

# COORDINATE REDUCTION STRATEGIES IN MULTIBODY DYNAMICS: A REVIEW

E. Pennestri, P.P. Valentini

Dip. di Ingegneria Meccanica, Università di Roma "Tor Vergata", Roma, ITALY, pennestri@mec.uniroma2.it

**Abstract:** *The paper reviews different methods of orthogonalization of constraints adopted in multibody dynamics formulations. A new approach, based on Schur decomposition, has been herein proposed. All the methods have been implemented in the MATLAB computing environment and CPU times required for a simple simulation have been recorded.*

## 1. Introduction

Multibody dynamics formalisms can be broadly splitted into two categories according to the set of coordinates used. Formalisms based on *redundant set of coordinates* are often used in commercial software such as MSC ADAMS or LMS Virtual Lab Motion. Formalisms based on a *minimum number of coordinates* are less used, but usually preferred in real-time computations.

Recursive dynamics appears an computationally efficient alternative

In this paper, the problem of coordinate reduction will be discussed. Mainly orthogonalization methods will be presented. In particular, the following numerical methods for coordinate reduction of state space dynamic formulations will be reviewed:

- Coordinate partitioning method;
- zero-eigenvalue method
- SVD decomposition;
- QR decomposition;
- Udwadia-Kalaba formulation
- PUTD method
- Schur decomposition

This last method is also herein proposed and compared with other approaches.

Recursive formulations (e.g. [6, 4]) are not included in this review.

Since numerical implementation and computational efficiency issues are not always fully addressed in review papers on constraints orthogonalization, the paper reports the MATLAB code and CPU time.

## 2. Nomenclature

- $I_k$ : Moment of inertia of the  $k^{th}$  gear;
- $F$ : degrees of freedom of the mechanical system;
- $[M]$ , mass matrix;
- $m$  number of independent constraint equations
- $n$  number of coordinates
- $\{q\}$ , vector of generalized coordinates;
- $\{Q\}$ , vector of external forces;
- $r$ , rank of the Jacobian matrix;
- $t$ , time;
- $\{u\}$ , vector of dependent generalized coordinates;
- $\{v\}$ , vector of independent generalized coordinates;
- $\{\lambda\}$ , vector of Lagrange's multipliers;
- $\{\Psi\}$ , vector of position constraints;
- $[\Psi_q]$ , Jacobian matrix of constraints;

- $\omega$ , angular velocity;
- $\text{chol}(\cdot)$  upper triangular matrix resulting from Cholesky factorization;
- dots denote differentiation with respect to time.

### 3. Theoretical bases

The dynamics equations deduced with a redundant number of coordinates the following of algebraic-differential equations (DAE) is obtained

$$[M] \{\ddot{q}\} + [\Psi_q]^T \{\lambda\} = \{Q\} \quad (1a)$$

$$\{\Psi(q, t)\} = \{0\} . \quad (1b)$$

The presence of actively controlled components may also introduce differential constraints and thus require DAE for mathematical modeling.

The total number of equations is  $n+m$  with  $n+m$  unknowns, *i.e.* the  $n$  generalized coordinates and  $m$  Lagrange's multipliers related to the constraint equations.

A common strategy for solving (1) is the differentiation, with respect to time, of position constraint equations. Hence (1) is transformed into the following DAE system of differential index 1:

$$\begin{bmatrix} M & \Psi_q^T \\ \Psi_q & 0 \end{bmatrix} \begin{Bmatrix} \ddot{q} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} Q \\ \gamma \end{Bmatrix} \quad (2)$$

where  $\{\gamma\} = -([\Psi_q] \{\dot{q}\})_q \{\dot{q}\} - 2[\Psi_{qt}] \{\dot{q}\} - \{\Psi_{tt}\}$ . The (2), assuming full rank Jacobian, can be solved into two steps:

For the purpose of block solution, the mass matrix is first decomposed such that  $[M] = [M]^{1/2} [M]^{1/2}$ , then (2) is rewritten as follows [10]

$$\begin{bmatrix} L & 0 \\ H^T & -L_1^T \end{bmatrix} \begin{bmatrix} L^T & H \\ 0 & L_1 \end{bmatrix} \begin{Bmatrix} \ddot{q} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} Q \\ \gamma \end{Bmatrix} \quad (3)$$

where

$$[L] = [M]^{1/2} , \quad (4)$$

$$[H] = [L^{-1}]^T [\Psi_q]^T , \quad (5)$$

$$[L_1] = \text{chol}([H]^T [H]) . \quad (6)$$

Thus, it can be demonstrated that [24]

$$\{\lambda\} = [L]^{-1} \{\lambda_1\} \quad (7a)$$

with

$$\{\lambda_1\} = [L_1]^{-1} ([H]^T [L]^{-1} \{Q\} - \{\gamma\}) . \quad (7b)$$

Since  $[L_1]$  is an upper triangular matrix, the use of specialized procedures (*e.g.* DTRDI of LAPACK) is recommended for its inversion.

Afterwards, the generalized accelerations vector is computed by means of

$$\{\ddot{q}\} = [L^{-1}]^T ([L]^{-1} \{Q\} - [H] \{\lambda\}) \quad (8)$$

It must be stressed that, for a reliable numerical integration procedure, the initial conditions must satisfy the position and velocity constraints [19].

### 4. Baumgarte constraint stabilization

Since after numerical integration of Eqs. (2) position and velocity constraints fail to be satisfied, the right side of the acceleration constraint is altered as follows

$$\{\bar{\gamma}\} = \{\gamma\} - 2\alpha \{\Psi\} - \beta \{\dot{\Psi}\} \quad (9)$$

where  $\alpha$  and  $\beta$  are the gain parameters usually chosen such that  $\alpha = \beta$ . The DAE system to be integrated is thus transformed to

$$\begin{bmatrix} M & \Psi_q^T \\ \Psi_q & 0 \end{bmatrix} \begin{Bmatrix} \ddot{q} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} Q \\ \bar{\gamma} \end{Bmatrix} \quad (10)$$

### 5. Orthogonalization of constraints

In order to reduce the DAE system to an ordinary differential equations (ODE) the elimination of the Jacobian matrix  $\Psi_q$  of the constraint equations from (1a) is necessary. This approach offers the following advantages:

- The elimination of Lagrange's multipliers when solving equations:
- The possibility to partition the entire set of generalized coordinates into independent variables and dependent ones;
- The transform of the DAE system into a ODE gives the opportunity of a wider choice of numerical integration subroutines;
- Mechanical systems with a redundant number of constraints or with changing d.o.f. can be analysed.

In this section the elimination of Lagrange multipliers through the orthogonalization of constraints will be briefly discussed. This approach dates back to G.A. Maggi [17, 9, 16, 7].

For our purposes a minimum set  $v$  of  $F$  independent coordinates is defined. Let us append to the constraint vector  $\{\Psi\}$  the equations  $\{\Phi\}$  that can be established between  $v$  and  $q$ . Thus, we obtain

$$\{\Gamma\}_{(m+F) \times 1} = \begin{Bmatrix} \Psi(q) \\ \Phi(v, q) \end{Bmatrix} = 0 \quad (11)$$

The time derivative of (11) leads to

$$[\Gamma_v]_{(m+F) \times F} \{\dot{v}\} + [\Gamma_q]_{(m+F) \times n} \{\dot{q}\} = 0 \quad (12)$$

Assuming  $[\Gamma_q]$  non singular and introduced the matrix

$$[V]_{n \times F} = -[\Gamma_q]^{-1} [\Gamma_v] \quad (13)$$

one obtains

$$\{\dot{q}\} = [V] \{\dot{v}\} \quad (14)$$

When there is not any explicit dependence on time of constraints equations, the following *orthogonality condition* is deduced

$$[\Psi_q][V] = 0 \quad (15)$$

and the accelerations  $\ddot{q}$  can be expressed in the form

$$\{\ddot{q}\} = [V] \{\ddot{v}\} + [\dot{V}] \{\dot{v}\} \quad (16)$$

Premultiplying both sides of the dynamic equation of system (1a) and taking into account (15) and (16), the vector of Lagrange's multipliers is eliminated from the differential equations of equilibrium and the following ODE is obtained

$$[V]^T [M] [V] \{\ddot{v}\} = [V]^T \{Q\} + [V]^T [M] [\dot{V}] \{\dot{v}\} \quad (17)$$

The matrix  $[V]$  is not unique, the main purpose of this paper is to illustrate the different ways of obtaining it.

The set of coordinates  $q \in \mathbb{R}^n$  is partitioned into two sets  $v \in \mathbb{R}^F$  and  $u \in \mathbb{R}^m$  of independent and dependent coordinates, respectively

$$\{q\} = \begin{Bmatrix} u \\ v \end{Bmatrix} \quad (18)$$

where  $\{u\}_{(n-F) \times 1}$  is the vector of dependent coordinates and  $\{v\}_{F \times 1}$  is the one of independent coordinates. The acceleration constraints  $[\Psi_q] \{\ddot{q}\} = \{\gamma\}$ , through row operations (*e.g.* Gaussian elimination), can be rearranged in the form

$$\begin{bmatrix} [\Psi_u] & [\Psi_v] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{v}\} \end{Bmatrix} = \begin{Bmatrix} \{\gamma\} \\ \{0\} \end{Bmatrix} \quad (19)$$

or

$$[\Psi_u]_{m \times (n-F)} \{\ddot{u}\} + [\Psi_v]_{m \times F} \{\ddot{v}\} = \{\gamma\} \quad (20)$$

where  $[\Psi_u]$  is a nonsingular square triangular matrix with unit values on the diagonal.

From the previous expression a relationship between  $\ddot{q}$  and  $\ddot{v}$  follows

$$\{\ddot{q}\} = \begin{Bmatrix} \ddot{u} \\ \ddot{v} \end{Bmatrix} = \begin{bmatrix} -[\Psi_u]^{-1} [\Psi_v] \\ [I] \end{bmatrix} \{\ddot{v}\} + \begin{Bmatrix} [\Psi_u]^{-1} \{\gamma\} \\ \{0\} \end{Bmatrix} \quad (21)$$

By comparing (21) and (17) one can deduce

$$[V]_{n \times F} = \begin{bmatrix} -[\Psi_u]^{-1} [\Psi_v] \\ [I] \end{bmatrix} \quad (22)$$

and

$$[\dot{V}] \{\dot{v}\} = \begin{Bmatrix} [\Psi_u]^{-1} \{\gamma\} \\ \{0\} \end{Bmatrix} \quad (23)$$

Now it is possible to define the matrix  $[V]$  in the presence of rheonomic constraints. In this case, which is common in applications, the expression of the constraint vector is  $\{\Psi(q, t)\} = 0$  and the expression of velocities changes into

$$[\Psi_q] \{\dot{q}\} = -\{\Psi_t\} \quad (24)$$

Let us assume the correctness of the expression

$$\{\dot{v}\} = [B]_{F \times n} \{\dot{q}\} \quad (25)$$

Because of singularities in the configuration of the mechanical system, the d.o.f. of the system may not be constant during the numerical integration process. Moreover, the matrix  $[B]$  has  $F$  rows independent from the  $m$  rows of the Jacobian matrix.

As a consequence, combining (24) and (25), the following expression is obtained

$$\begin{bmatrix} [\Psi_q] \\ [B] \end{bmatrix}_{(m+F) \times n} \{\dot{q}\} = \begin{Bmatrix} -\{\Psi_t\} \\ \{\dot{v}\} \end{Bmatrix} \quad (26)$$

where the left hand side matrix is not singular. The solution of the previous system gives

$$\{\dot{q}\} = \begin{bmatrix} [\Psi_q] \\ [B] \end{bmatrix}^{-1} \begin{Bmatrix} -\{\Psi_t\} \\ \{\dot{v}\} \end{Bmatrix} = -[S]_{n \times m} \{\Psi_t\} + [V] \{\dot{v}\} \quad (27)$$

where matrix  $[S]$  is formed from the first  $m$  columns of matrix  $\begin{bmatrix} [\Psi_q] \\ [B] \end{bmatrix}^{-1}$ . Equation (27) contains, as particular case, the expression (14).

By solving the system (26) the following expression is obtained

$$\begin{aligned} \begin{bmatrix} [\Psi_q] \\ [B] \end{bmatrix} \begin{bmatrix} [\Psi_q] \\ [B] \end{bmatrix}^{-1} &= \begin{bmatrix} [\Psi_q] \\ [B] \end{bmatrix} \begin{bmatrix} [S] & [V] \end{bmatrix} \\ &= \begin{bmatrix} [\Psi_q][S] & [\Psi_q][V] \\ [B][S] & [B][V] \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \end{aligned} \quad (28)$$

which confirms the property of orthogonality also in the case of rheonomic constraints:

$$[\Psi_q][V] = [0] \quad (29)$$

and introduce the new one

$$[B][V] = [I] \quad (30)$$

By differentiating (26) with respect to time the following equation is obtained

$$\begin{bmatrix} [\Psi_q] \\ [B] \end{bmatrix} \{\ddot{q}\} = \begin{Bmatrix} \{\gamma\} \\ \{\ddot{v}\} \end{Bmatrix} \quad (31)$$

which express the relationship between  $\{\ddot{v}\}$  and  $\{\ddot{q}\}$ . as

$$\{\ddot{q}\} = [S] \{\gamma\} + [V] \{\ddot{v}\} \quad (32)$$

The term  $[S] \{\gamma\}$  is obtained from (31) assuming  $\ddot{v} = 0$  as follows

$$[S] \{\gamma\} = \begin{bmatrix} [\Psi_q] \\ [B] \end{bmatrix}^{-1} \begin{Bmatrix} \{\gamma\} \\ \{0\} \end{Bmatrix} \quad (33)$$

## 6. An application of constraints orthogonalization to the stability analysis of gear trains

Some type of analysis do not require the explicit computation of meshing forces.

For instance, a model for the forward dynamic analysis of a two d.o.f. epicyclic gear train (EGT) can be described by the following differential equations system

$$[\mathcal{M}] \begin{Bmatrix} \dot{\omega}_{d1} \\ \dot{\omega}_{d2} \end{Bmatrix} = \begin{Bmatrix} T_{d1} - aT_r \\ T_{d2} - bT_r \end{Bmatrix}, \quad (34)$$

where

- $[\mathcal{M}]$  is reduced mass matrix,
- $T_{d1}, T_{d2}$  and  $T_r$  are respectively the driving and resisting torques,
- $\dot{\omega}_{d1}, \dot{\omega}_{d2}$  are the absolute angular accelerations at the driving shafts,
- $a$  and  $b$  are constants function of gear ratios.

Under certain working conditions, it can be shown that the behavior of the system is not stable [23].

For this purpose, let us assume that the motor speed-torques curves are linearized by means of the following equations

$$T_{d1} = T_{d1}^o + K_{d1} (\omega_{d1} - \omega_{d1}^o) , \quad (35)$$

$$T_{d2} = T_{d2}^o + K_{d2} (\omega_{d2} - \omega_{d2}^o) , , \quad (36)$$

where the upperscript  $^o$  refers to the values of variables at steady state condition. Similarly, the resisting torque-speed curve can be also described by means of the linear equation

$$T_r = T_r^o + K_r (\omega_r - \omega_r^o) , \quad (37)$$

where  $\omega_r$  denotes the angular speed of the output shaft. When (35), (36) and (37) are substituted into (34), after taking into account the existing linear relationship between  $\omega_r$  and the input angular velocities  $\omega_{d1}$  and  $\omega_{d2}$  of the two driving shafts, with obvious nomenclature, one obtains:

$$[\mathcal{M}] \begin{Bmatrix} \Delta \dot{\omega}_{d1} \\ \Delta \dot{\omega}_{d2} \end{Bmatrix} + [\mathcal{K}] \begin{Bmatrix} \Delta \omega_{d1} \\ \Delta \omega_{d2} \end{Bmatrix} = \begin{Bmatrix} T_{d1}^o - aT_r^o \\ T_{d2}^o - bT_r^o \end{Bmatrix} , \quad (38)$$

with  $[\mathcal{K}]$  symmetric matrix. The solution of the homogeneous o.d.e. associated with (38) will have the following algebraic form

$$\begin{Bmatrix} \Delta \omega_{d1} \\ \Delta \omega_{d2} \end{Bmatrix} = \{X\} e^{\lambda t} , \quad (39)$$

where  $\lambda$  is the solution of the eigenvalue problem

$$[\lambda \mathcal{M} + \mathcal{K}] \{X\} = \{0\} . \quad (40)$$

It is self evident that positive values of  $\lambda$ 's will reveal an unstable behavior of the system. Since this type of analysis requires a preliminary computation of  $[\mathcal{M}]$  and  $[\mathcal{K}]$ , in the section will be described a procedure for the computation of such matrices.

Let

$$\begin{aligned} \Phi_1 &\equiv \dot{v}_1 - \omega_{d1} = 0 , \\ \Phi_2 &\equiv \dot{v}_2 - \omega_{d2} = 0 , \end{aligned}$$

and

$$\Gamma (\omega, \dot{v}) = \begin{Bmatrix} \{\psi (\omega)\} \\ \{\phi (\omega, \dot{v})\} \end{Bmatrix} = \{0\} \quad (41)$$

the vector of constraints.

Once the matrix

$$[V] = -[\Gamma_\omega]^{-1} [\Gamma_{\dot{v}}] , \quad (42)$$

is introduced, the following orthogonality relationship

$$[\Psi_\omega] [V] = [0] \quad (43)$$

can be demonstrated [10, 20].

Making use of this condition one obtains

$$[\mathcal{M}] = [V]^T [M] [V] \quad (44)$$

and

$$[\mathcal{K}] = [V]^T \{T\} , \quad (45)$$

where

$$\{T\} = \begin{Bmatrix} \dots \\ K_{d1} (\omega_{d1} - \omega_{d1}^o) \\ \dots \\ K_{d2} (\omega_{d2} - \omega_{d2}^o) \\ \dots \\ K_r (\omega_r - \omega_r^o) \\ \dots \end{Bmatrix} \quad (46)$$

is the vector of externally applied torques <sup>1</sup>

<sup>1</sup>At this stage of the analysis procedure the values of  $T_{d1}^o$ ,  $T_{d2}^o$  and  $T_r^o$  can be omitted.

## 6.1 Application of the theory to a practical case

Let us assume that is required to deduce the matrices  $[\mathcal{M}]$  and  $[\mathcal{K}]$  for the 2 d.o.f. gear train shown in Figure 1. This EGT has been developed at the Ames Research Center, Moffet Field (CA) has been studied by Pennestrì [21, 22] regarding power-flow analysis, by Pennestrì and Valentini for stability analysis [23] and, more recently, by Popa, Pandrea and Stănescu [25] for dynamic analysis. From Table 1 one obtains the following system of equations

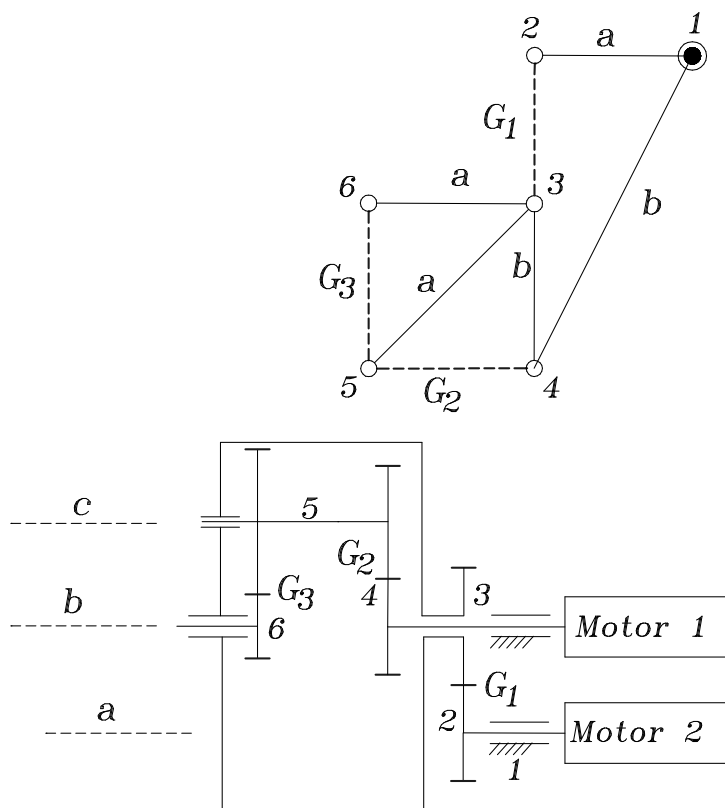


Figure 1: Graph and functional representation of a 2 d.o.f. rotary drive mechanism.

Fundamental circuit	Gear $i$	Gear $j$	Transfer vertex $k$	Gear ratio $\tau$	Turning pairs level
#1	2	3	1	$\tau_1$	$a, b$
#2	5	4	3	$\tau_2$	$b, c$
#3	5	6	3	$\tau_3$	$a, c$

Table 1: Fundamental circuits look-up table (see Figure 1)

$$\{\Psi\} \equiv [\Psi_\omega] \{\omega\} = \begin{bmatrix} 1 & -\tau_1 & 0 & 0 & 0 \\ 0 & \tau_2 - 1 & -\tau_2 & 1 & 0 \\ 0 & \tau_3 - 1 & 0 & 1 & -\tau_3 \end{bmatrix} \begin{Bmatrix} \omega_2 \\ \omega_3 \\ \omega_4 \\ \omega_5 \\ \omega_6 \end{Bmatrix} = \{0\} .$$

Solving this set of equations w.r.t.  $\omega_3$ ,  $\omega_5$  and  $\omega_6$  one obtains

$$\begin{aligned} \omega_3 &= \frac{\omega_2}{\tau_1} , \\ \omega_5 &= \frac{1 - \tau_2}{\tau_1} \omega_2 + \tau_2 \omega_4 , \\ \omega_6 &= \frac{\tau_3 - \tau_2}{\tau_1 \tau_3} \omega_2 + \frac{\tau_2}{\tau_3} \omega_4 . \end{aligned}$$

Since

$$\begin{aligned}\Phi_1 &\equiv \dot{v}_1 - \omega_2 = 0, \\ \Phi_2 &\equiv \dot{v}_2 - \omega_4 = 0,\end{aligned}$$

from (42) the matrix

$$[V] = \begin{bmatrix} 1 & 0 \\ \frac{1}{\tau_1} & 0 \\ 0 & 1 \\ \frac{1-\tau_2}{\tau_1} & \tau_2 \\ \frac{\tau_3-\tau_2}{\tau_1\tau_3} & \frac{\tau_2}{\tau_3} \end{bmatrix}$$

is deduced.

Finally, from the application of (44) and (45) one obtains, respectively,

$$\begin{aligned}\mathcal{M}_{11} &= I_2 + \frac{I_3 + m_5 \delta_{bc}^2}{\tau_1^2} + \left(\frac{\tau_2 - 1}{\tau_1}\right)^2 I_5 + \left(\frac{\tau_2 - \tau_3}{\tau_1 \tau_3}\right)^2 I_6, \\ \mathcal{M}_{12} = \mathcal{M}_{21} &= \frac{(1 - \tau_2) \tau_2}{\tau_1} I_5 + \frac{(\tau_3 - \tau_2) \tau_2}{\tau_1 \tau_3^2} I_6, \\ \mathcal{M}_{22} &= I_4 + I_5 \tau_2^2 + \frac{\tau_2^2}{\tau_3^2} I_6,\end{aligned}$$

and

$$\begin{aligned}\mathcal{K}_{11} &= K_{d1} + \left(\frac{\tau_2 - \tau_3}{\tau_1 \tau_3}\right)^2 K_r, \\ \mathcal{K}_{12} = \mathcal{K}_{21} &= \frac{\tau_2 (\tau_3 - \tau_2)}{\tau_1 \tau_3^2} K_r, \\ \mathcal{K}_{22} &= K_{d2} + \frac{\tau_2^2}{\tau_3^2} K_r.\end{aligned}$$

An application of the Maggi's like method to the dynamic analysis of the scotch-yoke mechanism is given by Vlase [29].

## 7. Numerical methods for elimination of Lagrange's multipliers

In the example just discussed the matrix  $[V]$  was obtained by hand computation. The methods reported in this section are suitable for implementation in a software of automatic multibody dynamic analysis (see Appendix).

### 7.2 Coordinate partitioning

In multibody dynamics the coordinate partitioning method is an historically important computational scheme. The scheme received extensive application in multibody dynamics by Haug and his coworkers [31, 12].

By definition, the sub-Jacobian  $[\Psi_u]$  is non singular. Based on this partitioning and the DAE equations can be rewritten in the form<sup>2</sup>

$$[M^{vv}] \{\ddot{v}\} + [M^{vu}] \{\ddot{u}\} + [\Psi_v]^T \{\lambda\} = \{Q^v\} \quad (47)$$

$$[M^{uv}] \{\ddot{v}\} + [M^{uu}] \{\ddot{u}\} + [\Psi_u]^T \{\lambda\} = \{Q^u\} \quad (48)$$

$$\{\Psi(u, v)\} = 0 \quad (49)$$

The non singularity of  $[\Psi_u]$  and the implicit function theorem guarantee that  $\{u\}$  can be locally computed as a function of  $\{v\}$ , i.e.

$$\{u\} = \{h(v)\} \quad (50)$$

The velocity

$$[\Psi_q] \{\dot{q}\} = -\{\Psi_t\} \quad (51)$$

and acceleration

$$[\Psi_q] \{\ddot{q}\} = \{\gamma\} \quad (52)$$

<sup>2</sup>It is assumed that the Jacobian  $[\Psi_q]^T$  has full row rank.

constraints can respectively rearranged in the form

$$[\Psi_u] \{\dot{u}\} + [\Psi_v] \{\dot{v}\} = - \{\hat{\Psi}_t\} \quad (53)$$

$$[\Psi_u] \{\ddot{u}\} + [\Psi_v] \{\ddot{v}\} = \{\gamma\} \quad (54)$$

Eliminating  $\lambda$  and  $\ddot{u}$  from (47), (48) and (54) one obtains [12]

$$[\hat{M}] \{\ddot{v}\} = \{\hat{Q}\} \quad (55)$$

where

$$\begin{aligned} [\hat{M}] &= [M^{vv}] - [M^{vu}] [\Psi_u]^{-1} [\Psi_v] \\ &\quad - [\Psi_v]^T \left( [\Psi_u]^{-1} \right)^T \left( [M^{uv}] - [M^{uu}] [\Psi_u]^{-1} [\Psi_v] \right) \end{aligned} \quad (56)$$

and

$$\begin{aligned} \{\hat{Q}\} &= \{Q^v\} - [M^{vu}] [\Psi_u]^{-1} \{\gamma\} \\ &\quad - [\Psi_v]^T \left( [\Psi_u]^{-1} \right)^T \left( \{Q^u\} - [M^{uu}] [\Psi_u]^{-1} \{\gamma\} \right) \end{aligned} \quad (57)$$

The solution of the DAE system is therefore reduced to a set of ODE system through the sequence of steps listed below

1. Partition the vector  $\{q\}$  of coordinates;
2. Determine  $\{\dot{u}\}$  and  $\{u\}$  at time  $t$  by means of Eqs. (53) and (50), respectively;
3. Solve Eqs. (47), (48), (49) w.r.t.  $\{\ddot{u}\}$ ,  $\{\ddot{v}\}$  and  $\{\lambda\}$ ;
4. Integrate and compute  $\{u\}$ ,  $\{v\}$ ,  $\{\dot{u}\}$ ,  $\{\dot{v}\}$  at time  $t + \Delta t$

A critical review of different dynamic formulations is offered in [28].

### 7.3 QR decomposition

According to this method, the Jacobian matrix  $[\Psi_q]^T$  is first rearranged as follows [15]:

$$[\Psi_q]^T = \begin{bmatrix} [Q_1]_{n \times (n-F)} & [Q_2]_{n \times F} \end{bmatrix} \begin{bmatrix} [R_1]_{r \times m} \\ [0]_{F \times m} \end{bmatrix} = [Q_1] [R_1] \quad (58)$$

where  $[Q_1]$  and  $[Q_2]$  simultaneously satisfy the conditions

$$[Q_2]^T [Q_1] = [0] \quad (59)$$

$$[Q_2]^T [Q_2] = [I] \quad (60)$$

The first condition is used to eliminate the Lagrange's multipliers in the equations of motion. In fact, considering equation (58) and (59), the following one is obtained

$$[Q_2]^T [\Psi_q]^T = [Q_2]^T [Q_1] [R_1] = [0] \quad (61)$$

Comparing (61) and (60) with (29) and (30), the matrices  $[B]$  and  $[V]$  are respectively obtained

$$[V] = [Q_2] \quad (62)$$

$$[B] = [Q_2]^T \quad (63)$$

Hence, the equations of motion are rearranged as follows:

$$[Q_2]^T [M] [Q_2] \{\ddot{v}\} = [Q_2]^T [Q] - [Q_2]^T [M] [S] \{\gamma\} \quad (64)$$

where

$$[S] \{\gamma\} = \begin{bmatrix} [\Psi_q] \\ [Q_2]^T \end{bmatrix}^{-1} \begin{Bmatrix} \{\gamma\} \\ \{0\} \end{Bmatrix} \quad (65)$$

The pseudo upper triangular decomposition (PUTD) method, developed by Amirouche et al. [2, 1], is somewhat equivalent to the method based on the QR decomposition.

In particular, the method's first step is to obtain the Householder transform matrix  $[H]$  of  $[\Psi_q]^T$ . Then, through the Gram-Schmidt process, it is identified a matrix  $[D_2]$  such that

$$[D_2]_{(n-r) \times n} [H]_{n \times n}^T [\Psi_q]^T = [0] \quad (66)$$

The matrix  $[D_2]$  has the following block structure

$$[D_2] = \begin{bmatrix} [0]_{(n-r) \times r} & [I]_{(n-r) \times (n-r)} \end{bmatrix} \quad (67)$$

One can verify that  $[D_2] [H]^T = [Q_2]^T$ .

#### 7.4 The Zero Eigenvalue method

The zero-eigenvalue theorem provides an alternative tool for the orthogonalization of the Jacobian matrix. This method has been originally proposed by Walton and Steeves [30] and later extended by Kamman and Huston [14].

Let us form the matrix

$$[H] = [\Psi_q]^T [\Psi_q] \quad (68)$$

Since  $[H]_{n \times n}$  is symmetric, all the eigenvalues are positive and there exists the similarity transform

$$[T]^T [H] [T] = [\Lambda] \quad (69)$$

where  $[T]$  is the orthogonal matrix of eigenvectors and  $[\Lambda]$  the diagonal matrix of eigenvalues.

The matrix  $[T]$  can be partitioned as follows

$$[T] = \begin{bmatrix} [T_1]_{n \times (n-r)} & [T_2]_{n \times r} \end{bmatrix} \quad (70)$$

where  $[T_1]$  are the eigenvectors associated to zero eigenvalues.

It can be easily verified the condition of orthogonality

$$[\Psi_q] [T_1] = [0] \quad (71)$$

#### 7.5 Singular value decomposition

Matrix  $[V]$  is now deduced by means of the Singular Value Decomposition (SVD) algorithm. Therefore, matrix  $[\Psi_q]^T$  has been rearranged as follows [26, 18]

$$\begin{aligned} [\Psi_q]^T &= \begin{bmatrix} [W_d]_{n \times r} & [W_i]_{n \times (n-r)} \end{bmatrix} \begin{bmatrix} [\Lambda_1]_{r \times r} \\ [0]_{(n-r) \times r} \end{bmatrix} [U]_{r \times r}^T \\ &= [W_d] [\Lambda_1] [U]^T \end{aligned} \quad (72)$$

The columns of the matrix  $\begin{bmatrix} [W_d] & [W_i] \end{bmatrix}$  are orthogonal, *i.e.*  $[W_i]^T [W_d] = [0]$  and  $[W_i]^T [W_i] = [I]$ . Therefore, combining this with (72) the following expression is obtained

$$[W_i]^T [\Psi_q]^T = [W_i]^T [W_d] [\Lambda_1] [U]^T = [0] \quad (73)$$

Matrix  $[V]$  is then defined as

$$[V] = [W_i] \quad (74)$$

and, as consequence, the following equalities are established

$$\{\dot{q}\} = [W_i] \{\dot{v}\} \quad (75)$$

and

$$[S] \{\gamma\} = \begin{bmatrix} [\Psi_q] \\ [W_i]^T \end{bmatrix}^{-1} \begin{Bmatrix} \{\gamma\} \\ \{0\} \end{Bmatrix} \quad (76)$$

The final expression of the equations of motion is

$$[W_i]^T [M] [W_i] \{\ddot{v}\} = [W_i]^T [Q] - [W_i]^T [M] [S] \{\gamma\} \quad (77)$$

## 7.6 The Schur decomposition

The Schur decomposition offers an alternative tool for computing matrix  $[V]$ .

The Schur decomposition of the square matrix  $[H]$  has the form [11]

$$[K]^T [H] [K] = [U] \quad (78)$$

where  $[K]$  is a unitary matrix and  $[U]$  an upper triangular matrix. The eigenvalues of  $[H]$  are on the diagonal of  $[U]$ .

The matrix  $[K]$  can be partitioned as follows

$$[K] = \begin{bmatrix} [K_1]_{n \times (n-r)} & [K_2]_{n \times r} \end{bmatrix} \quad (79)$$

where  $[K_1]$  are the generalized Schur vectors associated to zero eigenvalues. If we let  $[H] = [\Psi_q]^T [\Psi_q]$ , then the following condition of orthogonality holds

$$[\Psi_q] [K_1] = [0] . \quad (80)$$

## 7.7 The Udwadia-Kalaba formulation

The square matrix at the left-hand can be inverted by block partitioning

$$\begin{bmatrix} M & \Psi_q^T \\ \Psi_q & 0 \end{bmatrix}^{-1} = \begin{bmatrix} M^{-1} - M^{-1} \Psi_q^T (\Psi_q M^{-1} \Psi_q^T)^{-1} \Psi_q M^{-1} & M^{-1} \Psi_q^T (\Psi_q M^{-1} \Psi_q^T)^{-1} \\ (\Psi_q M^{-1} \Psi_q^T)^{-1} \Psi_q M^{-1} & - (\Psi_q M^{-1} \Psi_q^T)^{-1} \end{bmatrix} . \quad (81)$$

Introduced the vector

$$\{\ddot{q}_f\} = [M]^{-1} \{Q\} , \quad (82)$$

representing the acceleration vector of the unconstrained system, the solution of (2) follows

$$\{\ddot{q}\} = \{\ddot{q}_f\} + [M]^{-1} [\Psi_q]^T \left( [\Psi_q] [M]^{-1} [\Psi_q]^T \right)^{-1} (\{\gamma\} - [\Psi_q] \{\ddot{q}_f\}) \quad (83)$$

and

$$\{\lambda\} = \left( [\Psi_q] [M]^{-1} [\Psi_q]^T \right)^{-1} (\{\gamma\} - [\Psi_q] \{\ddot{q}_f\}) . \quad (84)$$

If we let

$$[M]^{-1} = [M]^{-\frac{1}{2}} [M]^{-\frac{1}{2}} , \quad (85)$$

$$[D] = [\Psi_q] [M]^{-\frac{1}{2}} , \quad (86)$$

then equation (83) can be rewritten in the form

$$\begin{aligned} \{\ddot{q}\} &= \{\ddot{q}_f\} + [M]^{-\frac{1}{2}} \left( [M]^{-\frac{1}{2}} [\Psi_q]^T \right) \left( [\Psi_q] [M]^{-\frac{1}{2}} [M]^{-\frac{1}{2}} [\Psi_q]^T \right)^{-1} (\{\gamma\} - [\Psi_q] \{\ddot{q}_f\}) , \\ &= \{\ddot{q}_f\} + [M]^{-\frac{1}{2}} [D]^T \left( [D] [D]^T \right)^{-1} (\{\gamma\} - [\Psi_q] \{\ddot{q}_f\}) \end{aligned} \quad (87)$$

Taken into account the definition of right pseudoinverse, we can let

$$[D]^+ = [D]^T \left( [D] [D]^T \right)^{-1} , \quad (88)$$

and the previous equation can be concisely expressed as follows [5]:

$$\{\ddot{q}\} = \{\ddot{q}_f\} + [M]^{-\frac{1}{2}} [D]^+ (\{\gamma\} - [\Psi_q] \{\ddot{q}_f\}) . \quad (89)$$

This is the dynamic formulation originally proposed by Udwadia and Kalaba [27] starting from Gauss Principle of Least Constraint.

If the Baumgarte stabilization is introduced [8, 13], then (89) is modified as follows

$$\{\ddot{q}\} = \{\ddot{q}_f\} + [M]^{-\frac{1}{2}} [D]^+ \left( \{\gamma\} - 2\alpha \left\{ \dot{\Psi} \right\} - \beta^2 \{\Psi\} - [\Psi_q] \{\ddot{q}_f\} \right) \quad (90)$$

## 8. Numerical example

The computational efficiency of methods herein discussed has been numerically tested. The MATLAB code enclosed in the Appendix has been used for such purpose. Standard math libraries have been used for all algebra operations [3].

The numerical example is the simulation of a simple pendulum (See Figure 2). Mass and Jacobian matrices,  $\gamma$  and  $Q$  vectors have the following algebraic form:

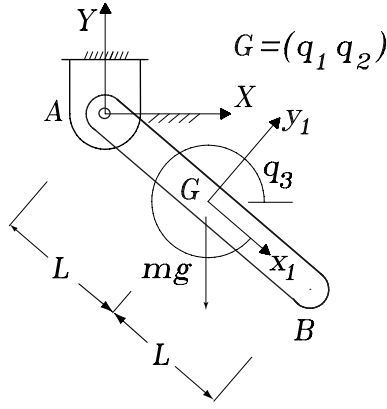


Figure 2:

- Mass matrix:  $[M] = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_g \end{bmatrix}$
- Jacobian matrix:  $[\Psi_q] = \begin{bmatrix} 1 & 0 & L \sin q_3 \\ 0 & 1 & -L \cos q_3 \end{bmatrix}$
- $\{\gamma\} = \{ -L\dot{q}_3^2 \cos q_3 \quad -L\dot{q}_3^2 \sin q_3 \}^T$
- $\{Q\} = \{ 0 \quad -9.8m \quad 0 \}^T$

The prescribed values are  $L = 0.5$  and  $I_g = 0.1$ . Simulation time is 10 s.

The null space matrix  $[V]$  is computed at each time integration step. The number of floating point operations (flops) for the computation of matrix  $[V]$  is currently of  $O(n^3)$ .

In order to simplify computer programming the value  $r$  of the Jacobian matrix has been always preliminary computed by means of the Matlab function `rank`. This could be avoided since the rank can be estimated making use of the results of Jacobian matrix decomposition.

In an effort to reduce CPU time some researchers hinted the possibility of delay in the computation of  $[V]$  by introducing various criteria. Bachau [7] defined such methods as *Maggi-like methods*.

The CPU times required by each method have been summarized in Table 2.

Table 2: Computational efficiency of the constraints orthogonalization methods.

Method	CPU Time (s)
Coordinate partitioning	11.45
QR	2.64
PUTD	4.83
SVD	2.73
Zero Eigenvalue	2.72
Schur	2.74
Udwadia-Kalaba	3.17

Finally, it has been observed that the numerical accuracy is not influenced by the method used for the orthogonalization. This statement may not be true when dealing with different and large problems.

The CPU times have been recorded after running the simulations under Windows XP operating system (AMD Athlon 1.92 GHz). Each simulation has been executed ten times and the lowest time recorded.

## 9. Conclusions

The computational efficiency of methods for constraints orthogonalization has been compared. The QR decomposition ranked best among the tested methods. The Schur decomposition has been herein proposed for the numerical computation of the null space of the Jacobian matrix. There is a subtle difference between the Zero-Eigenvalue method and Schur decomposition. The similarity transformation (69) may not exist, whereas Schur decomposition can be always obtained for any square matrix.

Comparable computational efficiencies have been observed for the SVD, Zero Eigenvalue and Schur decompositions. The CPU time herein reported could be reduced further through a more skilled MATLAB programming. However, all linear algebra operations have been carried out making use of library routines.

## References

- [1] F. Amirouche. *Fundamentals of Multibody Dynamics*. Birkhäuser, 2004.
- [2] F.M.L. Amirouche and S.K. Ider. Coordinate reduction in the dynamics of constrained multibody systems a new approach. *ASME Journal of Applied mechanics*, 55:1988, 1988.
- [3] E. et al. Anderson. *LAPACK - Users's Guide*. Society for Industrial and Applied Mathematics, Philadelphia, 1995.
- [4] J.H. Anderson, K.S. and Critchley. Improved "Order-N" Performance Algorithm for the Simulation Constrained Multi-Rigid-Body Dynamic Systems. *Multibody System Dynamics*, 9:185–212, 2003.
- [5] A. Arabyan and F. Wu. An Improved Formulation for Constrained Mechanical Systems. *Multibody System Dynamics*, 2:49–69, 1998.
- [6] E.J. Bae, D. and Haug. A Recursive Formulation of Constrained Mechanical Systems Dynamics: Part II, Closed Loop Systems. *Mechanisms, Structures, and Machines*, 15(4):481–506, 1987.
- [7] A. Bauchau, A.O. and Laulusa. Review of Classical Approaches for Constraint Enforcement in Multibody Systems. *Journal of Computational and Non Linear Dynamics*, 2007.
- [8] J.W. Baumgarte. Stabilization of Constraints and Integrals of Motion in Dynamical Systems. *Computer Methods in Applied Mechanics and Engineering*, 1:1–16, 1972.
- [9] M. Borri, C.L. Bottasso, and P. Mantegazza. Equivalence of Kane's and Maggi's Equations. *Meccanica*, 25:272–274, 1990.
- [10] E. Eich-Soellner and C. Führer. *Numerical Methods in Multibody Dynamics*. B.G. Teubner, 1998.
- [11] G. Golub and C. van Loan. *Matrix Computations*. The Johns Hopkins University Press, London, third edition, 1996.
- [12] E.J. Haug. *Computer-Aided Kinematics and Dynamics of Mechanical Systems*. Allyn and Bacon, 1989.
- [13] Baumgarte J.W. A New Method of Stabilization for Holonomic Constraints. *ASME Journal of Applied Mechanics*, 50:869–870, 1983.
- [14] R.L. Kamman, J.W. and Huston. Dynamics of Constrained Multibody Systems. *ASME Journal of Applied Mechanics*, 51:899–903, 1985.
- [15] S.S. Kim and M.J. Vanderploeg. QR Decomposition for State Space Representation of Constrained Mechanical Dynamic Systems. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, 108:176–182, 1986.
- [16] J.G. and Kamat Kurdila, A. and Papastavridis. Role of Maggi's equations in computational methods for constrained multibody systems. *Journal of Guidance, Control, and Dynamics*, 13:113–120, 1990.
- [17] G.A. Maggi. Di alcune nuove forme della dinamica applicabili ai sistemi anolonomi. *Rendiconti della Regia Accademia dei Lincei - Serie V*, X:287–291, 1901.
- [18] N.K. Mani, E.J. Haug, and K.E. Atkinson. Singular Value Decomposition for Analysis of Mechanical System Dynamics. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, 107:82–87, 1985.
- [19] P.E. Nikravesh. Initial condition correction in multibody dynamics. *Multibody System Dynamics*, 18:107–115, 2007.
- [20] D. Pennestrì, E. and de Falco. *Cinematica e Dinamica dei Sistemi Multibody*, chapter Riduzione del numero di coordinate, pages 703–733. Casa Editrice Ambrosiana, 2006.
- [21] E. Pennestrì. *On the Automatic Design Analysis of Gear Trains*. PhD thesis, Columbia Univerisity, 1991.
- [22] E. Pennestrì and F. Freudenstein. A Systematic Approach to Power-flow and Static-Force Analysis in Epicyclic Spur-gear Trains. *ASME Journal of Mechanical Design*, 115:639–644, 1993.

- [23] E. Pennestrì and P.P. Valentini. Dynamic analysis of epicyclic gear trains by means of computer algebra. *Multibody System Dynamics*, 7:249–264, 2002.
- [24] E. Pennestrì, P.P. Valentini, and Vita L. Multibody Dynamics Simulation of Planar Linkages with Dahl friction. *Multibody System Dynamics*, 17:321–347, 2007.
- [25] D. Popa, D. Stănescu, and N.D. Pandrea. On a Mechanism for Coupling the Power Sources. In C.L. Bottasso, editor, *Proc. of the Multibody Dynamics 2007, N.D. ECCOMAS*, June 2007.
- [26] R.P. Singh and P.W. Likins. Singular Value Decomposition for Constrained Mechanical Systems. *ASME Journal of Applied Mechanics*, 52:943–948, 1985.
- [27] F.E. Udewadia and R.E. Kalaba. *Analytical Dynamics a New Approach*. Cambridge University Press, Cambridge, 1996.
- [28] G. and Losantos Unda, J. and de Jalon. A Comparative Study of Different Formulations of the Dynamic Equations of Constrained Mechanical Systems. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, 109:466–474, 1987.
- [29] S. Vlase. A method for eliminating lagrangian multipliers from the equations of motion of interconnected mechanical systems. *ASME Journal of Applied Mechanics*, 54:235–237, 1987.
- [30] E.C. Walton, W.C. and Steeves. A New Matrix Theorem and Its Application for Establishing Independent Coordinates for Complex Dynamical Systems with Constraints. NASA Technical Report TR-327, NASA, 1969.
- [31] R.A. Wehage and E.J. Haug. Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems. *ASME Journal of Mechanical Design*, 134:247–255, 1982.

## APPENDIX: MATLAB code

```
%
% Multibody dynamic analysis of a simple pendulum
%
% Ettore Pennestri' - Universita' Roma Tor Vergata - Italy
% pennestri@mec.uniroma2.it
%
clear all;
T_initial=0.    % Initial time
T_final=100;    % Final time
% Parameters
global L; global mass; global Ig;global method;
L=0.5;mass=10;Ig=0.1;
%%%%%%%%%% Method choice %%%%%%%%%%%
% Uncomment the method you wish to use
% method='putd';
% method='svd';
  method='schur';
% method='qr';
% method='zeroeigenvalue';
% method='udwadiakalaba';
% method='coordinatepartitioning';
%%%%%%%%%%
ndiv=1;
dt=(T_final-T_initial)/ndiv;
% Initial conditions
y0=[L*sqrt(2)/2,-L*sqrt(2)/2,-pi/4,0,0,0];
delete('myresults.dat');
tout = [T_initial];
yout = y0;
t=0;
fprintf('Initial conditions,%6.2f \n',y0);
tstart=T_initial;
tfinal=T_initial+dt;
tempo=0;
profile on
while tfinal<=T_final
tic
options = odeset('RelTol',1e-4);
[t,y] = ode45(@pendulum_dynamics,[tstart tfinal],y0,options);
A=[t,y];
nt=length(t);
tempo=toc+tempo;
nt = length(t);    % Measure the length of the output time vector
tout = [tout; t]; % Append this vector to the previous one
yout = [yout; y]; % Append solution to the previous one
tstart = t(nt);
tfinal=tstart+dt;
y0=y(nt,:);
end
profile viewer
p = profile('info');
profsave(p,'profile_results')
final_y1=[tout yout ];
save('myresults.dat','final_y1','-ASCII');
fprintf('Simulation time %6.2f \n',tempo);
fclose('all');
% Plot angular displacement
figure;
plot(tout,yout(:,3));
grid on;
```

```

xlabel('Time (s)');          % Assign label for x-axis
ylabel('Theta');           % Assign label for y-axis
title('Angular displacement')
% Plot error on constraints
figure;
[mtot,ntot]=size(yout);
for i=1:mtot
    norm(i)=(yout(i,1)-L*cos(yout(i,3)))^2+(yout(i,2)-L*sin(yout(i,3)))^2;
end
semilogy(tout,norm);
grid on;
xlabel('Time (s)');          % Assign label for x-axis
ylabel('Norm');            % Assign label for y-axis
title('Constraint error')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function dy = pendulum_dynamics(t,y)
%
% Multibody dynamic analysis of a simple pendulum
%
% Ettore Pennestri' - Universita' Roma Tor Vergata - Italy
% pennestri@mec.uniroma2.it
%
global L; global mass; global Ig;global method;
s3=sin(y(3));
c3=cos(y(3));
% Mass matrix
M=[mass 0 0;0 mass 0;0 0 Ig];
% Jacobian matrix
Psiq=[1 0 L*s3;0 1 -L*c3];
[m,n]=size(Psiq);
% Force vector
Q=[0; -mass*9.8; 0];
% Gamma vector
gamma=[-L*y(6)*y(6)*c3;-L*y(6)*y(6)*s3];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
switch lower(method)
case('putd')
r=rank(Psiq);
[H,R]=qrhouse(Psiq');
D22=eye(n-r,n-r);
D21=zeros(n-r,r);
D2T=[D21 D22];
V=H*D2T';
case('zeroeigenvalue')
% Zero-eigenvalue method
[T,S] = eig(Psiq'*Psiq);
r=rank(Psiq);
V=T(1:n,1:(n-r));
case('qr')
[QQ,R]=qr(Psiq');
V=QQ(1:n,r+1:n);
case('svd')
r=rank(Psiq);
[Uh,S1,Vh] = svd(Psiq');
V=Uh(1:n,r+1:(n));
case('schur')
r=rank(Psiq);
[US,K] = schur(Psiq'*Psiq);
V=US(1:n,1:(n-r));
case('coordinatepartitioning')

```

```

[ddq]=coord_partitioning(M,Psiq,gamma,Q);
dy=[ y(4) y(5) y(6) ddq(1) ddq(2) ddq(3)]';
return;
case('udwadiakalaba')
[ddq]=udwadiakalaba(M,Psiq,gamma,Q);
dy=[ y(4) y(5) y(6) ddq(1) ddq(2) ddq(3)]';
return;
end
% Compute [S]*{gamma}
zero=zeros(n-m,1);
Sgamma=inv([Psiq;transpose(V)])*[gamma;zero];
Mh=V'*M*V;
Fh=V'*Q-V'*M*Sgamma;
ddv=Mh\Fh;
ddq=V*ddv+Sgamma;
dy=[ y(4) y(5) y(6) ddq(1) ddq(2) ddq(3)]';
end % End-of-function pendulum_dynamics

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function[ddq]=udwadiakalaba(M,Psiq,gamma,f)
% Purpose: Computation of accelerations in a multibody dynamics simulation
%          by means of the Udwadia-Kalaba formulation.
% It is assume that the dynamics equations are in the following form
%
%      |  M   Psiq^T  | | ddq | | f |
%      |  Psiq  0    | | lambda | = | gamma |
%      |  -      -   | | -      |
%
[m,n] = size(M);
for i =1:m
    v(i)=1/M(i,i);
    vl(i)=1/sqrt(M(i,i));
end
Minv=diag(v);
Minvl2=diag(vl);
ddqf=Minv*f;
D=Psiq*Minvl2;
Dps=pinv(D);
[m3,n3]=size(Psiq);
a1=(gamma-Psiq*ddqf);
ddq=ddqf+Minvl2*Dps*a1;
end % of function udwadiakalaba

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function[ddq]=coord_partitioning(M,Psiq,gamma,f)
%
% Ettore Pennestri' - Universita' Roma Tor Vergata - Italy
% pennestri@mec.uniroma2.it
%
% Purpose: Computation of accelerations in a multibody dynamics simulation
%          by means of coordinate partitioning
% It is assumed that the dynamics equations are in the following form
%
%      |  M   Psiq^T  | | ddq | | f |
%      |  Psiq  0    | | lambda | = | gamma |
%      |  -      -   | | -      |
%
[m,n]=size(Psiq);
[R,iu] = rref(Psiq);
it=cumsum(ones(n,1),1);
iv=setdiff(it,iu);

```

```

nu=length(iu);
nv=length(iv);
for i=1:nu
    for j=1:nu
        Muu(i,j)=M(iu(i),iu(j)) ;
    end
end
for i=1:nv
    for j=1:nv
        Mvv(i,j)=M(iv(i),iv(j));
    end
end
for i=1:nv
    for j=1:nu
        Mvu(i,j)=M(iv(i),iu(j));
    end
end
for i=1:nu
    for j=1:nv
        Muv(i,j)=M(iu(i),iv(j));
    end
end
for i=1:m
    for j=1:nu
        Psiu(i,j)=Psiq(i,iu(j));
    end
end
for i=1:m
    for j=1:nv
        Psiv(i,j)=Psiq(i,iv(j));
    end
end
for i=1:nv
    Qv(i)=f(iv(i));
end
for i=1:nu
    Qu(i)=f(iu(i));
end
Qv=Qv';Qu=Qu';
invPsiu=inv(Psiu);
Mh=Mvv-Mvu*invPsiu*Psiv-Psiv'*invPsiu'*(Muv-Muu*invPsiu*Psiv);
Qh=Qv-Mvu*invPsiu*gamma-Psiv'*invPsiu'*(Qu-Muu*invPsiu*gamma);
ddv=Mh\Qh;
ddu=invPsiu*(gamma-Psiv*ddv);
for i=1:nu
    ddq(iu(i))=ddu(i);
end
for i=1:nv
    ddq(iv(i))=ddv(i);
end
end % of function coord_partitioning

```

**Ettore PENNESTRI**, Professor, Universita' di Roma Tor Vergata - Italy, pennestri@mec.uniroma2.it  
**Pier Paolo VALENTINI**, Researcher, Universita' di Roma Tor Vergata - Italy, valentini@ing.uniroma2.it