Direct integration of the equations of multibody dynamics through central differences and linearization

Gorka Urkullu[1], Igor Fernandez de Bustos[2], Vanessa Garcia-Marina[3], Haritz Uriarte[4]

[1]*Department of Mechanical Engineering, Faculty of Engineering of Bilbao, University of the Basque Country, Alameda de Urquijo s/n, 48013, Bilbao, Spain*
e-mail: gorka.urkullu@ehu.eus
ORCID: 0000-0002-7937-5627

[2]*Department of Mechanical Engineering, Faculty of Engineering of Bilbao, University of the Basque Country, Alameda de Urquijo s/n, 48013,Bilbao, Spain*
e-mail: igor.fernandezdebustos@ehu.eus

[3]*Department of Mechanical Engineering, University School of Engineering of Vitoria-Gasteiz, University of the Basque Country, Nieves Cano 12, 01006, Vitoria-Gasteiz, Spain*
e-mail: vanessa.garcia@ehu.eus

[4]*Department of Mechanical Engineering, Faculty of Engineering of Bilbao, University of the Basque Country, Alameda de Urquijo s/n, 48013, Bilbao, Spain*
e-mail: haritz.uriarte@ehu.eus

A methodology for integrating the rigid body dynamics for the analysis of multibody systems is presented. The novelty lies in the fact that the equations system is solved directly by means of central differences as second order integration method, whilst to obtain the best approximation each time instant the equilibrium is solved iteratively applying the exact Newton method. Thus, it is possible to achieve the system solution directly without having to reduce the differential order. This decreases the number of unknowns. In return, it is necessary to set out a linearization of the equations. The rotation of each element is described by parametrization under unit quaternion. In this paper the necessary developments for the modelization of the spherical and rotational joints are included. The imposed constraints by these joints, as well as by the quaternions norm, are introduced in the model through the null space matrix. The reactions produced by these constraints are eliminated from the system by the null space, too. Several examples have been analysed through the implementation of the methodology in Octave. This allowed to verify the accuracy of the method comparing with results from different commercial software. The examples also include benchmark problems.

Keywords: Multibody systems, central differences, exact Newton method, quaternion.

1. Introduction

There are several possible approaches for the resolution of the multibody systems dynamics, depending on the variables employed, the parametrization of the orientation of every solid used, the formulation of the dynamical equilibrium followed to achieve the equations to integrate, the method applied for the integration of the equations, and the methodology chosen for the treatment of the constraints. [1–4]. The Principles of mechanics such as Lagrange's equations, Newton's laws, Hamilton's canonical equations or virtual powers constitute the basis for formulating the multibody systems dynamics and lead to the final form of the motion equations. These equations together with the constraints imposed by the joints form a system of differential algebraic equations (DAE's) [5,6].

A common way to solve the problem is to transform it in a system of ordinary differential equations (ODE’s), for which there are several options [7]. In the stabilized Lagrange formulation the constraint equations are derived twice. Afterwards the Baumgarte stabilization [8] is applied to avoid the breaking down of the system, due to constraint drift. Another solution method for constraint optimization problems is the well known augmented Lagrange method, Bayo applied it to the dynamics of constrained mechanical systems [9]. In this paper is described the use of penalty factors as a technique to enforce the satisfaction of kinematic constraints with a Lagrangian formulation. A wide definition of both formulations and their implementation in commercial codes is included in [10], by Cuadrado. In [11], Blajer exhibits in its work a geometric representation of the formulation of the augmented Lagrange, as well as the application in systems with singularities. Another interesting approach can be found in González [12], who uses the penalty factors present in the augmented Lagrange formulation with the aim to determine the reaction forces in redundant constrained dynamic systems, in order to introduce the physical properties of the system in the model. On this point it is worth noting the work of Pappalardo [13], who examines different computational methods including augmented formulation and, demonstrates that the algorithm based on the combination of the natural absolute coordinate formulation [14] with the Udwadia-Kalaba equations is the most effective. In the vast majority of works the Lagrange equations are taken as starting point but it is also possible to employ the canonical equation of Hamilton which leads to the formulation of augmented Hamilton as in [15]. Another formulation to reduce the system to the minimum variables number and reach to an ODE is through the null space of the constraints Jacobian, as can be found in [16]. In this respect, an excellent revision about the null space formulation can be found in [17] by Laulusa. Furthermore, Jalón in [18], it is contributed to the resolution and clarification of the multiplicity of solutions for the constraint equations and it is focused on three method mainly with each other the null space. The advantage of the null space formulation is that the Lagrange multipliers are eliminated, however, it has a disadvantage against the augmented Lagrange method and it is the need to solve the position and velocity problems of the mechanism in every instant. In the same way, other formulations can be mentioned such as the Kane method [19], which also leads to a system with the minimum set of coordinates. Once the DAE's system is transformed to get an ODE's system, It is frequent to reduce it to an order 1 system in exchange for increasing its dimension, afterwards solving it with the numerical tools existing for the purpose; these can be explicit methods such as Runge-Kutta or Adams-Bashforth, or implicit methods including the Trapezoidal Rule or Adams-Moulton.

Another alternative is to solve the DAE's through methods of direct integration. Traditionally, it has been employed methodologies based on backward differentiation formulas (BDF) [20–22]. The implementation of BDF in general purpose solvers presents serious numerical difficulties and these are more severe for index three problems such as constrained multibody systems. Furthermore, it is also possible to find works in which BDF have been replaced by another type of formulation. In [23] for example, unlike approaches based on the resolution of order 2 systems such as [24,25] or stabilization as in [26,27], an algorithm is presented that uses the implicit method of Newmark [28], to discretize the order 3 DAE's directly; as a consequence this requires that the constraints be fulfilled in every time interval, resulting in a non-linear algebraic system that is solved by some method of the quasi Newton type. In [29], the implementation of the method of Hilber-Hughes-Taylor [30], is presented, which is suitable for the solution of DAE's in the simulation of mechanical systems with contacts and flexible bodies. In addition to BDF, it is also used the implicit method of Runge-Kutta [31,32], being the trapezoidal method one of the best known. On the one hand, the main advantages are that implicit Runge-Kutta does not suffer from system discontinuities or changes in the order or time step as BDF. On the other hand, the resulting set of equations is quite large and involved.

The vast majority of these method reduce the differential order of the equations system to apply the integration method at the expense of increasing the size of the system. One of the target of this work is to integrate directly the second order differential equation system. Obviously, the integration method has to be able to accurately integrate up to second order. Normally, the model is defined with dependant coordinates, hence, another target is to obtain a system with a minimum set of variables but without having to solve the position or velocity problem to calculate all of coordinates of the system. The equations of movement presents one nonlinearity on account of the product of the velocities. It is common to approach the integration algorithm in terms of the acceleration and solve the system avoiding the linearization of the nonlinear equations. The third target is to improve the accuracy of the solution by linearizing the equations in an exact manner without using an approximate form.

In this work to be able to accomplish the targets and evaluate the methodology in a more diverse set of systems, the Cartesian coordinates are used by locating the centre of the reference system in the centre of mass of the body. Due to reasons of algebraic simplicity and computational efficiency, the unit representation of the quaternion is a convenient choice to define the orientation of each element [33]. The system of equations obtained through the equations of Newton-Euler is solved directly by a numerical integrator. The method employed is central differences, being a fast and easy to implement method. The benefit is to avoid having to duplicate the dimension of the system to reduce the differential order of the equations, as is usually done. In order to find the best approximation at each instant of time the system is solved analytically by means of exact Newton method [34–36]. Not only the nonlinear Newton-Euler equations are linearized but also the constraint equations. As it is known, this iterative process finds successively better approximation to the roots by means of the exact derivative of the function. Therefore, in this work, it will denominate exact Newton to differentiate it from other procedures that use approximate derivatives such as quasi Newton. On that account, the difference between the well-known direct application of the trapezoidal method (TM) to the index-three form lies in the use of exact Newton method apart from TM is an implicit method. A similar case of TM application can be found in Gavrea [23]. The algebraic constraints equation are introduced in the system by using the null space and hence, reducing the model dimension and eliminating the Lagrange multipliers. This fact allows to obtain a less demanding integration compared to augmented Lagrange or penalty formulation and it is possible to implement an explicit method. On account of having applied a second order numerical integrator, it is not necessary to resolved the position and velocity problem of the complete mechanism in every instant of time so that the calculation time is improved. This provides an important advantage compared with the classical null space formulation. The validity of the method is demonstrated by the resolution of several mechanical systems included in the study carried out by González [37], for the standardization of multibody problems and for the evaluation of the error. The methodology has been implemented with a preliminary programming in Octave [38]. Finally, the results obtained are compared to commercial software such as Msc Adams. Due to the size of problems and the type of programming language used, the comparison is focused only on the accuracy.

This work is organized as follows: the calculation of the basic magnitudes is included in the section 2. The rotation of the body is express by unit quaternion, therefore, it is convenient to define previously some properties. Section 3 presents the approach of the dynamic equations. The linearization of the Newton-Euler equations by exact Newton method provides an adequate and accurate expression to apply the integration method. In section 4, the central different method is used to solve the equation of movement and the in the section 5 the constraint equation is introduced by null space method. This leads to the final expression where the constraint reactions force have been wiped out and the independent coordinates are the only unknown variables. Section 5 also details the constraint equations of the quaternion norm as well as the spherical and rotation joint. The discussion of numerical application is shown in section 6. Three benchmarking examples have been developed including spatial redundantly constrained systems and mechanism with singular positions.

2. Calculation of basic magnitudes

In this section the basic magnitudes needed to raise the equations of the movement will be calculated, and after that, the constraint equations will be established.

2.1. Quaternion

A unit quaternion defines the orientation of a body respect to the global reference frame and may be represented in matrix form by 4-dimensional column matrix:

$$\boldsymbol{q}=\{q_0 \quad q_1 \quad q_2 \quad q_3\}^T \tag{1}$$

Applying the Rodrigues equation [39] to the unit quaternion, it can be obtained the rotation matrix:

$$R(\boldsymbol{q})=\begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1 q_2-q_0 q_3) & 2(q_1 q_3+q_0 q_2) \\ 2(q_1 q_2+q_0 q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2 q_3-q_0 q_1) \\ 2(q_1 q_3-q_0 q_2) & 2(q_2 q_3+q_0 q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{2}$$

At the same time, the angular velocity and acceleration matrices are defined in world-fixed frame as follows:

$$\Omega=\dot{R}(\boldsymbol{q})R(\boldsymbol{q})^T=2\begin{bmatrix}0 & -q_1\dot{q}_2+q_2\dot{q}_1+q_3\dot{q}_0-q_0\dot{q}_3 & -q_1\dot{q}_3+q_3\dot{q}_1+q_0\dot{q}_2-q_2\dot{q}_0\\ -q_2\dot{q}_1+q_1\dot{q}_2+q_0\dot{q}_3-q_3\dot{q}_0 & 0 & -q_2\dot{q}_3+q_3\dot{q}_2+q_1\dot{q}_0-q_0\dot{q}_1\\ -q_3\dot{q}_1+q_1\dot{q}_3+q_2\dot{q}_0-q_0\dot{q}_2 & -q_3\dot{q}_2+q_2\dot{q}_3+q_0\dot{q}_1-q_1\dot{q}_0 & 0\end{bmatrix} \tag{3}$$

And

$$\dot{\Omega}=2\begin{bmatrix}0 & -q_1\ddot{q}_2+q_2\ddot{q}_1+q_3\ddot{q}_0-q_0\ddot{q}_3 & -q_1\ddot{q}_3+q_3\ddot{q}_1+q_0\ddot{q}_2-q_2\ddot{q}_0\\ -q_2\ddot{q}_1+q_1\ddot{q}_2+q_0\ddot{q}_3-q_3\ddot{q}_0 & 0 & -q_2\ddot{q}_3+q_3\ddot{q}_2+q_1\ddot{q}_0-q_0\ddot{q}_1\\ -q_3\ddot{q}_1+q_1\ddot{q}_3+q_2\ddot{q}_0-q_0\ddot{q}_2 & -q_3\ddot{q}_2+q_2\ddot{q}_3+q_0\ddot{q}_1-q_1\ddot{q}_0 & 0\end{bmatrix} \tag{4}$$

Hence, the vectors corresponding to angular velocity and acceleration are:

$$\boldsymbol{\omega}=\begin{Bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{Bmatrix}=2\begin{Bmatrix}-q_3\dot{q}_2+q_2\dot{q}_3+q_0\dot{q}_1-q_1\dot{q}_0\\ -q_1\dot{q}_3+q_3\dot{q}_1+q_0\dot{q}_2-q_2\dot{q}_0\\ -q_2\dot{q}_1+q_1\dot{q}_2+q_0\dot{q}_3-q_3\dot{q}_0\end{Bmatrix} \tag{5}$$

And

$$\dot{\boldsymbol{\omega}}=\begin{Bmatrix}\dot{\omega}_x\\ \dot{\omega}_y\\ \dot{\omega}_z\end{Bmatrix}=2\begin{Bmatrix}-q_3\ddot{q}_2+q_2\ddot{q}_3+q_0\ddot{q}_1-q_1\ddot{q}_0\\ -q_1\ddot{q}_3+q_3\ddot{q}_1+q_0\ddot{q}_2-q_2\ddot{q}_0\\ -q_2\ddot{q}_1+q_1\ddot{q}_2+q_0\ddot{q}_3-q_3\ddot{q}_0\end{Bmatrix} \tag{6}$$

An additional advantage of this quaternion approach is that to be used with central differences, no first-derivative products appear, which happens when using Euler’s angles. This advantage is relative, because in the second term of Euler's equations, this problem still appears. In the other hand the disadvantage is to have to work with four variables and the non-linear constraint:

$$\sum_{i=0}^{3} q_i^2=1 \tag{7}$$

2.2. Global coordinates of a point starting from local ones

The idea is to get the global coordinates of a point c of the element p when knowing the local coordinates.

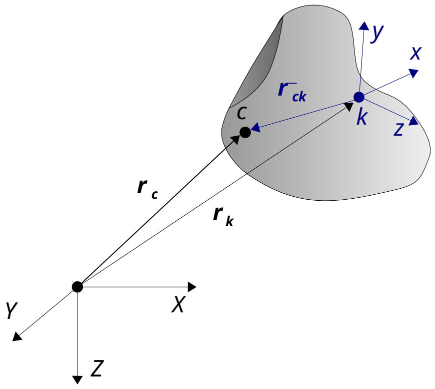

Figure 1: Point c in global coordinates.

If the overline means that it is expressed in the local body reference frame. The local coordinates of point c, referred to the local system of element k are:

$$\bar{\boldsymbol{r}}_{ck}=\{\bar{x}_{ck}\quad \bar{y}_{ck}\quad \bar{z}_{ck}\}^T \tag{8}$$

And $\boldsymbol{r}_{\boldsymbol{k}}$ the global coordinates of the centre of mass (CoM) of element k:

$$\boldsymbol{r}_{\boldsymbol{k}}=\{x_k \quad y_k \quad z_k\}^T \tag{9}$$

The position of the point c in the global system is:

$$\boldsymbol{r}_{\boldsymbol{c}}=\boldsymbol{r}_{\boldsymbol{k}}+R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}} \tag{10}$$

Let it be $\boldsymbol{q}_{\boldsymbol{k}}$ the quaternion that defines the orientation of element k and $R(\boldsymbol{q}_k)$ its rotation matrix. If one wants to carry out a linear approximation, it is necessary to develop the expression (10) under Taylor series. This is necessary to impose constraints in the like of the spherical joint or similar. If $\boldsymbol{q}_k^0$ is the quaternion in the position where the series development is carried out:

$$\boldsymbol{r}_{\boldsymbol{c}}=R(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}}-S_p(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}},\bar{\boldsymbol{r}}_{\boldsymbol{ck}})\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}}+\boldsymbol{r}_{\boldsymbol{k}}+S_p(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}},\bar{\boldsymbol{r}}_{\boldsymbol{ck}})\boldsymbol{q}_{\boldsymbol{k}} \tag{11}$$

Where $S_p(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}},\bar{\boldsymbol{r}}_{\boldsymbol{ck}})$ is a matrix resulting from the linearization, and all the terms are known:

$$S_p(\boldsymbol{q}_k^0,\bar{\boldsymbol{r}}_{ck})=$$
$$\begin{bmatrix} (2q_{k0}^0\bar{x}_{ck}-2q_{k3}^0\bar{y}_{ck}+2q_{k2}^0\bar{z}_{ck}) & (2q_{k1}^0\bar{x}_{ck}+2q_{k2}^0\bar{y}_{ck}+2q_{k3}^0\bar{z}_{ck}) & (-2q_{k2}^0\bar{x}_{ck}+2q_{k1}^0\bar{y}_{ck}+2q_{k0}^0\bar{z}_{ck}) & (-2q_{k3}^0\bar{x}_{ck}-2q_{k0}^0\bar{y}_{ck}+2q_{k1}^0\bar{z}_{ck}) \\ (2q_{k3}^0\bar{x}_{ck}+2q_{k0}^0\bar{y}_{ck}-2q_{k1}^0\bar{z}_{ck}) & (2q_{k2}^0\bar{x}_{ck}-2q_{k1}^0\bar{y}_{ck}-2q_{k0}^0\bar{z}_{ck}) & (2q_{k1}^0\bar{x}_{ck}+2q_{k2}^0\bar{y}_{ck}+2q_{k3}^0\bar{z}_{ck}) & (2q_{k0}^0\bar{x}_{ck}-2q_{k3}^0\bar{y}_{ck}+2q_{k2}^0\bar{z}_{ck}) \\ (-2q_{k2}^0\bar{x}_{ck}+2q_{k1}^0\bar{y}_{ck}+2q_{k0}^0\bar{z}_{ck}) & (2q_{k3}^0\bar{x}_{ck}+2q_{k0}^0\bar{y}_{ck}-2q_{k1}^0\bar{z}_{ck}) & (-2q_{k0}^0\bar{x}_{ck}+2q_{k3}^0\bar{y}_{ck}-2q_{k2}^0\bar{z}_{ck}) & (2q_{k1}^0\bar{x}_{ck}+2q_{k2}^0\bar{y}_{ck}+2q_{k3}^0\bar{z}_{ck}) \end{bmatrix}$$

Equation (11) allows to write in a compact way:

$$\boldsymbol{r}_{\boldsymbol{c}}=R(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}}^{\boldsymbol{L}}-S_v(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0},}\bar{\boldsymbol{r}}_{\boldsymbol{ck}})\boldsymbol{x}_k^0+S_s(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0},}\bar{\boldsymbol{r}}_{\boldsymbol{ck}})\boldsymbol{x}_k \tag{12}$$

Naming:

$$S_s(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0},}\bar{\boldsymbol{r}}_{\boldsymbol{ck}})=[I_{3x3} \quad S_p(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}},\bar{\boldsymbol{r}}_{\boldsymbol{ck}})] \tag{13}$$

$$S_v(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}},\bar{\boldsymbol{r}}_{\boldsymbol{ck}})=[0_{3x3} \quad S_p(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}},\bar{\boldsymbol{r}}_{\boldsymbol{ck}})] \tag{14}$$

$$\boldsymbol{x}_k=\{\boldsymbol{r}_k \quad \boldsymbol{q}_k\}^T \tag{15}$$

2.3. Global coordinates of a vector from the local ones

Let it be $\bar{\boldsymbol{v}}_{\boldsymbol{k}}$ the local coordinates of a vector belonging to a solid k:

$$\bar{\boldsymbol{v}}_{\boldsymbol{k}}=\{\bar{v}_{kx} \quad \bar{v}_{ky} \quad \bar{v}_{kz}\}^T \tag{16}$$

It can be expressed in global coordinates as follows:

$$\boldsymbol{v}_{\boldsymbol{k}}=R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{v}}_{\boldsymbol{k}} \tag{17}$$

By linearizing equation (17) through the Taylor series, one can obtain an expression similar to that of equation (11):

$$\boldsymbol{v}_{\boldsymbol{k}}=R(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}})\bar{\boldsymbol{v}}_{\boldsymbol{k}}-S_{p}(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}},\bar{\boldsymbol{v}}_{\boldsymbol{k}})\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}}+S_{p}(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0},}\bar{\boldsymbol{v}}_{\boldsymbol{k}})\boldsymbol{q}_{\boldsymbol{k}} \qquad (18)$$

If the expression is compact, it can be written as:

$$\boldsymbol{v}_{\boldsymbol{k}}=R(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}})\bar{\boldsymbol{v}}_{\boldsymbol{k}}-S_{v}(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0},}\bar{\boldsymbol{v}}_{\boldsymbol{k}})\boldsymbol{x}_{k}^{0}+S_{v}(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}},\bar{\boldsymbol{v}}_{\boldsymbol{k}})\boldsymbol{x}_{k} \qquad (19)$$

With:

$$S_{v}(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}},\bar{\boldsymbol{v}}_{\boldsymbol{k}})=\left[0_{3x3} \quad S_{p}(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{0}},\bar{\boldsymbol{v}}_{\boldsymbol{k}})\right] \qquad (20)$$

3. Approach of the equations

Considering that they are rigid bodies, each element can be defined by the position of one point in the space and the orientation of a local reference system attached to the body. Additionally, if the local reference system centre coincides with the centre of gravity, and its axis are in the inertia main directions, the inertia tensor will become a diagonal matrix.

$$I_{g}=\begin{bmatrix} I_{x} & 0 & 0 \\ 0 & I_{y} & 0 \\ 0 & 0 & I_{z} \end{bmatrix} \qquad (21)$$

Under these conditions and applying the principle of the kinetic moment, one can obtain Euler's equations:

$$\boldsymbol{T}_{\boldsymbol{e}}=I_{q}\dot{\boldsymbol{\omega}}+\boldsymbol{\omega}\times I_{q}\boldsymbol{\omega} \qquad (22)$$

Where $\boldsymbol{T}_{e}$ is the vector of external torques, and had already defined $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$ in equations (5) and (6), respectively. For a solid k this equations is:

$$\begin{Bmatrix} T_{ekx} \\ T_{eky} \\ T_{ekz} \end{Bmatrix}=\begin{bmatrix} I_{kx} & 0 & 0 \\ 0 & I_{ky} & 0 \\ 0 & 0 & I_{kz} \end{bmatrix}\begin{Bmatrix} \dot{\omega}_{kx} \\ \dot{\omega}_{ky} \\ \dot{\omega}_{kz} \end{Bmatrix}+\begin{bmatrix} 0 & \omega_{kz}I_{kz} & -\omega_{ky}I_{ky} \\ -\omega_{kz}I_{kz} & 0 & \omega_{kx}I_{kx} \\ \omega_{ky}I_{ky} & -\omega_{kx}I_{kx} & 0 \end{bmatrix}\begin{Bmatrix} \omega_{kx} \\ \omega_{ky} \\ \omega_{kz} \end{Bmatrix} \qquad (23)$$

However, one problem with this equation is that the angular velocity cannot be approached as the derivative of the rotation angles with respect to x, y or z. Therefore, it cannot be integrated directly in this way. To do so, it is necessary to set one expression out of the angular velocity under function of some parameters which define the orientation; these can be Euler's angles, quaternions, or even directly the elements of the rotation matrix.

According to Euler's equation, a differential problem with nonlinear equations must be addressed. In optimization, among all of the linearization methods, exact Newton method will be apply. As it is known, it is a quadratic converge method and provides higher level of accurate than quasi Newton method. This is because the exact derivative is used to find the best approximation at each instant of time. The algorithm is obtained from the truncated Taylor series development for the term of grade 2.

$$f(\boldsymbol{x})\simeq f(\boldsymbol{x}_{0})+f'(\boldsymbol{x}_{0})(\boldsymbol{x}-\boldsymbol{x}_{0})+\frac{f''(\xi)}{2}(\xi)^{2} \qquad (24)$$

Returning to Euler's equations of dynamics (equation (22)), expressed in a local system attached to the solid k:

$$\begin{Bmatrix} \bar{T}_{ekx} \\ \bar{T}_{eky} \\ \bar{T}_{ekz} \end{Bmatrix} = \begin{Bmatrix} I_{kx}\dot{\bar{\omega}}_{kx} \\ I_{ky}\dot{\bar{\omega}}_{ky} \\ I_{kz}\dot{\bar{\omega}}_{kz} \end{Bmatrix} + \begin{Bmatrix} \bar{\omega}_{ky}\bar{\omega}_{kz}\left(I_{kz}-I_{ky}\right) \\ \bar{\omega}_{kx}\bar{\omega}_{kz}\left(I_{kx}-I_{kz}\right) \\ \bar{\omega}_{kx}\bar{\omega}_{ky}\left(I_{ky}-I_{kx}\right) \end{Bmatrix} \tag{25}$$

If the expressions for the acceleration is added to equation (25):

$$\begin{Bmatrix} \bar{T}_{ekx} \\ \bar{T}_{eky} \\ \bar{T}_{ekz} \end{Bmatrix} = \begin{Bmatrix} 2I_{kx}\left(-\bar{q}_{k3}\ddot{\bar{q}}_{k2}+\bar{q}_{k2}\ddot{\bar{q}}_{k3}+\bar{q}_{k0}\ddot{\bar{q}}_{k1}-\bar{q}_{k1}\ddot{\bar{q}}_{k0}\right) \\ 2I_{ky}\left(-\bar{q}_{k1}\ddot{\bar{q}}_{k3}+\bar{q}_{k3}\ddot{\bar{q}}_{k1}+\bar{q}_{k0}\ddot{\bar{q}}_{k2}-\bar{q}_{k2}\ddot{\bar{q}}_{k0}\right) \\ 2I_{kz}\left(-\bar{q}_{k2}\ddot{\bar{q}}_{k1}+\bar{q}_{k1}\ddot{\bar{q}}_{k2}+\bar{q}_{k0}\ddot{\bar{q}}_{k3}-\bar{q}_{k3}\ddot{\bar{q}}_{k0}\right) \end{Bmatrix} + \begin{Bmatrix} \bar{\omega}_{ky}\bar{\omega}_{kz}\left(I_{kz}-I_{ky}\right) \\ \bar{\omega}_{kx}\bar{\omega}_{kz}\left(I_{kx}-I_{kz}\right) \\ \bar{\omega}_{kx}\bar{\omega}_{ky}\left(I_{ky}-I_{kx}\right) \end{Bmatrix} \tag{26}$$

The problem with this expression are the cross products of $\bar{\omega}_j\bar{\omega}_i$, which would result in a representation of cross products in the shape $\dot{\bar{q}}_j\dot{\bar{q}}_i$. To solve this, a development in Taylor series of the cross products is carried out:

$$\bar{\omega}_y\bar{\omega}_z \simeq \bar{\omega}_y^0\bar{\omega}_z^0+\bar{\omega}_y^0\left(\bar{\omega}_z-\bar{\omega}_z^0\right)+\bar{\omega}_z^0\left(\bar{\omega}_y-\bar{\omega}_y^0\right)=-\bar{\omega}_y^0\bar{\omega}_z^0+\bar{\omega}_y^0\bar{\omega}_z+\bar{\omega}_z^0\bar{\omega}_y \tag{27}$$

Due to the fact that central differences are going to be used, the only variation that can happen in the angular velocity is owing to the derivatives of the quaternions. Thus, the expression (27) can be written as:

$$\begin{gathered} \begin{Bmatrix} \bar{T}_{ekx} \\ \bar{T}_{eky} \\ \bar{T}_{ekz} \end{Bmatrix} + \begin{Bmatrix} \bar{\omega}_{ky}^0\bar{\omega}_{kz}^0\left(I_{kz}-I_{ky}\right) \\ \bar{\omega}_{kz}^0\bar{\omega}_{kx}^0\left(I_{kx}-I_{kz}\right) \\ \bar{\omega}_{kx}^0\bar{\omega}_{ky}^0\left(I_{ky}-I_{kx}\right) \end{Bmatrix} = \\ \begin{Bmatrix} 2I_{kx}\left(-\bar{q}_{k3}\ddot{\bar{q}}_{k2}+\bar{q}_{k2}\ddot{\bar{q}}_{k3}+\bar{q}_{k0}\ddot{\bar{q}}_{k1}-\bar{q}_{k1}\ddot{\bar{q}}_{k0}\right) \\ 2I_{ky}\left(-\bar{q}_{k1}\ddot{\bar{q}}_{k3}+\bar{q}_{k3}\ddot{\bar{q}}_{k1}+\bar{q}_{k0}\ddot{\bar{q}}_{k2}-\bar{q}_{k2}\ddot{\bar{q}}_{k0}\right) \\ 2I_{kz}\left(-\bar{q}_{k2}\ddot{\bar{q}}_{k1}+\bar{q}_{k1}\ddot{\bar{q}}_{k2}+\bar{q}_{k0}\ddot{\bar{q}}_{k3}-\bar{q}_{k3}\ddot{\bar{q}}_{k0}\right) \end{Bmatrix} + \begin{Bmatrix} \bar{\omega}_{ky}^0\bar{\omega}_{kz}\left(I_{kz}-I_{ky}\right) \\ \bar{\omega}_{kz}^0\bar{\omega}_{kx}\left(I_{kx}-I_{kz}\right) \\ \bar{\omega}_{kx}^0\bar{\omega}_{ky}\left(I_{ky}-I_{kx}\right) \end{Bmatrix} + \begin{Bmatrix} \bar{\omega}_{ky}\bar{\omega}_{kz}^0\left(I_{kz}-I_{ky}\right) \\ \bar{\omega}_{kz}\bar{\omega}_{kx}^0\left(I_{kx}-I_{kz}\right) \\ \bar{\omega}_{kx}\bar{\omega}_{ky}^0\left(I_{ky}-I_{kx}\right) \end{Bmatrix} \end{gathered} \tag{28}$$

Adding the expression of the angular velocity in the equation (28), one can write the next by expressing the quaternion in local coordinates:

$$\bar{\boldsymbol{T}}_{\boldsymbol{ek}}+\bar{\boldsymbol{T}}_{\boldsymbol{CRk}}\left(\bar{\boldsymbol{q}}_{\boldsymbol{k}},\dot{\bar{\boldsymbol{q}}}_k^0\right)=\bar{M}_{Rk}\left(\boldsymbol{q}_{\boldsymbol{k}}\right)\ddot{\bar{\boldsymbol{q}}}_{\boldsymbol{k}}+\bar{C}_{Rk}\left(\bar{\boldsymbol{q}}_{\boldsymbol{k}},\dot{\bar{\boldsymbol{q}}}_k^0\right)\dot{\bar{\boldsymbol{q}}}_{\boldsymbol{k}} \tag{29}$$

It is worth pointing out the Euler's Rotation Theorem, which states that a rigid body or coordinate frame can be brought from an arbitrary initial orientation to an arbitrary final orientation by a single angle rigid rotation through a principal angle θ about the principal axis $\boldsymbol{e}=\begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}^T$; the principal axis fixed in both the initial and final orientation. Thus, taking into account the Euler's Theorem it can be noted that $\bar{\boldsymbol{q}}=\boldsymbol{q}$ since as it is known quaternion is defined as:

$$\boldsymbol{q}=\begin{Bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{Bmatrix}=\begin{Bmatrix} \cos\left(\dfrac{\theta}{2}\right) \\ e_1\sin\left(\dfrac{\theta}{2}\right) \\ e_2\sin\left(\dfrac{\theta}{2}\right) \\ e_3\sin\left(\dfrac{\theta}{2}\right) \end{Bmatrix} \tag{30}$$

Thus, the equation (29) can be written:

$$\bar{\boldsymbol{T}}_{\boldsymbol{ek}}+\bar{\boldsymbol{T}}_{\boldsymbol{CRk}}(\boldsymbol{q_k},\dot{\bar{\boldsymbol{q}}}_k^0)=\bar{M}_{Rk}(\boldsymbol{q_k})\ddot{\bar{\boldsymbol{q}}}_{\boldsymbol{k}}+\bar{C}_{Rk}(\boldsymbol{q_k},\dot{\bar{\boldsymbol{q}}}_k^0)\dot{\bar{\boldsymbol{q}}}_{\boldsymbol{k}} \tag{31}$$

With:

$$\bar{M}_{Rk}(\boldsymbol{q_k})=\begin{bmatrix} -2I_{kx}q_{k1} & 2I_{kx}q_{k0} & -2I_{kx}q_{k3} & 2I_{kx}q_{k2} \\ -2I_{ky}q_{k2} & 2I_{ky}q_{k3} & 2I_{ky}q_{k0} & -2I_{ky}q_{k1} \\ -2I_{kz}q_{k3} & -2I_{kz}q_{k2} & 2I_{kz}q_{k1} & 2I_{kz}q_{k0} \end{bmatrix} \tag{32}$$

$$\bar{C}_{Rk}(\boldsymbol{q_k},\dot{\bar{\boldsymbol{q}}}_k^0)=$$
$$\begin{bmatrix} -2(\bar{\omega}_{ky}^0q_{k3}+\bar{\omega}_{kz}^0q_{k2})(I_{kz}-I_{ky}) & 2(\bar{\omega}_{kz}^0q_{k3}-\bar{\omega}_{ky}^0q_{k2})(I_{kz}-I_{ky}) & 2(\bar{\omega}_{kz}^0q_{k0}+\bar{\omega}_{ky}^0q_{k1})(I_{kz}-I_{ky}) & 2(\bar{\omega}_{ky}^0q_{k0}-\bar{\omega}_{kz}^0q_{k1})(I_{kz}-I_{ky}) \\ -2(\bar{\omega}_{kx}^0q_{k3}+\bar{\omega}_{kz}^0q_{k1})(I_{kx}-I_{kz}) & 2(\bar{\omega}_{kz}^0q_{k0}-\bar{\omega}_{kx}^0q_{k2})(I_{kx}-I_{kz}) & 2(\bar{\omega}_{kx}^0q_{k1}-\bar{\omega}_{kz}^0q_{k3})(I_{kx}-I_{kz}) & 2(\bar{\omega}_{kx}^0q_{k0}+\bar{\omega}_{kz}^0q_{k2})(I_{kx}-I_{kz}) \\ -2(\bar{\omega}_{kx}^0q_{k2}+\bar{\omega}_{ky}^0q_{k1})(I_{ky}-I_{kx}) & 2(\bar{\omega}_{ky}^0q_{k0}+\bar{\omega}_{kx}^0q_{k3})(I_{ky}-I_{kx}) & 2(\bar{\omega}_{kx}^0q_{k0}-\bar{\omega}_{ky}^0q_{k3})(I_{ky}-I_{kx}) & 2(\bar{\omega}_{ky}^0q_{k2}-\bar{\omega}_{kx}^0q_{k1})(I_{ky}-I_{kx}) \end{bmatrix} \tag{33}$$

$$\bar{\boldsymbol{T}}_{CRk}(\boldsymbol{q}_k,\dot{\bar{\boldsymbol{q}}}_k^0)=\begin{Bmatrix} \bar{\omega}_{ky}^0\bar{\omega}_{kz}^0(I_{kz}-I_{ky}) \\ \bar{\omega}_{kz}^0\bar{\omega}_{kx}^0(I_{kx}-I_{kz}) \\ \bar{\omega}_{kx}^0\bar{\omega}_{ky}^0(I_{ky}-I_{kx}) \end{Bmatrix} \tag{34}$$

Where $\bar{M}_{RK}(\boldsymbol{q_k})$ is the mass matrix, $\bar{\boldsymbol{T}}_{\boldsymbol{CRk}}$ is the indirect forces vector resulting from the linearization, $\bar{\boldsymbol{C}}_{\boldsymbol{Rk}}$ gathers the linearized components proportional to the velocity, and $\bar{\boldsymbol{T}}_{\boldsymbol{ek}}$ is the vector of the external torques.

These expressions are only valid when the terms are uttered in the coordinates system of the rigid body along its principal axes. However, it is desired to have them referred to the global system. So, under these conditions the rotation matrix has to be included. If one considers:

$$R_4^T(\boldsymbol{q}_k)=\begin{bmatrix} 1 & \boldsymbol{0}^{\boldsymbol{T}} \\ \boldsymbol{0} & R^T(\boldsymbol{q}_k) \end{bmatrix} \tag{35}$$

One can have:

$$\bar{\boldsymbol{q}}_k=R_4^T(\boldsymbol{q}_k)\boldsymbol{q}_k=\boldsymbol{q}_k \tag{36}$$

$$\dot{\bar{\boldsymbol{q}}}_k=R_4^T(\boldsymbol{q}_k)\dot{\boldsymbol{q}}_k \tag{37}$$

$$\ddot{\bar{\boldsymbol{q}}}_k=R_4^T(\boldsymbol{q}_k)\ddot{\boldsymbol{q}}_k \tag{38}$$

Under these circumstances, it can be suggested:

$$\bar{\boldsymbol{T}}_{ek}+\bar{\boldsymbol{T}}_{CRk}(\boldsymbol{q}_k,\dot{\bar{\boldsymbol{q}}}_k^0)=\bar{M}_{Rk}(\boldsymbol{q}_k)R_4^T(\boldsymbol{q}_k)\ddot{\boldsymbol{q}}_k+\bar{C}_{Rk}(\boldsymbol{q}_k,\dot{\bar{\boldsymbol{q}}}_k^0)R_4^T(\boldsymbol{q}_k)\dot{\boldsymbol{q}}_k \tag{39}$$

Therefore, taking into account:

$$\dot{\bar{\boldsymbol{q}}}_k^0 = \begin{Bmatrix} \dot{\bar{q}}_{k0}^0 \\ \dot{\bar{q}}_{k1}^0 \\ \dot{\bar{q}}_{k2}^0 \\ \dot{\bar{q}}_{k3}^0 \end{Bmatrix} = R_4^T(\boldsymbol{q}_k)\,\dot{\boldsymbol{q}}_k^0 \tag{40}$$

And premultiplying the expression (39) by the rotation matrix, it is obtained:

$$R(\boldsymbol{q}_k)\bar{\boldsymbol{T}}_{ek} + R(\boldsymbol{q}_k)\bar{\boldsymbol{T}}_{CRk}(\boldsymbol{q}_k, \dot{\bar{\boldsymbol{q}}}_k) = R(\boldsymbol{q}_k)\bar{M}_{Rk}(\boldsymbol{q}_k)R_4^T(\boldsymbol{q}_k)\ddot{\boldsymbol{q}}_k + R(\boldsymbol{q}_k)\bar{C}_{Rk}(\boldsymbol{q}_k, \dot{\bar{\boldsymbol{q}}}_k)R_4^T(\boldsymbol{q}_k)\dot{\boldsymbol{q}}_k \tag{41}$$

This allows to rewrite the equilibrium equation as follows:

$$\boldsymbol{T}_{\boldsymbol{ek}}(t) + \boldsymbol{T}_{\boldsymbol{CRk}}(\boldsymbol{q}_{\boldsymbol{k}}(t), \dot{\boldsymbol{q}}_{\boldsymbol{k}}(t)) = M_{Rk}(\boldsymbol{q}_{\boldsymbol{k}}(t))\ddot{\boldsymbol{q}}_{\boldsymbol{k}}(t) + C_{Rk}(\boldsymbol{q}_{\boldsymbol{k}}(t), \dot{\boldsymbol{q}}_{\boldsymbol{k}}(t))\dot{\boldsymbol{q}}_{\boldsymbol{k}}(t) \tag{42}$$

Where:

$$\boldsymbol{T}_{\boldsymbol{ek}}(t) = R(\boldsymbol{q}_{\boldsymbol{k}}(t))\bar{\boldsymbol{T}}_{\boldsymbol{ek}}(t)$$

$$\boldsymbol{T}_{CRk}(\boldsymbol{q}_k(t), \dot{\boldsymbol{q}}_k(t)) = R(\boldsymbol{q}_{\boldsymbol{k}}(t))\bar{\boldsymbol{T}}_{\boldsymbol{CRk}}(\boldsymbol{q}_{\boldsymbol{k}}(t), \dot{\bar{\boldsymbol{q}}}_k(t))$$

$$M_{Rk}(\boldsymbol{q}_{\boldsymbol{k}}(t)) = R(\boldsymbol{q}_{\boldsymbol{k}}(t))\bar{M}_{Rk}(\boldsymbol{q}_{\boldsymbol{k}}(t))R_4^T(\boldsymbol{q}_{\boldsymbol{k}}(t))$$

$$C_{Rk}(\boldsymbol{q}_{\boldsymbol{k}}(t), \dot{\boldsymbol{q}}_{\boldsymbol{k}}(t)) = R(\boldsymbol{q}_{\boldsymbol{k}}(t))\bar{C}_{Rk}(\boldsymbol{q}_{\boldsymbol{k}}(t), \dot{\bar{\boldsymbol{q}}}_k(t))R_4^T(\boldsymbol{q}_{\boldsymbol{k}}(t))$$

The forces and lineal displacements have still to be included by means of Newton equation:

$$\boldsymbol{F}_{ek} = m_k\ddot{\boldsymbol{r}}_k \tag{43}$$

Where $\boldsymbol{F}_{ek}$ is the vector of the external force, m_k is the mass of body k and, $\ddot{\boldsymbol{r}}_k$ is the acceleration of the centre of mass. Thus, the equilibrium equation can be written:

$$M_k(\boldsymbol{x}_{\boldsymbol{k}}(t))\ddot{\boldsymbol{x}}_k(t) + C_k(\boldsymbol{x}_k(t), \dot{\boldsymbol{x}}_k(t))\dot{\boldsymbol{x}}_k(t) = \boldsymbol{F}_{Ek}(t) + \boldsymbol{F}_{CRk}(\boldsymbol{x}_{\boldsymbol{k}}(t), \dot{\boldsymbol{x}}_{\boldsymbol{k}}(t)) \tag{44}$$

Being:

$$M_k = \begin{bmatrix} m_k I_{3x3} & 0_{3x4} \\ 0_{3x3} & M_{Rk} \end{bmatrix} \tag{45}$$

$$C_k = \begin{bmatrix} 0_{3x3} & 0_{3x4} \\ 0_{3x3} & C_{Rk} \end{bmatrix} \tag{46}$$

$$\boldsymbol{F}_{Ek} = \begin{Bmatrix} \boldsymbol{F}_{ek} \\ \boldsymbol{T}_{ek} \end{Bmatrix} \tag{47}$$

$$\boldsymbol{F}_{CRk} = \begin{Bmatrix} 0_{3x1} \\ \boldsymbol{T}_{CRk} \end{Bmatrix} \tag{48}$$

$$\boldsymbol{x}_{\boldsymbol{k}} = \{x_k \quad y_k \quad z_k \quad q_{k0} \quad q_{k1} \quad q_{k2} \quad q_{k3}\}^T \tag{49}$$

In spite of applying an explicit method, the matrix C_k and the vector $\boldsymbol{F}_{CRk}$ depend on the velocity and it will be necessary to solve the equation iteratively.

4. Numeric integration by central differences

As it is known, the method of Central Differences is B-stable and the maximum time increment that guarantees the stability is limited by the value of the highest frequency representative of the system, equation (50). Normally, this should not condition the calculation since in a system formed by rigid bodies the frequency is related to the stiffness of the springs and is not as high as it can be in systems formed by deformable solids. Nonetheless, if frequency of the system were too high, it would be possible to continue using central differences together with some modal truncation techniques but this assumption is not included in this work.

$$\Delta t_{max} = \frac{2}{\omega_{max}} \tag{50}$$

The system of equations (44) defines the behaviour of one body alone, but if one has a system composed by n bodies, the solution vector will be made up of:

$$\boldsymbol{x}(t) = \begin{Bmatrix} \boldsymbol{x_1}(t) \\ \boldsymbol{x_2}(t) \\ .. \\ \boldsymbol{x_n}(t) \end{Bmatrix} \tag{51}$$

Considering the equilibrium equation for several bodies, through expansion and assembly it can be laid out the subsequent set of equations:

$$M(\boldsymbol{x}(t))\ddot{\boldsymbol{x}}(t) + C(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t))\dot{\boldsymbol{x}}(t) = \boldsymbol{F}_E(t) + \boldsymbol{F}_{CR}(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t)) + \boldsymbol{F}_R(\boldsymbol{x}(t)) \tag{52}$$

Where:

M: is the Mass matrix. It is obtained from the assembly of the M_k of each body.

C: is the Damping matrix. It is obtained from the assembly of the C_k of each body and it is composed of the linearized components proportional to the velocity.

$\boldsymbol{F}_E$: is the External forces and torques vector applied on the system.

$\boldsymbol{F}_{CR}$: is the Indirect forces vector coming from linearizations.

$\boldsymbol{F}_R$: is the constraint Reaction forces vector; those generated by the joints between elements.

It is desired to solve equation (52) by means of central differences, as it has been formulated. The terms dependent on $\boldsymbol{x}(t)$ are already calculated, because of being an explicit method. Those dependent on $\dot{\boldsymbol{x}}(t)$, though, are not. Thus, each iteration will have to be solved in an iterative way. As it can be seen in Avilés's book [40], applying the central differences approach:

$$\begin{aligned} &\left(M(\boldsymbol{x}(t))\frac{1}{(\Delta t)^2} + C(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t))\frac{1}{2\Delta t} \right) \boldsymbol{x}(t+\Delta t) = \\ &= \boldsymbol{F}_E(t) + \boldsymbol{F}_{CR}(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t)) + \boldsymbol{F}_R(\boldsymbol{x}(t)) + \left(-M(\boldsymbol{x}(t))\frac{1}{(\Delta t)^2} + C(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t))\frac{1}{2\Delta t} \right) \boldsymbol{x}(t-\Delta t) + \left(M(\boldsymbol{x}(t))\frac{2}{(\Delta t)^2} \right) \boldsymbol{x}(t) \end{aligned} \tag{53}$$

The constraint imposed by the quaternion norm can be directly introduced in the equations system, but it is preferable to include it into the constraints. In a more compact way equation (53) can be written as follows:

$$A(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t))\boldsymbol{x}(t+\Delta t) = E(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t)) + \boldsymbol{F}_R(\boldsymbol{x}(t)) + B(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t))\boldsymbol{x}(t-\Delta t) + D(\boldsymbol{x}(t))\boldsymbol{x}(t) \tag{54}$$

Where:

$$A(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t))=\left(M\,(\boldsymbol{x}(t))\frac{1}{(\Delta t)^2}+C\,(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t))\frac{1}{2\,\Delta t}\right) \quad (55)$$

$$B(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t))=\left(\quad -M\,(\boldsymbol{x}(t))\frac{1}{(\Delta t)^2}+C\,(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t))\frac{1}{2\Delta t}\right) \quad (56)$$

$$D(\boldsymbol{x}(t))=\left(M\,(\boldsymbol{x}(t))\frac{2}{(\Delta t)^2}\right) \quad (57)$$

$$E\,(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t))=\boldsymbol{F}_{\boldsymbol{E}}(t)+\boldsymbol{F}_{\boldsymbol{CR}}(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t)) \quad (58)$$

It must be taken into account that, when introducing relationships between elements such as joints, it is necessary to solve the equilibrium equation for all the elements at the same time. The constraint reaction forces vector is not included because it will be removed by means of the null space of the restriction matrix.

Equation (54) is nonlinear, because in A, B and E appears $\dot{\boldsymbol{x}}\,(t)$ which, in its turn, depends on $\boldsymbol{x}\,(t+\Delta t)$. It is necessary to solve each equilibrium step iteratively. The sequence of the iterative proccess for an instant t will be:

1) Beginning from $\boldsymbol{x}\,(t)$, $\boldsymbol{x}\,(t-\Delta t)$, $\dot{\boldsymbol{x}}\,(t-\Delta t)$ and $\boldsymbol{F}_E(t)$.

2) With $\boldsymbol{x}\,(t)$ one can get $D(\boldsymbol{x}(t))$. This is valid for all calculation in this instant.

3) Find one initial value for the derivative, which can be the one corresponding to the previous iteration:

$$\dot{\boldsymbol{x}}(t)_j=\dot{\boldsymbol{x}}\,(t)_0\simeq\dot{\boldsymbol{x}}\,(t-\Delta t)=\frac{1}{2\Delta t}\left(\boldsymbol{x}(t)-\boldsymbol{x}(t-2\Delta t)\right)$$

4) With this $\dot{\boldsymbol{x}}\,(t)_j$ one can calculate $A\left(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t)_j\right)$, $B\left(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t)_j\right)$ and $E\left(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t)_j\right)$.

5) Apply the method of the null space to introduce the constraints. The term $\boldsymbol{F}_R(\boldsymbol{x}(t))$ is cancelled in this step.

6) Solve the system (54), including the null space matrices of the constraints, to find $\boldsymbol{x}\,(t+\Delta t)_j$.

7) With this $\boldsymbol{x}\,(t+\Delta t)_j$ one calculates again $\dot{\boldsymbol{x}}\,(t)_{j+1}$ through:

$$\dot{\boldsymbol{x}}(t)_{j+1}\simeq\frac{1}{2\Delta t}\left(\boldsymbol{x}(t+\Delta t)_j-\boldsymbol{x}(t-\Delta t)\right)$$

The stopping criterion can be based on the velocity or on the displacement. If $\dot{\boldsymbol{x}}\,(t)_{j+1}-\dot{\boldsymbol{x}}\,(t)_j<\varepsilon$, then one continues with the next time instant; if not, go back to (4)) with the new $\dot{\boldsymbol{x}}\,(t)_{j+1}$. The other option is $\boldsymbol{x}\,(t+\Delta t)_{j+1}-\boldsymbol{x}\,(t+\Delta t)_j<\varepsilon$. The latter requires finding a first approximation that could be $\boldsymbol{x}\,(t)$ which is known.

For the initial step, things are different. As $\dot{\boldsymbol{x}}\,(t_0)$ is known, it not necessary to iterate and it can be done:

1) One has $\boldsymbol{x}\,(t_0)$, $\dot{\boldsymbol{x}}\,(t_0)$ and $\boldsymbol{F}_E(t_0)$.

2) Matrices are expanded and assembled to obtain $M(\boldsymbol{x}(t_0))$, $C(\boldsymbol{x}(t_0), \dot{\boldsymbol{x}}(t_0))$ and $\boldsymbol{F}_{CR}(\boldsymbol{x}(t_0), \dot{\boldsymbol{x}}(t_0))$.

3) $\boldsymbol{x}(t_1)$ is calculated by solving the following equations system:

$$\dot{\boldsymbol{x}}(t_0) \simeq \frac{\boldsymbol{x}(t_1) - \boldsymbol{x}(t_{-1})}{2\Delta t}$$

$$\ddot{\boldsymbol{x}}(t_0) \simeq \frac{\boldsymbol{x}(t_1) - \boldsymbol{x}(t_0) + \boldsymbol{x}(t_{-1})}{\Delta t^2}$$

$$M(\boldsymbol{x}(t_0))\ddot{\boldsymbol{x}}(t_0) + C(\boldsymbol{x}(t_0), \dot{\boldsymbol{x}}(t_0))\dot{\boldsymbol{x}}(t_0) = \boldsymbol{F}_E(t_0) + \boldsymbol{F}_{CR}(\boldsymbol{x}(t_0), \dot{\boldsymbol{x}}(t_0)) + \boldsymbol{F}_R(\boldsymbol{x}(t_0))$$

Hence, it is a system formed by three equations and three unknowns: $\ddot{\boldsymbol{x}}(t_0)$, $\boldsymbol{x}(t_{-1})$ and $\boldsymbol{x}(t_1)$. It is important to keep in mind that to this equations should be added the constraints equations by means of null space.

5. Constraints

5.1. Implementation of the constraints

In multibody systems, the vast majority of constraint equations are nonlinear and to apply the null space method adequately, restrictions must be linearized. This can be realized by means of the development in Taylor series. Thus, the constraint equations can be expressed with the following form:

$$H\boldsymbol{x} = \boldsymbol{b} \tag{59}$$

Where the matrix H is formed by the coefficient of the variables; $\boldsymbol{x}$ are the coordinates of the system; and $\boldsymbol{b}$ is the vector of the independent terms. It is important to point out that matrix H is not the Jacobian matrix.

The system (59) is indefinite which implies that there are infinite solutions and, as it is known, it is possible to express all of them as:

$$\boldsymbol{x} = \boldsymbol{x}_p + N_H \boldsymbol{\alpha} \tag{60}$$

Where $\boldsymbol{x}_p$ is a particular solution, N_H is the null space of the matrix H and, $\boldsymbol{\alpha}$ is the vector of independent coordinates of the system. Within all these infinite solutions obtained, only those fulfilling the equilibrium equation will be part of the solution. Posing equation (60) for a time instant $t+\Delta t$, one reaches to:

$$\boldsymbol{x}(t+\Delta t) = \boldsymbol{x}_p(t+\Delta t) + N_H \boldsymbol{\alpha}(t+\Delta t) \tag{61}$$

If substituted in the equilibrium equation (54), an equation system in the following way is achieved:

$$\begin{gathered} A(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t))\left(\boldsymbol{x}_p(t+\Delta t) + N_H \boldsymbol{\alpha}(t+\Delta t)\right) = \\ = E(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t)) + \boldsymbol{F}_R(\boldsymbol{x}(t)) + B(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t))\boldsymbol{x}(t-\Delta t) + D(\boldsymbol{x}(t))\boldsymbol{x}(t) \end{gathered} \tag{62}$$

It should be noted that $\boldsymbol{x}(t)$ and $\boldsymbol{x}(t-\Delta t)$ are known, $\dot{\boldsymbol{x}}(t)$ however not, since it depends on $\boldsymbol{x}(t+\Delta t)$. For this reason the system must be solved iteratively.

According to it is proposed, the equation (62) can not be solved. The the constraint joints are express in term of quaternions meanwhile the Newton-Euler equations uses rotations. This fact provides that the force generated by the restrictions are not cancelled when the equilibrium is raised. In finite elements, under the supposition of small displacements, this situation can be resolved premultiplying by the transpose of the null space N_H. Nonetheless, this is not possible due to the matrix H does not correspond with the transpose of the directions of the constraint efforts in terms of rotations, as it has been mentioned. As a consequence, in order to find a solution, the constraints have to be posed under function of the angles. Taking into account any restriction equation system defined as $\boldsymbol{f}(\boldsymbol{x})=0$, it can be done:

$$\sum_{i=1}^{j} d\, f_i(\boldsymbol{x})=\sum_{i=1}^{j}\left(\frac{\partial f_i(\boldsymbol{x})}{\partial x}dx+\frac{\partial f_i(\boldsymbol{x})}{\partial y}dy+\frac{\partial f_i(\boldsymbol{x})}{\partial z}dz+\frac{\partial f_i(\boldsymbol{x})}{\partial \theta_x}d\,\theta_x+\frac{\partial f_i(\boldsymbol{x})}{\partial \theta_y}d\,\theta_y+\frac{\partial f_i(\boldsymbol{x})}{\partial \theta_z}d\,\theta_z\right)$$

Generating a system in the following way:

$$\begin{bmatrix}\frac{\partial f_1(\boldsymbol{x})}{\partial x} & \cdots & \frac{\partial f_1(\boldsymbol{x})}{\partial \theta_z}\\ \vdots & \ddots & \vdots\\ \frac{\partial f_j(\boldsymbol{x})}{\partial x} & \cdots & \frac{\partial f_j(\boldsymbol{x})}{\partial \theta_z}\end{bmatrix}\begin{Bmatrix}dx\\ dy\\ dz\\ d\,\theta_x\\ d\,\theta_y\\ d\,\theta_z\end{Bmatrix}=0$$

$$G(\boldsymbol{x}(t))\Delta\boldsymbol{x}=\boldsymbol{0} \tag{63}$$

The matrix $G(\boldsymbol{x}(t))$ represents the constraint efforts formulated in displacement and rotations. Therefore, solutions of this system are the possible infinitesimal movements of the set, approached under the rotations, not the quaternions. The null space of $G(\boldsymbol{x}(t))$, N_G, allows then to cancel the equilibrium equations in which constraint efforts appear:

$$N_G^T\boldsymbol{F}_R(\boldsymbol{x}(t))=0 \tag{64}$$

Consequently, constraints must not only be developed for the coordinates but also for the efforts. Now, equation (62) can be written as:

$$\begin{gathered}N_G^T A(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t))N_H\boldsymbol{\alpha}(t+\Delta t)=\\ N_G^T\left(E(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t))+B(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t))\boldsymbol{x}(t-\Delta t)+D(\boldsymbol{x}(t))\boldsymbol{x}(t)-A(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t))\boldsymbol{x}_{pt}(t+\Delta t)\right)\end{gathered} \tag{65}$$

From here on, a differentiation will be made between coordinates constraints matrix to refer to matrix H and efforts constraints matrix to matrix G. An important detail lays in the fact that if N_H varies each iteration inside the resolution of one $t+\Delta t$, N_G does not. On one hand, the coordinates constraints are applied to $\boldsymbol{x}(t+\Delta t)$, thus, within a same instant of time, matrix H and obviously N_H vary with each new approximation of $\boldsymbol{x}(t+\Delta t)_j$. On that account, the particular solution $\boldsymbol{x}_p(t+\Delta t)$ also has to be recalculated using the following expression, being known $H(\boldsymbol{x}(t+\Delta t)_j)$ and $\boldsymbol{b}(\boldsymbol{x}(t+\Delta t)_j)$:

$$H(\boldsymbol{x}(t+\Delta t)_j)\boldsymbol{x}_p(t+\Delta t)_j=\boldsymbol{b}(\boldsymbol{x}(t+\Delta t)_j) \tag{66}$$

On the other hand, as explicit method is applied, the equilibrium is posed at the instant t and consequently the forces do not vary with each iteration. Therefore, matrix G is valid for all the instant t.

In addition, it is essential to underline that now the efforts originated by the joints are under function of the independent coordinates, so when approaching the equilibrium of the complete system, they will be wiped out. Thus, the system (65) has the dimension of the number of independent coordinates and provides the value of those coordinates, $\boldsymbol{\alpha}(t+\Delta t)$. To obtain all the coordinates of the system, one only has to apply the equation (61).

In the classical null space formulation [17,18], due to only the independent accelerations are unknown, the integration method just provides the independent velocities and positions. Nevertheless, in order to built the system of equations for the null space approach it is necessary to know all the positions and velocities of the model. Consequently, it will must be solved the position and velocity problem at each instant of time. In contrast, applying the methodology presented in this article, since it is using an integration method of order 2, velocity and acceleration is calculates directly with equation (67) and (68), respectively:

$$\dot{\boldsymbol{x}}(t)=\frac{\boldsymbol{x}(t+\Delta t)-\boldsymbol{x}(t-\Delta t)}{2\Delta t} \tag{67}$$

$$\ddot{\boldsymbol{x}}(t)=\frac{\boldsymbol{x}(t+\Delta t)-\boldsymbol{x}(t)+\boldsymbol{x}(t-\Delta t)}{\Delta t^2} \tag{68}$$

Another important issue is how to obtain the constraints reaction forces. Once calculated displacements, velocities and accelerations, the equilibrium equation (54) is completely known except the reaction efforts. Hence, it can be easily modified in the following way:

$$\boldsymbol{F}_R(\boldsymbol{x}(t))=A(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t))\boldsymbol{x}(t+\Delta t)-E(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t))-B(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t))\boldsymbol{x}(t-\Delta t)-D(\boldsymbol{x}(t))\boldsymbol{x}(t) \tag{69}$$

At the initial step, as $\dot{\boldsymbol{x}}(t_0)$ is known, the constraint forces could be calculated directly with the following expression:

$$\begin{gathered}\boldsymbol{F}_R(\boldsymbol{x}(t_0))= \\ M(\boldsymbol{x}(t_0))\left[\frac{2\left(\boldsymbol{x}(t_1)-\Delta t\,\dot{\boldsymbol{x}}(t_0)\right)-\boldsymbol{x}(t_0)}{\Delta t^2}\right]+C(\boldsymbol{x}(t_0),\dot{\boldsymbol{x}}(t_0))\dot{\boldsymbol{x}}(t_0)-\boldsymbol{F}_E(\boldsymbol{x}(t_0))-\boldsymbol{F}_{CR}(\boldsymbol{x}(t_0),\dot{\boldsymbol{x}}(t_0))\end{gathered} \tag{70}$$

Although apparently, equations (69) and (70) seem costly, it is simply multiplication of known matrix and vector with a cost of n^2. Remember that this constraint forces are the equivalent forces of the restrictions at the centre of mass.

5.2. Spherical joint

The spherical joint is obtained by imposing the position of a point of the element, see figure (2). For further study of the spherical joint for flexible bodies it is advisable to consult the work of Tian [41]. Let it be c one point common from element k and element m.

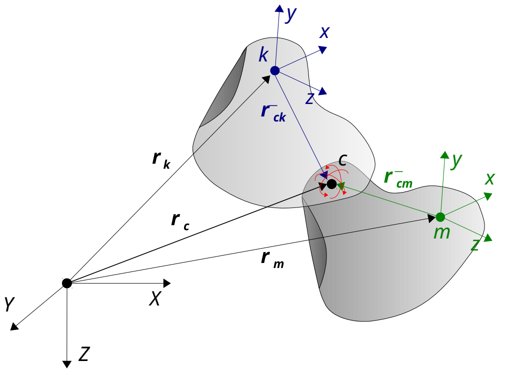

Figure 2: Spherical joint between two bodies.

Then, the constraint of spherical joint between both bodies establishes that:

$$f_s(\boldsymbol{x}_k,\boldsymbol{x}_m)=\boldsymbol{r}_k+R(\boldsymbol{q}_k)\bar{\boldsymbol{r}}_{ck}-\boldsymbol{r}_m-R(\boldsymbol{q}_m)\bar{\boldsymbol{r}}_{mc}=0 \tag{71}$$

To obtain the spherical joint equations for coordinates, if expression (71) is linearized, like in equation (11), one can get:

$$S_s(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\bar{\boldsymbol{r}}_{\boldsymbol{ck}})\,\boldsymbol{x}_{\boldsymbol{p}}^{\boldsymbol{j+1}}-S_s(\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}},\bar{\boldsymbol{r}}_{\boldsymbol{cm}})\,\boldsymbol{x}_{\boldsymbol{m}}^{\boldsymbol{j+1}}=-R(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}})\,\boldsymbol{r}_{\boldsymbol{ck}}^{\boldsymbol{L}}+S_p(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\bar{\boldsymbol{r}}_{\boldsymbol{ck}})\,\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}}+R(\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}})\,\bar{\boldsymbol{r}}_{\boldsymbol{cm}}-S_p(\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}},\bar{\boldsymbol{r}}_{\boldsymbol{cm}})\,\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}} \tag{72}$$

Expanding equations for a system of n elements and disposing in matrix form:

$$\begin{bmatrix} 0_{3x7} & \cdots & S_s(\boldsymbol{q}_k^j,\bar{\boldsymbol{r}}_{ck}) & 0_{3x7} & \cdots & -S_s(\boldsymbol{q}_m^j,\bar{\boldsymbol{r}}_{cm}) & 0_{3x7} & \cdots \end{bmatrix}_{3x\,7n} \begin{Bmatrix} \boldsymbol{x}_1^{j+1} \\ \vdots \\ \boldsymbol{x}_k^{j+1} \\ \vdots \\ \boldsymbol{x}_m^{j+1} \\ \vdots \\ \boldsymbol{x}_n^{j+1} \end{Bmatrix}_{7nx\,1} = \boldsymbol{b}_s(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}})$$

$$H_s(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}})\,\boldsymbol{x}^{\boldsymbol{j+1}}=\boldsymbol{b}_{\boldsymbol{s}}(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}}) \tag{73}$$

With:

$$\boldsymbol{b}_s(\boldsymbol{q}_k^j,\boldsymbol{q}_m^j)=-R(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}})\,\bar{\boldsymbol{r}}_{\boldsymbol{ck}}+S_p(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\bar{\boldsymbol{r}}_{\boldsymbol{ck}})\,\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}}+R(\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}})\,\bar{\boldsymbol{r}}_{\boldsymbol{cm}}-S_p(\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}},\bar{\boldsymbol{r}}_{\boldsymbol{cm}})\,\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}}$$

In the other hand, for the spherical joint equations for displacement, if equation (10) is particularized for an instant t, the position in global coordinates of a point c of element k is achieved. If a displacement and some infinitesimal rotations happen, the differential displacement will be as follows:

$$d\,\boldsymbol{r}_{\boldsymbol{c}}=\frac{d\,\boldsymbol{r}_{\boldsymbol{c}}}{dx_k}dx_k+\frac{d\,\boldsymbol{r}_{\boldsymbol{c}}}{dy_k}dy_k+\frac{d\,\boldsymbol{r}_{\boldsymbol{c}}}{dz_k}dz_k+\frac{d\,\boldsymbol{r}_{\boldsymbol{c}}}{d\,\theta_{xk}}d\,\theta_{xk}+\frac{d\,\boldsymbol{r}_{\boldsymbol{c}}}{d\,\theta_{yk}}d\,\theta_{yk}+\frac{d\,\boldsymbol{r}_{\boldsymbol{c}}}{d\,\theta_{zk}}d\,\theta_{zk} \tag{74}$$

Which can be expressed for the case of displacements as:

$$\frac{d\,\boldsymbol{r}_{\boldsymbol{c}}}{dx_k}=\begin{Bmatrix}1\\0\\0\end{Bmatrix};\frac{d\,\boldsymbol{r}_{\boldsymbol{c}}}{dy_k}=\begin{Bmatrix}0\\1\\0\end{Bmatrix};\frac{d\,\boldsymbol{r}_{\boldsymbol{c}}}{dz_k}=\begin{Bmatrix}0\\0\\1\end{Bmatrix}$$

Furthermore, for the case of the rotations, these will modify the term $R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}}$:

$$\frac{d\,\boldsymbol{r}_{\boldsymbol{c}}}{d\,\theta_{xk}}=\begin{Bmatrix}0\\-(R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}})_z\\(R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}})_y\end{Bmatrix};\frac{d\,\boldsymbol{r}_{\boldsymbol{c}}}{d\,\theta_{yk}}=\begin{Bmatrix}(R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}})_z\\0\\-(R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}})_x\end{Bmatrix};\frac{d\,\boldsymbol{r}_{\boldsymbol{c}}}{d\,\theta_{zk}}=\begin{Bmatrix}-(R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}})_y\\(R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}})_x\\0\end{Bmatrix}$$

Then it can be written:

$$d\,\boldsymbol{r}_{\boldsymbol{c}}=\begin{bmatrix}1&0&0&0&(R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}})_z&-(R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}})_y\\0&1&0&-(R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}})_z&0&(R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}})_x\\0&0&1&(R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}})_y&-(R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{r}}_{\boldsymbol{ck}})_x&0\end{bmatrix}\begin{Bmatrix}dx_k\\dy_k\\dz_k\\d\,\theta_{xk}\\d\,\theta_{yk}\\d\,\theta_{zk}\end{Bmatrix}$$

$$d\,\boldsymbol{r}_{\boldsymbol{c}}=Q_s(\boldsymbol{q}_{\boldsymbol{k}},\bar{\boldsymbol{r}}_{\boldsymbol{ck}})\,\Delta\,\boldsymbol{x}_{\boldsymbol{k}} \tag{75}$$

Where:

$$R(\boldsymbol{q}_k)\bar{\boldsymbol{r}}_{ck} = \begin{Bmatrix} (R(\boldsymbol{q}_k)\boldsymbol{r}_{ck})_x \\ (R(\boldsymbol{q}_k)\bar{\boldsymbol{r}}_{ck})_y \\ (R(\boldsymbol{q}_k)\bar{\boldsymbol{r}}_{ck})_z \end{Bmatrix}$$

For the case where c is a point with known position:

$$Q_s(\boldsymbol{q}_k, \bar{\boldsymbol{r}}_{ck})\,\Delta\,\boldsymbol{x}_k = \boldsymbol{0} \tag{76}$$

In the other hand, if it is a joint between two elements, then:

$$Q_s(\boldsymbol{q}_p, \bar{\boldsymbol{r}}_{cp})\boldsymbol{\Delta}\,\boldsymbol{x}_p - Q_s(\boldsymbol{q}_m, \bar{\boldsymbol{r}}_{cm})\,\boldsymbol{\Delta}\,\boldsymbol{x}_m = \boldsymbol{0} \tag{77}$$

These constraints have to be expanded to the total model size:

$$\begin{bmatrix} 0_{3x6} & \cdots & Q_s(\boldsymbol{q}_k^j, \bar{\boldsymbol{r}}_{ck}) & 0_{3x6} & \cdots & -Q_s(\boldsymbol{q}_m^j, \bar{\boldsymbol{r}}_{cm}) & 0_{3x6} & \cdots \end{bmatrix}_{3x6n} \begin{Bmatrix} \Delta\,\boldsymbol{x}_1 \\ \vdots \\ \Delta\,\boldsymbol{x}_k \\ \vdots \\ \Delta\,\boldsymbol{x}_m \\ \vdots \\ \Delta\,\boldsymbol{x}_n \end{Bmatrix}_{6nx1} = \boldsymbol{0}$$

$$G_s(\boldsymbol{q}_k, \boldsymbol{q}_m)\boldsymbol{\Delta}\,x = \boldsymbol{0} \tag{78}$$

5.3. Revolute joint

This is a joint of class I, where the movement is limited to a rotation around one axis, defined by its director vector v as in the figure (3).

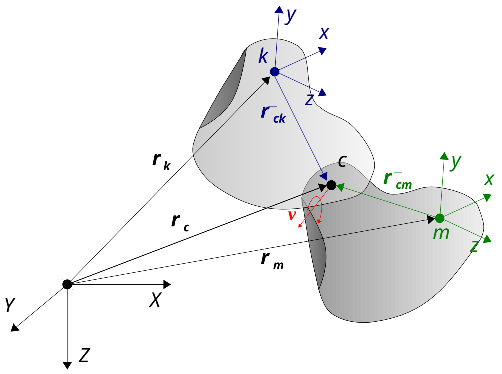

Figure 3: Revolute joint in point c between body k and body m.

Therefore, the rotation joint between two elements k and m is imposed by a common point and a common director vector, being its equations the following:

$$f_s(\boldsymbol{x}_k, \boldsymbol{x}_m) = \boldsymbol{r}_k + R(\boldsymbol{q}_k)\bar{\boldsymbol{r}}_{ck} - \boldsymbol{r}_m - R(\boldsymbol{q}_m)\bar{\boldsymbol{r}}_{mc} = 0$$

$$f_v(\boldsymbol{q}_k, \boldsymbol{q}_m) = R(\boldsymbol{q}_k)\bar{\boldsymbol{v}}_k - R(\boldsymbol{q}_m)\bar{\boldsymbol{v}}_m = 0 \tag{79}$$

These equations are referred to the common point. Next, equation (79) will be developed.

To find the equations for the revolute joint for coordinates, taking into account equation (19), one can write equation (79) for a time instant t_{j+1} as:

$$S_v(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\bar{\boldsymbol{v}}_{\boldsymbol{k}})\,\boldsymbol{x}_{\boldsymbol{k}}^{\boldsymbol{j+1}}-S_v(\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}},\bar{\boldsymbol{v}}_{\boldsymbol{m}})\,\boldsymbol{x}_{\boldsymbol{m}}^{\boldsymbol{j+1}}=-R(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}})\bar{\boldsymbol{v}}_{\boldsymbol{k}}+S_p(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\bar{\boldsymbol{v}}_{\boldsymbol{k}})\boldsymbol{q}_{\boldsymbol{p}}^{\boldsymbol{j}}+R(\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}})\bar{\boldsymbol{v}}_{\boldsymbol{m}}-S_p(\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}},\bar{\boldsymbol{v}}_{\boldsymbol{m}})\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}} \tag{80}$$

By expanding the equations for a system of n elements and disposing in matrix form:

$$\begin{bmatrix} 0_{3x7} & \cdots & S_v(\boldsymbol{q}_k^j,\bar{\boldsymbol{v}}_k) & 0_{3x7n} & \cdots & -S_v(\boldsymbol{q}_m^j,\bar{\boldsymbol{v}}_m) & 0_{3x7n} & \cdots \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_1^{j+1} \\ \vdots \\ \boldsymbol{x}_k^{j+1} \\ \vdots \\ \boldsymbol{x}_m^{j+1} \\ \vdots \\ \boldsymbol{x}_n^{j+1} \end{Bmatrix}_{7nx1} = \boldsymbol{b}_v(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}})$$

$$H_v(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}})\,\boldsymbol{x}^{\boldsymbol{j+1}}=\boldsymbol{b}_{\boldsymbol{v}}(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}}) \tag{81}$$

Where:

$$\boldsymbol{b}_v(\boldsymbol{k}_p^j,\boldsymbol{q}_m^j)=-R(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}})\boldsymbol{v}_{\boldsymbol{k}}^{\boldsymbol{L}}+\bar{S}_s(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\boldsymbol{v}_{\boldsymbol{k}}^{\boldsymbol{L}})\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}}+R(\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}})\boldsymbol{v}_{\boldsymbol{m}}^{\boldsymbol{L}}-\bar{S}_s(\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}},\boldsymbol{v}_{\boldsymbol{m}}^{\boldsymbol{L}})\,\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}}$$

To completely build the revolute joint link, one must include the equations concerning the point. Subsequently, the equations remain as follows:

$$\begin{bmatrix} H_s(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}}) \\ H_v(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}}) \end{bmatrix}_{6x7n} \boldsymbol{x}^{\boldsymbol{j+1}}=\begin{Bmatrix} \boldsymbol{b}_s(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}}) \\ \boldsymbol{b}_{\boldsymbol{v}}(\boldsymbol{q}_{\boldsymbol{k}}^{\boldsymbol{j}},\boldsymbol{q}_{\boldsymbol{m}}^{\boldsymbol{j}}) \end{Bmatrix}_{6x1} \tag{82}$$

In the case of the equations of revolute joint for displacements, returning to equation (17) that defines the position of a vector $\boldsymbol{v}_{\boldsymbol{k}}$ in the global system from its components in a local system $\bar{\boldsymbol{v}}_k$, if it is particularized for an instant t and is subjected to a displacement and an infinitesimal rotation, the differential remains as:

$$d\,\boldsymbol{v}_{\boldsymbol{k}}=\frac{d\,\boldsymbol{v}_{\boldsymbol{k}}}{dx_k}dx_k+\frac{d\,\boldsymbol{v}_{\boldsymbol{k}}}{dy_k}dy_k+\frac{d\,\boldsymbol{v}_{\boldsymbol{k}}}{dz_k}dz_k+\frac{d\,\boldsymbol{v}_{\boldsymbol{k}}}{d\,\theta_{xk}}d\,\theta_{xk}+\frac{d\,\boldsymbol{v}_{\boldsymbol{k}}}{d\,\theta_{yk}}d\,\theta_{yk}+\frac{d\,\boldsymbol{v}_{\boldsymbol{k}}}{d\,\theta_{zk}}d\,\theta_{zk} \tag{83}$$

It can be written:

$$d\,\boldsymbol{v}_{\boldsymbol{k}}=\begin{bmatrix} 0 & 0 & 0 & 0 & (R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{v}}_{\boldsymbol{k}})_z & -(R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{v}}_{\boldsymbol{k}})_y \\ 0 & 0 & 0 & -(R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{v}}_{\boldsymbol{k}})_z & 0 & (R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{v}}_{\boldsymbol{k}})_x \\ 0 & 0 & 0 & (R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{v}}_{\boldsymbol{k}})_y & -(R(\boldsymbol{q}_{\boldsymbol{k}})\bar{\boldsymbol{v}}_{\boldsymbol{k}})_x & 0 \end{bmatrix} \begin{Bmatrix} dx_k \\ dy_k \\ dz_k \\ d\,\theta_{xk} \\ d\,\theta_{yk} \\ d\,\theta_{zk} \end{Bmatrix}$$

$$d\,\boldsymbol{v}=Q_v(\boldsymbol{q}_{\boldsymbol{k}},\bar{\boldsymbol{v}}_{\boldsymbol{k}})\Delta\,\boldsymbol{x}_k \tag{84}$$

Naming:

$$R\,(\boldsymbol{q}_k)\bar{\boldsymbol{v}}_k=\left\{\begin{matrix}(R\,(\boldsymbol{q}_k)\bar{\boldsymbol{v}}_k)_x\\(R\,(\boldsymbol{q}_k)\bar{\boldsymbol{v}}_k)_y\\(R\,(\boldsymbol{q}_k)\bar{\boldsymbol{v}}_k)_z\end{matrix}\right\}$$

In the case where $\boldsymbol{v}_k$ is a known vector:

$$Q_v(\boldsymbol{q_k},\bar{\boldsymbol{v_k}})\,\Delta\,\boldsymbol{x}_k=0 \tag{85}$$

For the case of a vector belonging to two elements k and m, the system would be as follows:

$$Q_v(\boldsymbol{q_k},\bar{\boldsymbol{v}}_k)\Delta\,\boldsymbol{x}_k-Q_v(\boldsymbol{q_m},\bar{\boldsymbol{v}}_m)\Delta\,\boldsymbol{x}_m=\boldsymbol{0} \tag{86}$$

Thus, expanding for the total dimension of the model:

$$\begin{bmatrix}0_{3x6} & \cdots & Q_v(\boldsymbol{q}_k,\bar{\boldsymbol{v}}_k) & 0_{3x6} & \cdots & -Q_v(\boldsymbol{q}_m,\bar{\boldsymbol{v}}_m) & 0_{3x6} & \cdots\end{bmatrix}_{3x6n}\left\{\begin{matrix}\Delta\,\boldsymbol{x}_1\\\vdots\\\Delta\,\boldsymbol{x}_k\\\vdots\\\Delta\,\boldsymbol{x}_m\\\vdots\\\Delta\,\boldsymbol{x}_n\end{matrix}\right\}_{6nx1}=\boldsymbol{0}$$

$$G_v(\boldsymbol{q_k},\boldsymbol{q_m})\Delta\,\boldsymbol{x}=\boldsymbol{0} \tag{87}$$

Finally, including the effort equations of the spherical joint:

$$\begin{bmatrix}G_s(\boldsymbol{q_k},\boldsymbol{q_m})\\G_v(\boldsymbol{q_k},\boldsymbol{q_m})\end{bmatrix}_{6x6n}\Delta\,\boldsymbol{x}=\left\{\begin{matrix}\boldsymbol{0}\\\boldsymbol{0}\end{matrix}\right\} \tag{88}$$

5.4. Constraint imposed by the quaternion norm

This norm could be possible to include in the equilibrium equation but it is more convenient to add as a restriction. The constraint imposed by the quaternion norm presents a non linearity so it will be necessary to pose a linearization in the following way:

Let it be the norm function:

$$f\,(\boldsymbol{q})=q_0^2+q_1^2+q_2^2+q_3^2-1=0 \tag{89}$$

By linearizing the function by means of a Taylor series in a value of $\boldsymbol{q}_{\boldsymbol{j+1}}$ in the nearness of $\boldsymbol{q}_{\boldsymbol{j}}$:

$$f\,(\boldsymbol{q}^{\boldsymbol{j+1}})\simeq-1-q_0^j-q_1^j-q_2^j-q_3^j+2\{q_0^j\quad q_1^j\quad q_2^j\quad q_3^j\}^T\left\{\begin{matrix}q_0^{j+1}\\q_1^{j+1}\\q_2^{j+1}\\q_3^{j+1}\end{matrix}\right\}=0 \tag{90}$$

If one organizes equation (90) for an element k :

$$\{0\quad 0\quad 0\quad 2q_{0k}^j\quad 2q_{1k}^j\quad 2q_{2k}^j\quad 2q_{3k}^j\}^T\,\boldsymbol{x}_k^{j+1}=1+q_{0k}^j+q_{1k}^j+q_{2k}^j+q_{3k}^j$$

In a compact way:

$$S_q\left(\boldsymbol{q_k^j}\right)\boldsymbol{x}_k^{j+1}=\boldsymbol{b}\left(\boldsymbol{q}_k^j\right) \tag{91}$$

Expanding for the complete system of *n* elements:

$$\begin{bmatrix} 0_{1x7} & \cdots & S_q\left(\boldsymbol{q_k^j}\right) & \cdots & 0_{1x7}\end{bmatrix}_{1x7n}\begin{Bmatrix}\boldsymbol{x_1}\\ \vdots \\ \boldsymbol{x_k^{j+1}} \\ \vdots \\ \boldsymbol{x_n}\end{Bmatrix}=\boldsymbol{b}_q\left(\boldsymbol{q_k^j}\right)$$

$$H_q\left(\boldsymbol{q_k^j}\right)\boldsymbol{x^{j+1}}=\boldsymbol{b_q}\left(\boldsymbol{q_k^j}\right) \tag{92}$$

It is worth pointing out that the quaternion norm does not make any force so the restriction matrix for effort is not contemplated.

6. Numeric application

The methodology exposed has been implemented with a preliminary programming in the calculation software GNU Octave 4.4.0 [38]. Three mechanical systems are represented with the aim to compare the responses with different integrators present in the simulation software named MSC Adams 17.2 Student Edition such as GSTIFF, WSTIFF and HHT. The comparison will be centred mainly in the accuracy of the results. Simulations have been run in a computer equipped with Windows 10, CPU Intel Xenon, and 32 Gb of RAM. The examples include a Spinning Top, a planar Double Four-Bar mechanism and a spatial Bricard mechanism. In this context, several drawback have to be addressed by the formulation such as: external perturbations, singular configurations or over-constrained systems.

According to validation of the result, GNU Octave is an interpreted language a different from MSC Adams which use compiled language. The interpreted languages are usually more slower due to the need to translate the programme while it is executing. Even though the problem proposed are adequate for measuring the accuracy of the method, they have few elements. This implies that most of the computational cost is invested in this task. Therefore, although other formulations can be implemented in GNU Octave, the validity of the results could not be assured as far as a calculation time is concerned.

6.1. Measurements

Taking advantage of the fact that both the Double Bar and Bricard mechanisms are benchmark problems, it is easy to obtain results with sufficient accuracy from iftomm-multibody website. These results can be used to establish a total error by computing different variables and therefore measuring the accuracy. In each case, it will be mentioned under what circumstances these references results have been chosen.

The accuracy is a function of the error admitted for each iteration step, the increment of time and the type of integrator. Therefore, accuracy is measured as the maximum error between the solution and a reference value. In general, results are a function of time such as the position, the velocity, the force, etc.; it will be stated which of these variables are part of the solution to be evaluated. If the next is defined:

$x_j(t_i)$ as the solution in the instant t_i of the variable j .

$x_j^{ref}(t_i)$ as the solution of the reference variable j for the instant t_i .

The error for a variable is obtained through equation (93), whereas the accumulated error for n variables during a simulation of m intervals is calculated through equation (94). The value of threshold is introduced to avoid singularities when the reference value tends to zero, and its value has been fixed to 1e-3.

$$e_j(t_i)=\frac{\left|x_j(t_i)-x_j^{ref}(t_i)\right|}{max\left\{\left|x_j^{ref}(t_i)\right|,x_j^{threshold}\right\}} \tag{93}$$

$$e_{Total}=\sqrt{\frac{1}{m}\sum_{i=1}^{m}\frac{1}{n}\sum_{j=1}^{n}\left(e_j(t_i)\right)^2} \tag{94}$$

The variables of each problem taking part in the solution are gathered in table 1.

Table 1:Problems, duration and monitored variables

Problem	Duration	Variables
Double Bar Mechanism	10	Displacement in x, and velocity in x of point 1
Bricard Mechanism	10	Displacement in x, y, z of point 2

In relation to this aspect, there are some issues that are worth clarifying. Table 3 and 4 show the ERROR magnitude, this means the integration tolerance for each iteration step. Therefore, the same quantity has been agreed for all integration methods and obviously, the solutions will be affected for this value. Besides, the integrators of MSC Adams normally uses variable time step. It is possible to impose the maximum time step, however, the minimum is left free for assuring the converge of the results. On this sense, in spite of the variable step, MSC Adams allows obtaining the results in fixed steps and if the solutions are known in those instant of time, it is feasible to calculate the total error. In addition, constraint violation at the position level will be displayed. In tables this value is calculated using root mean square of the norm of the position violation vector. The algorithm of the method is formulated in function of the displacements, hence, the violation of velocity constraints will not be obtained.

6.2. Spinning top

This is a system formed by a single rigid body that starts with a velocity of rotation in the longitudinal axis, and to which a perturbation is applied in the centre of gravity in a given moment. The 3D model consists of a spherical joint in the contact between the spinning top and the ground.

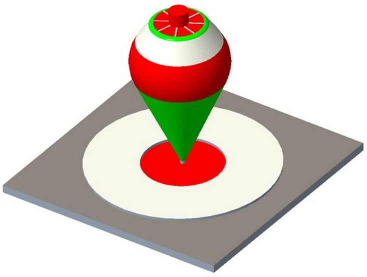

Figure 4: Spinning top and ground

In figure 5, the displacement in axis x of the spinning top’s mass centre is shown when applying the Direct Integration Method with Central Difference (DIMCD) with different integration steps in GNU Octave. It can be observed that a good convergence exists with a calculation step of 0.5 ms even with 1 ms.

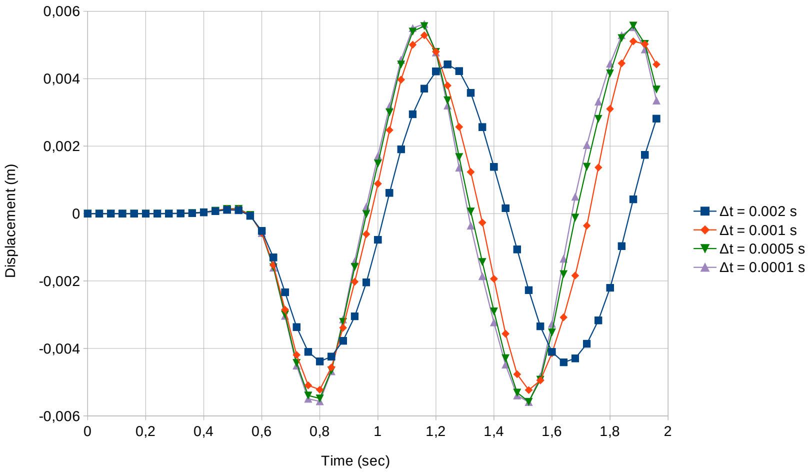

Figure 5: Displacement of the mass centre in direction x of the Spinning Top.

To evaluate the response, the results obtained with DIMCD with a constant step of 0.5 ms are compared to different integrators. Due to the fact that the GSTIFF is a multistep integrator, a maximum step of 0.5 ms and an ERROR of 1e-6 have been established, while the step diminishes to 0.2 ms in several periods of the simulation. In addition, the integrator WSTIFF is initiated with the same criteria and it is observed that the step diminishes to 0.1 ms in the moment of application of the perturbation. In figure can be shown the results of the displacement and velocity in the x axis of the CoM.

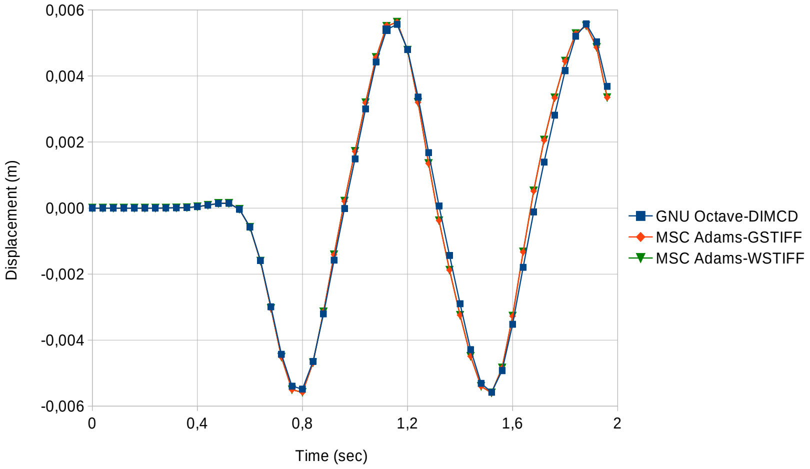

Figure 6: Displacement of the CoM in x axis of the Spinning Top for different integrators.

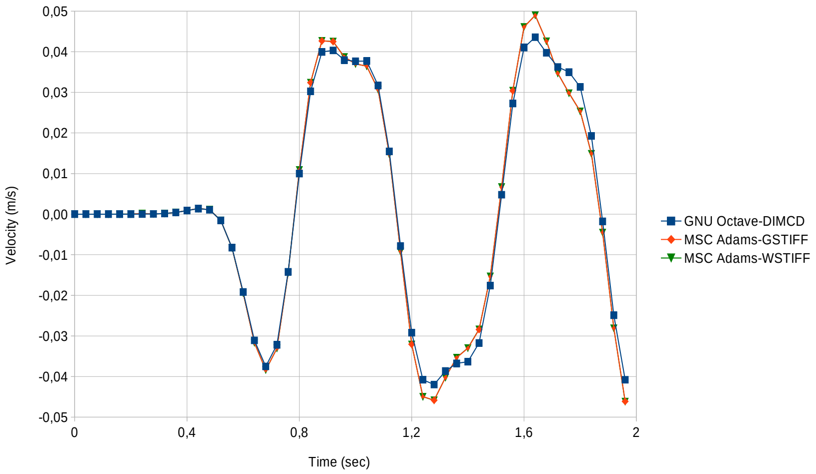

Figure 7: Velocity of the CoM in x axis of the Spinning Top for different integrators.

The discretization of the perturbation is fundamental in the correct modelling of the problem. It is observed that in the integrators GSTIFF and WSTIFF a good precision is achieved when decreasing the step and increasing the order of integration, whereas in the DIMCD method even maintaining a constant step and with an order of integration of 2 quite good results are obtained. This is mainly due to the fact that the exact Newton method is used to calculate the equilibrium at each instant of integration.

Table 2: Comparative of the different integrators

Method	Type of step	Δt (ms)	Order	ERROR	CPU time (s)	Violation of const.
DIMCD	Fixed	0.5	2	1e-6	59	1,59e-16
GSTIFF I3	Variable	0.2-0.5	6	1e-6	3	6,01e-15
WSTIFF I3	Variable	0.1-0.5	6	1e-6	5	2,43e-11

The constraint equations at the position level are achieved in the order of 1e-16, as can be observed in figure 8.

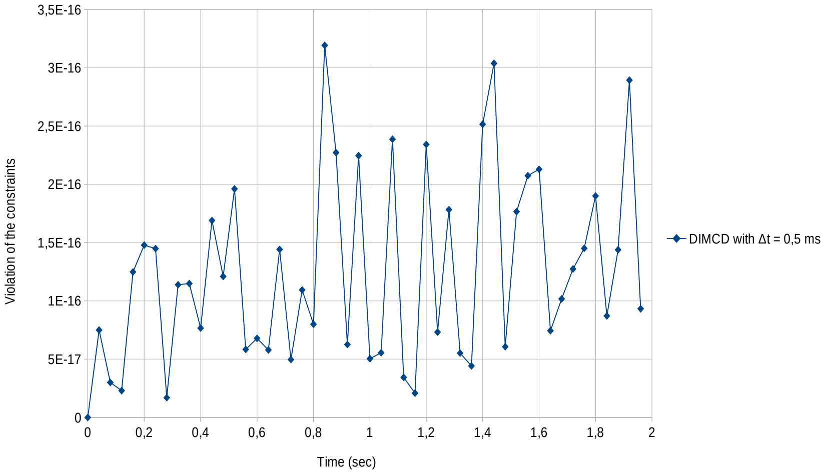

Figure 8:Violation of the constraint position equations of the Spinning Top.

6.3. Double Four-Bar

The particularity is that when the mechanism reaches the horizontal position, the first order number of degrees of freedom of the model increase from 1 to 3, which causes that not a few formulations have problems to overcome this situation. This problem was proposed by Bayo and Avello in [42].

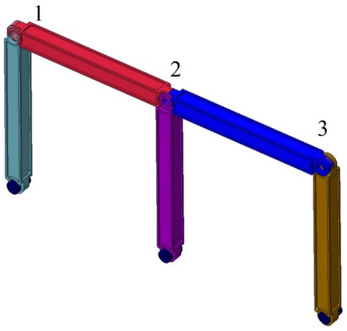

Figure 9: Double Four-Bar Mechanism

This is a problem where only the gravity force intervenes, therefore, the conservation of the mechanical energy of the system can be a good indicator of the quality of the simulation. Simulations where the deviation does not exceed 0,1 J are taken as acceptable. In figure 10 the result of the conservation of the energy is shown using DIMCD for different time steps with GNU Octave. According to the quality criterion, it can be considered that the simulation with time step of 10 ms is adequate. For these conditions, the violations of the equations of restrictions have been computed whose results are shown in figure 11. It can be observed that the positions of least conservation coincide with places where the greatest violations of the constraint equation is committed.

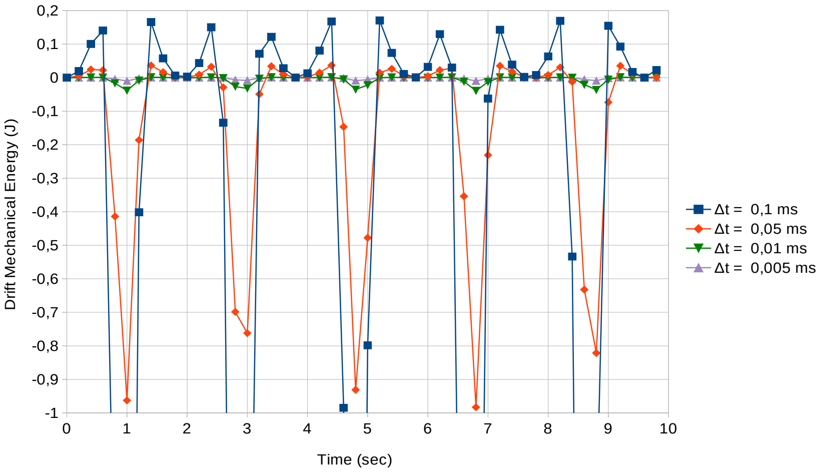

Figure 10: Conservation of the energy for different time step of the Double Four-Bar.

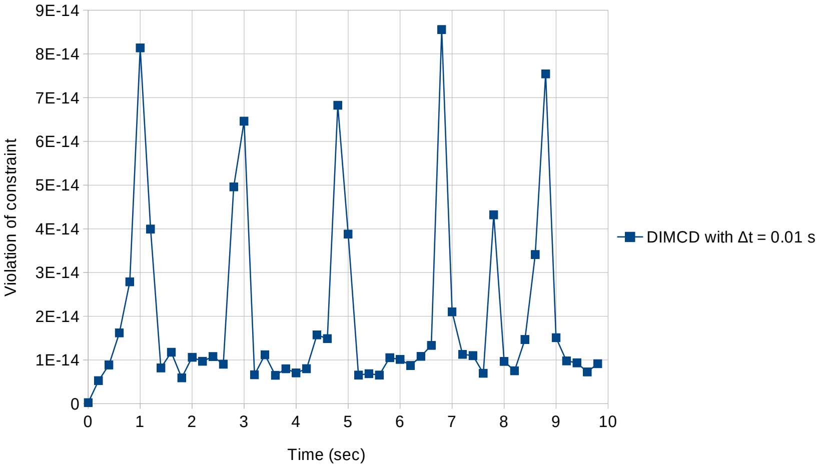

Figure 11: Violation of constraint position equation for Double Four-Bar mechanism.

As it has been commented, it is possible to obtain a solution as a reference. Hence, it is considered as the most accurate solutions those with greatest energy conservation. The reference simulation preserve the mechanical energy of the system up to 3.2e-7 J. These mention results can be used to calculate the total error by means variables for the x displacement and x velocity of the point 1. Fifty numerical results are sampled at the same instant of time of the three different integrator of MSC Adams and DIMCD. The figure 12 and 13 show the result of the simulations and a total error is included in table 3.

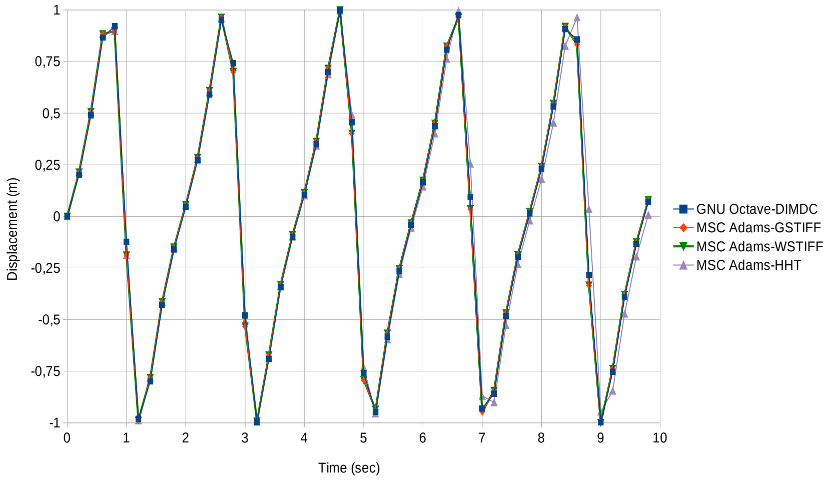

Figure 12: Displacement of the point 1 in x direction for different integrators of the Double Four-Bar.

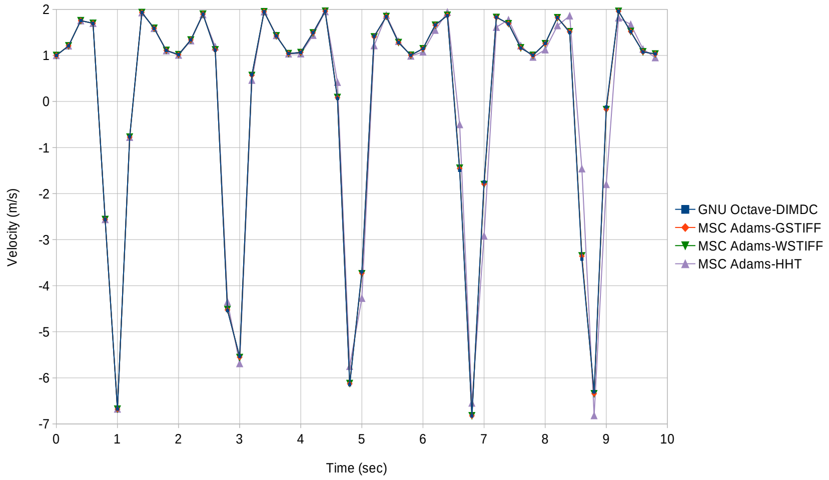

Figure 13: Velocity of point 1 in x direction for different integrators of the Double Four-Bar.

Table 3: Comparison between different simulations for the double fourbar mechanism

Method	**Type of step**	**Δt (ms)**	**Order**	**ERROR**	**CPU time (s)**	**Total Error**	**Violation of const.**
DIMCD	Fixed	10	2	1e-6	98	0.48	1,42e-14
GSTIFF I3	Variable	5-10	5	1e-6	1	0.57	
WSTIFF I3	Variable	5-10	6	1e-6	1	0.58	
HHT	Variable	3-10	2	1e-6	1	1,12	

In figure 14 is shown the level of energy conservation of the integrators. The first thing that is observed is that the HHT method does not comply with the minimum quality required, which is reflected in the accuracy of the results. Probably, the ERROR should be more stringent to keep the drift energy below 0,1 J. However, since HHT is a method of order 2 as is the DIMCD, then it is a good indicator to estimate the degree of accuracy of the method described in this document. The integrators GSTIFF and WSTIFF, when the mechanism is close to the singularity, they increase the order of integration and decrease the step, thus achieving the best response in terms of energy conservation. Despite, the total error is fairly similar to

DIMDC. These three integrators are far away from limitation of drift energy and obviously, their error parameter could be relaxed to improve the efficiency of the simulation.

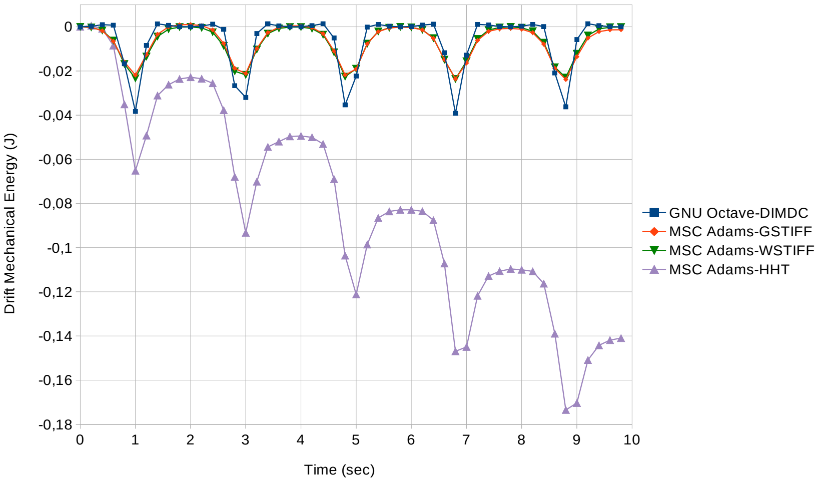

Figure 14: Energy conservation of the different integrators in Double Four-Bar.

6.4. Bricard mechanism

As it is known, it is typical example of an overrestricted system that, when the pairs are placed in a certain position, acquires 1 degree of freedom. Starting from the configuration of figure 15 it is released under the action of gravity.

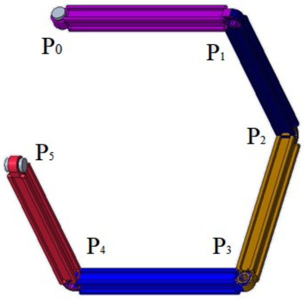

Figure 15: Bricard mechanism

In figure 16, the drift of the mechanical energy is shown for different time step using DIMCD. The authors of the benchmark example propose that the deviation of the total energy of the system should not exceed 0.001 J. Thus, simulation with a calculation time of 10 ms meet with this requirement. Under these conditions, the violation of the position restriction equations is approximately 1e-14, as can be seen in the figure 17.

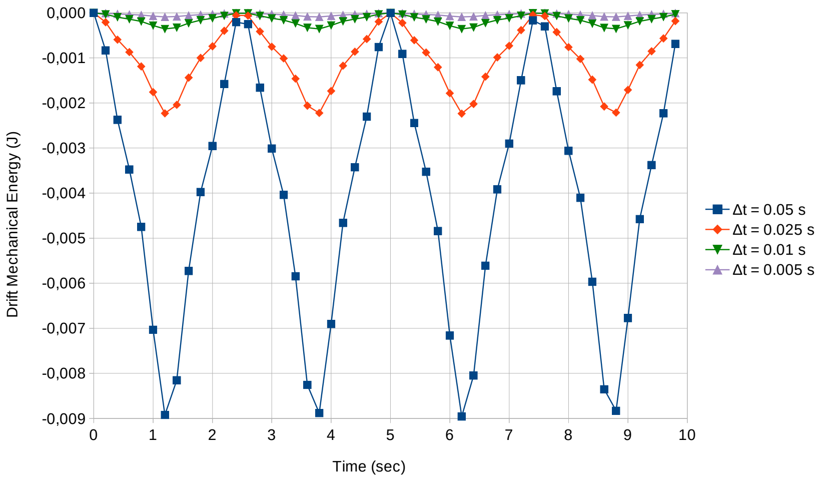

Figure 16: Energy conservation for different time step with DIMCD in Bricard mechanism.

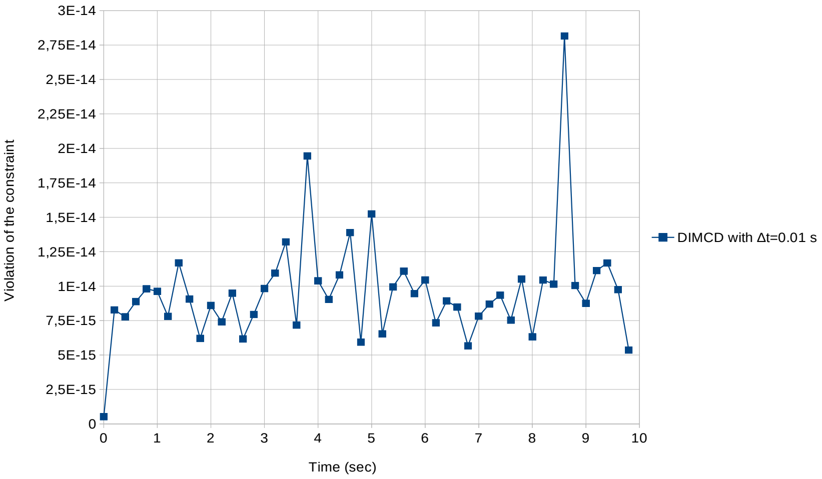

Figure 17: Violation of the position constraint equations of the Bricard mechanism.

As a consequence, in order to validate the solution, a comparison will be made following the same criterion of minimum drift energy. The reference value to obtain the total error has been acquired from a simulation with a deviation in the mechanical energy conservation of 9.6e-7 J. In figure 18 and 17 are shown the displacement of the point 2 in x and z direction, respectively. It can be appreciated how, as simulation progresses, the solutions are dispersed between MSC Adams and DIMCD. The parameters of the simulations are collected in table 4 where total error is also included. In figure 20, the conservation of the mechanical energy is shown.

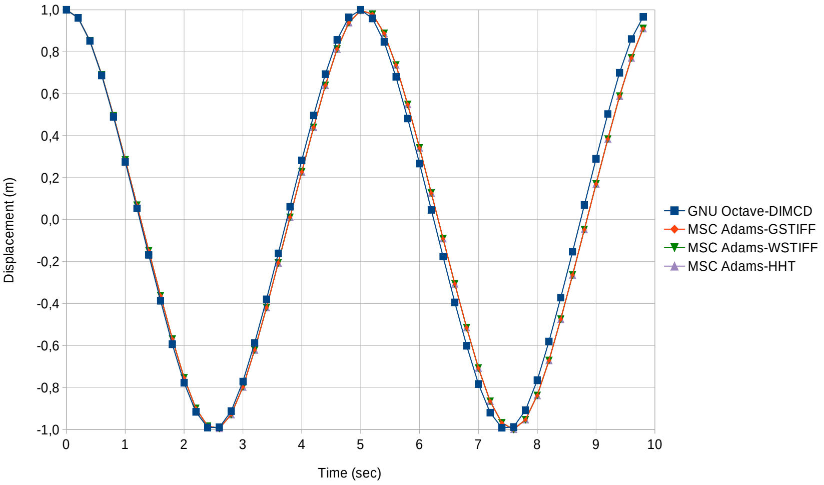

Figure 18: Displacement of point 2 in x direction of the Bricard mechanism.

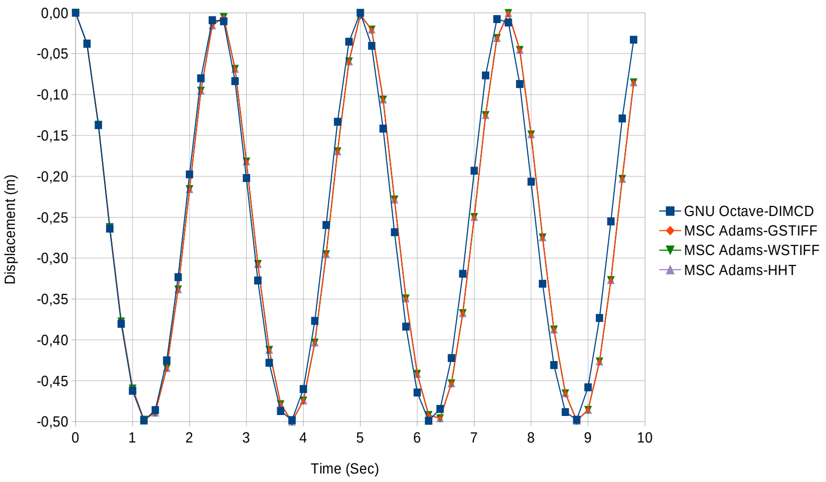

Figure 19: Displacement of point 2 in z direction of the Bricard mechanism.

Table 4: Comparative between different integrators for the Bricard mechanism

Method	Type of step	Δt (ms)	Order	ERROR	CPU time (s)	Total Error	Violation of const.
DIMCD	Fixed	10	2	1e-8	101	0,01	1,2e-14
GSTIFF I3	Variable	5-10	6	1e-8	2	0.98	
WSTIFF I3	Variable	10	6	1e-8	2	0.98	
HHT	Variable	1.7-10	2	1e-8	3	0.99	

It can be observed that the error achieved in the energy conservation committed by GSTIFF, WSTIFF and HHT is high compared to that obtained by the DIMCD and, of course, the total error is in accordance with this value. Therefore, the ERROR parameter should be adjusted to the quality requirement. In any case, the difference in accuracy that has been achieved with the DIMCD method is very significant. As a consequence, this makes it clear the influence of using exact Newton and not a quasi-Newton method to calculate the equilibrium.

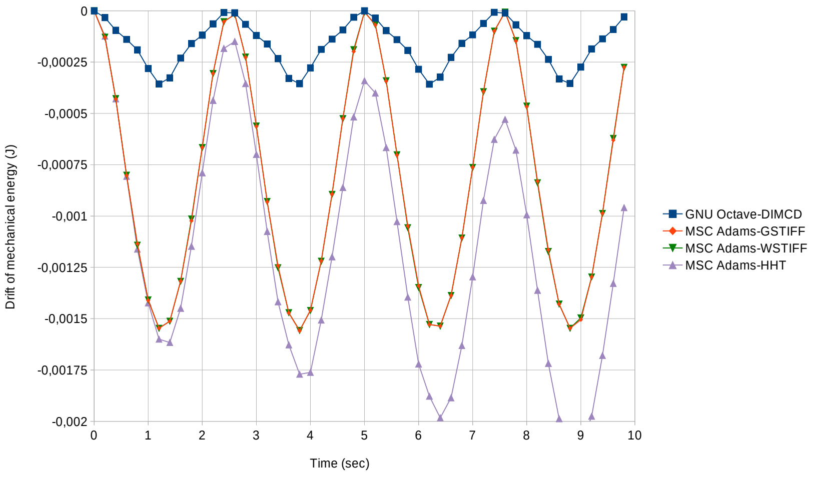

Figure 20: Conservations of the energy of the Bricard mechanism.

7. Conclusions

A new methodology was developed for the multibody analysis with rigid bodies where the system of equations is solved directly by the application of the method of central differences of order 2. In addition, in order to improve the accuracy, the equilibrium equations are solved in each instant of time in an analytical way, using the exact Newton method. When applying a method of order 2 it is not necessary to solve the problem of position and velocity at each moment apart from avoiding to duplicate the size of the problem, thus saving calculation time. The constraint equations are included through the null space, having to eliminate the equations incompatible with the infinitesimal movements of the system approached as a function of the rotations, obtaining a system of equal dimension to the degrees of freedom of the model. Therefore, it is demonstrated that the targets stated in the introduction have been accomplished.

With the aim of verifying the methodology, three different problems were solved including systems with external perturbation, singular positions and redundant constraint. The method was implemented in a preliminary form in GNU Octave, and the results were compared with those obtained through MSC Adams Student Edition for different integrators. According to the results, in examples with external forces and high rotation velocities, where the discretization is an essential issues, the method provides great level of precision even comparable to integrations with variable time step. Moreover, the method converges towards the solution with high accuracy and this fact is reflected in the conservation of the mechanical energy of the system. This degree of accuracy is largely due to the employ of exact derivatives to linearize the nonlinear equation of motion. Indeed, using the same admitted error for each iteration step in every integrator, the methodology affords remarkable results since it approaches the solution with a quadratic convergence.

As it was declared, the aim of this work was focused on verifying the accuracy of the method and, thus, the implementation is far from being optimized. In addition, making a comparison in terms of computational efficient would be inappropriate due to the size of the proposed problems and the use of interpreted language. In fact, the percentage of computational cost of the integration algorithm and the translation and execution process would be difficult to discern. Even though, a rough estimation suggests that the method could be quite efficient, taking into account the reduction in the among of degrees of freedom to solve for each iteration.

Future developments should include an optimized version of implementation and the inclusion of more types of joints and phenomena such as cams or contact problems.

Acknowledgements

Funding: The authors would like to thank to the Basque Government for the funding in a direct or indirect manner to the Research Group recognized under section IT 947-16; to the Spanish Ministry of Economy and Competitiveness for the grant through the project DPI2016-80372-R (AEI/FEDER, UE); and to the University of the Basque Country (UPV/EHU) for the predoctoral help for the training of research personnel.

The authors also wish to thank the research group of Professor Javier Cuadrado for their advice and guidance in the development of this work.

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

References

[1] García de Jalon, J., and Bayo, E., 1994, *Kinematic and Dynamic Simulation of Multibody Systemas*, Springer-Verlag New York.

[2] Shabana, A. A., 2009, *Computational Dynamics*, John Wiley & Sons.

[3] Wittenburg, J., 2007, *Dynamics of Multibody Systems*, Springer Science & Business Media.

[4] Featherstone, R., 2014, *Rigid Body Dynamics Algorithms*, Springer.

[5] Bauchau, O. A., 2010, *Flexible Multibody Dynamics*, Springer Science & Business Media.

[6] Shabana, A. A., 2013, *Dynamics of Multibody Systems*, Cambridge university press.

[7] Brenan, K. E., and Petzold, L. R., 1989, 'The Numerical Solution of Higher Iindex Differential/Algebraic Equations by Implicit Methods', SIAM J. Numer. Anal., **26**(4), pp. 976–996.

[8] Baumgarte, J., 1972, 'Stabilization of Constraints and Integrals of Motion in Dynamical Systems', Comput. Methods Appl. Mech. Eng., **1**(1), pp. 1–16.

[9] Bayo, E., De Jalon, J. G., and Serna, M. A., 1988, 'A Modified Lagrangian Formulation for the Dynamic Analysis of Constrained Mechanical Systems', Comput. Methods Appl. Mech. Eng., **71**(2), pp. 183–195.

[10] 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 Syst. Dyn., **4**(1), pp. 55–73.

[11] Blajer, W., 2002, 'Augmented Lagrangian Formulation: Geometrical Interpretation and Application to Systems with Singularities and Redundancy', Multibody Syst. Dyn., **8**(2), pp. 141–159.

[12] González, F., and Kövecses, J., 2013, 'Use of Penalty Formulations in Dynamic Simulation and Analysis of Redundantly Constrained Multibody Systems', Multibody Syst. Dyn., **29**(1), pp. 57–76.

[13] Pappalardo, C. M., and Guida, D., 2018, 'On the Computational Methods for Solving the Differential-Algebraic Equations of Motion of Multibody Systems', Machines, **6**(2), p. 20.

[14] Pappalardo, C. M., 2015, 'A Natural Absolute Coordinate Formulation for the Kinematic and Dynamic Analysis of Rigid Multibody Systems', Nonlinear Dyn., **81**(4), pp. 1841–1869.

[15] González, F., Dopico, D., Pastorino, R., and Cuadrado, J., 2015, 'Benchmarking of Augmented Lagrangian and Hamiltonian Formulations for Multibody System Dynamics', Proc. ECCOMAS Themat. Conf. Multibody Dyn. 2015, Multibody Dyn. 2015, pp. 1548–1559.

[16] Cuadrado, J., Cardenal, J., and Bayo, E., 1997, 'Modelling and Solution Methods for Efficient Real-Time Simulation of Multibody Dynamics', Multibody Syst. Dyn., **3**(1), pp. 259–280.

[17] Laulusa, A., and Bauchau, O. A., 2008, 'Review of Classical Approaches for Constraint Enforcement in Multibody Systems', J. Comput. Nonlinear Dyn., **3**(1), p. 11004.

[18] de Jalón, J. G., and Gutierrez-Lopez, M. D., 2013, 'Multibody Dynamics with Redundant Constraints and Singular Mass Matrix: Existence, Uniqueness, and Determination of Solutions for Accelerations and Constraint Forces', Multibody Syst. Dyn., **30**(3), pp. 311–341.

[19] Kane, T. R., and Levinson, D. A., 1985, *Dynamics, Theory and Applications*, McGraw Hill.

[20] Pogorelov, D., 1998, 'Differential–algebraic Equations in Multibody System Modeling', Numer. algorithms, **19**(1–4), pp. 183–194.

[21] Steigerwald, M. F., 1990, 'BDF Methods for DAEs in Multi-Body Dynamics: Shortcomings and Improvements', *Real-Time Integration Methods for Mechanical System Simulation*, Springer, pp. 345–352.

[22] Gear, C., 1971, ‘Simultaneous Numerical Solution of Differential-Algebraic Equations’, IEEE Trans. circuit theory, **18**(1), pp. 89–95.

[23] Gavrea, B., Negrut, D., and Potra, F. A., 2005, ‘The Newmark Integration Method for Simulation of Multibody Systems: Analytical Considerations’, Proc. ASME Des. Eng. Div. 2005, Pts A B, pp. 1079–1092.

[24] Yen, J., Petzold, L., and Raha, S., 1998, ‘A Time Integration Algorithm for Flexible Mechanism Dynamics: The DAE α-Method’, Comput. Methods Appl. Mech. Eng., **158**(3–4), pp. 341–355.

[25] Cardona, A., and Géradin, M., 1994, ‘Numerical Integration of Second Order Differential—Algebraic Systems in Flexible Mechanism Dynamics’, *Computer-Aided Analysis of Rigid and Flexible Mechanical Systems*, Springer, pp. 501–529.

[26] Neto, M. A., and Ambrôsio, J., 2003, ‘Stabilization Methods for the Integration of DAE in the Presence of Redundant Constraints’, Multibody Syst. Dyn., **10**(1), pp. 81–105.

[27] Cuadrado, J., Dopico, D., Naya, M. A., and Gonzalez, M., 2004, ‘Penalty, Semi-Recursive and Hybrid Methods for MBS Real-Time Dynamics in the Context of Structural Integrators’, Multibody Syst. Dyn., **12**(2), pp. 117–132.

[28] Newmark, N. M., 1959, ‘A Method of Computation for Structural Dynamics’, J. Eng. Mech. Div., **85**(3), pp. 67–94.

[29] Negrut, D., Rampalli, R., Ottarsson, G., and Sajdak, A., 2007, ‘On an Implementation of the Hilber-Hughes-Taylor Method in the Context of Index 3 Differential-Algebraic Equations of Multibody Dynamics (DETC2005-85096)’, J. Comput. Nonlinear Dyn., **2**(1), p. 73.

[30] Hilber, H. M., Hughes, T. J. R., and Taylor, R. L., 1977, ‘Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics’, Earthq. Eng. Struct. Dyn., **5**(3), pp. 283–292.

[31] Haug, E. J., Negrut, D., and Engstler, C., 1999, ‘Implicit Runge-Kutta Integration of the Equations of Multibody Dynamics in Descriptor Form’, J. Struct. Mech., **27**(3), pp. 337–364.

[32] Negrut, D., Haug, E. J., and German, H. C., 2003, ‘An Implicit Runge–Kutta Method for Integration of Differential Algebraic Equations of Multibody Dynamics’, Multibody Syst. Dyn., **9**(2), pp. 121–142.

[33] Pucheta, M. A., Paz, C. J., and Pereyra, M. E., 2014, ‘Representaciones Cinemáticas de Orientación y Ecuaciones de Estimación’, Mecánica Comput., **22**, pp. 2303–2324.

[34] Fox, R. L., 1971, *Optimization Methods for Engineering Design*, Addison-Wesley Pub. Co.

[35] Nocedal, J., and Wright, S. J., 2006, *Nonlinear Equations*, Springer.

[36] Burden, R. L., and Faires, J. D., 2010, ‘Numerical Analysis’, Cengage Learn., **9**.

[37] González, M., Dopico, D., Lugrís, U., and Cuadrado, J., 2006, ‘A Benchmarking System for MBS Simulation Software: Problem Standardization and Performance Measurement’, Multibody Syst. Dyn., **16**(2), pp. 179–190.

[38] John W. Eaton, David Bateman, Søren Hauberg, R. W., 2017, ‘GNU Octave Version 4.4.1 Manual: A High-Level Interactive Language for Numerical Computations.’

[39] Junkins, J. L., 1993, *Introduction to Dynamics and Control of Flexible Structures*, Aiaa.

[40] Avilés, R., 2003, *Métodos de Análisis Para Diseño Mecánico: Diseño Mecánico, Análisis Estático, Elementos Finitos En Estática, Elementos Finitos En Dinámica, Análisis de Fatiga*, Escuela Superior de Ingenieros.

[41] Tian, Q., Zhang, Y., Chen, L., and Flores, P., 2009, 'Dynamics of Spatial Flexible Multibody Systems with Clearance and Lubricated Spherical Joints', Comput. Struct., **87**(13–14), pp. 913–929.

[42] Bayo, E., and Avello, A., 1994, 'Singularity-Free Augmented Lagrangian Algorithms for Constrained Multibody Dynamics', Nonlinear Dyn., **5**(2), pp. 209–231.