A MODEL REDUCTION METHOD FOR THE CONTROL OF RIGID MECHANISMS

O. Brüls*, P. Duysinx†, and J.-C. Golinval*

* Département Aérospatiale, Mécanique et Matériaux (ASMA)
University of Liège
Chemin des Chevreuils 1, 4000 Liège, BELGIUM
e-mail: o.bruls@ulg.ac.be, jc.golinval@ulg.ac.be, web page: http://www.ulg.ac.be/ltas-vis

† Département Production, Mécanique et Thermodynamique (PROMETHE)
University of Liège
Chemin des Chevreuils 1, 4000 Liège, BELGIUM
e-mail: p.duysinx@ulg.ac.be, web page: http://www.ulg.ac.be/roboauto

Keywords: Model Reduction, Constraint Elimination, Interpolation, Parallel Mechanism.

Abstract. *This paper presents a reduction method to build closed-form dynamic equations for rigid multibody systems with a minimal kinematic description. Relying on an initial parameterization with absolute displacements and rotations, the method is able to tackle complex topologies with closed-loops in a systematic way and its extension to flexible multibody systems will be investigated in the future. Thus, it would be of great use in the framework of model-based control of mechanisms. The method is based on an interpolation strategy. The initial model is built and reduced for a number of selected points in the configuration space. Then, a piecewise polynomial model is adjusted to match the collected data. After the presentation of the reduction procedure and of the interpolation strategy, two applications of the reduction method are considered: a four-bar mechanism and a parallel kinematic machine-tool called "Orthoglide".*

1 INTRODUCTION

In the field of robotics, parallel kinematic mechanisms are the subject of a growing interest[1,2,3]: the motorized joints can be located on the base so that the moving mass is reduced, the stiffness is increased, and faster motions are possible. In some cases, the extended bandwidth of the actuators may excite the natural vibration modes of the system so that the flexible behaviour has to be taken into account. Anyway, even if the rigid-body assumption is valid, parallel mechanisms exhibit a complex kinematic and dynamic behaviour which requires powerful model-based control algorithms, such as the computed torque technique[4]. This technique is performed with an inverse dynamic model, which outputs the required joint actions given a set of joint values, rates and accelerations. This algebraic relationship is directly connected to the Ordinary Differential Equations (ODEs) of the underlying state-space model. Thus, an important problem is to implement this dynamic model in the control algorithm without violating the real-time constraints.

Modeling methods in multibody dynamics may be classified according to their generality and computational efficiency, which are usually conflicting with each other. On the one hand, efficient symbolic and/or recursive formulations, such as the Newton-Euler algorithm[4], are able to provide compact closed-form descriptions of open kinematic chains with a minimal set of coordinates. But their extension to more complex topologies is less attractive, since closed-loops introduce algebraic constraints between redundant variables. The system is then described by Differential Algebraic Equations (DAEs), and a numerical constraint elimination technique[5,6] is necessary to construct a reduced set of ODEs. Even if those formulations have been extended to mechanisms with flexible links[7], they are usually restricted to rigid body mechanisms. On the other hand, numerical Finite Element (FE) methods[8] are systematic and general in nature, and they are able to handle complex topologies and highly flexible components. However, the generated DAEs contain many redundant variables and the eventual constraint elimination involves expensive computations. Other methods lead to various compromises between efficiency and generality, such as the lumped approaches proposed by several authors[9,10] to deal with flexible mechanisms, which also rely on a rather large number of coordinates. Thus, none of these modeling methods meets the requirements for real-time control of parallel mechanisms.

In this work, a systematic and general methodology has been developed to provide closed-form ODEs for such mechanisms with complex topology. This paper presents the method for rigid mechanisms, and an extension to flexible multibody systems will be investigated in the future. The method relies on two ingredients: a numerical reduction procedure and an interpolation strategy. Starting from a redundant parameterization of the equations of motion, a constraint elimination procedure is developed to provide the reduced set of ODEs in any given configuration. Then, an interpolation is performed in order to get an explicit expression of the reduced model in the whole configuration space. In other words, the constraint elimination is performed offline once for all, so that the closed-form dynamic equations are almost directly available online. This method is similar to metamodeling techniques used in response surface optimization methods[11,12,13,14]. It may also be seen as an enhanced table look-up technique[15] adapted for parallel mechanisms, where the memory size needed to store the model is significantly reduced thanks to an efficient interpolation strategy.

This paper is organized as follows. Section 2 describes the minimal kinematic parameterization associated with the reduced model, as well as the redundant parameterization necessary to build systematically an initial model. Section 3 presents the local reduction procedure in a given configuration. The global interpolation strategy is developed in section 4. Two numerical applications are treated in section 5: a four-bar mechanism and a parallel kinematic machine tool, the Orthoglide[1]. The paper ends with a few conclusions and suggestions for future work.

2 PARAMETERIZATION OF THE EQUATIONS OF MOTION

The dynamic model of a rigid mechanism is directly connected to the parameterization of its motion. In this section, two parameterizations and the corresponding models are reviewed.

2.1 Minimal kinematic parameterization

The number of kinematic modes of a rigid mechanism is given by the Grübler formula. If we assume that the mechanism is fully actuated, this number s equals the number of actuated joints, and the s × 1 configuration vector $\boldsymbol{\theta}$ can be defined as the set of degrees of freedom conjugated with the corresponding generalized forces. For instance, for a motorized hinge the associated configuration variable is the angle between the connected links whereas for a linear actuator, it is the relative distance between the connected bodies.

The mechanism motion can be described with a set of Ordinary Differential Equations (ODEs) [4]:

$$\mathbf{H}(\boldsymbol{\theta})\,\ddot{\boldsymbol{\theta}} + \mathbf{h}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) - \mathbf{p}_{ext}(\boldsymbol{\theta}) = \boldsymbol{\tau} \tag{1}$$

where $\mathbf{H}$ is the $s \times s$ inertia tensor, $\mathbf{h}$ is the $s \times 1$ vector of centrifugal and Coriolis forces, $\mathbf{p}_{\mathrm{ext}}$ is the $s \times 1$ vector of external forces (e.g. the gravity forces) and $\boldsymbol{\tau}$ is the $s \times 1$ vector of the control actions. The centrifugal and Coriolis forces are quadratic in the velocities, thus the $s \times s \times s$ 3$^{\mathrm{rd}}$ order tensor $\mathbf{h}^*$ can be defined:

$$h_i = \sum_{j=1}^{s}\sum_{k=1}^{s} h^*_{ijk}(\boldsymbol{\theta})\,\dot{\theta}_j\,\dot{\theta}_k \tag{2}$$

This tensor could be computed analytically from the gradient of the inertia tensor $\mathbf{H}$. If we define the vector $\left[\dot{\boldsymbol{\theta}}\dot{\boldsymbol{\theta}}\right] = \left[\dot{\theta}_1^2 \quad \dot{\theta}_1\dot{\theta}_2 \quad \dots \quad \dot{\theta}_s^2\right]$, we get the matrix relationship:

$$\mathbf{h}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) = \mathbf{D}(\boldsymbol{\theta})\left[\dot{\boldsymbol{\theta}}\dot{\boldsymbol{\theta}}\right] \tag{3}$$

where $\mathbf{D}$ is the $s \times s(s+1)/2$ gyroscopic and centrifugal matrix directly connected with $\mathbf{h}^*$.

Our purpose is to build closed-form expressions for $\mathbf{H}(\boldsymbol{\theta}), \mathbf{D}(\boldsymbol{\theta})$ and $\mathbf{p}_{ext}(\boldsymbol{\theta})$, which are needed in equation (1) to estimate the required action $\boldsymbol{\tau}$ from the given values of $\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}$ and $\ddot{\boldsymbol{\theta}}$. For mechanisms with closed-loops, no direct formulation is available, and we have to start from a model based on a redundant kinematic description with constraint equations.

2.2 Redundant kinematic parameterization

Several parameterizations are well-established in rigid multibody dynamics. Since the method presented here will be later extended to flexible mechanisms, absolute coordinates have been chosen in order to be consistent with the Finite Element formulation[8]. However, for rigid mechanisms, the reduction could have been established from any formulation based on relative coordinates, leading to the same reduced model (1).

The motion of a rigid body is directly referred to an inertial frame, and this formalism is described in details by Géradin and Cardona[8]. The rigid body is represented by one master node at the center of inertia, and a set of slave nodes, for instance located at the attachment points of the joints. The motion of the master node is described with a vector of absolute displacements and a Cartesian rotation vector. The kinetic and gravity potential energy of the body can be formulated easily with respect to those coordinates. The absolute orientation of a slave node is directly identified to the orientation $\mathbf{\Psi}$ of the master, whereas its absolute position $\mathbf{x}_s$ satisfies:

$$\mathbf{x}_s - \mathbf{x}_m - \mathbf{R}(\mathbf{\Psi})\,\mathbf{X}_s = \mathbf{0} \tag{4}$$

where $\mathbf{x}_m$ is the master position, $\mathbf{R}$ is the rotation matrix and $\mathbf{X}_s$ is the position of the slave node in the material frame centered on the master node. If the slave displacements are considered as generalized coordinates, equation (4) can be regarded as a kinematic constraint. A kinematic joint connecting two rigid bodies is also modeled with an additional set of scleronomic constraint.

Adopting an updated Lagrangian point of view for the rotation parameters and applying the Lagrange multipliers method[8], the Hamilton principle leads to a system of DAEs of the form:

$$\begin{aligned} &\mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{g}_{iner}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{\Phi}_q^T\,\boldsymbol{\lambda} - \mathbf{g}_{ext} = \mathbf{C}\,\boldsymbol{\tau} \\ &\mathbf{\Phi}(\mathbf{q}) = \mathbf{0} \end{aligned} \tag{5}$$

where $\mathbf{q}$ is the $n \times 1$ vector of generalized coordinates and $\mathbf{\Phi}$ denotes the n-s scleronomic kinematic constraints. $\mathbf{M}$ is the $n \times n$ mass matrix, $\mathbf{g}_{iner}$ is the $n \times 1$ vector of complementary inertia forces, $\mathbf{\Phi}_q$ is the n-$s \times n$ matrix of constraint gradients, $\boldsymbol{\lambda}$ is the n-$s \times 1$ vector of Lagrange multipliers, $\mathbf{g}_{ext}$ is the $n \times 1$ vector of external forces and $\mathbf{C}$ is the $n \times s$ influence matrix of the control actions $\boldsymbol{\tau}$ on the generalized coordinates. As in the state-space model, the complementary inertia forces are quadratic in the velocities:

$$g_{iner\,i} = \sum_{j=1}^{n}\sum_{k=1}^{n} g^*_{ijk}(\mathbf{q})\,\dot{q}_j\,\dot{q}_k \tag{6}$$

where $\mathbf{g}^*$ is a $n \times n \times n$ third order tensor.

Rigid bodies contribute to this tensor through their rotation degrees of freedom. In particular, *for an updated configuration*, the complementary inertia forces of an isolated rigid body labeled "*e*" are:

$$\underbrace{\begin{bmatrix} g_{iner1} \\ g_{iner2} \\ g_{iner3} \end{bmatrix}^{(e)}}_{=\mathbf{g}_{iner}^{(e)}} = \underbrace{\begin{bmatrix} I_3 - I_2 & 0 & 0 \\ 0 & I_1 - I_3 & 0 \\ 0 & 0 & I_2 - I_1 \end{bmatrix}^{(e)}}_{=\mathbf{I}^{(e)}} \underbrace{\begin{bmatrix} \dot{\psi}_3\dot{\psi}_2 \\ \dot{\psi}_1\dot{\psi}_3 \\ \dot{\psi}_2\dot{\psi}_1 \end{bmatrix}^{(e)}}_{=\left[\dot{\mathbf{\Psi}}\dot{\mathbf{\Psi}}\right]^{(e)}} \tag{7}$$

where I_1, I_2 and I_3 are the principal moments of inertia. Those equations, formulated in the material principal axes, are closely related to the classical Euler equations. It is worth noticing that the matrix $\mathbf{I}^{(e)}$ does not depend on the configuration.

2.3 Outline of the reduction method

The initial model is the model associated with the redundant coordinates and the reduced model is the model associated with the minimal coordinates. The method to extract the reduced model from the initial model is summarized as follows:

$$\begin{Bmatrix}\boldsymbol{\theta}^1\\ \vdots \\ \boldsymbol{\theta}^r\end{Bmatrix} \underset{\text{Coordinate transform.}}{\Rightarrow} \begin{Bmatrix}\mathbf{q}^1\\ \vdots \\ \mathbf{q}^r\end{Bmatrix} \underset{\text{Initial Model}}{\Rightarrow} \begin{Bmatrix}\mathbf{M}^1,\mathbf{g}^1_{ext}\dots\\ \vdots \\ \mathbf{M}^r,\mathbf{g}^r_{ext}\dots\end{Bmatrix} \underset{\text{Reduction}}{\Rightarrow} \begin{Bmatrix}\mathbf{H}^1,\mathbf{D}^1,\mathbf{p}^1_{ext}\\ \vdots \\ \mathbf{H}^r,\mathbf{D}^r,\mathbf{p}^r_{ext}\end{Bmatrix} \underset{\text{Interpolation}}{\Rightarrow} \mathbf{H}(\boldsymbol{\theta}),\mathbf{D}(\boldsymbol{\theta}),\mathbf{p}_{ext}(\boldsymbol{\theta})$$

A set of r reference points are selected in the configuration space. For each of them, the redundant coordinates are computed (sections 3.1 to 3.3) and the initial model is built numerically. Then, the reduction method is achieved to define the reduced models locally around each reference configuration (section 3.4). Finally, a polynomial interpolation procedure generates the explicit model in the whole configuration space (section 4).

3 LOCAL REDUCTION IN A GIVEN CONFIGURATION

This section addresses the numerical determination of the reduced model (**H**, **D** and $\mathbf{p}_{ext}$) for any given value of $\boldsymbol{\theta}$. The proposed method is an adaptation of the constraint elimination method with coordinate partitioning[5,6]. Our starting point is an initial model of the mechanism and an initial assembled configuration described by the vector $\mathbf{q}^{init}$ satisfying the constraints. First, the characterization of the relationship between the coordinates $\mathbf{q}$ and $\boldsymbol{\theta}$ is studied.

3.1 Preliminary considerations

Since they have been defined as the actuator degrees of freedom, the configuration variables $\boldsymbol{\theta}$ are, in essence, relative coordinates. In order to establish their relation with the absolute coordinates $\mathbf{q}$, a mixed coordinates formulation is exploited[5], which means that both sets of coordinates are embedded in a unique and redundant parameterization. Each component θ_i is connected to the nodal coordinates $\mathbf{q}$ by an additional kinematic constraint:

$$\Phi_i^{(\theta)}(\mathbf{q},\theta_i)=0 \qquad i=1,\dots,s \tag{8}$$

In practice, this constraint comes from a generic formulation of the usual actuators, i.e. the motorized hinge and the linear actuator (see Géradin and Cardona[8]). For example, if we consider a hinge joint connecting two nodes n_1 and n_2, the orientation parameters of the frames attached to n_1 and n_2 can be extracted from the vector $\mathbf{q}$. Therefore, the relative orientation α between those frames is a function $\alpha(\mathbf{q})$ and a convenient choice for the constraint associated with the joint variable θ_i is:

$$\Phi_i^{(\theta)}(\mathbf{q},\theta_i) = \sin(\theta_i - \alpha(\mathbf{q})) = 0 \tag{9}$$

From here, the n-s constraints of the initial model are denoted $\mathbf{\Phi}^{(q)}(\mathbf{q})$, and we get a total of n constraints:

$$\mathbf{\Phi}(\mathbf{q},\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{\Phi}^{(q)}(\mathbf{q}) \\ \mathbf{\Phi}^{(\theta)}(\mathbf{q},\boldsymbol{\theta}) \end{bmatrix} = \mathbf{0} \tag{10}$$

This equation may be thought as an implicit definition for $\mathbf{q}(\boldsymbol{\theta})$. For a given configuration $\boldsymbol{\theta}^*$, we can thus ensure that a unique solution $\mathbf{q} = \mathbf{q}(\boldsymbol{\theta})$ exists in the neighbourhood of $\boldsymbol{\theta}^*$ if the Jacobian matrix of the constraints $\mathbf{\Phi}_q(\mathbf{q}^*,\boldsymbol{\theta}^*)$ is not singular.

The points where $\mathbf{\Phi}_q$ is singular are either limit points or bifurcation points[8], where the kinematic modes change qualitatively. Then, the kinematic description given by the configuration variables $\boldsymbol{\theta}$ breaks down: if the actuated joints were locked, variations of the generalized coordinates $\mathbf{q}$ would be allowed within the null space of $\mathbf{\Phi}_q$. A new set of configuration variables is necessary to analyse the behaviour of the mechanism at those points. In many control applications, those configurations are carefully avoided and in the following, we consider them as bounds for the motion of the mechanism.

We would expect that a univocal relationship $\mathbf{q}(\boldsymbol{\theta})$ would characterize the whole configuration space. Unfortunately, for a mechanism with s kinematic modes, several assembled solutions $\mathbf{q}$ can exist for a given vector $\boldsymbol{\theta}$ [16]. So, we have to keep in mind the local validity of the relationship $\mathbf{q}(\boldsymbol{\theta})$, which will be illustrated with the four-bar mechanism in section 5.1.

In equation (5), the constraints of the initial model are supposed to be independent, otherwise, we would get more than n constraints in equation (10). If this assumption is not satisfied but the rank of $\mathbf{\Phi}_q$ is still maximal, the problem can be tackled by the extraction of a minimal set of constraints with a regular gradient.

3.2 Computation of the Jacobian $\mathbf{q}_\theta$

The Jacobian of the coordinate transformation plays a key role in the reduction, as will be seen below. It is computed by implicit differentiation of (10):

$$\mathbf{q}_\theta = \frac{\partial \mathbf{q}}{\partial \boldsymbol{\theta}} = -\mathbf{\Phi}_q^{-1}\,\mathbf{\Phi}_\theta \tag{11}$$

which makes sense away from the singularities.

3.3 Computation of q

First, the generalized coordinates $\mathbf{q}$ associated with the given $\boldsymbol{\theta}$ have to be computed in order to satisfy the constraint equation (10). This equation can be thought as a set of *n* nonlinear equations with *n* unknowns, which can be solved with a standard Newton-Raphson procedure:

$$\begin{aligned}&\text{Prediction: } \mathbf{q}^{(0)} = \mathbf{q}^{init} + \mathbf{q}_\theta\,(\boldsymbol{\theta} - \boldsymbol{\theta}^{init})\\ &\text{Correction: } \mathbf{q}^{(k+1)} = \mathbf{q}^{(k)} - \boldsymbol{\Phi}_q^{-1}\,\boldsymbol{\Phi}^{(k)}\end{aligned} \tag{12}$$

If $\boldsymbol{\theta}$ is far from $\boldsymbol{\theta}^{init}$, a few intermediate configurations can be considered to improve the convergence.

3.4 Reduction

The first equation in (5) should be rewritten:

$$\mathbf{M}\,\ddot{\mathbf{q}} + \mathbf{g}_{iner} + \boldsymbol{\Phi}_q^{(q)T}\,\boldsymbol{\lambda} - \mathbf{g}_{ext} = \mathbf{C}\,\boldsymbol{\tau} \tag{13}$$

For a minimal kinematic description, the Jacobian $\mathbf{q}_\theta$ is in the null space of the constraint gradient $\boldsymbol{\Phi}_q^{(q)}$ [5]:

$$\left.\begin{aligned}\boldsymbol{\Phi}_q = \begin{bmatrix}\boldsymbol{\Phi}_q^{(q)}\\ \boldsymbol{\Phi}_q^{(\theta)}\end{bmatrix} \quad &\Rightarrow \quad \boldsymbol{\Phi}_q^{(q)} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\boldsymbol{\Phi}_q\\ \boldsymbol{\Phi}_\theta = \begin{bmatrix}\mathbf{0}\\ \boldsymbol{\Phi}_\theta^{(\theta)}\end{bmatrix} \quad &\underset{(11)}{\Rightarrow} \quad \mathbf{q}_\theta = -\boldsymbol{\Phi}_q^{-1}\begin{bmatrix}\mathbf{0}\\ \boldsymbol{\Phi}_\theta^{(\theta)}\end{bmatrix}\end{aligned}\right\} \Rightarrow \quad \boldsymbol{\Phi}_q^{(q)}\,\mathbf{q}_\theta = \mathbf{0}$$

In other words, the columns of the Jacobian $\partial \mathbf{q}/\partial \theta_i$ belong to the tangent space of the constraint manifold $\boldsymbol{\Phi}^{(q)} = \mathbf{0}$. This is consistent with the well-known principle that a projection of the dynamic equations (13) onto the configuration space eliminates the Lagrange multipliers term:

$$\mathbf{q}_\theta^T\,\mathbf{M}\,\ddot{\mathbf{q}} + \mathbf{q}_\theta^T\,\mathbf{g}_{iner} - \mathbf{q}_\theta^T\,\mathbf{g}_{ext} = \mathbf{q}_\theta^T\,\mathbf{C}\,\boldsymbol{\tau} \tag{14}$$

Comparing equation (14) with (1) allows the identification of the reduced model. First, we observe:

$$\mathbf{q}_\theta^T\,\mathbf{C} = \mathbf{I} \tag{15}$$

where $\boldsymbol{I}$ is the identity matrix; this relation results from the definition of $\boldsymbol{\theta}$ as conjugated to $\boldsymbol{\tau}$. We also have:

$$\boxed{\mathbf{p}_{ext} = \mathbf{q}_\theta^T\,\mathbf{g}_{ext}} \tag{16}$$

To estimate $\mathbf{H}$ and $\mathbf{h}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ should be eliminated:

$$\begin{aligned}\dot{\mathbf{q}} &= \mathbf{q}_\theta\,\dot{\boldsymbol{\theta}}\\ \ddot{\mathbf{q}} &= \mathbf{q}_\theta\,\ddot{\boldsymbol{\theta}} + \dot{\mathbf{q}}_\theta\,\dot{\boldsymbol{\theta}}\end{aligned} \tag{17}$$

And we get:

$$\boxed{\mathbf{H} = \mathbf{q}_\theta^T\,\mathbf{M}\,\mathbf{q}_\theta} \tag{18}$$

$$\mathbf{h} = \mathbf{q}_\theta^T\,\mathbf{g}_{iner} + \mathbf{q}_\theta^T\,\mathbf{M}\,\dot{\mathbf{q}}_\theta\,\dot{\boldsymbol{\theta}} = \mathbf{D}\left[\dot{\boldsymbol{\theta}}\dot{\boldsymbol{\theta}}\right] \tag{19}$$

The identification of the gyroscopic and centrifugal matrix $\mathbf{D}$ is not straightforward. First, let us focus on the term $\mathbf{q}_\theta^T\,\mathbf{g}_{iner}$. In equation (7), the elementary formulation of $\mathbf{g}_{iner}$ involves the rates of the Cartesian rotation vector, which are related with $\dot{\boldsymbol{\theta}}$ by equation (17):

$$\dot{\boldsymbol{\Psi}}^{(e)} = \mathbf{q}_\theta^{(e)}\,\dot{\boldsymbol{\theta}} \tag{20}$$

This relationship can be partitioned:

$$\begin{bmatrix}\dot{\psi}_1\\ \dot{\psi}_2\\ \dot{\psi}_3\end{bmatrix}^{(e)} = \begin{bmatrix}\mathbf{q}_{\theta 1}^{(e)T}\\ \mathbf{q}_{\theta 2}^{(e)T}\\ \mathbf{q}_{\theta 3}^{(e)T}\end{bmatrix}\dot{\boldsymbol{\theta}}$$

where $\mathbf{q}_{\theta i}^{(e)T}$ are row vectors. Thus we get:

$$\begin{aligned}\dot{\psi}_2\,\dot{\psi}_3 &= \left(\mathbf{q}_{\theta 2}^{(e)T}\,\dot{\boldsymbol{\theta}}\right)\left(\mathbf{q}_{\theta 3}^{(e)T}\,\dot{\boldsymbol{\theta}}\right)\\ &= \dot{\boldsymbol{\theta}}^{\,T}\left(\mathbf{q}_{\theta 2}^{(e)}\,\mathbf{q}_{\theta 3}^{(e)T}\right)\dot{\boldsymbol{\theta}}\end{aligned} \tag{21}$$

which can be reorganized:

$$\dot{\psi}_2\,\dot{\psi}_3 = L_1^{(e)T}\left[\dot{\boldsymbol{\theta}}\dot{\boldsymbol{\theta}}\right] \tag{22}$$

where $L_1^{(e)T}$ is a row vector. If this development is reproduced for $\dot{\psi}_1\dot{\psi}_3$ and $\dot{\psi}_1\dot{\psi}_2$, we get:

$$\left[\dot{\boldsymbol{\Psi}}\dot{\boldsymbol{\Psi}}\right]^{(e)} = \begin{bmatrix}\dot{\psi}_2\,\dot{\psi}_3\\ \dot{\psi}_1\,\dot{\psi}_3\\ \dot{\psi}_1\,\dot{\psi}_2\end{bmatrix}^{(e)} = \underbrace{\begin{bmatrix}L_1^{(e)T}\\ L_2^{(e)T}\\ L_3^{(e)T}\end{bmatrix}}_{=\mathbf{L}^{(e)}}\left[\dot{\boldsymbol{\theta}}\dot{\boldsymbol{\theta}}\right] \tag{23}$$

so that equation (7) becomes:

$$\mathbf{g}_{iner}^{(e)} = \underbrace{\mathbf{I}^{(e)}\,\mathbf{L}^{(e)}}_{=\mathbf{A}^{(e)}}\left[\dot{\boldsymbol{\theta}}\dot{\boldsymbol{\theta}}\right] \tag{24}$$

The assembly of the contributions of all the rigid bodies of the mechanism yields:

$$\mathbf{g}_{iner} = \mathbf{A}\left[\dot{\boldsymbol{\theta}}\dot{\boldsymbol{\theta}}\right] \tag{25}$$

Let us now consider the second term in equation (19): $\mathbf{q}_\theta^T\,\mathbf{M}\,\dot{\mathbf{q}}_\theta\,\dot{\boldsymbol{\theta}}$. We have

$$\left(\dot{\mathbf{q}}_\theta\,\dot{\boldsymbol{\theta}}\right)_i = \sum_{j=1}^{s}\left(\dot{\mathbf{q}}_\theta\right)_{ij}\dot{\theta}_j = \sum_{k=1}^{s}\sum_{j=1}^{s}\frac{\partial\left(\mathbf{q}_\theta\right)_{ij}}{\partial\theta_k}\,\dot{\theta}_j\,\dot{\theta}_k \qquad i=1,\ldots,n \tag{26}$$

where the derivatives of the Jacobian are computed numerically using a finite difference method. This equation can be restated:

$$\dot{\mathbf{q}}_\theta\,\dot{\boldsymbol{\theta}} = \mathbf{B}\left[\dot{\boldsymbol{\theta}}\dot{\boldsymbol{\theta}}\right] \tag{27}$$

Finally, from equations (19), (25) and (27), the gyroscopic and centrifugal matrix $\mathbf{D}$ is obtained in the form:

$$\boxed{\mathbf{D} = \mathbf{q}_\theta^T\,(\mathbf{A} + \mathbf{M}\,\mathbf{B})} \tag{28}$$

4 GLOBAL INTERPOLATION IN THE CONFIGURATION SPACE

In the previous section, we have developed an algorithm which produces the reduced model $\mathbf{H}$, $\mathbf{D}$, $\mathbf{p}_{\text{ext}}$ from a given configuration vector $\boldsymbol{\theta}$. All coefficients of the matrices $\mathbf{H}$ and $\mathbf{D}$ and of the vector $\mathbf{p}_{\text{ext}}$ may be collected in a single $t \times 1$ output vector $\mathbf{f}$ ($t = s(s+1)/2 + s^2\,(s+1)/2 + s$), so that the algorithm can be considered as a black box function:

$$\mathbf{f} = \mathbf{f}(\boldsymbol{\theta}) \tag{29}$$

As seen before, the evaluation of this function involves a large amount of computation, and a simplified analytical model would be more convenient for real-time applications. From the full black box model, we want to define a surrogate model, or metamodel[11] (model of the model) :

$$\hat{\mathbf{f}} = \hat{\mathbf{f}}(\boldsymbol{\theta}) \tag{30}$$

Obviously, there is a trade-off between the accuracy of the approximation and the complexity of the metamodel. But which method should be considered for the construction of $\hat{\mathbf{f}}$? Many researches have been presented in the literature regarding this problem, especially in the fields of identification and optimization. In identification, interpolation methods are widely used to fit experimental data, which are affected by noise. In optimization, response surface methods[12] take advantage of an interpolated model to reduce the number of runs of the full model. Our problem is closer to these latter "metamodeling" applications, where $\mathbf{f}$ is defined by deterministic computer analysis codes.

The metamodel is elaborated according to the following generic procedure:

1. select a set of inputs $\boldsymbol{\theta}^1,\ldots,\ \boldsymbol{\theta}^r$;
2. run the full model for each input and get back $\mathbf{f}^1,\ldots,\ \mathbf{f}^r$;
3. define a generic approximated function $\hat{\mathbf{f}}$;
4. fit the parameters of $\hat{\mathbf{f}}$ on the data obtained in step 3.

Step 1 is also referred to as the experimental design problem[17]. Step 2 was the subject of section 3. In step 3, we restrict our study to linear models:

$$\hat{f}_i(\boldsymbol{\theta}) = \sum_j w_{ij}\ u_j(\boldsymbol{\theta}) \tag{31}$$

where u_j are fixed basis functions and w_{ij} are the free weights allowing to fit the model. In that case, step 4 is a linear regression problem which can be solved in the least-square sense using standard methods. In the following, the choice of the basis functions (step 3) and the experimental design (step 1) are addressed and a few remarks are given about the implementation of the method.

4.1 Choice of the basis functions

The simplest basis functions u_j are low order polynoms, which are well suited for local interpolation. A piecewise interpolation strategy allows to define the approximated function $\hat{\mathbf{f}}$ in the whole configuration space, but the discontinuous transitions in the model description can then be a source of trouble for some applications.

Instead of polynoms, one can use radial basis functions[13,14] $u_j = u\left(\left\|\boldsymbol{\theta} - \boldsymbol{\theta}^j\right\|\right)$. As there are as many basis functions as given configurations, exactness can be achieved at the data points. For large data set, the estimation of $\hat{\mathbf{f}}$ becomes very costly and the number of basis function has to be reduced. Anyway, the choice of adequate radial functions remains complicated and sometimes requires nonlinear optimization strategies.

Other enhanced strategies are extensively developed in the literature, but this work is a first step in the construction of metamodels in multibody dynamics. Therefore, we retain the simple, efficient and powerful piecewise polynomial fitting strategy, with second order polynoms (linear polynoms would require a finer discretization of the workspace which would lead to a prohibitive increase in the memory needed to store the reduced model). This method is expected to be efficient for a sufficiently smooth function $\mathbf{f}$.

4.2 Experimental design

Figure 1 illustrates classical experimental designs[17]: the full factorial design, the fractional factorial design and the Central Composite Design (CCD). More sophisticated methods, such as the Taguchi approach, seem unnecessarily complicated for our problem.

We have retained the CCD method, which provides sufficient information to estimate the quadratic effects required for a second order polynomial fitting. The configuration space is discretized into a set of non-overlapping boxes whose size may be adapted according to the smoothness of **f** (for instance, near singularities, smaller boxes are required to capture the stiff behaviour of the system). Small discontinuous transitions can thus arise in the interpolated function at the sides of the boxes. In CCD, the position of the "star points" on the axis has to be selected; for the sake of simplicity we put them at the intersection of the axis with the box sides.

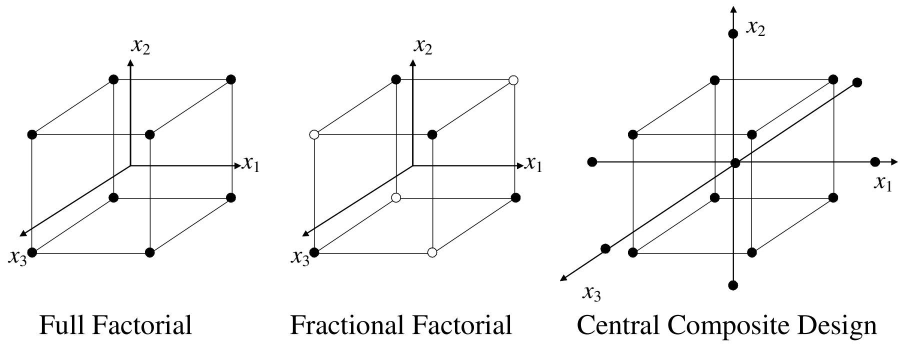

Figure 1: Experimental designs, the 3D case.

4.3 Implementation

Our developments are implemented in the OOFELIE[18]environment (Object Oriented Finite Elements Led by Interactive Executor). The function **f** is computed according to the algorithm presented in section 3.

The set of boxes is recorded as a tree structure: the root is defined as a box which contains the whole configuration space. A bisection criterion decides if the box is small enough or if a bisection is needed in a chosen direction. The result of the bisection is the definition of two child boxes. If a generated box is outside the configuration space, it is discarded, otherwise the recursive definition is pursued until the leaf boxes of the tree. The leaf boxes are small enough according to the bisection criterion which guarantees the relevance of a local polynomial approximation.

The topology of the tree is thus specified in a very flexible way by the bisection criterion. In the examples presented below, the sizes of the boxes are compared with maximal dimensions specified by the user, leading to a uniform discretization of the configuration space. A more advanced criterion can rely on an error analysis of the local approximation, which will be considered in future developments.

For each leaf box, the CCD is applied to define a set of configurations where the full model **f** is computed. The couples of data ($\boldsymbol{\theta}^p - \mathbf{f}^p$) are stored as they are generated, so that they can be reused for neighbouring boxes. A quadratic polynom is fitted on those data according to a standard linear regression algorithm.

To evaluate the approximated reduced model $\hat{\mathbf{f}}$ in a given configuration, we have to find the right box with the right polynomial parameters. This problem is solved with an efficient binary search algorithm which takes advantage of the tree topology of the recorded data. The total number of floating point operations n_{fp} involved in the evaluation of $\hat{\mathbf{f}}$ can be estimated by a detailed inspection of the algorithm. If we consider a rectangular configuration space discretized into 2^x intervals along each dimension, which leads to 2^{xs} boxes, we get:

$$n_{fp} = \underbrace{x\,s}_{\substack{\text{binary}\\ \text{search}}} + \underbrace{s + \frac{s\,(s+1)}{2}}_{\mathbf{u}(\boldsymbol{\theta})} + t\,\underbrace{(s^2+3s)}_{\hat{f}_i(\boldsymbol{\theta})} \quad (32)$$

5 APPLICATIONS

Two applications are presented in this section: a four-bar mechanism (1D configuration space) and an Orthoglide (3D configuration space).

5.1 The four-bar mechanism

The figure below illustrates a four-bar mechanism with large configuration changes.

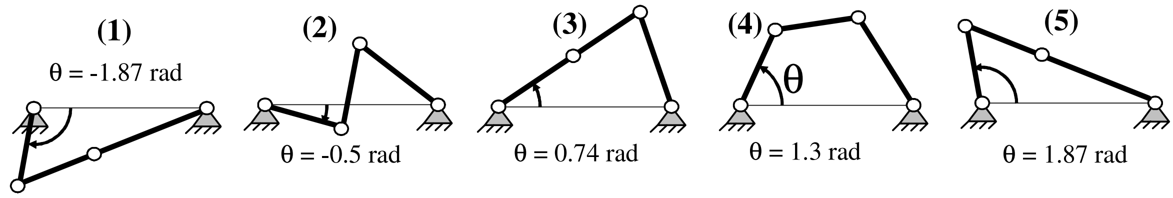

Figure 2: The four-bar mechanism.

The singular configurations (1) and (5) define the limits of our study in the configuration space. The implicit function theorem can not be applied at those points, and no univocal expression $\mathbf{q}(\theta)$ is available in their neighbourhood. From those configurations, two motions are possible for the same θ, whether the aligned links bend upward or downward. If another articulation had been selected for the actuation, the singularity would disappear at those specific configurations. These observations illustrate the discussion in section 3.1 about the possible solutions of the direct kinematic problem.

The full model is elaborated and the reduced model is built in the interval [-1.75,1.75] rad of the configuration space which is subdivided into 16 equal size boxes. The gravity forces are taken into account. According to equation (32), 18 floating point operations are required to obtain the reduced model in a given configuration.

The results are presented in Figure 3. Close to the singularities, the Jacobian defined in equation (11) grows to infinity and vertical asymptotes can be guessed in the results. The gyroscopic and centrifugal matrix is related to the gradient of the inertia tensor, which appears clearly in these results. As expected, the accuracy is remarkable if the function is smooth (relative error below 1%), but near the singularities the equations become stiff and the reliability of the interpolation decreases. Small discontinuous transitions are observed at the borders of each box, especially in the neighbourhood of the singularities. Using smaller boxes would improve the accuracy at the price of a global increase in the computational load to build the model and in the memory size to store it. An adaptive strategy of the box sizes could be implemented in order to achieve a better trade-off.

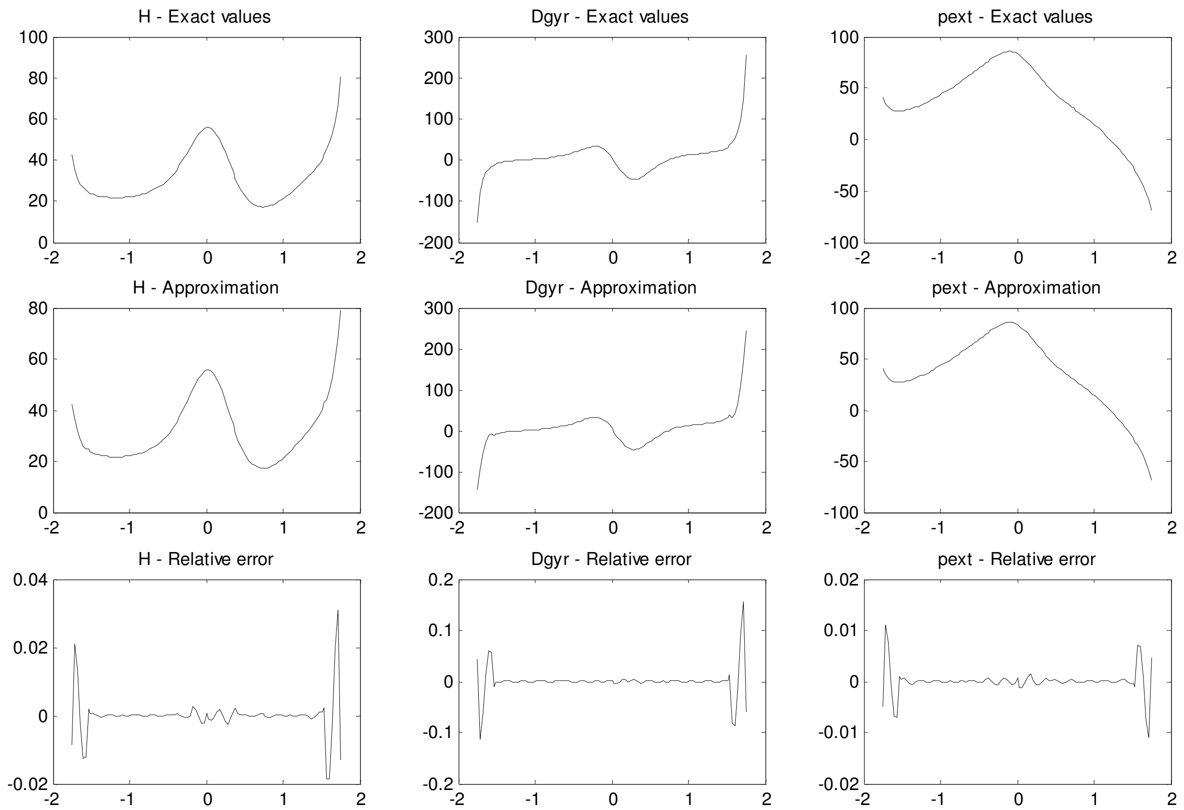

Figure 3: Four-bar mechanism properties (MKS units) with respect to the configuration angle θ (radians). 1st line: full model, 2nd line: reduced model, 3rd line: relative error between both models.

5.2 The Orthoglide

We now consider a more elaborated mechanism with an actual 3 dimensional motion: the Orthoglide (Figure 4). This machine tool with a parallel architecture has been designed at the IRCCyN research center in Nantes[1] in order to exhibit an isotropic behaviour in the center of its workspace. In the reference configuration, the 3 links are parallel to the inertial frame axes. The motion of each actuator is limited to the interval [-0.26,0.06] (all data are specified in MKS units) and the corresponding workspace does not contain any singular configuration. According to equation (32), 504 floating point operations are required to obtain the reduced model in a given configuration.

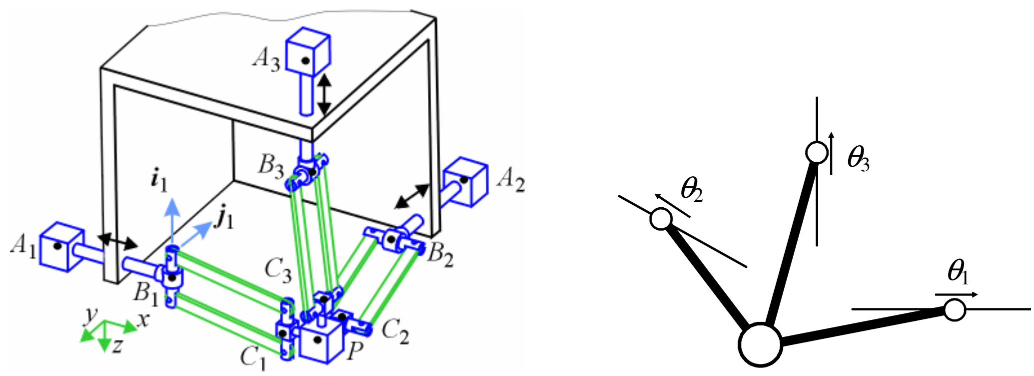

Figure 4: The Orthoglide. Complete topological description / Representation of the dynamic model.

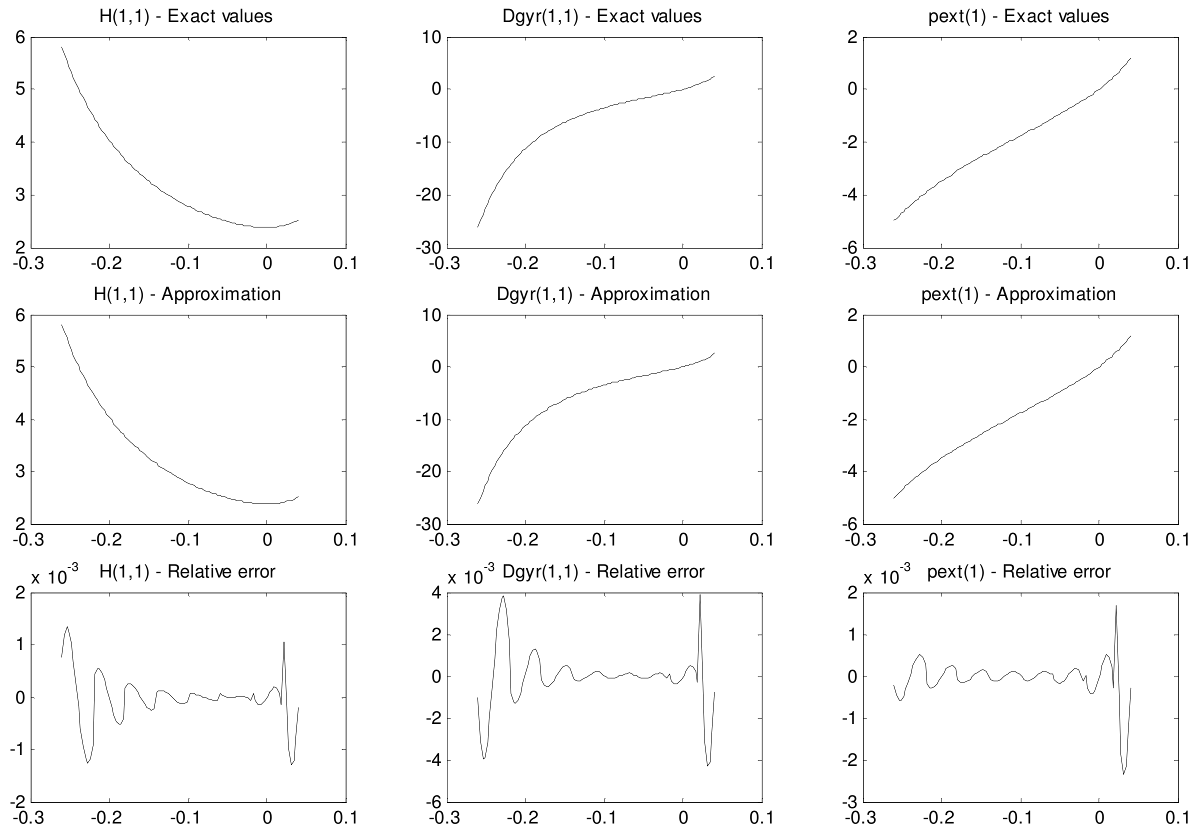

Figure 5: Evolution of the model parameters of the Orthoglide (MKS units) along a diagonal line which crosses the configuration space from [-0.26 –0.26 –0.26] to [0.06 0.06 0.06].

The initial model contains 27 variables: 15 generalized coordinates and 12 Lagrange multipliers. Limited to a box [-0.26,0.06] × [-0.26,0.06] × [-0.26,0.06], the configuration space is bisected into $8^3 = 512$ sub-boxes so that a set of 3000 input configurations is considered according to the CCD.

The results are presented in Figure 5. The interpolation appears to be very accurate: the relative errors are below 0.5% even at the border of the workspace. These excellent results may be attributed to the smooth behaviour of the system, since we keep away from the mechanism singularities.

6 CONCLUSIONS

This paper presents a method to build closed-form dynamic equations of rigid multibody systems with a minimal kinematic description. Starting from a redundant parameterization of the motion, it is able to treat complex topologies with closed-loops in a systematic way. Thus, the method would be of great use for model-based control of rigid mechanisms.

The model extraction follows a two level procedure:

- the local reduction of the dependent coordinate model in a given configuration according to a constraint elimination technique, which assumes a coordinate partitioning and is possible only for non-singular configurations of the mechanism;
- the global interpolation of the reduced model in the configuration space using a piecewise second-order polynomial approximation.

The method has been tested with two applications: a four-bar mechanism and a parallel kinematic machine-tool, the Orthoglide. It provides efficiently an accurate reduced model away from the singularities. Near the singularities, the stiff behaviour of the system is difficult to capture by interpolation. This is not really problematic for most control applications where singular configurations are carefully avoided. The clear physical interpretation of the coefficients of the reduced model gives new insights into the dynamics of the mechanism, which is valuable for the designer of the control system.

These results are thus stimulating for further investigations. More elaborated interpolation strategies could be considered, such as radial basis functions which have better continuity properties than piecewise polynoms. An automatic adaptation of the configuration space discretization would improve the efficiency of the method and allow the user to specify at the highest level the required accuracy of the interpolation. The extension to flexible multibody systems, a challenging project, is possible if a linear elastic behaviour is assumed.

ACKNOWLEDGEMENTS

M. Brüls is supported by a grant from the Belgian National Fund for Scientific Research (FNRS) which is gratefully acknowledged. This work also presents research results of the Belgian programme on Inter-University Poles of Attraction initiated by the Belgian state, Prime Minister's office, Science Policy Programme (AMS IUAP 5/06). The scientific responsibility is assumed by its authors. The authors thank Prof. P. Wenger (IRCCyN) for making the Orthoglide data available.

REFERENCES

[1] P. Wenger and D. Chablat, "Kinematic Analysis of a New Parallel Machine Tool: the Orthoglide", *Proc. of the 7th International Symposium on Advances in Robot Kinematics*, Slovenia, 2000.

[2] M. Honegger, A. Codourey, and E. Burdet, "Adaptive Control of the Hexaglide, a 6 dof Parallel Manipulator", *Proc. of the IEEE Int. Conf. on Robotics and Automation ICRA'97*, Albuquerque NM, USA, 1997.

[3] F. Caccavale, B. Siciliano, and L. Villani, "The Tricept Robot: Dynamics and Impedance Control", *IEEE/ASME Transactions on Mechatronics* **8**(2), 2003, 263–268.

[4] H. Asada and J.J.E. Slotine, *Robot Analysis and Control*, John Wiley & Sons, 1986.

[5] J. García de Jalón and E. Bayo, *Kinematic and Dynamic Simulation of Multibody Systems – The Real-Time Challenge*, Springer Verlag, 1994.

[6] R.A. Wehage and E.J. Haug, "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems", *J. of Mechanical Design*, **104**, 1982, 247-255.

[7] W.J. Book, "Recursive Lagrangian Dynamics of Flexible Manipulator Arms", *Int. J. of Robotics Research* **3**(3), 1984, 87-101.

[8] M. Géradin and A.Cardona, *Flexible Multibody Dynamics : A Finite Element Approach*, John Wiley & Sons, 2001.

[9] R.L. Huston, "Computer Methods in Flexible Multibody Dynamics", *Int. J. for Num. Meth. in Engrg.* **32**, 1991, 1657–1668.

[10] E. Wittbrodt and S. Wojciech, "Application of Rigid Finite Element Method to Dynamic Analysis of Spatial Systems", *J. of Guidance, Control, and Dynamics* **18**(4), 1995, 891–898.

[11] J.P.C.Kleijnen, *Statistical tools for simulation practitioners*, Marcel Dekker, NY, 1987.

[12] R.H. Myers, D.C. Montgomery, *Response Surface Methodology*, Wiley Series in Probability and Statistics, 1995.

[13] T.W. Simpson, J. Peplinski, P.N. Koch, and J.K. Allen, "On the Use of Statistics in Design and the Implications for Deterministic Computer Experiments", *Design Theory and Methodology, ASME Design Engineering Technical Conferences,* Sacramento, 1997.

[14] J. Tu, "Cross-Validated Multivariate MetaModeling Methods for Physics-Based Computer Simulations", *Proc. of the 21st IMAC International Modal Analysis Conference,* Florida, 2003.

[15] M.H.Raibert, "Analytical Equations Vs. Table Look-up for Manipulation: a Unifying Concept", *Proc. of the IEEE Conf. Decision and Control*, New Orleans, 1977.

[16] E. Dombre and W. Khalil, *Modélisation et Commande des Robots*, Hermes, 1988.

[17] D.C. Montgomery, *Design and Analysis of Experiments*, John Wiley & Sons, 1997.

[18] A. Cardona, I. Klapka, and M. Géradin, "Design of a New Finite Element Programming Environment", *Engineering Computations*, **11**, 1994, 365-381.