INTERNATIONAL JOURNAL OF MECHANICS
DOI: 10.46300/9104.2023.17.2 Volume 17, 2023

Kinematic Analysis of Constrained General Planner Mechanisms Using MATLAB Graphical User Interface (GUI)

[1]Marian William, [2]Moutaz M. Hegaze, [3]Mohamed I. Abu El-Sebah, [4]Yasser Elshaer
[1,2,4]Arab Academy for Science and Technology and Maritime Transport "AAST", Egypt
[3]Electronics Research Institute, Cairo, Egypt. Smart Village "AAST", 12577, Egypt

Received: May 26, 2022. Revised: January 7, 2023. Accepted: February 14, 2023. Published: March 3, 2023.

Abstract—Mechanisms are significant in mechanical engineering as they are required for proper motion transition. This work studies the kinematic analysis of the planar mechanisms using constraint properties between links and joints. A joint library has been built by two types of joints (revolute and prismatic) and has been modeled to be extended in the future. The Kinematic Analysis of General Planer Mechanisms (KAGPM) has been described and implemented through the Graphical User Interface in MATLAB. The Newton-Raphson method was utilized as a numerically computational technique to resolve the kinematic constraint equations. The proposed KAGPM program has been in the kinematic analysis of a slider crank mechanism for the purpose of validation with Haug. For more validation of the program, the effects of some effective parameters have been investigated (the geometric of the mechanism, the initial conditions, and the transient of the mechanism).

Keywords—Computer-aided, Constraints Mechanisms, Kinematic analysis, MATLAB GUI, Planar mechanisms.

I. Introduction

Mechanisms are systems of rigid bodies connected by joints that result in a system with at least one degree of freedom. The transformation of an input motion into a complicated output motion is the main feature [1],[2].

Mechanisms are a topic covered in many mechanical engineering curricula around the world. The way we teach kinematics is evolving with time. By-hand graphical approaches to kinematic analysis of position, velocity, and acceleration were popular in the 1970s. In the past few years, more and more people have been programming algebraic solutions on computers [3].

The multibody analysis approach has a strong foundation in computer-based mechanism analysis nowadays. Pro Engineer, SolidWorks, Solid Edge, Adams, and AutoCAD are some of the most popular licensed software programs. There are many ways to use open-source software, and some of them require the user to learn how to program [4],[5],[6].

Recent CAD systems are smarter and use more automated processes. Engineers have created approaches to help designers utilize automated CAD systems in concurrent engineering. Parametric analysis and intelligence are illustrations. Parametric modeling revolutionized computer-aided design. This accelerates design iteration in commercial CAD software. Most parameters relate to fundamental equations in parametric technology [7].

Expertise and insight from engineers are critical to the design process. Performance may be improved by altering design parameters. There may be numerous iterations on a design based on experience and calculations.

Computer-Aided Design Synthesis (CADSYN) is a CAD program that lets you see and interact with planar four-bar systems [8].

MATLAB's COM Server has a Visual Basic.NET GUI, so users may easily change design parameters. This lets researchers study the kinematics of gantry robots and find the best way to design and set them up [9].

In many areas of research, but particularly in engineering, the use of digital computer programs has grown in importance [10],[11],[12].

On the other hand, little study focuses on their application to the kinematics and dynamics of mechanical systems [13],[14].

The purpose of this study is to introduce the kinematic analysis of the planar mechanisms using constraint properties between links and joints. The Kinematic Analysis of General Planer Mechanisms (KAGPM) has been described and implemented through the Graphical User Interface in MATLAB. The kinematic performance of the slide-crank mechanism has been evaluated to validate the quality and efficacy of the KAGPM program. The results demonstrated a clear correlation with a published result with the same conditions and geometry by Haug [1]. For more validation of the program, the effects of some effective parameters have been investigated (geometric of the mechanism, the initial conditions and the response of the mechanism in the transient state).

II. Kinematic Analysis Of Planar Mechanisms

In kinematics, a mechanism's location, velocity, and acceleration can be studied without regard to the forces that cause the motion to occur. The kinematics of the system must be quantitatively specified in terms of generalized coordinates before equations of motion can be formulated. Generalized coordinates are defined as any combination of variables that uniquely identify the location and orientation of everybody in a mechanism or the configuration of that mechanism.

The global generalized coordinates are defined in this work by vector q. Two types of fames have been introduced to describe the motion of each body i in the mechanism. One of them is the moving frame of reference x_i-y_i, attached to the body, and the other one is the fixed frame x_0-y_0, as shown in Fig. 1. The origin of each moving frame i can be defined with respect to the fixed frame by a vector $r_i = [x_i \quad y_i]^T$ while the rotation of the body is defined by the rotation angle φ_i of the moving frame with respect to the fixed frame. A 3 x 1 column vector $q_i = [x_i \quad y_i \quad \varphi_i]^T$ can be defined to represent planar cartesian generalized coordinates for body i. For a mechanism of Nb bodies, the number of planar cartesian generalized coordinates N is N = 3 x Nb, while the global generalized coordinate N x 1 vector q for the mechanism can be represented by $q = [{q_1}^T \quad {q_2}^T \quad \cdots \quad {q_{Nb}}^T]^T$ [1]-[6].

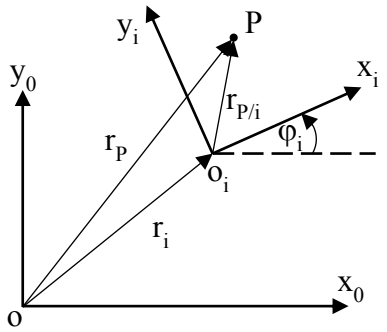

Fig.1 Moving and fixed frames of reference.

The relation between the moving and fixed frames can be defined by the position of point P with respect to them as:

$$r_P = r_i + r_{P/i} \tag{1}$$

Where r_p and r_i are the position of point P on the body i in vector form with respect to the fixed and moving frame, respectively. The position of point P can be represented in matrix form by $P_0 = [P_{x0} \quad P_{y0} \quad 1]^T$ and $P_i = [P_{xi} \quad P_{yi} \quad 1]^T$ in the fixed and moving frame of body i, respectively.

Equation (1) can be represented in matrix form to transfer the position of point P from the moving frame i to the fixed frame.

$$P_0 = T_{0i}\, P_i \tag{2}$$

Where T_{0i} is the transformation matrix from frame i to frame 0 and is defined as; $T_{0i} = \begin{bmatrix} R_{0i} & S_{0i} \\ 0 & 1 \end{bmatrix}$ in the homogenous form.

R_{0i}, S_{0i} is the rotation and translation matrices from frame i to frame 0, respectively.

$$R_{0i} = \begin{bmatrix} C_i & -S_i \\ S_i & C_i \end{bmatrix} \qquad S_{0i} = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

Where C_i and S_i are $\cos(\varphi_i)$ and $\sin(\varphi_i)$, respectively.

Kinematics is the mathematical basis for the position, velocity, and acceleration analysis of planar mechanical systems. A set of kinematic constraint equations between any two bodies which regulate their relative motion can be represented by an algebraic equation $\varphi^k(q,t) = 0$. Kinematic constraints are functions of the system's generalized coordinates but are not explicitly dependent on time. These restrictions characterize a machine's physical structure and provide one or more degrees of freedom. Actuator inputs that display the evolution of some position coordinates over time are frequently employed to represent the movement of mechanical systems. The system of kinematics constraints equations cannot be solved without a driving constraint equation $\varphi^D(q,t) = 0$ that uniquely determines the motion of the system when creating a library of kinematic equations for broad classes of mechanisms [1]-[4].

$$\phi(q,t) = \begin{bmatrix} \varphi^k(q,t) \\ \varphi^D(q,t) \end{bmatrix} = 0 \tag{3}$$

where $\varphi(q,t)$ represents a set of Nc constraint equations. Assume that numerical techniques were employed to solve the algebraic refer to (3) for q at discrete times. Since q is not explicitly known as a function of time, it cannot be differentiated to yield $\dot{q}$ or $\ddot{q}$. Using the chain rule to different refer to (1) to find the derivatives of both sides with respect to time [1]-[4]. The velocity equation is as shown.

$$\frac{\partial \varphi}{\partial q}\dot{q} + \frac{\partial \varphi}{\partial t} = 0 \quad \Longrightarrow \quad \varphi_q \dot{q} = -\varphi_t \tag{4}$$

For the nonsingular Nc x N Jacobian matrix φ_q, refer to (4), which can be solved at discrete time periods to obtain $\dot{q}$. Similarly, using the chain rule to different refer to (4) to find the derivatives of both sides with respect to time [1],[2]-[4]. The acceleration equation is as shown.

$$\frac{\partial \varphi_q}{\partial q}\frac{\partial q}{\partial t}\dot{q} + \frac{\partial \varphi_q}{\partial t}\dot{q} + \frac{\partial \varphi}{\partial q}\frac{\partial \dot{q}}{\partial t} + \frac{\partial \varphi_t}{\partial q}\frac{\partial q}{\partial t} + \frac{\partial \varphi_t}{\partial t} = 0$$
$$\varphi_q \ddot{q} + \left(\varphi_q \dot{q}\right)_q \dot{q} + 2\varphi_{qt}\dot{q} = -\varphi_{tt} \tag{5}$$

III. Joints Library

In this part, a library of joints has been created that consists of all the constraint equations necessary to describe the motion of those joints. With this library, the user can choose the type of joint in the system, making it easy to model and analyze any mechanism. The most common joints used with the planer mechanisms have been represented in this work: revolute and prismatic joints. The joint library created in this work has been programmed as some modules to be extended in future work.

A. Revolute Joint

The revolute joint allows relative rotation about a point P that is common to bodies i and j, *as* shown in Fig. 2. If one body is held fixed, the other body has only one rotational degree of freedom from the pair. The point P is defined with respect to frame i and frame j by P_i and P_j, respectively. As the point P is common, the constraint equations necessary to describe the motion of that joint.

$$P_0 = T_{0i}\, P_i = T_{0j}\, P_j$$

$$\begin{bmatrix} C_i & -S_i & x_i \\ S_i & C_i & y_i \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{xi} \\ P_{yi} \\ 1 \end{bmatrix} = \begin{bmatrix} C_j & -S_j & x_j \\ S_j & C_j & y_j \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{xj} \\ P_{yj} \\ 1 \end{bmatrix}$$

$$C_i P_{xi} - S_i P_{yi} + x_i - C_j P_{xj} + S_j P_{yj} - x_j = 0 \quad (6)$$

$$S_i P_{xi} + C_i P_{yi} + y_i - S_j P_{xj} - C_j P_{yj} - y_j = 0 \quad (7)$$

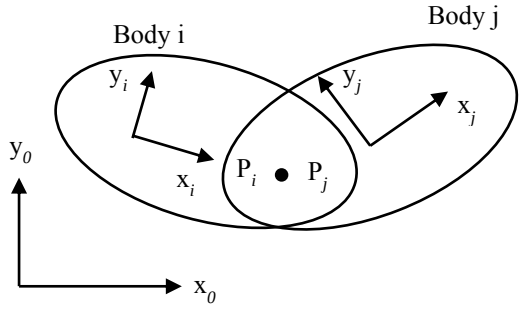

Fig.2 Revolute Joint.

B. Prismatic Joint

A translational joint allows relative translation of a pair of bodies along a common axis but no relative rotation of the bodies. Such a joint may be defined as a straight block on one body that fits precisely into a straight slot (or keyway) on the second body. Let the points P_i, P_j, Q_i and Q_j be specified on a line that is parallel to or on the path of relative translation between bodies i and j for the translational joint depicted in Fig.3. P_i and Q_i are noncoincident locations on body i, while P_j and Q_j are noncoincident points on body j.

Vectors V_i and U_j must always be collinear. i.e., $V_i = A_i\ V_i'$ where $V_i' = [x_i^p - x_i^Q,\ y_i^p - y_i^Q\ ]^T$, and V_j is similarly defined by data that locates P_j and Q_j on body j [1] .

$$V_i = Q_i - P_i \quad V_i = \begin{bmatrix} Q_{xi} - P_{xi} \\ Q_{yi} - P_{yi} \\ 0 \end{bmatrix} \quad \text{OR} \quad V_i = \begin{bmatrix} Q_{xi} - P_{xi} \\ Q_{yi} - P_{yi} \end{bmatrix}$$

$$U_j = Q_j - P_j \quad U_j = \begin{bmatrix} Q_{xj} - P_{xj} \\ Q_{yj} - P_{yj} \\ 0 \end{bmatrix} \quad \text{OR} \quad U_j = \begin{bmatrix} Q_{xj} - P_{xj} \\ Q_{yj} - P_{yj} \end{bmatrix}$$

The rotation of vector V_i by an angle 90° has been introduced to define the perpendicular vector $V_i^{\perp}$ to the vector V_i.

$$V_i^{\perp} = R_{90} V_i \qquad V_i^{\perp} = \begin{bmatrix} Q_{xi} - P_{xi} \\ Q_{yi} - P_{yi} \end{bmatrix}$$

Where $R_{90} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ is the rotation matrix by 90° to obtain the perpendicular.

As the scalar product between any two perpendicular vector equal zero, the constraints equations necessary to describe the motion of that joint can be achieved with the following limitations.

With respect to the moving frames of each body,

$$\left(V_i^{\perp}\right)^T U_j = 0$$

Which can be represented with respect to the fixed frame as:

$$\left(R_{0i} V_i^{\perp}\right)^T R_{0j} U_j = 0 \quad (8)$$

The position vector of the point P in the two frames is represented with respect to the moving frames by:

$$D_{ij} = P_j - P_i$$

With respect to the moving frames of each body,

$$\left(V_i^{\perp}\right)^T D_{ij} = 0$$

Which can be represented with respect to the fixed frame as:

$$\left(V_i^{\perp}\right)^T D_{ij} = \left(R_{0i} V_i^{\perp}\right)^T \left(R_{0j} P_j - R_{0i} P_i\right) = 0 \quad (9)$$

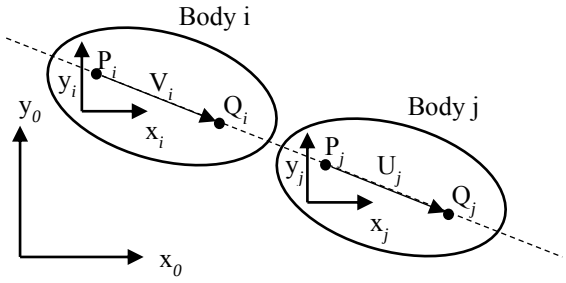

Fig. 3 Prismatic Joint.

IV. Computational Techniques In Kinematics

The kinematic equations that define the system's location are highly nonlinear, but the velocity and acceleration equations are linear and can be solved using matrix factorization and solution techniques that are well suited for computer implementation. Because of the large number of variables involved, it is impossible to write out the explicit governing equations for large-scale systems, much less solve them analytically. Numerical algorithms that quickly converge and are efficient can be used to solve the kinematic position equations on a lime grid at each moment after an assembled configuration has been achieved with independent kinematic and driving restrictions. Mathematical methods, such as the Newton-Raphson method and matrix factorization, will be discussed here to solve nonlinear algebraic constraint equations [1]-[5]. Fig. 4 shows the algorithm of the iterative Newton-Raphson method.

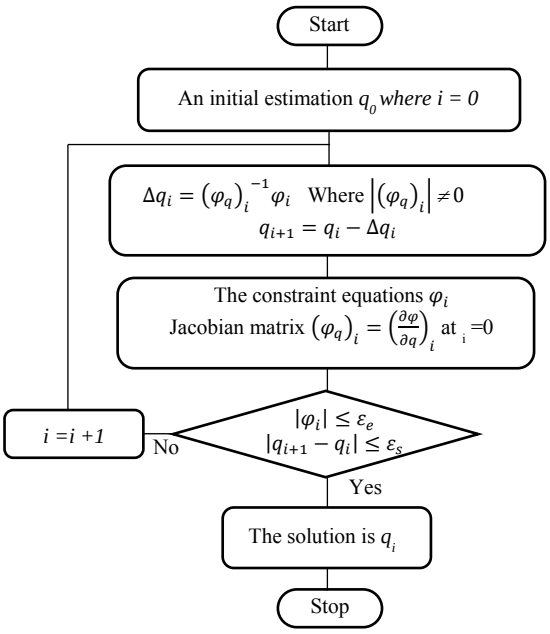

Fig. 4 Newton-Raphson algorithm for constrained equations.

V. Description of KAGPM Program

Fig. 5 displays the KAGPM algorithm, which describes how the KAGPM program operates using MATLAB Graphical User Interface (GUI). The Graphical User Interface makes it much easier to enter data, especially for people who have never used MATLAB before.

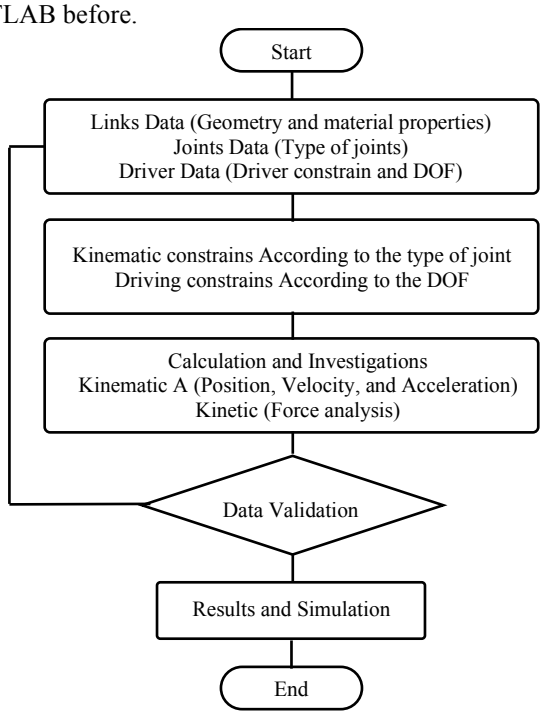

Fig. 5 The KAGPM algorithm.

The general panel of the KAGPM program is shown in Fig. 6, which was created with the goal of making the multi-body mechanism analysis process more user-friendly for people who have never used it before. KAGPM explains the initial model design mechanism by selecting a certain number of links and joints. Assuming that the previous design was correctly implemented, the second step requires user input for each link and describes the type of each joint before selecting the driver constraint joint type and input data for it. The results and simulations have been presented through the curves of the time history of the position, velocity, and acceleration of any point on the mechanism.

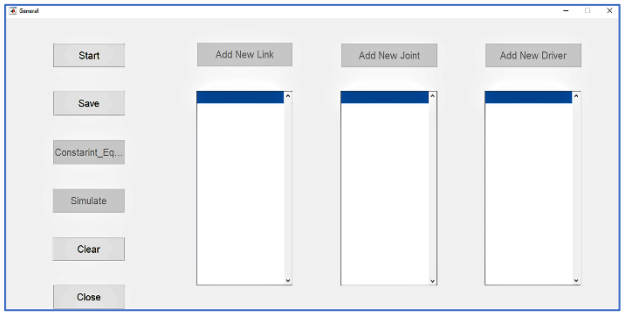

Fig.6 The KAGPM GUI program's panel.

VI. Validation of the Proposed KAGPM Program

A slider crank mechanism has been used for the purpose of validation of the proposed program. This type of mechanism is often found in car engines and sewing machines. Fig.7 depicts the sliding-crank mechanism. To model the slider crank mechanism, the number of links and joints, the kind of joints, and the driver have to be declared. Table I shows the main geometric configurations of the slider crank mechanism used in the validation of the proposed GUI program [1].

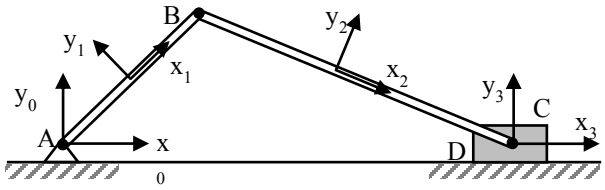

Fig. 7 Slider Crack Mechanism.

Table I. Slider Crank mechanism configurations

Crank length	2 (m)
Connecting rod length	3.5 (m)
Number of links	3
Number of joints	4
Number of Revolute joints	3 joints (A, B and C)
Number of Prismatic joints	1 joint (D)

The global generalized coordinates are defined by:

$$q = [x_1 \quad y_1 \quad \varphi_1 \quad x_2 \quad y_2 \quad \varphi_2 \quad x_3 \quad y_3 \quad \varphi_3]^T$$

The start panel shown in Fig. 8 has been designed to add the links, joints, and driver. Select the link to enter its configurations and its initial conditions as shown in Fig. 9. The initial conditions q_0 of the mechanism have been chosen as:

$$q_0 = [1 \quad 0 \quad 0 \quad 3.75 \quad 0 \quad 0 \quad 5.5 \quad 0 \quad 0]^T$$

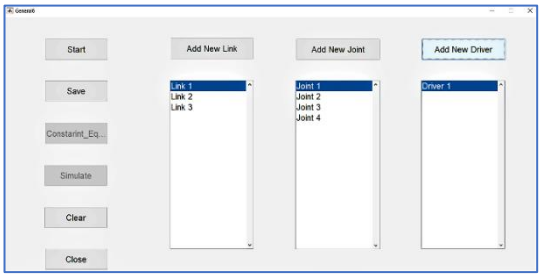

Fig. 8 Start panel of KAGPM.

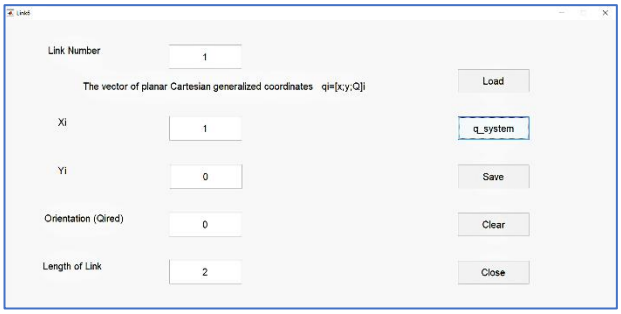

Fig. 9 Initial conditions and configurations of any link.

The slider crank mechanism includes four joints (three revolute joints and one prismatic joint). The details of which are summarized in Table II to show its configuration and the local position with respect to the moving frame of each link. The number of joints has been chosen as indicated in the KAGPM homepage, as shown in Fig. 8, and the information for each joint

has been entered, as shown in Fig. 10. The constraint equations 6-9 of each joint have been created 8 equations according to the data in Table 2 (two constraint equations for each joint). It is necessary to specify a driver constraint equation to complete the set of constraint equations and the two links to which the actuator is attached.

The driving constraint at joint A between links 0 and 1 as:

$$\varphi_1 = \frac{\pi}{4} + \omega t \quad (10)$$

Where the angular velocity constant ω is considered for the validation by 4π [1].

Table. II Joints configuration and location

Joint	Type	Link i	x_i	y_i	Link j	x_j	y_j
A	Revolute	0	0	0	1	-1	0
B	Revolute	1	1	0	2	-1.75	0
C	Revolute	2	1.75	0	3	0	0
D	Prismatic	3	0	0	0	0	0

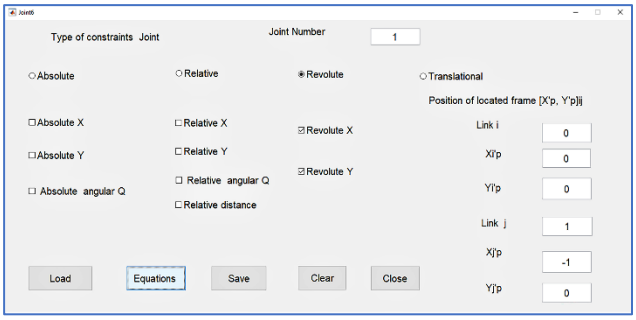

Fig. 10 The local position and configurations of any joint.

Fig.11, Fig.12, and Fig.13 show the output results of the KAGPM program, the position, velocity, and acceleration of the slider.

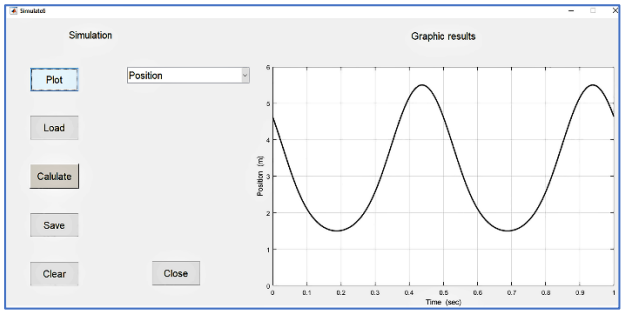

Fig. 11 Time history of the position of the slider.

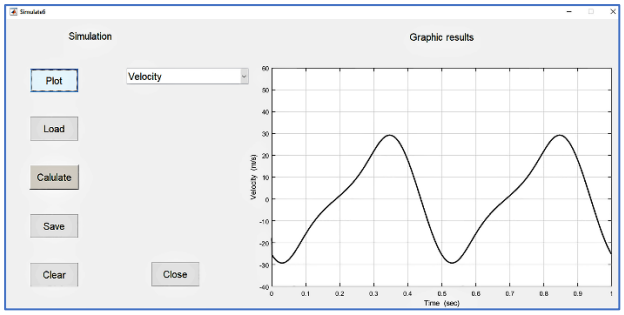

Fig. 12 Time history of the velocity of the slider.

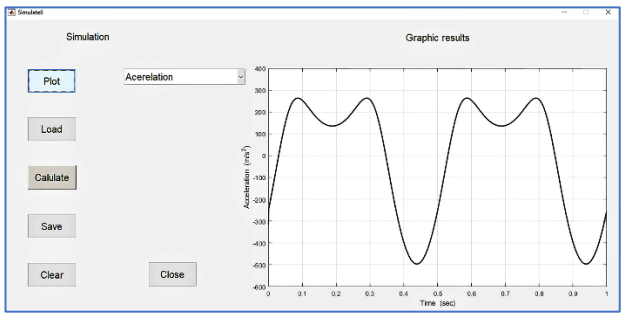

Fig. 13 Time history of the acceleration of the slider.

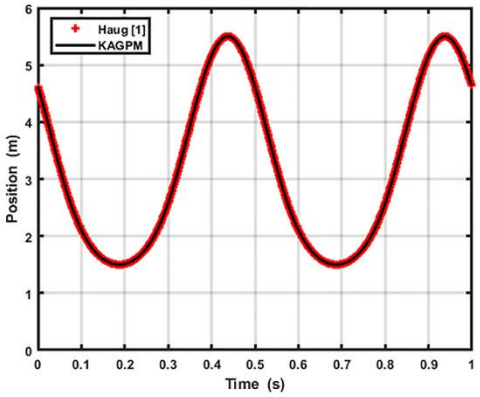

Fig. 14 The position of the slider by KAGPM and Haug [1].

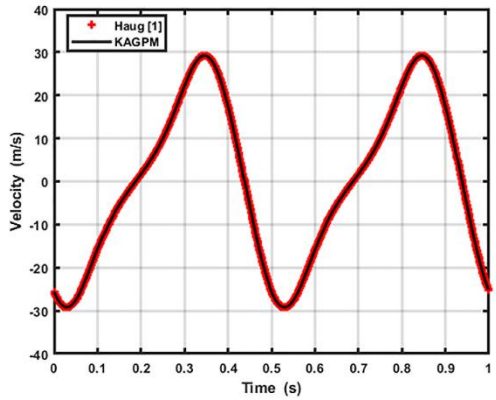

Fig. 15 The velocity of the slider by KAGPM and Haug [1].

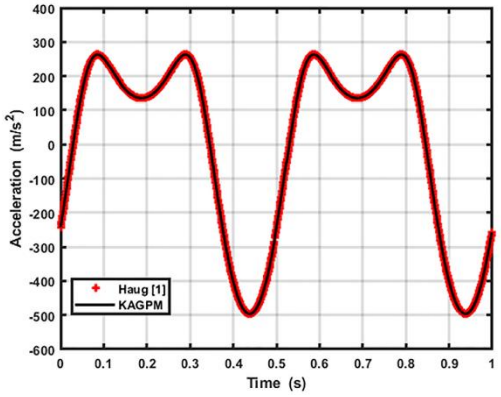

Fig. 16 The acceleration of the slider by KAGPM and Haug [1].

Fig.14, Fig15, and Fig16 show the comparison between the KAGPM results and the published results with the same conditions by Haug [1]. Clearly, the results are in total agreement with the good visualization of kinematics by the KAGPM program.

VII. Parametric Study of Some Effective Parameters

For more validation of the proposed KAGPM program in different conditions and with different parameters, a parametric study of some effective parameters on the response of the mechanisms, the initial conditions with different time intervals, and the length of the connecting rod, has been investigated.

A. Effect of the initial conditions (crank angular velocity).

In the validation, the system has been investigated at an initial angular velocity of the crank ω=4π (rad/s), which remains constant during the interval of time as indicated refer to (10). In this part, the effect of the initial angular velocity of the crank has been studied during three different intervals of time to reach the same final angular velocity $\omega_f = 4\pi$ (rad/s) in all cases with the last second. The computational process has been carried out using Laptop DELL, Processor Core i7-5500U CPU, 2 cores, 4 threads, 3 Ghz, and RAM of 16 GB.

The first interval of time is from 0 to 1.5 seconds. The angular acceleration needed in the transient period of time (0: 0.5 s) to reach the final angular velocity $\omega_f = 4\pi$ (rad/s) when the system starts its motion from rest is angular acceleration $\alpha=8\pi$ (rad /s^2).

The computational time needed for the transient state (0:0.5 s) is (33 min), while the time needed for the steady state period of time (0.5:1 s) is (3 min).

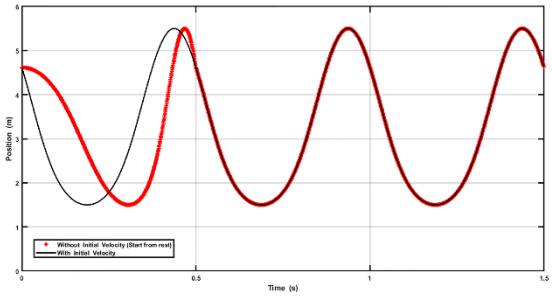

Fig.17 The position of the slider during the time interval (0:1.5 s)

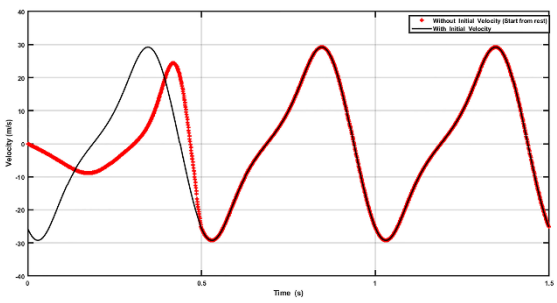

Fig.18 The velocity of the slider during the time interval (0:1.5 s)

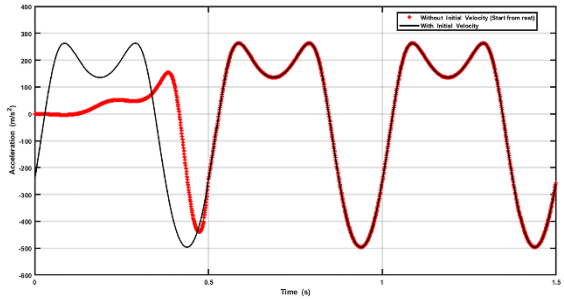

Fig.19 The acceleration of the slider during the time interval (0:1.5 s).

Fig.17, Fig.18, and Fig.19 show the time history of the slider position, velocity, and acceleration during an interval of time from (0 to 1.5 s). The response of the system is divided into two stages. The first stage is a transient stage from (0 to 0.5 s) where the velocity and acceleration response are less than the response in the steady state stage. The inertia effect on the slider has been improved when the system starts its motion from rest. The second stage is a steady state stage from (0.5 to 1.0 s) which is in total agreement with the case of initial angular velocity.

The second interval of time is from 0 to 2.0 second. The angular acceleration needed in the transient period of time (0:1.0 s) to reach the final angular velocity ω_f =4π(rad/s) when the system starts its motion from rest is α=4π (rad /s^2).

The computational time needed for the transient state (0:1.0 s) is (75 min), while the time needed for the steady state period of time (1.0: 2.0 s) is (9 min).

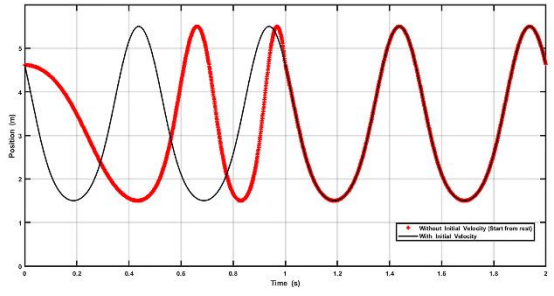

Fig. 20 The position of the slider during the time interval (0:1.5 s)

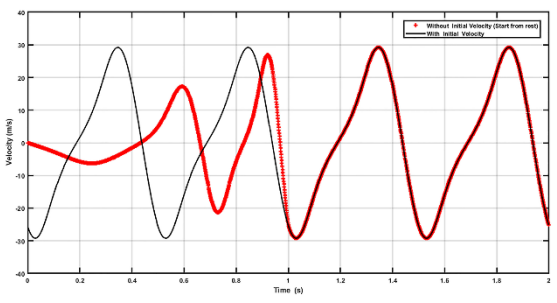

Fig. 21 The velocity of the slider during the time interval (0:1.5 s)

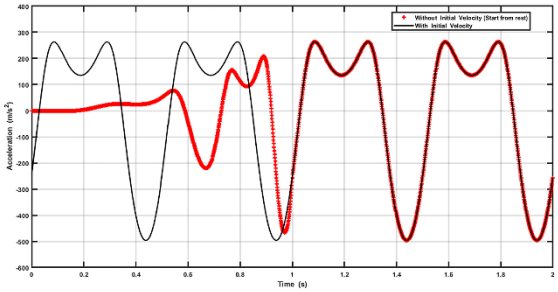

Fig. 22 The acceleration of the slider during the time interval (0:1.5 s).

Fig.20, Fig.21, and Fig.22 show the time history of the slider position, velocity, and acceleration during an interval of time from 0 to 2.0 second. The response of the system is divided into two stages. The first stage is a transient stage from (0 :1.0 s) where the velocity and acceleration response are less than the response in the steady state stage. The inertia effect on the slider has been improved when the system starts its motion from rest. The second stage is a steady state stage from (1.0 to 2.0 s) which is in a total agreement with the case of initial angular velocity.

The third interval of time is from 0 to 3.0 second. The angular acceleration needed in the transient period of time (0:3.0 s) to reach the final angular velocity $\omega_f = 4\pi$ (rad/s) when the system starts its motion from rest is $\alpha = 2\pi$ (rad /s^2).
The computational time needed for the transient state (0:2.0 s) is (240 min), while the time needed for the steady state period of time (1.0:2.0 s) is (17 min).

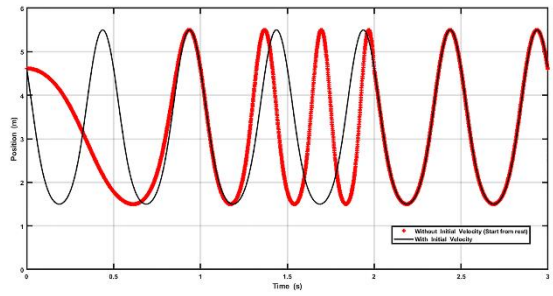

Fig. 23 The position of the slider during the time interval (0:3.0 s).

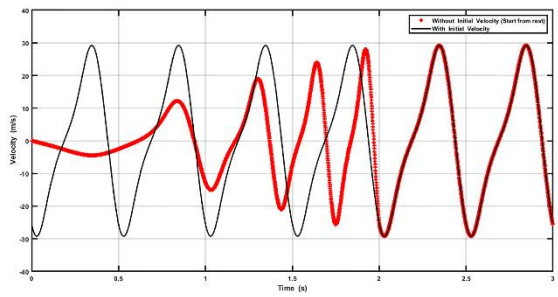

Fig. 24 The velocity of the slider during the time interval (0:3.0 s).

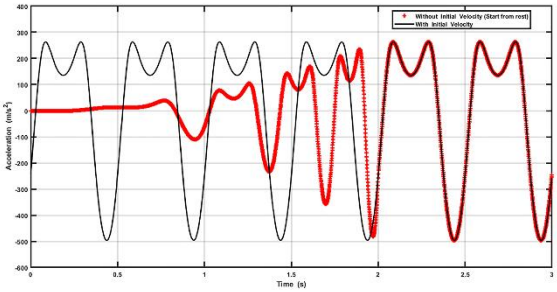

Fig. 25 The acceleration of the slider during the time interval (0:3.0s).

Fig.23, Fig.24, and Fig.25 show the time history of the slider position, velocity, and acceleration during an interval of time from 0 to 3.0 second. The response of the system is divided into two stages. The first stage is a transient stage from (0 to 2.0 s) where the velocity and acceleration response are less than the response in the steady state stage. The inertia effect on the slider has been improved when the system starts its motion from rest. The second stage is a steady state stage from (2.0 to 3.0 s) which is in a total agreement with the case of initial angular velocity.

B. Geometric effect (length of the connecting rod).

In this part the effect of the geometric of the mechanism has been studied with different lengths of the connecting rod while the crank length is constant for all cases. The effect has been investigated on the kinematic performance of the mechanism. As declared in the previous section, the effect of the initial conditions on the performance of the mechanism, the kinematic analysis of the mechanism in this investigation has been carried out during the time interval from (0 to 2.0 s) and the system started its motion from rest. Three different lengths of the connecting rod have been considered in the investigation 2.5 (m), 2.2 (m) and 2.1 (m).

Fig.26, Fig.27, and Fig.28 show the position, velocity, and acceleration of the slider with the three proposed length compared with the original one in the previous section 3.5(m). the figures show that the sensitivities of the kinematic response of the mechanism to the length of the connecting rod. The acceleration and velocity of the slider affected by the length of the connecting rod which increase by decrease the length of the connecting rod. The inertia effect on the slider has been improved by increase the connecting rod length.

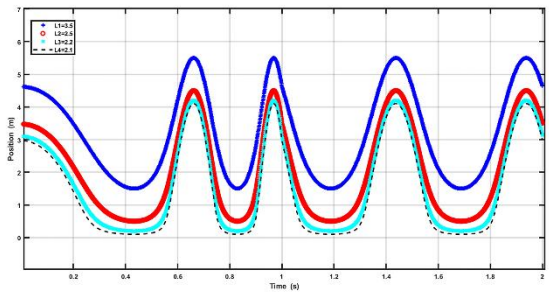

Fig. 26 The position of the slider with different length of connecting rod.

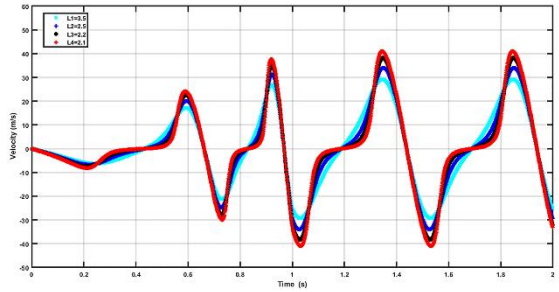

Fig. 27 The velocity of the slider with different length of connecting rod.

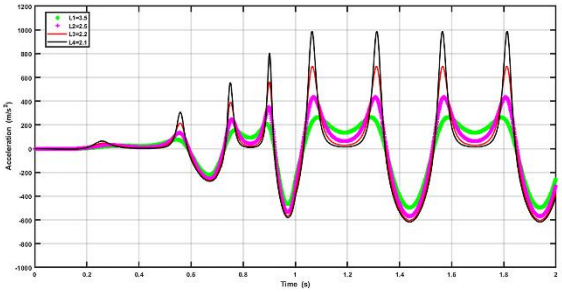

Fig. 28 The acceleration of the slider with different length of connecting rod.

VIII. Conclusion

The kinematic analysis of the planar mechanisms has been studied using constraint properties between links and joints. A joint library has been built by two types of joints (revolute and prismatic) and is modeled to be extended in the future. Kinematic Analysis of General Planer Mechanisms (KAGPM) has been described and implemented through the Graphical User Interface in MATLAB. To validate the quality and efficacy of the KAGPM program, the kinematic performance of the slide-crank mechanism has been evaluated. The results demonstrated a clear correlation with a published result with the same conditions and geometry by Haug. For more validation of the program, the effect of some effective parameters has been investigated (geometric of the mechanism- the initial conditions and the response of the mechanism in the transient state).

References

[1] E. J. Haug, "Computer-Aided Kinematics and Dynamics of Mechanical System", United States, Iowa, vol. 1, 1989, pp.48–200.

[2] M. Stanisic, "Mechanisms and Machines: Kinematics, Dynamics, and Synthesis", USA, SI Edition, Apr 15, 2014.

[3] J. Zhao, and Z. Feng, and F. Chu, and N. Ma, "Advanced Theory of Constraint and Motion Analysis for Robot Mechanisms," Chinese, vol.1, November 19, 2013.

[4] K. Waldron, G. Kinzel, and S. Agrawal, "Kinematics, Dynamics, and Design of Machinery," vol. 3, May 2016.

[5] E. Y. Rodríguez, and J. R. Mckinley, "Multibody approach MATLAB GUI for kinematic and dynamic analysis of planar mechanisms,", Barranquilla. Colombia, vol. 23, pp. 433-442, 5 December 2018.

[6] Z. Terze, "Multibody Dynamics: Computational Methods and Applications," in Springer, vol. 35, Barcelona, Spain, Jun 26, 2014.

[7] P. E. Nikravesh," Planar Multibody Dynamics: Formulation, Programming with MATLAB," U.S.A, second edition, 2007.

[8] E. Demir, "Kinematic Design of Mechanisms in A Computer- Aided Design Environment", Middle East Technical University, May 2005.

[9] M. Ismai, and M. Khalafalla, "Gantry Robot Kinematic Analysis User Interface Based on Visual Basic and MATLAB", Tianjin. China, vol. 4, February 2015.

[10] A. Shabana, "Dynamics of Multibody Systems,", University of Illinois. Chicago, Fifth Edition ,5 March 2020.

[11] H. Tan, Y. Hu, and L. Li "Effect of friction on the dynamic analysis of slider-crank mechanism with clearance Wuhan, China, vol 115, October 2019, pp. 20-40.

[12] A. El-Nady, "application of computational techniques in dynamic analysis of mechanical systems,", Military Technical College Chair of Base Science, Cairo ,2007, unpublished.

[13] E.C. Cobb, E. Brigham, C. Destefano, and "Slider –Crank Mechanism for Demonstration and Experimentation MQP,", April 25, 2013.

[14] S. Gramblička, R. Kohár, and M. Stopka, "Dynamic analysis of mechanical conveyor drive system," Zilina, Slovakia, vol. 192, pp. 259-264, 22 June 2017.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US