Proceedings of DETC'97
1997 ASME Design Engineering Technical Conferences
September 14-17, 1997, Sacramento, California

DETC97/VIB-4228

A STRATEGY TO ACCELERATE THE NUMERICAL INTEGRATION IN THE DYNAMIC SIMULATION OF MULTIBODY SYSTEMS

Jesús Cardenal
Department of Mathematical Methods in Engineering.
University of La Coruña.
Mendizábal, s/n.
15403 Ferrol. SPAIN
Phone: +34 81 354130
Fax: +34 81 354463
e-mail: jcarde@cdf.udc.es

Javier Cuadrado
Department of Mechanical Engineering.
University of La Coruña.
Mendizábal, s/n.
15403 Ferrol. SPAIN
Phone: +34 81 354130
Fax: +34 81 354463
e-mail: javicuad@cdf.udc.es

Eduardo Bayo
Department of Mechanical Engineering.
University of La Coruña.
and School of Eng. & Arch.
University of Navarre
31009 Pamplona. SPAIN
Phone: +34 48 425600
Fax: +34 48 425629
e-mail: ebayo@cdf.udc.es

ABSTRACT

This paper presents a multi-index variable time step method for the integration of the equations of motion of constrained multibody systems in *descriptor form*. The basis of the method is the augmented Lagrangian formulation with projections in index-3 and index-1. The method takes advantage of the better performance of the index-3 formulation for large time steps and of the stability of the index-1 for low time steps, and automatically switches from one method to the other depending on the required accuracy and values of the time step.

The variable time stepping is accomplished through the use of an integral of motion, which in the case of conservative systems becomes the total energy. The error introduced by the numerical integrator in the integral of motion during consecutive time steps provides a good measure of the local integration error, and permits a simple and reliable strategy for varying the time step. Overall, the method is efficient and powerful; it is suitable for stiff and non-stiff systems, robust for all time step sizes, and it works for singular configurations, redundant constraints and topology changes. Also, the constraints in positions, velocities and accelerations are satisfied during the simulation process. The method is robust in the sense that becomes more accurate as the time step size decreases.

INTRODUCTION

Kinematic and dynamic simulations of multibody systems allow the accurate prediction of the behavior of heavy machinery, spacecraft, automobile suspensions and steering systems, graphic arts and textile machinery, robots, packaging machinery, machine tools, etc. The first issue to consider in the simulation process is that of modeling the system; that is, the selection of a set of parameters or coordinates that will allow to unequivocally define at all times the position, velocity and acceleration of the system. The most useful kinds of coordinates currently used to define the motion of multibody systems are relative coordinates, reference point (or Cartesian) coordinates, and natural (or fully Cartesian) coordinates. These coordinates, when combined with the principles of dynamics, lead to the final form of the equations of motion. Dynamic principles such as Lagrange's formulation, Newton's Laws, canonical equations of Hamilton, Virtual Power, Hamilton's Principle and Gibbs-Apell equations, constitute the basis for the formulations of multibody dynamics (Haug, 1989, García de Jalón and Bayo, 1994, Nikravesh, 1988, Shabana, 1989). The choice of dynamic formulation determines the subsequent choice of numerical integration schemes.

The method of Lagrange's multipliers leads to a representation of the equations of motion in *descriptor form* constituting a set of index-3[1] differential algebraic equations (DAE). The addition of stabilization techniques, such as the method of Baumgarte (1972), reduces the index and makes the solution tractable by means of standard ordinary differential equations (ODE) solvers, however, it does not provide full constraint satisfaction, leads to a limited control of ac-

[1] Following the notation used by Brenan et al. (1989), given a DAE $F(t,y,y')=0$, we call *index* of that DAE the minimum number of times that all or part of the original DAE must be differentiated with respect to the independent variable (in this case t) in order to determine the derivative of the function, y', as a continuous function of y and t.

curacy, and in addition provides no way for choosing the values of the coefficients used by the method. An augmented Lagrangian formulation with projections (Bayo and Ledesma, 1996) has been proposed which, in addition to transforming the set of equations into a stabilized set, is solvable by standard ODE methods and assure Lyapunov stability of the simulation process (Kurdila et al., to appear). This method also has the advantages of being robust under singular configurations, topology changes and with redundant constraints, and provides full constraint satisfaction.

State-space methods, such as coordinate partitioning (Wehage and Haug, 1982), Kane's method (Kane and Levinson, 1985) and virtual power with projection matrices (Serna et al., 1982), transform the equations of motion to a minimum set of coordinates that are directly solvable by ODE methods. State-space representations may also be obtained by means of *velocity transformations* (Jerkovsky, 1978, Kim and Vanderploeg, 1986, Nikravesh and Gim, 1989, Bae and Won, 1990, Avello et al., 1993). State-space representations are more suitable for ODE integration than the descriptor counterparts at the expense of solving the velocity and position problem at each time step. However, they do not handle topology changes and singular configurations well. In addition, they cannot support stiff integrators and consequently may not be a good choice when the system has built-in numerical stiffness.

The numerical mathematics community has sought solutions to the index reduction problem and has proposed many different ways. Recent advances have been made which have yielded stabilized index reduction methods and accurate ways of projecting the DAE onto the underlying ODE for more stable and accurate solutions. Key developments are the work of Brenan, et al. (1989), Griepentrog, et al. (1992), Führer and Leimkuhler (1991), Hairer and Wanner (1996) (developers of RADAU5), Lubick (1989 and 1992) (developer of MEXX), Petzold (1982) (developer of DASSL) and Arnold (1993 and 1996) (developer of HEX5).

In regard to numerical integration, the backward difference formula (BDF) methods have been customarily used for the solution of differential algebraic equations because the artificial damping thereby introduced helps to stabilize the solution and provides convergence particularly in the index-3 setting. However, the actual implementation of BDF algorithms in general-purpose solvers is not free from serious numerical difficulties, which become more acute for index-3. A few of such difficulties are:

- For an index-*m* DAE the tangent or quasi-tangent matrix used in the Newton-Raphson iteration has a condition number of order $O(1/\Delta t^m)$ (Brenan et al., 1989). Hence, the practical implementation of the method is bound to have large round-off errors for small time steps (usually starting at $\Delta t=10^{-5}$).
- Instabilities may result from sudden changes in system variables and constraints, such as impacts, sudden appearances or disappearances of constraints and topology changes. Any time there is a discontinuity in the response the multi-step BDF tries to fit a polynomial through the discontinuity and therefore the time step size must be severely reduced. As explained in the previous point, this results in an ill-conditioned iteration matrix. Consequently the Newton-Raphson iteration may end up near a solution and yet not be able to converge to it. These problems can be circumvented, but at the expense of re-initializing the integration, thus producing delays in the integration process.
- Also, the multi-step methods are not self-starting. A *k*-step method requires sufficiently accurate *k*-1 starting values which have to be obtained by other methods, thus this may render the method sensitive to the starting values.

Due to the reasons suggested above we propose a multi-index, augmented Lagrangian formulation of the equations of motion in descriptor form for index-1 and index-3. The index-3 form is more efficient than the index-1; however, for time step sizes smaller than 10^{-5} the ill-conditioning of the tangent matrix in index-3 affects the performance of the method, while the index-1 form becomes more accurate and robust (see Cuadrado et al., 1996). A single-step numerical integration scheme with variable time step size is developed based on the strategy mentioned above. Mass-orthogonal projections to the constraint space, as described in Bayo and Ledesma (1996), are performed to assure constraint satisfaction to machine precision during the integration process. At the end of the paper, we devote a section to present numerical simulations that illustrate the performance of the proposed approach.

PRELIMINARIES ON MULTI-BODY DYNAMIC ANALYSIS IN DESCRIPTOR FORM

Formulation in Fully Cartesian coordinates

Let us consider a multibody system whose configuration is characterized by *n* fully Cartesian (or natural) coordinates (García de Jalón and Bayo, 1994) denoted by vector $\mathbf{q}$ that are interrelated through the *m* holonomic kinematic constraint conditions:

$$\mathbf{\Phi}(\mathbf{q},t)=\mathbf{0} \tag{1}$$

The use of the Principle of Virtual Power directly leads to the equations of motion:

$$\delta\dot{\mathbf{q}}^T\left(\mathbf{M}\ddot{\mathbf{q}}-\mathbf{Q}(\mathbf{q},\dot{\mathbf{q}})+\mathbf{\Phi}_\mathbf{q}^T\boldsymbol{\lambda}\right)=0 \tag{2}$$

which for a general multibody system leads to:

$$\mathbf{M}\ddot{\mathbf{q}}+\mathbf{\Phi}_\mathbf{q}^T\boldsymbol{\lambda}=\mathbf{Q}(\mathbf{q},\dot{\mathbf{q}}) \tag{3}$$

where **M** is the mass matrix, **Q** is the vector of the external forces as well as the velocity dependent inertia forces, $\Phi_{\mathbf{q}}$ is the Jacobian of the constraint equations, and λ is a vector containing the Lagrange's multipliers. Note that the use of fully Cartesian coordinates leads to a constant mass matrix **M** and the absence of velocity dependent inertia forces in the vector **Q**; consequently, equation (3) is greatly simplified.

Equations (1) and (3) constitute a set of $n+m$ mixed DAE's of index-3 (Brenan et al., 1989), with **q** and λ as unknowns. It is a common practice in multibody dynamics to differentiate twice the constraints, thus transforming the equation to index-1, and append the resulting equations to (3) to yield:

$$\begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^T \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{Q} \\ -\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} - \dot{\boldsymbol{\Phi}}_t \end{Bmatrix} \tag{4}$$

These equations can now be integrated using standard numerical integration techniques with each function evaluation performed using equation (4). In addition, equation (4) may also be easily modified to include Baumgarte stabilization (Haug, 1989).

Index-1 Augmented Lagrangian Formulation with Projections

Bayo and Ledesma (1996) introduced an index-1 augmented Lagrangian method with mass-orthogonal projections of the positions and velocities to their constraint manifolds. Following, a brief summary of their formulation is included. The equation of motion is as follows:

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^T \alpha \ddot{\boldsymbol{\Phi}} + \boldsymbol{\Phi}_{\mathbf{q}}^T \boldsymbol{\lambda}^* = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}) \tag{5}$$

where λ^* are the Lagrange multipliers and α is the penalty factor (typically 10^7). Introducing the expression of the second derivative of the constraint equations (1) with respect to time into equation (5) the following matricial equation is obtained:

$$\left[\mathbf{M} + \boldsymbol{\Phi}_{\mathbf{q}}^T \alpha \boldsymbol{\Phi}_{\mathbf{q}}\right]\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^T \boldsymbol{\lambda}^* = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}) - \boldsymbol{\Phi}_{\mathbf{q}}^T \alpha\left(\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} + \dot{\boldsymbol{\Phi}}_t\right) \tag{6}$$

The following iteration process yields the unknown multipliers λ^*:

$$\lambda_{i+1}^* = \lambda_i^* + \alpha \ddot{\Phi}_{i+1}, \quad i = 0,1,2,\ldots \tag{7}$$

with $\lambda_0^* = \mathbf{0}$ for the first iteration. Equation (7) physically represents the introduction at iteration $i+1$ of forces that tend to compensate for the addition of all the constraint terms which are not exactly zero. Experience shows that when the constraints are scaled to unity, penalty factors ranging from 10^5 to 10^7 give good convergence rates, and only 1 to 2 iterations are required to converge to the solution. The leading matrix remains constant during the iteration process.

As a result of using the index-1 formulation, the solution of equation (6) yields a set of accelerations that not only satisfies dynamic equilibrium but also the constraint conditions $\ddot{\boldsymbol{\Phi}} = \mathbf{0}$ at $n+1$. But the solution at each stage should also satisfy the first and zero derivatives of the kinematic constraints expressed by equation (1). As a consequence, in order to assure that these constraints are accomplished, it is necessary to project the positions **q** and velocities $\dot{\mathbf{q}}$ onto their constraint manifolds (see Bayo and Ledesma, 1996).

Index-3 Augmented Lagrangian Formulation with Projections

Also presented in Bayo and Ledesma (1996) is an index-3 augmented Lagrangian method with mass-orthogonal projections of the velocities and accelerations on their constraint manifolds. This formulation leads to the following equations of motion:

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^T \alpha \boldsymbol{\Phi} + \boldsymbol{\Phi}_{\mathbf{q}}^T \boldsymbol{\lambda}^* = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}) \tag{8}$$

where λ^* are the Lagrange multipliers and α the penalty factor. In this case, the following iteration process yields the unknown multipliers λ^*:

$$\boldsymbol{\lambda}_{i+1}^* = \boldsymbol{\lambda}_i^* + \alpha \boldsymbol{\Phi}_{i+1}, \quad i = 0,1,2,\ldots \tag{9}$$

with $\lambda_0^* = \mathbf{0}$ for the first iteration. Similar to the index-1 formulation, the value of the penalty factor α affects the convergence rate. Experience shows that, when the constraints are scaled to unity, penalty factors ranging from 10^7 to 10^9 give good convergence rates for the index-3 formulation, and once again only 1 to 2 iterations are required to converge to the solution.

As a result of using the index-3 formulation, the solution of equation (8) yields a set of $\mathbf{q}_{n+1}$ that not only satisfies dynamic equilibrium but also the constraint conditions $\boldsymbol{\Phi} = \mathbf{0}$ at $n+1$. But, as it has been stated for index-1 approach, the solution should also satisfy the first and second derivatives with respect to time of these constraint conditions. As a consequence, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ need to be projected to satisfy $\dot{\boldsymbol{\Phi}} = \mathbf{0}$ and $\ddot{\boldsymbol{\Phi}} = \mathbf{0}$, respectively (see Bayo and Ledesma, 1996).

THE TOTAL SYSTEM ENERGY VARIATION AS A LOCAL ERROR ESTIMATOR

In order to illustrate the correlation between the system energy variation and the local error, we perform the simulation of a double pendulum which moves from rest in the

horizontal position under gravity effects (Figure 1). Each link of the pendulum has a distributed unit mass and a unit length. We use natural coordinates for the modeling process (García de Jalón and Bayo, 1994) and for the integration we use the trapezoidal rule which is a single step, second order, implicit, *A*-stable method and it is also energy preserving in the linear regime. The total time of simulation is 10 seconds and the penalty factor is 10^7.

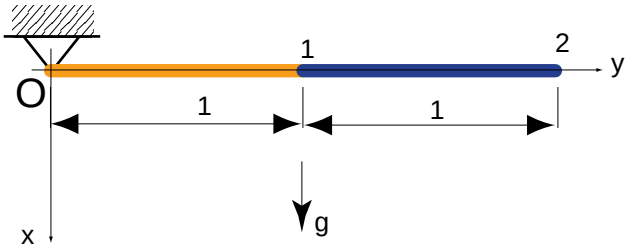

Figure 1. Double pendulum initial position.

There are neither external forces, other than the gravity, nor devices that dissipate energy, hence the total system energy should be constant. The reference level for the potential energy is located at the support level. As the pendulum starts from rest conditions, its total energy should remain equal to zero over the whole simulation.

Table 1. Comparative results double pendulum.

	Index-1		Index-3	
	Error	Time	Error	Time
$\Delta t = 10^{-2}$	3.848e-1	0.39	2.241e-1	0.26
$\Delta t = 10^{-3}$	4.680e-3	2.95	4.854e-3	1.49
$\Delta t = 10^{-4}$	4.421e-5	24.94	4.399-5	14.70
$\Delta t = 10^{-5}$	5.061e-7	238.8	3.962e-7	149.2
$\Delta t = 10^{-6}$	1.258e-7	2370	1.086e-4	1470

Table 1 shows the maximum error in the energy norm and the CPU time[2] in seconds taken by each method for different integration time steps.

The change in the energy can only arise from the numerical error during the integration process. Figures 2 and 3 illustrate the total system energy and the local integration error as a function of time[3]. It may be seen that large local errors are simultaneous to sudden variations of the total system energy. This pattern occurs for all time step sizes, and for both index-1 and index-3 approaches.

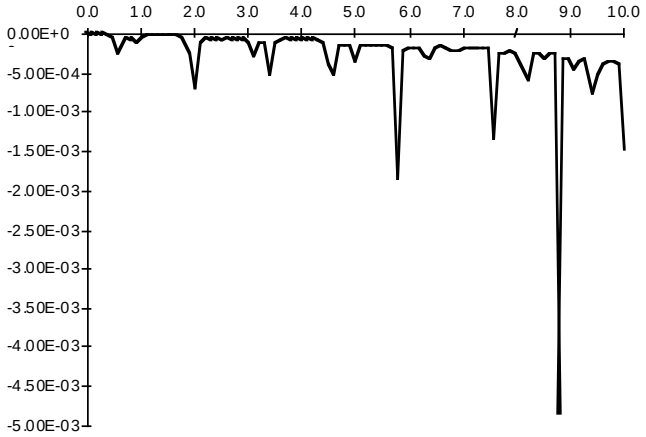

Figure 2. Total system energy vs. time

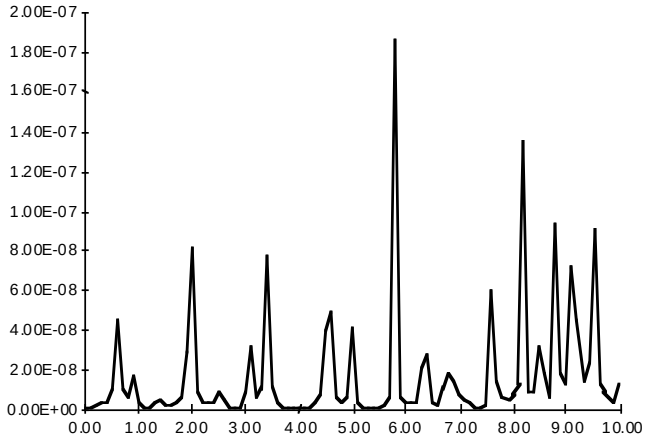

Figure 3. Local integration error vs. time

A MULTI-INDEX VARIABLE SINGLE STEP METHOD

Although the results shown in Table 1 are good for both methods, it may be seen that the index-1 method is more accurate than the index-3 for small time steps, and the index-3 method is more efficient than the index-1 (with same accuracy) at large time steps. This trend is even more pronounced for large scale systems (Cuadrado et al, 1996). Therefore, a multi-index formulation would take advantage of the good qualities of both methods for different time step sizes. Index-3 is the right choice for time steps between 10^{-2} and 10^{-4} (where is accurate and efficient); however, as we decrease the size of the time step, starting in the neighborhood of $\Delta t = 10^{-4}$, the index-1 method becomes more accu-

[2] All tests presented in the paper have been preformed on a Silicon Graphics Indigo[2] workstation equiped whith a MIPS R4400 200 MHz processor. The interest of the authors is mainly focused in comparing the methods, hence the absolute figures are not so important as their relative value.

[3] Both figures belong to a simulation with Index-3 approach and time step size of 10^{-3} seconds. The local error has been evaluated performing each integration step twice: one with the actual time step size and other in two halves of the step size. Richardson extrapolation has been used to obtain a refined result for each step.

rate and robust. At a time step of 10^{-6}, the index-3 method fails.

When performing a simulation, it is desirable to obtain the most accurate results with the least computational effort. Therefore, it is preferable to use an index-3 approach rather than an index-1 whenever possible (*possible* in this case means when the accuracy in the results is enough for the given purposes). Hence, choosing an integration time step is a critical task: if the Δt is too small, the integration process will last more than needed, probably with no evident benefits in the accuracy of the numerical results. Conversely, if the time step is too large, the integration process may diverge or the results obtained may end up being wrong. Consequently, the optimum combination of integration time step and index approach for each problem should be sought.

Considering a fixed single step algorithm, the best combination of index approach and time step will be determined by the nature of the problem and by the worst function conditioning in the sense of its time history. That is, if the function being integrated is smooth all over the time interval, a moderate time step size combined with an index-3 approach will be a good choice. However, in cases where the function has sharp zones, even of short duration, the whole integration process would have to be carried out using a small time step, otherwise it would fail. The choice for the right index is determined by the step size. As a general rule, if the time step size is smaller than 10^{-4} seconds an index-1 should be the choice. Conversely, for time step sizes over 10^{-3} seconds the choice should be index-3. There is a gap between 10^{-4} and 10^{-5} in which both methods could perform well and the choice is highly dependent on the characteristics of the problem.

A variable step integrator would certainly lead to an economy in CPU time for the case considered above. If the time step is variable then it will be possible to accommodate it to obtain constant local error or to maintain this error below some given reference value. Classical approaches in ODE establish a variable time step strategy based on a measure of the local truncation error, which is determined by evaluating either the integrated functions using two different order methods (the Runge-Kutta-Fehlberg, for example), or by integrating two successive time intervals with step size h_n and $h_n/2$. Any of these strategies require many more extra function evaluations, thus compromising the numerical efficiency of the method.

Cardenal et al. (1996) proposed the use of the kinetic energy stored in the penalty system as a measure of the local integration error. This energy, evaluated in the following way:

$$K_\alpha = \frac{1}{2}\alpha\dot{\mathbf{\Phi}}^T\dot{\mathbf{\Phi}} \tag{10}$$

can be computed easily and efficiently. Although good in principle, this manner of evaluating the local integration error has an important drawback: it is quite difficult to establish a general quantitative relationship between the local integration error and the energy stored in the penalty system.

For these reasons, taking into account the existing correlation between the energy and the local error (see Figure 2 and Figure 3), we propose to use an integral of motion (system total energy in conservative systems) as a measure of the local integration error. For non-conservative systems and considering the use of fully Cartesian coordinates, the following integral of motion (energy invariant) may be established: premultiplying the equations of motion by $\dot{\mathbf{q}}^T$ and integrating over time the following expression may be obtained:

$$\int_0^t \dot{\mathbf{q}}^T\left(\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q}\right)d\tau = constant \tag{11}$$

The integration of this equation leads to:

$$\Pi = T(t) + V(t) - \int_0^t \dot{\mathbf{q}}^T\mathbf{Q}d\tau = constant \tag{12}$$

which clearly yields $T + V = constant$ for conservative systems.

THE MULTI-INDEX AUTO-TIME-STEPPING ALGORITHM

The time history of the energy invariant is taken as a measure of the local integration error. The variable time step size strategy is based on modifying the step size when the change in the energy invariant exceeds an allowable value. When this occurs a new time step size is calculated in order to achieve the desired accuracy. The new time step size $\overline{\Delta t}_n$ can be expressed as a function of the previous time step size Δt_n *in* the following way (Ralston and Rabinowitz, 1978):

$$\overline{\Delta t}_n = \eta_r \Delta t_n \tag{13}$$

where η_r, is a parameter defined as:

$$\eta_r = \left(\frac{\nu\varepsilon h_n}{|T_r|}\right)^{1/r} \tag{14}$$

where T_r is a function of the local error estimation; ε is the allowed local error; r the integration index and ν a safety factor typically taken as 0.8.

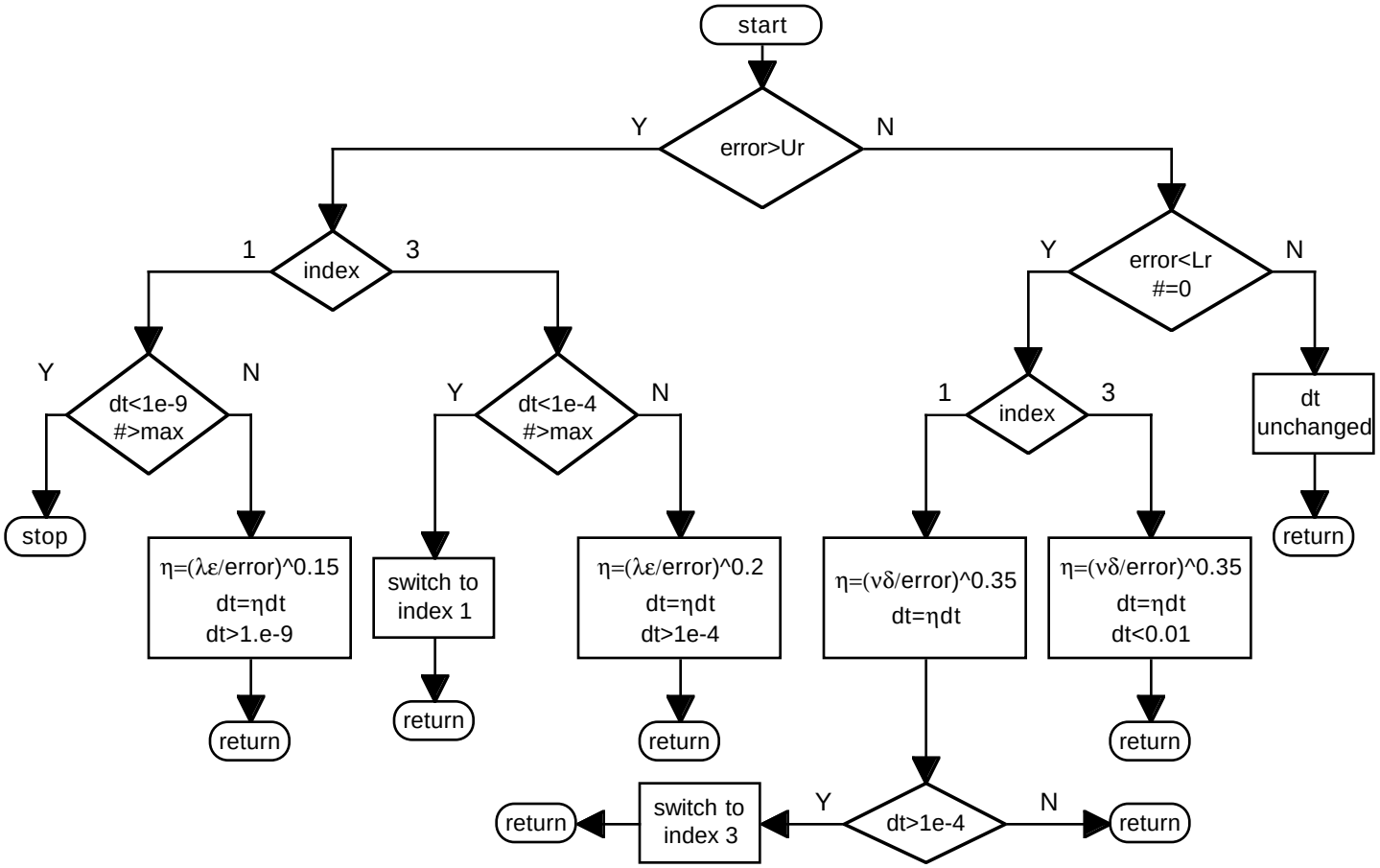

Figure 4. Multi-index and time-stepping strategy flowchart

The strategy to perform the change of index is based on the general consideration that when integrating in index-3, the larger the time step size, the better the performance, and that the time step should never be smaller than 10^{-4}. On the contrary, for the index-1 approach, the smaller the step size, the better the results. There is an overlapping zone, ranging from $\Delta t = 10^{-4}$ to 10^{-5} in which both approaches are suitable. In this zone, it is preferable to use the more economical index-3 rather than the index-1 approach.

Figure 4 illustrates in a flowchart manner the general strategy established for the modification of time step size.

The index change is also based on convergence considerations. If any integration step does not converge, or the local error is larger than a specified allowable value U_r, a time step reduction is performed. If the problem persists after several consecutive time step reductions (denoted as *max* in the flowchart of Figure 4) using the index-3 approach, even for step sizes larger than 10^{-4} then it is necessary to change to index-1. On the contrary, when the local error is below some specified value L_r the step size is enlarged and a change of index is performed (from index-1 to index-3) provided the step size is larger than 10^{-4}.

It is important to remark that prior to the change to index-1 from index-3, it is necessary to recalculate the accelerations $\ddot{\mathbf{q}}$ and velocities $\dot{\mathbf{q}}$ to satisfy equation (6) for the current positions $\mathbf{q}$.

NUMERICAL RESULTS

In this section two examples are presented. First of all the double pendulum described earlier in this paper and the front suspension system of an off-road vehicle.

Double Pendulum

As it has been written before, the *double pendulum case* consist on the simulation of the behavior of a double pendulum released from its horizontal resting position. The simulation lasts for 10 seconds. Each bar of the pendulum has a length of 1 meter and weights 1 Kg.

Figure 5 and Figure 6 show the results obtained in this simulation. It may be seen that index and time step size changes occur when sudden variations in the system energy invariant take place. In this simple example the overhead introduced by the variable-time-stepping algorithm is comparatively large to the total CPU time consumed (see Table

1). Hence no benefits, in terms of CPU time, are obtained. However, considerable accuracy is obtained when integrating at variable time step size, as seen in Figure 6, when compared to fixed step size.

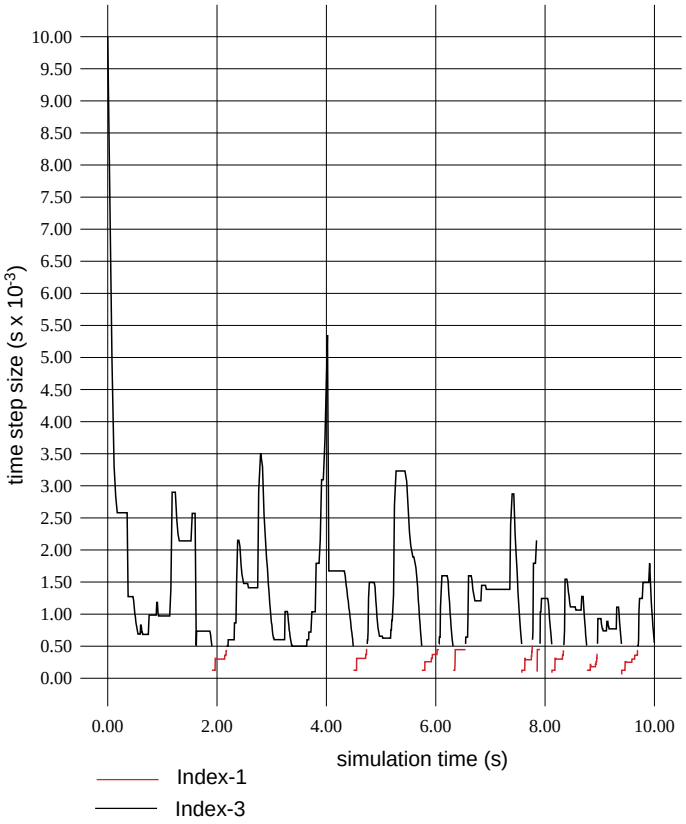

Figure 5. Time step size history.

Off-road vehicle suspension system

The 1/4 ton 4x4 Iltis vehicle (Iltis Data Package, 1990) has been proposed as a benchmark problem by the European automobile industry to check multibody dynamic codes.

We have performed the simulation of the front suspension of the vehicle (see Figure 7) under the following conditions: at the starting time, the suspension is at rest but its position does not correspond to the static equilibrium, so, it is left to freely oscillate until the static equilibrium position is reached. After three seconds, the suspension at 5 m/s goes over a road bump given by a cosine profile and afterwards is left to freely oscillate until the equilibrium is reached again. The complete analysis lasts for 6 seconds. The characteristics of the system are: a leaf spring, modeled as a linear spring of stiffness 35,900 N/m; a shock absorber, which provides an elastic force due to an external polymer, given by,

$$F_S = -4.0092\cdot 10^6 + 2.8397\cdot 10^7 x - 6.7061\cdot 10^7 x^2 + \\ +5.2796\cdot 10^7 x^3 \text{ N} \qquad (15)$$

and a damping force given by the following formula:

$$\begin{aligned} F_D &= -416.42 + 1844.3v \text{ N} \quad v < -0.2 \text{ m/s} \\ F_D &= 9945.627v + 33955.72v^2 - 59832.25v^3 - 395651.0v^4 \text{ N} \\ &-0.2 < v < 0.21 \text{ m/s} \\ F_D &= 1919.1638 + 1634.727v \text{ N} \quad v > 0.21 \text{ m/s} \end{aligned} \qquad (16)$$

where the distance x is in meters; the tyre, modeled by means of a linear vertical spring of stiffness 460,000 N/m. The model requires a total of 23 variables, related through 22 constraint equations, since there is only one degree of freedom.

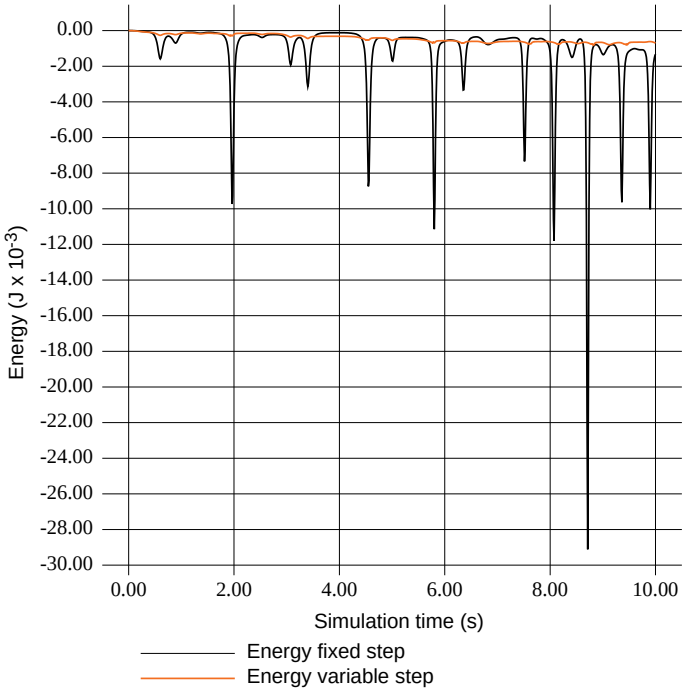

Figure 6. Total system energy.

We perform the first simulation using a fixed index-1 approach with a fixed time step of $5\cdot 10^{-4}$ seconds. The second simulation is performed at variable time step and index, starting with index-3 and a step size of 10^{-3} seconds. The tolerance values U_r and L_r are $2.65\cdot 10^{-5}$ and 10^{-7}, respectively. The first simulation took 54.45 seconds of CPU time and the second took 15.18 seconds.

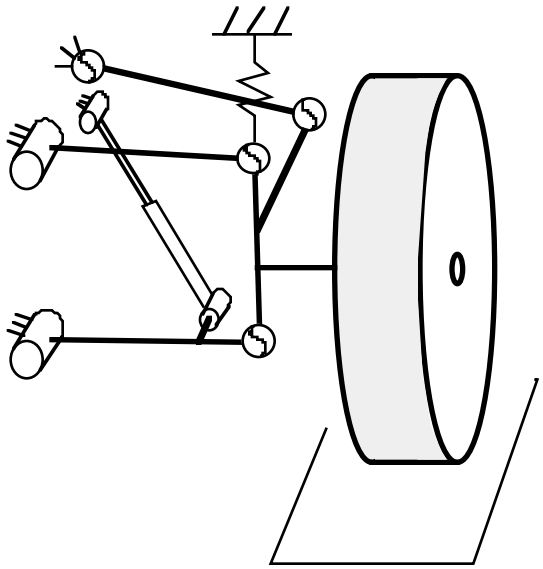

Figure 7. Iltis suspension.

Figure 8 shows the time step size evolution. It may be clearly seen that, for the specified tolerance it is necessary to shorten the time step size only when the wheel is passing over the bump. At the beginning the step size is quickly adjusted to speed up the process while maintaining the local error between the allowable limits.

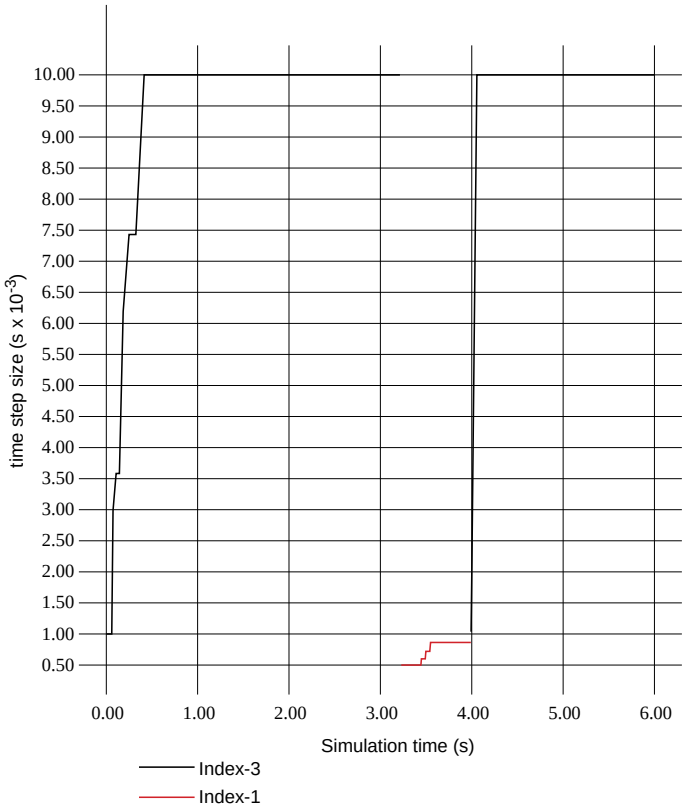

Figure 8. Time step size history.

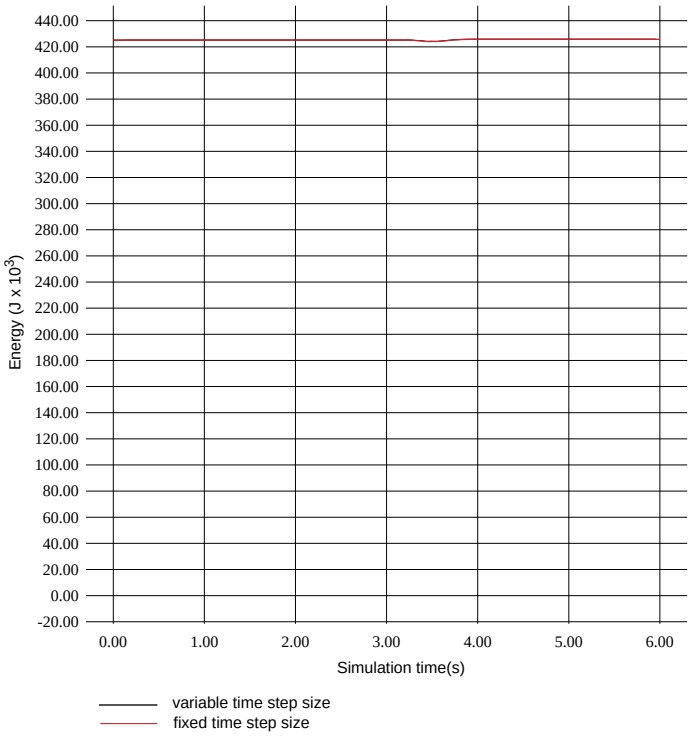

Figure 9. Total system energy.

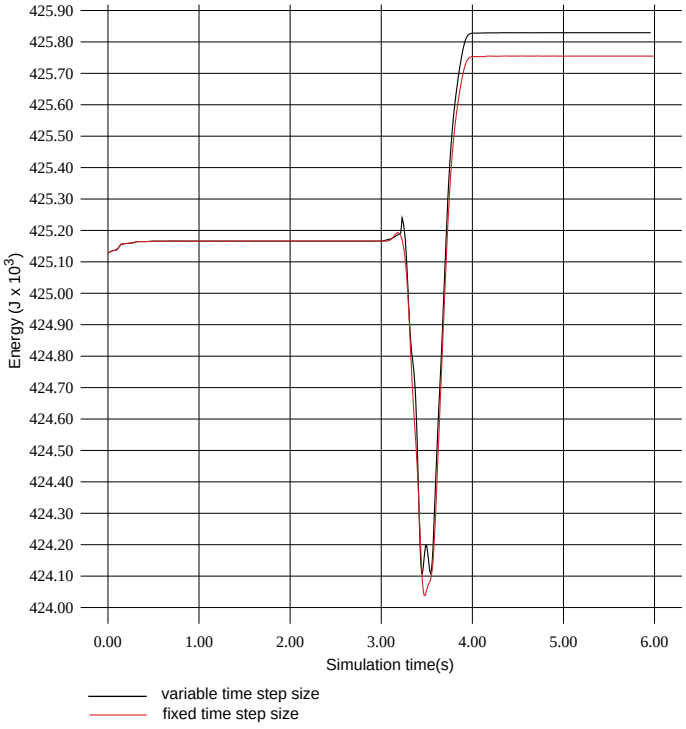

Figure 10. Total system energy (zoomed).

Figure 9 and Figure 10 show the total system energy in both cases. It is important to remark that only when the energy is represented scaling the vertical axis (Figure 10) we can notice its change. The biggest deviation from the initial value (425,128 J) is about 0.2% of the total system energy in both cases. And the difference between results obtained in both cases is about 0.02% of the total system energy, while there is a 72% gain in CPU time.

CONCLUSIONS

The proposed multi-index formulation shows a behavior that could be defined as complementary. On one hand, index-1 leads to worse results than index-3 for large time steps, whereas, index-3 provides wrong results for very small time steps due to numerical ill conditioning. On the other hand, index-3 is more efficient while index-1 is more robust. In order to accommodate all these features a multi-index variable time step size strategy has been devised based on the following criteria:

1. Whenever possible the time step is increased.
2. The time step size is decreased based on the measure of the total energy invariant, which has been show to maintain a connection with the local error.
3. The threshold to change from index-3 to index-1 is set to 10^{-4}.

These are the major conclusions:

- The system energy invariant has proved to be a good measure of the local integration error. This feature makes unnecessary the use of traditional (but less efficient) error criteria used in standard numerical integration methods.
- The variable time step strategy and the switch from one method to the other do not cause problems during the integration process.
- The use of the proposed technique allows a speedup gain in CPU time for large scale systems, thus helping to achieve real-time behavior.
- The method is general and can also be applied to solve the dynamics of flexible multibodies.

ACKNOWLEDGMENTS

The support of this work provided by the Army Research Office under grant DAAH04-95-1-0662, and the CICYT of the Spanish Ministry of Education under grant TAP95-0226 is gratefully acknowledged.

REFERENCES

Arnold, M., 1993, "Stability of Numerical Methods for Differential-Algebraic Equations of Higher Index", *Applied Numerical Mathematics*, Vol. 13, pp. 5-14.

Arnold, M., 1996, "Half-explicit Runge-Kutta methods with explicit stages for differential-algebraic systems of index 2, *to be published*.

Avello, A., Jiménez, J.M., Bayo, E. and García de Jalón, J., 1993, "A Simple and Parallelible Method for Real-Time Dynamic Simulation Based on Velocity Transformations", *Journal of Computer Methods in Applied Mechanics and Engineering*, Vol. 107, pp. 313-339.

Bae, D.S. and Won, Y.S., 1990, "A Hamiltonian Equation of Motion for Real-Time Vehicle Simulation", *Advances in Design Automation*, Vol. 2, pp. 151-157.

Baumgarte, J., 1972, "Stabilization of Constraints and Integrals of Motion in Dynamical Systems", *Journal of Computer Methods in Applied Mechanics and Engineering*, Vol. 1, pp. 1-16.

Bayo, E. and Ledesma, R., 1996, "Augmented Lagrangian and Mass-Orthogonal Projection Methods for Constrained Multibody Dynamics". *Journal of Nonlinear Dynamics*, Vol. 9, pp. 113-130.

Bayo, E., García de Jalón, J. and Serna, M.A., 1988, "A Modified Lagrangian Formulation for the Dynamic Analysis of Constrained Mechanical Systems", *Computer Methods in Applied Mechanics and Engineering*, Vol. 71, pp. 183-195.

Brenan, K.E, Campbell, S.L. and Petzold, L.R., 1989, "The Numerical Solution of Initial Value Problems in Differential-Algebraic Equations", Elsevier Science Publishing, Co., North-Holland, New York.

Cardenal, J., Cuadrado, J. and Bayo, E., 1996, "A multi-index variable time step method for the dynamic simulation of multibody systems", 16th Computers in Engineering Conference (ASME), Irvine (California).

Cuadrado, J., Cardenal, J. and Bayo, E., 1996, "Modeling issues for the intelligent simulation of Multibody Dynamics", 24th Biennal Mechanism Conference (ASME), Irvine (California).

Führer, C. and Leimkuhler, B.J., 1991, "Numerical solution of Differential Algebraic Equations for Constrained Mechanical Motion", *Numerische Mathematik*, Vol. 59, pp. 55-69.

García de Jalón, J. and Bayo, E., 1994, "Kinematic and Dynamic Simulation of Multibody Systems. The Real-Time Challenge", Springer-Verlag, New York.

Griepentrog, E., Hanke, M and Marz, R., 1992, "Berlin Seminar on Differential Algebraic Equations", Fachbereich Mathematik der Humboldt-Universitat zu Berlin.

Hairer, E. and Wanner, G., 1996, "Solving Ordinary Differential Equations II", Springer-Verlag, Berlin.

Haug, E.J., 1989, "Computer-Aided Kinematics and Dynamics of Mechanical Systems, Volume I: Basic Methods", Allyn and Bacon.

"Iltis Data Package", IAVSD Workshop, Herbertov, Czechoslovakia, Sept., (1990).

Jerkovsky, W., 1978, "The Structure of Multibody Dynamic Equations", *Journal of Guidance and Control*, Vol. 1, pp. 173-182.

Kane, T.R. and Levinson, D.A., 1985, "Dynamics: Theory and Applications", McGraw-Hill, New York.

Kim, S.S. and Vanderploeg, M.J., 1986, “A General and Efficient Method for Dynamic Analysis of Mechanical Systems Using Velocity Transformations”, *Journal of Mechanisms, Transmissions and Automation in Design*, Vol. 108, pp. 176-182.

Kurdila, A. J., Junkins, J. L., and Hsu, S., to appear, “Lyapunov Stable Penalty Methods for Imposing Nonholonomic Constraints in Multibody System Dynamics”, *Journal of Nonlinear Dynamics*.

Lubich, Ch., “h2-extrapolation methods for differential-algebraic systems”, *Impact Comput. Sc. Eng*., Vol. 1, pp. 260-268

Lubich, Ch., Nowak, U., Pöhle, U. and Engstler, Ch., 1992, “MEXX–Numerical Software for the Integration of Constrained Mechanical Multibody Systems”, preprint SC 92-12, Konrad-Zuse-Zentrum, Berlin.

Nikravesh, P.E. and Gim, G., 1989, “Systematic Construction of the Equations of Motion for Multibody Systems Containing Closed Kinematic Loops”, *Advances in Design Automation*, Vol. 3, pp. 27-33, ASME Press.

Nikravesh, P.E., 1988, “Computer-Aided Analysis of Mechanical Systems”, Prentice-Hall.

Petzold, L.R., 1982, “A Description of DASSL: A Differential/Algebraic System Solver”, Proceedings of IMACS World Congress, Montreal, Canada.

Ralston, A. and Rabinowitz, P., 1978, “A First Course in Numerical Analysis. 2nd Edition”, McGraw-Hill, Inc., New York.

Schiehlen, W.O., 1990, “Multibody System Handbook”, Springer-Verlag, Heidelberg.

Serna, M.A., Avilés, R. and García de Jalón, J., 1982, “Dynamic Analysis of Planar Mechanisms with Lower-Pairs in Basic Coordinates”, *Mechanism and Machine Theory*, Vol. 17, pp. 397-403.

Shabana, A.A., 1989, “Dynamics of Multibody Systems”, Wiley.

Wehage, R.A. and Haug, E.J., 1982, “Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems”, *ASME Journal of Mechanical Design*, Vol. 104, pp. 247-255.