Review of Contemporary Approaches for Constraint Enforcement in Multibody Systems *

Olivier A. Bauchau,
Daniel Guggenheim School of Aerospace Engineering,
Georgia Institute of Technology, Atlanta, GA, USA,

and André Laulusa
SIMUDEC Pte Ltd,
Singapore Science Park II, Singapore 117628.

Abstract

A hallmark of multibody dynamics is that most formulations involve a number of constraints. Typically, when redundant generalized coordinates are used, equations of motion are simpler to derive but constraint equations are present. Approaches to dealing with high index differential algebraic equations, based on index reduction techniques, are reviewed and discussed. Constraint violation stabilization techniques that have been developed to control constraint drift are also reviewed. These techniques are used in conjunction with algorithms that do not exactly enforce the constraints. Control theory forms the basis for a number of these methods. Penalty based techniques have also been developed, but the augmented Lagrangian formulation presents a more solid theoretical foundation. In contrast to constraint violation stabilization techniques, constraint violation elimination techniques enforce exact satisfaction of the constraints, at least to machine accuracy. Finally, as the finite element method has gained popularity for the solution of multibody systems, new techniques for the enforcement of constraints have been developed in that framework. The goal of this paper is to review the features of these methods, assess their accuracy and efficiency, underline the relationship among the methods, and recommend approaches that seem to perform better than others.

Keywords: Constrained multibody dynamics; Constraint conditions; Numerical methods; Differential/Algebraic equations.

1 Introduction

The theoretical foundations and classical approaches to the enforcement of constraints in multibody systems were presented by Laulusa and Bauchau [1]. The goal of the present review paper is to focus on the features of more recent techniques used to enforce constraints in multibody systems, assess their accuracy and efficiency, underline the relationship among the methods, and recommend approaches that seem to perform better than others.

An alternative approach to dealing with high index differential algebraic equations (DAEs) is the use of index reduction techniques, as discussed in section 2. Many of the proposed methods

* *Journal of Computational and Nonlinear Dynamics*, **3**(1), pp 011005 1-8, 2008.

do not enforce constraints exactly, rather, constraints are allowed to "drift" in time; this problem gave rise to the development of many constraint violation stabilization techniques that alleviate this problem, see section 3, as well as constraint violation elimination techniques that completely circumvent this problem, see section 4. Finally, in recent years, the finite element method has played an increasingly important role in multibody dynamics formulations and the tools and techniques used in this framework are the subject of section 5.

In this paper, a subscript $(\cdot)_a$ will be used to refer to equations, figure, sections or citations in ref. [1].

2 Index reduction techniques

Index reduction techniques are typically presented as mathematical processes that reduce the index of DAEs. Numerical analysis techniques are then used to prove that the application of specific types of time integrators to the reduced order DAEs provides a reliable solution of the problem. Gear *et al.* $[14]_a$ proposed a method, called the *stabilized index-2* or *GGL* method, that reduces the index from 3 to 2 and showed that variable-order, variable-step backward difference methods (BDF) converge for the resulting index-2 problem. Later, Gear [2] developed an approach to further reduce the problem to index-1 DAEs. These approaches imply additional computational cost in the form of additional Lagrange multipliers to be solved for.

Lötstedt and Petzold $[3]_a$ proved that k^{th}-order, constant step size BDF converge when applied to index-1, -2, or -3 DAEs; the numerical solution is accurate to order $\mathcal{O}(h^k)$, where h is the time step size. The same authors $[4]_a$ further investigated the practical difficulties of implementing variables step size integration methods for the same types of problems. The difficulties associated with the solution of index-3 DAEs were underlined: the condition number of the Newton iteration matrix, *i.e.* the tangent matrix used to solve the discretized nonlinear algebraic equations is $\mathcal{O}(h^{-3})$, resulting in increasingly ill conditioned problems for decreasing time step sizes. This conditioning problem can be partially remedied by scaling the constraint equations.

Eich [3] also provided a convergence analysis for a coordinate projection approach combined with BDF methods to integrate index-1 DAEs. The approach projects the numerical solution of the underlying ordinary differential equations (ODEs) onto the position and velocity invariants to reduce constraint violations. The accuracy of the projected solution was shown to be identical to that of BDF methods applied to the ODEs. For linear systems, it was shown that only the errors lying in the invariants were propagated, rendering the solution more accurate.

For holonomic systems, Yen *et al.* [4, 5] reduced the index-1 DAEs to ODEs by means of local parametrizations. The ODEs, which are similar to Boltzmann-Hamel equations, feature *local parameters* that implicitly define independent generalized coordinates and speeds. Then, using the local parametrization mapping and the constraint equations, the original generalized coordinates and velocities are recovered. A convergence analysis is presented that demonstrate an $\mathcal{O}(h^k)$ accuracy when k^{th}-order linear multistep methods are used. In numerical applications, local parameterizations were obtained using the generalized coordinate partitioning and tangent space methods. Local parameters, determined by the tangent space based on QR decomposition, were also used by Potra and Yen [6]. Similarly, Haug and Yen [7] determined local parameters using the generalized coordinate partitioning technique. Based on BDF methods, the discretized Boltzmann-Hamel equations were shown to be equivalent to a set of discretized DAEs, involving the constraints at the displacement, velocity and acceleration levels. For practical applications, these DAEs were used.

Yen *et al.* [8, 9] introduced the "coordinate-split formulation" for the solution of the index-2 DAEs characteristic of flexible multibody systems. A family of second order α-methods with user controllable numerical dissipation was proposed, which extend the corresponding methods used for ODEs in structural dynamics, the HHT algorithm of Hilber *et al.* [10] and the generalized-α algo-

rithm of Chung and Hulbert [11]. The coordinate-split method is a numerical implementation of the null space formulation (NSF) applied to the stabilized index-2 approach of Gear *et al.* $[14]_a$; it eliminates the two sets of Lagrange multipliers associated with this approach. Mass projections were used to impose the constraints at both position and velocity levels. To deal with the highly oscillatory nature of the response of flexible multibody systems, the authors introduced a modification of the Newton iteration process, denoted "modified coordinate-split iteration." Improved convergence was proved mathematically and demonstrated by means of examples.

For holonomic systems, Tseng *et al.* [12] devised an algorithm, called "Maggi's equations with perturbation iteration," which further develops the modified coordinate-split iteration by perturbing the solution that is projected onto the constraint manifold to eliminate constraint violations. In this approach, the determination of the generalized accelerations, velocities and displacements is separated from that of the Lagrange multipliers, which are recovered in a post-processing operation. Good numerical results were obtained, although the authors stressed the need for further validation of the approach, especially in the presence of flexible bodies. Note that the authors considered the coordinate-split formulation to be a numerical implementation of Maggi's equations, but in the classification introduced herein, this approach belongs to NSFs.

Another approach to index reduction is the embedded projection method developed by Borri *et al.* [13], which can be used to systematically reduce the index of the original DAEs system from 3 to 1. Furthermore, the method splits the original problem into its algebraic and differential parts, which can then be solved sequentially. While the accuracy and robustness of the procedure were demonstrated, its complexity is also apparent.

Parczewski and Blajer [14, 15] investigated systems subjected to *program constraints*, *i.e.* systems forced to follow a prescribed path. The control forces that impose the prescribed motion might have components in directions both tangential and orthogonal to the constraint manifold. This feature of control problems contrasts with the classical theory of constrained dynamics, for which constraint forces are acting in the direction normal to the constraint manifold. A classification of program constraint realizations was developed, which includes both orthogonal and tangent realizations, involving normal and tangent control forces, respectively. The authors provided several examples of program constraint realizations and determined the associated control forces; the difficulties inherent to non orthogonal realizations were underlined. Constraint forces that have components in directions both tangential and orthogonal to the constraint manifold are said to be "non-ideal" constraint forces. Udwadia and Kalaba $[29]_a$ also studied such systems and gave explicit expressions for both ideal and non-ideal constraint forces in terms of Moore-Penrose generalized inverses.

3 Constraint violation stabilization techniques

A number of techniques impose the constraints at the acceleration level, as is the case for the index-1 formulation, null space formulation, or Udwadia-Kalaba's formulation (UKF). Considering holonomic constraints, let $\underline{\mathcal{C}} = 0$, $\underline{\dot{\mathcal{C}}} = 0$ and $\underline{\ddot{\mathcal{C}}} = 0$ represent the displacement, velocity and acceleration level constraints, respectively. The system consisting of the equations of motion and the acceleration level constraints then forms a set of index-1 DAEs *with invariants*. Indeed, for the exact solution, $\underline{\mathcal{C}} = 0$ and $\underline{\dot{\mathcal{C}}} = 0$ represent two invariants of the system.

Unfortunately, due to numerical approximations and round-off errors, numerical solutions will not evolve along the invariant manifolds, resulting in $\underline{\mathcal{C}} \neq 0$ and $\underline{\dot{\mathcal{C}}} \neq 0$. In other words, the constraints "drift away" from the invariant manifold. From a mathematical standpoint, equation $\underline{\ddot{\mathcal{C}}} = 0$ is not stable since its poles are located at the origin of the s-plane, where s is the variable of Laplace's transform; consequently, $\underline{\mathcal{C}}$ and $\underline{\dot{\mathcal{C}}}$ will not converge to zero if any deviation occurs. The constraint violation stabilization techniques presented in this section attempt to minimize or

eliminate this drift; they are not, *per se*, solution methods for constrained dynamical problems, but rather, are used in conjunction with various solution techniques that are sensitive to the drift phenomenon.

If the mechanical system is conservative, the total mechanical energy, E, is an additional invariant of the system, $\dot{E} = 0$. If nonconservative forces are externally applied, the work they perform can be added to the total mechanical energy to form an invariant of the system. Likewise, the solution will drift away from this manifold, *i.e.* the total mechanical energy will not be preserved. In fact, the energy preservation constraint is a particular case of a nonholonomic constraint $\underline{\mathcal{D}}(\underline{q}, \underline{\dot{q}}) = 0$.

3.1 Control theory based stabilization techniques

The most popular stabilization technique probably is Baumgarte's method, which can be interpreted within the framework of control theory. Several researchers improved Baumgarte's original method and these efforts are described below.

3.1.1 Baumgarte's constraint violation stabilization

Baumgarte [16] introduced a stabilization method in which the original acceleration level constraint, $\underline{\ddot{\mathcal{C}}} = 0$, is replaced by

$$\underline{\ddot{\mathcal{C}}} + 2\alpha\underline{\dot{\mathcal{C}}} + \beta^2\underline{\mathcal{C}} = 0, \tag{1}$$

where α and β are user defined, positive parameters. In practical implementation, the choice $\alpha = \beta$ is often appropriate because critical damping is achieved. In the case of nonholonomic constraints, see eq. $(2)_a$, the velocity level constraint, $\underline{\mathcal{D}}(\underline{\dot{q}}, \underline{q}, t) = B(\underline{q}, t)\ \underline{\dot{q}} + \underline{b}(\underline{q}, t)$, is replaced by

$$\underline{\dot{\mathcal{D}}} + \gamma\underline{\mathcal{D}} = 0, \tag{2}$$

where γ is a user defined, positive parameter. The total energy constraint was treated in a similar manner. Baumgarte's stabilization method has been very widely used in multibody dynamics because it is easily implemented in conjunction with a variety of formulations of the equations of motion and time integration procedures. Ostermeyer [17] explained the effects of Baumgarte's stabilization method within the framework of control theory.

Unfortunately, parameters α and β are problem dependent, and no general procedure exists for their determination; hence, the approach tends to be unreliable and cannot be recommended for general purpose use in multibody dynamics because the constraints are never exactly satisfied. Eich and Hanke [18] mention that: "Choosing α and β too large results in stiff ODEs and a great amount of computing time." Nevertheless, some authors reported successful computations with Baumgarte's method. For instance, Nikravesh *et al.* [19] found an index-1 formulation in conjunction with Baumgarte's stabilization to be significantly more efficient computationally than the coordinate partitioning approach. They mention that: "Experience has shown that for most practical problems, positive values less than 5 for α and β are adequate. When $\alpha = \beta$, critical damping is achieved, which usually provides the fastest error reduction." A number of researchers have used various approaches to estimate these parameters in an effort to improve the reliability of Baumgarte's method, see section 3.1.2.

Park and Haug [20] have combined Baumgarte's stabilization method with the generalized coordinate partitioning method and shown that this hybrid approach outperforms both methods applied individually. They mention that: "Thus, the constraint stabilization method alone cannot handle every situation accurately and efficiently." Their rational for this conclusion is that the choice of α and β at each integration step is difficult and expensive, and erroneous solutions can appear when the constraint matrix is nearly singular.

3.1.2 Improvements of Baumgarte's stabilization method

Chang and Nikravesh [21] proposed an approach to adaptively determine the damping coefficient as the simulation proceeds. They assumed $\alpha = \beta$ and used adaptive control concepts to estimate optimal damping coefficients that are different for each constraint. Numerical examples demonstrate the better control of constraint violations achieved with the adaptive approach. Another improvement of Baumgarte's method was proposed by Ostermeyer [17] who added to eq. (1) a term involving the time integral of the constraint violation, based on optimum control theory.

Bae and Yang [22] also proposed an approach to the evaluation of the stabilization parameters. First, they replaced eq. (1) by $\ddot{\underline{\mathcal{C}}} + \alpha\dot{\underline{\mathcal{C}}} + \alpha\underline{\mathcal{C}} = 0$, where α represents the magnitude of the penalty factor for both position and velocity constraint violations, arguing that both violations are equally undesirable. Larger values of α will yield smaller constraint violations. However, if α is too large, the system becomes unstable. Hence, the value of α is limited by the stability characteristics of the numerical procedure used to integrate the equations of motion, the Adams-Bashforth integrator was used in this work. This condition yields a closed form expression for α as a function of the time step size and stability boundaries of the integrator.

A similar study was undertaken by Yoon *et al.* [23] who showed that under suitable assumptions, the constraint equation, written as $\ddot{\underline{\mathcal{C}}} + \alpha\dot{\underline{\mathcal{C}}} + \beta\underline{\mathcal{C}} = d$, where d represent the disturbances due to truncation errors, is indirectly integrated with the same numerical scheme as that used for the dynamics equations. This enables a rigorous study of the accuracy and stability characteristics of Baumgarte's method to be performed. However, in view of the complexity of the analysis, results were only shown for one case, the simple pendulum. The authors also pointed out the importance of stabilizing the energy preservation constraint.

Based on the input-output feedback linearization technique, Chiou and Wu [24] transformed the nonlinear governing DAEs into a set of linear equations. Next, they showed that a pole placement technique leads to Baumgarte's method and proposed a new approach to stabilization based on the variable structure control technique. While they demonstrated the superiority of their approach over Baumgarte's method by means of examples, no guidelines were provided on how to select the constants appearing in either approach.

Control theory concepts are also the basis for Lin and Hong's [25] stability analysis of Baumgarte's method using digital control theory. They notice that selecting α and β to be positive numbers is not sufficient to guarantee convergence of $\underline{\mathcal{C}}$ and $\dot{\underline{\mathcal{C}}}$ to zero as implied by stability analysis applied to eq. (1). Hence, they performed a stability analysis of the *discretized equations* using the Z-transform concept. They defined two parameters, $\hat{\alpha} = \alpha/h$ and $\hat{\beta} = \beta/h^2$, and concluded that while $\hat{\alpha}$ and $\hat{\beta}$ are independent of the problem and time step size, they do depend on the time integration scheme used for the simulation. Desirable values of $\hat{\alpha}$ and $\hat{\beta}$ were given for the Adams-Bashforth and Adams-Moulton predictor-corrector integrators.

3.2 Penalty based stabilization techniques

In penalty formulations, constraints are enforced by means of a *penalty term* added to the Lagrangian of the system, $1/2\ \underline{\mathcal{C}}^T\mathcal{P}\underline{\mathcal{C}}$, where $\mathcal{P} = \mathrm{diag}(p_i^2)$, and p_i are the *penalty factors*. It is common practice to use the same penalty factor, $p = p_i$, for all constraints and hence, the penalty term is often written as $1/2\ p^2\ \underline{\mathcal{C}}^T\underline{\mathcal{C}}$. The idea behind this formulation is to choose large penalty factors so as to drive the constraints to zero, *i.e.* $p \to \infty$ and $\underline{\mathcal{C}} \to 0$. Of course, in practical applications, a finite value of the penalty factor must be selected to avoid numerical ill conditioning and hence, the constraints are never exactly enforced and the quantities $\underline{\lambda} \approx p^2\underline{\mathcal{C}}$ approximate Lagrange's multipliers.

3.2.1 The staggered stabilization technique

Park and Chiou [26], derived a stabilization technique based on a penalty formulation. The Lagrange multipliers associated with holonomic constraints were written as $\underline{\lambda} = \underline{\mathcal{C}}(\underline{q}, t)/\epsilon$, where $\epsilon = 1/p^2$ is the penalty factor; time differentiation of this expression then leads to

$$\dot{\underline{\lambda}} = \frac{1}{\epsilon}\ (B\dot{\underline{q}} + \frac{\partial \underline{\mathcal{C}}}{\partial t}). \tag{3}$$

Taken together with eq. $(1)_a$, this equation forms a set of coupled ODEs. For nonholonomic constraints, a similar procedure can be followed by selecting Lagrange's multipliers as $\dot{\underline{\lambda}} = 1/\epsilon\ (B\ddot{\underline{q}} + \partial\underline{\mathcal{D}}/\partial t)$. Introducing governing eq. $(1)_a$ leads to $\epsilon\dot{\underline{\lambda}} + (BM^{-1}B^T)\underline{\lambda} = BM^{-1}\underline{F} + \partial\underline{\mathcal{D}}/\partial t$. If Lagrange's multipliers are written as $\underline{\lambda} = \bar{\underline{\lambda}}\ \exp(\sigma t)$, the homogeneous part of this equations becomes $(\sigma + BM^{-1}B^T/\epsilon)\bar{\underline{\lambda}} = 0$. This implies that the constraint decay rates, σ_i, are the eigenvalues of matrix $BM^{-1}B^T/\epsilon$; in other words, the decay rates are a function of the physical characteristics of the system, in contrast with Baumgarte's method that depends on abstract coefficients unrelated to system properties.

Note that a single derivative of the constraints was taken, hence this approach will be less sensitive to the drift phenomenon than methods requiring two time derivatives. Furthermore, it depends on a single coefficient, the penalty factor. Examples showed improved accuracy for displacement level constraint invariants as compared to the results of Baumgarte's method [26]. This stabilized technique is robust as it can accommodate nearly rank deficient constraint matrices, while Baumgarte's technique cannot.

Park *et al.* $[69]_a$ presented an *explicit-implicit, staggered* procedure to implement the stabilization procedure described in the previous paragraph. The approach calls for the developments of two distinct modules, one integrates the generalized coordinates knowing the constraint forces, the other integrates the Lagrange multipliers knowing the generalized coordinates. Calls to the two modules alternate, hence the approach is called a "staggered procedure." Examples were given, demonstrating the accuracy and effectiveness of the procedure, which is robust but conditionally stable.

3.2.2 Augmented Lagrangian formulation

The augmented Lagrangian formulation (ALF) developed by Bayo *et al.* [27] starts as a penalty formulation of the problem. Corresponding to the k^{th} holonomic constraint, the following terms are added to the Lagrangian of the system: a penalty term, $1/2\ \alpha_k\omega_k\mathcal{C}_k^2$, a Rayleigh dissipative forces terms, $-2\ \alpha_k\omega_k\mu_k\dot{\mathcal{C}}_k$, and a fictitious kinetic energy term, $1/2\ \alpha_k\dot{\mathcal{C}}_k^2$. The governing equations of the system now become index-1 equations

$$M\ddot{\underline{q}} + B^T\alpha(\ddot{\underline{\mathcal{C}}} + 2\Omega\mu\dot{\underline{\mathcal{C}}} + \Omega^2\underline{\mathcal{C}}) = \underline{F}, \tag{4}$$

where $\alpha = \text{diag}(\alpha_k)$, $\Omega = \text{diag}(\omega_k)$ and $\mu = \text{diag}(\mu_k)$. This penalty formulation will only yield accurate predictions for large penalty factors, $\alpha_k \to \infty$; the coefficients ω_k and μ_k play a stabilizing role similar to that of the corresponding coefficients of Baumgarte's method. In the ALF, a set of Lagrange multipliers is introduced together with the penalty terms to yield

$$M\ddot{\underline{q}} + B^T\alpha(\ddot{\underline{\mathcal{C}}} + 2\Omega\mu\dot{\underline{\mathcal{C}}} + \Omega^2\underline{\mathcal{C}}) = \underline{F} - B^T\underline{\lambda}^*. \tag{5}$$

Had the sole Lagrange multipliers been introduced, the governing equations would have been $M\ddot{\underline{q}} = \underline{F} - B^T\underline{\lambda}$, and hence, $\underline{\lambda} = \underline{\lambda}^* + \alpha(\ddot{\underline{\mathcal{C}}} + 2\Omega\mu\dot{\underline{\mathcal{C}}} + \Omega^2\underline{\mathcal{C}})$. Since the Lagrange multipliers are sufficient, *per se*, to impose the constraints, the penalty coefficient is no longer required to be large; however,

the formulation now involves m additional unknowns. In the proposed approach, the Lagrange multipliers, $\underline{\lambda}^*$, are not treated as unknowns; rather, they are computed through an iterative process

$$\underline{\lambda}^{*(i+1)} = \underline{\lambda}^{*(i)} + \alpha(\underline{\ddot{\mathcal{C}}} + 2\Omega\mu\underline{\dot{\mathcal{C}}} + \Omega^2\underline{\mathcal{C}})^{(i+1)}, \tag{6}$$

where the superscript $(\cdot)^{(i)}$ indicates the iteration number and $\underline{\lambda}^{*(0)} = 0$. Combining this iterative scheme with eq. (5) leads to

$$(M + B^T\alpha B)\underline{\ddot{q}}^{(i+1)} = M\underline{\ddot{q}}^{(i)} - B^T\alpha(\dot{B}\underline{\dot{q}} + 2\Omega\mu\underline{\dot{\mathcal{C}}} + \Omega^2\underline{\mathcal{C}}), \tag{7}$$

where $M\underline{\ddot{q}}^{(0)} = \underline{F}$.

The ALF reduces the problem to a set of ODEs with no additional unknowns. The iterative solution of the Lagrange multipliers is inexpensive since iterations are already required for the solution of the nonlinear equations of motion. Numerical experimentation shows that accurate solutions can be obtained for a wide range of penalty factors, $\alpha_k \in [10^3, 10^9]$. The formulation can be generalized to accommodate nonholonomic constraints.

Bayo *et al.* [28] further elaborated on the ALF. The penalty term was simplified to keep two terms only, resulting in $\underline{\lambda} = \underline{\lambda}^* + \alpha(\underline{\mathcal{C}} + \mu\underline{\dot{\mathcal{C}}})$. They observed that the velocity level constraint in the penalty factor was necessary to prevent the appearance of high frequency numerical oscillations during the simulation. To integrate the equations of motion, the trapezoidal rule was used with accelerations or displacements as primary variables: the latter were shown to provide superior performance.

4 Constraint violation elimination techniques

In contrast to the constraint violation stabilization techniques presented in the previous section, constraint violation elimination techniques are method which result in satisfaction of the constraints within machine accuracy.

4.1 Geometric projection approach to stabilization

Yoon *et al.* [29] developed another approach to constraint violation stabilization. Let $\underline{\bar{q}}_n$ and $\underline{\bar{v}}_n$ be the generalized coordinates and velocities, respectively, predicted by the integration of the equations of motion at the end of time step n. In general, $\underline{\mathcal{C}}(\underline{\bar{q}}_n, t_n) \neq 0$ and $\underline{\mathcal{D}}(\underline{\bar{q}}_n, \underline{\bar{v}}_n, t_n) \neq 0$. The approach consists in correcting or perturbing the generalized coordinates and velocities, $\underline{q}_n = \underline{\bar{q}}_n + \underline{\hat{q}}_n$, and $\underline{v}_n = \underline{\bar{v}}_n + \underline{\hat{v}}_n$, respectively, where $\underline{\hat{q}}_n$ and $\underline{\hat{v}}_n$ are the unknown coordinate and velocity corrections, respectively, both assumed to be small. The updated coordinates, $\underline{q}_n$, and velocities, $\underline{v}_n$, satisfy the holonomic and nonholonomic constraints, *i.e.* $\underline{\mathcal{C}}(\underline{q}_n, t_n) = 0$ and $\underline{\mathcal{D}}(\underline{q}_n, \underline{v}_n, t_n) = 0$, respectively.

At first, the generalized coordinate corrections are evaluated by linearizing the holonomic constraints to find $B(\underline{\bar{q}}_n, t_n)\ \underline{\hat{q}}_n \approx -\underline{\mathcal{C}}(\underline{\bar{q}}_n, t_n)$. Since these equations are overdetermined, it is assumed that the corrections lie in the subspace defined by the constraint matrix, *i.e.* $\underline{\hat{q}}_n = B^T\underline{\epsilon}_n$, where $\underline{\epsilon}_n$ is an unknown array. It then follows that $\underline{\epsilon}_n = -(BB^T)^{-1}\underline{\mathcal{C}}(\underline{q}_n, t_n)$, and finally

$$\underline{\hat{q}}_n = -B^T(BB^T)^{-1}\underline{\mathcal{C}}(\underline{\bar{q}}_n, t_n). \tag{8}$$

Next, the generalized velocity corrections are evaluated by linearizing the nonholonomic constraints to find $B(\underline{q}_n, \underline{\bar{v}}_n, t_n)\ \underline{\hat{v}}_n \approx -\underline{\mathcal{D}}(\underline{q}_n, \underline{\bar{v}}_n, t_n)$. In this second phase, the generalized coordinates are kept constant, since they were corrected in the first phase of the procedure. Here again, these equations are overdetermined and it is assumed that $\underline{\hat{v}}_n = B^T\underline{\gamma}_n$. It then follows that $\underline{\gamma}_n = -(BB^T)^{-1}\underline{\mathcal{D}}(\underline{q}_n, \underline{\bar{v}}_n, t_n)$, and finally

$$\underline{\hat{v}}_n = -B^T(BB^T)^{-1}\underline{\mathcal{D}}(\underline{q}_n, \underline{\bar{v}}_n, t_n). \tag{9}$$

It should be noted that the procedure alleviates constraint violations without modifying the equations of motion, in contrast with Baumgarte's method. The approach is geometric in nature: $\Gamma^T(\underline{q}_n - \bar{\underline{q}}_n) = \Gamma^T B^T \underline{\epsilon}_n = 0$, the corrected solution is a projection of the approximate solution onto the constraint manifold. Clearly, the geometric procedure alleviates the constraint violations without eliminating them; complete elimination would require an iterative solution of the constraint equations. Yoon *et al.* [29] demonstrated the effectiveness of the procedure for holonomic, nonholonomic and energy constraints.

Blajer $[41]_a$, [30] developed a similar approach to constraint elimination based on the geometric interpretation of constrained dynamics he presented with his coworkers in ref. $[73]_a$. Based on geometric arguments, the following correction schemes are found for the generalized coordinates and velocities,

$$\hat{\underline{q}}_n = -M^{-1}B^T(BM^{-1}B^T)^{-1}\underline{\mathcal{C}}(\bar{\underline{q}}_n, t_n), \tag{10a}$$

$$\hat{\underline{v}}_n = -M^{-1}B^T(BM^{-1}B^T)^{-1}\underline{\mathcal{D}}(\underline{q}_n, \bar{\underline{v}}_n, t_n), \tag{10b}$$

respectively. Blajer's corrections, eqs. (10a) and (10b), are more physically consistent than Yoon's, eqs. (8) and (9), respectively, because the matrix $(BM^{-1}B^T)$ involves terms that are of consistent units whereas matrix (BB^T) does not. Indeed, when generalized coordinates have different units, such as displacements and rotations, matrix (BB^T) weighs all components equally; in contrast, matrix $(BM^{-1}B^T)$ weighs each component by an appropriate inertial term. In numerical applications, the position corrections, eq. (10a), are iterated on first, until constraint violations are completely eliminated, *i.e.* until the constraint equations are satisfied to machine accuracy. If nonholonomic constraints are present, the same process is applied to correct the velocities.

Baumgarte [31] developed a new stabilization method, which is derived from a modified statement of Hamilton's principle. The resulting equations of motion feature non classical Lagrangian multipliers and the holonomic constraints need to be differentiated only once with respect to time. Unfortunately, no applications were presented, making the assessment of the approach rather difficult.

Terze *et al.* [32] formulated a constraint elimination method within the framework of the NSF. Using the projective criterion defined by Blajer *et al.* $[73]_a$, they identified a set of independent variables. Displacement constraint violations were then iteratively eliminated by adjusting the sole dependent variables to satisfy the displacement level constraint equations. In a second step, the velocity constraint violations were eliminated using the velocity level constraint equations. During both correction steps, the independent displacements and velocities were kept unchanged.

4.2 The mass-orthogonal projection formulation

Bayo and Avello [33] proposed an ALF based on the canonical equations of Hamilton. Compared to the index-1 based formulation derived earlier by Bayo *et al.* [27], the new formulation exhibits better accuracy and robustness in the presence of singular configurations. The improved performance was credited to the fact that a single differentiation of the holonomic constraints is required in canonical formulations, rather than the double differentiation associated with index-1 formulations. Effectiveness of the new approach was illustrated by numerical examples. A similar formulation was derived for nonholonomic constraints although no example was given.

While the ALF presented in section 3.2.2 satisfies the weighted constraint, $\ddot{\underline{\mathcal{C}}} + 2\Omega\mu\dot{\underline{\mathcal{C}}} + \Omega^2\underline{\mathcal{C}}$, see eq. (4), to machine accuracy, individual constraints at the position, $\underline{\mathcal{C}} = 0$, velocity, $\dot{\underline{\mathcal{C}}} = 0$, and acceleration levels, $\ddot{\underline{\mathcal{C}}} = 0$, are not necessarily satisfied to the same level of accuracy. To improve this situation, Bayo and Ledesma [34] combined the ALF with a mass orthogonal projection technique. To impose the position level constraint, they propose to minimize $V = 1/2\ (\underline{q} - \underline{q}^*)M(\underline{q} - \underline{q}^*)$ subject to the constraint $\underline{\mathcal{C}}(\underline{q}, t) = 0$, where $\underline{q}^*$ is the solution obtained at the end of a time step using the

ALF. This minimization problem is itself solved using an ALF, transforming V into

$$V^* = \frac{1}{2}\,(\underline{q} - \underline{q}^*)^T M(\underline{q} - \underline{q}^*) + \frac{1}{2}\,\underline{\mathcal{C}}^T \alpha \underline{\mathcal{C}} + \underline{\mathcal{C}}^T \underline{\lambda}. \tag{11}$$

Simple algebraic manipulations lead to the following iterative scheme to impose the position constraint

$$(M + B^T \alpha B)\Delta^{(i+1)} = -M(\underline{q}^{(i)} - \underline{q}^*) - B^T \underline{\lambda}^{(i)} \tag{12}$$

where $\Delta^{(i+1)} = \underline{q}^{(i+1)} - \underline{q}^{(i)}$ and $\underline{\lambda}^{(i+1)} = \underline{\lambda}^{(i)} + \alpha \underline{\mathcal{C}}^{(i+1)}$. From a computational view point, this iterative procedure is not expensive because the system matrix, $(M + B^T \alpha B)$, is identical to that of eq. (7). Hence, this matrix is factorized once only and the additional computational cost consists of the evaluation of the right hand side of eq. (12) followed by forward reductions and backward substitutions. This contrasts with the approaches presented in section 4.1 that typically involve more computational effort. The constraints at the velocity and acceleration levels can be treated in a similar manner and are formulated in such a way that the resulting system matrix is identical to that of eq. (12), minimizing computational cost.

Bayo and Ledesma [34] illustrated their approach with several numerical examples. Application of the mass-orthogonal projection at each time step eliminates constraint violations to machine accuracy and dramatically increases the accuracy of the simulation. A mechanism presenting singular configurations was successfully simulated to demonstrate the robustness of the ALF. The trapezoidal rule was used to integrate the equations of motion with accelerations or displacements as primary variables: the latter were shown to provide superior performance.

4.2.1 Comparative studies

For holonomic systems, Schiehlen [35] derived governing DAEs and ODEs, the Boltzmann-Hamel equations, as well as equations of motion based on a recursive approach. The recursive approach, suitable for chain topologies, can be much more competitive than the ODE formulation, although for small numbers of dofs, the latter is still competitive as the former is rather complex. In general, recursive approaches require $\mathcal{O}(n)$ operations, in contrast with ODE formulations that may need up to $\mathcal{O}(n^3)$ arithmetic operations. A comparative study showed that the ODE formulations are more efficient than their DAEs counterpart, although this conclusion was based on very simple, rigid multibody systems examples involving very few dofs.

Cuadrado *et al.* [36] compared four methods that are used to simulate multibody dynamics with constraints. These methods are: the augmented Lagrangian formulation index-1 and index-3 with projections, ALF-1 and ALF-3, respectively, a modified state-space formulation (equations of motion in independent coordinates) and a fully recursive formulation. Modifications were performed to the classical state-space formulation to improve its performance in the presence of stiff systems. The ALF-1 and ALF-3 approaches used natural or fully Cartesian coordinates, as described in ref. [37]. These coordinates have the advantage of leading to a constant mass matrix.

A number of rigid multibody problems were solved with all four methods to compare their performance; none was found to be fully satisfactory. The ALF-3 approach failed to converge when using time step sizes smaller than 10^{-5} sec, while for time step sizes larger than 10^{-2} sec, the ALF-1 approach failed to converge. The space-state and the fully recursive formulation lacked robustness as they failed to handle singular configurations. In addition, the fully recursive formulation behaved poorly in the presence of stiff systems or systems presenting redundant constraints. Nevertheless, for non-stiff problems of large size, this method became competitive. Of all the methods tested, the ALF-3 formulation was the most efficient, while the ALF-1 was the most robust. The authors suggested that a combined index-1 and index-3 formulation would constitute a very good tool for solving multibody dynamics with constraints. However, further evaluation of the methods was

recommended, especially for large scale industrial problems. It is not clear how the various methods presented in this study would perform for elastic multibody systems.

5 Finite Element based techniques

Multibody dynamics analysis was originally developed as a tool for modeling mechanisms with simple tree-like topologies composed of rigid bodies, but has considerably evolved to the point where it can handle nonlinear flexible systems with arbitrary topologies. In finite element based multibody dynamics approaches, as described by Géradin and Cardona [38], a given mechanism is modeled by an idealization process that represents each component of the mechanism by an "element" chosen from an extensive library of elements implemented in the code.

Shabana [39] presented a thorough review of flexible multibody dynamics formulations. Traditionally, elasticity has been taken into account in an approximate manner using modal expansion techniques, *i.e.* using a small number of dofs. This contrasts with finite element formulations, which typically discretize 1-, 2-, or 3-D elasticity problems in terms of nodal coordinates consisting of displacements or rotations of points, *i.e.* using a large number of dofs. However, the distinction between the two approaches might sometimes become blurry; for instance, component mode synthesis [40, 41] is a powerful tool used in both multibody dynamics and finite element formulations.

Because of the large number of dofs involved in finite element formulations and the likely presence of high frequencies associated with the discretization process, time integration relies almost exclusively on implicit schemes; for linear systems, the HHT integrator [10], the workhorse used in most commercial codes, is second order accurate, unconditionally stable, and present high frequency numerical damping; these three features are considered indispensable for the successful integration of large finite element systems, see textbooks such as Hughes [42] or Bathe [43]. This contrasts with multibody formulations that tend to use explicit, predictor multi-corrector algorithms such as the Adams-Bashforth integrator [44]. Although of a much higher order of accuracy, this integrator is conditionally stable. Implicit integrators require the solution of a linear system at each time step, which dominates the cost of the analysis. Efficient solutions procedures make use of the sparsity of the system of equation which form enabling technique of the finite element method, see Bathe [43].

An important implication of these observations is that any formulation that destroys the sparsity of the system matrix generated by the finite element method is unlikely to be effective. For instance, applications of Maggi's formulation requires the computation of the null space of the constraint matrix. The various algorithms used to compute the null space, whether the *LU* factorization with pivoting, GSO algorithm, or SVD, all alter the band structure of the system matrix. The index-1 formulation, see eq. $(13a)_a$, requires the inverse of the mass matrix, another band destroying operation; of course, the NSF requires the evaluation of the null space; finally, the computation of the pseudo-inverse called for by UKF is once more an operation that does not preserve sparsity.

Clearly, far fewer methods are available for the effective enforcement of constraints when bandedness of the system matrix must be preserved, and hence, it should not come as a surprise that the sparsity based, index-3 DAEs formulation of Orlandea *et al.* $[33, 34]_a$, as discussed in section 3_a, has been used within the framework of finite element formulations. Penalty based stabilization techniques, as discussed in section 3.2, have also been used in this framework; in fact, the staggered stabilization technique of Park and Chiou [26] and the augmented Lagrangian formulation of Bayo *et al.* [27] were originally presented for finite element formulations.

5.1 Time stepping algorithms

In view of the difficulties associated with the solution of index-3 DAEs, considerable effort was devoted to the development of time integration techniques suitable for large finite element systems.

Cardona and Géradin [45, 46] showed that the classical Newmark [47] trapezoidal rule is unconditionally *unstable* for linear systems in the presence of constraints. However, the use of dissipative algorithms such as HHT [10] resulted stable behavior, even for nonlinear systems. Further work by Farhat *et al.* shows that both HHT and generalized-α [11] methods achieve stability for a class of constrained hybrid formulations. In these approaches, stabilization of the integration process is inherently associated with the dissipative nature of the algorithms. While stability is mathematically proven for linear systems, there is no guarantee when it comes to nonlinear systems [48].

To remedy this situation, considerable work has been done in recent years with energy preserving schemes. In these schemes, unconditional nonlinear stability is achieved by proving a discrete energy preserving statement, $E_f = E_i$, where E denotes the total mechanical energy of the system, and the subscripts $(\cdot)_i$ and $(\cdot)_f$ denote the value of the corresponding quantity at the initial and final times of the time step, respectively, denoted t_i and t_f, respectively. This algorithmic preservation property is a direct consequence of the specific discretization used for the inertial and elastic forces acting on the system. In view of the positive definite nature of the total mechanical energy, this discrete conservation law guarantees the stability of the computational scheme for nonlinear problems.

However, while nonlinear unconditional stability is the first step towards the development of robust algorithms, energy preserving schemes are not well suited for large finite element problems because high frequency oscillations, especially in the velocity and stress fields, can corrupt the computed system response, as observed by Bauchau *et al.* [49, 50]; furthermore, unrealistic energy transfer between modes can take place. Hence, the presence of high frequency numerical dissipation is an indispensable feature of robust time integrators for multibody systems, a fact that was already observed for linear systems and prompted the development of numerically dissipative algorithms such as the HHT [10] or generalized-α [11] methods. Numerically dissipative schemes that feature nonlinear unconditional stability can be developed by proving a discrete energy decay statement, $E_{i+1} = E_i - E_d$, where $E_d > 0$ is the energy dissipated within the time step.

5.2 Enforcement of the constraints

The main idea behind energy preserving and decaying algorithms is to develop discretizations of the equations of motion that imply the preservation of a known first integral of the motion, the total mechanical energy. When it comes to enforcement of the constraints, a similar path has been followed: the well known fact that the work done by the constraint forces must vanish is implemented at the algorithmic level [49, 51]. The work done by the constraint forces is $W^c = \int \underline{F}^{cT}\underline{\dot{q}}\ \mathrm{d}t = \int \underline{\lambda}^T B\underline{\dot{q}}\ \mathrm{d}t = \int \underline{\lambda}^T\underline{\dot{\mathcal{C}}}\ \mathrm{d}t$ and hence, the vanishing of this work is intimately linked to the vanishing of the constraint derivatives. This observation helps understand why it is important to enforce constraints at both displacement and velocity levels. Here again, it must be noted that an approximate solution of the constrained equations of motion will not necessarily imply the vanishing of the work done by the constraint forces at the algorithmic level. This provides a potential source of "numerical energy," which could destabilize the integration scheme.

Focusing on holonomic constraints, the following relationship is used to define the algorithmic constraint matrix, B_m, as

$$\underline{\mathcal{C}}_f - \underline{\mathcal{C}}_i = B_m(\underline{q}_f - \underline{q}_i), \tag{13}$$

where the subscript $(\cdot)_m$ indicates quantities evaluated at the midpoint of the time step. Note that the mean value theorem guarantees the existence of B_m. The discretized forces of constraint now become $\underline{F}^c_m = B^T_m\underline{\lambda}_m$, where $\underline{\lambda}_m$ are midpoint Lagrange's multipliers, and the work done by these discretized forces of constraint follows as $W^c = (\underline{q}_f - \underline{q}_i)^T\underline{F}^c_m = \underline{\lambda}^T_m B_m(\underline{q}_f - \underline{q}_i) = \underline{\lambda}^T_m(\underline{\mathcal{C}}_f - \underline{\mathcal{C}}_i)$. Clearly, the vanishing of the work done by the algorithmic forces of constraints implies $\Delta t\ \underline{\dot{\mathcal{C}}}_m = \underline{\mathcal{C}}_f - \underline{\mathcal{C}}_i = 0$, which echoes, at the algorithmic level, the condition required for the exact solution, $\underline{\dot{\mathcal{C}}} = 0$. Rather than imposing the condition $\underline{\mathcal{C}}_f - \underline{\mathcal{C}}_i = 0$, it is preferable to enforce $\underline{\mathcal{C}}_f = 0$ at each time step, to

avoid the drift phenomenon. Discretizations of numerous constraints that satisfy eq. (13) can be found in the following references [52, 53, 54, 55].

Gonzalez [56] formulated an integration scheme for solving the equations of motion of Hamiltonian system expressed in the form of DAEs. Holonomic constraints were considered. The numerical scheme was based on the notion of *discrete derivative*, which satisfied properties such as directionality, consistency and orthogonality; eq. (13) is an example of discrete derivative. The proposed scheme satisfies the constraints, and leads to the conservation of the Hamiltonian and linear and angular momenta, but constraints are not satisfied at the velocity level. Bauchau [57] showed that the approach to modeling constraints characterized by eq. (13) is closely related to the stabilized index-2 method of Gear *et al.* $[14]_a$, although no additional unknowns are required.

The approach summarized in the last two sections combines two algorithmic features: preservation/dissipation of energy and vanishing of the work done by the constraint forces. This provides a formal proof of numerical stability for the integration of nonlinear, flexible multibody systems, and constraints are enforced to machine accuracy, both at the displacement and velocity levels. The price to pay for these desirable features is that the discretization of inertial, elastic and constraint forces must be carefully crafted for each element type so that the preservation characteristics of the algorithms can be proved. This stands in sharp contrasts with the more traditional approach to multibody simulations that use a variety of formulations of the equations of motion, but rely on "black box" integration routine, which are designed for the solution of DAEs, but are otherwise unaware of the specific features and characteristics of the equations being solved.

5.3 The discrete null space approach

Betsch *et al.* [58, 59] have recently proposed an original method for the time integration of constrained dynamical systems, based on Maggi's formulation. In this approach, the index-3 DAEs are first discretized with an energy/momentum preserving scheme based on the algorithms of Gonzalez [56], and Betsch and Steimann [60]. Next, the discrete Lagrange multipliers are eliminated using a *discrete null space*: using the notation of eq. (13), the discrete null space, Γ_m, is the orthogonal complement of the discrete constraint matrix, B_m, such that $\Gamma_m^T B_m^T = 0$. As discussed in section 5.2, analytical expressions of the discrete constraint matrix can be derived for a wide range of constraints; the originality of proposed approach is to show that analytical expressions of the discrete null space can also be obtained for numerous constraints. This approach bypasses the need for the numerical evaluation of the null space using the many techniques described in sections 4.1_a and 4.2_a, and the associated numerical cost. Furthermore, the discrete null space is computed for each element of the system independently, and hence, can be used within the framework of finite element methods without harming the bandwidth of the system.

5.4 Scaling of the governing equations

Petzold and Lötstedt $[4]_a$ have shown that index-3 DAEs are severely ill conditioned for small time step sizes when using BDF formulas: unless corrective actions are taken, the condition number of the iteration matrix is $\mathcal{O}(h^{-3})$. Furthermore, errors in the displacement, velocity and multiplier fields are shown to propagate at rates of $\mathcal{O}(h^{-1})$, $\mathcal{O}(h^{-2})$ and $\mathcal{O}(h^{-3})$, respectively. A perturbation analysis by Arnold [61] indicated that errors and constraint violations grow very rapidly as the time step size is reduced, preventing the practical use of time refinement procedures, and imposing tight tolerances on the solution of the non-linear discrete equations.

Petzold and Lötstedt $[4]_a$ presented a simple scaling transformation of the DAEs that yields a condition number $\mathcal{O}(h^{-2})$ and an improvement of one order in the errors for all solution fields. Although the sensitivity to perturbations is reduced, numerical problems are still observed in practice. Their scaling, termed "left preconditioning," consists of dividing the constraint equations by

the time step size, while the dynamic equilibrium equations are multiplied by the same quantity. Bottasso *et al.* [62] proposed the addition of a "right preconditioning" which scales the unknowns of the problem: Lagrange multipliers are divided by the square of the time step size, while velocities are multiplied by the time step size. The combination of both preconditioning techniques leads to a remarkable result: both error propagation and conditioning are $\mathcal{O}(h^0)$, *i.e.* the behavior of the numerical solution of index-3 DAEs is identical to that of ODEs. Clearly, scaling can and should be used in conjunction with other techniques for the solution of DAEs: it is easily implemented, does not require a reformulation of the equations of motion, and does not introduce additional unknowns.

6 Conclusions

Index reduction techniques are formal mathematical procedures that reduce the index of DAEs from 3 to 2 or 1. In many cases, properties of the proposed schemes are proved mathematically. While the order of accuracy of the solution is often provided, the problem of violation of the constraints was rarely addressed.

Due to approximations and round-off errors, many numerical solutions do not satisfy the constraints exactly, a phenomenon known as "drift." Numerous constraint violation stabilization techniques have been developed to remedy this problem, but Baumgarte's method is probably the most widely used. It presents two shortcomings: first, constraints are never exactly satisfied, and second, no general procedure exists to determine the problem dependent parameters appearing in the formulation. Consequently, this approach cannot be recommended for general used in multibody systems.

Penalty based formulations have also been used to control the drift phenomenon. The augmented Lagrangian formulation is probably the most robust and efficient method of that type. Next, constraint violation elimination techniques have been developed to enforce satisfaction of the constraints to machine accuracy, typically by means of iterative process that projects the solution onto the constraint manifold.

Next, new algorithms have been developed for the enforcement of constraints within the framework of the finite element method. Based on the physical concepts of energy preservation and vanishing of the work done by the constraint forces, robust algorithms have been developed that present mathematical proofs of stability. Finally, scaling algorithms have recently been developed that completely eliminate the ill conditioning traditionally associated with index-3 DAEs.

References

[1] A. Laulusa and O.A. Bauchau. Review of classical approaches for constraint enforcement in multibody systems. *Journal of Computational and Nonlinear Dynamics*, 3(1):011004 1–8, January 2008.

[2] C.W. Gear. Differential-algebraic equation index transformations. *SIAM Journal on Scientific and Statistical Computing*, 9(1):40–47, January 1988.

[3] E. Eich. Convergence results for a coordinate projection method applied to mechanical systems with algebraic constraints. *SIAM Journal on Numerical Analysis*, 30(5):1467–1482, October 1993.

[4] J. Yen, E.J. Haug, and T.O. Tak. Numerical methods for constrained equations of motion in mechanical system dynamics. *Mechanics of Structures and Machines*, 19(1):41–76, 1991.

[5] J. Yen. Constrained equations of motion in multibody dynamics as ODEs on manifolds. *SIAM Journal on Numerical Analysis*, 30(2):553–568, 1993.

[6] F.A. Potra and J. Yen. Implicit numerical integration for euler-lagrange equations via tangent space parameterization. *Mechanics of Structures and Machines*, 19(1):77–98, 1991.

[7] E.J. Haug and J. Yen. Implicit numerical integration of constrained equations of motion via generalized coordinate partitioning. *ASME Journal of Mechanical Design*, 114:296–304, June 1992.

[8] J. Yen and L.R. Petzold. An efficient Newton-type iteration for the numerical solution of highly oscillatory constrained multibody dynamic systems. *SIAM Journal on Scientific Computing*, 19(5):1513–1534, September 1998.

[9] J. Yen, L.R. Petzold, and S. Raha. A time integration algorithm for flexible mechanism dynamics: The DAE α-method. *Computer Methods in Applied Mechanics and Engineering*, 158:341–355, 1998.

[10] H.M. Hilber, T.J.R. Hughes, and R.L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering and Structural Dynamics*, 5:283–292, 1977.

[11] J. Chung and G.M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-α method. *Journal of Applied Mechanics*, 60:371–375, 1993.

[12] F.C. Tseng, Z.D. Ma, and G.M. Hulbert. Efficient numerical solution of constrained multibody dynamics systems. *Computer Methods in Applied Mechanics and Engineering*, 192:439–472, 2003.

[13] M. Borri, L. Trainelli, and A. Croce. The embedded projection method: A general index reduction procedure for constrained system dynamics. *Computer Methods in Applied Mechanics and Engineering*, 195(50-51):6974–6992, 2006.

[14] J. Parczewski and W. Blajer. On realization of program constraints: Part I - Theory. *Journal of Applied Mechanics*, 56:676–679, September 1989.

[15] W. Blajer and J. Parczewski. On realization of program constraints. Part II - Practical implications. *Journal of Applied Mechanics*, 56:680–684, September 1989.

[16] J.W. Baumgarte. Stabilization of constraints and integrals of motion in dynamic systems. *Computer Methods in Applied Mechanics and Engineering*, 1:1–16, 1972.

[17] G.P. Ostermeyer. On Baumgarte stabilization for differential algebraic equations. In E.J. Haug and R.C. Deyo, editors, *Real-Time Integration Methods for Mechanical System Simulation*, pages 193–207. Springer-Verlag, Berlin, Heidelberg, 1990.

[18] E. Eich and M. Hanke. Regularization methods for constrained mechanical multibody systems. *Zeitschrift für angewandte Mathematik und Mechanik*, 75(10):761–773, 1995.

[19] P.E. Nikravesh, R.A. Wehage, and O.K. Kwon. Euler parameters in computational dynamics and kinematics. Part I and Part II. *Journal of Mechanisms, Transmissions, and Automation in Design*, 107(3):358–369, September 1985.

[20] T.W. Park and E.J. Haug. A hybrid numerical integration method for machine dynamic simulation. *Journal of Mechanisms, Transmissions, and Automation in Design*, 108:211–216, June 1986.

[21] C.O. Chang and P.E. Nikravesh. An adaptive constraint violation stabilization method for dynamic analysis of mechanical systems. *Journal of Mechanisms, Transmissions, and Automation in Design*, 107:488–492, December 1985.

[22] D.S. Bae and S.M. Yang. A stabilization method for kinematic and kinetic constraint equations. In E.J. Haug and R.C. Deyo, editors, *Real-Time Integration Methods for Mechanical System Simulation*, pages 209–232. Springer-Verlag, Berlin, Heidelberg, 1990.

[23] S. Yoon, R.M. Howe, and D.T. Greenwood. Stability and accuracy analysis of Baumgarte's constraint violation stabilization method. *ASME Journal of Mechanical Design*, 117:446–453, September 1995.

[24] J.C. Chiou and S.D. Wu. Constraint violation stabilization using input-output feedback linearization in multibody dynamic analysis. *Journal of Guidance, Control, and Dynamics*, 21(2):222–228, March-April 1998.

[25] S.T. Lin and M.C. Hong. Stabilization method for numerical integration of multibody mechanical systems. *ASME Journal of Mechanical Design*, 120:565–572, December 1998.

[26] K.C. Park and J.C. Chiou. Stabilization of computational procedures for constrained dynamical systems. *Journal of Guidance, Control, and Dynamics*, 11(4):365–370, July-August 1988.

[27] E. Bayo, J. García de Jalón, and M.A. Serna. A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 71:183–195, November 1988.

[28] E. Bayo, J. García de Jalón, A. Avello, and J. Cuadrado. An efficient computational method for real time multibody dynamic simulation in fully Cartesian coordinates. *Computer Methods in Applied Mechanics and Engineering*, 92:377–395, 1991.

[29] S. Yoon, R.M. Howe, and D.T. Greenwood. Geometric elimination of constraint violations in numerical simulation of lagrangian equations. *ASME Journal of Mechanical Design*, 116:1058–1064, December 1994.

[30] W. Blajer. Elimination of constraint violation and accuracy aspects in numerical simulation of multibody systems. *Multibody System Dynamics*, 7:265–284, 2002.

[31] J.W. Baumgarte. A new method of stabilization for holonomic constraints. *ASME Journal of Applied Mechanics*, 50:869–870, December 1983.

[32] Z. Terze, D. Lefeber, and O. Muftić. Null space integration method for constrained multibody system simulation with no constraint violation. *Multibody System Dynamics*, 6:229–243, 2001.

[33] E. Bayo and A. Avello. Singularity-free augmented Lagrangian algorithms for constrained multibody dynamics. *Nonlinear Dynamics*, 5:209–231, 1994.

[34] E. Bayo and R. Ledesma. Augmented Lagrangian and mass-orthogonal projection methods for constrained multibody dynamics. *Nonlinear Dynamics*, 9:113–130, 1996.

[35] W.O. Schiehlen. Computational aspects in multibody system dynamics. *Computer Methods in Applied Mechanics and Engineering*, 90:569–582, 1991.

[36] J. Cuadrado, J. Cardenal, and Bayo. E. Modeling and solution methods for efficient real-time simulation of multibody dynamics. *Multibody System Dynamics*, 1:259–280, 1997.

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

[38] M. Géradin and A. Cardona. *Flexible Multibody System: A Finite Element Approach*. John Wiley & Sons, New York, 2001.

[39] A.A. Shabana. Flexible multibody dynamics: Review of past and recent developments. *Multibody System Dynamics*, 1(2):189–222, June 1997.

[40] A.A. Shabana. Substructure synthesis methods for dynamic analysis of multi-body systems. *Computers & Structures*, 20:737–744, 1985.

[41] A. Cardona and M. Géradin. A superelement formulation for mechanism analysis. *Computer Methods in Applied Mechanics and Engineering*, 100:1–29, 1992.

[42] T.J.R. Hughes. *The Finite Element Method*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1987.

[43] K.J. Bathe. *Finite Element Procedures*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1996.

[44] L.F. Shampine and M.K. Gordon. *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*. W.J. Freeman, San Francisco, CA, 1975.

[45] A. Cardona and M. Géradin. Time integration of the equations of motion in mechanism analysis. *Computers & Structures*, 33(3):801–820, 1989.

[46] A. Cardona. *An Integrated Approach to Mechanism Analysis*. PhD thesis, Université de Liège, Belgium, 1989.

[47] N.M. Newmark. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division*, 85:67–94, 1959.

[48] T.J.R. Hughes. Analysis of transient algorithms with particular reference to stability behavior. In T. Belytschko and T.J.R. Hughes, editors, *Computational Methods for Transient Analysis*, pages 67–155. North-Holland, Amsterdam, 1983.

[49] O.A. Bauchau, G. Damilano, and N.J. Theron. Numerical integration of nonlinear elastic multi-body systems. *International Journal for Numerical Methods in Engineering*, 38(16):2727–2751, 1995.

[50] O.A. Bauchau, C.L. Bottasso, and L. Trainelli. Robust integration schemes for flexible multi-body systems. *Computer Methods in Applied Mechanics and Engineering*, 192(3-4):395–420, 2003.

[51] O.A. Bauchau and N.J. Theron. Energy decaying schemes for nonlinear elastic multi-body systems. *Computers & Structures*, 59(2):317–331, 1996.

[52] O.A. Bauchau. Computational schemes for flexible, nonlinear multi-body systems. *Multibody System Dynamics*, 2(2):169–225, 1998.

[53] O.A. Bauchau and C.L. Bottasso. On the design of energy preserving and decaying schemes for flexible, nonlinear multi-body systems. *Computer Methods in Applied Mechanics and Engineering*, 169(1-2):61–79, 1999.

[54] M. Borri, C.L. Bottasso, and L. Trainelli. Integration of elastic multibody systems by invariant conserving/dissipating algorithms. Part I: Formulation. *Computer Methods in Applied Mechanics and Engineering*, 190:3669–3699, 2001.

[55] M. Borri, C.L. Bottasso, and L. Trainelli. Integration of elastic multibody systems by invariant conserving/dissipating algorithms. Part II: Numerical schemes and applications. *Computer Methods in Applied Mechanics and Engineering*, 190:3701–3733, 2001.

[56] O. Gonzalez. Mechanical systems subject to holonomic constraints: Differential-algebraic formulations and conservative integration. *Physica D*, 132:165–174, 1999.

[57] O.A. Bauchau. A self-stabilized algorithm for enforcing constraints in multibody systems. *International Journal of Solids and Structures*, 40(13-14):3253–3271, 2003.

[58] P. Betsch. The discrete null space method for the energy consistent integration of constrained mechanical systems. Part I: Holonomic constraints. *Computer Methods in Applied Mechanics and Engineering*, 194(50-52):5159–5190, 2005.

[59] P. Betsch and S. Leyendecker. The discrete null space method for the energy consistent integration of constrained mechanical systems. Part II: Multibody dynamics. *International Journal for Numerical Methods in Engineering*, 67:499–552, 2006.

[60] P. Betsch and P. Steinmann. A DAE approach to flexible multibody dynamics. *Multibody System Dynamics*, 8:367–391, 2002.

[61] M. Arnold. A perturbation analysis for the dynamical simulation of mechanical multibody systems. *Applied Numerical Mathematics*, 18:37–56, 1995.

[62] C.L. Bottasso, O.A. Bauchau, and A. Cardona. Time-step-size-independent conditioning and sensitivity to perturbations in the numerical solution of index three differential algebraic equations. *SIAM Journal on Scientific Computing*, 29(1):397–414, 2007.