A variable time-step and variable penalty method for the index-3 augmented Lagrangian formulation with velocity and acceleration projections

Daniel Dopico[1], Emilio Sanjurjo[1], Javier Cuadrado[1], Alberto Luaces[1]

[1] Laboratorio de Ingeniería Mecánica,
University of A Coruña.
Mendizábal s/n, 15403. Ferrol, Spain.
{ddopico, emilio.sanjurjo, javier.cuadrado, aluaces}@udc.es

Abstract

The Index-3 augmented Lagrangian formulation with velocity and acceleration projections (the ALI3-P formulation) is an efficient and robust method to carry out the forward dynamics simulation of multibody systems modeled in dependent coordinates. The ALI3-P formulation was recently extended to non-holonomic systems and it was extensively used for the real-time simulation of different systems with human- and hardware-in-the-loop, some of them including complex phenomena like flexibility, contact with friction or non-holonomic constraints.

For almost all the applications tackled so far, a constant time step was always employed, but for the case of non-real-time systems with intermittent contacts of high stiffness this is not the right approach, since the time step selected has to achieve the accuracy and robustness required in the hardest part of the whole simulation, thus affecting the overall simulation speed. On the contrary, a variable time step can speed-up the simulation when the integration is easier, allowing very small time steps when the integration is harder.

Even if there is a vast literature on variable time step algorithms and at least one previous work for the mentioned formulation, the relation between time step and penalty factors is not covered. This relation is crucial if a broad range of time steps are required. In this work, a variable time step and variable penalty algorithm is proposed, specially suited for the formulation.

Finally, the algorithms developed are applied to the simulation of a real machine involved in a real situation: a diesel forklift rollover. The model includes some problematic phenomena from the integration point of view, like tire forces and intermittent contacts with friction when the machine rolls over and slams into the pavement.

1. Introduction

There are a vast amount of formulations of the equations of motion based on different techniques to enforce the constraints [1]. Among them, the index-3 augmented Lagrangian formulation with projections of velocities and accelerations (the ALI3-P formulation) uses a constraint violation elimination technique based on geometric projections of velocities and accelerations onto the constraint manifolds.

The ALI3-P formulation constitutes an efficient and robust scheme for the dynamics of multibody systems. It combines a reduced number of equations of motion, an implicit integration scheme along with Newton-Raphson iterations, and projections of velocity and accelerations onto their constraint manifolds to enforce position, velocity and acceleration constraints [2]. The ALI3-P formulation was recently extended to non-holonomic systems in [3] and it was extensively used for the real-time simulation of different systems with human- and hardware-in-the-loop, some of them including complex phenomena like flexibility, contact with friction or non-holonomic constraints.

One of the main disadvantages of the formulation comes from the fact that it is derived from the penalty formulation [4] and, therefore, it includes penalty factors in the equations of motion and their selection is problematic, especially for systems with very different masses connected, demanding different penalties for each constraint. The selection of penalty factors is a topic not properly covered yet in the literature. Moreover, the acceptable values of the penalty factors for the ALI3-P formulation in order to obtain stable and accurate solutions strongly depend on the integration time step selected.

For most of the applications tackled so far, a constant time step was employed but, for the case of non-real-time systems with intermittent contacts of high stiffness, this is not the right approach, since the time step selected has to achieve the accuracy and robustness required in the hardest part of the whole simulation, thus affecting the overall simulation speed. On the contrary, a variable time step can speed-up the simulation when the integration is easier, allowing very small time steps when the integration becomes harder. In [5], a variable time-step method for the ALI3-P and ALI1-P (Index-1 Augmented Lagrangian with projections) was proposed, nevertheless aspects like the selection of the penalty factor and the way to estimate the integration error for the iterative scheme of the implicit integrator, were not properly considered.

It is very important for the ALI3-P formulation, to relate the time step to the penalty factors selected if a broad range of time steps are required. In this work, a variable time step and variable penalty algorithm is proposed, taking advantage of the fact that the integrator is implicit to estimate the integration error and to establish a strategy to modify these parameters. The strategy to vary the parameters is going to be explored in the next sections.

2. ALI3-P formulation for holonomic systems

The Index-3 Augmented Lagrangian formulation with Projections (ALI3-P) for non holonomic systems was introduced in [3], based on the index-3 formulations described in [2] and [6]. The formulation is summarized here for holonomic systems.

Let us consider a multibody system modeled with $\mathbf{q} \in \mathbb{R}^{n_c}$ coordinates related by m holonomic constraints (some of them rheonomic). The equations of motion have the following expressions

$$[\mathbf{M}\ddot{\mathbf{q}}]_{\delta m} + \left[\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}^{*(i+1)} + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}\mathbf{\Phi}\right]_{\delta f} = \mathbf{Q}_{\delta f} \tag{1a}$$

$$\boldsymbol{\lambda}_{n+1}^{*(i+1)} = \boldsymbol{\lambda}_{n+1}^{*(i)} + \boldsymbol{\alpha}\mathbf{\Phi}_{n+1}^{(i+1)};\, i > 0 \tag{1b}$$

where

$$[\cdots]_{\delta m} = (1-\delta_m)\,[\cdots]_{n+1}^{(i)} + \delta_m[\cdots]_n \tag{2a}$$

$$[\cdots]_{\delta f} = \left(1-\delta_f\right)[\cdots]_{n+1}^{(i)} + \delta_f[\cdots]_n \tag{2b}$$

$$[\cdots]_{\delta f}^{(i+1)} = \left(1-\delta_f\right)[\cdots]_{n+1}^{(i+1)} + \delta_f[\cdots]_n \tag{2c}$$

and $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n_c \times n_c}$ is the mass matrix, $\mathbf{Q}(\mathbf{q},\dot{\mathbf{q}},t) \in \mathbb{R}^{n_c}$ is the vector of generalized forces, $i = 0,1,2,...,$ $\boldsymbol{\alpha}$ is a diagonal matrix that contains the penalty factors associated with the constraints, δ_f and δ_m are scalar parameters of the generalized-α method, n is the time step index, i is the iteration index of the approximate Lagrange multipliers $\boldsymbol{\lambda}_{n+1}^{*}$.

The time-stepping equations for the method are the Newmark expressions [7] and substituting them into the equations of motion (1a), a nonlinear system of equations for $\mathbf{q}_{n+1}$ is obtained, which can be solved by means of a Newton-Raphson iteration. The detailed expressions are provided in Figure 1.

Equations (1) enforce constraint equations at position level only. In order to enforce velocity and acceleration level constraints, the formulation makes use of velocity and acceleration projections. The general form of the velocity and acceleration projections was provided in [3] but, for holonomic systems, the simplified non-iterative expressions have the following form:

$$\left(\mathbf{P} + \varsigma\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}\mathbf{\Phi}_{\mathbf{q}}\right)\dot{\mathbf{q}} = \mathbf{P}\dot{\mathbf{q}}^{*} - \varsigma\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}\mathbf{\Phi}_{t} \tag{3a}$$

$$\left(\mathbf{P} + \varsigma\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}\mathbf{\Phi}_{\mathbf{q}}\right)\ddot{\mathbf{q}} = \mathbf{P}\ddot{\mathbf{q}}^{*} - \varsigma\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}\left(\dot{\mathbf{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} + \dot{\mathbf{\Phi}}_{t}\right) \tag{3b}$$

In equations (3), $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are the projected magnitudes while $\dot{\mathbf{q}}^{*}$ and $\ddot{\mathbf{q}}^{*}$ are the magnitudes coming from the solution of the equations of motion (1) before the projection.

For the selection of the projection matrix $\mathbf{P}$ and the constraints weighting parameters ς, the following choice is assumed here:

$$\mathbf{P} = \mathbf{M}_{n+1} + \frac{1-\delta_f}{1-\delta_m}\left(\gamma h\mathbf{C}_{n+1} + \beta h^2\mathbf{K}_{n+1}\right) \tag{4}$$

$$\varsigma = \frac{1-\delta_f}{1-\delta_m}\beta h^2, \tag{5}$$

where $\mathbf{K} = -\partial\mathbf{Q}/\partial\mathbf{q}$ and $\mathbf{C} = -\partial\mathbf{Q}/\partial\dot{\mathbf{q}}$ are the stiffness and damping matrices.

Figure 1 summarizes the steps required to implement the formulation for a constant time step.

3. Variable penalty algorithm

It is a well known fact that one of the main disadvantages of penalty and augmented Lagrangian formulations is the selection of the penalty factors needed to enforce the constraints, and this problem can dissuade inexpert users from using this kind of approaches.

In the case of pure penalty formulations, the selection of the penalty factors required has an impact on both the accuracy and the stability of the integration of the equations of motion. In the case of the ALI3-P formulation used here, the

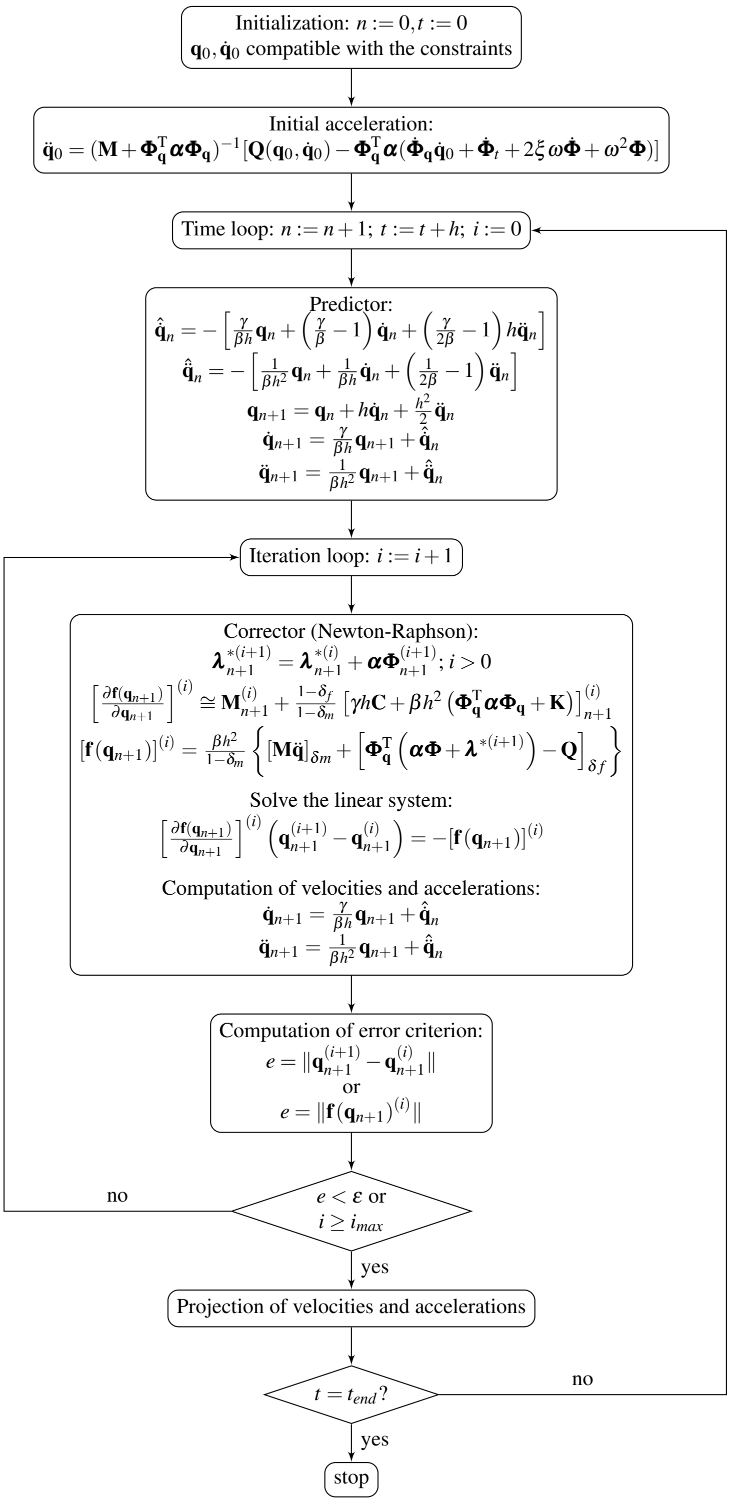

Figure 1: Flowchart for the ALI3-P formulation with fixed time step.

effect is more on the side of the stability than on the accuracy and improper selections can easily lead to slow down the convergence of the Newton-Raphson algorithm or even to the integration process failure. The selection of penalty factors is particularly difficult for systems with very different frequencies due to small and big masses connected to each other. This aspect is not properly covered in the bibliography.

Another uncovered aspect is the relation between penalty factors and time step for implicit integrators based on residual-tangent matrix schemes like the one proposed in this work. In this case the selection of the penalty factor has more to do with numerical aspects of the equations. The key is the scaling of tangent matrix terms in the corrector of Figure 1, recalled here

$$\left[\frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}}\right] \cong \mathbf{M} + \frac{1-\delta_f}{1-\delta_m}\left[\gamma h \mathbf{C} + \beta h^2 \left(\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}} \boldsymbol{\alpha} \mathbf{\Phi}_{\mathbf{q}} + \mathbf{K}\right)\right] \tag{6}$$

Observe, in (6), that the best possible situation is to achieve the same order of magnitude in every term of the tangent matrix. Assuming that the stiffness and damping matrices, $\mathbf{K}$ and $\mathbf{C}$, are fixed, the order of magnitude of the term $\frac{1-\delta_f}{1-\delta_m}\beta h^2 \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}} \boldsymbol{\alpha} \mathbf{\Phi}_{\mathbf{q}}$ should be close to the order of magnitude of $\mathbf{M}$.

Let us suppose a hypothetical system with $\|\mathbf{M}\| \approx 1$, and terms in the Jacobian transposed times Jacobian such that $\|\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}} \mathbf{\Phi}_{\mathbf{q}}\| \approx 1$. Selecting a typical time step $h_n = 10^{-3}$ s, a scalar penalty factor $\alpha_n = 10^6$ and integrator coefficients $\delta_f = 0$, $\delta_m = 0$, $\beta = 1/4$

$$\frac{1-\delta_f}{1-\delta_m}\beta h_n^2 \|\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}} \boldsymbol{\alpha}_n \mathbf{\Phi}_{\mathbf{q}}\| = \frac{1}{4} \tag{7}$$

Observe that the order of magnitude of the term is comparable to the order of magnitude of the masses. Now, a variation of the time step from h_n to h_{n+1} will affect as follows

$$\frac{1-\delta_f}{1-\delta_m}\beta h_{n+1}^2 \|\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}} \boldsymbol{\alpha}_n \mathbf{\Phi}_{\mathbf{q}}\| = \frac{1}{4}\left(\frac{h_{n+1}}{h_n}\right)^2 \tag{8}$$

The following update formula for the penalty factor aims at keeping the same order of magnitude in the term

$$\boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n \left(\frac{h_n}{h_{n+1}}\right)^2 \tag{9}$$

Therefore,

$$\frac{1-\delta_f}{1-\delta_m}\beta h_{n+1}^2 \|\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}} \boldsymbol{\alpha}_{n+1} \mathbf{\Phi}_{\mathbf{q}}\| = \frac{1}{4} \tag{10}$$

Equation (9) keeps the same order of magnitude for the penalty term, in the tangent matrix and, therefore, without further knowledge of the tangent matrix, it can be a good option to update the penalty factors in terms of the variation of the time step. The mathematical interpretation of the expression is that smaller time steps require much bigger penalty factors.

4. Variable time step and variable penalty algorithm

It was mentioned before that for almost all the applications tackled so far, a constant time step was employed. This approach is the right one for systems with human [8] or hardware-in-the-loop applications, but it is not always reasonable for offline simulations, especially for those with intermittent contacts of high stiffness or abrupt maneuvers in which the tire forces can vary quickly, like the simulation solved in this work. In this case, a variable time step can speed-up the simulation when the integration is easier, allowing very small time steps if one or more contacts appear, or when the maneuver is more demanding in terms of tire forces and the integration of the equations of motion becomes harder.

Classical approaches base the time step variation strategy on the estimation of the local truncation error of the integration process, either by integrating the current time step using two different order methods or by integrating two successive time intervals with different step sizes [9, 10]. These classical strategies require of an error estimator for the integrator considered and some extra function evaluations. In case of predictor corrector schemes, it is possible to take advantage of the predictor and corrector expressions to estimate this error [10].

Instead of this, in [5] a more physical criterion was proposed: for non-conservative systems, the energy invariant integral was established as error estimator for the time step variation strategy, thus avoiding the need for pure mathematical estimators and extra function evaluations. Nevertheless, they do not consider important aspects like how to combine the proposed estimator with the predictor-corrector implicit integration scheme based on a Newton-Raphson iteration.

In this paper an approach specially suited to the implicit scheme described in Figure 1 is proposed. The approach requires minimum changes with respect to the constant time step scheme in order to keep the computational overhead associated to the variable time step algorithm almost negligible.

Playing the role of integrator error estimators, the same errors proposed for the fixed time step algorithm are used.

$$e = \|\mathbf{q}_{n+1}^{(i+1)} - \mathbf{q}_{n+1}^{(i)}\| \tag{11a}$$

or

$$e = \|\mathbf{f}(\mathbf{q}_{n+1})^{(i)}\| \tag{11b}$$

In order to calculate the new time step based on the previous time step, the following expression is used

$$h_{n+1} = r\, h_n \tag{12a}$$

$$r = \left(\frac{\nu\varepsilon}{e}\right)^{\frac{1}{p+1}} \tag{12b}$$

where h_{n+1}, h_n are the new and old time steps, ε is the constraint tolerance, $p = 2$ is the order of the integration method and $\nu \approx 0.9$ is a safety factor aimed at a conservative selection of the new time steps.

For the penalty matrix update, $\boldsymbol{\alpha}$, expression (9) is used.

A flowchart of the variable time step algorithm is provided in Figure 2. Only the relevant details for the variable time step strategy are included, because the equations at every stage of the algorithm are the same already included in the flowchart of Figure 1. For details about the different steps in the flowchart, the reader is referred to Figure 1.

5. Diesel forklift simulation

In this section, the algorithms developed have been applied to the simulation of a diesel forklift involved in a real situation: a reversing maneuver resulting in the machine rollover. The model includes some problematic phenomena from the integration point of view, like abrupt tire forces and intermittent contacts with friction when the machine rolls over and slams into the pavement.

5.1. Multibody model

The multibody model is composed of 10 bodies (see Figure 3 and table 1). The model was built in the MBSLIM software package with mixed coordinates: 90 natural coordinates (12 points and 18 unit vectors) plus 9 angles and 2 distances, making a total of $n = 101$ coordinates. A total number of $d = 11$ mechanical degrees of freedom (DOF) have been considered: 6 DOF for the chassis rigid body motion, 4 DOF for the wheels rotations and 1 DOF for the rear rigid axle tilt. Besides, 5 additional kinematically guided motions, corresponding to degrees of freedom in the real machine have been considered: the mast swinging angle, the fork and upper mast lift and the rear wheels steering angles (related by the Ackerman steering condition). For these guided motions, rheonomic constraints have been employed and therefore they cannot be considered as degrees of freedom from the mechanical point of view, nevertheless they are true additional DOF in the real machine that have been considered, making a total of 16 real degrees of freedom.

The total number of constraints is $m = 103$, that means $m - (n - g) = 13$ redundant constraints present in the model, which is a typical situation modeling in MBSLIM.

The same model described in [11] was used for the tire forces with an improved collision detection algorithm that makes possible to come into contact on the tire shoulder, essential for rollover situations. The collisions between chassis or cabin and pavement have been modeled too, with the same approach proposed in [8].

5.2. Numerical results

The maneuver consists in the machine moving straight backward at full speed (10 km/h initially). One second after the simulation starts, the driver turns right completely.

Some snapshots of the interactive simulation are included in Figure 4, proving that the careless maneuver results in the machine rollover. Some important magnitudes are represented in 3D: the green vectors are normal tire forces, the red ones are tangential tire forces, the yellow ones are center of mass accelerations for each body, a black vector represents the acceleration suffered by the driver and the purple one represents the velocity of the central point in the front axle.

The maneuver can be understood by examining Figure 5. It starts following a straight backward direction and after 1 s, the driver turns right during 1 second (red line), keeping the steering completely turned for the rest of the maneuver.

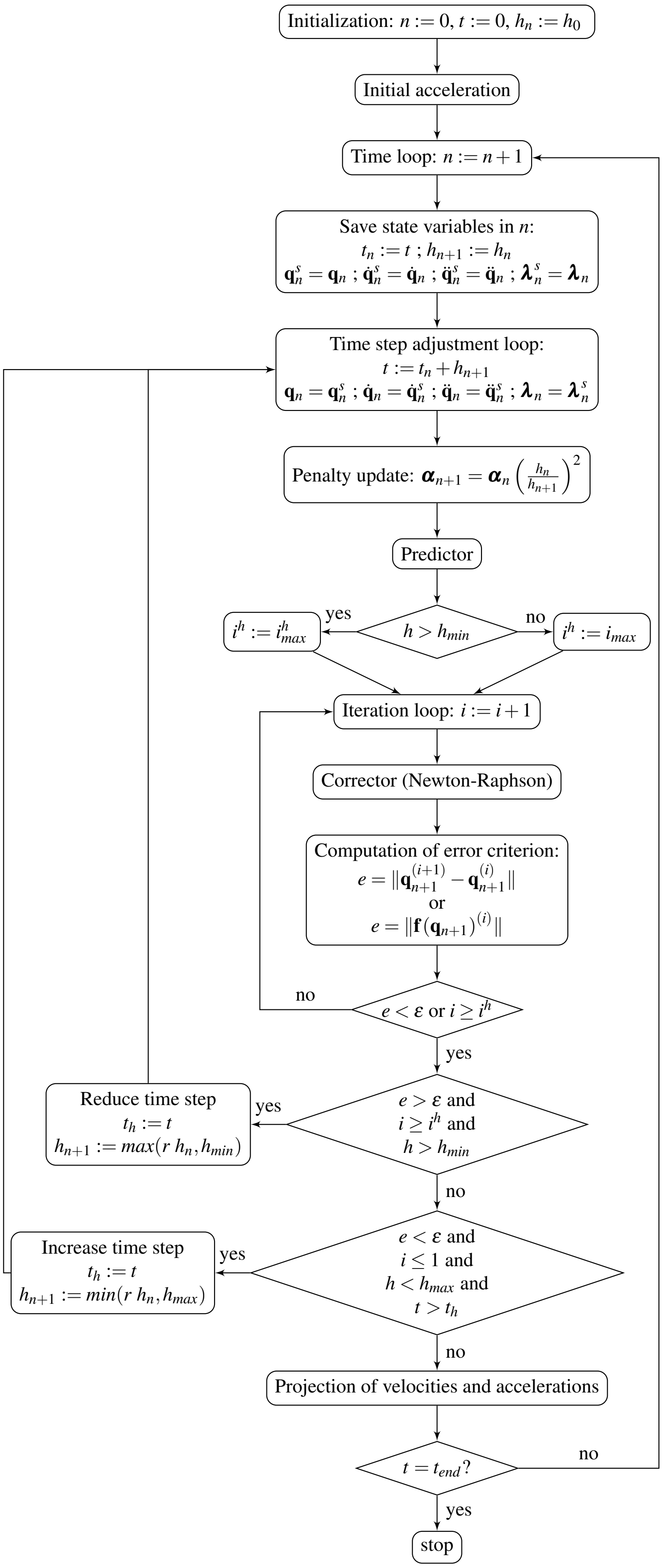

Figure 2: Flowchart for the variable time step algorithm.

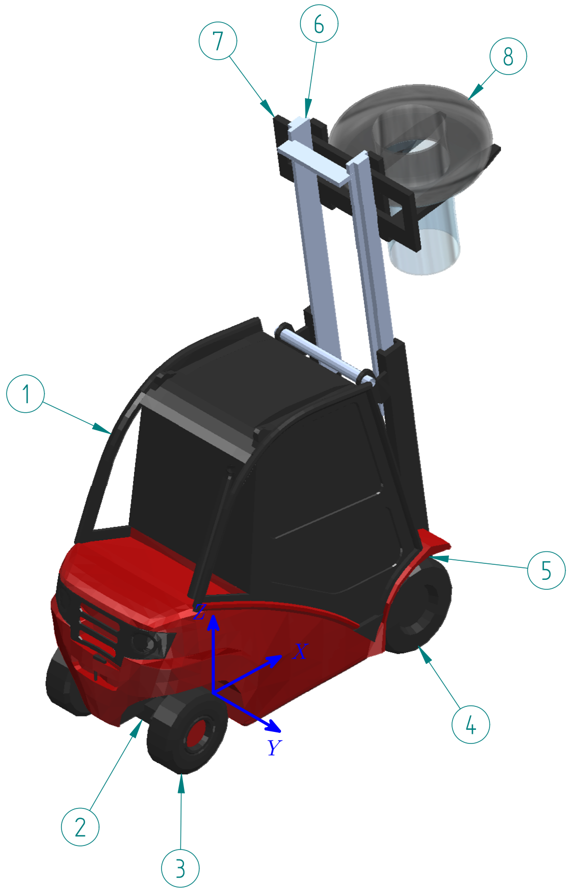

Figure 3: Diesel forklift model

The load is initially lifted at 3.2 m high (black line). Around $t = 1.5$ s, the machine starts loosing stability (see blue and green lines corresponding to the z coordinate of the right wheels centers) and the load hits the pavement at 4.25 s. Figure 6 shows the trajectory followed by the machine.

The evolution of the time step during the simulation is shown in Figure 7 for the proposed error criteria and two different error tolerances. The maximum and minimum time steps allowed were $h_{max} = 0.02$ s and $h_{min} = 10^{-4}$ s.

Observe that the algorithm reduces the time step size when the integration is harder, this happens especially when the load slams into the pavement. Moreover, with some combinations of positions criterion and tolerance, the algorithm increases the time step when the integration is easier, reaching very large values.

It is important to remark that the boundary value h_{min} is rarely reached in this simulation. Strangely, when the machine is overturned and completely stopped, the bound could be reached because some tires are contacting the ground and the tire model is not regularized for null velocities, and this lack of regularization explains the erratic behavior of the time step size at the end of the simulation for looser tolerances. Nevertheless, for tighter tolerances, the algorithm performs much better for very low velocities. This behavior, apparently contradictory, is perfectly understandable on the basis of the tire model implemented.

Compared to the fixed time step algorithm, the variable time step strategy performs much better in terms of efficiency and stability of the solution. For constant time step $h = 0.01$ s, the simulation crashes very soon and for $h = 0.001$ s the solution is very similar to that shown in Figure 5, but the overall efficiency is seriously affected.

Finally, the variation of the penalty factor proved to be crucial. Otherwise the simulation fails as well with the variable time step algorithm.

6. Conclusions

A variable time step and variable penalty algorithm, suited to the ALI3-P formulation, has been developed.

Table 1: List of bodies composing the model

Body ref.	Name	No. of bodies
1	Chassis and cabin	1
2	Rear rigid axle	1
3	Rear wheels	2
4	Front wheels	2
5	Lower mast	1
6	Upper mast	1
7	Fork	1
8	Load	1

Figure 4: Rollover situation.

The variation of the time step in this kind of formulations is problematic, but the ability of altering the penalty factor proved to be crucial in order to get the algorithm working. Without the penalty factors adjustment the simulations fail unless a very narrow range of time step sizes are allowed. The algorithm achieves much faster and more robust simulations than the fixed time step algorithms already existent for this formulation.

The algorithm developed was applied to the simulation of a real machine involved in a real situation: a diesel forklift rollover. The model includes some problematic phenomena from the integration point of view, like tire forces and intermittent contacts with friction when the machine rolls over and slams into the pavement.

References

[1] Bauchau, O. A., and Laulusa, A., 2007. "Review of contemporary approaches for constraint enforcement in multibody systems". *Journal of Computational and Nonlinear Dynamics,* ***3***(1), Nov., pp. 011005–011005.

[2] Bayo, E., and Ledesma, R., 1996. "Augmented lagrangian and mass-orthogonal projection methods for constrained multibody dynamics". *Nonlinear Dynamics,* ***9***(1-2), 02, pp. 113–130.

[3] Dopico, D., González, F., Cuadrado, J., and Kovecses, J., 2014. "Determination of holonomic and nonholonomic constraint reactions in an index-3 augmented lagrangian formulation with velocity and acceleration projections". *Journal of Computational and Nonlinear Dynamics,* ***9***(4), July, pp. 041006–041006.

[4] Bayo, E., García de Jalon, J., and Serna, M., 1988. "A modified lagrangian formulation for the dynamic analysis of constrained mechanical systems". *Computer Methods in Applied Mechanics and Engineering,* ***71***(2), 11, pp. 183–195.

[5] Cardenal, J., Cuadrado, J., Morer, P., and Bayo, E., 1999. "A multi-index variable time step method for the dynamic simulation of multibody systems". *International Journal for Numerical Methods in Engineering,* ***44***(11), 04, pp. 1579–1598.

[6] Cuadrado, J., Cardenal, J., Morer, P., and Bayo, E., 2000. "Intelligent simulation of multibody dynamics: Space-state and descriptor methods in sequential and parallel computing environments". *Multibody System Dynamics,* ***4***(1), 02, pp. 55–73.

[7] Newmark, N., 1959. "A method of computation for structural dynamics". *Journal of the Engineering Mechanics Division,ASCE*, pp. 67–94.

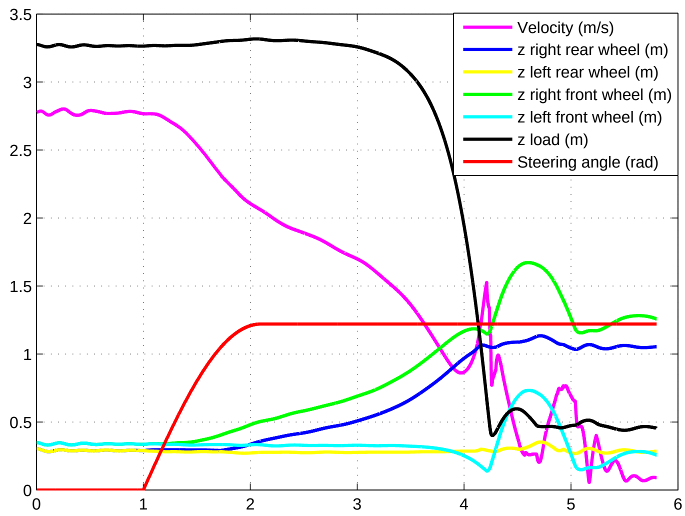

Figure 5: Maneuver results.

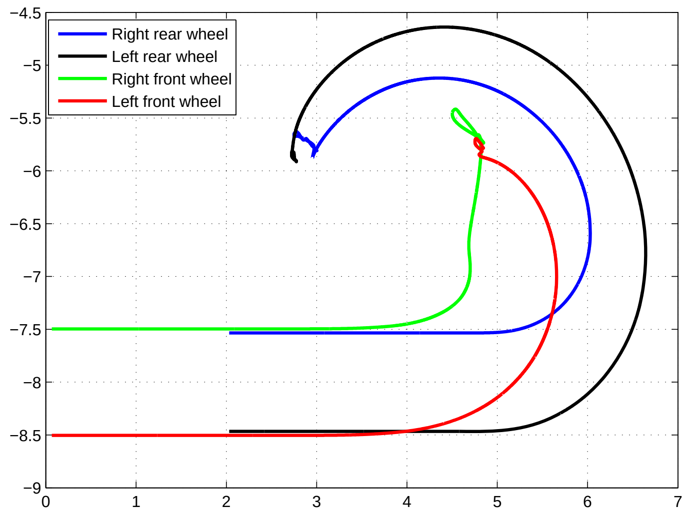

Figure 6: Wheels trajectories.

[8] Dopico, D., Luaces, A., Gonzalez, M., and Cuadrado, J., 2011. "Dealing with multiple contacts in a human-in-the-loop application". *Multibody System Dynamics,* ***25***(2), pp. 167–183.

[9] Hairer, E., Norsett, S., and Wanner, G., 1993. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer- Verlag, Berlin Heidelberg.

[10] Ascher, U., and Petzold, L., 1998. *Computer methods for ordinary differential equations and differential-algebraic equations*. Philadelphia Society for Industrial and Applied Mathematics.

[11] Zhu, Y., Dopico, D., Sandu, C., and Sandu, A., 2015. "Dynamic response optimization of complex multibody systems in a penalty formulation using adjoint sensitivity". *Journal of Computational and Nonlinear Dynamics,* ***10***(3), May, pp. 1–9.

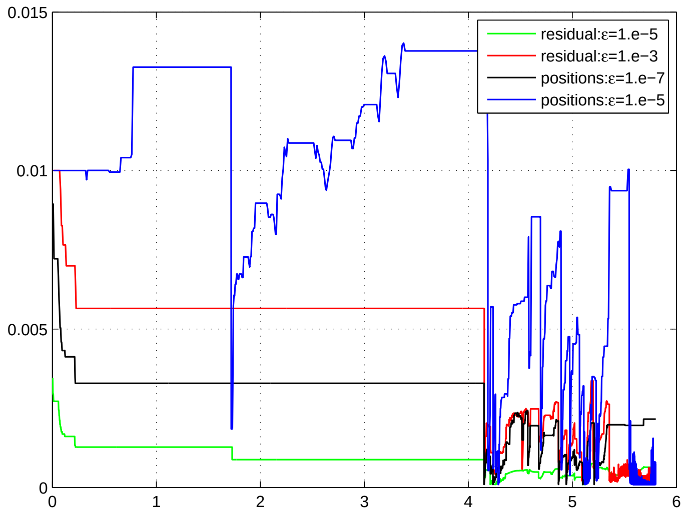

Figure 7: Time step evolution with residual error $e = \|\mathbf{f}(\mathbf{q}_{n+1})^{(i)}\|$ and positions error $e = \|\mathbf{q}_{n+1}^{(i+1)} - \mathbf{q}_{n+1}^{(i)}\|$