The 5th Joint International Conference on Multibody System Dynamics
June 24 – 28, 2018, Lisbon, Portugal

Dynamic Analysis of Planar Multibody Systems with Fully Cartesian Coordinates

Ivo Roupa[1], Sérgio B. Gonçalves[1] and Miguel Tavares da Silva[1]

[1] *IDMEC, LAETA, Instituto Superior Técnico, Universidade de Lisboa, Portugal*
{ivo.roupa, sergio.goncalves, MiguelSilva}@tecnico.ulisboa.pt

ABSTRACT — Multibody dynamics formulations have been successfully applied in the analysis of mechanical systems, as they allow for an efficient modelling and simulation of complex structures. These formulations require the use of a set of parameters called generalized coordinates that uniquely defines the position and orientation of each body in space. Different multibody formulations can be found in literature, having its own advantages and limitations.
This work proposes a novel formulation, Fully Cartesian Coordinates, that presents a new kinematic structure, based on points and vectors, to describe a generic rigid body. This concept eliminates the use of angular variables and body-dependent equations, two of the drawbacks of the most common formulations. The assessment of the formulation is performed by applying it in the simulation of two classical mechanisms, a simple planar pendulum and a planar slider crank, and their results compared with benchmark data from IFToMM library.
The formulation enabled to model with success the mechanisms, generating constraint equations and contributions to the Jacobian matrix and the right-hand side vectors that are constant, linear or quadratic. When compared the results with the benchmark data, high agreement scores were found in the system coordinates for both pendulum (ICC = 1.000) and slider crank (ICC = 0.998).
Fully Cartesian Formulation proved to be an interesting option in the modelling and analysis of multibody systems, due to the low degree of non-linearity of the equations it generates, high level of systematization of the model, intuitiveness and easiness of utilization and reliability and validity of the results.

1. Introduction

Classical formulations, based on analytical dynamics, have been used to describe simple mechanical systems. Although, these formulations tend to use a reduced set of coordinates, they generate highly non-linear equations, as well as they require a considerable user expertise to discretize the system and achieve a solution. On the other hand, formulations based on computational dynamics are a good approach for the systematization of the modeling process and analysis of medium and large systems. Within these formulations, multibody dynamics stand out, as they allow for an efficient description of mechanical systems, allowing the inverse and forward dynamic analysis of large systems.

Multibody systems are described as a collection of bodies interconnected by kinematic pairs and acted upon by external forces. All these systems, without exception, require the use of a set of parameters that uniquely defines the position and orientation of each body during the period of analysis. These parameters are referred to as generalized coordinates. Ideally, a multibody formulation should be able to represent in a systematic form a given mechanical system with a minimum set of coordinates and it should generate kinematic constraints with a low degree of non-linearity, being simultaneously simple and fast to evaluate. However, we can hardly reconcile such characteristics in a single formulation, as so compromises have to be found. The differences in the multibody formulations are essentially related with the total number of generalized coordinates necessary to model the system and the number and degree of non-linearity of the kinematic constraint equations that these generate. Hence, its selection has direct influence on the complexity of the problem to solve, on its computational performance, systematization ability, and, equally important, on the intuitiveness and easiness of utilization.

Although several formulations are available in the literature, the most applied are the *relative coordinates formulation* (RCF) [1], the *reference point coordinate formulation* (RPCF) [2] also known as *Cartesian coordinates* and the *natural coordinate formulation* (NCF) [3].

Relative coordinates use a minimal number of generalized coordinates per body and are particularly useful to represent the structure of open chain mechanisms. In this formulation, the position and the orientation of each body is defined with respect to an adjacent element and the equations of motion are compact but dense. Furthermore, if the multibody system is composed by several closed chains, the choice of the loop equations is a non-trivial task [1].

RPCF or Cartesian Coordinates formulation is the most commonly methodology used in analysis of rigid multibody systems. These are a suitable choice for both open and closed chain mechanisms, since it directly defines the absolute position and orientation of each element of the system, regardless the motion of the adjacent model elements. In spatial systems, RPCF requires the use of six or seven generalized coordinates per body, depending if the formulation makes use of Euler angles or Euler parameters [2]. On the other hand, three generalized coordinates per body are required to describe the kinematics of planar systems. Usually, these are the cartesian coordinates of a point of the body (the *reference point*, which often is the center of mass (CoM) and an angle between its local reference frame and a system of inertial axes. In this way, although the RPCF needs a larger number of variables, when compared to RCF, it generates sparse matrices, which are usually more numerically efficient. Moreover, this formulation allows for a systematization of different mechanical systems, being also intuitive due to the nature of its coordinates. However, RPCF generates kinematic constraint equations with transcendental terms that are computationally expensive to evaluate [2].

NCF is an alternative formulation developed to avoid the computational problems reported for the RPCF and RCF. This formulation uses only Cartesian coordinates of points, usually located at relevant positions of the body (e.g. joints and extremities), and vectors to define the bodies kinematic structure. This is in fact the reason why these coordinates were originally referred to as Fully Cartesian coordinates. The term Natural coordinates arose solely from the fact that, with the purpose of reducing the number of coordinates required to model a given system, points can be shared by different bodies. Therefore, several kinematic joints, such as the revolute joints in 2D and spherical in 3D, would appear 'naturally' without the need of an explicit joint definition, i.e. without the use of specific kinematic constraint equations [3]. Despite presenting the reported advantages, moving the generalized coordinates to relevant points introduces some intricacies. The rigid body definition and the mass matrices vary with the specific structure of the modelled element. Additionally, with shared points, system matrices become coupled and since there is no explicit joint definition, reaction forces cannot be calculated directly from the equations of motion of the system [3]. When compared to Cartesian coordinates, NCF requires, on average, less coordinates if points are shared between bodies and generates kinematic constraint equations that are quadratic or linear in nature. However, and despite its simplicity, this formulation is less systematic than the Cartesian coordinates and its use far less intuitive.

An alternative approach to modeling planar (2D) multibody systems is the Fully Cartesian Coordinates formulation (FCCF), based on a methodology previously presented in Gameiro et al. [4] and applied in Uhlar et al. and Pappalardo et al. [5], [6]. In this formulation the two major characteristics of Natural coordinates formulation are maintained, i.e., multibody systems are still described using only with Cartesian coordinates and kinematic constraint equations are still quadratic or linear, but the reported disadvantages are eliminated, i.e., the definition of the rigid bodies are not body-specific dependent and the matrices are not coupled due to points sharing. This is accomplished by introducing the concept of a generic rigid body, which presents a predetermined kinematic structure. In planar motion, this body is defined by one point, located at its center of mass, and one unit vector. The two Cartesian coordinates of the relevant point are used to describe rigid body translations and the two Cartesian coordinates of the unit vector its orientation. The introduction of this generic rigid body presents several advantages regarding the formulation and systematization of the model, as it will be seen later in the paper. The adoption of a constant rigid body structure brings the modeling approach closer to the one adopted with Cartesian coordinates and familiar to most users.

Taking in consideration the advantages and disadvantages of each multibody formulation, this work aims the development of a novel formulation based on Fully Cartesian Coordinates. The formulation characteristics will be explored, alongside with its strengths and disadvantages. In order to prove its reliability and validity, the proposed methodology was implemented in an in-house software, developed using Python language, being applied to perform a forward dynamics analysis of two simple mechanisms, respectively, a planar pendulum and a planar slider crank. The results will be posteriorly compared with benchmark data for the same mechanisms.

2. Methodology

The present work is divided in two major sections, the first one comprises the theoretical framework behind the Fully Cartesian Coordinates formulation, namely the equations necessary to model any constrained multibody system, and the second one explores its application in the study of two IFToMM benchmark problems. Results obtained through a forward dynamics simulation will be compared with benchmark data through specific statistical procedures to prove formulation validity and reliability.

2.1. Fully Cartesian Coordinates Formulation

2.1.1. Kinematic structure of the generic Rigid Body

Contrarily to the Natural Coordinates, Fully Cartesian coordinates introduce a new kinematic structure, which presents a general predetermined structure. This fact allows for an easier systematization of the model implementation, avoiding the need of body-specific equations, and brings this formulation closer to the one adopted in RPCF from a modelling perspective. This generic rigid body is defined by one point (point $\mathbf{O}_i'$ located at its center of mass), which describes the body translations, and one unit vector ($\mathbf{U}_i$) to define its orientation (Fig. 1). Therefore, the generalized coordinates of the body *i* ($\mathbf{q}_i$) can be defined as:

$$\mathbf{q}_i = \left\{ r_{i_x} \quad r_{i_y} \quad U_{i_x} \quad U_{i_y} \right\}^T \tag{1}$$

Where r_{i_x} and r_{i_y} are the x and y coordinates of point $\mathbf{O}_i'$ and U_{i_x} and U_{i_y} are the x and y components of vector $\mathbf{U}_i$. Since $\mathbf{U}_i$ is defined as a unit vector, a unit module condition is introduced in the constraint equations ($\mathbf{\Phi}$) to relate these two coordinates (see Tab. 1). One of the major advantages of FCCF is the efficient definition of an arbitrary point **P** as function of the coordinates of body *i*:

$$\mathbf{r}_{P_i} = \begin{Bmatrix} r_{i_x} \\ r_{i_y} \end{Bmatrix} = \mathbf{r}_{0'} + \mathbf{r}_{0'P_i} \tag{2}$$

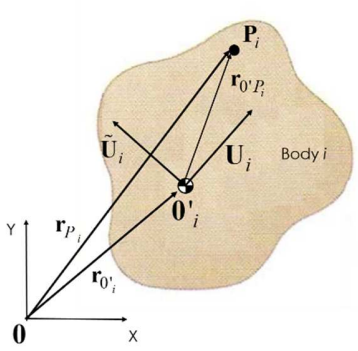

Fig. 1: Description of the generic rigid body in Fully Cartesian Coordinates

In its turn, vector $\mathbf{r}_{0'P_i}$ can be written also as a linear combination of the body *i* local reference frame:

$$\mathbf{r}_{0'P_i} = c_1 \mathbf{U}_i + c_2 \tilde{\mathbf{U}}_i \tag{3}$$

where $\tilde{\mathbf{U}}_i$ is a unitary vector normal to $\mathbf{U}_i$ and coefficients c_1 and c_2 are the local coordinates of a generic point $\mathbf{P}_i$ with respect to body *i* local reference frame ($\mathbf{0}'_i$). Therefore, the global coordinates of point $\mathbf{P}_i$ can be computed directly from the generalized coordinates of body *i*:

$$\mathbf{r}_{i_p} = \mathbf{C}_{P_i} \mathbf{q}_i \tag{4}$$

where $\mathbf{C}_{P_i}$ is a constant transformation matrix with dimensions 2x4:

$$\mathbf{C}_{P_i} = \begin{bmatrix} 1 & 0 & c_1 & -c_2 \\ 0 & 1 & c_2 & c_1 \end{bmatrix} \tag{5}$$

Considering the first and second time derivatives of equation (4, the velocity ($\dot{\mathbf{r}}_{i_p}$) and acceleration($\ddot{\mathbf{r}}_{i_p}$) of point $\mathbf{P}_i$, can also be calculated directly from the vector of the generalized velocities ($\dot{\mathbf{q}}_i$) and accelerations ($\ddot{\mathbf{q}}_i$) of body *i*:

$$\dot{\mathbf{r}}_{i_p} = \mathbf{C}_{P_i} \dot{\mathbf{q}}_i \tag{6}$$

$$\ddot{\mathbf{r}}_{i_p} = \mathbf{C}_{P_i} \ddot{\mathbf{q}}_i \tag{7}$$

2.1.2. Kinematic Constraints

The modelling of a multibody system requires the implicit or explicit definition of the elements that compose it. Their description is performed through mathematical equations, commonly named kinematic constraints, which relate or state dependencies between different variables, such as generalized coordinates, reference points, angular drivers, among others. This section presents the constraint equations ($\mathbf{\Phi}$) for the most common elements applied during the modelling of mechanical systems, considering a Fully Cartesian Coordinates formulation. Their equations, as well as the respective contributions to the Jacobian matrix ($\mathbf{\Phi}_q$); and right-hand-side vector of the velocities ($\boldsymbol{\upsilon}$) and accelerations ($\boldsymbol{\gamma}$) are compiled in **Error! Reference source not found.**.

2.2. Equations of Motion and Dynamic Analysis

The equations of motion describe mathematically the dynamic behavior of a given physical system in time. In a framework of a constrained multibody system, these equations relate the kinematics, inertial forces and generalized accelerations of the system and the internal and external forces that act on it as expressed in [7]:

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_q^T \\ \mathbf{\Phi}_q & 0 \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{g} \\ \boldsymbol{\gamma} \end{Bmatrix} \tag{8}$$

where **M** is the mass matrix of the system, $\ddot{\mathbf{q}}$, $\boldsymbol{\lambda}$ and **g** are respectively the vector of the generalized accelerations, Lagrange multipliers and generalized forces. From a numerical point of view, the use of the kinematic acceleration equation in forward dynamics simulations can result in convergence problems [7]. Hence, it is advised to use a stabilization method to avoid constraint violations and lead to a more stable integration procedure. In this work, a Baumgarte stabilization methodology [8] was implemented, being analyzed the influence of different Baumgarte coefficients (α and β) in the convergence of the method.

Kinematic Constraint	$\boldsymbol{\Phi}$	$\boldsymbol{\Phi}_q$	υ	γ
Unit Vector(1)	$\mathbf{u}^T\mathbf{u}-1=0$	$\left[2\mathbf{u}^T\right]_{1x2}$	0	$-2\dot{\mathbf{u}}^2$
Pinned Joint(2)	$\mathbf{C}_{P_i}\mathbf{q}_i-\mathbf{r}^*=0$	$\left[\mathbf{C}_{P_i}\right]_{2x4}$	$\dot{\mathbf{r}}^*$	$\ddot{\mathbf{r}}^*$
Roller Joint(3)	$\mathbf{b}\left[\mathbf{C}_{P_i}\mathbf{q}_i-\mathbf{r}^*\right]=0$	$\mathbf{b}\left[\mathbf{C}_{P_i}\right]_{1x4}$	0	0
Revolute Joint(4)	$\mathbf{C}_{P_i}\mathbf{q}_i-\mathbf{C}_{P_j}\mathbf{q}_j=0$	$\left[\mathbf{C}_{P_i} \quad -\mathbf{C}_{P_j}\right]_{2x8}$	0	0
Translation Joint (5)	$\begin{bmatrix}\tilde{\mathbf{u}}^T.\mathbf{v}-LuLv.\sin(\theta\langle\mathbf{u},\mathbf{v}\rangle)\\ \mathbf{x}^T.\mathbf{w}-LxLw.\cos(\theta\langle\mathbf{x},\mathbf{w}\rangle)\end{bmatrix}=0$	$\begin{bmatrix}\mathbf{E}.\mathbf{q}_j & -\mathbf{q}_j{}^T(\mathbf{G}^T+\mathbf{G})\\ \mathbf{x} & \mathbf{w}\end{bmatrix}_{2x4}$	$\begin{bmatrix}0\\ -Lx.Lw.\sin(\theta\langle\mathbf{x},\mathbf{w}\rangle).\dot{\theta}\end{bmatrix}_{2x1}$	$\begin{bmatrix}2\dot{\mathbf{u}}^T\dot{\mathbf{v}}\\ Lx.Lw.(\cos(\theta\langle\mathbf{x},\mathbf{w}\rangle\ddot{\theta}^2+\sin\theta\langle\mathbf{x},\mathbf{w}\rangle\dot{\theta})-2\dot{\mathbf{x}}^T.\dot{\mathbf{w}}\end{bmatrix}_{2x1}$
Translation Joint (6)	$\begin{bmatrix}\tilde{\mathbf{u}}^T.\mathbf{v}-LuLv.\sin(\theta\langle\mathbf{u},\mathbf{v}\rangle)\\ \tilde{\mathbf{x}}^T.\mathbf{w}-LxLw.\sin(\theta\langle\mathbf{x},\mathbf{w}\rangle)\end{bmatrix}=0$	$\begin{bmatrix}\mathbf{E}.\mathbf{q}_j & -\mathbf{q}_j{}^T(\mathbf{G}^T+\mathbf{G})\\ \tilde{\mathbf{x}} & \tilde{\mathbf{w}}\end{bmatrix}_{2x4}$	$\begin{bmatrix}0\\ -Lx.Lw.\cos(\theta\langle\mathbf{x},\mathbf{w}\rangle).\dot{\theta}\end{bmatrix}_{2x1}$	$\begin{bmatrix}2\dot{\mathbf{u}}^T\dot{\mathbf{v}}\\ -Lx.Lw.(\sin\theta\langle\mathbf{x},\mathbf{w}\rangle\dot{\theta}^2-\cos(\theta\langle\mathbf{x},\mathbf{w}\rangle\ddot{\theta})-2\dot{\mathbf{x}}^T.\dot{\mathbf{w}}\end{bmatrix}_{2x1}$
Translation Revolute joint (7)	$\tilde{\mathbf{u}}^T.\mathbf{v}-LuLv\sin(\theta\langle\mathbf{u},\mathbf{v}\rangle)=0$	$\left[\mathbf{E}.\mathbf{q}_j \quad -\mathbf{q}_j{}^T(\mathbf{G}^T+\mathbf{G})\right]_{2x4}$	0	$2\dot{\mathbf{u}}^T\dot{\mathbf{v}}$
Angular Driver (Dot Product)	$\mathbf{u}^T.\mathbf{v}-LuLv.\cos(\theta\langle\mathbf{u},\mathbf{v}\rangle)=0$	$\left[\mathbf{v} \quad \mathbf{u}\right]_{1x4}$	$-Lu.Lv.\sin(\theta\langle\mathbf{u},\mathbf{v}\rangle).\dot{\theta}$	$Lu.Lv.(\cos(\theta\langle\mathbf{u},\mathbf{v}\rangle\ddot{\theta}^2+\sin\theta\langle\mathbf{u},\mathbf{v}\rangle\dot{\theta})-2\dot{\mathbf{u}}^T.\dot{\mathbf{v}}$
Angular Driver (Cross Product)	$\tilde{\mathbf{u}}^T.\mathbf{v}-LuLv.\sin(\theta\langle\mathbf{u},\mathbf{v}\rangle)=0$	$\left[\tilde{\mathbf{v}} \quad \tilde{\mathbf{u}}\right]_{1x4}$	$-Lu.Lv.\cos(\theta\langle\mathbf{u},\mathbf{v}\rangle).\dot{\theta}$	$-Lu.Lv.(\sin\theta\langle\mathbf{u},\mathbf{v}\rangle\dot{\theta}^2-\cos(\theta\langle\mathbf{u},\mathbf{v}\rangle\ddot{\theta})-2\dot{\mathbf{u}}^T\dot{\mathbf{v}}$
Linear Driver	$\mathbf{C}_{P_i}\mathbf{q}_i-\mathbf{r}_t^*=0$	$\left[\mathbf{C}_{P_i}\right]_{2x4}$	$\dot{\mathbf{r}}_t^*$	$\ddot{\mathbf{r}}_t^*$
(1)	$\mathbf{u}$: generic vector belonging to body i $\dot{\mathbf{u}}$: vector $\mathbf{u}$ angular velocity.			
(2)	$\mathbf{C}_{P_i}$ - transformation matrix of point **P** with respect to body i i = index of th body $\mathbf{r}^*$: prescribed joint global coordinates / $\dot{\mathbf{r}}^*$: prescribed joint global velocities / $\ddot{\mathbf{r}}^*$: prescribed joint global accelerations			
(3)	$\mathbf{b}=\begin{bmatrix}1 & 0\end{bmatrix}$ if roller joint is aligned with horizontal direction (x axis) $\mathbf{b}=\begin{bmatrix}0 & 1\end{bmatrix}$ if roller joint is aligned with vertical direction (y axis)			
(4)	$\mathbf{C}_{P_i}$ - transformation matrix of point **P** with respect to body i. $\mathbf{C}_{P_j}$ - transformation matrix of point P with respect to body j. $\mathbf{q}_i$: vector of generalized coordinates of body i $\mathbf{q}_j$: vector of generalized coordinates of body j			

(5)	$\tilde{\mathbf{u}}$: vector perpendicular to vector $\mathbf{u}$ $\mathbf{u}, \mathbf{x}$: generic vectors belonging to body j $\mathbf{v}, \mathbf{w}$: generic vectors belonging to body j $\theta\langle \mathbf{u}, \mathbf{v} \rangle$: angle between $\mathbf{u}$ and $\mathbf{v}$ $\mathbf{u}$: $\mathbf{C}_{A_i}\mathbf{q}_i - \mathbf{C}_{B_j}\mathbf{q}_j$ $\mathbf{v}$: $\mathbf{C}_{C_j}\mathbf{q}_j - \mathbf{C}_{B_j}\mathbf{q}_j$ Lu and Lv : vector $\mathbf{u}$ and $\mathbf{v}$ length $\mathbf{A}, \mathbf{B}, \mathbf{C}$: three collinear points belonging to body *i*, body *j* and body *j* respectively $\mathbf{C}_{A_i}$: transformation matrix of point A with respect to body *i* $\mathbf{C}_{B_j}$ and $\mathbf{C}_{C_j}$: transformation matrices of points $\mathbf{B}$ and $\mathbf{C}$ with respect to body *j* $\theta\langle \mathbf{x}, \mathbf{w} \rangle$: constant angle between vectors $\mathbf{x}$ and $\mathbf{w}$ $\mathbf{E} = \mathbf{C}_{A_i}^T \mathbf{R}_{90}^T \mathbf{C}_{CB_j}$ $\mathbf{G} = \mathbf{C}_{B_j}^T \mathbf{R}_{90}^T \mathbf{C}_{CB_j}$ $\mathbf{R}_{90}^T$ - transpose of a 2x2 90 degrees rotation matrix $\mathbf{C}_{CB_j} = \mathbf{C}_{B_j} - \mathbf{C}_{C_j}$ $\dot{\theta}$: angular velocity $\ddot{\theta}$: angular acceleration $\dot{\mathbf{v}}$: vector $\mathbf{v}$ linear velocity
(6)	$\tilde{\mathbf{x}}$: vector perpendicular to vector $\mathbf{x}$ $\tilde{\mathbf{w}}$: vector perpendicular to vector $\mathbf{w}$
(10)	$\mathbf{r}_t^*$ - prescribed linear trajectory / $\dot{\mathbf{r}}_t^*$ - prescribed linear velocity / $\ddot{\mathbf{r}}_t^*$ - prescribed linear acceleration.

Tab. 1: Most common elements constraint equations ($\mathbf{\Phi}$) and their contributions to the Jacobian matrix ($\mathbf{\Phi}_q$), right-hand-side vector of the velocity ($\boldsymbol{\upsilon}$) and acceleration ($\boldsymbol{\gamma}$)

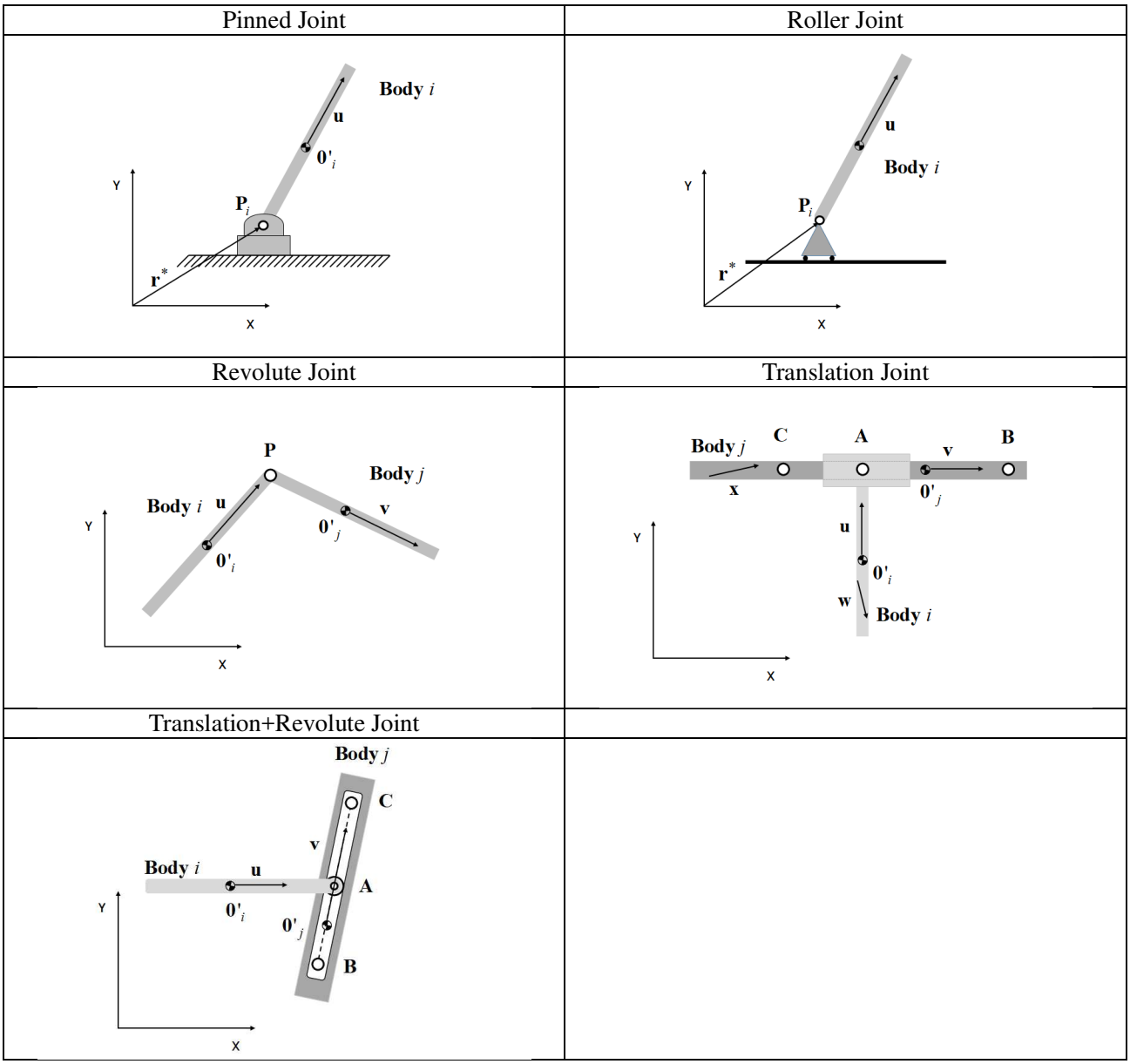

Fig. 2: Multibody System: most common joints

2.3. Mechanisms modelling

The validation of the model was performed by analyzing its efficiency in the study of two mechanisms and their results compared with a collection of benchmark problems available on the Library of Computational Benchmark Problems [9]. The formulation was implemented in an in-house software developed in Instituto Superior Técnico using Python language (v2.7.10). Both models followed the specifications presented in the benchmark database [9]: a) *Simple Pendulum*: mechanism composed by a point with mass (m = 1 kg) and a massless rod (L = 1 m). The rod is connected to the ground via a revolute joint that constrains the system motion to the x-y plane (see Fig. 3). The system moves under gravity effects (-9.81 m.s^{-2} along the global y axis) from an initial position in which the coordinates of point $P1$ are x_0 = -1 m, y_0 = 0 m. The initial velocity of the pendulum is zero and the total simulation time is 10 s; b) *Slider-Crank*: mechanism composed by two rods with equal length (L = 1 m), a uniformly distributed mass (m = 1 kg), and a square cross section of width r = 0.1 m (see Fig. 3). The slider is considered to be massless. There is no friction between the slider and the ground. Point $P3$ is constrained to move on the x axis. The system is in a singular configuration, when the value of the angle of the first rod with respect to the x axis is $\theta = n\pi/2$, with n = 0, 1, 2… The system moves under gravity effects (-9.81 m/s2 along the

global y axis) from the initial position (presented in Fig 3), in which $\theta = \pi / 4$. The initial velocity of point $P3$ is 4 m/s in the negative direction of the global x axis. The total simulation time is 10 s.

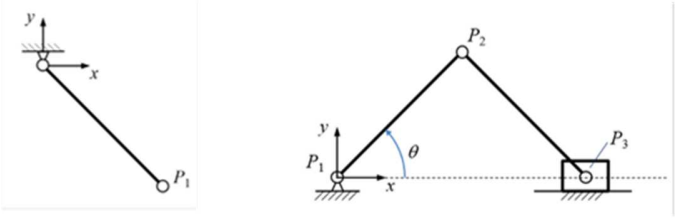

Fig. 3: Planar Pendulum (Left); Planar Slider-Crank (Right) (adapted from IFToMM benchmark library [9])

For both mechanisms a forward dynamics simulation was performed considering a computer with the following characteristics: a) CPU: Intel® Core ™ i5 CPU 750 @ 3.67Ghz b) RAM: 12 GB; c) GPU: NVidia Quadro FX 380; d) Operative System: Windows 10 Pro 64 bits. To solve the equations of motion (8) the *scipy.integrate.odeint* integrator was used with the default parameters (variable step integration). In order to understand the influence of the Baumgarte stabilization in the simulation convergence, different coefficients were tested. Following Nikravesh recommendations, equal values were considered for α and β [2], varying from 0 to 100 with a step of 5. The 'x' and 'y' coordinates of point P1 in pendulum and P3 in slider crank, the variation of the system mechanical energy, the constraints violations and other computational outcomes were compared with data available in the database of the Library of Computational Benchmark Problems, specifically the models implemented by Francisco González using Natural Coordinates formulation [9]. A statistical analysis was performed considering a Bland-Altman Limits of Agreement methodology [10] and Intraclass Correlation (ICC) [11] for assessment of the data consistency and agreement.

3. Results

The validation of the formulation presented in this work was performed by applying it in the kinematic and dynamic analysis of simple mechanisms. In general, the kinematic formulation allowed to model in an intuitive way the mechanisms in study, resulting in constraint equations only dependent of linear and quadratic terms. No problems were found regarding the convergence of the methodology in the inverse kinematic and dynamic analysis of these mechanisms. In all cases, the outcomes presented an excellent agreement with the results obtained using the respective algebraic equations for both kinematic and dynamic patterns. The formulation was also accessed by comparing its performance during forward dynamic simulation with benchmark data for two mechanisms available in IFToMM website. In the pendulum case, the methodology converged with success to a feasible solution for the tested Baumgarte coefficients, presenting a high correlation with the provided data. On other hand, the slider crank methodology converged for almost all the tested coefficients. However, for a set of parameters ($0 \leq \alpha = \beta < 45$]), the solution presented an abnormal movement, behaving like a pendulum after a given time instant, i.e. point P_3 stops in the same position as point P_1 and the two rods start to swing together. For the purpose of presenting the data, sections 3.1 and 3.2 will present respectively the results obtained considering the Baumgarte coefficients equal to 0 and 50.

3.1. Simple Pendulum

Fig. 4 presents the x and y coordinates of point P1 during the simulation. As expected, the coordinates followed a pattern typical of a pendulum. The comparison with the benchmark data does not allow to observe significant differences in the pattern and magnitude of point P1 for the entire simulation time. This fact is supported

by the analysis of the Bland-Altman plot (see Fig. 5) and Intraclass Correlation Coefficients (ICC) between the two cases, which present an excellent score both for consistency and absolute agreement (see Tab. 2).

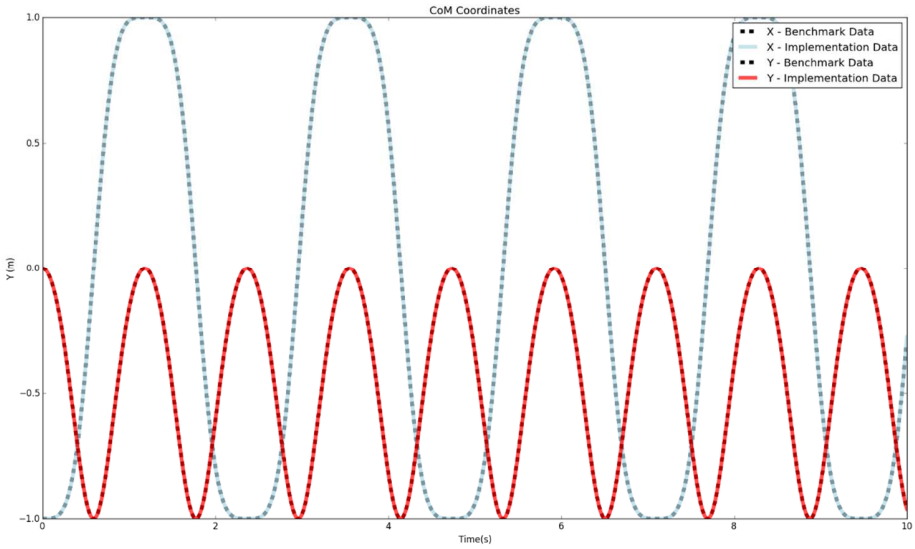

Fig. 4: Time-history of the x and y coordinates of point $P1$ for the pendulum simulation

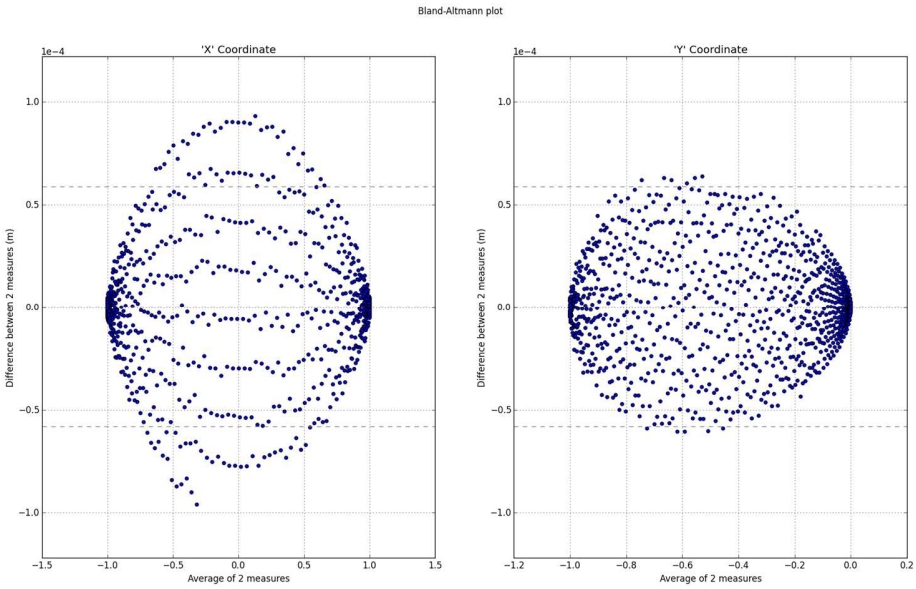

Fig. 5: Bland-Altman plot for x and y coordinates of point $P1$ for the pendulum simulation

	Intraclass Correlation	
	Consistency	Absolute Agreement
'x' coordinate	1.0	1.0
'y' coordinate	1.0	1.0

Tab. 2: Intraclass Correlation Coefficients for the pendulum simulation

Fig. 6 and Fig. 7 show the variation of the mechanical energy of the system and the violations of the constraints during the simulation. The mechanical energy is obtained as the sum of the kinetic and potential energy of the mechanism and the constraints violations are calculated as the norm of the kinematic constraints vector. In the first case is possible to observe variations in the system mechanical energy in the order of 10^{-6}, lower than the value presented in the benchmark data. However, the obtained results show an increase of the variation of mechanical energy along the simulation time, a pattern not observed in the benchmark data (see Fig. 6). A similar

behavior is observed for the constraints violations, increasing its value with the evolution of the simulation time. Despite presenting lower values in both cases, it is important to note the differences observed in the order of magnitude of the constraints violation, probably related with differences in the error tolerance of the integrator (Benchmark: 10^{-9}; RFFC: 10^{-8}). Other computational parameters related with the formulation implementation can be consulted in Tab. 3

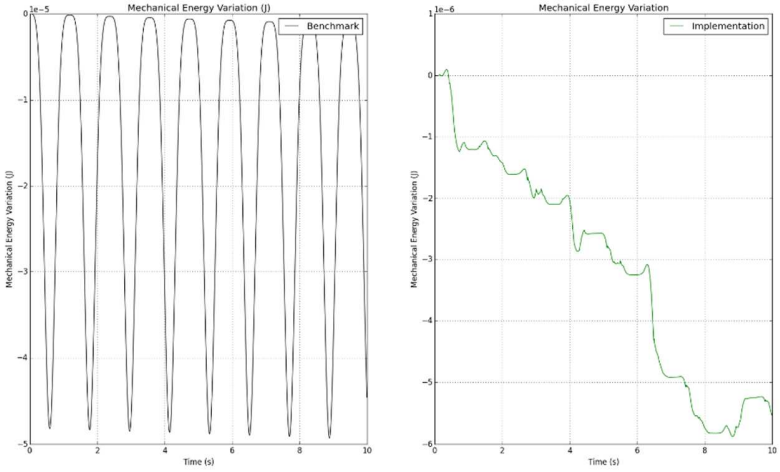

Fig. 6: Variation of the mechanical energy for the pendulum simulation

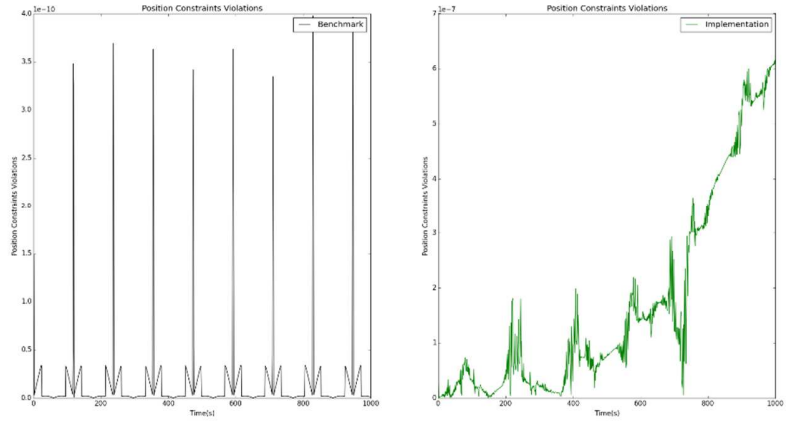

Fig. 7: Constraints violations for the pendulum simulation

	Benchmark	Simulation
Programming Language	C++	Python
Computer	Intel Core i5-4250U @ 1.30 GHz 1.90 GHz - 8 GB RAM	Intel® Core i5 CPU 750 @ 3.67 Ghz - 12 GB
Number of simulations	NA	1.0E3
Average CPU Time (s)	0.53	0.36
Integrator	trapezoidal rule	Adams / BDF
Maximum admissible error:	1.0E-9	1.5E-8
Time step duration (s)	1.0E-3	Variable
Number of time steps	-	1416
Maximum number of iterations per time	12	27

Tab. 3: Computer specifications, integrator parameters and simulation time for the pendulum simulation

3.2. Slider Crank

As in the pendulum case, the results obtained for the simulation of the slider crank mechanism followed the patterns presented in the benchmark data. The ICC analysis of point $\mathbf{P}_2$ shows a high correlation between simulation and benchmark data both for consistency and absolute agreement (see Tab. 4). Despite the high correlation values, a detailed analysis of the differences indicates an increase of this value along the simulation time, presenting a maximum value of 0.03m for '*x*' coordinate and 0.05m for '*y*' coordinate at t=10s (see Fig. 9).

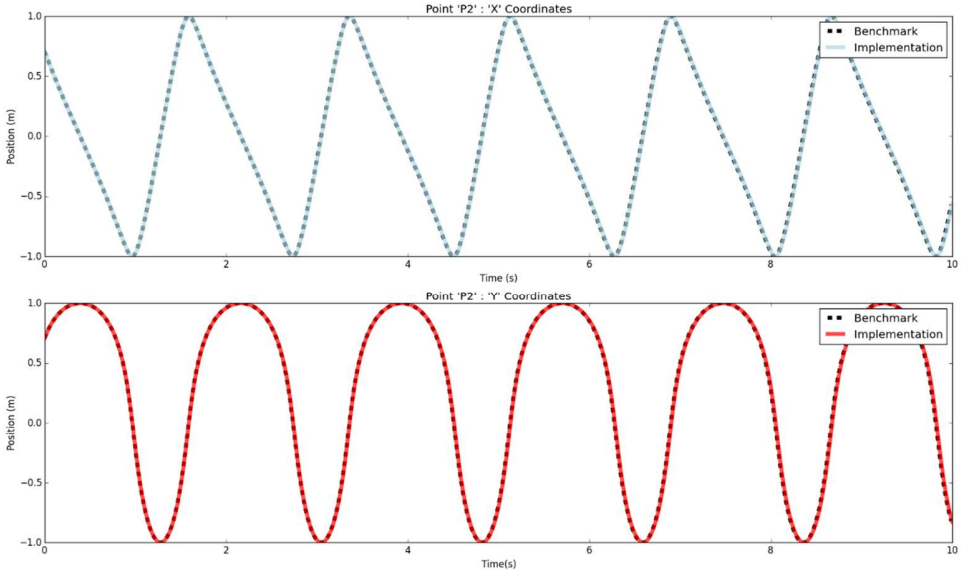

Fig 8: Time-history of the x and y coordinates of point P2 for the slider crank simulation

Although the simulation results for the mechanical energy variation indicates an increase of one order of magnitude when compared with the benchmark data, the variations oscillated around 10^{-3} J. Low values were also observed in the constraints violation (FCCF: 10^{-8}; Benchmark: 10^{-11}), indicating the convergence of the methodology along the simulation time. When compared with the pendulum case, an increase of approximately 1.8s was observed in the simulation time, as well as an increase in the number of time steps and maximum number of iterations (see Tab. 5).

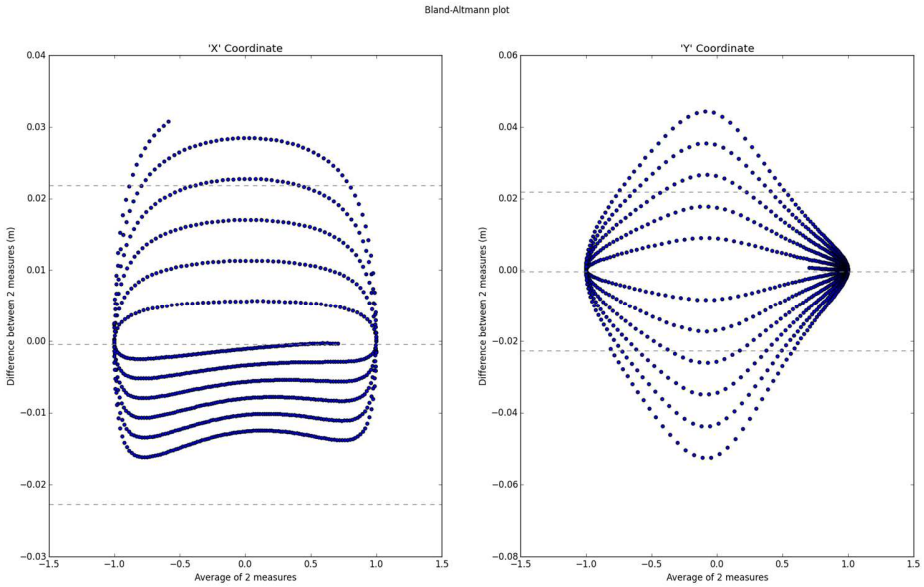

Fig. 9: Bland-Altman plot for *x* and *y* coordinates of point P2 for the slider crank simulation

	Intraclass Correlation	
	Consistency	Absolute Agreement
‘x’ coordinate	0.998	0.998
‘y’ coordinate	0.998	0.998

Tab. 4: Intraclass Correlation Coefficients for slider crank simulation

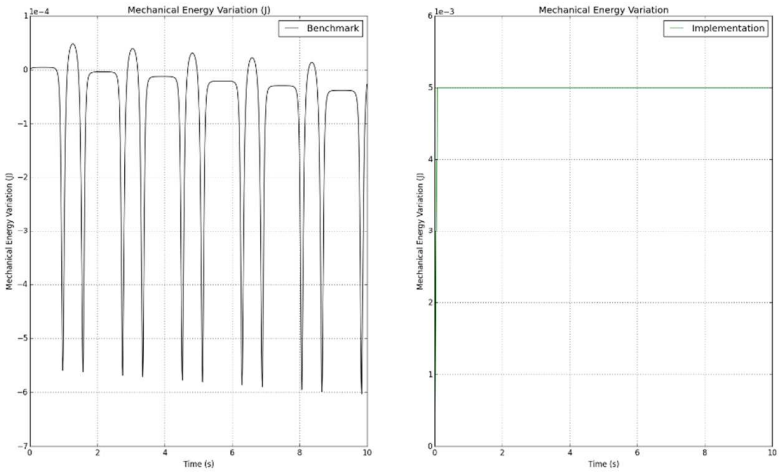

Fig. 10: Variation of the system mechanical energy for slider crank simulation

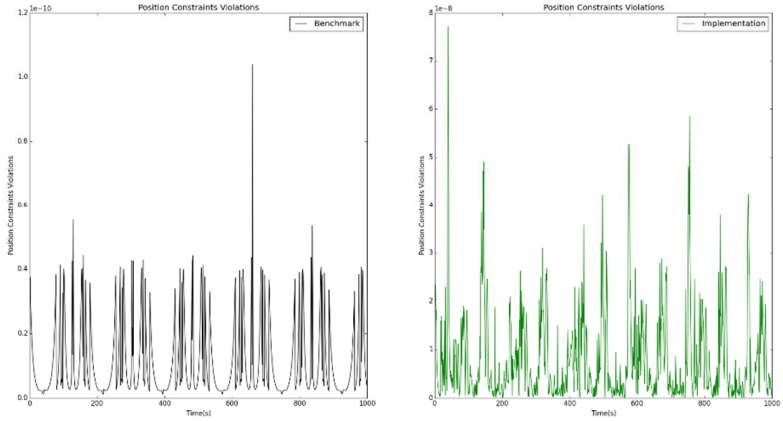

Fig. 11: Constraints violations for slider crank simulation

	Benchmark	Simulation
Programming Language	C++	Python
Computer	Intel Core2Duo @ 3.16GHz	Intel® Core i5 CPU 750 @ 3.67 Ghz - 12 GB
Number of simulations	NA	1.0E3
Average CPU Time (s)	0.328	2.21
Integrator	trapezoidal rule	Adams / BDF
Maximum admissible error:	1.0E-9	1.5E-8
Time step duration (s)	1.0E-3	Variable
Number of time steps	-	2430
Maximum number of iterations	1.2E1	4.5E1

Tab. 5: Computer specifications, integrator parameters and simulation time for the slider crank simulation

4. Discussion

Fully Cartesian Coordinates Formulation introduces a novel concept of a generic rigid body, which considers a predetermined kinematic structure composed by one point and unit vectors. Despite using the same kinematic elements as the Natural Coordinates Formulation, the body structure in FCCF considers only one point located on its center of mass instead of different relevant points. This approach avoids the need of body-specific constraint equations, approximating it from the RPCF. However, when compared with the Cartesian Coordinates, FCCF presents the advantage of using only Cartesian coordinates, eliminating terms dependent of angular variables that are in general more computational expensive.

In planar models, the rigid body in FCCF can be represented by one point and one vector to describe respectively the body translation and orientation. As this new structure implies that the body coordinates are independent from the adjacent ones, no points are shared. Therefore, when compared to the NCF, FCCF requires an explicit definition of joints, resulting in a larger number of generalized coordinates. Nevertheless, the problems related with the use of implicit joints in NCF are avoided, namely the appearance of coupled matrices and the impossibility of computing directly the reaction forces from the equations of motion. Moreover, since the coordinates of a given point can be described using the generalized coordinates and a constant transformation matrix (**C**), the calculation of the system relevant points is more computational efficient (e.g. joints, points of application of force, etc.), not requiring an update of the rotation matrix for each time frame as in RPCF. This simplicity in points calculation also reflects in the kinematic modelling of the system, since FCCF generates linear or quadratic constraint equations and their contributions to the Jacobian matrix ($\mathbf{\Phi_q}$) and right-hand side vectors ($\boldsymbol{\upsilon}$ and $\boldsymbol{\gamma}$) are also constant, linear or quadratic.

Another important aspect of this formulation is the definition of the system mass matrix. If the origin of each local reference frame is set at the body center of mass and its vectors are orientated with body principal axis of inertia, then the mass and principal moments of inertia are directly the entries of the mass matrix. Moreover, the mass matrix is constant, diagonal and easy to define, in opposition to NCF, in which the contribution of each matrix entry need to be calculated considering the body inertial parameters and the location of the generalized coordinates in relation to its CoM.

In order to validate the FCCF, this formulation was implemented in an in-house software developed in Python and tested through the analysis of different mechanisms. No problems were found regarding the system modelling, as well as the inverse kinematic and dynamic analyses did not presented problems related with the convergence of the method. Its performance was also analyzed by applying it in a forward dynamic simulation of two simple mechanisms, a simple planar pendulum and a planar slider crank, and its results compared with benchmark data.

In the pendulum example, the simulation did not present convergence-related issues for the range of Baumgarte coefficients tested ($0 \leq \alpha = \beta \leq 100$]). An analysis of the coordinates correlation between the simulation and benchmark outputs indicates an excellent score both for consistency and absolute agreement. This agreement is contradicted by the Bland-Altman plot, which shows that some values exceed the limits of agreement. However, this limit, which is dependent of the standard deviation of the differences, presents a very low value (1.5E-4 m) not having a physical meaning. Regarding mechanical energy variation, benchmark data shows a periodic behavior while simulation data increases over time. However, the magnitude of the variation in both cases is smaller than 5E-5 and therefore negligible, as expected since no dissipative terms were included in the simulation. The analysis of the evolution of the constraints violations during the simulation time also shows a different pattern. The constraints violation for the FCCF increased along the time with a maximum value of 4E-6 while benchmark data shows a very regular pattern with a maximum value of 4E-10. These variations can result from the differences on the integrator parameters, namely the tolerance errors, integration algorithm, time step, among others. For the methodology implemented in this work the default values from the *scipy.integrate.odeint* were used.

Contrarily to the pendulum example, the slider crank presented some convergence issues in the forward dynamic simulations. For a set of Baumgarte coefficients, the system after running some time started to behave like an erratic pendulum, with the point $\mathbf{P}_3$ fixed in the same position as the point $\mathbf{P}_1$. Results showed that values of α and β between 0 and 40 enable the convergence of the method, presenting, however, the behavior above described. For an α and β equal to 45, no convergence was achieved. Finally, for values of α and β higher than 50, no problems were found in the convergence of the simulation, resulting in the expected outcome. For this range of coefficients, the simulation results achieved an excellent agreement with the benchmark data. As in the pendulum case, the Bland-Altman plot also indicate the existence of differences higher than the limits of agreement in some points. An analysis of these variations allows to observe an increase of the differences with the

simulation time, achieving values on the order of 0.03m to x and 0.05m to y in the end of the simulation. As expected, the variation of the mechanical energy for the slider crank presents also small values (6E-3 J), since no dissipative terms were considered. The constraints violations for the RCCF does not show an increase with the simulation time, as in the pendulum example, reaching a maximum value of approximately 5E-8, inferior the tolerance error.

The computational times in the present work were calculated based on 1000 simulations. The increase in the complexity in the second example resulted in a significant increase in the computational time of approximately 514%. When compared with the benchmark data, the results show a decrease in the computational time for the pendulum (FCCF: 0.36 s; Benchmark: 0.53 s) and an increase for the slider crank (FCCF: 2.21s; Benchmark: 0.328 s). However, no conclusions can be taken between the two formulations, since the programming languages are substantially different and the computer used in the simulations do not present similar specifications to the ones used in the benchmark problem. The computational outcomes for the two examples shows also an increase in the number of iteration per time step when compared with the benchmark. These differences are not directly comparable, since the integrator is not the same and the benchmark considered a fixed time step. It is important to mention also that no speed optimization packages like *Numba* or *Cython* were used in this work.

5. Conclusions

This work aimed the development of an alternative multibody formulation, which would avoid the disadvantages of the common methodologies. The proposed formulation, Fully Cartesian Coordinates, presents a new concept of rigid body that is not body-specific dependent as in the Natural Coordinates, but it is defined recurring exclusively to cartesian coordinates, through the use of points and vectors. This fact eliminates the use of angular variables, such as in the Cartesian Coordinates, reducing the computational complexity of the problem to solve, since the constraint equations and their contributions to the Jacobian matrix and right-hand side vectors of velocities and accelerations are constant, linear or quadratic.

The proposed formulation allowed an effective modelling and analysis of different mechanical systems. No convergence-related issues were found in the inverse dynamic analyses. The formulation performance was also evaluated by applying it in the forward dynamics simulation of two benchmark problems. The results showed an excellent agreement with the provided data allowing to conclude that the presented formulation is an effective and efficient option to model and perform dynamic analysis of rigid multibody systems with different topologies.

Acknowledgements

The authors would like to thank Fundação para a Ciência e Tecnologia and LAETA for its support (SFRH/BD/51574/2011 and UID/EMS/50022/2013).

References

[1] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, 1st ed. Phoenix: Springer, 2008.

[2] P. Nikravesh, *Computer-aided analysis of mechanical systems*, 1st ed. New Jersey,07632: Prentice Hall, 1988.

[3] G. Jalon and E. Bayo, *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge*. New York: Springer Verlag, 1993.

[4] P. Gameiro and M. Tavares da Silva, "A systematic approach to the simulation of multibody systems with natural coordinates," in *Multibody Dynamics*, 2005.

[5] S. Uhlar and P. Betsch, "A rotationless formulation of multibody dynamics: Modeling of screw joints and incorporation of control constraints," *Multibody Syst. Dyn.*, vol. 22, no. 1, pp. 69–95, 2009.

[6] C. M. Pappalardo and D. Guida, "On the Lagrange multipliers of the intrinsic constraint equations of rigid multibody mechanical systems," *Arch. Appl. Mech.*, vol. 88, no. 3, pp. 419–451, 2018.

[7] M. Tavares da Silva, "Human Motion Analysis using Multibody Dynamics and Optimization Tools,"

Universidade Técnica de Lisboa - Instituto Superior Técnico, 2003.

[8] J. Baumgarte, “Stabilization of constraints and integrals of motion in dynamical systems,” *Comput. Methods Appl. Mech. Eng.*, vol. 1, no. 1, pp. 1–16, 1972.

[9] IFToMM Technical Committee for Multibody Dynamics, “Library of Computational Benchmark Problems.” .

[10] J. M. Bland, D. G. Altman, J. Martin Bland, and D. G. Altman, “Statistical methods for assessing agreement between two methods of clinical measurement.,” *Lancet*, vol. 1, no. 8476, pp. 307–310, 1986.

[11] P. E. Shrout and J. L. Fleiss, “Intraclass correlations: uses in assessing rater reliability.,” *Psychol. Bull.*, vol. 86, no. 2, pp. 420–8, Mar. 1979.