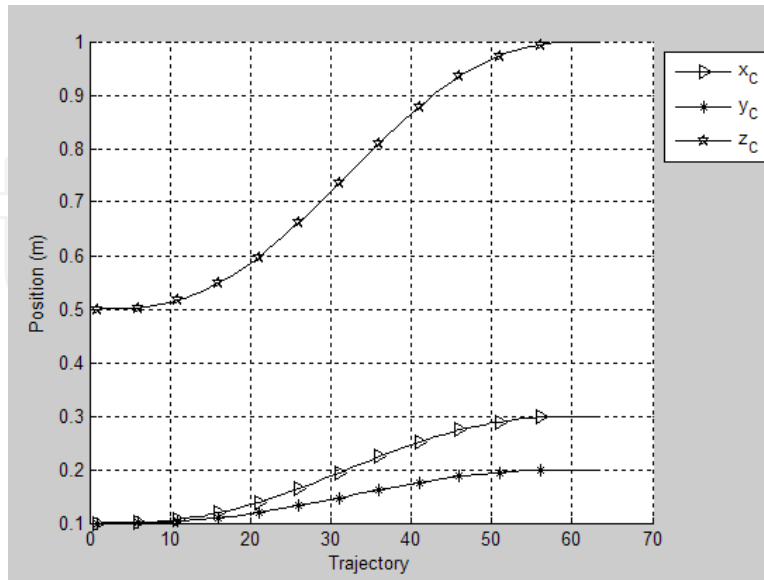


Fig. 7. Mobile platform cartesian trajectory.

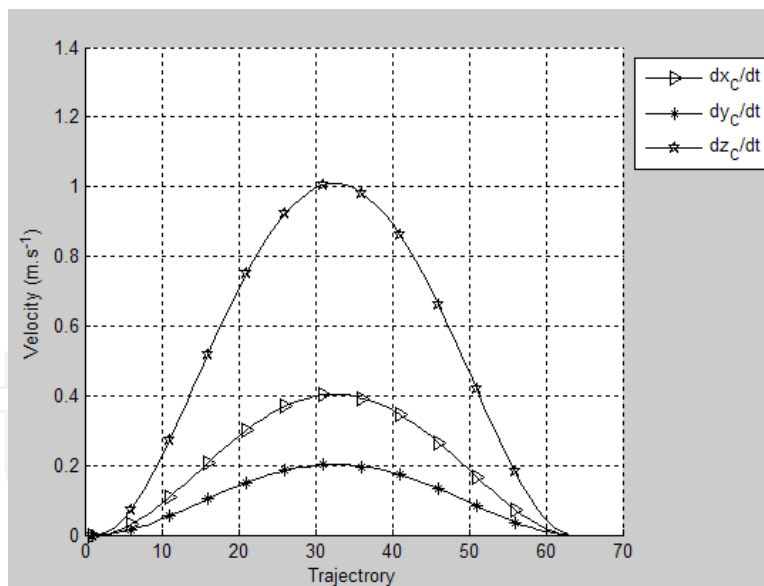


Fig. 8. Mobile platform linear velocity profile.

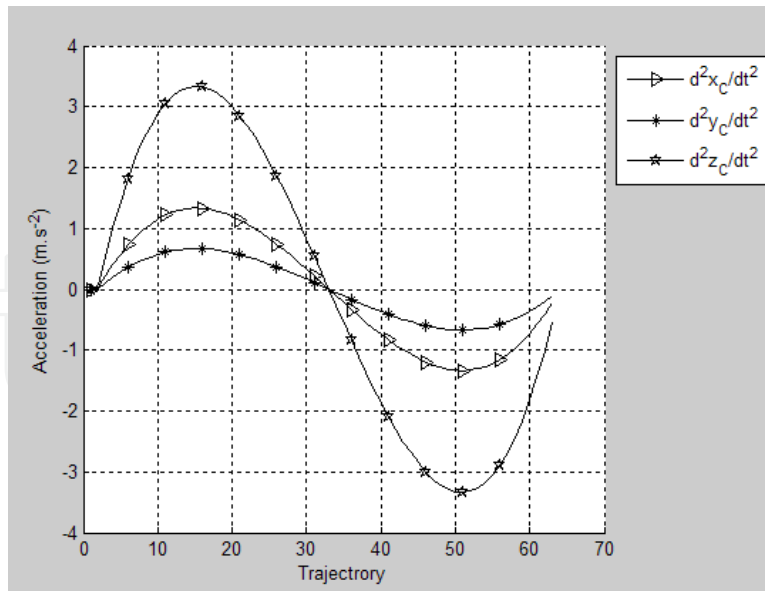


Fig. 9. Mobile platform linear acceleration profile.

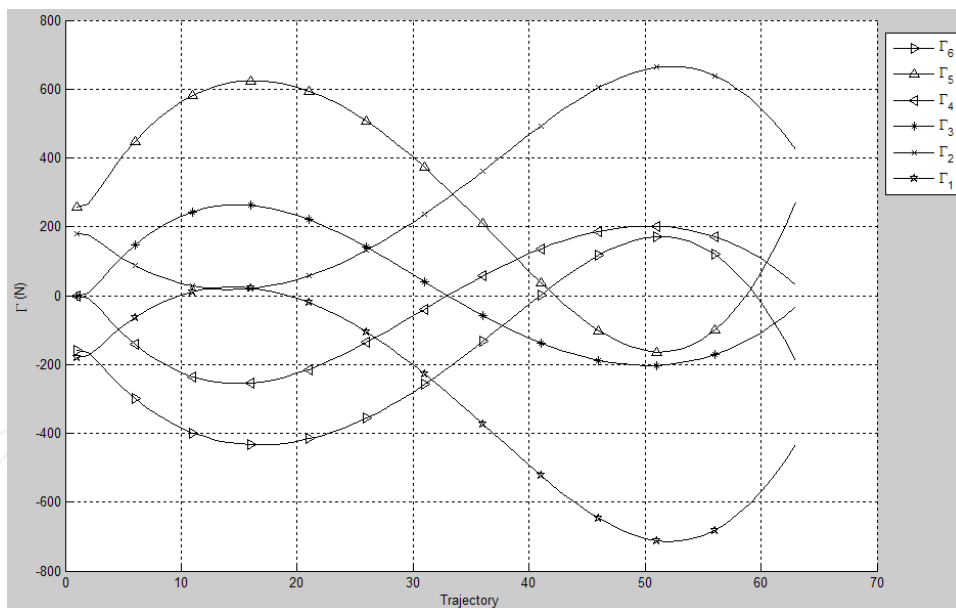


Fig. 10. Active joint forces  $\Gamma_i$  along the trajectory.

## 10. Conclusion

In this paper, we first presented a review of various proposed schemes for kinematics and inverse dynamics computation of the C5 parallel robot. We presented a new and efficient scheme for derivation of the Jacobian matrix in a factored form as a product two highly sparse matrices. We also presented a new scheme for fast and direct computation of the joint accelerations, given the desired acceleration of the mobile platform. We also reviewed the proposed scheme by Khalil et al (Khalil & Guegan, 2004)(Khalil & Ibrahim, 2007) for computation of the inverse dynamics of the C5 parallel robot. We showed that by using our new scheme for computation of the joint acceleration as well as the use of Jacobian in a factored form a much better efficiency in the computation can be achieved.

The proposed methodology by Khalil et al (Khalil & Guegan, 2004)(Khalil & Ibrahim, 2007) for computation of the inverse dynamics is very systematic and indeed has been applied to several parallel robots. However, we still believe that more efficient formulations can be derived by a further exploitation of specific structures of the parallel robots. We are currently investigating new approaches for the inverse dynamics computation of parallel robots. We are also extending our derivation of the Jacobian matrix in the factored form to other parallel robots.

## 11. Appendix: dynamic parameters

Body	Mass	Inertia
1	2 kg	$J_{G_{i1}} = \begin{bmatrix} 0.0025 & 0 & 0 \\ 0 & 0.00125 + 0.1667 (Q_{i1})^2 & 0 \\ 0 & 0 & 0.00125 + 0.1667 (Q_{i1})^2 \end{bmatrix}$
2	1.67 kg	$J_{G_{i2}} = \begin{bmatrix} 0.00167 & 0 & 0 \\ 0 & 0.0084 & 0 \\ 0 & 0 & 0.0084 \end{bmatrix}$

Table 1. Dynamic parameters of the segment  $i = 1$  and 2

Body	Mass	Inertia
1	2 kg	$J_{G_{i1}} = \begin{bmatrix} 0.00125 + 0.1667 (Q_{i1})^2 & 0 & 0 \\ 0 & 0.0025 & 0 \\ 0 & 0 & 0.00125 + 0.1667 (Q_{i1})^2 \end{bmatrix}$
2	1.67 kg	$J_{G_{i2}} = \begin{bmatrix} 0.0084 & 0 & 0 \\ 0 & 0.00167 & 0 \\ 0 & 0 & 0.0084 \end{bmatrix}$

Table 2. Dynamic parameters of the segment  $i = 3$  and 4

Body	Mass	Inertia		
1	2 kg	$J_{G_{i1}} = \begin{bmatrix} 0.00125 + 0.1667 (Q_{i1})^2 & 0 & 0 \\ 0 & 0.00125 + 0.1667 (Q_{i1})^2 & 0 \\ 0 & 0 & 0.0025 \end{bmatrix}$		
2	1.67 kg	$J_{G_{i2}} = \begin{bmatrix} 0.0084 & 0 & 0 \\ 0 & 0.0084 & 0 \\ 0 & 0 & 0.00167 \end{bmatrix}$		

Table 3. Dynamic parameters of the segment  $i = 5$  and 6

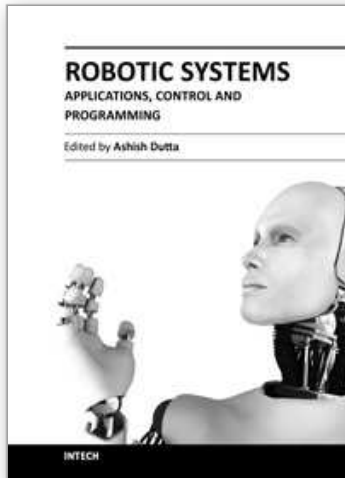
Body	Mass	Inertia		
Mobile platform	10 kg	$I_C = \begin{bmatrix} 0.375 & 0 & 0 \\ 0 & 0.1875 & 0 \\ 0 & 0 & 0.1875 \end{bmatrix}$		

Table 4. Dynamic parameters of the mobile platform

## 12. References

- Bi, Z. M. & Lang, S. Y. T. (2006). Kinematic and dynamic models of a tripod system with a passive leg, *IEEE/ASME Transactions on Mechatronics* Vol. 11(No. 1): 108–112.
- Cork, P. I. (1996). A robotics toolbox for matlab, *IEEE Robotics and Automation Magazine* Vol. 3: 24–32.
- Dafaoui, M., Amirat, Y., François, C. & Pontnau, J. (1998). Analysis and design of a six dof parallel robot. modeling, singular configurations and workspace, *IEEE Trans. Robotics and Automation* Vol. 14(No. 1): 78–92.
- Dasgupta, B. & Choudhury, P. (1999). A general strategy based on the newton-euler approach for the dynamic formulation of parallel manipulators, *Mech. Mach. Theory* Vol. 4: 61–69.
- Fichter, E. F. (1986). A stewart platform based manipulator: general theory and practical construction, *International Journal of Robotics Research* Vol. 5(No. 2): 157–182.
- Fijany, A., Djouani, K., Fried, G. & J. Pontnau, J. (1997). New factorization techniques and fast serial and parallel algorithms for operational space control of robot manipulators, *Proceedings of IFAC, 5<sup>th</sup> Symposium on Robot Control*, Nantes, France, pp. 813–820.
- Fijany, A., Sharf, I. & Eleuterio, G. M. T. (1995). Parallel  $o(\log n)$  algorithms for computation of manipulator forward dynamics, *IEEE Trans. Robotics and Automation* Vol. 11(No. 3): 389–400.
- Fried, G., Djouani, K., Borojeni, D. & Iqbal, S. (2006). Jacobian matrix factorization and singularity analysis of a parallel robot, *WSEAS Transaction on Systems* Vol. 5/6: 1482–1489.
- Gosselin, C. M. (1996). Parallel computational algorithms for the kinematics and dynamics of planar and spatial parallel manipulator, *Journal of Dynamic System, Measurement and Control* Vol. 118: 22–28.
- Khalil, W. & Guegan, S. (2004). Inverse and direct dynamic modeling of gough-stewart robots, *IEEE Transactions on Robotics* Vol. 20(No. 4): 745–762.

- Khalil, W. & Ibrahim, O. (2007). General solution for the dynamic modeling of parallel robots, *Journal of Intelligent and Robotic Systems* Vol. 49(No. 1): 19–37.
- Khan, W. A. (2005). Recursive kinematics and inverse dynamics of a 3r planar parallel manipulator, *Journal of Dynamic System, Measurement and Control* Vol. 4(No. 4): 529–536.
- Leroy, N., Kokosy, A. M. & Perruquetti, W. (2003). Dynamic modeling of a parallel robot. Application to a surgical simulator, *Proceedings of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, pp. 4330–4335.
- Luh, J. Y. S., Walker, W. & Paul, R. P. C. (1980). On line computational scheme for mechanical manipulator, *ASME Journal of Dynamic System, Meas Control* Vol. 102: 69–76.
- Merlet, J. P. (2000). *Parallel Robots*, Kluwer Academic Publishers.
- Merlet, J. P. (2002). Optimal design for the micro parallel robot mips, *Proceedings of IEEE International Conference on Robotics and Automation, ICRA '02*, Washington DC, USA, pp. 1149–1154.
- Nakamura, Y. & Ghodoussi, M. (1989). Dynamic computation of closed link robot mechanism with non redundant actuators, *IEEE Journal of Robotics and Automation* Vol. 5: 295–302.
- Neugebauer, R., Schwaar, M. & F. Wieland, F. (1998). Accuracy of parallel structured of machine tools, *Proceedings of the International Seminar on Improving Tool Performance*, Spain, pp. 521–531.
- Plitea, N., Pisla, D., Vaida, C., Gherman, B. & Pisla, A. (2008). Dynamic modeling of a parallel robot used in minimally invasive surgery, *Proceedings of EUCOMES 08*, Cassino, Italy, pp. 595–602.
- Poignet, P., Gautier, M., Khalil, W. & Pham, M. T. (2002). Modeling, simulation and control of high speed machine tools using robotics formalism, *Journal of Mechatronics* Vol. 12: 461–487.
- Reboulet, C. (1988). *Robots parallèles*, Eds Hermès, Paris.
- Stewart, D. (1965). A platform with 6 degrees of freedom, *Proceedings of the institute of mechanical engineers*, pp. 371–386.
- Tsai, L. W. (2000). Solving the inverse dynamics of a gough-stewart manipulator by the principle of virtual work, *Journal of Mechanism Design* Vol. 122: 3–9.
- Zhu, Z., Li, J., Gan, Z. & Zhang, H. (2005). Kinematic and dynamic modelling for real-time control of tau parallel robot, *Journal of Mechanism and Machine Theory* Vol. 40(No. 9): 1051–1067.



## **Robotic Systems - Applications, Control and Programming**

Edited by Dr. Ashish Dutta

ISBN 978-953-307-941-7

Hard cover, 628 pages

**Publisher** InTech

**Published online** 03, February, 2012

**Published in print edition** February, 2012

This book brings together some of the latest research in robot applications, control, modeling, sensors and algorithms. Consisting of three main sections, the first section of the book has a focus on robotic surgery, rehabilitation, self-assembly, while the second section offers an insight into the area of control with discussions on exoskeleton control and robot learning among others. The third section is on vision and ultrasonic sensors which is followed by a series of chapters which include a focus on the programming of intelligent service robots and systems adaptations.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Georges Fried, Karim Djouani and Amir Fijany (2012). Kinematic and Inverse Dynamic Analysis of a C5 Joint Parallel Robot, *Robotic Systems - Applications, Control and Programming*, Dr. Ashish Dutta (Ed.), ISBN: 978-953-307-941-7, InTech, Available from: <http://www.intechopen.com/books/robotic-systems-applications-control-and-programming/kinematic-and-inverse-dynamic-analysis-of-a-c5-joint-parallel-robot>

# **INTECH**

open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821



© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen