

Multibody approach Matlab GUI for kinematic and dynamic analysis of planar mechanisms

GUI en Matlab de enfoque multi-cuerpos para el análisis cinemático y dinámico de mecanismos planos

Eugenio Yime Rodríguez¹, Javier Roldán Mckinley¹, Milton Coba Salcedo²

1: Grupo DIMER-Programa de Ing. Mecánica, Universidad del Atlántico, Barranquilla, Colombia

2: Grupo IMTEF-Programa de Ing. Mecánica, Universidad del Atlántico, Barranquilla, Colombia

eugenioyime@mail.uniatlantico.edu.co

javierroldan@mail.uniatlantico.edu.co

miltoncoba@mail.uniatlantico.edu.co

Abstract— The description and use of a Matlab programmed software with graphical interface created in GUIDE are presented in this paper. The report begins with the introduction of multibody approach fundamentals for planar mechanisms. Vector description of rigid elements, revolute and prismatic kinematic pairs, and rotatory or linear drivers (actuators) are presented with the set of the restriction equations. Numerical solution of the equations for both kinematics and dynamics of the mechanism elements is made with the Newton-Raphson algorithm. Sliding-crank four-link planar mechanism case study is used for familiarizing the user with the basic level multibody approach software that is introduced in the paper. The use of the software is painstakingly illustrated for the sliding-crank mechanism at every stage: definition of elements, joints, and actuators, followed by motion specification and the final visualization of the simulation results. A satisfactory validation is made by comparing the software findings against a SolidWorks simulation.

Key Word — Multibody approach, Planar mechanisms, GUI, Matlab-GUIDE, Mechanism and machine design

Resumen— La descripción y uso de un software desarrollado en Matlab con una interfaz gráfica creada en GUIDE son presentados en este artículo. El reporte inicia con la introducción de fundamentos de enfoque multicuerpos para mecanismos planos. La descripción vectorial de elementos rígidos, pares cinemáticos prismático y de revoluta, y actuadores (drivers) lineales y rotatorios son presentadas con el establecimiento de las ecuaciones de restricción. La solución numérica de las ecuaciones cinemáticas y dinámicas se hace por medio del algoritmo Newton-Raphson. El mecanismo de cuatro barras manivela-corredera es el caso de estudio para familiarizar el usuario con el software multicuerpos de nivel básico que se presente en el artículo. El uso del software es meticulosamente ilustrado para el mecanismo manivela-corredera en cada etapa: definición de elementos, juntas y actuadores; seguido de la especificación del movimiento para visualización final de los resultados. Se hizo una

satisfactoria comparación de los resultados obtenidos con el software con los obtenidos de una simulación con SolidWorks.

Palabras clave— Enfoque multicuerpos, Mecanismos planos, GUI, Matlab-GUIDE, Diseño de mecanismos y máquinas

I. INTRODUCTION

“Mechanisms” are defined as mechanical systems comprised by rigid bodies that are linked through joints, leading to a system with at least one degree-of-freedom [1], [2], [3]. Mechanisms main feature is the transformation of an input motion, usually under an uniform pattern either linear or rotational, into a complex output motion that can describe planar and/or spatial trajectories [4], [5]. Mechanisms play an important role in Mechanical Engineering, since they are needed for the right motion transformation in mechanical systems. In example, the internal combustion engines inside the vehicles use a four-link mechanism to transform the linear input motion from the cylinder into a rotatory continuous output motion that is transmitted to the wheels. Another example is the sewing machine, in which a four-link mechanism is used to convert rotatory input motion from an electric motor, into linear output motion for the alternate motion of the needle.

As from the educational stand-point, the most of the Mechanical Engineering programs in the world include at least one course that deals with the study and analysis of mechanisms. In Colombia, for instance, the Colombian association of electrical, mechanical and related engineering majors-ACIEM [6] (ACIEM because of its Spanish translation Asociación Colombiana de Ingenieros Eléctricos, Mecánicos y Afines) states that Mechanisms is a core subject for the Mechanical Engineering programs in Colombia. On the other

hand, teaching and learning tools for this subject is actively changing in accordance with technological advancements. In the 70s, by-hand graphical approach for position, velocity and acceleration kinematic analysis was used. Later, during the 80s, algebraic methods that are based on vector analysis and complex numbers were introduced. More recently, it has been emphasized the use of computer for the programming of the algebraic solutions, so that the solution might be visualized not only for table or a plot, but for the simulation of a whole work-cycle.

Nowadays, the computer based mechanisms analysis has a solid background on the multibody analysis approach [7], [8], [9], [10], [11], [12], [13] [14], [15], thus allowing the kinematic and dynamic analysis of mechanisms in specialized CAD/CAE computer packages, such as ProEngineer, SolidWorks, SolidEdge and Adams, among the most widely used licensed software. Other licensed mechanisms analysis software packages are SAM [16] and Universal Mechanisms [17]. There are other two types of software available for the users, in general, besides the above mentioned packages. GIM [18] and Linkage [19] are free software for the mechanisms designer. There is also the open code software in mechanisms that are, basically, the result of applied research in computational mechanics. These software require that the user acquire some knowledge not only about the subject area but also about programming, in order to perform the modifications that are suitable to his particular research needs. Some examples of open code software in mechanisms are Ch Mechanism Toolkit [20], Mesa Verde [21] y MBDyn [22].

This paper presents a software for the kinematic and dynamic analysis of planar mechanisms. The programming was based on the multibody mechanical systems approach. Multibody fundamentals are presented trying to synthesize knowledge in the computational analysis of mechanisms. An illustrative example for using the software is explained for planar mechanisms with rotational and prismatic joints. From the user standpoint, it is highlighted that the user does not need experience in drawing mechanical elements. The mechanism to be analyzed can be built by making click and defining lengths inside textboxes. The first part of the report is an introduction to the mathematical fundamentals. Later, the relation between the mathematical approach and the programming methods are presented; in order to provide the user with the indications for further modification of the programming code, if required. An explanation of the GUI is next, followed by the study case. Results are compared against findings that were obtained using another computational tool.

II. FUNDAMENTALS OF MULTIBODY ANALYSIS FOR PLANAR MECHANISMS [7]

In computational analysis for mechanisms based on multibody approach, a coordinate system with respect to an inertial reference frame is used for defining the position of each body. If the reference frame moves with its body, then the frame is

called a local reference system. Fig. 1(a) depicts a four-link mechanism with four revolute joints whose movable elements position has been defined by a reference system $\{x_i, y_i\}$ at its center of mass, and a position angle ψ_i . In contrast, Fig. 1(b) shows the joint angles label type that is used in Robotics and Mechanisms and Machine Theory; by knowing the joint angles and the link distances is also possible to know the behavior of any element of the mechanism at any moment. Due to the greater amount of variables used for describing the system in Figure 1(a), some equations (usually linear) must be established to describe the manner how all these variables are related. The stated equations must propose a system with independent variables as the same degrees-of-freedom (DOF) of the system.

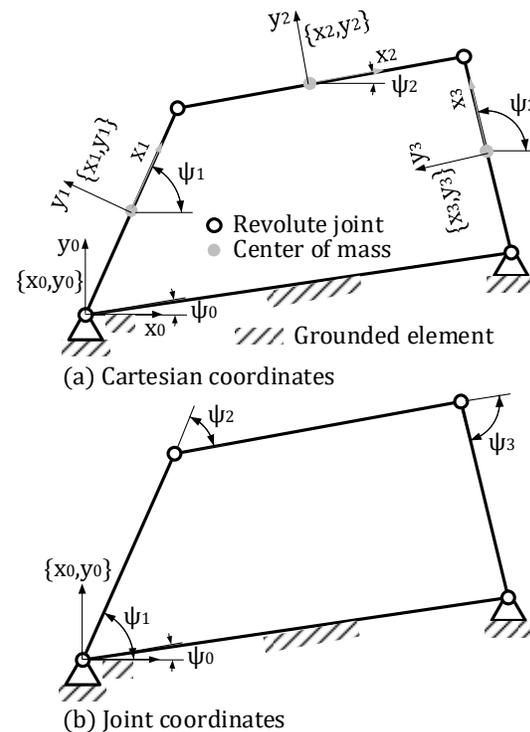


Figure 1. Different coordinates for a four-linkage planar mechanism.

At least nine Cartesian generalized coordinates are needed for the four-link mechanism in Fig. 1(a), in order to describe the three movable elements of the system, they are elements 1, 2 and 3. Three coordinates are needed for each movable element: abscissa x_i , ordinate y_i , and the orientation angle ψ_i . The system possesses one DOF, then eight equations must be written to relate the variables. In this multibody approach with generalized coordinates, restriction equations are directly linked to the joint connecting elements. Fig. 2 shows the first joint of the mechanism in Fig. 1(a). In this joint, point P is a common point for both ground and element 1, and the coordinates, Φ_1 and Φ_2 , of point P in the fixed coordinate system are given by equations 1 and 2, respectively, where all variables in these equations are shown in Fig. 2 and l_1 is the length of the element with its center of mass located at $l_1/2$.

In multibody approach, the equations that relate the variables are called motion restriction equations, and they are written in compact format as a vector named restrictions vector, labelled by Φ . If \mathbf{q} is the vector that represent the generalized coordinates, then the restriction equation yields a single vector equation as in equation 3, where t is the elapsed time. In order to facilitate linear algebra operations in multibody approach for mechanisms analysis, it is desirable that the restriction vector be a square matrix. The square matrix is comprised by two vectors that represent two different types of restrictions in the system. The drive restrictions vector depends on the DOF of the system and is represented by the vector Φ^D . The joint restriction vector, labelled by Φ^K , targets the joint types (revolute, prismatic, screw, etc.) of the system, to generate the matrix shown in equation (4).

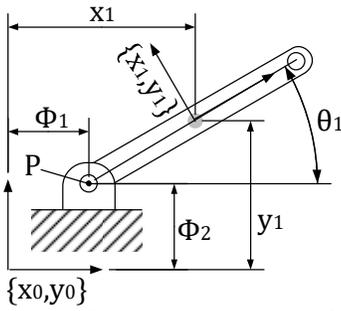


Figure 2. Generalized coordinates for a single element with revolute joint.

$$\Phi_1 = x_1 - \frac{1}{2} \cos \theta_1 = 0 \quad (1)$$

$$\Phi_2 = y_1 - \frac{1}{2} \sin \theta_1 = 0 \quad (2)$$

$$\Phi(\mathbf{q}, t) = \mathbf{0} \quad (3)$$

$$\Phi(\mathbf{q}, t) = \begin{bmatrix} \Phi^K \\ \Phi^D \end{bmatrix} = \mathbf{0} \quad (4)$$

III. DRIVE AND JOINT RESTRICTIONS

The most common types of joints for planar mechanisms are the rotational and the prismatic joints. As for actuators, in a similar manner, rotational and linear motors are the most used input motion devices. Fig. 3 presents a rotational joint at point P that allows relative rotational motion between bodies i and j , and each body has a vector \mathbf{r}_i from the global system $\{x_0, y_0\}$ origin to the body center of gravity CG $_i$. Point P with respect to the local systems $\{x_i, y_i\}$ and $\{x_j, y_j\}$ is described by using the local vectors \mathbf{s}_i^{P} and \mathbf{s}_j^{P} for the bodies i and j , respectively. The equation that describes the kinematic restriction for the revolute joint is given by equation (5), where Φ_{rot}^K is the restriction vector for the revolute joint, and \mathbf{A}_i is the planar rotation matrix of the rigid body given by equation (6), where θ_i is the orientation angle of the body i .

$$\Phi_{rot}^K = \mathbf{r}_i + \mathbf{A}_i \mathbf{s}_i^{P} - \mathbf{r}_j - \mathbf{A}_j \mathbf{s}_j^{P} = \mathbf{0} \quad (5)$$

$$\mathbf{A}_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \quad (6)$$

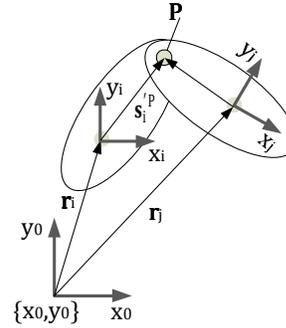


Figure 3. Vector analysis nomenclature for two bodies with relative rotational motion in a revolute joint.

The substitution of equation (5) into equation (6) yields equation (7), where $r_{i,x}$ is the x component of the vector \mathbf{r}_i , $r_{i,y}$ is the y component of the vector \mathbf{r}_i , $s_{i,x}^{P}$ is the x component of the vector \mathbf{s}_i^{P} , and $s_{i,y}^{P}$ is the y component of the vector \mathbf{s}_i^{P} . From equation (7) there will be obtained two non-linear equations, one for each component: x axis and y axis.

$$\Phi_{rot}^K = \begin{bmatrix} r_{i,x} + \cos \theta_i s_{i,x}^{P} - \sin \theta_i s_{i,y}^{P} - r_{j,x} - \cos \theta_j s_{j,x}^{P} + \sin \theta_j s_{j,y}^{P} \\ r_{i,y} + \sin \theta_i s_{i,x}^{P} + \cos \theta_i s_{i,y}^{P} - r_{j,y} - \sin \theta_j s_{j,x}^{P} - \cos \theta_j s_{j,y}^{P} \end{bmatrix} = \mathbf{0} \quad \dots(7)$$

Fig. 4 depicts the vector analysis labelling for the prismatic joint that allows only relative translational motion between consecutive elements i and j . Similar to the rotational joint, there are two position vectors, \mathbf{r}_i and \mathbf{r}_j , from the origin of the global reference frame $\{x_F, y_F\}$ to the center of mass of each element. There are also the joint position vectors with respect to the local reference system of each body, they are vectors \mathbf{s}_i^{P} and \mathbf{s}_j^{P} . Parallel local reference systems $\{V_i, V_i^\perp\}$ and $\{V_j, V_j^\perp\}$ for bodies i and j , respectively, establish the reference system for the joint. From the labelling in Fig. 4, the prismatic joint is defined as the connection that allows only motion along the line that is perpendicular to axes V_i^\perp and V_j^\perp and connects points \mathbf{P}_i and \mathbf{P}_j . The equation that kinematically relates the vectors defined in Fig. 4 is given by equation (8), where $\mathbf{d}_{i,j}$ is the vector from the origin of local reference system $\{V_i, V_i^\perp\}$ to the origin of the local reference system $\{V_j, V_j^\perp\}$. The restriction vector for the linear motion of the prismatic joint, Φ_{lin}^K , is given by equation (9).

$$\mathbf{d}_{i,j} = \mathbf{r}_j + \mathbf{A}_j \mathbf{s}_j^{P} - \mathbf{r}_i - \mathbf{A}_i \mathbf{s}_i^{P} \quad (8)$$

$$\Phi_{lin}^K = \begin{bmatrix} (v_i^\perp) \mathbf{d}_{i,j} \\ (v_j^\perp) \mathbf{d}_{i,j} \end{bmatrix} = \mathbf{0} \quad (9)$$

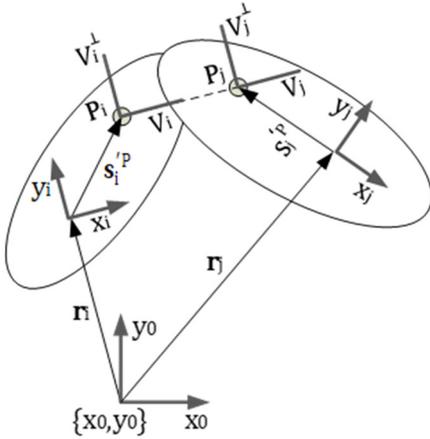


Figure 4. Vector analysis nomenclature for two bodies with relative linear motion in a prismatic joint.

IV. ACTUATORS' RESTRICTIONS

Fig. 5 presents the vector labelling for a rotational actuator that turns in about point P, that is used for stating the restriction equation (10) under the condition of the rotational driver Φ_{rot}^D , where θ_i and θ_j are the position angles of links i and j, respectively, and $C(t)$ is the function that defines the pair motion with respect to time. For the linear actuator whose labelling is depicted in Fig. 6, the restriction equation is given by equation (11), where the restriction condition is defined as Φ_{lin}^D , \mathbf{d}_{ij} is the vector from the origin of local reference system $\{V_i, V_i^\perp\}$ to the origin of the local reference system $\{V_j, V_j^\perp\}$, origins separated a distance d_{ij} along the vector \mathbf{d}_{ij} , and $\text{dot}(\mathbf{d}_{ij}^T, \mathbf{d}_{ij})$ is the dot product between the vectors.

$$\Phi_{rot}^D = \theta_j - \theta_i - C(t) = 0 \tag{10}$$

$$\Phi_{lin}^D = \text{dot}(\mathbf{d}_{ij}^T, \mathbf{d}_{ij}) - C(t)^2 = 0 \tag{11}$$

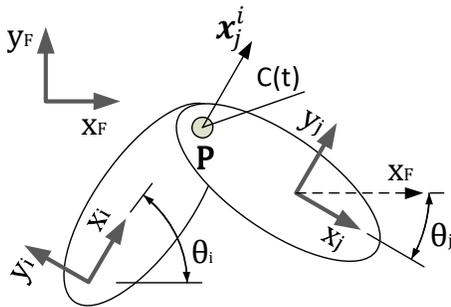


Figure 5. Vector analysis nomenclature for a rotational actuator.

V. MECHANISMS ANALYSIS USING KINEMATIC AND DYNAMIC MULTIBODY APPROACH [7]

Multibody analysis approach consists in stating a restriction vector that is later derived with respect to time. Next, position, velocity and acceleration kinematic analyses are made from

the generated equations, and subsequently, these results are used for the dynamic analyses of acting forces and torques, and the reactions at the pairs (joints). The latter is the inverse dynamic analysis.

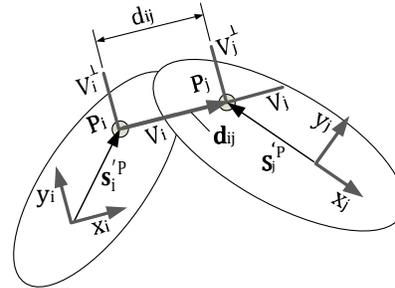


Figure 6. Vector analysis nomenclature for a linear actuator.

A. Position analysis.

Multibody position analysis lies in finding the numerical solution to the restriction vector in equation (3) for a determined time instant. The physical interpretation is that there is knowledge of the approximated position values for each body, and it is desired obtaining the real positions of the mechanism so that the restriction vector reaches a value closer to zero, within the acceptable error margin. The Newton-Raphson method is widely used in this problem, which can be stated as in equation (12) where q_i is the value of the generalized coordinate at the i-th position, and q_{i+1} is the value of the generalized coordinate at the (i+1)-th position, after a time t has elapsed, where the term Φ_q indicates the partial derivative of the restriction vector Φ with respect to q position of the mechanism, it is to say $\frac{\partial \Phi}{\partial q}$.

$$q_{i+1} = q_i - \Phi_q^{-1}(q_i, t)\Phi(q_i, t) \tag{12}$$

The Newton-Raphson method is an iterative routine used to recalculate the new system position from the actual value. As any other numerical method, convergence criteria are to be established. An usual condition is the infinite norm for the vector Φ , that cannot be greater to an error defined very close to zero. A non-convergence indicator is taken as exceeding a maximum number of iterations.

B. Velocity analysis.

Once the position analysis is done, the velocity analysis arises by taking derivative with respect to time of the restriction condition, leading to the relation of the bodies' velocities in a vector equation that is used to recalculate the velocity of each body. When taking derivatives, the restriction vector Jacobian vector, Φ_q , appears as the combination of the partial derivatives in each restriction element. Table 1 summarizes the partial derivative terms according to the restriction type, that are used to build the Jacobian vector; where the matrix terms \mathbf{B}_i , \mathbf{A}_i and \mathbf{R} are given by equation (13). After taking derivative of the restriction vector, it yields equation (14),

where from it is solved for the velocity vector, $\dot{\mathbf{q}}$, as shown in equation (15) (Haug [7]), with $\mathbf{v} = -\Phi_{\mathbf{t}}$ is the partial derivative of the restriction vector with respect to time, also known as the vector of the velocity analysis of the system, to be found for each type of actuator.

$$\mathbf{B}_i = \begin{bmatrix} -\sin\theta_i & -\cos\theta_i \\ \cos\theta_i & -\sin\theta_i \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{A}_i = \mathbf{R}\mathbf{A}_i \quad (13)$$

$$\dot{\Phi}(\mathbf{q}, t) = \Phi_{\mathbf{q}}\dot{\mathbf{q}} + \Phi_{\mathbf{t}} = 0 \quad (14)$$

$$\dot{\mathbf{q}} = \Phi_{\mathbf{q}}^{-1}\mathbf{v} \quad (15)$$

Restriction type	$\frac{\partial\Phi}{\partial\mathbf{r}_i}$	$\frac{\partial\Phi}{\partial\theta_i}$
$\Phi_{\text{rot}}^{\text{K}}$	$\mathbf{1}_{2 \times 2}$	$\mathbf{B}_i \mathbf{s}_i^{\text{P}}$
$\Phi_{\text{lin}}^{\text{K}}$	$\begin{bmatrix} -\mathbf{v}_i^{\text{T}} \mathbf{B}_i^{\text{T}} \\ \mathbf{0}_{1 \times 2} \end{bmatrix}$	$\begin{bmatrix} -\mathbf{v}_i^{\text{T}} \mathbf{A}_i^{\text{T}} (\mathbf{r}_j - \mathbf{r}_i + \mathbf{A}_j \mathbf{s}_j^{\text{P}}) \\ -\mathbf{v}_i^{\text{T}} \mathbf{A}_i^{\text{T}} \mathbf{A}_j \mathbf{v}_j \end{bmatrix}$
$\Phi_{\text{rot}}^{\text{D}}$	$\mathbf{0}_{1 \times 2}$	-1
$\Phi_{\text{lin}}^{\text{D}}$	$-2\mathbf{d}_{ij}^{\text{T}}$	$-2\mathbf{d}_{ij}^{\text{T}} \mathbf{R} \mathbf{A}_i \mathbf{s}_i^{\text{P}}$
Restriction type	$\frac{\partial\Phi}{\partial\mathbf{r}_j}$	$\frac{\partial\Phi}{\partial\theta_j}$
$\Phi_{\text{rot}}^{\text{K}}$	$-\mathbf{1}_{2 \times 2}$	$-\mathbf{B}_j \mathbf{s}_j^{\text{P}}$
$\Phi_{\text{lin}}^{\text{K}}$	$\begin{bmatrix} \mathbf{v}_i^{\text{T}} \mathbf{B}_i^{\text{T}} \\ \mathbf{0}_{1 \times 2} \end{bmatrix}$	$\begin{bmatrix} \mathbf{v}_i^{\text{T}} \mathbf{A}_i^{\text{T}} \mathbf{A}_j \mathbf{s}_j^{\text{P}} \\ \mathbf{v}_i^{\text{T}} \mathbf{A}_i^{\text{T}} \mathbf{A}_j \mathbf{v}_j \end{bmatrix}$
$\Phi_{\text{rot}}^{\text{D}}$	$\mathbf{0}_{1 \times 2}$	1
$\Phi_{\text{lin}}^{\text{D}}$	$2\mathbf{d}_{ij}^{\text{T}}$	$2\mathbf{d}_{ij}^{\text{T}} \mathbf{R} \mathbf{A}_j \mathbf{s}_j^{\text{P}}$

Table 1. Summary of equations used to construct the restriction Jacobian vector.

From the definition of the restrictions, it is apparent that the \mathbf{v} vector only exists for the actuators, since only the definitions of the type of vectors contain the time in an explicit manner. A summary of vector \mathbf{v} equations is given in Table 2.

Restriction type	Component
$\Phi_{\text{rot}}^{\text{D}}$	$\dot{C}(t)$
$\Phi_{\text{lin}}^{\text{D}}$	$2C(t)\dot{C}(t)$

Table 2. Component of \mathbf{v} vector according with the restriction type.

C. Acceleration analysis.

The acceleration analysis is stated from the second derivative of the restriction vector, to yield equation (16) (Haug [7]), where it has been considered that the Jacobian $\Phi_{\mathbf{q}}$ does not depend on time but only of the position, therefore $\Phi_{\mathbf{q}\mathbf{t}} = \mathbf{0}$. From equation (16), the acceleration vector $\boldsymbol{\gamma}$, is given by equation (17) and the acceleration velocity vector of the system yields (18). Definitions of $\boldsymbol{\gamma}$ vector for the most used plane joints and actuators are shown in Table 3. Vector \mathbf{g}_i is given by equation (19).

$$\ddot{\Phi}(\mathbf{q}, t) = \Phi_{\mathbf{q}}\ddot{\mathbf{q}} + (\Phi_{\mathbf{q}}\dot{\mathbf{q}})_{\mathbf{q}}\dot{\mathbf{q}} + \Phi_{\mathbf{tt}} = \mathbf{0} \quad (16)$$

$$\boldsymbol{\gamma} = -(\Phi_{\mathbf{q}}\dot{\mathbf{q}})_{\mathbf{q}}\dot{\mathbf{q}} - \Phi_{\mathbf{tt}} \quad (17)$$

$$\ddot{\mathbf{q}} = \Phi_{\mathbf{q}}^{-1}\boldsymbol{\gamma} \quad (18)$$

$$\mathbf{g}_i = \mathbf{A}_j \mathbf{R} \mathbf{s}_j^{\text{P}} (\theta_j - \theta_i)^2 - \mathbf{R}^{\text{T}} (\mathbf{r}_j - \mathbf{r}_i) \dot{\theta}_i^2 - 2(\dot{\mathbf{r}}_j - \dot{\mathbf{r}}_i) \dot{\theta}_i \quad (19)$$

Restriction type	Vector $\boldsymbol{\gamma}$
$\Phi_{\text{rot}}^{\text{K}}$	$\mathbf{A}_i \mathbf{s}_i^{\text{P}} \dot{\theta}_i^2 - \mathbf{A}_j \mathbf{s}_j^{\text{P}} \dot{\theta}_j^2$
$\Phi_{\text{lin}}^{\text{K}}$	$\begin{bmatrix} -\mathbf{v}_i^{\text{T}} \mathbf{A}_i^{\text{T}} \mathbf{g}_i \\ 0 \end{bmatrix}$
$\Phi_{\text{rot}}^{\text{D}}$	$\ddot{C}(t)$
$\Phi_{\text{lin}}^{\text{D}}$	$-2\mathbf{d}_{ij}^{\text{T}} \dot{\mathbf{d}}_{ij} - 2\mathbf{d}_{ij}^{\text{T}} (\mathbf{A}_i \mathbf{s}_i^{\text{P}} \dot{\theta}_i^2 - \mathbf{A}_j \mathbf{s}_j^{\text{P}} \dot{\theta}_j^2) + 2\dot{C}^2(t) + 2C(t)\ddot{C}(t)$

Table 3. Definitions of acceleration vector $\boldsymbol{\gamma}$.

D. Dynamic analysis.

Multibody approach for dynamic analysis is based on generalized coordinates (Haug [7]) in order to set the dynamic equation (20), where \mathbf{M} is a diagonal matrix comprised by each of the generalized mass matrixes of the bodies of the system, $\Phi_{\mathbf{q}}^{\text{T}}$ is the traspose of the Jacobian matrix of the restrictions, $\boldsymbol{\lambda}$ is the Lagrange multipliers vectors related with the joints or restrictions, and \mathbf{Q} is the generalized forces vector acting on the gravity center of each body. For the inverse dynamic analysis the \mathbf{Q} vector only contains the Coriolis term from cross($-\boldsymbol{\omega}$, $\mathbf{I}\boldsymbol{\omega}$), where $\boldsymbol{\omega}$ is the angular velocity vector of the body, \mathbf{I} is the inertia matrix with respect to the gravity center of the body, and ‘‘cross’’ defines the cross product operation. For the inverse dynamic analysis, it is solved for the Lagrange multipliers vector from equation (20). In the direct dynamic analysis, the motion equations of the system are to be found from the forces on the actuators, in order to satisfy the relation established in equation (21), where the augmented matrix containing \mathbf{M} is square, and the system is formed by algebraic equations that set a differential equations system.

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^{\text{T}}\boldsymbol{\lambda} = \mathbf{Q} \quad (20)$$

$$\begin{bmatrix} \mathbf{M} & \Phi_{\mathbf{q}}^{\text{T}} \\ \Phi_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q} \\ \boldsymbol{\gamma} \end{bmatrix} \quad (21)$$

E. Algorithm for multibody analysis

The procedure for a multibody analysis for a system is summarized in the following algorithm.

- State an initial position for each mechanism element
- Set the system position vector \mathbf{q}_i
- Assign an initial value \mathbf{t}
- Use the Newton-Raphson approximation until there is convergence
- Calculate the velocity $\dot{\mathbf{q}}$
- Calculate the acceleration $\ddot{\mathbf{q}}$
- Find the Lagrange multipliers vector $\boldsymbol{\lambda}$
- Solve the differential equations system in equation (22)

- Projects the new position for the next time instant t_{i+1} , equation (23), and position in equation (24)
- Repeat from second step until the end position is reached

$$t_{i+1} = t_i + \Delta t \tag{22}$$

$$q_{i+1} = q_i + \dot{q}_i \Delta t + \frac{\ddot{q}_i \Delta t^2}{2} \tag{23}$$

VI. PLANAR MECHANISM DESCRIPTION USING MULTIBODY APPROACH

Each element of a mechanism has to be described for the multibody formulation by using the joint and actuator definitions. For rigid bodies are required the mass, inertia and initial position values. Fig. 7 presents a widely used mechanism known as the sliding-crank type, commonly found in car engines and sewing machines, among others. The example of the four-link mechanism used as an example, with the respective information is given in Table 4. First information needed is the number of elements. Grounded link or fixed frame is taken as zero element. Next, the mass and the inertia of the rigid body are given, required for the dynamic analysis. The initial position of each rigid body is given next, by a three element vector that contains abscissa of the center of mass, ordinate of the center of mass, and orientation of the element given in radians, for each element. The mechanism in Figure 7 has three rotational joints and one translational joint, whose information is summarized in Table 5. The joints requires the specification of the connected elements and the coordinates of the contact point in each body local system, this is the s_i^p local vector. Only an actuator is needed, set as rotational type. Multibody information for the rotational actuator is presented in Table 6. The two bodies that the actuator is connecting are to be specified. An initial angle of the movable element, θ_0 , is needed, in addition to the motion function with its first and second derivatives.

Link	Mass (kg)	Center of mass initial position [m; m]	Initial position [rad]	Inertia (kg·m ²)
1	0.02806	[0, 0.05]	$\pi/2$	0.096
2	0.02753	[0.05528, 0.089443]	-0.4637	0.3713
3	0.33777	[178.89, 0]	0	0.1406

Table 4. Multibody information of four-link mechanism elements.

No	Type	Body i	Body j	i-th position	j-th position
1	Rotational	0	1	[0, 0]	[-0.05, 0]
2	Rotational	1	2	[0.05, 0]	[-0.1, 0]
3	Rotational	2	3	[0.1, 0]	[0, 0]
4	Translational	0	3	[0, 0, 0]	[0, 0, 0]

Table 5. Multibody information of mechanism revolute joints.

Type	Body i	Body j	θ_i (°)	θ_j (°)	F(t)	F'(t)	F''(t)
Rotational	0	1	0	0	$0.6283 * t + \pi/2$	0.6283	0

Table 6. Multibody information of rotational actuator in Fig. 7.

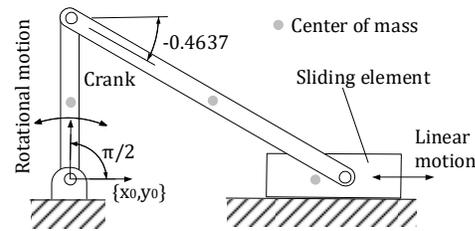


Figure 7. Sliding-crank type four-link mechanism.

VII. CODE STRUCTURE IN MATLAB

A data structure is created for the described mechanism from the multibody information to handle information of each element in an efficient manner. Table 7 presents the code lines for adding the element one. The kinematic pair, rotational in this case, is defined based on the code in Table 8. Next, the lines for defining the rotational actuator are in Table 9. Finally the simulation of motion type for the actuator is shown in Table 10. After the definition of elements, joints, actuator and motion, the function “inv_dynamics” is called, Table 11, in order to solve first the kinematic analysis for position, velocity and acceleration of the mechanism, and later performing the inverse dynamic analysis to obtain the reactions at the joints and the required forces and/or torques at the actuators. The function is called under the syntax $[q, qp, qpp, t, F] = \text{inv_dynamics}(\text{FourBar})$.

```

%% Slider-Crank mechanism creation
% First link
MBC.links(1).linknum = 1;
MBC.links(1).mass = 0.02806;
MBC.links(1).inertia = 0.096;
MBC.links(1).position = [0; 0.05; pi/2];
    
```

Table 7. Code for defining (adding) a rigid element to the system.

```

%% Slider-Crank mechanism creation
% First joint
MBC.joints(1).jointnum= 1;
MBC.joints(1).type = 'rotational';
MBC.joints(1).link_i = 0;
MBC.joints(1).link_j = 1;
MBC.joints(1).pos_i = [0; 0];
MBC.joints(1).pos_j = [-0.05; 0];
    
```

Table 8. Code for adding a revolute pair to the system.

```

%% Slider-Crank mechanism creation
% Traslational joint
MBC.joints(4).jointnum= 4;
MBC.joints(4).type = 'traslational';
MBC.joints(4).link_i = 0;
MBC.joints(4).link_j = 3;
MBC.joints(4).pos_i = [0; 0];
MBC.joints(4).pos_i.theta = 0;
MBC.joints(4).pos_j = [0; 0];
MBC.joints(4).pos_j.theta = 0;
    
```

Table 9. Code for adding a translational pair to the system.

```

%% Slider-Crank mechanism creation
FourBar.drivers(1).DriverNum = 1;
FourBar.drivers(1).type = 'rotational';
FourBar.drivers(1).link_i = 0;
FourBar.drivers(1).link_j = 1;
FourBar.drivers(1).theta_i = 0;
    
```

```
FourBar.drivers(1).theta_j = 0;
FourBar.drivers(1).function = '\0.6283*t+pi/2';
FourBar.drivers(1).Deriv1 = '\0.6283';
FourBar.drivers(1).Deriv2 = '\0';
```

Table 10. Code for adding a revolute actuator to the system.

```
% Slider-Crank mechanism creation
% Simulation parameters
FourBar.parameters.t_start = 0;
FourBar.parameters.t_end = 10.0;
FourBar.parameters.delta_t = 0.01;
FourBar.parameters.maxerror = 0.01;
FourBar.parameters.maxiter = 10;
FourBar.parameters.gravity = [0; -9.8];
```

Table 11. Code for defining simulation parameters to the system.

VIII. MATLAB GUI FOR MULTIBODY ANALYSIS OF PLANAR MECHANISMS

Although the code can be executed either from the Command Window or from the Run command from menu, a Matlab Graphical User Interface-GUI under GUIDE was designed for facilitating the use of the developed software. The GUI will mainly ease the data input, particularly for users with few experience with Matlab programming. Fig. 8 depicts the MMA GUI (Multibody Mechanisms Analysis) that was designed under the simplicity basis in order to facilitate the first time interaction of the users with the multibody approach for mechanism analysis.

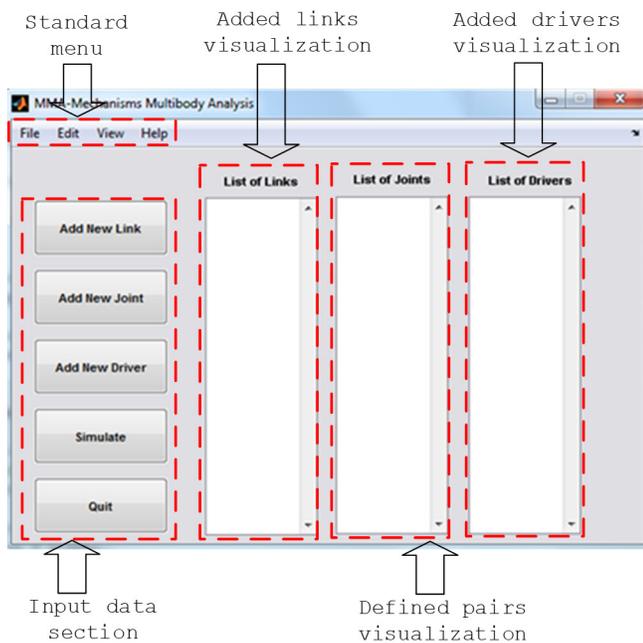


Figure 8. Matlab GUI panels.

There are five different panels that collect all push buttons, textboxes and lists. Fig. 9 shows the algorithm for using the interface. The example corresponding to the data summarized in Tables IV to VI for the sliding-crank mechanism in Figure 7 will be executed in the MMA GUI. The initial position

configuration in Table 4 is shown in Fig. 10. By clicking on the Add New Link button in the “Input data section” panel, the window “Add new link data” in Fig. 11 is displayed. The three movable elements 1 to 3 will be defined, as shown in Fig. 11 (a) through (c). The fixed reference system $\{x_0, y_0\}$ has been located at joint J_0 , Fig. 10. Information entered in Fig. 11 windows is taken from Table 4.

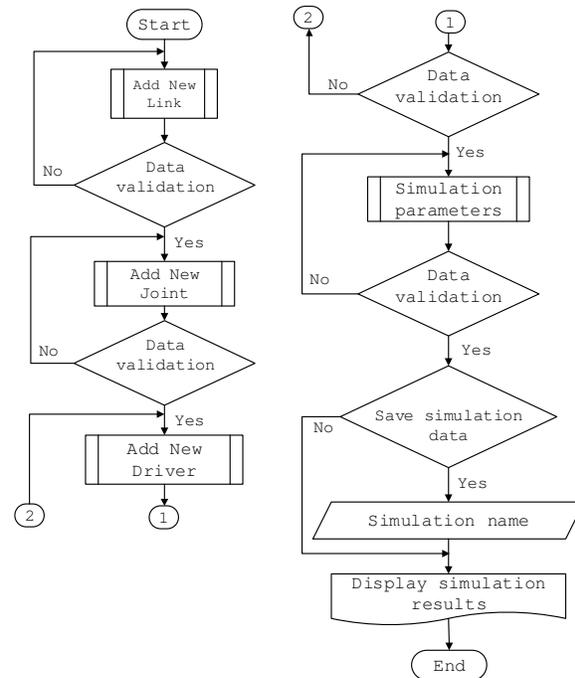


Figure 9. Algorithm for using the MMA GUI.

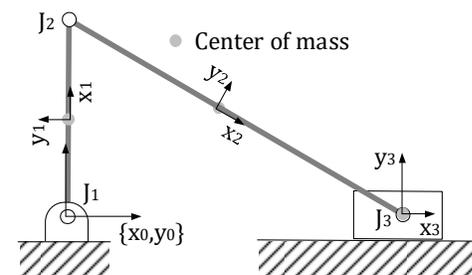


Figure 10. Initial configuration, joints and references systems of the sliding-crank mechanism for using the MMA GUI.

The joints are defined after defining the links, by clicking in the button “Add New Joint” in the “Input data section” panel. The window “Add new joint data” will open, and the joint information is entered as shown in Fig. 12 (a) through (c) for the rotational joints, and Fig. 12(d) for the translational joint. For the rotational joints, Table 5 summarizes the information of the joint and the number the connected bodies, the pair position with respect to the center of mass of each body, it is to say vectors s_1^p and s_2^p . Table 5 also shows the information for the translational joint: joint number, connected bodies, and position vectors from a point of the reference line to the acting line of the translational joint, they are vectors s_1^p and s_2^p , and

finally the vectors that define the direction of the pair with respect to that reference point: \mathbf{v}_i^j and \mathbf{v}_j^i .

The next step is defining the actuator of the system. The driver is entered by clicking of the “Add New Driver” button inside the “Input data section” panel. The window in Fig. 13 allows the definition of the driver at the joint J1 (Fig. 10), connecting the ground element and the crank. At this point of the simulation, all elements are entered, as seen in the panels “Added links visualization”, “Defined pairs visualization” and “Added drivers visualization” in Fig. 14. The motion type is set by clicking on the “Simulate” button inside the “Input data section” panel, to open the “Add simulation parameters window” in Fig. 15. The gravity vector is also defined in this stage of the simulation.

GUI and SolidWorks. The presented software is registered at National Bureau of Copyrights-Colombia (Dirección Nacional de Derecho de Autor-Colombia) www.derechodeautor.gov.co, under the license Software 13-54-133, name MMA-Mechanism Multibody Analysis [23].

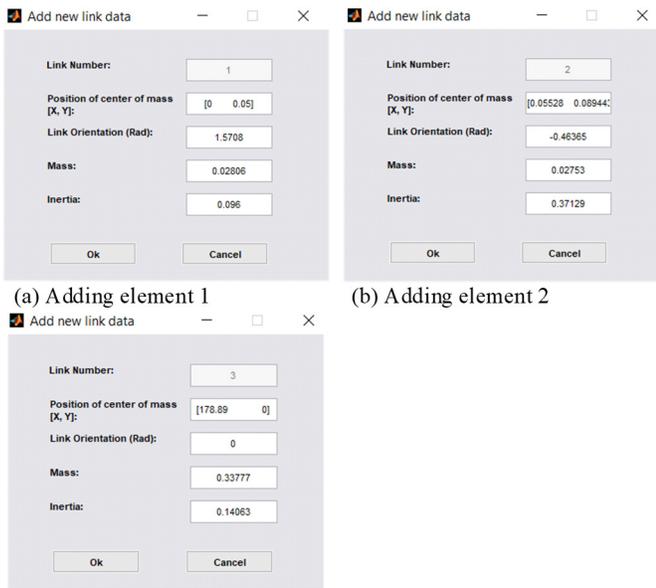


Figure 11. Input of elements 1 to 3 for the sliding-crank mechanism.

Once the simulation is performed, the MMA GUI generates the results of the position, velocity and acceleration kinematics for the center of mass of each movable element, and the dynamic analysis as the reactions at the joints along the motion. By clicking in the “View” button of the “Standard menu” (Fig. 8), it can be selected, from the choices “Position Graphic”, “Velocity Graphic”, “Acceleration Graphic”, “Reaction Graphic” and “Simulation”, from the menu in Fig. 15. In this example, it is of interest the motion of the sliding element. Kinematic results were compared against the motion simulation of a SolidWorks CAD model. SolidWorks results were exported as “.csv” files for plotting and comparing in Matlab, yielding the plots generated in Fig. 16(a), (b) and (c), for the kinematics position, velocity and acceleration of element 3, respectively. Full agreement of the results can be seen. The software allows visualizing dynamics of all elements, and reaction forces of all joints. Fig. 17 presents the x-axis reactions at the joint 1, from the dynamics analysis. There is also full agreement in the compared results: MMA

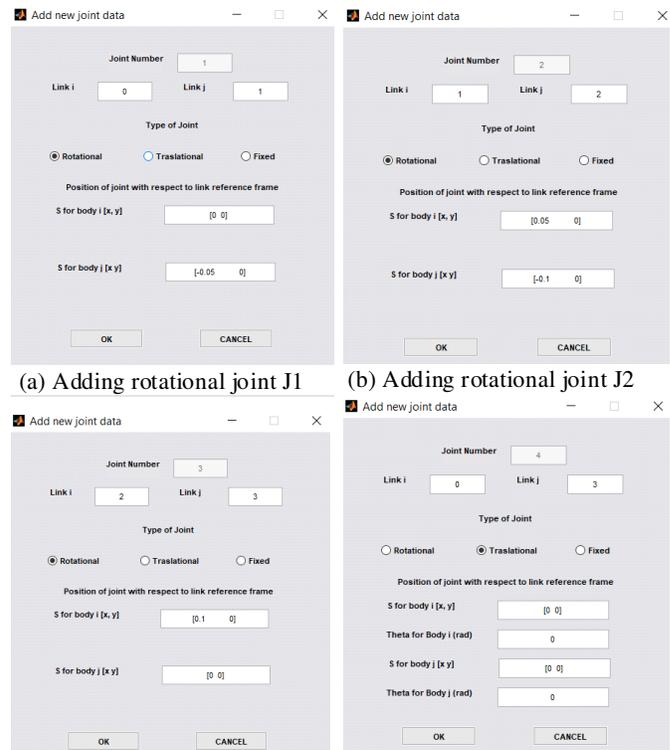


Figure 12. Adding the joints of the sliding-crank mechanism.

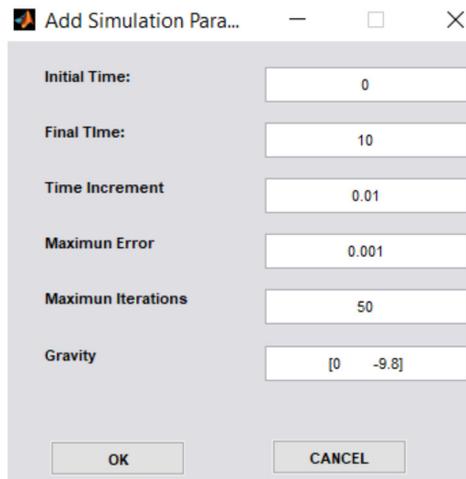


Figure 13. Entering simulation parameters.

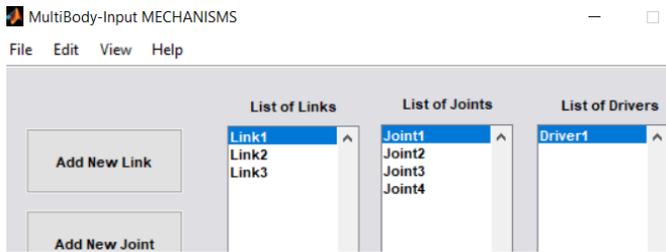


Figure 14. Links, joints and drivers entered in the MMA GUI.

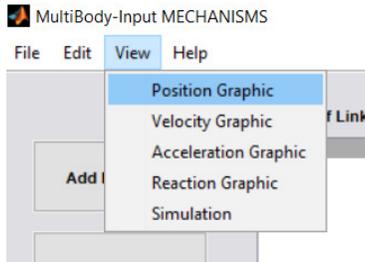


Figure 15. MMA GUI options for plotting the kinematic and dynamic results.

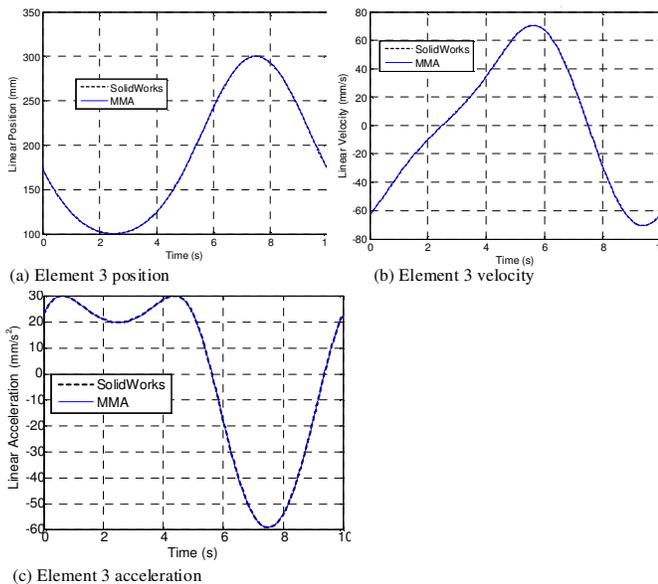


Figure 16. Kinematic results for the sliding element, both MMA GUI and SolidWorks results.

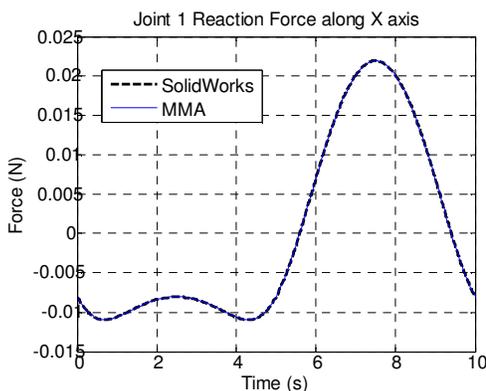


Figure 17. Reaction forces during motion along x axis for the joint 1.

IX. CONCLUSION

A software with graphical user interface created in GUIDE of Matlab was developed from the fundamentals of multibody analysis. In order to facilitate the understanding of the subject and to properly appreciate the effectiveness of the created tool, some multibody fundamental aspects were presented. Some presented aspects include the element type, joint type: rotational or prismatic, and the actuator type: rotatory motor or linear motor. For each element, the definition and vector analysis are presented by showing the relationships between reference frames of the elements, in order to state restriction vector equations that are solved by numerical methods with the Newton-Raphson algorithm. A total of 14 “.m” files and 4 “.fig” Matlab files allow the solution of the kinematics and the dynamics of planar mechanisms, and the interaction with the user. Previous presented background information on the subject is intended in order to facilitate the use of the software by users with little to no previous experience in multibody subject. An illustrative case of the crank-sliding mechanism (it contains both translational and revolute joints) is used for both understating the multibody information of a planar mechanism, and showing the step-by-step use of the software. The user defines elements, joints and actuators, successively, with no need of previous drawing of a CAD model. Kinematic results from the software are obtained and compared against SolidWorks outputs of the same considered motion, showing full agreement.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support provided by the Research Vice-Rector Division at Universidad del Atlántico (Barranquilla, Colombia).

REFERENCES

- [1] M. Stanisic, *Mechanisms and Machines: Kinematics, Dynamics, and Synthesis*. Cengage Learning, 2014.
- [2] K. Waldron, G. Kinzel and S. Agrawal, *Kinematics, Dynamics, and Design of Machinery*. Wiley, 2016.
- [3] J. Uicker, G. Pennock and J. Shigley, *Theory of Machines and Mechanisms*, Oxford University Press, 2010.
- [4] R. Norton, *Design of Machinery: An Introduction to the Synthesis and Analysis of Mechanisms and Machines*. McGraw-Hill Education, 2011.
- [5] H. Mabie and C. Reinholtz, *Mechanisms and Dynamics of Machinery*. Wiley, 1987.
- [6] ACIEM-Asociación Colombiana de Ingenieros Eléctricos, Mecánicos y Afines, “Caracterización profesional de ocho especialidades de la ingeniería – Competencias y funciones de los profesionales recién egresados,” ACIEM, Bogotá, Colombia, 2006.

- [7] E. Haug, *Computer Aided Kinematics and Dynamics of Mechanical Systems: Basic Methods*. Prentice Hall College, 1989.
- [8] A. Shabana, *Dynamics of Multibody Systems*. Cambridge University Press, 2013.
- [9] P. Nikravesh, *Computer-Aided Analysis of Mechanical Systems*. Prentice Hall, 1988.
- [10] J. Wittenburg, *Dynamics of Multibody Systems*. Springer, 2008.
- [11] P. Nikravesh, *Planar Multibody Dynamics: Formulation, Programming and Applications*. CRC Press, 2007.
- [12] M. Coutinho, *Dynamic Simulations of Multibody Systems*. Springer, 2001.
- [13] F. Amirouche, *Fundamentals of Multibody Dynamics. Theory and Applications*. Birkhäuser, 2005.
- [14] J. Font-Llagunes, *Multibody Dynamics: Computational Methods and Applications*. Springer, 2016.
- [15] Z. Terze, *Multibody Dynamics: Computational Methods and Applications*. Springer, 2014.
- [16] A. Engineering, "SAM," February 2017. [Online]. Available: <https://www.artas.nl/es/sam>
- [17] S. Lab, "Universal Mechanism," January 2017. [Online]. Disponible: <http://www.universalmechanism.com/en/pages/index.php?id=1>
- [18] CompMech, "GIM," Universidad del País Vasco, [En línea]. Available: <http://www.ehu.es/compmech/software/>
- [19] D. Rector, "Linkage," January 2017. [Online]. Available: <http://blog.rectorsquid.com/linkage-mechanism-designer-and-simulator/>
- [20] SoftIntegration, "Ch Mechanism Toolkit," Enero 2017. [Online]. Available: <https://www.softintegration.com/webservices/mechanism/>
- [21] A. Schmidt, "MESA VERDE Generation and Application of Complete simulation models for multibody systems," *Vehicle System Dynamics*, vol. 22, no. 1, pp. 158-161, 1993.
- [22] P. Masarati, M. Morandini and P. Mantegazza, "An efficient formulation for general-purpose multibody/multiphysics analysis," *ASME J. Comput. Nonlinear Dyn.*, vol. 9, no. 4, pp. 1-9, 2014.
- [23] E. Yime Rodríguez y J. Roldán Mckinley, "Software MMA-Mechanism Multibody Analysis (Registro 13-54-133)," Dirección Nacional de Derechos de Autor, Bogotá, Colombia, 2016.