Costa Castelló, Ramón; Griñó, Robert; Basañez, Luis
DAE Methods in Constrained Robotics System Simulation
Instituto Politécnico Nacional
Distrito Federal, México

Available in: http://www.redalyc.org/articulo.oa?id=61510304
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,basanex}@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 application 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 part of a kinematic chain is restricted to holonomic or nonholonomic constraints. These systems are a current research topic in modeling [Unseren & Koivisto, 1991; Harris, 1986] and simulation [Murphy & Ricker, 1990; McMillan et al., 1994; McMillan et al., 1998] and control [Paljug et al., 1994; Nakamura et al., 1991; Schneider & Cannon, 1989; Laroussi et al., 1994]. This work deals with modeling and simulation using 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 method which allows a quite simple formulation but it does not seem very efficient [Murphy et al., 1990]. This last approach is others like the Reduction Transformation [McClamroch, 1986] manipulate the initial differential–algebraic models to obtain purely differential models. As will be shown, the obtained models are usually quite complex, they hide underlying physical phenomena which are difficult to analyse and generate simulation problems due to the introduction of constraints on the differential model. These approaches can be formulated over Configuration Space (C) for classical Lagrange formulation, but recently...
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 constrained robotics systems is presented in the paper.

### 2 DAE Basic Theory

**Definition 1** A DAE system is a set of equations which can be expressed in general

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

where $\gamma: \mathbb{R}^{1+2n+m} \rightarrow \mathbb{R}^n$, $\frac{\partial \gamma}{\partial x}$ is singular \{ $\frac{\partial \gamma}{\partial x} < n$, $x \in \mathbb{R}^n$, and $u \in \mathbb{R}^m$. $u$ is the input system.

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

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

Present knowledge of nonlinear DAE systems to some morphologies, the most typical presented in Table 1.

### 2.1 DAE Index

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

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

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

Table 2 shows the relation between the index and the morphology of some non–linear explicit DAE systems. In order to determine the index of a DAE, this is differented with respect to $t$.
$$\begin{align*}
\text{semi explicit} & \quad \begin{cases}
\text{const.} & \quad \begin{cases}
y(t) + B_{11}y(t) + B_{12}z(t) = f_1(t) \\
B_{21}y(t) + B_{22}z(t) = f_2(t)
\end{cases} \\
\text{time} & \quad \begin{cases}
y(t) + B_{11}(t)y(t) + B_{12}(t)z(t) = f_1(t) \\
B_{21}(t)y(t) + B_{22}(t)z(t) = f_2(t)
\end{cases} \\
\text{var.} & \quad \begin{cases}
y(t) + B_{11}(t)y(t) + B_{12}(t)z(t) = f_1(t) \\
B_{21}(t)y(t) + B_{22}(t)z(t) = f_2(t)
\end{cases}
\end{cases} \\
\text{fully implicit} & \quad \begin{cases}
\text{const.} & \quad \begin{cases}
A\dot{x}(t) + Bx(t) = f(t)
\end{cases} \\
\text{time} & \quad \begin{cases}
A(t)\dot{x}(t) + B(t)x(t) = f(t)
\end{cases} \\
\text{var.} & \quad \begin{cases}
A(t)\dot{x}(t) + B(t)x(t) = f(t)
\end{cases}
\end{cases}
\end{align*}$$

**Table 1:** Typical Homogeneous DAE Morphology with $x(t) = [y(t), z(t)]^T$.

<table>
<thead>
<tr>
<th>Index</th>
<th>DAE system</th>
<th>x</th>
<th>Condition $\det{\mathbf{p}} \neq 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>$\dot{y} = r(t,y,w,u)$, $0 = g(t,y,w,u)$</td>
<td>$[y,w]^T$</td>
<td>$\det\frac{\partial g}{\partial \omega^T} \neq 0$</td>
</tr>
<tr>
<td>II</td>
<td>$\dot{y} = r(t,y,w,u)$, $0 = g(t,y)$</td>
<td>$[y,w]^T$</td>
<td>$\det\left{\frac{\partial g}{\partial y^T} \cdot \frac{\partial r}{\partial \omega^T}\right} \neq 0$</td>
</tr>
<tr>
<td>III</td>
<td>$\dot{y} = r(t,y,z,u)$, $\dot{z} = k(t,y,z,w,u)$, $0 = g(t,y)$</td>
<td>$[y,z,w]^T$</td>
<td>$\det\left{\frac{\partial g}{\partial y^T} \cdot \frac{\partial r}{\partial \omega^T} \cdot \frac{\partial k}{\partial \omega^T}\right} \neq 0$</td>
</tr>
</tbody>
</table>

**Table 2:** Semiexplicit Homogeneous DAE morphology and Index.
This matrix is called the reduction index matrix \( \varphi \) of the DAE. If \( \varphi \) is singular it has no inverse and it is not possible to obtain the underlying ODE. The condition for \( \varphi \) to be full rank 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 \( x \) of \( \gamma(t, \dot{x}, x, u) = \delta(t) \) as \( \delta(t) \to 0 \). 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:

\[
\begin{align*}
\dot{x}_1 &= \gamma_1(t, x_1, x_2, \ldots, x_r) \\
\dot{x}_2 &= \gamma_2(t, x_1, x_2, \ldots, x_{r-1}) \\
&\vdots \\
\dot{x}_i &= \gamma_i(t, x_{i-1}, x_1, \ldots, x_{r-1}), \quad 3 \leq i \leq r-1 \\
&\vdots \\
0 &= \gamma_r(t, x_{r-1})
\end{align*}
\]

with \( x = [x_1, x_2, \ldots, x_r]^T \) and matrix

\[
\left( \frac{\partial \gamma_r}{\partial x_{r-1}} \right) \left( \frac{\partial \gamma_{r-1}}{\partial x_{r-2}} \right) \cdots \left( \frac{\partial \gamma_2}{\partial x_1} \right) \left( \frac{\partial \gamma_1}{\partial x_r} \right)
\]

nonsingular.

**Theorem 1** [Brenan et al., 1989] Assuming \( \gamma_i (i = 1, 2, \ldots, r) \) 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 \( \varphi = \left( \frac{\partial \gamma_1}{\partial x_3} \right) \left( \frac{\partial \gamma_2}{\partial x_3} \right) \left( \frac{\partial \gamma_3}{\partial x_3} \right) \) are.

is:

\[
\begin{align*}
\dot{x}_1 &= \gamma_1(t, x_1) \\
\dot{x}_2 &= \gamma_2(t, x_1, x_2) \\
\dot{x}_2 &= \gamma_2(t, x_1, x_2) \\
0 &= \gamma_r(t, x_{r-1})
\end{align*}
\]

with \( x = (x_1, x_2)^T \).

**Definition 6** The Triangular Chain Form (TCF) DAE is:

\[
\begin{align*}
\gamma_1(t, x_1, x_1) &= 0 \\
\gamma_2(t, x_1, x_2, x_1, x_2) &= 0 \\
&\vdots \\
\gamma_r(t, x_1, \ldots, x_r, x_1, \ldots, x_r) &= 0
\end{align*}
\]

with \( x = [x_1, x_2, \ldots, x_r]^T \) and \( \gamma_i (i = 1, 2, \ldots, r) \) either an implicit form of an ODE or a Hessenberg form of a Standard Canonical Form DAE.

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

### 3 Numerical Solution

Two main approaches to numerically solve DAEs have been proposed. The first one consists in reformulating the DAE as an ODE with the same DAE's behavior. This reformulates the DAE in such a way that ordinary ODE solvers can deal with it.

Another problem in DAE integration is the initial conditions, since most DAE solvers need consistent initial set of constraints. Although it is possible to obtain consistent initial conditions for some classes of DAEs exist [Brown et al., 1995; Gracia and Pantelides, 1988], this point can be hard for general DAE systems and in the general case it must be obtained by using some kind of heuristic.

It is not of great relevance in mechanical systems when initial conditions are considered; in this case the inverse kinematics needs to be solved.

#### 3.1 ODE Approach
second order they can be expressed in the form:

\[
\dot{y} = z \\
\dot{z} = A^{-1}(y) [h(y, z, u) + G^T(y)f] \\
0 = g(y)
\]

(5) (6) (7)

where \(y\) is the vector of position variables, \(z\) is the vector of velocities, \(g(y)\) are holonomic constraints over the system, \(f\) are the constraining forces, \(A\) is the inertia tensor, \(h\) represent the free dynamics and \(G(y) = \frac{\partial g(y)}{\partial y}\) is the jacobian matrix of \(g(y)\). It can be shown that when \(\phi = G(y)A^{-1}(y)G^T(y)\) 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 \((y, z)\). The procedure is as follows. First, constraining equation 7 is derivated twice in order to obtain \(f\) as a function of \((y, z)\). At this point the following overdetermined DAE [Eich et al., 1990] is obtained:

\[
\dot{y} = z \\
\dot{z} = A^{-1}(y) [h(y, z, u) + G^T(y)f] \\
0 = g(y)
\]

(8) (9) (10) (11) (12)

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

\[
\dot{y} = z \\
\dot{z} = A^{-1}(y) [h(y, z, u) - G^T(y) [G(y)A^{-1}(y)G^T(y)]^{-1} [G(y)A^{-1}(y)h(y, z, u) + \dot{G}(y)z]]
\]

(13) (14)

where constraints in the system are implicit.

A different approach is to derivate again to obtain an explicit expression of \(f\). This new system corresponds to a conditionally stiff system, which needs to be considered carefully and solved carefully.

The position constraints do not appear in the resulting expressions due to the problem reformulation. This reformulation is mathematically correct; however, for numerical discretization such constraints are not satisfied, and they will tend to drift away from the position constraint, resulting in states not satisfying those constraints.

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

### 3.2 DAE Approach

Although a lot of work has been done in recent years on developing DAE solvers [Arevalo et al., 1995], a general-purpose DAE solver is not yet available. Nevertheless, there are powerful tools like GELDA [Kunkel et al., 1994] which are able to handle almost any linear time varying DAE system.

The main drawback of general DAE solvers is that they can only deal with systems with index III. For the aforementioned, most mechanical systems are index II, and it is thus necessary to reduce the DAE system in order to apply a DAE solver. Some index I solvers based on a fixed coefficient implementation of the implicit ODE solver HEM5 [Brasey, 1994] based on a half explicit ODE solver of order 5 [Brasey, 1992]; LSODI [Hindmarsh, 1983], based on a fixed coefficient implementation of BDF formulas; and DASSL [Brown et al., 1989] based on a variable stepsize order five fixed coefficient implementation of BDF formulas. For work DASSL, the most popular DAE solver for solving stiff DAE systems, is used to obtain numerical results (see section 3.2). In addition to DAE solvers, other tools with similar 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 equations by its derivative. The resulting DAE is index II, and it can be integrated with available DAE solvers. A drawback of this approach is that, as in the flow chart given in the last section, position constraints may not appear in the formulation and the numerical solution may drift away from these invariants. One method which solves this problem is to introduce the additional Lagrange multiplier.
\[
\begin{align*}
\dot{z} &= \mathbf{A}^{-1}(y) [h(y, z, u) + \mathbf{G}^T(y) f] \\
0 &= g(y) \\
0 &= \mathbf{G}(y) \dot{y}
\end{align*}
\]

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

\[
\begin{align*}
\dot{y} &= z + \mathbf{G}^T(y) \mu \\
\dot{z} &= w \\
0 &= -\mathbf{A}(y) w + h(y, z) + \mathbf{G}^T(y) f \\
0 &= g(y) \\
0 &= \mathbf{G}(y) \dot{y}
\end{align*}
\]

This last formulation is more efficient, owing to the fact that the numerical algorithm does not need to invert the \(\mathbf{A}(y)\) 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:

\[
0 = \mathbf{G}(y) \dot{y} + \alpha \mathbf{g}(y)
\]

or

\[
0 = \dot{\mathbf{G}}(y) \dot{y} + \mathbf{G}(y) \dot{y} + \alpha_1 \mathbf{G}(y) \dot{y} + \alpha_0 \mathbf{g}(y)
\]

where \(\alpha_i\) 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 \(\alpha_i\) 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

\[
\begin{align*}
\mathbf{P}(y) \dot{z} &= \mathbf{P}(y) \mathbf{A}^{-1}(y) [h(y, z, u) + \mathbf{G}^T(y) f] \\
0 &= \mathbf{g}(y) \\
0 &= \mathbf{G}(y) \dot{y}
\end{align*}
\]

where

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

\(\mathbf{X}\) and \(\mathbf{Y}\) being permutation matrices such that

\[
y = \mathbf{X} y_1 + \mathbf{Y} y_2
\]

where \(y_1\) and \(y_2\) are nonintersecting sets of \(y\). As can be seen, Lagrange multipliers come from the formulation, so it is not necessary to solve for them at each iteration. The Coordinate-formulation is advantageous in dealing with highly oscillatory multibody systems [Yeong, 1992].

The above formulations can deal with index-\(2\) but they cannot be applied when the right hand\-side matrix is singular. One formulation which handles singular points [Petzold et al., 1993], uses the idea of Baumgarte as starting point, but instead of applying the constraints as hard constraints, they are approximated by the following minimization problem:

\[
\min_f \quad \frac{1}{2} \mathbf{M}_c^T \mathbf{M}_c \quad \text{subject to} \quad \frac{h^4}{2} f^T f \leq \delta
\]

where

\[
\mathbf{M}_c = \mathbf{g}(y) + h \mathbf{G}(y) \dot{y} + \frac{1}{2} h^2 \mathbf{G}(y) B(y) f + \frac{1}{2} h^2 \mathbf{G}(y) B(y) f
\]

Solving for \(f\) arises :

\[
\begin{align*}
f &= \left[ \mathbf{G}(y) \mathbf{B}(y) \right]^T \mathbf{g}(y) \\
&= + h \left[ \mathbf{G}(y) \mathbf{B}(y) \right]^T \mathbf{h}(y, z) \\
&+ \left[ \frac{2}{h} \mathbf{G}(y) \mathbf{B}(y) \right]^T \mathbf{G}(y) z \\
&+ \left[ \frac{2}{h^2} \mathbf{G}(y) \mathbf{B}(y) \right]^T \mathbf{g}(y)
\end{align*}
\]

where \(\mathbf{B}(y) = \mathbf{A}^{-1}(y) \mathbf{G}^T(y), \epsilon \) and \(h\) represent the errors. The well known of Fromm et al. [1958],

\[
\frac{1}{2} \mathbf{M}_c^T \mathbf{M}_c \quad \text{subject to}
\]

\[
\frac{h^4}{2} f^T f \leq \delta
\]
freedom generated by the kernel of $G(y)B(y)$ are used to minimize the norm of $f$. 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 theoretical 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 nonholonomic 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:

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

(36)

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

$$x = \text{kin} (\theta)$$

(37)

where $x \in SO(3)$ represents the position and orientation of the terminal element in the Operational Space, and $\text{kin} (\theta)$ is the kinematic function.

<table>
<thead>
<tr>
<th>curve equation</th>
<th>singularities</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a \cdot y + b \cdot x = c$</td>
<td>$\theta_2 \in [0, \pi], \theta_1 = \arctan\left(\frac{c}{b}\right)$</td>
</tr>
<tr>
<td>$\frac{(x-a)^2}{b^2} + \frac{(y-c)^2}{d^2} = 1$</td>
<td>$\theta_2 \in [0, \pi], \theta_1 = \arctan\left(\frac{c}{b}\right)$</td>
</tr>
</tbody>
</table>

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 moving its terminal element moving over a rigid surface are:

$$M(\theta)\ddot{\theta} + c(\theta, \dot{\theta}) + g(\theta) = \tau - J^T(\theta)f \quad \phi(\text{kin} (\theta)) = 0$$

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

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

$$f = \frac{\phi_x^T(\text{kin} (\theta))}{\|\phi_x^T(\text{kin} (\theta))\|}f$$

where $\phi_x$ represents the gradient vector $\left(\frac{\partial \phi}{\partial x}\right)$. The obtained system has $f$ as a scalar representing the force module. In this formulation the DAE with only one undetermined variable (the terminal restriction $g$) is obtained.

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

$$\dot{\theta} = \omega$$

$$\dot{\omega} = M^{-1}(\theta)\left[-c(\theta, \omega) - g(\theta) + J^T(\theta)\frac{\phi_x^T(\text{kin} (\theta))}{\|\phi_x^T(\text{kin} (\theta))\|}f\right]$$

$$0 = \phi(\text{kin} (\theta))$$

Remark 5 For this kind of system the independent variable is $	heta$. The evolution of $\omega$ is calculated from the above equation, and the position and orientation of the terminal element are known through the kinematic function.
4.2.2 Singularity Identification

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

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

**Proof:**

As $\mathbf{M}$ represents the inertia tensor, it is a positive definite full range matrix, so $\mathbf{44}$ is a positive semidefinite quadratic form.

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

$$\mathbf{J}(\theta)^T \phi_x(\text{kin}(\theta))^T = 0 \quad (45)$$

which means that

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

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

Space; so, when $\mathbf{J}^T(\theta)$ and $\phi_x^T(\text{kin}(\theta))$ the forces in the Operational Space and the constraining surface correspond to null configuration space. So it is not possible to determine the constraining forces.

**Remark 7** In the singularity analysis, no side relations are taken into account, so a singular configuration will necessarily verify all restrictions.

As has been shown, the proposed method works with some kinematic singularities, although it is not able to deal with the singularities identified before. The location of the singularities will depend strongly on the surface and robot kinematics. For a planar robot constrained to some classes, a complete characterization of the singular configurations has been developed by the authors [Costa et al.]. Figure 1 shows some examples.

5 Simulations

In order to show the effectiveness of...
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 task, 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 manner 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.

5.2 Robot Arm Constrained to a Plane

In this section simulation results are presented to complement the theoretical developments. All cases are based on a planar robot, and the case parametrization has no singularities. Some singularities are due to kinematic ill-conditioning.

The simulation of the behavior of the robot introduced in Section 4.2 has been performed for a robot, a planar 2 d.o.f robot with 1m long links of weight and 1 kg·m² inertia moment at the end effector restricted to a curve has been used. The 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 \( t = 0.5 \) the torques are taken off and the whole 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 singularities are \((0,0), (\pi,0), (0,\pi), \) and \((\pi,\pi)\), but singularities do not satisfy position constraints and then the system is free of singularities; in the second one the constraining line is \( y = 0 \); in this case the theoretical singularities are the same as stated before, but now \((\pi,\pi)\) and \((0,\pi)\) satisfy constraints so they are real singularities of the system (Figure 4), because that simulation cannot be performed at these points. Finally the curve \( x^2 + y^2 = 1 \) is 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 great advantages in terms of development and numerical efficiency. The theoretical formulation is much more clear and better conditioned than other approaches like the Reduction Transformation.

The simulation of two examples is presented. One of them is a comparative analysis of the Reduction Transformation and the Index Reduction method. In both cases the simulation has been performed. In this way, the validation of theoretical developments and the great improvement in numerical conditions are confirmed.
Figure 3: Evolution for $x = 1$ (Absolute tolerance $10^{-3}$, relative tolerance $10^{-3}$, integration maximum $5 \times 10^{-6}$, error $5.249 \times 10^{-11}$)
Figure 4: Evolution for $x = 0$ (Absolute tolerance $5 \cdot 10^{-5}$, relative tolerance $5 \cdot 10^{-3}$, integration method steps size $7 \cdot 10^{-6}$, error $1.9 \cdot 10^{-16}$)
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 max 5 \times 10^5, steps 5 \times 10^4, 10^{-11})
methodology combines symbolic manipulation with numerical analysis.

References


**Ramon Costa-Castelló** was born in Lleida, Spain on June 1970. He received the MS 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 Cibernetica (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 MS degree in Electrical Engineering (1989) and the PhD degree in Automatic Control (1997) from the Universitat Politècnica de Catalunya (UPC), Barcelona. During 1992 and 1991 he worked as research assistant at the Instituto de Cibernetica (UPC), and since 1992, he is assistant professor at the Systems Engineering and Automatic Control Department and at the Instituto de Cibernetica (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 Cibernetica (UPC) and Director from 1987 to 1990. From 1986 he was 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.