David J. Braun[1]
School of Informatics,
University of Edinburgh,
10 Crichton Street,
Edinburgh EH8 9AB,UK
e-mail: david.braun@ed.ac.uk;
david.braun@vanderbilt.edu

Michael Goldfarb
Department of Mechanical Engineering,
Vanderbilt University,
VU Station B 351592,
Nashville, TN 37235
e-mail: michael.goldfarb@vanderbilt.edu

Simulation of Constrained Mechanical Systems—Part II: Explicit Numerical Integration

This paper presents an explicit to integrate differential algebraic equations (DAEs) method for simulations of constrained mechanical systems modeled with holonomic and nonholonomic constraints. The proposed DAE integrator is based on the equation of constrained motion developed in Part I of this work, which is discretized here using explicit ordinary differential equation schemes and applied to solve two nontrivial examples. The obtained results show that this integrator allows one to precisely solve constrained mechanical systems through long time periods. Unlike many other implicit DAE solvers which utilize iterative constraint correction, the presented DAE integrator is explicit, and it does not use any iteration. As a direct consequence, the present formulation is simple to implement, and is also well suited for real-time applications. [DOI: 10.1115/1.4005573]

1 Introduction

Precise numerical integration of constrained dynamical systems, are usually performed with specifically developed implicit integration methods enhanced with iterative constraint corrections, a review of which can be found in Refs. [1]–[6]. Such an implicit differential algebraic equation (DAE) integrator is nontrivial to implement but can perform stable and precise integration regardless of whether the original dynamical system is stiff or nonstiff by nature. Compared to explicit integrators, implicit integrators are computationally more expensive but also more stable and are usually required if the equation of motion is stiff. Unlike for ordinary differential equations, explicit numerical methods have not been proven as efficient alternatives for DAEs. This is however not because DAEs are mainly stiff by nature, but rather because usual reformulations of the original DAE problem (utilized to prevent error accumulation) make stable DAE integration require a stiff (iterative) solver, Gear [7].

In this paper, we employ the equation of motion derived in Part I of this work [8] which does not require an implicit integrator or iterative correction to provide a precise long time solution for DAEs. As derived, the equation can be directly discretized using explicit ODE schemes (forward Euler, Runge-Kutta, etc.) and integrated in the same fashion as one would integrate ordinary differential equations. As frequently argued by numerous authors, ten Dam [9], Ascher et al. [10], Burgermeister et al. [11], an explicit integration (as the one offered here) is preferred for real-time DAE integration, see Petzold [12]: *"For real-time simulation, it is essential to be able to make use of explicit methods wherever possible; if the solution is not computed in the allotted time, it will be useless."*

Beyond this targeted feature, the DAE solver employed here can be used to simulate systems with *nonholonomic constraints* [13], and to perform *energy conserving DAE integration* where enforcement of the energy conservation law does not interfere with the constraint enforcement, (see Ref. [14] for an alternative formulation). Furthermore, the proposed equation can also be used to treat systems modeled with *redundant constraints*, and to predict motion through *kinematic singularities* [15].

In the remainder of this paper, a numerical implementation of the equation of motion derived in Part I of this work is presented. Application of the proposed explicit DAE integrator is then demonstrated on two nontrivial examples.

[1]Corresponding author.
Manuscript received February 11, 2010; final manuscript received October 21, 2011; accepted manuscript posted February 1, 2012; published online May 16, 2012. Assoc. Editor: Wei-Chau Xie.

2 Explicit Numerical Integration of DAEs

The model of the constrained dynamical system considered here is defined by: (a) the differential equation of the unconstrained dynamical system: $\mathbf{M}(t, \mathbf{q})\ddot{\mathbf{q}} = \mathbf{Q}(t, \mathbf{q}, \dot{\mathbf{q}})$, (where t denotes the time, $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ are the generalized coordinates, velocities and acceleration of the unconstrained system, $\mathbf{M}$ is a mass matrix while $\mathbf{Q}$ represents the inertial and applied forces); (b) the holonomic constraint set: $\mathbf{\Phi}_q(t, \mathbf{q}) = \mathbf{0}$; (c) the nonholonomic constraint set: $\mathbf{\Phi}_v(t, \mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$; and, (d) the energy-type conservation law: $\Phi_e(t, \mathbf{q}, \dot{\mathbf{q}}) = 0$.

The equation of motion for explicit numerical integration for such constrained dynamical system, proposed in Part I of this work [8], is given as:

$$\dot{\mathbf{q}} = \mathbf{v} + \mathbf{R}^{-1}\mathbf{C}_q^{+}(\mathbf{b}_q - \mathbf{A}_q\mathbf{v} - \Phi_q/dt) \tag{1}$$

and

$$\begin{aligned}\dot{\mathbf{v}} = \mathbf{a} &+ \mathbf{R}^{-1}\mathbf{C}_v^{+}(\mathbf{b}_v - \mathbf{A}_v\mathbf{a} - (\dot{\Phi}_q^T, \Phi_v^T)^T/dt) \\ &+ \mathbf{R}^{-1}\mathbf{N}_v(\mathbf{C}_e\mathbf{N}_v)^{+}(b_{ev} - \mathbf{A}_{ev}\mathbf{a} - \Phi_{ev}/dt)\end{aligned} \tag{2}$$

where $\mathbf{q}$ and $\mathbf{v}$ are the coordinates and the velocities of the constrained system, while the definition of the other terms is provided in Ref. [8]. Now instead of utilizing a special numerical algorithm to integrate (a)–(d), one can apply traditional ODE integrators to solve the reformulated Eqs. (1) and (2), without having problems of constraint and energy drift.

2.1 Discretization. By choosing a small time step dt, the domain of integration $t \in [0, T]$ can be equidistantly discretized as $t_n = nT/N$ where $n \in \{0, 1, ..., N\}$. By means of numerical solution, we seek the discrete values of all positions $\{\mathbf{q}_n\}$ and velocities $\{\mathbf{v}_n\}$ which are (numerically) constraint consistent. Let us assume that approximately constraint consistent initial conditions $\mathbf{q}_0 = \mathbf{q}(0)$ and $\mathbf{v}_0 = \mathbf{v}(0)$, $\mathbf{\Phi}_q(0, \mathbf{q}_0) \approx \mathbf{0}$, $\dot{\mathbf{\Phi}}_q(0, \mathbf{q}_0, \mathbf{v}_0) \approx \mathbf{0}$, $\mathbf{\Phi}_v(0, \mathbf{q}_0, \mathbf{v}_0) \approx \mathbf{0}$ are provided. In this case, a precise solution can be obtained by integrating the system dynamics successively between discrete time instants. To illustrate a concrete implementation, a numerical integration of Eqs. (1) and (2) is presented here based on a forward Euler method.

(1) Based on $\mathbf{q}_n = \mathbf{q}(t_n)$ and $\mathbf{v}_n = \mathbf{v}(t_n)$, known from the previous integration step (or defined by initial conditions for the starting step), one can evaluate: $\mathbf{Q} = \mathbf{Q}(t_n, \mathbf{q}_n, \mathbf{v}_n)$, $\mathbf{M} = \mathbf{M}(t_n, \mathbf{q}_n)$, $\mathbf{A}_q = \mathbf{A}_q(t_n, \mathbf{q}_n)$, $\mathbf{A}_v = \mathbf{A}_v(t_n, \mathbf{q}_n, \mathbf{v}_n)$,

$\mathbf{b}_q = \mathbf{b}_q(t_n, \mathbf{q}_n)$, $\mathbf{b}_v = \mathbf{b}_v(t_n, \mathbf{q}_n, \mathbf{v}_n)$, $\mathbf{\Phi}_q = \mathbf{\Phi}_q(t_n, \mathbf{q}_n)$, $\dot{\mathbf{\Phi}}_q = \mathbf{\Phi}_q(t_n, \mathbf{q}_n, \mathbf{v}_n)$ and $\mathbf{\Phi}_v = \mathbf{\Phi}_v(t_n, \mathbf{q}_n, \mathbf{v}_n)$. The upper triangular Cholesky factor of the mass matrix $\mathbf{R}$ is computed, where $\mathbf{M} = \mathbf{R}^T\mathbf{R}$, and the pseudoinverse $\mathbf{C}^+_{q/v}$ is computed based on $\mathbf{C}_{q/v} = \mathbf{A}_{q/v}\mathbf{R}^{-1}$.

(2) By exploiting the Cholesky factorization one can solve the unconstrained acceleration $\mathbf{a}_n$ from $\mathbf{R}^T\mathbf{R}\mathbf{a}_n = \mathbf{Q}$ with a successive forward and backward substitution.

(3) In order to perform energy correction, one needs to evaluate: $\Phi_e = \Phi_e(t_n, \mathbf{q}_n, \mathbf{v}_n)$, $\mathbf{A}_e = \mathbf{A}_e(t_n, \mathbf{q}_n, \mathbf{v}_n)$, $b_e = b_e(t_n, \mathbf{q}_n, \mathbf{v}_n)$, $\mathbf{C}_e = \mathbf{A}_e\mathbf{R}^{-1}$, $\mathbf{N}_v = \mathbf{I} - \mathbf{C}^+_v\mathbf{C}_v$, $(\mathbf{C}_e\mathbf{N}_v)^+$ and $((b_{ev} - \mathbf{A}_{ev}\mathbf{a}_n)dt - \Phi_{ev})$, where $(*)_{ev} = (*)_e - \mathbf{C}_e\mathbf{C}^+_v(*)_v$.

(4) The endpoint position is computed from:

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \mathbf{v}_n dt + \mathbf{R}^{-1}\mathbf{C}^+_q((\mathbf{b}_q - \mathbf{A}_q\mathbf{v}_n)dt - \Phi_q) \tag{3}$$

and the endpoint velocity from:

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{a}_n dt + \mathbf{R}^{-1}\mathbf{C}^+_v((\mathbf{b}_v - \mathbf{A}_v\mathbf{a}_n)dt - (\dot{\Phi}^T_q, \Phi^T_v)^T) + \mathbf{R}^{-1}\mathbf{N}_v(\mathbf{C}_e\mathbf{N}_v)^+((b_{ev} - \mathbf{A}_{ev}\mathbf{a}_n)dt - \Phi_{ev}) \tag{4}$$

At the end of an integration step, the new position and velocity $(\mathbf{q}_{n+1}, \mathbf{v}_{n+1})$ is obtained. These values are used to initialize the next integration cycle.

2.2 Comments on Application and Implementation. In the following, we discuss specific details related to the numerical implementation of Eqs. (3) and (4).

2.2.1 Inconsistent Initialization. Unlike in ODE integration, where the generalized coordinates (selected to exactly satisfy the kinematic constraints) eliminate the issue of inconsistent initialization, in DAE integration, inconsistency in initial conditions is a common issue. Practically, any initial condition $\mathbf{q}_0$ and $\mathbf{v}_0$ which does not *exactly* satisfy the constraints, $\mathbf{\Phi}_q(0, \mathbf{q}_0) = \mathbf{0}$, $\mathbf{\Phi}_v(0, \mathbf{q}_0, \mathbf{v}_0) = \mathbf{0}$, is considered inconsistent. Such an inconsistent initialization is a source of error generation in practice [16,17]. However, as derived and implemented, the presented method assures that slightly inconsistent initial conditions which do not exactly satisfy the constraints cannot cause constraint or energy drift along the time integration. This however does not make inconsistent initialization advisable, namely any integration started with an inconsistent position and velocity will require some starting (transient) time to regain its precision as discussed in Sec. 3.2. To avoid this effect, it is recommended to first algebraically find initial data which is constraint consistent in a numerical sense.[2]

2.2.2 Redundant Constraints. The pseudoinverse notation utilized in Eqs. (3) and (4) is a convenient theoretical tool which allows an explicit representation of the equation of motion regardless of whether the constraints are redundant or not. Computation of the right hand side of Eqs. (3) and (4) however, which does not require direct computation of the pseudoinverses, is an implementation detail discussed in the following.

If the constraints are nonredundant and the system does not move through kinematically singular configurations, the pseudoinverse is explicitly defined $\mathbf{C}^+_* = \mathbf{C}^T_*(\mathbf{C}_*\mathbf{C}^T_*)^{-1}$, otherwise it can be computed utilizing singular value decomposition (SVD) $\mathbf{C}_* = \mathbf{U}\mathbf{S}\mathbf{V}^T$, $\mathbf{C}^+_* = \mathbf{V}\mathbf{S}^+\mathbf{U}^T$ (since $\mathbf{S}$ is a diagonal matrix, computation of its pseudoinverse is trivial). However, instead of defining $\mathbf{C}^+_*$, and using it as a multiplier on the right hand side of Eqs. (3) and (4) directly, it is preferable to use an appropriate matrix decomposition algorithm and to compute the right hand side in Eqs. (3) and (4) by matrix-vector multiplications. Specifically, one can utilize Gram-Schmidt orthogonalization (GSO) [18]; orthogonal triangular decomposition (QR-decomposition), [19,20]; complete orthogonal decomposition (URV-decomposition), [21,22]; or singular value decomposition, [23,24]. From the former to the latter, the algorithms become more numerically stable but also less computationally efficient [5]. While these two objectives need to be negotiated during the implementation, it is also desirable to use off-the-shelf, state-of-the-art numerical routines instead of custom built codes. An additional objective in the present context was to implement the approach in real-time. For that purpose, the rapid prototyping environment provided by MATLAB/SIMULINK was used.

2.2.3 Nonholonomic Constraints. In numerous DAE approaches, the constraint set is assumed full-rank holonomic. While incorporation of rank deficient constraints is a separate issue addressed in the Sec. 2.2.2, utilization of nonlinear nonholonomic constraints (instead of only holonomic or linear nonholonomic once) is an added feature offered here. In order to demonstrate the difficulty at hand, let us point out that any formulation which assumes linearity of the holonomic constraints on velocity cannot be used once nonlinear nonholonomic constraints are applied on the system. In Part I of this work, we provided a systematic derivation which takes care of this practical limitation. This utility has also been recognized as beneficial in the implementation of energy conserving DAE integration.

2.2.4 Energy Conserving DAE Integration. Preventing error accumulation along the motion constraints is often considered as a primary objective in DAE integration. However, utilizing only constraint correction, one can obtain a solution which precisely satisfies the constraints, but which for example, can still be delayed in time with respect to the exact solution of the system. This kind of error, which accumulates on the constraint consistent motion (also present during ODE integration), *cannot* be eliminated, but can be reduced using kinematic constraint correction and energy correction together.

Let us now consider the generalized force

$$\mathbf{Q}_{\|} = \mathbf{R}^T\mathbf{N}_v(\mathbf{C}_e\mathbf{N}_v)^+((b_{ev} - \mathbf{A}_{ev}\mathbf{a}_n)dt - \Phi_{ev}) \tag{5}$$

which is embedded in the equation of motion. Eq. (4) (the last term in Eq. (4) represents $\mathbf{M}^{-1}\mathbf{Q}_{\|}$) to perform the energy correction during the simulation. Note that since $\mathbf{C}_e\mathbf{N}_v$ is a (row) vector, the pseudoinverse $(\mathbf{C}_e\mathbf{N}_v)^+$ required for the proposed energy correction can be calculated

$$(\mathbf{C}_e\mathbf{N}_v)^+ = \begin{cases} (\mathbf{C}_e\mathbf{N}_v)^T/((\mathbf{C}_e\mathbf{N}_v)(\mathbf{C}_e\mathbf{N}_v)^T), & \text{if}(\mathbf{C}_e\mathbf{N}_v)(\mathbf{C}_e\mathbf{N}_v)^T > tol \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{6}$$

without a computationally demanding routine. Moreover, the tolerance *tol* (small nonzero value) in Eq. (6) prevents $(\mathbf{C}_e\mathbf{N}_v)^+$, and also the correction force, from becoming arbitrarily large. In order to gain insight into this mechanism, let us consider an *unconstrained conservative autonomous system* with a classical energy conservation law $\Phi_e = T(\mathbf{q}, \dot{\mathbf{q}}) + V(\mathbf{q}) - T(\mathbf{q}_0, \dot{\mathbf{q}}_0) - V(\mathbf{q}_0) = 0$[3]. In this case, $(\mathbf{N}_v = \mathbf{I})$, $\mathbf{C}_e = \dot{\mathbf{q}}^T\mathbf{R}^T$ while $(\mathbf{C}_e\mathbf{N}_v)^+ = \mathbf{R}\dot{\mathbf{q}}/2T$ if $2T > tol$, and $(\mathbf{C}_e\mathbf{N}_v)^+ = \mathbf{0}$ otherwise. One can recognize that whenever the kinetic energy of the system is close to zero, enforcement of the energy law may require a large corrective force. Note however that the pseudoinverse solution proposed above automatically

[2]Practically, accuracy of the initial data depends on the accuracy required on the kinematic constraints.

[3]$T(\mathbf{q}, \dot{\mathbf{q}})$ is the kinetic energy while $V(\mathbf{q})$ denotes the potential energy of the system.

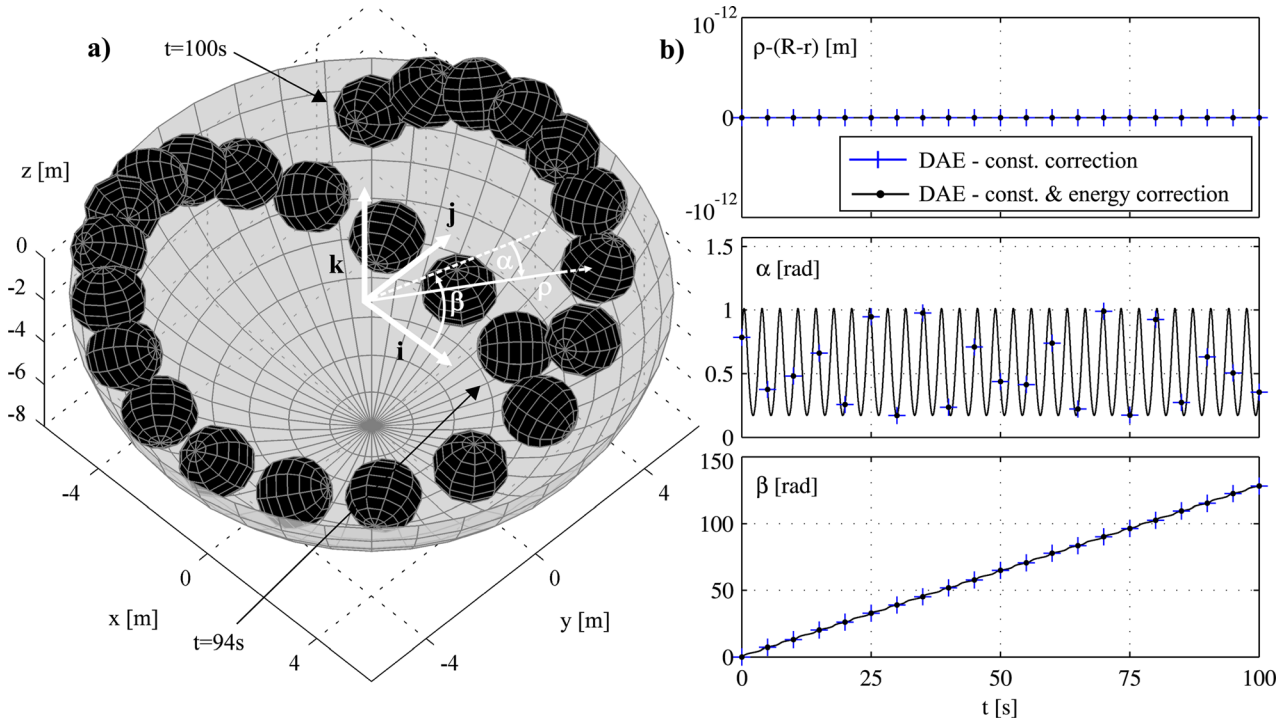

Fig. 1 Ball of mass $m = 1$ kg and $r = 1$ m rolling in a bowl of radius $R = 8$ m. (*a*) Stroboscopic view of the rolling motion. (*b*) Position of the center of the ball $(\rho(t), \alpha(t), \beta(t))$. The simulations are performed with 10^{-2} s time step, through $t \in [0, 100]$ s.

resolves this issue, namely it does not perform any correction in such a case (i.e., $(\mathbf{C}_e\mathbf{N}_v)^+ = \mathbf{0}$ if $2T \leq tol$). Accordingly, the error level, defined by *tol*, can be set[4] to avoid energy correction with large correction forces.

In the next Section (and in Part I of this paper) the proposed energy correction strategy is demonstrated as an effective means to perform precise energy conserving DAE integration through long time periods.

3 Application

In this Section we demonstrate the utility of the proposed formulation, Eqs. (3) and (4) (i.e., equation of motion derived in Part I of this work [8]), on two nontrivial differential-algebraic systems. The examples are solved with a fourth order fixed step Runge-Kutta method (RK4) [25]. The solver is implemented in MATLAB and compiled to a C-code. Using this code, the simulations are performed on a 2.4 MHz Intel Core2 Quad PC computer. All physical quantities used in the simulations have standard SI units [kg, m, s], while $g = 9.81\,\mathrm{m/s^2}$.

3.1 Ball Rolling in a Bowl. Consider a ball with radius r, and mass m which is rolling without slip in a bowl of radius R, $(R > r)$, see Fig. 1(*a*). The ball in the bowl is modeled as a constrained system using twelve coordinates, $\mathbf{q} = [\rho, \alpha, \beta, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]^T$ where $\mathbf{e}_i$, $i \in \{1, 2, 3\}$ are orthogonal unit vectors fixed to the ball at its center [14]. These vectors are used to define the orientation of the ball in the fixed Cartesian reference system $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ placed in the center of the bowl.

Using the selected coordinates, the equation of unconstrained motion, $\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}$, is given by

$$\mathbf{M}\ddot{\mathbf{q}} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m\rho^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & m\rho^2 cos(\alpha)^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}\mathbf{I} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & \mathbf{0} & \frac{1}{2}\mathbf{I} & \mathbf{0} \\ 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & \frac{1}{2}\mathbf{I} \end{bmatrix} \begin{bmatrix} \ddot{\rho} \\ \ddot{\alpha} \\ \ddot{\beta} \\ \ddot{\mathbf{e}}_1 \\ \ddot{\mathbf{e}}_2 \\ \ddot{\mathbf{e}}_3 \end{bmatrix}$$

where $\mathbf{I} = diag(I, I, I)$, $I = \frac{2}{5}mr^2$, $\mathbf{0} = 0_{3\times 3}$ and

$$\mathbf{Q} = -\begin{bmatrix} mgsin(\alpha) \\ mg\rho cos(\alpha) \\ 0 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} -m\rho(\dot{\alpha}^2 + \dot{\beta}^2 cos(\alpha)^2) \\ m\rho(2\dot{\rho}\dot{\alpha} + \rho\dot{\beta}^2 sin(\alpha)cos(\alpha)) \\ 2m\rho(\dot{\rho}\dot{\beta}cos(\alpha)^2 - \rho\dot{\alpha}\dot{\beta}sin(\alpha)cos(\alpha)) \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

where $\mathbf{0} = 0_{3\times 1}$. The holonomic kinematic constraints are,

$$\mathbf{\Phi}_q = \begin{bmatrix} \rho - (R - r) \\ \frac{1}{2}(\mathbf{e}_1^T\mathbf{e}_1 - 1) \\ \frac{1}{2}(\mathbf{e}_2^T\mathbf{e}_2 - 1) \\ \frac{1}{2}(\mathbf{e}_3^T\mathbf{e}_3 - 1) \\ \mathbf{e}_1^T\mathbf{e}_2 \\ \mathbf{e}_1^T\mathbf{e}_3 \\ \mathbf{e}_2^T\mathbf{e}_3 \end{bmatrix} = \mathbf{0}$$

In order to incorporate the nonslip constraints between the ball and the bowl, let us first formally define the velocity at the contact point P, $\mathbf{v}_P = \mathbf{v}_C + \mathbf{v}_P^C$, where $\mathbf{v}_C = 0\mathbf{e}_\rho + (R - r)\dot{\alpha}\mathbf{e}_\alpha + (R - r)cos(\alpha)\dot{\beta}\,\mathbf{e}_\beta$ is the velocity of the ball center, while $\mathbf{v}_P^C = \sum_{i=1}^{3} r(\mathbf{e}_i^T\mathbf{e}_\rho)\dot{\mathbf{e}}_i$ is

[4] In this paper and in Part I of this work [8] we used $tol = 10^{-6}$ in all examples. Note that computation of $(\mathbf{C}_e\mathbf{N}_v)^+$ can also be performed with standard routines that automatically set *tol*.

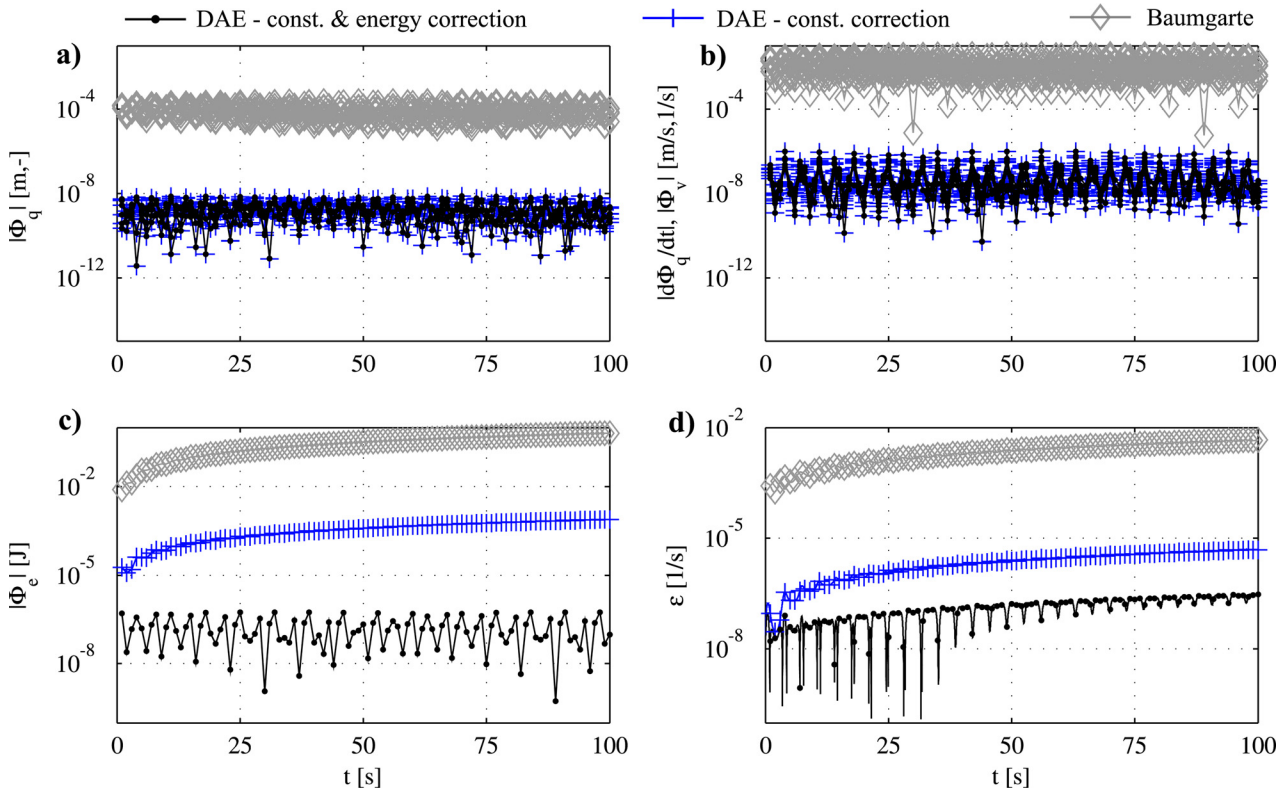

Fig. 2 The semilog figures depict the constraint error, energy error, and an overall error measure $\varepsilon = \left|\dot{\beta}\cos(\alpha)^2 + \frac{2r}{7(R-r)}\omega_\rho \sin(\alpha) - \dot{\beta}(0)\cos(\alpha(0))^2 - \frac{2r}{7(R-r)}\omega_\rho(0)\sin(\alpha(0))\right|$ where $\omega_\rho = \frac{1}{2}\sum_{i=1}^{3}(\mathbf{e}_i \times \dot{\mathbf{e}}_i)^T \mathbf{e}_\rho$. Note that, $\dot{\beta}\cos(\alpha)^2 + \frac{2r}{7(R-r)}\omega_\rho \sin(\alpha) = const.$ is a first integral of the considered nonholonomic system, and accordingly ε is identically zero along the exact solution (i.e., $\varepsilon \equiv 0$). In the current simulation, Baumgarte's constrained stabilization is implemented with optimal parameters [10].

the velocity of the contact point with respect to the center.[5] Using these relations, the two nonholonomic kinematic constraints: $\mathbf{e}_\alpha^T \mathbf{v}_P = 0$, $\mathbf{e}_\beta^T \mathbf{v}_P = 0$ have the following form,

$$\mathbf{\Phi}_v = \begin{bmatrix} (R-r)\dot{\alpha} + \mathbf{e}_\alpha^T r((\mathbf{e}_1^T\mathbf{e}_\rho)\dot{\mathbf{e}}_1 + (\mathbf{e}_2^T\mathbf{e}_\rho)\dot{\mathbf{e}}_2 + (\mathbf{e}_3^T\mathbf{e}_\rho)\dot{\mathbf{e}}_3) \\ (R-r)\cos(\alpha)\dot{\beta} + \mathbf{e}_\beta^T r((\mathbf{e}_1^T\mathbf{e}_\rho)\dot{\mathbf{e}}_1 + (\mathbf{e}_2^T\mathbf{e}_\rho)\dot{\mathbf{e}}_2 + (\mathbf{e}_3^T\mathbf{e}_\rho)\dot{\mathbf{e}}_3) \end{bmatrix} = \mathbf{0},$$

while the energy conservation law is given as,

$$\Phi_e = \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{M}\dot{\mathbf{q}} - mg\rho\sin(\alpha) - E_0 = 0$$

Figure 1(*b*) depicts the motion of the ball integrated with the proposed explicit DAE solver using constraint consistent initial conditions:[6] $\rho(0) = R - r$, $\alpha(0) = \pi/4$, $\beta(0) = 0$, $\dot{\rho}(0) = 0$, $\dot{\alpha}(0) = 0.5\,\mathrm{s}^{-1}$ and $\dot{\beta}(0) = 1\,\mathrm{s}^{-1}$. Figures 2(*a*)–2(*c*) depict the kinematic constraint error, and the error on the energy conservation law. The results demonstrate that *the proposed constraint correction is able to precisely satisfy and steadily maintain the holonomic and the nonholonomic constraints, while the additional energy correction prevents energy drift*. Since, for this example, the exact solution is not provided, the overall precision of the depicted solutions is assessed through a *first integral* of this nonholonomic system, reported in Fig. 2(*d*).

3.2 Biped Robot. In this section, we report a *real-time simulation* with a seven link biped robot modeled as a constrained mechanical system. For this purpose, the explicit equation of constrained motion, developed in Part I of this work, was implemented in SIMULINK and compiled to a C-code. This code was employed in the development and experimental realization of a control approach to dynamic walking by the authors [26,27] and is used here to simulate an oscillatory motion of the biped robot.

Consider a seven-link planar biped robot depicted in Fig. 3(*a*). The configuration of the biped is specified with nine absolute coordinates $\mathbf{q} = [x, y, \theta, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^T$. The unconstrained equation is defined as $\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}$ where, $\mathbf{M} \in \mathbb{R}^{9\times 9}$ is a positive definite mass matrix, $\mathbf{Q} \in \mathbb{R}^9$ collects the centrifugal forces and the gravitational forces. For the present simulation purpose, the constraints are defined to keep the forward heel and the backward toe on ground, and also to emulate an oscillatory motion of the robot[7]. The corresponding kinematic constraints are:

$$\mathbf{\Phi}_q = \begin{bmatrix} y - l_c\sin(\theta) - l_1\sin(\theta_1) - l_2\sin(\theta_2) - h\cos(\theta_3) - b\sin(\theta_3) \\ y - l_c\sin(\theta) - l_1\sin(\theta_4) - l_2\sin(\theta_5) - h\cos(\theta_6) + a\sin(\theta_6) \\ \theta_2 - \theta_1 + \frac{\pi}{10}(1 - \cos(\omega t)) \\ \theta_5 - \theta_4 + \frac{\pi}{10}(1 - \cos(\omega t)) \\ \theta - \left(\frac{4\pi}{9} - \frac{\pi}{18}(1 - \cos(\omega t))\right) \\ \theta_3 - \frac{\pi}{18}(1 + \cos(\omega t)) \\ \theta_6 + \frac{\pi}{18}(1 - \cos(\omega t)) \end{bmatrix} = \mathbf{0}$$

[5]The velocity of the center is expressed in the spherical coordinate system $(\mathbf{e}_\rho, \mathbf{e}_\alpha, \mathbf{e}_\beta)$; $\mathbf{e}_\rho = \cos(\alpha)\cos(\beta)\mathbf{i} + \cos(\alpha)\sin(\beta)\mathbf{j} - \sin(\alpha)\mathbf{k}$, $\mathbf{e}_\alpha = -\sin(\alpha)\cos(\beta)\mathbf{i} - \sin(\alpha)\sin(\beta)\mathbf{j} - \cos(\alpha)\mathbf{k}$, $\mathbf{e}_\beta = -\sin(\beta)\mathbf{i} + \cos(\beta)\mathbf{j} + 0\mathbf{k}$. The position of P can be calculated as $\mathbf{r}_P = \mathbf{r}_C + \sum_{i=1}^{3} X_i^P \mathbf{e}_i = \mathbf{r}_C + r\mathbf{e}_\rho$ where X_i^P are the coordinates of the material point on the ball in contact. By multiplying this last relation (from the left) with $\mathbf{e}_i^T$ one obtains $X_i^P = r(\mathbf{e}_i^T\mathbf{e}_\rho)$, and subsequently $\mathbf{v}_P^C = \sum_{i=1}^{3} X_i^P \dot{\mathbf{e}}_i = \sum_{i=1}^{3} r(\mathbf{e}_i^T\mathbf{e}_\rho)\dot{\mathbf{e}}_i$.

[6]The initial orientation of the ball is defined as: $\mathbf{e}_1(0) = [1, 0, 0]^T$, $\mathbf{e}_2(0) = [0, 1, 0]^T$, $\mathbf{e}_3(0) = [0, 0, 1]^T$, while the corresponding consistent initial velocities are calculated as $\dot{\mathbf{e}}_i(0) = \boldsymbol{\omega}(0) \times \mathbf{e}_i(0)$ where $\boldsymbol{\omega}(0) = \omega_\rho(0)\mathbf{e}_\rho(0) + ((R-r)\cos(\alpha(0))/r)\dot{\beta}(0)\mathbf{e}_\alpha(0) - ((R-r)/r)\dot{\alpha}(0)\mathbf{e}_\beta(0)$ and $\omega_\rho(0) = 1\,\mathrm{s}^{-1}$.

[7]The desired motion of the robot is chosen to avoid knee collision and ground collision with the feet such that the necessary smoothness condition on constraints is provided.

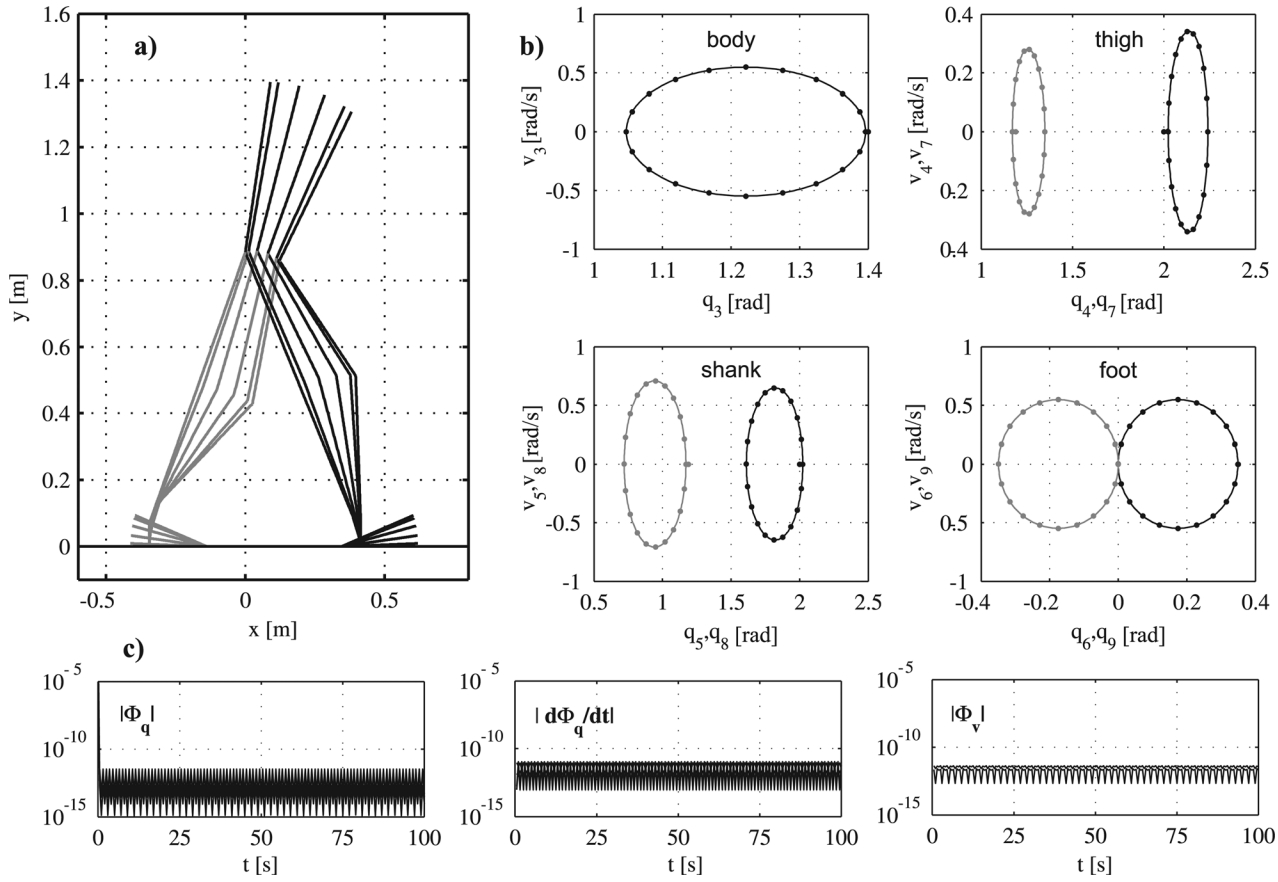

Fig. 3 Anthropomorphic biped, with model parameters given in Ref. [28] and $\omega = \pi s^{-1}$. (*a*) Stroboscopic plot of the oscillatory motion. (*b*) Phase plots. The real-time simulation is performed for $t \in [0, 100]s$ with 10^{-2} s time step, started from (slightly) *inconsistent initial conditions*: $\mathbf{q}(0) = [0.056, 1.2, 1.4, 2, 2, 0.35, 1.19, 1.19, 0]^T$, $\mathbf{v}(0) = 0$. (*c*) The kinematic constraints are precisely satisfied and steadily maintained.

and

$$\mathbf{\Phi}_v = \begin{bmatrix} \dot{x} + l_c sin(\theta)\dot{\theta} + l_1 sin(\theta_1)\dot{\theta}_1 + l_2 sin(\theta_2)\dot{\theta}_2 + (hcos(\theta_3) + bsin(\theta_3))\dot{\theta}_3 \\ \dot{x} + l_c sin(\theta)\dot{\theta} + l_1 sin(\theta_4)\dot{\theta}_4 + l_2 sin(\theta_5)\dot{\theta}_5 + (hcos(\theta_6) - asin(\theta_6))\dot{\theta}_6 \end{bmatrix} = \mathbf{0}$$

In this example, the constraints in $\mathbf{\Phi}_v$ are integrable, and as such they could be embedded in the holonomic constraint set $\mathbf{\Phi}_q$. In the present case, however, we consider the integration constants to be unknown, under which condition the velocity level constraints cannot be used on the position level.

The stroboscopic view of the simulated motion is depicted in Fig. 3(*a*). Figure 3(*b*) shows that during the *50 motion cycles*, $t \in [0, 100]s$, the corresponding configurations of the robot in successive periods are overlapped. Note that *despite the slight inconsistency in initialization, all constraints are precisely and steadily maintained, see Fig. 3(c)*. In order to demonstrate the effectiveness of the kinematic correction approach alone, we have chosen not to perform energy correction in this example. This could be justified in practice during short time integration (where energy correction may not have a prominent effect) and also if the energy law is either not provided or if it cannot be precisely computed. Regardless of whether energy correction is implemented or not however, the present formulation provides a physically viable constraint consistent solution.

4 Conclusion

This paper presents an explicit integration method which allows precise simulation of constrained mechanical systems modeled with holonomic and nonholonomic constraints. Unlike many other DAE integrators which are implicit and require iterative constraint correction, the presented solver is explicit and it does not use any iteration. This DAE solver is simple to implement and is also well suited to real-time applications. Simulation results demonstrate the effectiveness of the proposed *explicit DAE integration method*.

Acknowledgment

This work was done while D. J. Braun was with Vanderbilt University—Center for Intelligent Mechatronics as a research associate.

References

[1] Schiehlen, W., 1997, "Multibody System Dynamics: Roots and Perspectives," Multibody Syst. Dyn., **1**(2), pp. 149–188.

[2] Shabana, A. A., 1997, "Flexible Multibody Dynamics: Review of Past and Recent Developments," Multibody Syst. Dyn., **1**(2), pp. 189–222.

[3] Brogliato, B., ten Dam, A. A., Paoli, L., Génot, F., and Abadie, M., 2002, "Numerical Simulation of Finite Dimensional Multibody Nonsmooth Mechanical Systems," Appl. Mech. Rev., **55**(2), pp. 107–150.

[4] Eberhard, P., and Schiehlen, W., 2006, "Computational Dynamics of Multibody Systems: History, Formalisms, and Applications," J. Comput. Nonlinear Dyn., **1**(1), pp. 3–12.

[5] Laulusa, A., and Bauchau, O. A., 2008, "Review of Classical Approaches for Constraint Enforcement in Multibody Systems," J. Comput. Nonlinear Dyn., **3**(1), p. 011004.
[6] Bauchau, O. A., and Laulusa, A., 2008, "Review of Contemporary Approaches for Constraint Enforcement in Multibody Systems," J. Comput. Nonlinear Dyn., **3**(1), p. 011005.
[7] Gear, C. W., 2006, "Towards Explicit Methods for Differential Algebraic Equations," BIT, **46**(3), pp. 505–514.
[8] Braun, D., and Goldfarb, M., 2010, "Simulation of Constrained Mechanical Systems—Part I: An Equation of Motion," J. Appl. Mech., **79**(4), p. 041017.
[9] ten Dam, A. A., 1992, "Stable Numerical Integration of Dynamical Systems Subject to Equality State-Space Constraints," J. Eng. Math., **26**(2), pp. 315–337.
[10] Ascher, U. M., Chin, H., and Reich, S., 1994, "Stabilization of DAEs and Invariant Manifolds," Numerische Mathematik, **67**(2), pp. 131–149.
[11] Burgermeister, B., Arnold, M., and Esterl, B., 2006, "DAE Time Integration for Real-Time Applications in Multi-Body Dynamics," Z. Angew. Math. Mech., **86**(10), pp. 759–771.
[12] Petzold, L., 1992, "Numerical Solution of Differential-Algebraic Equations in Mechanical Systems Simulation," Physica D, **60**(1–4), pp. 269–279.
[13] Blajer, W., 2002, "Elimination of Constraint Violation and Accuracy Aspects in Numerical Simulation of Multibody Systems," Multibody Syst. Dyn., **7**(3), pp. 265–284.
[14] Betsch, P., 2006, "Energy-Consistent Numerical Integration of Mechanical Systems With Mixed Holonomic and Nonholonomic Constraints," Comput. Methods Appl. Mech. Eng., **195**(50-51), pp. 7020–7035.
[15] Kövecses, J., Piedboeuf, J.-C., and Lange, C., 2003, "Dynamics Modeling and Simulation of Constrained Robotic Systems," IEEE/ASME Trans. Mechatron., **8**(2), pp. 165–177.
[16] Leimkuhler, B., Petzold, L. R., and Gear, C. W., 1991, "Approximation Methods for the Consistent Initialization of Differential-Algebraic Equations," SIAM J. Numer. Anal., **28**(1), pp. 205–226.
[17] Nikravesh, P. E., 2008, "Initial Condition Correction in Multibody Dynamics," Multibody Syst, Dyn,, **18**(1), pp. 107–115.
[18] Udwadia, F. E., Kalaba, R. E., and Phohomsiri, P., 2004, "Mechanical Systems With Nonideal Constraints: Explicit Equations Without the Use of Generalized Inverses," J. Appl. Mech., **71**(5), pp. 618–621.
[19] Kim, S. S., and Vanderploeg, M. J., 1986, "QR Decomposition for State Space Representation of Constrained Mechanical Dynamical Systems," ASME J. Mech., Transm., Autom. Des., **108**(2), pp. 183–188.
[20] Neto, M. A., and Ambrósio, J., 2003, "Stabilization Methods for the Integration of DAE in the Presence of Redundant Constraints," Multibody Syst. Dyn., **10**(1), pp. 81–105.
[21] Fierro, R. D., Hansen, P. C., and Hansen, P. S. K., 1999, "UTV Tools: Matlab Templates for Rank-Revealing UTV Decompositions," Numer. Algorithms, **20**(2-3), pp. 165–194.
[22] Stewart, G. W., 1999, "The QLP Approximation to the Singular Value Decomposition," SIAM J. Comput., **20**(4), pp. 1336–1348.
[23] Singh, R. P., and Likins, P. W., 1985, "Singular Value Decomposition for Constrained Dynamical Systems," J. Appl. Mech., **52**(4), pp. 943–948.
[24] Golub, G., and Loan, C. V., 1996, *Matrix Computations*, 3rd ed., The John Hopkins University Press, Baltimore, MD.
[25] Abramowitz, M., and Stegun, I. A., 1972, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, New York.
[26] Braun, D. J., and Goldfarb, M., 2009, "A Control Approach for Actuated Dynamic Walking in Biped Robots," IEEE Trans. Robot., **25**(6), pp. 1292–1303.
[27] Braun, D. J., Mitchell, J. E., and Goldfarb, M., 2012, "Actuated Dynamic Walking in a Seven-Link Biped Robot," IEEE/ASME Trans. Mechatron., **17**(1) pp. 147–156.
[28] Braun, D. J., and Goldfarb, M., 2009, "Eliminating Constraint Drift in the Numerical Simulation of Constrained Dynamical Systems," Comput. Methods Appl. Mech. Eng., **198**(37–40), pp. 3151–3160.