Computación y Sistemas Vol. 1 No. 3 pp. 145-160
© 1998, CIC - IPN. *ISSN 1405-5546* Impreso en México

DAE Methods in Constrained Robotics System Simulation

Ramon Costa-Castelló **Robert Griñó** **Luis Basañez**

Institut de Cibernètica (UPC)
Av. Diagonal 647, 2a planta
08028-Barcelona, SPAIN
{costa,grino,basanez}@ic.upc.es

Article received on September, 1997; accepted on January 30, 1998.

Abstract

In this paper a DAE formulation is used to model the behaviour of constrained robotic systems. This formulation allows to specify in an easy and clear way the constrained behaviour of a robotic system.

In order to better understand DAE systems some key concepts in analytic and numerical solution of DAE are presented. Once DAE properties and its aplication to Constrained Robotics Systems have been studied, a complete characterization of singularities appearing in the model are presented for a class of constrained robotic systems.

Finally, the work is completed with a set of simulations in order to validate numerically the theoretical developments.

Keywords: Constrained Robotics, Singularities, Dynamics, Kinematics, Simulation

1 Introduction

Constrained robotic systems are those in which the movement of the end effector or of any point of the kinematic chain is restricted to holonomic or non-holonomic constraints. These systems are an open research topic in modeling [Unseren & Koivo, 1989; Li *et al.*, 1991; Harris, 1986] simulation [Murphy *et al.*, 1990; McMillan *et al.*, 1994; McMillan *et al.*, 1994] and control [Paljug *et al.*, 1994; Nakamura *et al.*, 1989; Schneider & Cannon, 1989; Laroussi *et al.*, 1988] areas. This work deals with modeling and simulation through a differential–algebraic approach.

Simulation of constrained robotics systems is a current research topic in both applied mathematics and mechanical engineering fields. One of the most used tools in this context is the Lagrange multipliers method which allows a quite simple formulation but leads to complex resolution with numerical problems [Ellis & Ricker, 1994]. Other methods based on the elimination of some differential equations have simpler implementation but they do not seem very rigorous [Murphy *et al.*, 1990]. This last approach and many others like the Reduction Transformation approach [McClamroch, 1986] manipulate the initial differential–algebraic models to obtain purely differential models. As will be shown, the obtained models are of such complexity they hide underlying physical laws, make the system analysis difficult and generate numerical problems due to the introduction of constraints into the differential model. These approaches are usually formulated over Configuration Space ($\mathcal{C}$) following the classical Lagrange formulation, but recently Lilly [Lilly, 1993] has developed a method based on the Operational Space which has proved to be very efficient and can be implemented on parallel machines to work in real–time [McMillan *et al.*, 1994]. This method is used for the simulation of multi–robot systems and it is also applicable when one or several arms are in a singular configuration [Murphy *et al.*, 1990]. This point is of great interest since other methods cannot deal with this situation. A drawback of the method is the need of complex algorithms to simulate system behavior.

In this work a formulation based on the differential–algebraic system theory is presented [McClamroch, 1986]. This formulation differs from the classical Lagrange method in the interpretation and resolution methodologies, although the initial equation set is exactly the same in both cases. In the Differential–Algebraic Equation (DAE) formulation the equation set is considered as a differential equation over the manifold defined by the constrains. This kind of systems are not directly integrable by classical Ordinary Differential Equation (ODE) solvers [Harier *et al.*, 1991], although they can be integrated by Differential–Algebraic Equation (DAE) solving methods [Brenan *et al.*, 1989]. In this line DAE solvers have been used to simulate multibody systems [Fuhrer & Leimkuhler, 1991; Simeon *et al.*, 1994] and the DAE formulation leads to a simple modeling methodology in contrast with other methods which have their principal drawback in their complexity.

The use of a general purpose tool like the DAE systems is of great interest due to the fact that the that same theory can be used to understand, analyze and control the system behavior [Sira, 1992; Kumar & Daoutidis, 1995; McClamroch, 1990; Krishnan & McClamroch, 1993; Yim, 1993]. DAE solvers are an open research topic, and the parallel and real–time algorithms are some of their most interesting aspects. These solvers will allow simulation of DAE systems in real–time environments without the need of further analysis or model modification.

The simplicity of this approach allows it to be used in robotic cell simulators where every body has its own model. When interaction between different bodies occurs, only the introduction of a new set of equations representing the interaction between the involved bodies is needed [Yen, 1995]. Other methods could not handle this problem in such a simple way.

In the next sections the on modeling and simulation of constrained robotics systems following a DAE approach is presented. First of all some basic theory on DAE systems is introduced, then some topics dealing with robust numerical formulation are presented, and finally the obtained results are used to formulate models for a class of constrained robotics systems. These models are analyzed and their singularities are characterized. In addition to the theoretical results, the simulation of some constrained robotics systems is presented at the end of the paper.

2 DAE Basic Theory

Definition 1 A DAE system is a set of differential equations which can be expressed in general form as:

$$\gamma\ (t, \dot{\mathbf{x}}, \mathbf{x}, \mathbf{u})\ =\ \mathbf{0} \tag{1}$$

where $\gamma : \Re^{1+2n+m} \rightarrow \Re^{n}$, $\frac{\partial\gamma}{\partial\dot{\mathbf{x}}}$ is singular (i.e. Rank $\{\frac{\partial\gamma}{\partial\dot{\mathbf{x}}}\} < n$), $\mathbf{x} \in \Re^{n}$, and $u \in \Re^{m}$. $\mathbf{u}$ is the input to the system.

Remark 1 If $\frac{\partial\gamma}{\partial\dot{\mathbf{x}}}$ is nonsingular, the implicit differential equation (1) can be formally converted into an ordinary differential equation.

There exists a theory for linear DAE systems [Brenan *et al.*, 1989; Dai, 1989], but that is not the case for general nonlinear systems. The linear DAE theory, based on the pencil[1] analysis, is of great interest in numerical analysis of nonlinear DAE system due to the fact that they are locally linearized during the numerical integration process.

Present knowledge of nonlinear DAE system is limited to some morphologies, the most typical of which are presented in Table 1.

2.1 DAE Index

Definition 2 The differential index of a DAE system is the minimum number of times that all or part of the implicit differential equation (1) must be differented with respect to t in order to determine $\dot{\mathbf{x}}$ as a continuous function $\mathbf{\Psi}$ of $t, \mathbf{x}, \mathbf{u}$.

Definition 3 $\dot{\mathbf{x}} = \mathbf{\Psi}(t, \mathbf{x}, \mathbf{u})$ is called the underlying ODE of the DAE.

Remark 2 In nonlinear systems, the differential index and the underlying ODEs of a nonlinear DAE system are local properties.

Table 2 shows the relation between the differential index and the morphology of some nonlinear semi–explicit DAE systems. In order to determine the index of a DAE, this is differented with respect to t until the underlying ODE is obtained. During this process a matrix needs to be inverted to make the ODE explicit.

[1] For a system in the form $\mathbf{A}(t)\,\dot{\mathbf{x}}(t) + \mathbf{B}(t)\,\mathbf{x}(t) = \mathbf{f}(t)$, where $\mathbf{A}(t)$ and $\mathbf{B}(t)$ are $m \times n$ matrices, the matrix pencil is defined as $\lambda\mathbf{A}(t) + \mathbf{B}(t)$, λ being a complex parameter.

$$
\begin{cases}
\textit{linear} & \begin{cases}
\textit{semi explicit} & \begin{cases}
\textit{const.} & \begin{cases} \dot{\mathbf{y}}(t) + \mathbf{B}_{11}\mathbf{y}(t) + \mathbf{B}_{12}\mathbf{z}(t) = \mathbf{f}_1(t) \\ \mathbf{B}_{21}\mathbf{y}(t) + \mathbf{B}_{22}\mathbf{z}(t) = \mathbf{f}_2(t) \end{cases} \\
\textit{time var.} & \begin{cases} \dot{\mathbf{y}}(t) + \mathbf{B}_{11}(t)\mathbf{y}(t) + \mathbf{B}_{12}(t)\mathbf{z}(t) = \mathbf{f}_1(t) \\ \mathbf{B}_{21}(t)\mathbf{y}(t) + \mathbf{B}_{22}(t)\mathbf{z}(t) = \mathbf{f}_2(t) \end{cases}
\end{cases} \\
\textit{fully implicit} & \begin{cases}
\textit{const.} & \{\mathbf{A}\dot{\mathbf{x}}(t) + \mathbf{B}\mathbf{x}(t) = \mathbf{f}(t) \\
\textit{time var.} & \{\mathbf{A}(t)\dot{\mathbf{x}}(t) + \mathbf{B}(t)\mathbf{x}(t) = \mathbf{f}(t)
\end{cases}
\end{cases} \\
\textit{nonlin.} & \begin{cases}
\textit{semi explicit} & \begin{cases}
\textit{general} & \begin{cases} \mathbf{f}(t, \mathbf{y}(t), \mathbf{z}(t), \mathbf{y}(t)) = \mathbf{0} \\ \mathbf{0} = \mathbf{g}(t, \mathbf{y}(t), \mathbf{z}(t)) \end{cases} \\
\textit{linear deriv.} & \begin{cases} \dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t), \mathbf{z}(t)) \\ \mathbf{0} = \mathbf{g}(t, \mathbf{y}(t), \mathbf{z}(t)) \end{cases}
\end{cases} \\
\textit{fully implicit} & \begin{cases}
\textit{general} & \{\mathbf{f}(t, \dot{\mathbf{x}}(t), \mathbf{x}(t)) = 0 \\
\textit{linear deriv.} & \{\mathbf{A}(t, \mathbf{x}(t))\dot{\mathbf{x}}(t) + \mathbf{f}(t, \mathbf{x}(t)) = 0
\end{cases}
\end{cases}
\end{cases}
$$

Table 1: Typical Homogeneous DAE Morphology with $\mathbf{x}(t) = [\mathbf{y}(t), \mathbf{z}(t)]^T$

Index	DAE system	$\mathbf{x}$	Condition $\det\{\wp\} \neq 0$
I	$\dot{\mathbf{y}} = \mathbf{r}(t, \mathbf{y}, \boldsymbol{w}, \mathbf{u})$ $\mathbf{0} = \mathbf{g}(t, \mathbf{y}, \boldsymbol{w}, \mathbf{u})$	$[\mathbf{y}, \boldsymbol{w}]^T$	$\det\{\frac{\partial \mathbf{g}}{\partial \boldsymbol{w}^T}\} \neq 0$
II	$\dot{\mathbf{y}} = \mathbf{r}(t, \mathbf{y}, \boldsymbol{w}, \mathbf{u})$ $\mathbf{0} = \mathbf{g}(t, \mathbf{y})$	$[\mathbf{y}, \boldsymbol{w}]^T$	$\det\{\frac{\partial \mathbf{g}}{\partial \mathbf{y}^T} \cdot \frac{\partial \mathbf{r}}{\partial \boldsymbol{w}^T}\} \neq 0$
III	$\dot{\mathbf{y}} = \mathbf{r}(t, \mathbf{y}, \mathbf{z}, \mathbf{u})$ $\dot{\mathbf{z}} = \mathbf{k}(t, \mathbf{y}, \mathbf{z}, \boldsymbol{w}, \mathbf{u})$ $\mathbf{0} = \mathbf{g}(t, \mathbf{y})$	$[\mathbf{y}, \mathbf{z}, \boldsymbol{w}]^T$	$\det\{\frac{\partial \mathbf{g}}{\partial \mathbf{y}^T} \cdot \frac{\partial \mathbf{r}}{\partial \mathbf{z}^T} \cdot \frac{\partial \mathbf{k}}{\partial \boldsymbol{w}^T}\} \neq 0$

Table 2: Semiexplicit Homogeneous DAE morphology and Index.

This matrix is called the reduction index matrix ($\wp$) of the DAE. If $\wp$ is singular it has no inverse and it is not possible to obtain the underlying ODE. The condition for $\wp$ to be full range is presented in Table 2 for some nonlinear semi–explicit DAE systems.

Although the differential index is the most often used, there are other DAEs–related indices like the perturbation index [Hairer *et al.*, 1989]. The perturbation index describes the continuity of the solutions $\mathbf{x}$ of $\gamma(t, \dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}) = \delta(t)$ as $\delta(t) \to \mathbf{0}$. The perturbation index is equal to or greater than the differential index for many classes of DAEs. In the following, index always refers to differential index.

The DAE index is of great interest in order to determine the complexity of a DAE and if DAE solvers can be applied to it.

2.2 Canonical Forms

Most analysis and results of DAE system theory are related to certain forms. Owing to it, the most important forms and their relation to the DAE index are presented in this section.

Definition 4 The Hessenberg form of size r of a DAE is:

$$\begin{aligned}
\dot{\mathbf{x}}_1 &= \gamma_1(t, \mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_r}) \\
\dot{\mathbf{x}}_2 &= \gamma_2(t, \mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_{r-1}}) \\
&\vdots \\
\dot{\mathbf{x}}_i &= \gamma_i(t, \mathbf{x_{i-1}}, \mathbf{x_i}, \cdots, \mathbf{x_{r-1}}), \ 3 \le i \le r-1 \\
&\vdots \\
\mathbf{0} &= \gamma_r(t, \mathbf{x_{r-1}})
\end{aligned} \tag{2}$$

with $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_r}]^T$ and matrix $\left(\frac{\partial \gamma_r}{\partial \mathbf{x_{r-1}}}\right)\left(\frac{\partial \gamma_{r-1}}{\partial \mathbf{x_{r-2}}}\right)\cdots\left(\frac{\partial \gamma_2}{\partial \mathbf{x_1}}\right)\left(\frac{\partial \gamma_1}{\partial \mathbf{x_r}}\right)$ nonsingular.

Theorem 1 [Brenan *et al.*, 1989] Assuming γ_i ($i = 1, 2, \cdots, r$) is sufficiently differentiable, the Hessenberg form of size r is solvable and has index r.

Remark 3 As will be shown later, most mechanical systems and, in particular constrained robotic systems, can be expressed in Hessenberg form of size 3 and if its reduction index matrices ($\wp = \left(\frac{\partial \gamma_3}{\partial \mathbf{x_2}}\right)\left(\frac{\partial \gamma_2}{\partial \mathbf{x_1}}\right)\left(\frac{\partial \gamma_1}{\partial \mathbf{x_3}}\right)$) are nonsingular, this DAE will be index 3. Most constrained robotic systems have configurations where $\wp$ is singular, so these systems are only locally index 3.

Definition 5 The *Standard Canonical Form* of a DAE is:

$$\begin{aligned}
\dot{\mathbf{x}}_1 &= \gamma_1\ (t, \mathbf{x_1}) \\
\zeta\ (t, \mathbf{x_1}, \mathbf{x_2})\, \dot{\mathbf{x}}_2 &= \mathbf{x_2} + \xi\ (t, \mathbf{x_1})
\end{aligned} \tag{3}$$

with $\mathbf{x} = (\mathbf{x_1}, \mathbf{x_2})^T$.

Definition 6 The *Triangular Chain Form* of size r of a DAE is:

$$\begin{aligned}
\gamma_1\ (t, \dot{\mathbf{x}}_1, \mathbf{x_1}) &= \mathbf{0} \\
\gamma_2\ (t, \dot{\mathbf{x}}_1, \dot{\mathbf{x}}_2, \mathbf{x_1}, \mathbf{x_2}) &= \mathbf{0} \\
&\vdots \\
\gamma_r\ (t, \dot{\mathbf{x}}_1, \cdots, \dot{\mathbf{x}}_r, \mathbf{x_1}, \cdots, \mathbf{x_r}) &= \mathbf{0}
\end{aligned} \tag{4}$$

with $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_r}]^T$ and γ_i ($i = 1, 2, \cdots, r$) is either an implicit ODE, a Hessenberg form DAE or a Standard Canonical Form DAE.

Theorem 2 [Brenan *et al.*, 1989] Triangular Chain Form DAEs are always solvable.

3 Numerical Solution Approaches

Two main approaches to numerically solve DAE systems have been proposed. The first one consist in obtaining an ODE with the same DAE's behavior. The second one reformulates the DAE in such a way that available DAE solvers can deal with it.

Another problem in DAE integration are the initial conditions, since most DAE solvers need a consistent initial set of constraints. Although algorithms to obtain consistent initial conditions for some classes of DAEs exist [Brown *et al.*, 1995; Gracia & Pons, 1992; Pantelides, 1988], this point can be hard for some type of DAE systems and in the general case it must be manually obtained by using some kind of heuristic. This problem is not of great relevance in mechanical systems if static initial conditions are considered; in this cases only the inverse kinematics needs to be solved.

3.1 ODE Approach

There are several methods for obtaining an ODE from a DAE. In order to show how these methods can be applied the following explanation will be centered on mechanical systems. Since most mechanical systems are second order they can be expressed in the form:

$$\dot{\mathbf{y}} = \mathbf{z} \tag{5}$$
$$\dot{\mathbf{z}} = \mathbf{A}^{-1}(\mathbf{y})\left[\mathbf{h}(\mathbf{y},\mathbf{z},\mathbf{u}) + \mathbf{G}^T(\mathbf{y})\mathbf{f}\right] \tag{6}$$
$$\mathbf{0} = \mathbf{g}(\mathbf{y}) \tag{7}$$

where $\mathbf{y}$ is the vector of position variables, $\mathbf{z}$ is the vector of velocities, $\mathbf{g}(\mathbf{y})$ are holonomic constraints over the system, $\mathbf{f}$ are the constraining forces, $\mathbf{A}$ is the inertia tensor, $\mathbf{h}$ represent the free dynamics and $\mathbf{G}(\mathbf{y}) = \frac{\partial \mathbf{g}(\mathbf{y})}{\partial \mathbf{y}^T}$ is the jacobian matrix of $\mathbf{g}(\mathbf{y})$. It can be shown that when $\wp = \mathbf{G}(\mathbf{y})\mathbf{A}^{-1}(\mathbf{y})\mathbf{G}^T(\mathbf{y})$ is nonsingular this system has Hessenberg form of size 3; so it will be index III.

One way to obtain an ODE from (5)-(7) is by using the Reduction Transformation approach [McClamroch, 1986], which gives rise to a reduced ODE in $(\mathbf{y},\mathbf{z})$. The procedure is as follows. First, constraining equation 7 is derivated twice in order to obtain $\mathbf{f}$ as a function of $(\mathbf{y},\mathbf{z})$. At this point the following overdetermined DAE [Eich *et al.*, 1990] is obtained:

$$\dot{\mathbf{y}} = \mathbf{z} \tag{8}$$
$$\dot{\mathbf{z}} = \mathbf{A}^{-1}(\mathbf{y})\left[\mathbf{h}(\mathbf{y},\mathbf{z},\mathbf{u}) + \mathbf{G}^T(\mathbf{y})\mathbf{f}\right] \tag{9}$$
$$\mathbf{0} = \mathbf{g}(\mathbf{y}) \tag{10}$$
$$\mathbf{0} = \mathbf{G}(\mathbf{y})\dot{\mathbf{y}} \tag{11}$$
$$\mathbf{f} = -\left(\mathbf{G}(\mathbf{y})\mathbf{A}^{-1}(\mathbf{y})\mathbf{G}(\mathbf{y})^T\right)^{-1} \left(\mathbf{G}(\mathbf{y})\mathbf{A}^{-1}(\mathbf{y})\mathbf{h}(\mathbf{y},\mathbf{z},\mathbf{u}) + \dot{\mathbf{G}}(\mathbf{y})\mathbf{z}\right) \tag{12}$$

Then the value of $\mathbf{f}$ is substituted in (9), and the following reduced ODE is obtained:

$$\dot{\mathbf{y}} = \mathbf{z} \tag{13}$$
$$\dot{\mathbf{z}} = \mathbf{A}^{-1}(\mathbf{y})\big[\mathbf{h}(\mathbf{y},\mathbf{z},\mathbf{u}) - \mathbf{G}^T(\mathbf{y})\left[\mathbf{G}(\mathbf{y})\mathbf{A}^{-1}(\mathbf{y})\mathbf{G}^T(\mathbf{y})\right]^{-1} \left[\mathbf{G}(\mathbf{y})\mathbf{A}^{-1}(\mathbf{y})\mathbf{h}(\mathbf{y},\mathbf{z},\mathbf{u}) + \dot{\mathbf{G}}(\mathbf{y})\mathbf{z}\right] \tag{14}$$

where constraints in the system are implicit.

A different approach is to derivate again to obtain an explicit expression of $\dot{\mathbf{f}}$. This new system corresponds to the underlying ODE of (5)-(7). None of these two approaches generates a state space representation, while other approaches produce an ODE over a state vector representation through the parametrization of the constraining manifold [Fuhrer and Leimkuhler, 1991].

The position constraints do not appear in the resulting expressions due to the problem reformulation. This reformulation is mathematically correct but under discretization such constraints are not satisfied in general, so the numerical solution of the obtained ODE will tend to drift away from the position constraints, resulting in states not satisfying those constraints.

Remark 4 The above expressions are only correct for states if $\mathbf{G}(\mathbf{y})\mathbf{A}^{-1}(\mathbf{y})\mathbf{G}^T(\mathbf{y})$ (reduction index matrix) is nonsingular. As it will be seen later, in most mechanical systems there exist states where this matrix is singular (in practice, they are a subset of singular kinematic configurations).

3.2 DAE Approach

Although a lot of work has been done in recent years in developing DAE solvers [Arevalo *et al.*, 1995], a general DAE solver is not yet available. Nevertheless some powerful tools like GELDA [kunkel *et al.*, 1995] can deal with almost any linear time varying DAE system.

The main drawback of general DAE solvers is that they can only deal at most with index II DAE systems. As aforementioned, most mechanical systems are index III, and it is thus necessary to reduce the DAE system index in order to apply a DAE solver. Some index II solvers are : HEM5 [Brasey, 1994] based on a half explicit method of order 5 [Brasey, 1992]; LSODI [Hindmarsh, 1983] based on a fixed coëfficient implementation of Backward Differentiation formulas (BDF); and DASSL [Brenan *et al.*, 1989] based on a variable stepsize order fixed leading coefficient implementation of BDF formulas. In this work DASSL, the most popular DAE solver, has been used to obtain numerical results (see section 5). In addition to DAE solvers, other tools with additional features like sensitivity analysis of DAE systems, are under development [Maly *et al.*, 1995].

The most forward way to reduce the index of a DAE like (5)-(7) is to substitute the constraint equation (7) by its derivative. The resulting DAE is index II, so it can be integrated with available DAE solvers. The drawback of this approach is that, as in the examples given in the last section, position constraint does not appear in the formulation and the numerical integration could drift away from this invariants. One approach which solves this problem is to reintroduce the position constraints with an additional Lagrange multiplier (μ). This formulation is called the GGL (Gear, Gupta & Leimkuhler) formulation and its application to problems in the form (5)-(6) gives rise to :

$$\dot{\mathbf{y}} = \mathbf{z} + \mathbf{G}^T(\mathbf{y})\mu \tag{15}$$

$$\dot{\mathbf{z}} = \mathbf{A}^{-1}(\mathbf{y})\left[\mathbf{h}\ (\mathbf{y},\mathbf{z},\mathbf{u}) + \mathbf{G}^T(\mathbf{y})\,\mathbf{f}\right] \quad (16)$$
$$\mathbf{0} = \mathbf{g}(\mathbf{y}) \quad (17)$$
$$\mathbf{0} = \mathbf{G}(\mathbf{y})\,\dot{\mathbf{y}} \quad (18)$$

Some formulations, similar to the GGL one, enforce additionally the acceleration through the introduction of the acceleration constraint jointly with an additional Lagrange multiplier. These formulations are called stabilized formulations of the Euler-Lagrange Equations. Although they are very robust from the numerical point of view, they may be inefficient in some cases [Petzold *et al.*, 1993].

In the GGL formulation it is necessary to compute the inverse of the $\mathbf{A}(\mathbf{y})$ matrix. This is a computationally expensive task, and it can be problematic if $\mathbf{A}(\mathbf{y})$ is closely singular. One formulation that has the advantages of GGL and obviates this disadvantage is the following:

$$\dot{\mathbf{y}} = \mathbf{z} + \mathbf{G}^T(\mathbf{y})\,\mu \quad (19)$$
$$\dot{\mathbf{z}} = \mathbf{w} \quad (20)$$
$$\mathbf{0} = -\mathbf{A}(\mathbf{y})\,\mathbf{w} + \mathbf{h}(\mathbf{y},\mathbf{z}) + \mathbf{G}^T(\mathbf{y})\,\mathbf{f} \quad (21)$$
$$\mathbf{0} = \mathbf{g}(\mathbf{y}) \quad (22)$$
$$\mathbf{0} = \mathbf{G}(\mathbf{y})\,\dot{\mathbf{y}} \quad (23)$$

This last formulation is more efficient, owing to the fact that the numerical algorithm does not need to invert the $\mathbf{A}(\mathbf{y})$ matrix, and it has all the advantages of GGL.

Another approach is the Baumgarte's technique [Baumgarte, 1972] which replaces the constraints by a linear combination of them and their derivatives, such as:

$$0 = \mathbf{G}(\mathbf{y})\,\dot{\mathbf{y}} + \alpha\mathbf{g}(\mathbf{y}) \quad (24)$$

or

$$0 = \dot{\mathbf{G}}(\mathbf{y})\,\dot{\mathbf{y}} + \mathbf{G}(\mathbf{y})\,\ddot{\mathbf{y}} + \alpha_1\mathbf{G}(\mathbf{y})\,\dot{\mathbf{y}} + \alpha_0\mathbf{g}(\mathbf{y}) \quad (25)$$

where α_i are selected so that the system in $\mathbf{g}$ defined by the above expressions be a Hurwitz polynomial. Baumgarte's method leads to a regularization of the DAE system, so the Baumgarte DAE and the original DAE have identical analytical solution. The main drawback of the method is that the adequate values of α_i depend on the integration stepsize, so it cannot be selected without a numerical study [Ascher *et al.*, 1992].

In contrast to index reduction techniques which apply differentiation to the Lagrange formulation, there exists another technique which applyes the Coordinate-Split operator to the variational form of the constrained dynamics [Yen *et al.*, 1994]. This formulation applied to (5)-(6) systems gives rise to:

$$\mathbf{P}(\mathbf{y})\,\dot{\mathbf{y}} = \mathbf{P}(\mathbf{y})\,\mathbf{z} \quad (26)$$
$$\mathbf{P}(\mathbf{y})\,\dot{\mathbf{z}} = \mathbf{P}(\mathbf{y})\,\mathbf{A}^{-1}(\mathbf{y})\left[\mathbf{h}(\mathbf{y},\mathbf{z},\mathbf{u})\right] \quad (27)$$
$$\mathbf{0} = \mathbf{g}(\mathbf{y}) \quad (28)$$
$$\mathbf{0} = \mathbf{G}(\mathbf{y})\,\dot{\mathbf{y}} \quad (29)$$

where

$$\mathbf{P}(\mathbf{y}) = \mathbf{X}^T - \left[(\mathbf{G}(\mathbf{y})\,\mathbf{Y})^{-1}\mathbf{G}(\mathbf{y})\,\mathbf{X}\right]^T\mathbf{Y}^T \quad (30)$$

$\mathbf{X}$ and $\mathbf{Y}$ being permutation matrices such that:

$$\mathbf{y} = \mathbf{X}_{\mathbf{y1}} + \mathbf{Y}_{\mathbf{y2}} \quad (31)$$

where $\mathbf{y}_1$ and $\mathbf{y}_2$ are nonintersecting sets of components of $\mathbf{y}$. As can be seen, Lagrange multipliers disappear from the formulation, so it is not necessary to calculate them at each iteration. The Coordinate-Splitting (CS) formulation is advantageous in dealing with certain highly oscillatory multibody systems [Yen *et al.*, 1996].

The above formulations can deal with index III DAEs, but they cannot be applied when the reduction index matrix is singular. One formulation which can deal with singular points [Petzold *et al.*, 1993], uses the formulation of Baumgarte as starting point, but instead of taking the constraints as hard constraints, they are formulated as the following minimization problem:

$$\min_{\mathbf{f}} \quad \frac{1}{2}{\mathbf{M}_c}^T\mathbf{M}_c \ \text{subject to} \quad (32)$$
$$\frac{h^4}{2}\mathbf{f}^T\mathbf{f} \leq \delta \quad (33)$$

where

$$\mathbf{M}_c = \mathbf{g}(\mathbf{y}) + h\mathbf{G}(\mathbf{y})\,\dot{\mathbf{y}} + \frac{1}{2}h^2\mathbf{G}(\mathbf{y})\,\mathbf{h}\ (\mathbf{y},\mathbf{z},\mathbf{u}) + \frac{1}{2}h^2\mathbf{G}(\mathbf{y})\,\mathbf{B}(\mathbf{y})\,\mathbf{f} \quad (34)$$

Solving for $\mathbf{f}$ arises :

$$\begin{aligned}\mathbf{f} = {} & \left[(\mathbf{G}(\mathbf{y})\,\mathbf{B}(\mathbf{y}))^T(\mathbf{G}(\mathbf{y})\,\mathbf{B}(\mathbf{y})) + \epsilon\mathbf{I}\right]^{-1} \\ & \left[(\mathbf{G}(\mathbf{y})\,\mathbf{B}(\mathbf{y}))^T\mathbf{G}(\mathbf{y})\,\mathbf{h}\ (\mathbf{y},\mathbf{z},\mathbf{u})\right] \\ & + \left[\frac{2}{h}(\mathbf{G}(\mathbf{y})\,\mathbf{B}(\mathbf{y}))^T(\mathbf{G}(\mathbf{y})\,\mathbf{z})\right] \\ & + \left[\frac{2}{h^2}(\mathbf{G}(\mathbf{y})\,\mathbf{B}(\mathbf{y}))^T\mathbf{g}(\mathbf{y})\right]\end{aligned} \quad (35)$$

where $\mathbf{B}(\mathbf{y}) = \mathbf{A}^{-1}(\mathbf{y})\,\mathbf{G}^T(\mathbf{y})$, ϵ and h are numerical parameters. Then, the value of $\mathbf{f}$ can be substituted in (6). With this transformation it is obtained a regularized ODE which can be integrated even when $\mathbf{G}(\mathbf{y})\,\mathbf{B}(\mathbf{y})$ is singular. It is important to note that whether or not $\mathbf{G}(\mathbf{y})\,\mathbf{B}(\mathbf{y})$ is singular, the minimum takes place for $\mathbf{M_c} = 0$. For the singular case the degrees of freedom generated by the kernel of $\mathbf{G}(\mathbf{y})\mathbf{B}(\mathbf{y})$ are used to minimize the norm of $\mathbf{f}$. This approach is also used in other algorithms [Murphy *et al.*, 1990] which can deal with singular cases.

This methodology gives rise to an ODE system which must be integrated by an ODE solver. For this reason it properly belongs to section 3.1, but has been described here because it is based on the Baumgarte's technique.

In this paper the Lagrange formulation and the modified GGL approach are used. In order to validate the methodology a theorical study of singular points is also presented.

4 DAE Representation of Constrained Robotics Systems

Constrained robotic systems are those in which the movement of the terminal element or of any point of the kinematic chain is constrained by holonomic or non-holonomic restrictions. In the following sections DAE systems appearing in constrained robotic systems are introduced and studied.

4.1 Robot Kinematics and Dynamics

The unconstrained dynamics equations of a robot can be written as:

$$\mathbf{M}(\theta)\ddot{\theta} + \mathbf{c}\left(\theta, \dot{\theta}\right) + \mathbf{g}(\theta) = \tau - \mathbf{J}^T(\theta)\mathbf{f} \tag{36}$$

where $\theta, \dot{\theta}$, and $\ddot{\theta}$ are the $n \times 1$ vectors of the joint variables, velocities and accelerations, $\mathbf{M}$ is the inertia tensor (expressed as a $n \times n$ symmetric and positive definite inertia matrix), $\mathbf{c}$ is the $n \times 1$ vector representing centrifugal and coriolis effects, $\mathbf{g}$ is the $n \times 1$ vector representing the effects of gravity, $\mathbf{J}$ is the $6 \times n$ manipulator Jacobian matrix, $\mathbf{f}$ is the 6×1 vector of exerted forces on the robot end effector and τ is the $n \times 1$ vector of exerted torques in the joints. The free kinematics is described by

$$\mathbf{x} = \mathbf{kin}(\theta) \tag{37}$$

where $\mathbf{x} \in SO(3)$ represents the position and orientation of the terminal element in the Operational Space, and $\mathbf{kin}$ is a nonlinear mapping between the Configuration Space and the Operational Space. Orientation can be represented by a set of 3 parameters in several ways: Euler angles [Meirovitch, 1988], Pitch-Roll-Yaw angles [Craig, 1986], or quasicoordinates [Paljug *et al.*, 1994].

curve equation	singularities
$a \cdot y + b \cdot x = c$	$\theta_2 \in [0, \pi], \theta_1 = \arctan\left(\frac{a}{b}\right)$
$\frac{(x-a)^2}{b} + \frac{(y-c)^2}{d} = 1$	$\theta_2 \in [0, \pi], \theta_1 = \arctan\left(\frac{c}{a}\right)$

Table 1: Curve analysis

4.2 Robot Arm Constrained to a Surface

4.2.1 System Modeling

The equations describing the behavior of a robot with its terminal element moving over a rigid surface are :

$$\mathbf{M}(\theta)\ddot{\theta} + \mathbf{c}\left(\theta, \dot{\theta}\right) + \mathbf{g}(\theta) = \tau - \mathbf{J}^T(\theta)\mathbf{f} \tag{38}$$

$$\phi(\mathbf{kin}(\theta)) = \mathbf{0} \tag{39}$$

where $\phi(\mathbf{x})$ is the equation of the constraining surface.

In this formulation, $\mathbf{f}$ has dimension 6 while ϕ has dimension 1; so there are more undetermined variables than restrictions. Since this formulation generates an ill-conditioned problem, it is necessary to reduce the number of undetermined variables. This can be done taking into account that $\mathbf{f}$ is normal to the constraining surface [Goldstein, 1950], so:

$$\mathbf{f} = \frac{{\phi_x}^T(\mathbf{kin}(\theta))}{\|{\phi_x}^T(\mathbf{kin}(\theta))\|} f \tag{40}$$

where $\phi_{\mathbf{x}}$ represents the gradient vector $(\frac{\partial \phi}{\partial \mathbf{kin}^T(\theta)})$, and f is a scalar representing the force module. In this way a DAE with only one undetermined variable (f) and only one restriction (g) is obtained.

Expressing (38)-(39) in the form (5)-(7), they become

$$\begin{aligned}
\dot{\theta} &= \omega && (41)\\
\dot{\omega} &= \mathbf{M}^{-1}(\theta)\left[-\mathbf{c}(\theta,\omega) - \mathbf{g}(\theta) + \tau \right. \\
&\quad \left. -\mathbf{J}^T(\theta)\frac{{\phi_x}^T(\mathbf{kin}(\theta))}{\|{\phi_x}^T(\mathbf{kin}(\theta))\|} f\right] && (42)\\
0 &= \phi(\mathbf{kin}(\theta)) && (43)
\end{aligned}$$

Remark 5 For this kind of system the index reduction matrix (which in this case is a scalar) is

$$\phi_{\mathbf{x}}(\mathbf{kin}(\theta))\,\mathbf{J}(\theta)\,\mathbf{M}^{-1}(\theta)\,\mathbf{J}^T(\theta)\,{\phi_x}^T(\mathbf{kin}(\theta)) \tag{44}$$

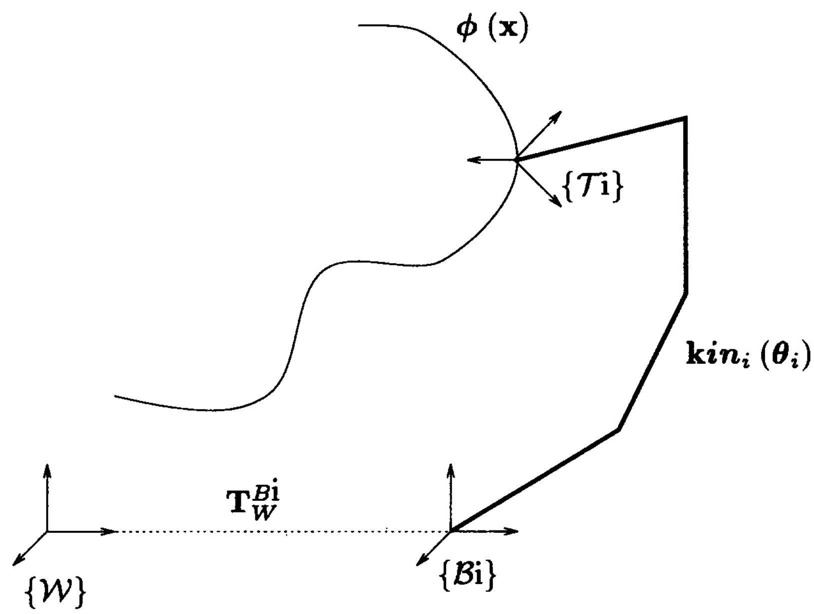


Figure 1: Robot arm constrained to a surface

4.2.2 Singularity Identification

In this section the relation between the singularities of the kinematics Jacobian matrix (**J**) and those of the index reduction matrix ($\wp$) will be analyzed.

Proposition 1 The index reduction matrix 44 is singular iff ${\phi_x}^T(\mathbf{kin}(\theta)) \in \mathrm{Ker}\{\mathbf{J}^T(\theta)\}$

Proof :

As **M** represents the inertia tensor, it is a positive definite full range màtrix, so 44 is a positive semidefinite quadratic from.

As $\wp$ is a scalar, it is singular when it is equal to 0. This can only happen when

$$\mathbf{J}(\theta)^T \phi_{\mathbf{x}}(\mathbf{kin}(\theta))^T = \mathbf{0} \tag{45}$$

which means that

$${\phi_x}^T(\mathbf{kin}(\theta)) \in \mathrm{Ker}\{\mathbf{J}^T(\theta)\} \tag{46}$$

Remark 6 In the particular case that $\mathrm{Rank}\{\mathrm{Ker}\{\mathbf{J}^T(\theta)\}\} = 1$, 46 happens iff the basis vector of $\mathrm{Ker}\{\mathbf{J}^T(\theta)\}$ and ${\phi_x}^T(\mathbf{kin}(\theta))$ are aligned.

This remark has a clear physical meaning: it is well known that constraining forces are normal to the constraining surface, and that $\mathbf{J}^T(\theta)$ relates forces in the Operàtional Space with forces in the Configuration Space; so, when $\mathbf{J}^T(\theta)$ and ${\phi_x}^T(\mathbf{kin}(\theta))$ are aligned, the forces in the Operational Space normal to the constraining surface correspond to null forces in the configuration space. So it is not possible to uniquely determine the constraining forces.

Remark 7 In the singularity analysis only velocity relations are taken into account, so not all the singular configuration will necessarily verify the position restrictions.

As has been shown, the proposed method can deal with some kinematic singularities, although it can not deal with the singularities identified before. The nature of the singularities will depend strongly on constraining surface and robot kinematics. For a 2d. D.D.F. planar robot constrained to some classes of surfaces a complete characterization of the singularities of $\wp$ has been developed by the authors [Costa *et al.*, 1995]. Table 1 shows some examples.

5 Simulations

In order to show the effectiveness of the proposed methodology some examples are presented.

First of all some implementation topics are related, secondly two simulations, corresponding to the case studies are presented. In both cases no controllers are assumed, so the system behaves as an autonomous system.

5.1 Implementation Tools

One of the most tedious and hardest tasks in simulation is the development of the whole set of equations and the validation of the models. To facilitate this tasks, a methodology which makes them in a more natural and optimal way has been developed. The whole process is shown in Figure 2.

First of all, a high level description of the system, similar to the ones described in this paper, is built. This description is made in a symbolic maner with the support of a Robotic Toolbox [Costa *et al.*, 1996], so, no numerical data is needed. From this description, the symbolic manipulator automatically generates all needed expressions like, for example, the derivatives and the jacobian matrix. After that, a complete and simplified set of equations is obtained. Also, jointly to this set of equations, a symbolic analysis of the problem can be performed. This kind of methodology is being highly used in numerical analysis of control problems [Campbell *et al.*, 1994].

Next step is the introduction of the numerical data in order to generate the C or Fortran code. This step is automatically performed, saving a lot of time and assuring a good implementation that, in addition, could be easily modified.

Once the C or Fortran code is available, the only task to do is to simulate the system behavior with a DAE solver. The use of numerical software like DAE solvers needs the manual tuning of some parameters if optimal computation times are desired. After simulation, numerical data representing the system behavior are obtained. These data can be visualized and analyzed with any visualization package.

Although the method is presented for a DAE model, it can be used in many other approaches. For example this method has also been used to generate Reduction Transformation models.

In this work, MapleV R3© [Char, 1991] has been used as symbolic manipulator, DASSL [Brenan *et al.*, 1989] as DAE solver and MatLab^{TM} [MATLAB, 1992] as postprocessing graphical tool.

5.2 Robot Arm Constrained to a Surface

In this section simulation results are presented in order to complement the theoretical developments. All the cases are based on a planar robot, and the orientation parametrization has no singularities. So, all the singularities are due to kinematic ill-conditionaments.

The simulation of the behavior of the system introduced in Section 4.2 has been performed. As test robot, a planar 2 d.o.f robot with 1m long links, 5kg of weight and 1 kg·m^2 inertia momentum and its end effector restricted to a curve has been used. The designed experiment is the following: from time $t = 0$ to $t = 0.5$ the robot remains in the initial position due to the fact that compensating torques are exerted. Then at time $t = 0.5$ the torques are taken off and the whole system evolves freely.

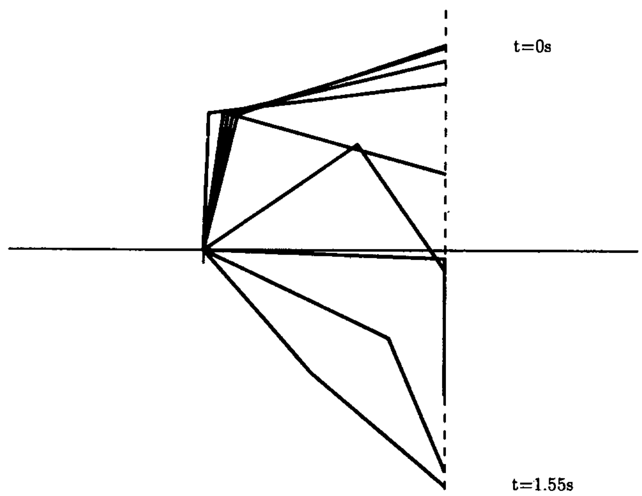
Three different cases have been selected for the simulation. In the first one the robot end effector is constrained to $x = 1$; in this case the theorical singularities are $(0,0)$, $(\pi,0)$, $(0,\pi)$, and (π,π), but these singularities do not satisfy position constraints (Figure 3) and then the system is free of singularities. In the second one the constraining line is $x = 0$; in this case theoretical singularities are the same as that in the case stated before, but now (π,π) and $(0,\pi)$ satisfy position constraints so they are real singularities of this system (Figure 4), because that simulation cannot go on near these points. Finally the curve $x^2 + y^2 = 1$ has been used; in this case there are no theoretical singularities, so there are no real singularities in the system (Figure 5).

6 Conclusions

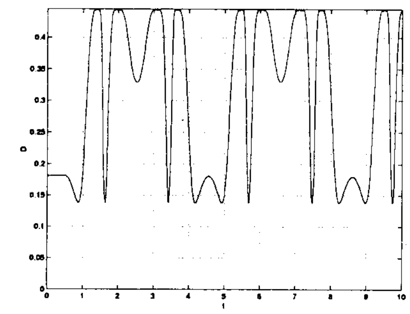
A methodology for modeling constrained robotic systems has been presented. This methodology offers simplicity of development and numerical efficiency. The presented formulation is much more clear and better numerically conditioned than other approaches like the Reduction Transformation.

The simulation of two examples is presented. For one of them a comparative analysis of the Reduction Transformation and the Index Reduction approaches has been performed. In this way, the validity of the theoretical developments and the great interest of the proposed approach have been confirmed.

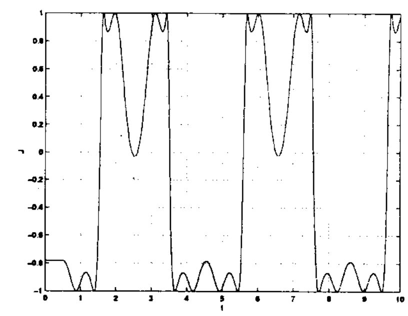
In order to make the model development easier, a methodology which allows comfortable model design and implementation, has been also proposed. This



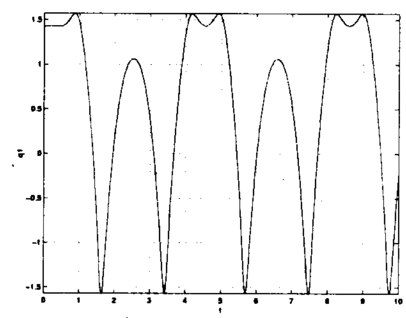
(a) Robot arm trajectory



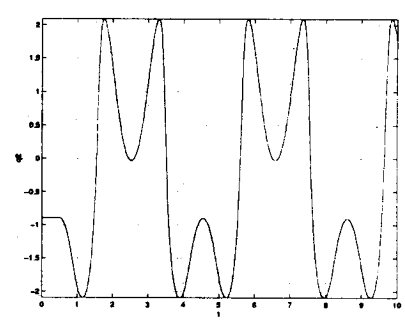
(b) Determinant



(c) Jacobian

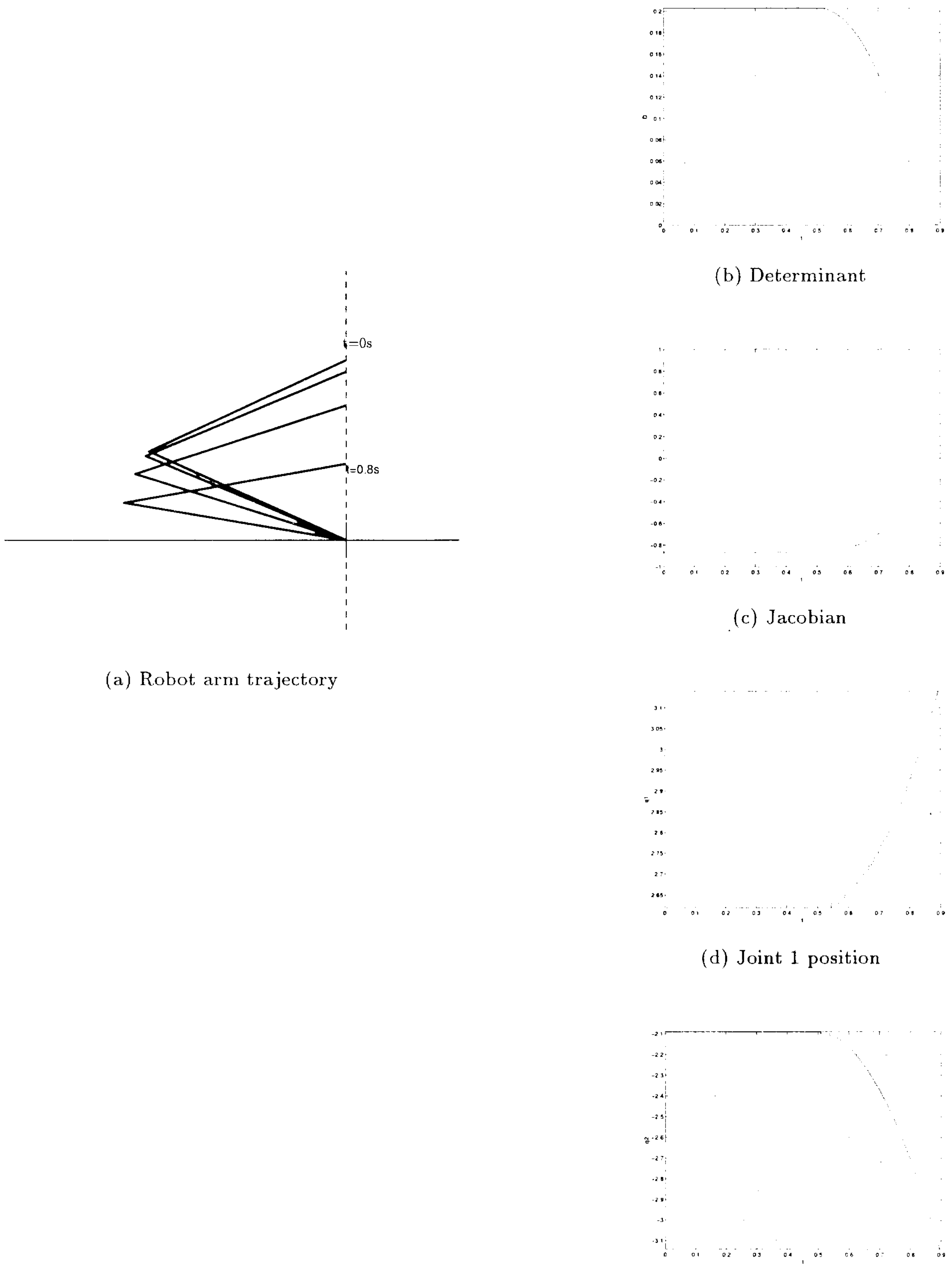


(d) Joint 1 position



(e) Joint 2 position

Figure 3: Evolution for $x = 1$ (Absolute tolerance 10^{-3}, relative tolerance 10^{-3}, integration maximum stepsize $5 \cdot 10^{-6}$, error $5.249 \cdot 10^{-11}$)



(a) Robot arm trajectory

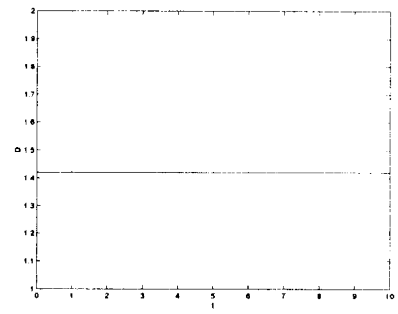
(b) Determinant

(c) Jacobian

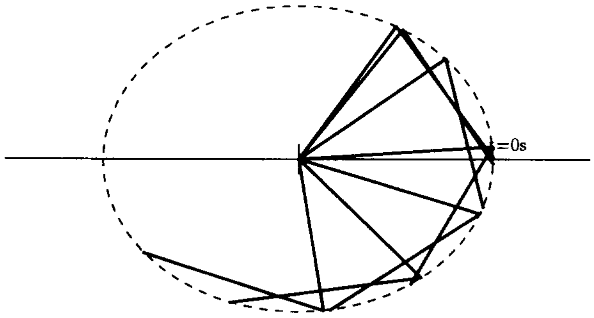
(d) Joint 1 position

(e) Joint 2 position

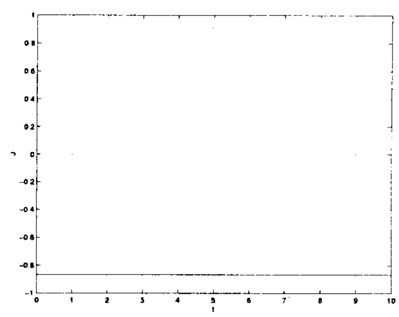
Figure 4: Evolution for $x = 0$ (Absolute tolerance $5 \cdot 10^{-5}$, relative tolerance $5 \cdot 10^{-3}$, integration maximum stepsize $7 \cdot 10^{-6}$, error $1.9 \cdot 10^{-16}$)



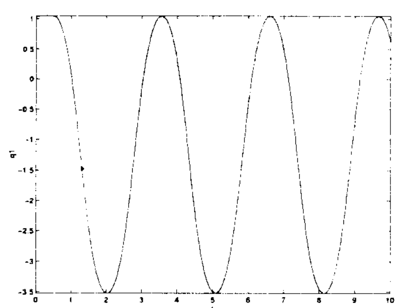
(b) Determinant



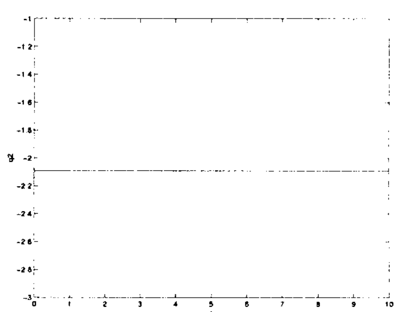
(a) Robot arm trajectory



(c) Jacobian



(d) Joint 1 position



(e) Joint 2 position

Figure 5: Evolution for $x^2 + y^2 = 1$ (a) Robot arm trajectory, (b) Determinant, (c) Jacobian, (d) Joint 1 position, (e) Joint 2 position. (Absolute tolerance 10^{-3}, Relative tolerance 10^{-3}, integration maximum stepsize $5 \cdot 10^{-6}$, error $5.2490 \cdot 10^{-11}$)

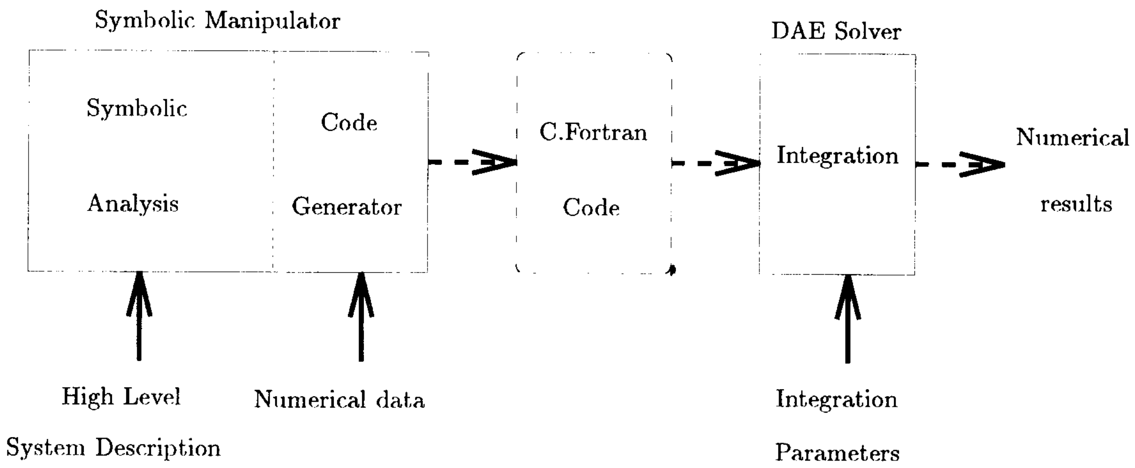


Figure 2: Methodology scheme.

methodology combines symbolic manipulation with numerical analysis.

References

Alan C. H.. "Odepack, a Systematized Collection of ODE Solvers." *Scientifc Computing*, 1983.

Arévalo C., and F. Claus and S. Gustaf. "Stabilized Multistep Methods For Index 2 Euler-Lagrange DAEs." *BIT 3n*, 1995.

Ascher U., H. Chin and S. Reich. Department of Computer Science, UBC, Vancouver. "Stabilization of DAE's and Invariant Manifolds." Technical Report, 1992.

Baumgarte J.. "Stabilization of Constraints and Integrals of Motion in Dynamical Systems." *Comp. Math. Appl. Mech. Eng*, 1972.

Brasey V.. *HEM5 User's Guide*, 1994.

Brasey V.. "A Half-Explicit Runge-Kutta Method of order 5 for solving constrained mechanical systems." *Computing*, 1992.

Brenan K. A., S.L. Campbell and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* SIAM, 1989.

Brown N. P., and C. Alan and L. Petzold. Univ. of Minnesota, Supercom. Inst. "Consistent Initial Condition Calculation for Differential-Algebraic Systems", 1985.

Bruce W. C.. *Maple V Language Reference Manual.* Springer-Verlag, 1991.

Constantinos C. P.. "The Consistent Initialization Of Differential-Algebr aic Systems." *SIAM J. Sci. Stat. Comput.*, 1988.

Costa-Castelló R., R. Griñó and L. Basañez. "Análisis de la Matriz de Reducción de Indice en el Modelado de Sistemas Robotizados Sometidos a Restricciones: Formulación en Forma de Sistemas Singulares." Report Tècnic IC-DT-9533, Institut de Cibernètica, 1995.

Costa-Castelló R., R. Griñó and L. Basañez. *Una Robot Toolbox Per Maple* Report Tècnic IC-DT-9619, Institut de Cibernètica, 1996.

Craig J.J.. *Introduction to Robotics, Mechanics Control.* Addison Wesley, 1986.

Dai L.. *Singular Control Systems.* Springer-Verlag New York Berlin, Lecture Notes in Control and Information

Sciences. Berlin, 1989.

Eich E., B. Leimkuhler and S. Reich. Helsinky Univ. of Tech., Inst. of Mathematics. "Stabilization and Projection Methods for Multibody Dynamics." 1990.

Ellis R. E. and S.L. Ricker. "Two Numerical Issues In Simulating Constrained Robot Dynamics." *IEEE Trans. On Systems. Man and Cybernetics*, 1994.

Führer C. and B.J. Leimkuhler. *Numerical Solution Of Differential -Algebraic Equations for Constrained Mechanical Motion.* Numerische Mathematik, Springer-Verlang, 1991.

Goldstein H.. *Classical Mechanics.* Addison-Wesley, 1950.

Gracia X. and J.M. Pons. "A Generalized Geometric Framework for Constrain ed Systems." *Dif. Geom. and Appl.*, 1992.

Hairer E., C. Lubich and M. Roche. *The Numerical Solution of Differ ential-Algebraic Systems by Runge-Kutta Methods.* Springer-Verlag Berlin, 1989.

Harier E., S.P. Norsett and G. Wanner. *Solving Ordinary Differential Equations.* Springer Verlag Series In Computational Mathematics, 1991.

Harris N. M.. "Feedback Stabilizaction of Control Systems Described by a Class of Nonlinear Differential-Algebraic Equations." *Systems & Control Letters*, 1990.

Harris N.. "Singular Systems of Differential Equations as Dynamic Models for Constrained Robot Systems." *Proc. of the 1986 IEEE Int. Conf. on Robotics and Automation.* San Francisco, California, 1986.

Kathryn W. L.. *Efficient Dynamic Simulation Of Robotic Mechanisms.* The Kluwer Int. Series in Enginnering and Computer Science. Kluwer Academic Publishers, 1993.

Krishnan H. and N. H. McClamroch. "Tracking in Control Systems Described by Nonlinear Differential-Algebraic Equations with Applications to Constrained Robot Systems." *Proc. of the American Control Conference.* Sn. Fco, Califo rnia, 1993.

Kumar A. and D. Prodromos. "Feedback Control of Nonlinear Differential- Algebraic-Equation Systems." *AIChE journal*, 1995.

Kunkel P., M. Volker, R. Werner and W. Jö. Technische Universit ät Chemnitz-Zwickau. "GELDA: A software package for the solution of General Linear Differential Algebraic Equations." 1995.

Laroussi K., H. Hemami and E. G. Ralph." Coordination Of Two Planar Robots In Lifting." *IEEE Journal of Robotics and Automation*, 1988.

Maly T. and L. Petzold. Univ. of Minnesota, Dept. of Comp Science. *Numerical Methods and Software for Sensitivity Analysis of Differential-Algebraic Systems.* 1995.

McMillan S., P. Sadayappan and D. E. Orin. "Parallel Dynamic Simulation Of Multiple Manipulator Systems: Temporal Versus Spatial Methods." *IEEE Trans. On Systems, Man And Cybernetics*, 1994.

McMillan S., P. Sadayappan and D. E. Orin. "Efficient Dynamic Simulation Of Multiple Manipulator Systems with Singular Configurations." *IEEE Trans. On Systems. Man And Cybernetics*, 1994.

Meirovitch L.. Methods Of Analytical Dynamics. Series In Advanced Engineering, Classic Textbook Reissue Series, Irving Shames, Mc Graw-Hill, 1988.

Nakamura Y. K. Nagai and Y. Tsuneo. "Massachusetts Institute of Technology. Dynamics and Stability In Coordination Of Multiple Robotics Mechanisms." *The Int. Journal of Robotics Research*, 1989.

Paljug E., Y. Xiapoing and K. Vijay. "Control of Rolling Contacts In Multi-Arm Manipulation." *IEEE Trans. On Robotics And Automation.* Nice France, 1994.

Petzold L. R., Ren Y. and M. Timothy. University of Minnesota, Department of Computer Science. *Numerical Solution of Differential-Algebraic Equations with Ill-Conditioned Constraints.* Technical Report, 1993.

Simeon B., F. Grupp, C. Führer and P.Rentrop. "A Nonlinear Truck Model and its Treatment as a Multibody System." *Journal of Computational and Applied Mathematics.* Elsevier Sience B.V, 1994.

Sira-Ramirez H.. "The Differential Algebraic Approach in Nonlinear Dynamical Feedback Controlled Landing Maneuvers." *IEEE Transactions on Automatic Control*, 1992.

Stanley A. S. and R. Cannon and H. Robert. "Object Impedance Control for Control for Cooperative Manipulation: Theory and Experimental Results." *IE EE Trans. on Robotics and Automation*, 1992.

Stephen L. C., E. Moore and Z. Yangchum. "Utilization of Automatic Differentiation in Control Algorithms." *IEEE Trans. on Aut. Control*, 1994.

Steve H. M., T. W. John and G. Saridis.

"Simulation Of Cooperating Robot Manipulators On a Mobile Platform." *Proc. of the 1990 IEEE Int. Conf. On Robotics And Automation.* Hyatt Regency Cincinnati, Ohio, 1990.

The MathWorks, Inc. *MATLAB Reference Guide,* 1992.

Unseren, M. A. and A.J. Koivo. "Reduced Order Model and Decoupled Control Architecture for Two Manipulators Holding an Object." *In Proc. of the 1989 IEEE Int. Conf. on Robotics and Automatization.* Scottsdale Arizona, 1989.

Yen J. and R. P. Linda. "An Efficient Newton-Type Iteration for the Numerical Solution of Highly Oscillatory Constrained Multibody Dynamic Systems." Technical Report, AHPCRC TR 96-001, University of Minnesota, Department of Computer Science, 1996.

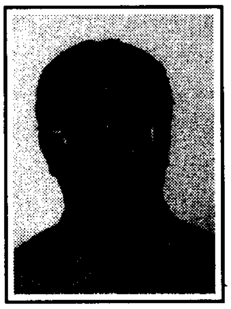
Yen J.. "Center for Computer-Aided Design. College of Engineering." The University of Iowa, *Dynamic Simulation of Contact Between Geometric Objects in Three Dimensional Space.* Technical Report, 1995.

Yen J and R. P. Linda. "On the Numerical Solution of Constrained Multibody Dynamic Systems." Technical Report, AHPCRC TR 94-038, University of Minnesota, Department of Computer Science, 1994.

Yim W.. "Feedback Linearization of Differential-Algebraic Systems and For ce and Position Control of Manipulators." *Proc. of the American Control Conference.* San Fco. California, 1993.

Zuofeng L., T. Tarn and K. B. Antal. "Dynamic Workspace Analysis Of Multiple Cooperating Robot Arms." *IEEE Trans. On Robotics And Automation,* 1991.

***Ramon Costa-Castelló** was born in Lleida, Spain on June 1970. He received the MSc degree in Computer Science by the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain in 1993, and he is currently a PhD candidate in Robotics and Automatic Control at the Instituto de Cibernética (UPC). From July 1996 he is assistant professor of Automatic Control at the UPC. Ramon Costa-Castelló is a student member of IEEE, SIAM and ASME. His research interest includes motion planning and control of multi-robot systems, nonlinear control theory and differential algebraic systems.*



***Robert Griñó** was born in Barcelona, Spain on April 1965. He received the MSc degree in Electrical Engineering (1989) and the PhD degree in Automatic Control (1997) from the Universitat Politècnica de Catalunya (UPC), Barcelona. During 1990 and 1991 he worked as research assistant at the Instituto de Cibernética (UPC), and since 1992, he is assistant professor at the Systems Engineering and Automatic Control Department and at the Instituto de Cibernética (UPC), Universitat Politècnica de Catalunya. Robert Griñó is member of IEEE, SIAM, SCS, and affiliate member of IFAC. His research interests includes nonlinear control, stability theory, robust control, sensitivity theory, differential algebraic systems and identification.*

***Luis Basañez** received the PhD degree in Electrical Engineering by the Universitat Politècnica de Catalunya (UPC) in 1975. From 1976 to 1987 he was Vice-director of the Instituto de Cibernética (UPC) and Director from 1987 to 1990. From 1986 he is full professor of System Engineering and Automatic Control at the Universitat Politècnica de Catalunya, in Barcelona, Spain. He has served as member of the Executive Committee of the International Federation of Robotics (IFR) from 1987 to 1992 and, presently, he is the Spanish Delegate at the IFR. From 1996 he is Vice-chairman of the Technical Board of the International Federation of Automatic Control (IFAC). In 1995 he received the Best Paper Award of the IEEE Int. Symp. on Assembly and Task Planning in Pittsburgh. His present research interest includes task planning and coordination of multi-robot systems, sensor integration and active perception.*