

DETC2005-84915

RECURSIVE ALGORITHM BASED ON CANONICAL MOMENTA FOR FORWARD DYNAMICS OF MULTIBODY SYSTEMS: NUMERICAL RESULTS

Joris Naudet

Vrije Universiteit Brussel
Faculty of Applied Sciences
Department of Mechanical Engineering
Pleinlaan 2, 1050 Brussel, Belgium
Email: joris.naudet@vub.ac.be

Dirk Lefeber

Vrije Universiteit Brussel
Faculty of Applied Sciences
Department of Mechanical Engineering
Pleinlaan 2, 1050 Brussel, Belgium
Email: dlefeber@vub.ac.be

ABSTRACT

In this paper, a recursive $O(n)$ method to obtain a set of Hamiltonian equations for open-loop and constrained multibody system is briefly discussed. The method is then used to perform a numerical comparison of acceleration based and canonical momenta based equations of motion. A relatively simple example consisting of a biped during double support phase is used for that purpose. While no significant difference in efficiency is found when using a fixed step numerical integration method, the Hamiltonian equations perform considerably better when using an adaptive method. This is at least the case when the error control is applied straightforwardly. Both methods can be made equally efficient by removing the error control on the velocities for the acceleration based equations.

INTRODUCTION

Multibody systems (MBS) dynamics is the study of the motion of systems of interconnected bodies and the forces and torques exerted on them. The simulation of the motion of mechanical systems has a wide variety of applications such as virtual prototyping, virtual reality, computer animation and advanced robot control. One can state that two main challenges are on the focus of research. On one hand, a lot of research has been and is still being done to reduce the computation time of dynamical simulations, while the complexity of the simulated systems keeps growing. The efficiency of simulations can be increased by

designing more efficient algorithms for obtaining the equation of motion [1–8] or by achieving better numerical integration [9–11] for advancing the state in time. On the other hand, many efforts are done to incorporate events in the simulations, while retaining computational efficiency. Contact detection and impacts, user-interaction and time-varying topologies need all to be taken into account when realistic simulations are needed, this is especially true for physics based computer games [12].

In previous publications [13, 14], an attempt was made to address the first challenge. A new, canonical momenta based algorithm was presented to solve the forward dynamics problem in a very efficient way, by reducing the number of operations required to obtain the equations of motion. It is also possible to handle constrained multibody systems with the algorithm, as shown in [15]. The purpose of this paper is to demonstrate some numerical examples and to point out some interesting features of the Hamiltonian equations related to the numerical integrations of these specific examples. In sections and , a short background of the Lagrangian and Hamiltonian equations is given. In the following sections, the recursive algorithm is presented and is it briefly described how the joint accelerations can be obtained. It is handy to be able to calculate the derivatives of the canonical momenta and the accelerations with the same algorithm, as it enables a comparison between the numerical integrations of both sets of equations. Finally, an example consisting of a constrained multibody system is discussed.

LAGRANGE'S EQUATIONS

The equations of motion for a multibody system described by n generalized coordinates q can be found using the well-known Lagrangian approach [16–18]:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial L}{\partial \mathbf{q}} = \mathbf{Q} \quad (1)$$

$L = T - V$ is the Lagrangian function, where T and V are respectively the total kinetic and the total potential energies of the system. \mathbf{Q} are the external generalized (non-conservative) forces. Equations (1) form a set of ordinary differential equations (ODE) of second order. If the coordinates \mathbf{q} are related to each other by means of (holonomic) constraint equations

$$\Phi(\mathbf{q}, t) = \mathbf{0} \quad (2)$$

a term embedding the generalized reaction forces is added to the equations:

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial L}{\partial \mathbf{q}} + \Phi_{\mathbf{q}}^T \lambda &= \mathbf{Q} \\ \Phi(\mathbf{q}, t) &= \mathbf{0} \end{aligned} \quad (3)$$

$\Phi_{\mathbf{q}}$ is the Jacobian matrix of the constraints and λ are the Lagrange multipliers. Equations (3) form a set of mixed differential algebraic equations (DAE). The constraint equations are often differentiated twice to time, to convert the DAE into an ODE:

$$\Phi_{\mathbf{q}} \ddot{\mathbf{q}} + \dot{\Phi}_{\mathbf{q}} \dot{\mathbf{q}} + \dot{\Phi}_t = \mathbf{0} \quad (4)$$

$\dot{\Phi}_t$ is the partial derivative of the constraints with respect to time. The combination of (3) and (4) results in

$$\begin{pmatrix} \mathbf{M} & \Phi_{\mathbf{q}}^T \\ \Phi_{\mathbf{q}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{q}} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{Q} + L_{\mathbf{q}} - \dot{\mathbf{M}}\dot{\mathbf{q}} \\ -\dot{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} - \dot{\Phi}_t \end{pmatrix} \quad (5)$$

\mathbf{M} is the mass matrix. The use of the constraint equations at the acceleration level (4) does not have any theoretical repercussion, it does however induce problems during numerical integration. As small numerical errors are introduced on the acceleration level, these will be integrated twice and will result in uncontrolled errors on the velocity and position levels. Therefore

the need for stabilization methods for the constraint violation errors. Well known procedures are the penalty methods [10, 19] and the coordinate partitioning method [20]. Differential algebraic approaches based on projections on the constraint manifold are promising alternatives [21, 22].

HAMILTON'S EQUATIONS

The Hamiltonian equations can be found by applying a Legendre transformation on the Lagrangian [16]. This transformation changes the description of the system in terms of generalized coordinates \mathbf{q} and velocities $\dot{\mathbf{q}}$ to a description in terms of the same coordinates \mathbf{q} and their conjugated canonical momenta \mathbf{p} . These canonical momenta are defined as:

$$\mathbf{p} = \frac{\partial L}{\partial \dot{\mathbf{q}}} \quad (6)$$

They are an extension of the concept of linear and angular momenta to generalized coordinates. Applying the Legendre transformation yields

$$\dot{\mathbf{q}} = \frac{\partial H}{\partial \mathbf{p}} \quad (7a)$$

$$\dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{q}} + \mathbf{Q} - \Phi_{\mathbf{q}}^T \lambda \quad (7b)$$

$$\Phi(\mathbf{q}, t) = \mathbf{0} \quad (7c)$$

This is a set of DAE's with $2n$ first order differential equations and m kinematic constraint equations. $H = \mathbf{p}^T \dot{\mathbf{q}} - L$ is the Hamiltonian function. DAE's are characterized by a so-called differential index. The acceleration based formulations have an index of 3, the Hamiltonian formulation has index 2 [18]. As shown by Brenan et al. [23], index 2 DAE's have a better behavior during numerical integration. Numerical evidence is provided in [10, 24]. Hence, the use of canonical momenta may be numerically advantageous compared to the use of accelerations. Transforming DAE (7) into a set of ODE's, in analogy with the Lagrangian formulation, leads to the same conclusion. This can be seen by considering the equations of motion as the solution of a variational problem with constraints [9] [10]. This implicates the definition of the so-called augmented Lagrangian which includes the constraints at the velocity level:

$$L^* = L + \dot{\Phi}^T \sigma \quad (8)$$

It results in following set of equations:

$$\begin{pmatrix} \mathbf{M} & \Phi_q^T \\ \Phi_q & \mathbf{0} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}} \\ \sigma \end{pmatrix} = \begin{pmatrix} \mathbf{p} \\ -\Phi_t \end{pmatrix} \quad (9)$$

together with

$$\dot{\mathbf{p}} = L_q + \mathbf{Q} + \dot{\Phi}_q^T \sigma \quad (10)$$

As the constraints were used at the velocity level, the numerical errors are integrated only once, resulting in smaller constraint violations.

O(n) FORMULATIONS

There exists a myriad of methods to derive sets of equations of motion. Equations (5) and (9) enunciate $O(n^3)$ methods by constructing and inverting the mass matrix. These methods can be quite efficient for few DOF, certainly in combination with symbolical optimization. They tend to become computationally demanding for higher numbers of bodies however, and it may be a more judicious choice to revert to recursive $O(n)$ algorithms in that case. For open-loop systems, a wide choice of algorithms is available. They mainly differ in details as the use of quasi-velocities or generalized speeds [3], the chosen reference points (center of gravity, joint axis), description in relative or inertial axes, the use of barycentric coordinates... Many algorithms use the concept of articulated mass matrix introduced in [1]. It is also possible to derive a set of *Hamiltonian equations* with a $O(n)$ method [13]. Constrained multibody systems are more difficult to solve recursively. Baraff solved the problem using the descriptor form and exploiting the sparse structure of the matrix $(\Phi_q \mathbf{M}^{-1} \Phi_q^T)$ [6]. In [25], a method using the reduction of the dependent spatial velocities can be found, this method is extended and explained with the concept of phantom bodies in [7]. In [15] the same ideas are used for the Hamiltonian formalism.

RECURSIVE HAMILTONIAN ALGORITHM

The recursive Hamiltonian algorithm for open-loop as well as for constrained multibody systems will now be described. For those who will not settle for less than a complete derivation, [26] and [15] may give satisfaction.

Rigid Body

The algorithm introduces a 6-dimensional momentum vector \mathbf{P} , which is nothing else than the aggregation of the linear \mathbf{p}_l and angular \mathbf{p}_a momenta:

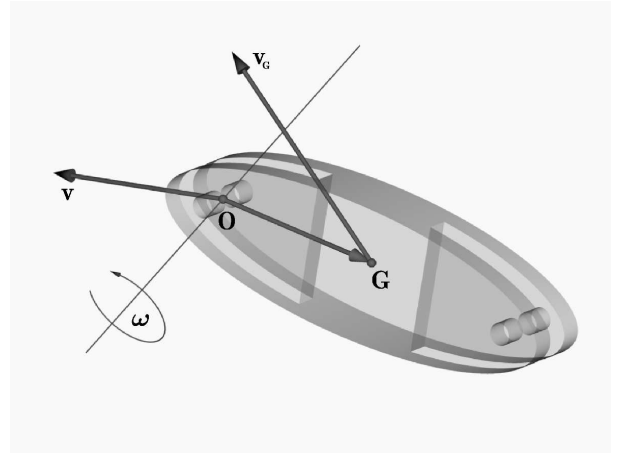


Figure 1. KINEMATICS NOTATION ON A RIGID BODY.

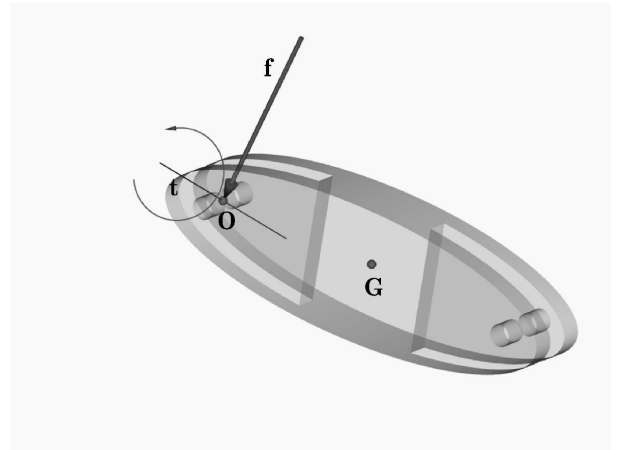


Figure 2. DYNAMICS NOTATION ON A RIGID BODY.

$$\mathbf{P} = \begin{pmatrix} \mathbf{p}_l \\ \mathbf{p}_a \end{pmatrix} = \begin{pmatrix} m\mathbf{I} & m\widetilde{\mathbf{OG}} \\ m\widetilde{\mathbf{OG}} & \mathbf{J} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \omega \end{pmatrix} = \mathbf{M}\Omega \quad (11)$$

m is the mass of the body, \mathbf{I} is a unity dyadic, \mathbf{J} is the inertia tensor and \mathbf{OG} is the position vector from the origin O of the local reference frame to the center of mass G of the body (fig.1). \mathbf{v} is the linear velocity of O and ω the angular velocity of the body referred to the inertial axes. Both the linear and angular velocities are brought together in the spatial velocity vector Ω . \mathbf{M} is the (symmetrical) mass matrix.

$\tilde{\mathbf{x}}$ is a skew-symmetric matrix constructed from the vector \mathbf{x} and is an alternative notation for the cross product.

$$\mathbf{x} \times \mathbf{a} = \tilde{\mathbf{x}} \mathbf{a} = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \quad (12)$$

Writing the Newton-Euler equations in terms of the mass matrix and taking the time derivatives with respect to the *local* reference frame results in following concise formula [26]:

$$\dot{\mathbf{P}} + \boldsymbol{\Omega} \times \mathbf{P} = \mathbf{T} \quad (13)$$

in which $\mathbf{T} = (\mathbf{f}^T \quad \mathbf{t}^T)^T$ is the spatial force containing both the resultants of the forces \mathbf{f} and the torques \mathbf{t} (fig. 2). The cross product for spatial vectors is defined as

$$\boldsymbol{\Omega} \times = \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{pmatrix} \times \triangleq \begin{pmatrix} \tilde{\boldsymbol{\omega}} & \mathbf{0} \\ \tilde{\mathbf{v}} & \tilde{\boldsymbol{\omega}} \end{pmatrix} \quad (14)$$

Using definition (6), one can see that the canonical momenta are the projection of the momentum vector on the subspace \mathbf{E} of the virtual motion:

$$\mathbf{p} = \frac{\partial L}{\partial \dot{\mathbf{q}}} = \frac{\partial T}{\partial \dot{\mathbf{q}}} = \frac{\partial \boldsymbol{\Omega}^T}{\partial \dot{\mathbf{q}}} \mathbf{M} \boldsymbol{\Omega} = \mathbf{E}^T \mathbf{M} \boldsymbol{\Omega} = \mathbf{E}^T \mathbf{P} \quad (15)$$

Matrix \mathbf{E} is called the joint matrix and consists of the partial derivatives of the spatial velocity with respect to the joint velocities.

Open-Loop Multibody System

For open-loop multibody systems, the so-called articulated momentum vector \mathbf{P}^* and accumulated forces \mathbf{T}^* are needed. They follow from the concept of articulated bodies [1].

$$\mathbf{P}_K^* = \mathbf{P}_K + {}^K \mathcal{T}_{K+1}^F \mathbf{P}_{K+1}^* \quad (16)$$

$$\mathbf{T}_K^* = \mathbf{T}_K + {}^K \mathcal{T}_{K+1}^F \mathbf{T}_{K+1}^* \quad (17)$$

${}^K \mathcal{T}_{K+1}^F$ denotes the spatial force transport operator (also called spatial force shift dyadic) from outboard body $K+1$ to actual body K . It can again be shown that the canonical momenta are obtained by projecting the articulated momentum vector on the joint subspace:

$$\mathbf{p}_K = \mathbf{E}_K^T \mathbf{P}_K^* \quad (18)$$

The principle of virtual power states that the reaction forces and torques do not produce a net power. This can be mathematically formulated as

$$\sum_i \boldsymbol{\Omega}_i^{*T} (\dot{\mathbf{P}}_i + \boldsymbol{\Omega}_i \times \mathbf{P}_i - \mathbf{T}_i) = \mathbf{0} \quad (19)$$

The summation is taken over all rigid bodies of the system, $\boldsymbol{\Omega}^*$ denotes a *virtual* spatial velocity. Further developments of these equations and introduction of (16) and (17) eventually leads to

$$\dot{\mathbf{P}}_K^* + \boldsymbol{\Omega}_K \times \mathbf{P}_K^* = \mathbf{T}_K^* \quad (20)$$

which is of the same form (13) as for a single rigid body. The time derivative of the canonical momenta can explicitly be calculated by projecting these equations on the joint subspace:

$$\dot{\mathbf{p}}_K = \mathbf{E}_K^T (\dot{\mathbf{T}}_K^* - \boldsymbol{\Omega}_K \times \mathbf{P}_K^*) + \dot{\mathbf{E}}_K \mathbf{P}_K^* \quad (21)$$

Constrained Multibody Systems

When additional constraints are imposed on the system, for example in the case of loop closure, the principle of virtual power can be used as well. However, the virtual coordinate velocities complicate matters: while they were independent in the case of open-loop systems, they are related to each other in the constrained case. A $O(n)$ solution consists of reducing the dependent spatial velocities to the independent ones.

Consider figure 3 for example. It represent a chain with N elements interconnected by means of pin-joints. Solving the problem implies a choice of dependent coordinates. Let it be the coordinates conjugated to the two last elements $N-1$ and N . It can be shown that the velocities of these bodies can be written as

$$\boldsymbol{\Omega}_N = \mathbf{C}_N {}^N \mathcal{T}_{N-1}^V \boldsymbol{\Omega}_{N-1} \quad (22)$$

$$\boldsymbol{\Omega}_{N-1} = \mathbf{C}_{N-1} {}^{N-1} \mathcal{T}_{N-2}^V \boldsymbol{\Omega}_{N-2} \quad (23)$$

with the constrained matrices \mathbf{C} being independent on the coordinate velocities. \mathcal{T}^V is the velocity transport operator. Inspired by these equations, one can define a generalized articulated momentum vector as

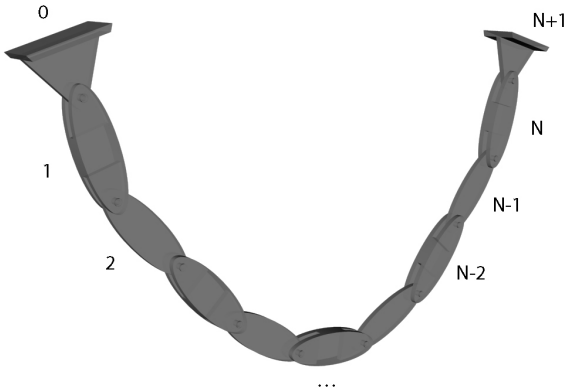


Figure 3. CHAIN WITH BOTH ENDS FIXED BY MEANS OF PIN-JOINTS.

$$\mathbf{P}_K^c = \mathbf{P}_K + \mathbf{C}_K^{T \kappa_{T_{K+1}^F}} \mathbf{P}_{K+1}^c \quad (24)$$

This vector will further be called the constrained (articulated) momentum vector. Note that one can use this definition for open-loop systems, assigning $\mathbf{C} = \mathbf{I}$. Like for single rigid bodies and open-loop systems, the canonical momenta can be found by projection of \mathbf{P}^c on the joint subspace:

$$\mathbf{p}_K = \mathbf{E}_K^T \mathbf{P}_K^c \quad (25)$$

Quite tedious manipulations of (19) yield

$$\dot{\mathbf{P}}_K^c + \boldsymbol{\Omega}_K \times \mathbf{P}_K^c = \mathbf{T}_K^c \quad (26)$$

Unfortunately, the constrained force vector \mathbf{T}_K^c does not have a simple form. This is obviously the price to pay for the added constraints.

$$\begin{aligned} \mathbf{T}_K^c = & \mathbf{T}_K + {}^{\kappa_{T_{K+1}^F}} \mathbf{C}_{K+1}^T \mathbf{T}_{K+1} \\ & + {}^{\kappa_{T_{K+1}^F}} [\dot{\mathbf{C}}_{K+1}^T + (\boldsymbol{\Omega}_{K+1} \times \mathbf{I}) \mathbf{C}_{K+1}^T - \mathbf{C}_{K+1}^T (\boldsymbol{\Omega}_{K+1} \times \mathbf{I})] \mathbf{P}_{K+1} \end{aligned} \quad (27)$$

On the other hand, \mathbf{C} has a sparse structure and above equation does not involve as many operations as may appear. Choosing arbitrary dependent bodies is quite more complicated, but it has no repercussions on the global shape of the equations.

Velocities And Accelerations

A set of Hamiltonian equations (7) is not complete without the time derivative of the coordinates $\dot{\mathbf{q}}$. These coordinate velocities can be computed by means of equation (11), which establishes the link between the momentum vectors and the spatial velocities. Development of this equation together with the definition for the articulated momentum vector results in a method to obtain the velocities using one backward and one forward recursion. Moreover, a similar method can be conceived for the calculation of the accelerations. While the accelerations are not necessary for the simulation itself, they indirectly provide the joint reaction forces at a moderate additional cost. A user can also easily switch from a canonical based formulation to an acceleration based formulation. This provides a convenient tool for comparison of the numerical integration of both formalisms.

NUMERICAL EXAMPLE: A BIPED

A numerical comparison between acceleration based and canonical momenta based equations of motion will now be performed. The purpose is to study the behavior of both sets of equations during numerical integration. First, fixed step numerical integrators are used to compare the introduced errors for equal time steps. Subsequently, adaptive methods are used and the number of function calls needed to achieve a given tolerance is recorded. Only one example is considered, the results should not be interpreted beyond this context.

Model Description

The constrained multibody system that will be discussed is based on the model of a biped robot during the double support phase (fig.4), so both feet are fixed to the ground during the simulation. The model is kept simple and only gravitational forces are considered, while no friction is present and no contact detection is performed. The biped actually moves through the ground in the presented simulation. This does not matter, as the only purpose of this example is to compare the behavior of acceleration-based and canonical momenta-based equations during numerical integration.

The system has five links and three degrees of freedom. Each link has a mass of $1kg$ and a length of $0.25m$, resulting in a moment of inertia of $0.0208kg.m^2$ around the joint axis. Using the recursive Hamiltonian algorithm, a singularity occurs if the two dependent links 4 and 5 are along the same line. It is not the purpose of this paper to show how singularities can be coped with, so singularities are simply avoided. The total simulation time is limited to $0.85s$ for that purpose. The initial velocities are all zero, while the initial coordinates are respectively $q_1 = 1.7, q_2 = -0.7$ and $q_3 = 0.3$. The two ankles touch the ground at the same height, $0.25m$ apart.

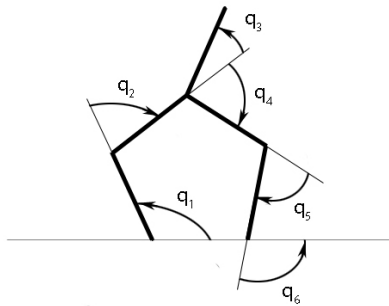


Figure 4. MODEL OF THE BIPED.

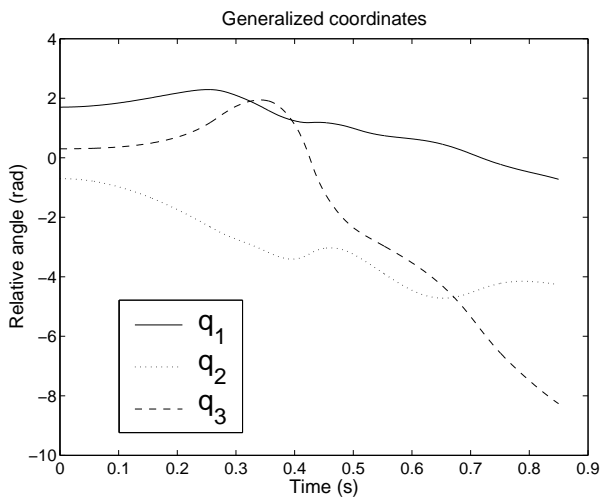


Figure 5. GENERALIZED COORDINATES.

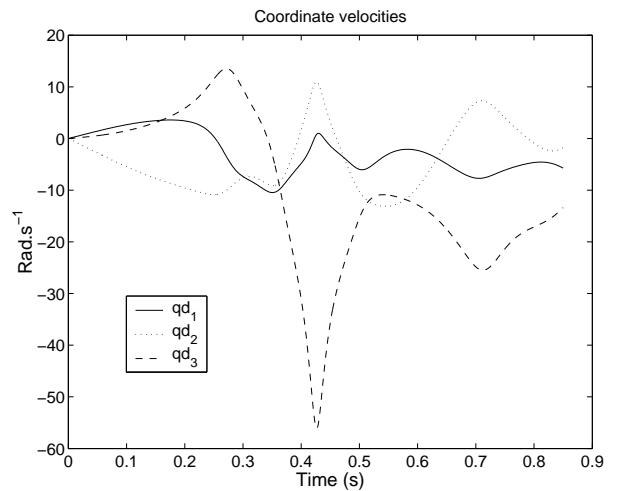


Figure 6. COORDINATE VELOCITIES.

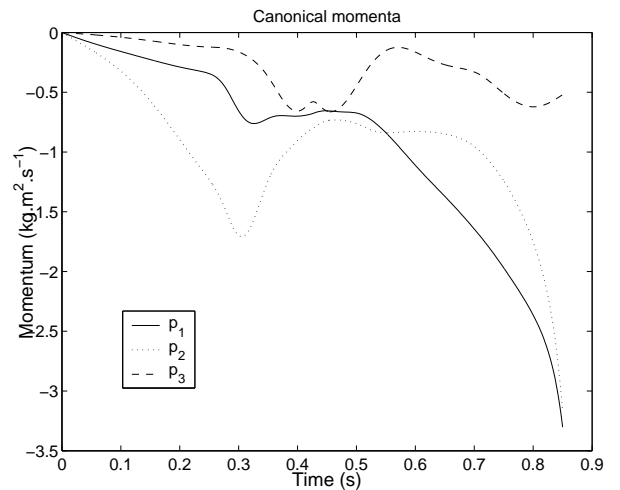


Figure 7. CANONICAL MOMENTA.

Numerical Integration With Fixed Step

For a first approach to the simulation, a fixed step method is used. The Runge-Kutta-Fehlberg 4th order method is suitable, as it provides an estimation of the local truncation error, which will give interesting information. The simulation is performed several times, with different time steps ranging from $0.0001s$ to $0.01s$ and with both a set of equations based on accelerations (1) and a set of Hamiltonian equations (7). The evolutions of the coordinates, the coordinate velocities and the canonical momenta are given in figures 5, 6 and 7.

In table 1, one can see a synthesis of the results. The estimated truncation error, averaged over the total simulation time and over all generalized coordinates, is calculated as well as the mean end error on the coordinates (at $t = 0.85s$). This end error is obtained by comparing the simulation results with a reference trajectory calculated with the *adaptive* Runge-Kutta Fehlberg method. Absolute and relative tolerances of $1.0e - 18$

and $2.2e - 14$ (minimum allowed by Matlab's ODE45) were taken. Although the numbers are consistently smaller for the Hamiltonian equations, the measure is too coarse and the differences are too small to make any conclusion. Similar results were obtained using a second order Runge-Kutta method and the implicit trapezoidal method (ODE15S). Additionally, fig.8 demonstrates the positive effect of the Hamiltonian equations on the evolution of the constraint violation errors.

It is also interesting to note that the local truncation error on the velocities are consistently larger than on the coordinates. For a fixed time step of $0.001s$, the averaged error is $1.1e - 09$ for the velocities, while it is only $1.4e - 11$ for the generalized coordinates. The canonical momenta have an even smaller average

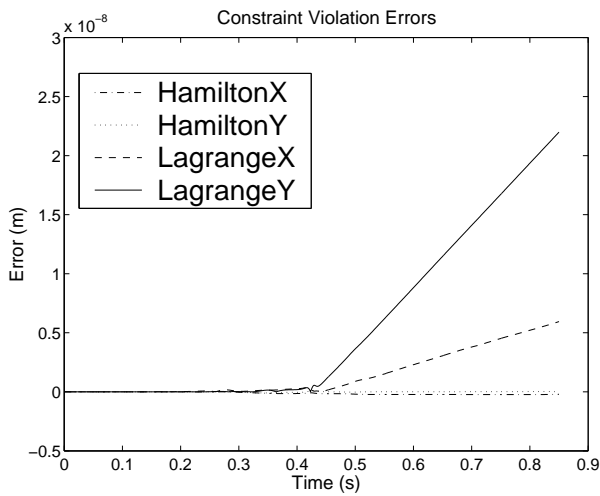


Figure 8. CONSTRAINT VIOLATION ERRORS FOR $\delta t=0.001s$.

Table 1. FIXED STEP INTEGRATION WITH RK45 METHOD.

Time step	Average end error		Mean local truncation error	
	Accelerations	Hamilton	Accelerations	Hamilton
0.01s	2.8e-03	1.2e-03	1.6e-06	9.7e-07
0.005s	1.9e-04	5.4e-05	4.3e-08	3.9e-08
0.001s	4.5e-08	1.2e-08	1.4e-11	1.3e-11
0.0005s	1.1e-09	3.8e-10	4.4e-13	4.2e-13
0.0001s	3.8e-11	2.9e-11	1.4e-16	1.3e-16

truncation error of $4.5e - 12$.

Numerical Integration With Adaptive Steps

The same simulation is now performed with two adaptive methods (ODE45 and ODE15S). Table 2 gives a list of simulations using different tolerance levels, both the absolute and the relative tolerances are set to the same value. There is one column for the average end error and two columns giving the number of function evaluations (F.E.) needed for the simulation. The function which is referred to is that which returns the accelerations ($\ddot{\mathbf{q}}$) or the derivatives of the canonical momenta and the generalized velocities ($\dot{\mathbf{p}}, \dot{\mathbf{q}}$). For example: the Runge-Kutta-Fehlberg method requires 6 F.E. for each time step (sometimes only 5 if the tolerance is not met, therefore the odd numbers in the table). The Hamiltonian equations clearly need to compute the equations of motion less often (30% to 40%).

Let's now take a closer look at the case of $1e - 06$ tolerance. As told in previous section, the truncation errors for the velocities are much higher than the errors for the coordinates. The numerical solver will therefore adapt its step size to the former errors, resulting in even smaller truncation errors for the coordinates.

Table 2. FIXED STEP INTEGRATION WITH RK4 METHOD

Tolerance	Method	End error	F.E. Lagrange	F.E. Hamilton
1e-04	ODE45	1e-02	313	206
1e-05	ODE45	4e-04	457	320
1e-06	ODE45	1e-05	661	434
1e-09	ODE45	4e-09	2545	1646
1e-12	ODE45	2e-11	10135	6560
1e-13	ODE45	2e-11	16063	10400
1e-03	ODE15S		336	235
1e-06	ODE15S		682	460
1e-09	ODE15S		1542	998
1e-12	ODE15S		7916	4815

Table 3. ADAPTIVE STEP SIZE WITH RK45 METHOD (TOL.= 10^{-6})

Tol= 10^{-6}	Mean trunc. err.		Average end err.		# of F.E.	
	Acc.	H	Acc.	H	Acc.	H
Full err. control	4.2e-09	4.4e-08	1.4e-05	2.0e-05	1578	1177
Partial err. control	4.7e-08	4.7e-08	4.8e-05	2.1e-05	1164	1159

On the other hand, the canonical momenta do not exhibit such big errors and their errors are treated at the same level as the coordinates. It is interesting to investigate what happens when the tolerance on the velocities and canonical momenta is relaxed or even removed (see table 3). With full error control, it is meant that all the state coordinates are checked for errors, while only the generalized coordinates are checked if partial control is applied. One can see that the errors on the velocities fully account for the extra computing cost using an adaptive method, while the overall accuracy does not deteriorate significantly. Figures 9 and 10 emphasize this: per time step of 0.01s, it has been calculated how many iterations were needed by the solver. The figures show the results for respectively full and partial error control. Ignoring the velocities while performing the error control is of course not recommended, but this example shows that an equal tolerance level for each state coordinate may not be the most judicious choice.

CONCLUSIONS

In this paper, a recursive algorithm for the forward dynamics of multibody systems was introduced. The algorithm is based on canonical momenta, resulting in a set of Hamiltonian equations. This formed the inspiration for a comparison of the behavior of acceleration based equations and Hamiltonian equations during numerical integration. A biped robot was chosen as example. It appeared that, while the accuracy is similar when using fixed step methods, the adaptive methods have a harder time tackling the acceleration based equations: more iterations are necessary while no significantly better accuracy is achieved. However, re-

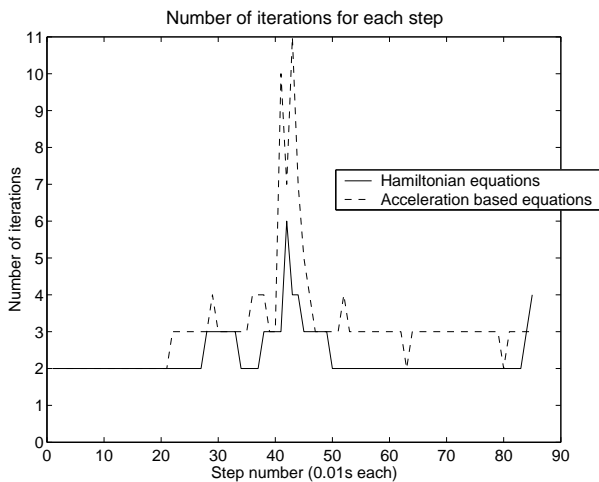


Figure 9. NUMBER OF ITERATIONS, FULL ERROR CONTROL.

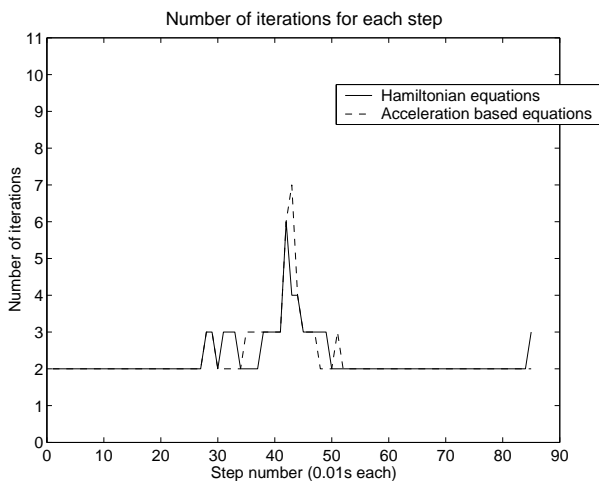


Figure 10. NUMBER OF ITERATIONS, PARTIAL ERROR CONTROL.

laxing the absolute tolerance on the coordinate velocities brings the number of iterations to the same level as in the case of Hamiltonian equations. One should thus consider using different tolerance levels for different state coordinates prior to simulation. It would be interesting to apply the same numerical analysis on other, more complex models and investigate whether the conclusions can be extended to them. This would help to gain more insight and make a better comparison between both sets of equations.

ACKNOWLEDGMENT

This research is supported by the Institute for the Promotion of Innovation by Science and Technology in Flanders (Belgium).

REFERENCES

- [1] Featherstone, R., 1987. *Robot Dynamics Algorithms*. Kluwer Academic Publishers.
- [2] Kane, T., and Levinson, D., 1985. *Dynamics: Theory and Applications*. McGraw-Hill.
- [3] Rosenthal, D., 1990. "An Order n Formulation for Robotic Systems". *The Journal of the Astronautical Sciences*, **38**(4), October-December, pp. 511–529.
- [4] Vukobratović, M., Filaretov, V., and Korzun, A., 1994. "A unified approach to mathematical modelling of robotic manipulator dynamics". *Robotica*, **12**, pp. 411–420.
- [5] Jerkovsky, W., 1978. "The Structure of Multibody Dynamics Equations". *Journal of Guidance and Control*, **1**(3), pp. 173–182.
- [6] Baraff, D., 1996. "Linear-Time Dynamics using Lagrange multipliers". *Computer Graphics Proceedings, Annual Conference Series*, pp. 137–146.
- [7] Anderson, K., and Crouch, J., 2003. "Order- N Performance Algorithm for the Simulation of Constrained Multi-Rigid-Body Dynamic Systems". *Multibody System Dynamics*, **9**, pp. 185–212.
- [8] Samin, J.-C., and Fiset, P., 2003. *Symbolic Modeling of Multibody Systems*. Kluwer Academic Publishers.
- [9] Lankarani, H., and Nikravesh, P., 1988. "Application of the Canonical Equations of Motion in Problems of Constrained Multibody Systems with Intermittent Motion". *Advances in Design Automation*, **1**, pp. 417–423.
- [10] Bayo, E., and Avello, A., 1994. "Singularity-Free Augmented Lagrangian Algorithms for Constrained Multibody Dynamics". *Nonlinear Dynamics*, **5**, pp. 209–231.
- [11] Rodríguez, J., Jiménez, J., Funes, F., and de Jalón, J., 2004. "Recursive and Residual Algorithms for the Efficient Numerical Integration of Multi-Body Systems". *Multibody System Dynamics*, **11**(4), pp. 295–320.
- [12] Novodex, 2004. "<http://www.novodex.com/>".
- [13] Naudet, J., Lefebvre, D., and Terze, Z., 2003. "Forward Dynamics of Open-Loop Multibody Mechanisms Using an Efficient Recursive Algorithm Based on Canonical Momenta". *Multibody Systems Dynamics*, **10**(1), pp. 45–59.
- [14] Naudet, J., Lefebvre, D., and Terze, Z., 2003. "General Formulation of an Efficient Recursive Algorithm based on Canonical Momenta for Forward Dynamics of Open-Loop Multibody Systems". *ECCOMAS Thematic Conference on Multibody Systems*, pp. 2003–33.
- [15] Naudet, J., and Lefebvre, D., 2005. "General Formulation of an Efficient Recursive Algorithm based on Canonical Momenta for Forward Dynamics of Closed-Loop Multibody Systems". *Proceedings of the XI DINAME (to be published)*.
- [16] Goldstein, H., 2002. *Classical Mechanics (3rd edition)*. Addison-Wesley Publishing Company.
- [17] Pars, L., 1965. *A Treatise of Analytical Dynamics*. William

Heinemann.

- [18] de Jalón, J. G., and Bayo, E., 1994. *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge*. Springer-Verlag, New York.
- [19] Baumgarte, J., 1983. “A New Method of Stabilization for Holonomic Constraints”. *ASME Journal of Applied Mechanics*, **50**, pp. 869–870.
- [20] Haug, E., 1989. *Computer-Aided Kinematics and Dynamics of Mechanical Systems. Volume I: Basic Methods*. Allyn and Bacon.
- [21] Blajer, W., 2002. “Elimination of Constraint Violation and Accuracy Aspects in Numerical Simulation of Multibody Systems”. *Multibody System Dynamics*, **7**, pp. 265–284.
- [22] Terze, Z., Lefeber, D., and Muftić, O., 2001. “Null space integration method for constrained multibody systems with no constraint violation”. *Multibody System Dynamics*, **6**(3), pp. 229–244.
- [23] Brenan, K., Campbell, S., and Petzold, L., 1989. *The Numerical Solution of Initial Value Problems in Differential-Algebraic Equations*. Elsevier Science Publishing.
- [24] Ruhoff, P., Præstgaard, E., and Perram, J., 1996. “Efficient algorithms for simulating complex mechanical systems using constraint dynamics”. *Proc. Roy. Soc. Lond. A*, **452**, pp. 1139–1165.
- [25] Stejskal and Michael Valášek, V., 1996. *Kinematics and Dynamics of Machinery*. Marcel Dekker, Inc.
- [26] Naudet, Dirk Lefeber and Zdravko Terze, J., 2003. “Forward Dynamics of Open-Loop Multibody Mechanisms Using an Efficient Recursive Algorithm Based on Canonical Momenta”. *Multibody Systems Dynamics*, **10**(1), pp. 45–59.