

# Augmented Lagrangian and Mass-Orthogonal Projection Methods for Constrained Multibody Dynamics

E. BAYO

*School of Engineering and Architecture, University of Navarra, 31080 Pamplona, Spain*

and

R. LEDESMA

*Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, U.S.A.*

(Received: 7 September 1994; accepted: 28 September 1994)

**Abstract.** This paper presents a new method for the integration of the equations of motion of constrained multibody systems in descriptor form. The method is based on the penalty-Augmented Lagrangian formulation and uses mass-orthogonal projections for the solution to satisfy the kinematic constraint conditions. The number of equations being solved is equal to the number of states, and does not depend on the number of constraint conditions. Therefore, the method is particularly suitable for systems with redundant constraints, singular configurations or topology changes. The major advantage of the new method relies on the fact that for a low computational cost, the constraints in positions, velocities and accelerations are satisfied to machine precision during the numerical integration. This process is efficiently done by means of a mass-orthogonal projection without the need for coordinate partitioning or reduction to a minimum set of coordinates. The projection scheme allows for a more accurate and robust integration of the equations of motion since constraint violations constitute one of the primary sources of numerical errors and instabilities during the integration process. The proposed projection is also applied to the classical Lagrangian approach, thus eliminating the need for further stabilization as well as the selection of parameters in Baumgarte's method.

**Key words:** Constrained multibody systems, penalty and augmented Lagrangian methods, constrained stabilization, projection methods.

## 1. Introduction

The kinematics and dynamics of multibody systems constitute an important part of what is referred to as CAD (*Computer Aided Design*) and MCAE (*Mechanical Computer Aided Engineering*). The systems included under the definition of *multibodies* comprise heavy machinery, spacecraft, automobile suspensions and steering systems, graphic arts and textile machinery, robots, packaging machinery, machine tools, etc. In recent years operating speeds have increased, and consequently there has been an increase in accelerations and inertia forces. These large forces inevitably lead to the appearance of dynamic problems that one must be able to accurately predict, thus requiring more sophisticated simulation tools.

Computer simulations performed by CAD and MCAE tools allow designers to predict the kinematic and dynamic behavior of different types of multibody systems in great detail, during all the design stages: from the first design concepts to the final prototypes. The kinematic and dynamic analysis of multibody systems are processes which are most appropriately performed using *interactive simulations*. The analyst is interested in visualizing a whole set of successive responses of the multibody system, a simulation of its behavior and operation over all the

workspace and over a certain period of time. In certain cases it may be necessary to obtain a *real-time* response, and even introduce the analyst, or hardware, as additional elements in the simulation process, thus leading to *man-in-the-loop* or *hardware-in-the-loop* situations which introduce external forces or control over specific degrees of freedom. Obviously the class and size of problems that may be solved in real-time will increase as the computer hardware and numerical algorithms improve in the ensuing years.

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 important types of coordinates currently used to define the motion of multibody systems are: *relative* coordinates, *reference point* or *Cartesian* coordinates, and *natural* or *full Cartesian* coordinates. Another consideration is the choice of the dynamic formulation, obtained from the application of the principles of mechanics, that will lead to the final form of the equations of motion. Dynamic principles such as Lagrange's equations, Newton's laws, canonical equations of Hamilton, virtual power, Hamilton's principle, Gibbs–Apell equations, etc. constitute the basis for the formulations of multibody dynamics. The choice of the dynamic formulation very much determines the subsequent choice of numerical integration schemes.

The widely used method of Lagrange's multipliers leads to a representation of the equations of motion in its *descriptor form* constituting a set of index-three differential algebraic equations (DAE), that is not solvable by state of the art numerical software. The addition of stabilization techniques, such as the method of Baumgarte [1], reduces the index and makes the solution tractable by means of standard ODE solvers, however, this method does not provide full constraint satisfaction, and no ways are available for choosing the values of the coefficients used by the method. A staggered stabilization based on a penalty approach has been presented [2] which integrates two different sets of equations one for the coordinates and another for the Lagrange multipliers. The use of penalty and augmented Lagrangian methods [3] in addition to transforming the set of equations into a stabilized set of index one that is solvable by standard ODE methods, assure Lyapunov stability of the simulation process [4]. These methods also have the advantages of being robust under singular configurations, topology changes and with redundant constraints [5], however, they do not provide full constraint satisfaction either, and as a consequence accuracy may degrade in long simulations.

The principles of mechanics may also be used to directly reduce the index of the equations. For example, the canonical equations of Hamilton lead to an index-two representation with the position coordinates and canonical momenta as unknown variables [5]. Other formulations, such as coordinate partitioning [6], Kane's method [7, 8] and virtual power with projection matrices [9] transform the equations of motion to a minimum set of coordinates or *state-space form* that is directly solvable by ODE methods. State-space representations may also be obtained by means of *velocity transformations* initially introduced by Jerkovsky [10] and subsequently extended by other authors [11–15]. These representations are more suitable for ODE integration than the descriptor counterparts, and as a consequence, they yield a better control of accuracy 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 numerical integration and index reduction problem, which has led to a proliferation of methods too vast to summarize herein. Recent advances include stabilized index reduction methods and accurate ways of

projecting the DAE into the underlying ODE for more stable and accurate solutions. Key developments are the work of Brenan *et al.* [16] and Griepentrog *et al.* [17] who summarize the state of the art in this area. More specifically, in regard to the index-three DAE arising in the modeling of nonlinear dynamics, a method has been proposed for stabilized index-one formulation [18, 19], and shown that is equivalent to the space-state generalized inverse form. These methods lead to numerically tractable index-three DAE but require a sizable computational effort because the number of unknowns of the problem increases by three times the number of constraint equations. A recent work [20] studies the stability, efficiency and accuracy of a new index reduction method based on dummy differentiation. Again the major drawback seems to be the cost of increasing the number of unknowns by three times the number of constraint equations.

In this paper we combine the augmented Lagrangian formulation with a mass-orthogonal projection technique to the constraint manifold, which yields a new method that has all the advantages of the former and yields no constraint errors. The major advantage of the method relies on the fact that for a low computational cost, the constraints in positions, velocities and accelerations are satisfied to machine precision during the integration process which becomes much more accurate than the regular augmented Lagrangian formulation. As explained below, this process is done without the need for coordinate partitioning or reduction to a minimum set of coordinates and only requires the solution of a set of equations equal to the number of states of the system.

## 2. Preliminaries on the Equations for Multibody Dynamics in Descriptor Form

### 2.1. FORMULATION IN LAGRANGIAN COORDINATES

Let us consider a multibody system whose configuration is characterized by  $n$  generalized coordinates  $\mathbf{q}$  that are interrelated through the  $m$  holonomic kinematic constraint conditions:

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

The use of the principle of virtual power directly leads to the following equations [22]

$$\delta \dot{\mathbf{q}}^T (\mathbf{M} \ddot{\mathbf{q}} - \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}) + \Phi_{\mathbf{q}}^T \lambda) = 0, \quad (2)$$

where  $\mathbf{M}$  is the mass matrix,  $\mathbf{Q}$  is the vector that contains the external and non-conservative forces, as well as the velocity dependent inertia terms; and  $\Phi_{\mathbf{q}}$  is the Jacobian of the constraint equations. The Lagrange multipliers  $\lambda$  are introduced so that the expression in parenthesis becomes zero, thus leading to the following equations of motion:

$$\mathbf{M} \ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \lambda = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}). \quad (3)$$

Equations (1) and (3) constitute a set of  $n + m$  mixed differential algebraic equations (DAE) of index three [16], with  $\mathbf{q}$  and  $\lambda$  as unknowns. In order to avoid the direct integration of DAE, a double differentiation of the constraints equations may be carried out, yielding:

$$\begin{bmatrix} \mathbf{M} & \Phi_{\mathbf{q}}^T \\ \Phi_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{Q} \\ -\dot{\Phi}_{\mathbf{q}} \dot{\mathbf{q}} - \dot{\Phi}_t \end{Bmatrix}. \quad (4)$$

These equations can now be integrated using standard numerical integrators with each function evolution performed using equation (4). In addition, this equation may also be easily modified to include Baumgarte's stabilization [1].

## 2.2. AUGMENTED LAGRANGIAN FORMULATION

The augmented Lagrangian formulation leads to the following equations of motion [3]:

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \alpha (\ddot{\Phi} + 2\Omega\mu\dot{\Phi} + \Omega^2\Phi) + \Phi_{\mathbf{q}}^T \lambda^* = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}), \quad (5)$$

where  $\lambda^*$  are the new Lagrange multipliers of the modified system. Note that the second term in the LHS of equation (5) represents the projection in the direction of the coordinates  $\mathbf{q}$  of all the internal forces that are generated by the dynamic penalty system when the constraints  $\Phi$ ,  $\dot{\Phi}$  and  $\ddot{\Phi}$  are violated. Introducing  $\ddot{\Phi} = \Phi_{\mathbf{q}\mathbf{q}}\ddot{\mathbf{q}} + \dot{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} + \ddot{\Phi}_t$  the following final result is obtained

$$(\mathbf{M} + \Phi_{\mathbf{q}}^T \alpha \Phi_{\mathbf{q}})\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \lambda^* = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}) - \Phi_{\mathbf{q}}^T \alpha (\dot{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} + \ddot{\Phi}_t + 2\Omega\mu\dot{\Phi} + \Omega^2\Phi). \quad (6)$$

This index-one formulation may be viewed as the “generic penalty method” [3] to which the Lagrange’s multipliers are added. As shown in [3] the augmented Lagrangian formulation allows the analyst to choose from a wide range of penalty values that assure convergence and avoid numerical ill-conditioning, thus avoiding the problems customarily attributed to the penalty formulations in multibody dynamics and allowing the use of non-stiff integrators.

It is important to note that there is very important difference between equation (6) and the Lagrange’s multiplier approach represented by (4). The leading matrix of equation (4) becomes singular in singular configurations and in the presence of redundant constraints. However, although the mass matrix  $\mathbf{M}$  is in general positive semi-definite, the leading matrix of (6),  $(\mathbf{M} + \Phi_{\mathbf{q}}^T \alpha \Phi_{\mathbf{q}})$ , is always positive definite, even in singular positions and/or with redundant constraints [5].

We are seeking the solution of (6) without having to use the algebraic constraint equations (1). This requires that the correct values of  $\lambda^*$  be known so that they can be inserted in (6). The following iteration process [3] yields the unknown multipliers  $\lambda^*$

$$\lambda_{i+1}^* = \lambda_i^* + \alpha (\ddot{\Phi} + 2\Omega\mu\dot{\Phi} + \Omega^2\Phi)_{i+1}, \quad i = 0, 1, 2, \dots \quad (7)$$

with  $\lambda_0^* = 0$  for the first iteration. Equation (7) physically represents the introduction at iteration  $i + 1$  of forces that tend to compensate the fact that the addition of all the constraint terms are not exactly zero. It turns out that with the augmented Lagrangian formulation, the penalty numbers do not need to be very large (thus leading to a better numerical conditioning) since the resulting error in the constraint equations will be eliminated by the Lagrange’s terms during the iteration procedure.

The matrix formulation of (6), including the iterative process defined in (7), is given by the following expression:

$$(\mathbf{M} + \Phi_{\mathbf{q}}^T \alpha \Phi_{\mathbf{q}})\ddot{\mathbf{q}}_{i+1} = \mathbf{M}\ddot{\mathbf{q}}_i - \Phi_{\mathbf{q}}^T \alpha (\dot{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} + \ddot{\Phi}_t + 2\Omega\mu\dot{\Phi} + \Omega^2\Phi), \quad i = 0, 1, 2, \dots, \quad (8)$$

where the subscript  $i$  represents the iteration number, and  $\mathbf{M}\ddot{\mathbf{q}}_0 = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}})$  for the initial iteration. Equation (8) may be used to iterate until  $\|\ddot{\mathbf{q}}^{(i+1)} - \ddot{\mathbf{q}}^{(i)}\| < \varepsilon$ , where  $\varepsilon$  is a user-specified tolerance.

The main advantage of using equation (8) is that the penalty terms are in fact used as an intermediate tool in order to compute the Lagrange’s multipliers for which no new equations are integrated, i.e., only  $n$  equations are solved in the integration process. Therefore, the value of the penalty factor  $\alpha$  does not affect the solution, but only the convergence rate. Experience shows that when the constraints are scaled to unity, penalty factors ranging from  $10^5$  to  $10^7$

give a good convergence rate, and only 2 to 3 iterations are required to converge to the solution.

Note that the added cost of using equation (8) to refine the solution and obtain the Lagrange multipliers is fairly small, since its leading matrix remains constant during the iteration process needed for a function evaluation. Therefore, at each iteration step only the computation of the acceleration-dependent term, a forward reduction, and a back-substitution are required. The numerical implementation of the algorithm using standard integrators [21], available in commercial mathematical libraries, is rather simple and may be described as follows.

Using standard ODE integrators with accelerations as primary variables the algorithm for the augmented Lagrangian formulation becomes:

#### ALGORITHM ALF

1. Given  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  at time step  $l$ .
2. Use (8) iteratively to solve for  $\ddot{\mathbf{q}}$  (function evaluation), with  $\mathbf{M}\ddot{\mathbf{q}}_0 = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}})$  for the initial iteration. Then use (7) to calculate the Lagrange multipliers  $\lambda^*$ , if desired.
3. Call the numerical integration subroutine (n.i.s.) to compute  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  at time step  $l + 1$ .
4. Upon convergence of the n.i.s. update the time variable and go to step 1.

### 3. Augmented Lagrangian Formulation with Projections

#### 3.1. MOTIVATION

As explained above the Augmented Lagrangian Formulation (ALF) shows very good characteristics and robustness as a general way of formulating and solving the equations of motion in descriptor form. At the time of coupling the equations of motion with the ODE integrator and performing the numerical integration, we pose the following question:

*Is there a computationally efficient way of bringing down to zero (within machine precision) not only the weighted term  $(\dot{\Phi} + 2\Omega\mu\dot{\Phi} + \Omega^2\Phi)$  but also each of the individual constraints  $\dot{\Phi}$ ,  $\dot{\Phi}$  and  $\Phi$  during the integration process?*

If this were possible, the ODE integrator would be alleviated from the already small constraint errors and would yield higher accuracy. We impose, however, the condition that for the sake of numerical efficiency the new approach should make the individual constraints – position, velocity or accelerations – be satisfied (we call this process *constraint cleaning*) without having to resort to coordinate partitioning or transformation to a minimum set of coordinates because these lead to expensive computations.

#### 3.2. PROJECTION OF THE SOLUTION TO THE CONSTRAINT MANIFOLD

During the integration process the augmented Lagrangian formulation yields a solution set of  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}}$  that does not enforce each individual constraint to be exactly equal to zero. In order to achieve full constraint satisfaction at each time step we propose a mass-orthogonal projection that takes the solution to the constraint manifold where  $\dot{\Phi}$ ,  $\dot{\Phi}$ ,  $\Phi$  are all equal to zero. In essence, our objective is to force the set of DAE to meet the underlying ODE at each time step. It is important to note that this projection scheme can also be carried out in the same manner within the classical Lagrange multiplier technique, and in doing so one eliminate the need of further stabilization such as the method of Baumgarte [1].

*(a) Projection in  $\mathbf{q}$* 

During the time integration process the numerical integration scheme yields a set of coordinates  $\mathbf{q}^*$  that does not completely satisfy the constraint coordinates  $\Phi = \mathbf{0}$ . In order to satisfy the constraints, we perform a mass-orthogonal projection of the solution to the constraint manifold and obtain a new set of positions  $\mathbf{q}$  that satisfies  $\Phi = \mathbf{0}$ . This can be enforced by the solution of the following constrained minimization problem

$$\min_{\mathbf{q}} V = \frac{1}{2} (\mathbf{q} - \mathbf{q}^*)^T \mathbf{M} (\mathbf{q} - \mathbf{q}^*) \quad \text{subject to} \quad \Phi(\mathbf{q}, t) = 0. \quad (9)$$

It may be seen how the mass matrix  $\mathbf{M}$  has been intentionally added to the objective function with the idea that the resulting equations will have the same leading matrix as that used in the dynamic analysis (equation (6)). In what follows and for the sake of clarity we will assume that the mass matrix is constant. The ensuing procedure is perfectly valid for a non-constant mass matrix.

In order to solve the problem posed by equation (9), we use an augmented Lagrangian method and minimize the following function

$$V^* = \frac{1}{2} (\mathbf{q} - \mathbf{q}^*)^T \mathbf{M} (\mathbf{q} - \mathbf{q}^*) + \frac{1}{2} \Phi^T \alpha \Phi + \Phi^T \lambda, \quad (10)$$

where  $\alpha$  is the penalty factor that we will assume is identical for all the constraints (although the constraints may be scaled so that each one has a different penalty value). As mentioned above a good choice for  $\alpha$  when working in double precision arithmetics is  $10^7$ .

We now differentiate  $V^*$  with respect to  $\mathbf{q}$  and equate to zero, yielding

$$\mathbf{H}(\mathbf{q}, t) = \frac{\partial V^*}{\partial \mathbf{q}} = \mathbf{M}(\mathbf{q} - \mathbf{q}^*) + \Phi_{\mathbf{q}}^T \alpha \Phi + \Phi_{\mathbf{q}}^T \lambda = 0, \quad (11)$$

where  $\mathbf{H}(\mathbf{q}, t)$  is the vector of  $n$  nonlinear functions. Equation (11) constitutes a system of nonlinear algebraic equations that can be solved by Newton–Raphson iteration. The first order expansion of  $\mathbf{H}(\mathbf{q}, t)$  yields:

$$\begin{aligned} \mathbf{H}(\mathbf{q} + \Delta \mathbf{q}, t) &= \mathbf{M}(\mathbf{q} - \mathbf{q}^*) + \Phi_{\mathbf{q}}^T \alpha \Phi + \Phi_{\mathbf{q}}^T \lambda + \mathbf{M} \Delta \mathbf{q} \\ &\quad + (\Phi_{\mathbf{q}}^T \alpha \Phi_{\mathbf{q}} + \Phi_{\mathbf{q}\mathbf{q}}^T \alpha \Phi + \Phi_{\mathbf{q}\mathbf{q}}^T \lambda) \Delta \mathbf{q}. \end{aligned} \quad (12)$$

From equation (12), we can set up the following iterative procedure to calculate  $\mathbf{q}$

$$(\mathbf{M} + \Phi_{\mathbf{q}}^T \alpha \Phi_{\mathbf{q}}) \Delta^{(i+1)} = -\gamma_p^{(i)} \quad (13)$$

with

$$\gamma_p^{(i)} = \mathbf{M}(\mathbf{q}^{(i)} - \mathbf{q}^*) + \Phi_{\mathbf{q}}^T \lambda^{(i)} \quad (14)$$

and

$$\mathbf{q}^{(i+1)} = \mathbf{q}^{(i)} + \Delta^{(i+1)} \quad \text{and} \quad \lambda^{(i+1)} = \lambda^{(i)} + \alpha \Phi^{(i+1)}, \quad (15)$$

where the superscript indicates the iteration number. The terms involving  $\Phi_{\mathbf{q}\mathbf{q}}$  have been removed from the tangent matrix because they are much smaller than  $\Phi_{\mathbf{q}}^T \alpha \Phi_{\mathbf{q}}$ . Equations (13) through (15) can be used iteratively until  $|\Delta \mathbf{q}| < \epsilon$ , where  $\epsilon$  is a user-specified tolerance.

Note the important feature that the leading matrix in equation (13) is identical to that used in equation (6) for the dynamic analysis. Furthermore, since the solution  $\mathbf{q}$  is close to the initial values  $\mathbf{q}^*$ , the projection problem can be solved using a modified Newton–Raphson method with no need for updating the tangent matrix and therefore, *the only cost associated with the projection is the evaluation of a new right hand side and a forward and backward substitution.*

REMARK I. The constraint minimization problem of equation (9) can also be solved using the classical Lagrange multiplier method. In this case the projection scheme can be performed using the following equation:

$$\begin{bmatrix} \mathbf{M} & \Phi_{\mathbf{q}}^T \\ \Phi_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \Delta \mathbf{q} \\ \Delta \lambda \end{Bmatrix}^{(i+1)} = \begin{Bmatrix} -\gamma_p^{(i)} \\ \mathbf{0} \end{Bmatrix} \quad (16)$$

which has the same leading matrix as that of equation (4). Therefore, the formation and decomposition of the leading matrix can be avoided.

(b) *Projection in  $\dot{\mathbf{q}}$*

Similarly during the time marching integration process the numerical integration scheme yields a set of velocities  $\dot{\mathbf{q}}^*$  that does not completely satisfy the constraint conditions  $\dot{\Phi} = \mathbf{0}$ . Again, we perform a mass-orthogonal projection of the solution to the velocity constraint manifold and obtain a new set of velocities  $\dot{\mathbf{q}}$  that will satisfy  $\dot{\Phi} = \mathbf{0}$ . Since the projection in  $\mathbf{q}$  has been previously carried out,  $\Phi = \mathbf{0}$ , and therefore the mass-orthogonal projection to  $\dot{\Phi} = \mathbf{0}$  is enforced by the solution of the following constrained minimization problem

$$\min_{\dot{\mathbf{q}}} V = \frac{1}{2} (\dot{\mathbf{q}} - \dot{\mathbf{q}}^*)^T \mathbf{M} (\dot{\mathbf{q}} - \dot{\mathbf{q}}^*) \quad \text{subject to} \quad \dot{\Phi}(\mathbf{q}, \dot{\mathbf{q}}, t) = 0. \quad (17)$$

The projection will lead to equations that have the same leading matrix as that used in the dynamic analysis (equation (6)). The time derivative of the constraint equation can be expressed as

$$\dot{\Phi}(\mathbf{q}, \dot{\mathbf{q}}, t) = \Phi_{\mathbf{q}} \dot{\mathbf{q}} + \Phi_t. \quad (18)$$

The augmented Lagrangian method is used again to minimize the following function

$$V^* = \frac{1}{2} (\dot{\mathbf{q}} - \dot{\mathbf{q}}^*)^T \mathbf{M} (\dot{\mathbf{q}} - \dot{\mathbf{q}}^*) + \frac{1}{2} \dot{\Phi}^T \alpha \dot{\Phi} + \dot{\Phi}^T \sigma, \quad (19)$$

where  $\sigma$  are the new Lagrangian multipliers. We now differentiate  $V^*$  with respect to  $\dot{\mathbf{q}}$  and equate to zero. Accordingly,

$$\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}, t) = \frac{\partial V^*}{\partial \dot{\mathbf{q}}} = \mathbf{M} (\dot{\mathbf{q}} - \dot{\mathbf{q}}^*) + \Phi_{\mathbf{q}}^T \alpha \dot{\Phi} + \Phi_{\mathbf{q}}^T \sigma = 0, \quad (20)$$

where we have made use of the fact that  $\Phi_{\mathbf{q}}^T = \dot{\Phi}_{\mathbf{q}}^T$ . Substituting equation (18) into equation (20), the following set of algebraic equations for the generalized velocities are obtained:

$$(\mathbf{M} + \Phi_{\mathbf{q}}^T \alpha \Phi_{\mathbf{q}}) \dot{\mathbf{q}} = \mathbf{M} \dot{\mathbf{q}}^* - \Phi_{\mathbf{q}}^T (\alpha \Phi_t + \sigma), \quad (21)$$

where the Lagrange multipliers are updated by the following expression:

$$\sigma^{(i+1)} = \sigma^{(i)} + \alpha \dot{\Phi}^{(i+1)}. \quad (22)$$

A more efficient implementation of the augmented Lagrangian penalty formulation for the projection of velocities is given by the following recursive set of equations:

$$(\mathbf{M} + \Phi_{\mathbf{q}}^T \alpha \Phi_{\mathbf{q}}) \dot{\mathbf{q}}^{(i+1)} = \mathbf{M} \dot{\mathbf{q}}^{(i)} - \Phi_{\mathbf{q}}^T \alpha \Phi_t \quad (23)$$

and the recursion is started by setting

$$\dot{\mathbf{q}}^{(i)} = \dot{\mathbf{q}}^* \quad \text{and} \quad \sigma^{(i)} = \alpha \dot{\Phi}^{(i)}. \quad (24)$$

Equation (23) constitutes a system of linear algebraic equations in  $\dot{\mathbf{q}}$  that may be readily solved. Since the leading matrix in equation (23) depends on the generalized coordinates only, the leading matrix needs to be triangularized only once, and the recursion expressed by equation (23) will only require a forward reduction and a back-substitution.

REMARK II. Similar to the projections in  $\mathbf{q}$ , if the classical Lagrangian multiplier method is used, the projection in velocities can be performed solving the following equation:

$$\begin{bmatrix} \mathbf{M} & \Phi_{\mathbf{q}}^T \\ \Phi_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{q}} \\ \sigma \end{Bmatrix} = \begin{Bmatrix} \mathbf{M} \dot{\mathbf{q}} - \Phi_{\mathbf{q}}^T \alpha \Phi_t \\ \mathbf{0} \end{Bmatrix}, \quad (25)$$

which again has the same leading matrix as that of equation (4).

### (c) Projections in $\ddot{\mathbf{q}}$

Following the same procedure as in the velocity analysis, the projection of the generalized accelerations onto the acceleration constraint manifold can be obtained through the following constrained minimization problem:

$$\min_{\ddot{\mathbf{q}}} V = \frac{1}{2} (\ddot{\mathbf{q}} - \ddot{\mathbf{q}}^*)^T \mathbf{M} (\ddot{\mathbf{q}} - \ddot{\mathbf{q}}^*) \quad \text{subject to} \quad \ddot{\Phi}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) = 0. \quad (26)$$

For the holonomic case, the second time derivative of the constraint equations can be expressed as:

$$\ddot{\Phi}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) = \Phi_{\mathbf{q}\mathbf{q}} \ddot{\mathbf{q}} + \dot{\Phi}_{\mathbf{q}\mathbf{q}} \dot{\mathbf{q}} + \ddot{\Phi}_t. \quad (27)$$

The augmented Lagrangian formulation of the above minimization problem is established by defining a modified objective function:

$$V^* = \frac{1}{2} (\ddot{\mathbf{q}} - \ddot{\mathbf{q}}^*)^T \mathbf{M} (\ddot{\mathbf{q}} - \ddot{\mathbf{q}}^*) + \frac{1}{2} \ddot{\Phi}^T \alpha \ddot{\Phi} + \ddot{\Phi}^T \kappa, \quad (28)$$

where  $\kappa$  are the new Lagrange multipliers. We now differentiate the modified objective function with respect to the generalized accelerations to obtain

$$\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) = \frac{\partial V^*}{\partial \ddot{\mathbf{q}}} = \mathbf{M} (\ddot{\mathbf{q}} - \ddot{\mathbf{q}}^*) + \Phi_{\mathbf{q}}^T \alpha \ddot{\Phi} + \Phi_{\mathbf{q}}^T \kappa = 0, \quad (29)$$



where we have made use of the identity

$$\Phi_{\mathbf{q}}^T = \ddot{\Phi}_{\dot{\mathbf{q}}}^T. \quad (30)$$

Substituting equation (27) into equation (29) the following set of algebraic equations for the generalized accelerations are obtained:

$$(\mathbf{M} + \Phi_{\mathbf{q}}^T \alpha \Phi_{\mathbf{q}}) \ddot{\mathbf{q}} = \mathbf{M} \ddot{\mathbf{q}}^* - \Phi_{\dot{\mathbf{q}}}^T \{ \alpha (\dot{\Phi}_{\mathbf{q}} \dot{\mathbf{q}} + \Phi_t) + \kappa \}, \quad (31)$$

where the associated Lagrange multipliers are updated by the following expression:

$$\kappa^{(i+1)} = \kappa^{(i)} + \alpha \ddot{\Phi}^{(i+1)}. \quad (32)$$

The numerical implementation of the augmented Lagrangian formulation for the projection of accelerations is given by the recursive set of equations

$$(\mathbf{M} + \Phi_{\mathbf{q}}^T \alpha \Phi_{\mathbf{q}}) \ddot{\mathbf{q}}^{(i+1)} = \mathbf{M} \ddot{\mathbf{q}}^{(i)} - \Phi_{\dot{\mathbf{q}}}^T \alpha (\dot{\Phi}_{\mathbf{q}} \dot{\mathbf{q}} + \Phi_t) \quad (33)$$

and the recursion is started by setting

$$\ddot{\mathbf{q}}^{(0)} = \ddot{\mathbf{q}}^* \quad \text{and} \quad \kappa^{(0)} = \alpha \ddot{\Phi}^{(0)}. \quad (34)$$

Equation (33) constitutes a system of linear algebraic equations in  $\ddot{\mathbf{q}}$  that are solved repeatedly until convergence in the generalized accelerations is achieved. *Since the leading matrix in equation (33) is the same as that of equation (23) used for the velocity projection, the acceleration analysis will require only a forward reduction and a back-substitution.*

## 4. Numerical Implementation of the Mass-Orthogonal Projections

### 4.1. INDEX-1 FORMULATION

Equation (6) or equivalent (8) can be solved using standard ODE integration [21] with accelerations as primary variables. The projections in displacement and velocities can be done at each time step to yield a set of corrected values for  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  that lie on the constraint manifold. The numerical implementation of the augmented Lagrangian formulation with projections and with accelerations as primary variables becomes:

#### ALGORITHM ALF-INDEX 1

1. Given  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  at time step  $l$ .
2. Use (8) iteratively to solve for  $\ddot{\mathbf{q}}$  (function evaluation), with  $\mathbf{M} \ddot{\mathbf{q}}_0 = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}})$  for the initial iteration. Then use (7) to calculate the Lagrange multipliers  $\lambda^*$ , if desired.
3. Call the numerical integration subroutine (n.i.s.) to compute  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  at time step  $l + 1$ .
4. Use equations (13) and (23) to perform the projections in position and velocities, and obtain a clean set of  $\mathbf{q}$  and  $\dot{\mathbf{q}}$ .
5. Upon convergence of the n.i.s. and update the time variable and go to step 1.

Note that it may be desired not to perform the projection (step 4 in the above algorithm) at each time step. Instead, one may wish to carry out the projection at every given number of steps. The numerical examples of Section 5 below show the trade off between accuracy and numerical cost when performing the projection at each time step or at a given number of steps.

Also note that when using the accelerations as primary variables the projection in  $\ddot{\mathbf{q}}$  is not necessary because the condition  $\ddot{\Phi} = \mathbf{0}$  is implicitly satisfied in the dynamic analysis (equation (6)) as a consequence of imposing that  $\Phi$  and  $\dot{\Phi}$  be equal to zero. In addition, if the projection in  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  are performed at every time step *the choice of parameters for  $\mu$  and  $\Omega$  is irrelevant, in fact they may be set equal to zero since the constraints in positions and velocities are now being satisfied at each time step.* The same conclusions apply for the classical Lagrange multipliers technique (equations (4), (16) and (25)).

#### 4.2. INDEX-THREE FORMULATION

We describe in this section a way of integrating the equations of motion and carrying out the projection scheme using an index-three formulation, in which the positions  $\mathbf{q}$  become the primary variables during the integration process. Let us start by rewriting the equations of motion (equation (6)) in the following form:

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \alpha \Phi + \Phi_{\mathbf{q}}^T \lambda^* = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}), \quad (35)$$

where the terms  $\ddot{\Phi}$  and  $\dot{\Phi}$  have been eliminated due to the fact that the projections in positions and velocities enforce that these constraints are equal to zero.

The difference equations of the numerical integration scheme may be expressed in general as:

$$\hat{\mathbf{q}}_{n+1} = a\mathbf{q}_{n+1} + \hat{\mathbf{q}}_{n+1}, \quad \hat{\ddot{\mathbf{q}}}_{n+1} = b\mathbf{q}_{n+1} + \hat{\dot{\mathbf{q}}}_{n+1}, \quad (36)$$

where  $a$  and  $b$  are constants that depend on the numerical integration scheme and the time step,  $\hat{\mathbf{q}}_{n+1}$  and  $\hat{\dot{\mathbf{q}}}_{n+1}$  are known quantities that depend on the positions, velocities and accelerations at step  $n$  and/or previous steps. Note that if the method is explicit,  $a$  and  $b$  will be equal to zero. The substitution of equation (36) into equation (35) yields:

$$b\mathbf{M}\mathbf{q}_{n+1} + \Phi_{\mathbf{q}}^T(\alpha\Phi + \lambda^*) = \mathbf{Q}_{n+1} - \mathbf{M}\hat{\ddot{\mathbf{q}}}_{n+1}, \quad (37)$$

which constitutes a set of nonlinear algebraic equations with  $\mathbf{q}_{n+1}$  as unknowns. We use Newton–Raphson iteration and evaluate the solution by means of an iterative process,

$$\left[ \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \right]^{(i)} \Delta \mathbf{q}^{(i+1)} = \mathbf{f}(\mathbf{q}^{(i)}) \quad (38)$$

where  $i$  is the iteration number. The function  $f$  is defined as

$$\mathbf{f}(\mathbf{q}_{n+1}) = \mathbf{M}\hat{\ddot{\mathbf{q}}}_{n+1} + \Phi_{\mathbf{q}}^T(\alpha\Phi + \lambda^*) - \mathbf{Q}_{n+1} \quad (39)$$

and the tangent matrix is approximated by

$$\frac{\partial \mathbf{f}}{\partial \mathbf{q}} \cong b\mathbf{M} + \Phi_{\mathbf{q}}^T \alpha \Phi_{\mathbf{q}} - \mathbf{Q}_{\mathbf{q}}. \quad (40)$$

Similar to what was done in the projection in  $\mathbf{q}$ , the terms involving  $\Phi_{\mathbf{q}\mathbf{q}}$  have been eliminated because they are negligible compared to the terms given in (40).

As a result of using the positions  $\mathbf{q}$  as primary variables, the solution of equation (37) yields a set of  $\mathbf{q}_{n+1}$  that not only satisfies dynamic equilibrium, but also the constraint conditions

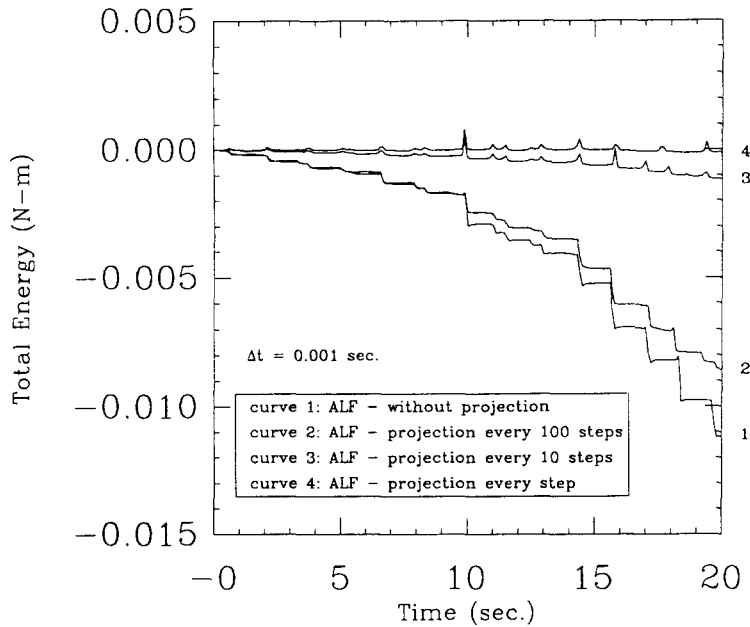


Fig. 1. Double pendulum. Energy time history.

$\dot{\Phi} = 0$  to machine precision. As a consequence the projection in  $\mathbf{q}$  is not necessary, and only  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}}$  need to be projected to satisfy  $\dot{\Phi} = 0$  and  $\ddot{\Phi} = 0$ .

The numerical implementation of the augmented Lagrangian formulation with projections and index-three formulation becomes:

#### ALGORITHM ALF-INDEX3

1. Given  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}}$  at time step  $l$ .
2. Use (38) iteratively to solve for  $\mathbf{q}_{l+1}$ . Then use (15) to calculate the Lagrange multipliers  $\lambda^*$ , if desired.
3. Use equations (23) and (33) to perform the projections in velocities and accelerations, and obtain a clean set of  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}}$ .
4. Update the time variable and go to step 1.

## 5. Numerical Results

*First Example.* The simulation of a double pendulum that moves from rest in the horizontal position under gravity effects is carried out. Each link has a distributed unit mass and a unit length. We use natural coordinates for the modeling process [22] and for the integration we use the midpoint rule which is second order, implicit,  $A$ -stable and  $B$ -stable method, symplectic and also energy preserving in the linear regime [23]. The time step of integration is 0.001 seconds and the total time of simulation is 20 seconds.

Figure 1 illustrates the time histories of the energy resulting from the analysis using the index-one formulation. The first plot shows the energy when no projection is made, the second when the projection is applied every 100 time steps, the third at every 10 steps and the fourth at every time step. These figures show that the errors are small in all cases (less than 0.04%) even when there are no projections to the constraint manifold. However, the proposed projection process helps reducing the integration errors.

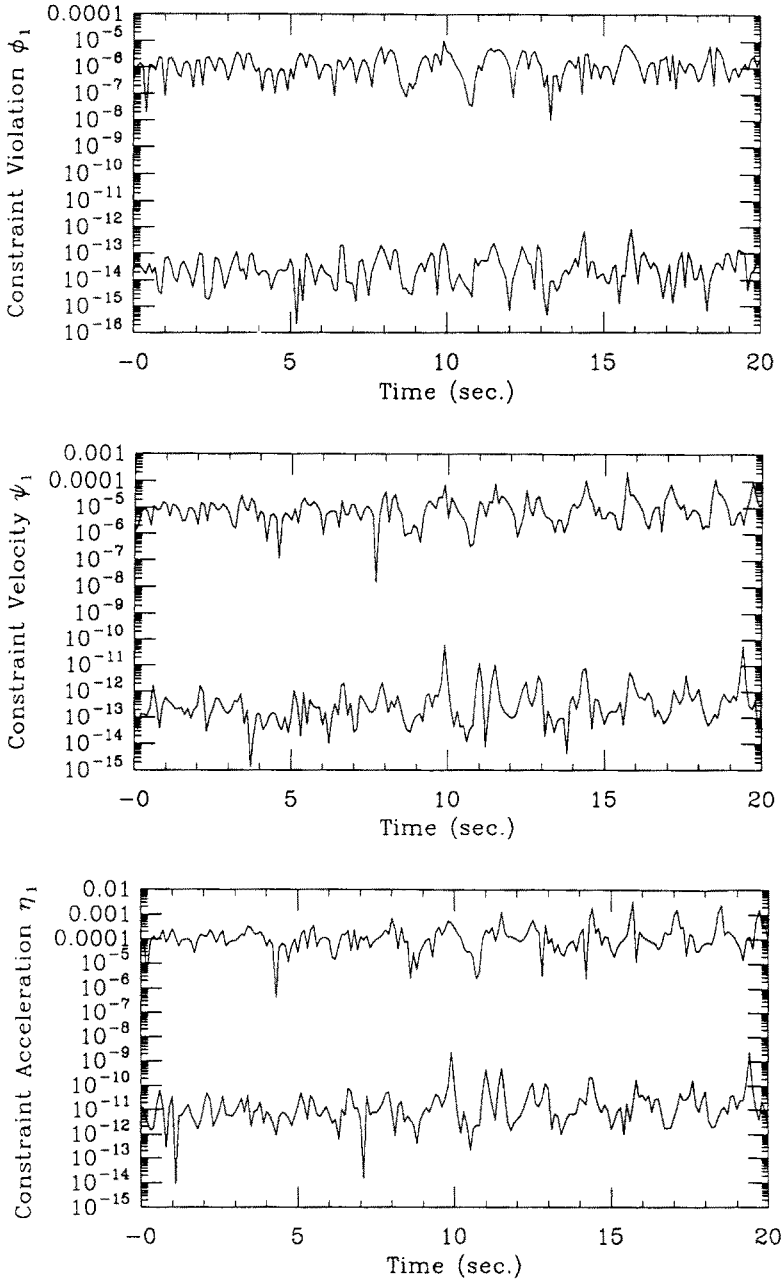


Fig. 2. Constraint violations in positions, velocities and accelerations with and without projections.

The results obtained in the constraint stabilization using the proposed projection process are shown in Figure 2 which compare the errors in  $\Phi$ ,  $\dot{\Phi}$  and  $\ddot{\Phi}$  with and without projections. It may be observed how the projection process reduces the constraint errors to machine precision.

Table 1 shows the results obtained, and illustrate the improvements in accuracy as measured by the error in the total energy when compared with the maximum kinetic energy of the system. Table 1 also illustrates the increment in CPU time necessary for the projections. It

Table 1. Comparative results with and without projections using ALF-INDEX 1.

Method	% Increment in CPU time	Max. error in % of max. kinetic energy	% Increment in accuracy
Standard A.L.F.	–	0.0377%	–
Projecting every 100 steps	1.9%	0.0287%	76.13%
Projecting every 10 steps	8.5%	0.0039%	966.67%
Projecting every step	23.40%	0.0004%	10668.37%

Table 2. Comparative results of a long simulation with and without projections.

Method	$\Delta E$	$\Delta L$	Drift $\Delta E$	Drift $\Delta L$	CPU time
Lagran. with no Stabiliz	$2.4 \cdot 10^0$	2.5	quadratic	linear	380
ALF $\mu = 1, \Omega = 10$	$9.0 \cdot 10^{-3}$	$5.0 \cdot 10^{-6}$	linear	none	314
ALF-INDEX1	$3.0 \cdot 10^{-4}$	$\sim 10^{-14}$	linear	none	391
ALF-INDEX3	$3.0 \cdot 10^{-4}$	$\sim 10^{-14}$	linear	none	211

may be seen how only 23.40% increment in CPU time, resulting from the projection at each time step, yields a much higher increase in accuracy.

*Second Example.* We solve the simulation problem proposed in [16] as a benchmark to observe the accuracy of index reduction methods in long simulations. The problem consists of a single pendulum of unit mass and length, modeled with Cartesian coordinates  $x$  and  $y$ , subject to a fictitious gravity of unit value. A large oscillation problem is studied with horizontal initial position ( $x = 1$  and  $y = 0$ ) and initial velocity ( $\dot{x} = 0$  and  $\dot{y} = -1$ ).

The numerical simulation is performed for 1000 units of time using the trapezoidal rule with fixed step size equal to 0.01 in a SPARC I workstation. The results of the simulation are shown in Table 2 where the four algorithms are compared: first, the Lagrange’s multiplier approach with no constraint stabilization (equation (4)); second, the ALF with no projections and parameters  $\mu = 1$  and  $\Omega = 10$ ; third the ALF-INDEX1 (accelerations as primary

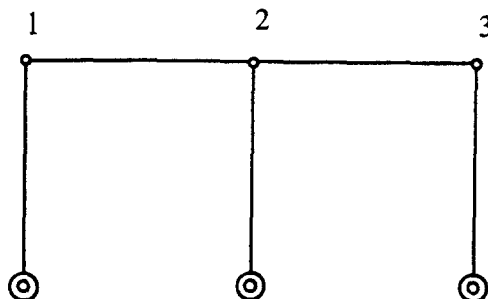


Fig. 3. Assembly of two four-bar linkages.

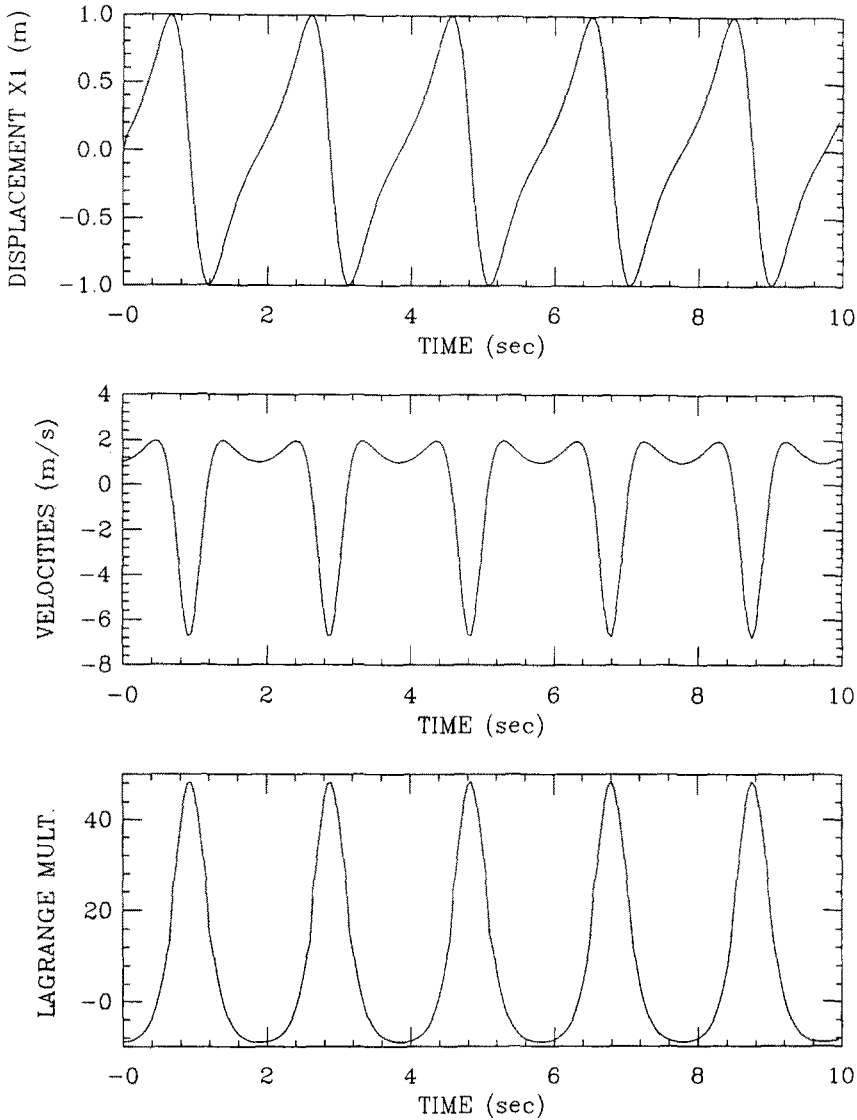


Fig. 4. Two four-bar linkage: coordinate  $x_1$  (top), velocity  $\dot{x}_1$  (middle) and Lagrange multiplier (bottom).

variables) with projection at every time step; and fourth the ALF-INDEX3 (positions as primary variables) with projections at every time step.

The results are compared in Table 2, where  $\Delta E$  and  $\Delta L$  indicate, respectively, the total energy error and drift of the length of the pendulum (the only constraint) at the end of the simulation. Table 2 also illustrates the total CPU time used by each of the methods as well as the drifts in  $\Delta E$  and  $\Delta L$ . It may be seen that the first case yields, as expected, unacceptable results. The augmented Lagrangian formulation with projection yields a substantial improvement in accuracy as well as constraint satisfaction over the standard ALF. In addition, an important observation comes from the comparison of the CPU time taken by the index-one and index-three formulations: the difference is almost 100%.

*Third Example.* Figure 3 shows the initial position of a one degree-of-freedom assembly

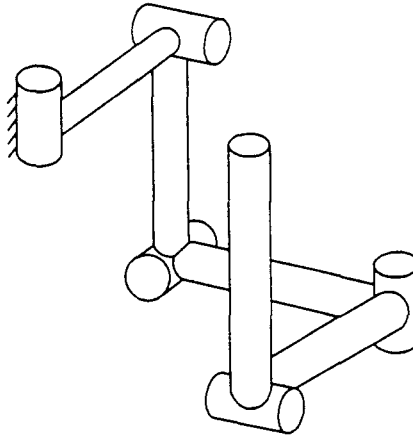


Fig. 5. Five link open-chain multibody.

of two four-bar linkages. This mechanism constitutes a particularly critical example, because it undergoes a singular configuration any time it reaches the horizontal position. Then, the number of degrees of freedom increases instantaneously from 1 to 3. To define the position of the system, we use the 6 position variables  $(x_1, y_1, x_2, y_2, x_3, y_3)$ . All the links are of length  $\ell = 1$  m and have a uniformly distributed mass  $m = 1$  Kg. The gravity force acts in the negative  $Y$  direction, with a value  $g = 8.81$  m/s<sup>2</sup>. At  $t = 0$  the initial velocity is  $\dot{x}_1 = 1$ . We integrate the motion for 10 seconds with time step of 0.025 seconds using the trapezoidal rule and the algorithm ALF-INDEX3. The value of the penalty parameter is  $\alpha = 10^7$ .

The results obtained are displayed in Figure 4. The top plot shows the time history of the coordinate  $x_1$ , the middle shows the velocity  $\dot{x}_1$ , and the bottom one shows the Lagrange multiplier  $\lambda_1$ , corresponding to a constant distance constraint between point 1 and the fixed end of the leftmost link. During the simulation the mechanism goes through the singular position ( $|x_1| = 1$ ) 10 times with no difficulty and without locking the simulation.

*Fourth example.* The mechanism of Figure 5 is composed of five links interconnected by revolute joints. It is modeled with 30 natural coordinates [22] and 25 constraint equations. The motion is due to self weight and the simulation is carried out using the trapezoidal rule with the algorithm ALF-INDEX3 during 10 seconds. The time step is 0.008 seconds and  $\alpha = 10^7$ .

The results obtained are displayed in Figure 6. The top plot shows the time history of the tip acceleration which shows peak values of 170 m/sec<sup>2</sup>, thus indicating the good convergence characteristics of the method under very strong motions. The middle plot shows the total energy, which is nicely preserved during the integration process, and the bottom one the norm of the constraint violations with and without the augmented Lagrangian iteration. This iteration leads to a constraint satisfaction within machine precision.

*Fifth Example.* The Andrew's squeezing mechanisms, described in detail in [24], has been proposed as an example to test numerical algorithms in multibody dynamics. We perform the simulation of this mechanism using the index-one formulation with a time increment of  $2 \cdot 10^{-5}$  and  $\alpha = 10^7$ , and the trapezoidal rule for the numerical integration. The total time of the simulation is 0.03 seconds, and the time history plots of the angles between the different

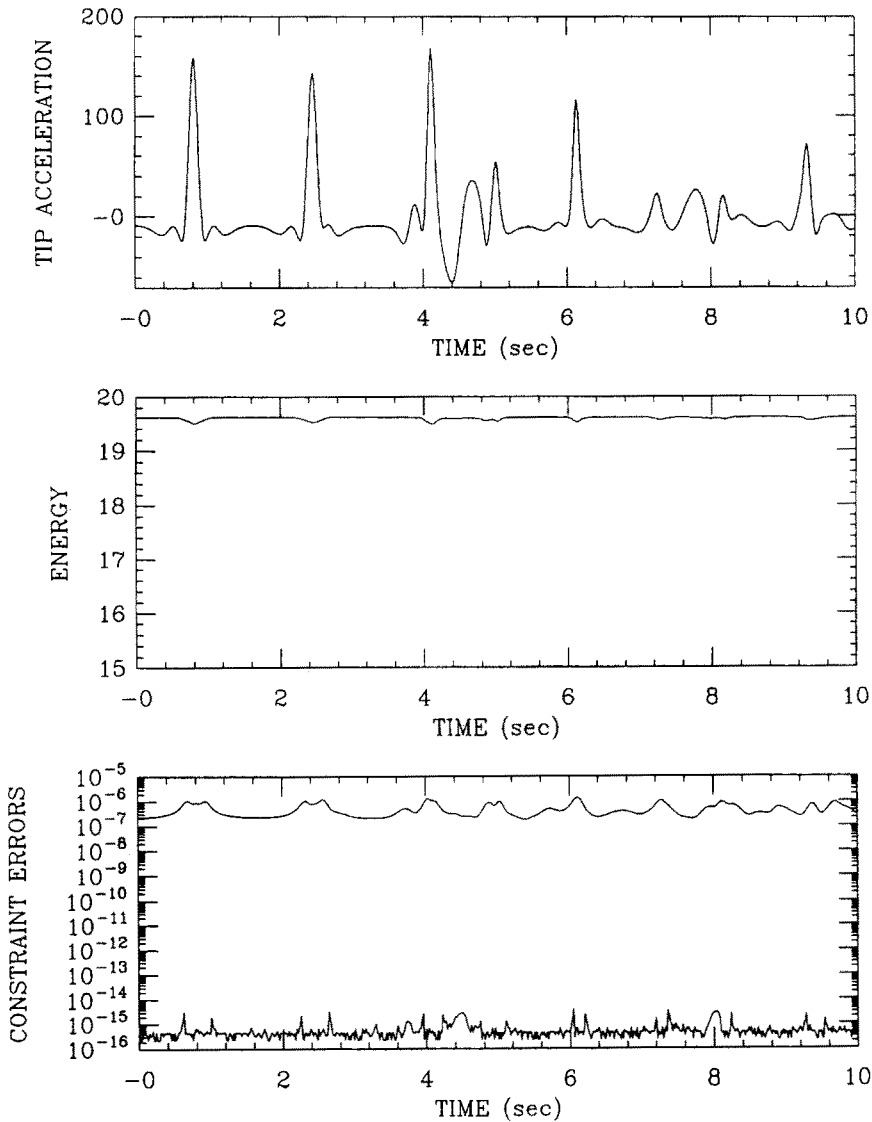


Fig. 6. Five-bar linkage: tip acceleration (top), energy (middle) and constraint errors (bottom).

elements of the mechanisms are illustrated in Figure 7. Then one of the elements is substituted by a very stiff spring with constant equal to  $10^8$ , thus adding additional numerical stiffness to the problem. The simulation is performed with no additional computational cost and yields the same results. All the constraint errors in position, velocities and accelerations are kept below  $10^{-14}$ .

## 6. Conclusions

We have presented in this paper a new method for multibody dynamics based on a combination of the augmented Lagrangian formulation and mass-orthogonal projections. We have shown through numerical simulation that the method yields a robust and efficient integration of the equations of motion. The new method maintains all the advantages of the augmented



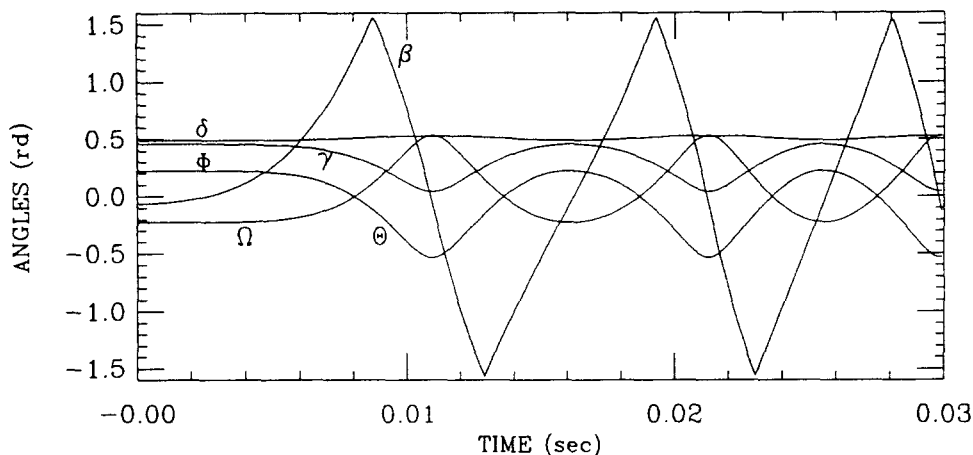


Fig. 7. Time history of joint angles in Andrew's squeezing mechanism.

Lagrangian method such as: robustness with redundant constraints, positive definite and symmetric leading matrices even in changing configurations and singular positions; and easiness of implementation. In addition, it yields a much more accurate integration, as well as constraint satisfaction to machine precision.

The mass-orthogonal projections are performed by a minimization process that eliminates the errors in  $\ddot{\Phi}$ ,  $\dot{\Phi}$  and  $\Phi$  (constraint cleaning). This process requires a reduced computational effort since the matrices used during the dynamic analysis are also used during the projections. Consequently, new matrix formation and triangularization for the solution of unknowns is kept to a minimum.

The simulations illustrate the best performance/price ratio is obtained when the projections are made at every time step. The method shows good convergence characteristics under strong motions, for very stiff systems and under singular configurations.

### Acknowledgement

The partial support of this work provided by the AFOSR under grant F496200-91-C-0095 is greatly acknowledged.

### References

1. Baumgarte, J., 'Stabilization of constraints and integrals of motion in dynamical systems', *Computer Methods in Applied Mechanics and Engineering* **1**, 1972, 1-16.
2. Park, K. C. and Chiou, J. C., 'Stabilization of computational procedures for constrained dynamical systems', *Journal of Guidance, Control, and Dynamics* **11**, 1988, 365-370.
3. Bayo, E., García de Jalón, J., and Serna, M. A., 'A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems', *Computer Methods in Applied Mechanics and Engineering* **71**, 1988, 183-195.
4. Kurdila, A. J., Junkins, J. L., and Hsu, S., 'Lyapunov stable penalty methods for imposing nonholonomic constraints in multibody system dynamics', *Nonlinear Dynamics* (to appear).
5. Bayo, E. and Avello, A., 'Singularity free augmented Lagrangian algorithms for constraint multibody dynamics', *Nonlinear Dynamics* **5**, 1994, 209-231.
6. Wehage, R. A. and Haug, E. J., 'Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems', *Journal of Mechanical Design* **104**, 1982, 247-255.
7. Kane, T. R. and Levinson, D. A., *Dynamics: Theory and Applications*, McGraw-Hill, New York, 1985.
8. Huston, R. L., *Multibody Dynamics*, Butterworth-Heinemann, New York, 1990.

9. Serna, M. A., Avilés, R., and García de Jalón, J., 'Dynamic analysis of planar mechanisms with lower-pairs in basic coordinates', *Mechanism and Machine Theory* **17**, 1982, 397–403.
10. Jerkovsky, W., 'The structure of multibody dynamic equations', *Journal of Guidance and Control* **1**, 1978, 173–182.
11. Kim, S. S. and Vanderploeg, M. J., 'A general and efficient method for dynamic analysis of mechanical systems using velocity transformations', *Journal of Mechanisms, Transmissions and Automation in Design* **108**, 1986, 176–182.
12. Nikravesh, P. E. and Gim, G., 'Systematic construction of the equations of motion for multibody systems containing closed kinematic loops', *Advances in Design Automation* **3**, 1989, 27–33.
13. García de Jalón, J., Avello, A., Jiménez, J. M., Martín, F., and Cuadrado, J., 'Real-time simulation of complex 3-D multibody systems with realistic graphics', in *Real-Time Integration Methods for Mechanical System Simulation*, NATO ASI Series, Vol. 69, Springer-Verlag, Heidelberg, 1990, pp. 265–292.
14. Bae, D.-S. and Won, Y. S., 'A Hamiltonian equation of motion for real time vehicle simulation', *Advances in Design Automation* **2**, 1990, 151–157.
15. Avello, A., Jiménez, J. M., Bayo, E., and García de Jalón, J., 'A simple and highly parallelizable method for real-time dynamic simulation based on velocity transformations', *Computer Methods in Applied Mechanics and Engineering* **107**, 1993, 313–339.
16. Brenan, K. E., Campbell, S. L., and Petzold, L. R., *The Numerical Solution of Initial Value Problems in Differential-Algebraic Equations*, Elsevier, New York, 1989.
17. Griepentrog, E., Hanke, M., and Marz, R., 'Berlin seminar on differential algebraic equations', Fachbereich Mathematik der Humboldt-Universität z Berlin, 1992.
18. Fuhrer, C. and Schwertassek, R., 'Generation and solution of multibody systems equations', *International Journal of Non-Linear Mechanics* **25**, 1990, 127–141.
19. Fuhrer, C. and Leimkuhler, B. J., 'Numerical solution of differential algebraic equations for constrained mechanical motion', *Numerische Mathematik* **59**, 1991, 55–69.
20. Matsson, S. E. and Soderlind, G., 'Index reduction in differential-algebraic equations using dummy derivatives', *SIAM Journal of Scientific Computation* **14**, 1993, 667–692.
21. Shampine, L. F. and Gordon, M. K., *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*, W. J. Freeman, San Francisco, CA, 1975.
22. García de Jalón, J. and Bayo, E., *Kinematic and Dynamic Simulation of Multibody Systems*, Springer-Verlag, New York, 1994.
23. Sanz-Serna, J. M., 'Symplectic integrators for Hamiltonian problems: An overview', in *Acta Numerica 1992*, Cambridge University Press, Cambridge, England, 1992, pp. 243–286.
24. Schiehlen, W. O., *Multibody System Handbook*, Springer-Verlag, Heidelberg, 1990.