Defence Science Journal, Vol. 73, No. 6, November 2023, pp. 642-655, DOI : 10.14429/dsj.73.6.18941

Closed-Loop Vector Formulation in Euler's Complex Numbers for Multi-Loop Planar Mechanisms With N-bars: A Novel Modeling Approach and Algorithm

Rahul V.M., Sandesh Bhaktha B.* and Gangadharan K.V.

National Institute of Technology Karnataka, Surathkal, Mangalore - 575 025, India
**E-mail: sandesh.bhaktha@gmail.com*

ABSTRACT

This paper presents a novel iterative algorithm incorporated in a user-friendly GUI for modeling the kinematics of multiple looped N-bar closed-loop mechanisms. Past research works have used custom coding or expensive commercial software to analyze the mechanisms of specific applications. The proposed algorithm focuses on kinematics and offers a quick, easy-to-use, cost-effective solution to analyze a wide range of generic mechanisms, reducing the need for custom coding and lowering computational costs. The algorithm employs algebraic equations, such as solving complex closed-loop vector equations using the Euler form of complex numbers, to simulate and derive the unknowns necessary to characterise any generic closed-loop mechanism. The Python code implemented in the algorithm adapts to various scenarios by utilising available information on the position, velocity, and acceleration variables of the mechanisms. The simulation tool can display real-time color contour plots (RGB color scale) for linear and angular velocities and accelerations, simulate mechanisms with multiple loops and switch configurations, and find inverse mechanisms. The approach for solving multiple loop problems and the algorithm utilized to solve the configurations, methods, equations used and GUI features implementation are all described in this study. The case study considered for a four-bar mechanism indicates a strong agreement between the results obtained from the proposed kinematics-based simulator and ANSYS software. These results demonstrate the simulator's effectiveness in providing low-cost and user-friendly simulation results for various generic mechanisms involving multiple interconnected loops.

Keywords: Closed-loop mechanisms; Algorithm; Novel simulation tool; Multiple loops; Mechanism design

NOMENCLATURE

r (m)	Length of link
θ (rad)	The angle of the link from horizontal
ω (rad / s)	Angular velocity of link
α (rad / $s^{2)}$	Angular acceleration of link
ν (m / s)	Velocity of link
a (m/s^2)	Acceleration of link
N	Number of links

1. INTRODUCTION

Simulation of kinematic mechanisms is an essential aspect of modern engineering practices that enables designers to analyse and evaluate complex motion characteristics of various mechanisms. It involves the use of simulation tools and modeling techniques that have proven to be highly beneficial in better understanding the kinematic behavior of these mechanisms. The four-bar mechanism is a prime example of a mechanism that benefits from such analysis, and it finds extensive application in the automotive industry such as windshield wiper mechanisms[1]. Simulation tools and modeling techniques are crucial in designing efficient mechanisms, and they play a critical role in the decision-making process. Kinematics concepts such as velocity, acceleration, and position analysis are essential for designing these mechanisms, and simulators can be used to verify the design decisions based on these concepts[2]. The use of simulation tools and modeling techniques has revolutionised the design process, making it more efficient and reliable. These tools enable designers to visualise and analyse the motion characteristics of mechanisms, which helps them to make proper design decisions. They allow for the testing of different design concepts, making it possible to evaluate the performance of the mechanism under different conditions. Furthermore, simulation tools can also help designers identify potential problems and provide solutions to overcome them, leading to better and more robust designs.

Research projects focusing on inverse kinematics utilised different modeling and simulation techniques. Marques, *et al.* proposed a method for modeling and simulating closed-loop kinematic chains[3], while Ding, *et al.* used genetic algorithms to improve a five-bar mechanism[4]. Müller focused on removing unnecessary constraints in loop closure[5], and Han[6], *et al.* presented a systematic modeling and analysis approach for wheeled mobile robots[6]. Sünkün[7], *et al.* developed a toolbox for simulating and analysing the toe trajectory of legged robots[7]. Adorno and Marinho employed robotic controlled and modeling libraries using dual quaternion algebra[8]. Dang[9],

Received : 09 March 2023, Revised : 19 May 2023
Accepted : 23 May 2023, Online published : 03 November 2023

et al. developed a kinematic simulation model based on SimMechanics and Adams for a specific mechanism[9], while Wang[10], *et al.* developed a 5-dof parallel mechanism[10]. Wang[11], *et al.* examined the shield machine's thrust mechanism[11], while Yamamoto[12], *et al.* suggested performing a kinematic analysis of a closed-loop mechanism using an automatic procedure extraction algorithm[12]. Bai[13], *et al.* carried out thorough kinematic modeling for parallel hexapod robot applications[13], and Bieze[14], *et al.* presented a finite element modeling approach using closed-loop kinematics for soft manipulators[14]. Bhattacherjee and Saha developed an accurate six-degree-of-freedom simulation model for homing missiles incorporating detailed servo-system modeling[15]. Additionally, Virtual Labs has developed simulators that simplify the calculation of position, speed, and acceleration variables for various mechanisms[16].

Some research works conducted have also used simulation software tools to analyse and optimise mechanisms with adjustable parameters. Programming languages such as Simulink, MATLAB, and Python were utilized to provide visual and interactive representations of the system's behavior and assist designers and engineers in creating accurate and efficient mechanism designs[17-20]. Kinematics and mechanism design principles, including the analysis of linkages, inverse and forward kinematics, and the D-H method, were also studied[21-25]. These concepts were found to be crucial in designing and building various mechanical systems, including robots and power transmission mechanisms. Python and MATLAB were commonly used programming languages to analyze and design kinematic and mechanism systems. Several research works also explored the practical application of mechanism design[26-28]. These applications include predicting ground workpiece surface roughness, power transmission, and mechanical stability on different terrains and planning in high-speed press lines. The surveys emphasised the importance of accurately designing and analysing mechanisms for optimal performance and efficiency in various applications, including manufacturing and robotics.

Prior studies on mechanism design, synthesis, and experimental verification have primarily focused on specific mechanisms, utilising simulation tools and algorithm codes with limited built-in mechanisms. The analysis of complex mechanisms featuring multiple loops and links necessitated costly commercial software and high computational power. Furthermore, developing codes for detailed modeling such mechanisms for in-depth analysis was time-intensive. To address these issues, this study proposes an iterative algorithm to solve any generic feasible kinematic closed-loop mechanisms using Euler form complex numbers calculations which is cost and computationally effective. The methodology allows for easy data input for closed-loop mechanisms having n loops and m links, which the algorithm can solve effortlessly. Any open-loop mechanisms can be converted into closed-loop mechanisms by adding a stationary ground link, enabling the simulation to run. To visualize output results, the study developed a GUI capable of plotting color contour and displaying running time data. The simulation results are compared with commercially available softwares as they are widely used for academic research purposes[29-30] like ANSYS[31]. The proposed algorithm simplifies developing codes for analyzing custom mechanisms, providing an accurate and effective solution.

2. THEORETICAL BACKGROUND

The closed-loop vector loop equations solved for various cases considered are briefly discussed in this section and implemented in code to model the kinematics of closed-loop linkages. The fundamental equation of close loop linkage formation could be written as a set of vectors in the Euler form, as shown in Eqn (1). Where, $\vec{r_k}$ is the vector making an angle θ with the positive x-axis, r_k is the actual length of the link. Subscript k indicates it is the K^{th} linkage member. Since all the linkages in a closed loop are arranged head-to-tail, their summation should become a null vector.

$$\vec{0} = \sum_{n} r_k e^{i\theta_k} \tag{1}$$

Upon differentiating $\vec{r_k}$ concerning time, it yields a velocity vector of the K^{th} linkage having the Euler form,

$$\overrightarrow{v_k} = \left(v_k + ir_k\omega_k\right)e^{i\theta_n} \tag{2}$$

$$\vec{0} = \sum_{n}\left(v_k + ir_k\omega_k\right)e^{i\theta_k} \tag{3}$$

$$\overrightarrow{a_k} = \left(\left(a_k - r_k\omega_k^2\right) + i\left(2v_k\omega_k + r_k\alpha_k\right)\right)e^{i\theta_k} \tag{4}$$

$$\vec{0} = \sum_{n}\left(\left(a_k - r_k\omega_k^2\right) + i\left(2v_k\omega_k + r_k\alpha_k\right)\right)e^{i\theta_k} \tag{5}$$

The three summation equations of a loop requiring summation must be solved and equated to zero to determine the closed-loop kinematics of any linkage configuration. If there are multiple loops, each loop is solved separately. These solutions are used to compute the unknown data for the next loop. Any loop that can be solved proceeds in steps, first solving equations to get length and angle, then velocity, and finally acceleration. In further sections, the specifics of the equations implemented are described.

2.1 Concepts and Equations

This work utilised the concept of mechanisms[2] to resolve all 4 cases for each of the three sets of unknowns (position, velocity, and acceleration), forming 12 cases as described in this section. These cases are obtained by assuming a maximum of 2 unknown variables for each set of unknowns for a given loop. They are as follows (where subscript *p* and *q* denote p^{th} and q^{th} link's variables).

Position variables include a combination of unknown parameters (Fig. 1), such as:

- r_p and θ_p
- r_p and θ_q
- r_p and r_q
- θ_p and θ_q

Similarly, velocity variables include:

- v_p and ω_p
- v_p and ω_q
- v_p and v_q
- ω_p and ω_q

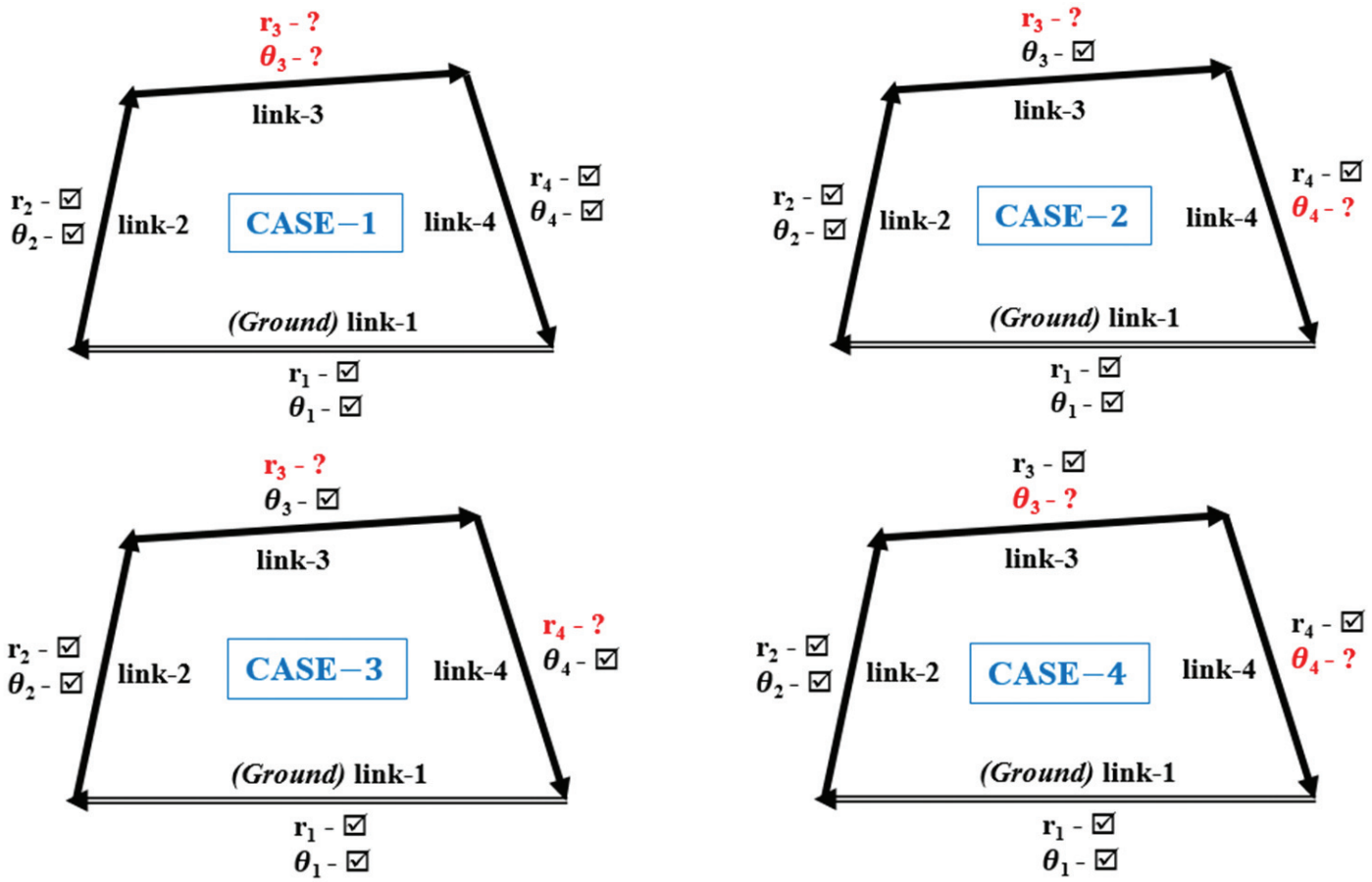


Figure 1. Four cases for solving position variables taking the 4-bar mechanism as an example.

Further, acceleration variables include:

- a_p and α_p
- a_p and α_q
- a_p and a_q
- α_p and α_q

The fundamental solution concept involved constructing equations by considering a single closed loop that could have a maximum of two unknowns for each position, velocity, and acceleration per loop. This assumption enabled simulation and coverage of all potential problem-solving scenarios. The equation was rearranged to place unknown vector values at the first two links, but its vector addition features allowed the unknown values to exist anywhere between the links of a closed loop, greatly simplifying the coding process. The program used subscript p or q to indicate unknown values for the first or second linkages and computed all position variables before determining acceleration and speed variables using input values. The modified algebraic equations for each position, velocity, and acceleration are described as follows.

2.2 Position

The solution found in the position part can fully define the mechanism in terms of length and orientation of links. As a result, if there is any discrepancy in finding the solution to either length or angle for any link, it implies that the mechanism cannot be solved for the given set of data values, and the mechanism is unlikely to be feasible. The closed loop vector equation in Euler form for N-links can be written as shown in Eqn. (1), and a resultant of the vector addition of fully defined links can be written as Eqn. (6),

$$r_k e^{i\theta_k} = \sum_n r_n e^{i\theta_n} \tag{6}$$

where, *r* and θ are the length and angle measured from the positive x-axis, respectively. The LHS of Eqn. (6) represents the resultant of all the link vectors fully defined in terms of *r* and θ. If the resultant vector $r_k e^{i\theta_k}$ is null, the known data values form a closed loop altogether, and no unknown variable can be solved in the current loop chosen.The closed loop equation can only have a maximum of two unknowns per loop between *r* and θ because there are two equations for each loop (obtained by separating the real and imaginary parts). All four scenarios for solving position variables are considered (because finding one unknown is a subset of finding two unknowns per loop). These four scenarios are:

Case-1 (r_p and θp are unknowns)

$$r_k e^{i\theta_k} = \sum_{j\neq p} r_j e^{i\theta_j} \tag{7}$$

$$r_p e^{i\theta_p} = -r_k e^{i\theta_k} \tag{8}$$

Case-2 (r_p and θ_q are unknowns)

$$\theta_q = \theta_p - \sin^{-1}\left(-\frac{r_k \sin(\theta_k - \theta_p)}{r_q}\right) \tag{9}$$

$$r_p = -\frac{r_k \sin\left(\theta_k - \theta_q\right)}{\sin\left(\theta_p - \theta_q\right)}$$

Case-3 (r_p and r_q are unknowns)

$$r_q = -\frac{r_k \sin\left(\theta_k - \theta_p\right)}{\sin\left(\theta_q - \theta_p\right)} \tag{10}$$

$$r_p = -\frac{r_k \sin\left(\theta_k - \theta_q\right)}{\sin\left(\theta_p - \theta_q\right)} \tag{11}$$

Case-4 (θ_p and θ_q are unknowns)

$$\theta_p - \theta_q = \cos^{-1}\left(\frac{r_k^2 - r_p^2 - r_q^2}{2 r_p r_q}\right) \tag{12}$$

$$\text{let } \varphi = \theta_p - \theta_q \tag{13}$$

$$\theta_p = \theta_k - \sin^{-1}\left(-\frac{r_q \sin\left(\varphi\right)}{r_k}\right) \tag{14}$$

$$\theta_q = \theta_p - \varphi \tag{15}$$

After completing this process, all position variables are clearly defined in the array of unknowns. It should be noted that the given condition might have various solutions because there are numerous possible angles (by solving inverse-trigonometric equations), and each corresponds to a different configuration. The proposed simulator can cover all these scenarios by switching between several configurations based on user input which is an important feature.

2.3 Velocity

The velocity part of the equation is to find $\left(V = \frac{d\vec{r}}{dt}\right)$ and $\left(\omega = \frac{d\theta}{dt}\right)$. This equation is obtained by differentiating any general j^{th} position vector with respect to time.

$$\frac{d\vec{r}}{dt} = \vec{v} = \left(v_j + i r_j \omega_j\right) e^{i\theta_j} \tag{16}$$

Similar to position equations, v, and ω can only have a combined total of 2 unknowns per loop. All four potential scenarios for obtaining velocity variables are considered because finding a single unknown is a subset of finding two unknowns per loop. They are:

Case-1 (v_p and ω_p are unknowns)

$$v_p = -v_k \cos\left(\theta_k - \theta_p\right) \tag{17}$$

$$\omega_p = -\frac{v_k \cos\left(\theta_k - \theta_p\right)}{r_p} \tag{18}$$

Case-2 (v_p and ω_q are unknowns)

$$\omega_q = -\frac{v_v \sin\left(\theta_k - \theta_p\right)}{r_q \sin\left(\theta_q + \frac{\pi}{2} - \theta_p\right)} \tag{19}$$

$$v_p = -\frac{v_v \sin\left(\theta_k - \theta_q - \frac{\pi}{2}\right)}{\sin\left(\theta_p - \theta_q - \frac{\pi}{2}\right)} \tag{20}$$

Case-3 (v_p and v_q are unknowns)

$$v_q = -\frac{v_v \sin\left(\theta_v - \theta_q\right)}{\sin\left(\theta_q - \theta_p\right)} \tag{21}$$

$$v_p = -\frac{v_v \sin\left(\theta_v - \theta_p\right)}{\sin\left(\theta_p - \theta_q\right)} \tag{22}$$

Case-4 (ω_p and ω_q are unknowns)

$$\omega_q = -\frac{v_v \sin\left(\theta_v - \frac{\pi}{2} - \theta_p\right)}{r_q \sin\left(\theta_q - \theta_p\right)} \tag{23}$$

$$\omega_p = -\frac{v_v \sin\left(\theta_v - \frac{\pi}{2} - \theta_q\right)}{r_p \sin\left(\theta_p - \theta_q\right)} \tag{24}$$

These are the main equations and the four possible scenarios to calculate the velocity unknowns for each loop. All velocity-related variables for every link in the loops are clearly defined after this process.

2.4 Acceleration

The acceleration part is to solve the acceleration equations to find $\left(\vec{a} = \frac{d\vec{v}}{dt}\right)$ and $\left(\alpha = \frac{d\omega}{dt}\right)$. This equation is obtained by differentiating any general j^{th} velocity vector with respect to time.

$$\frac{d\vec{v}}{dt} = \vec{a} = \left(\left(a_j - r_j \omega_j^2\right) + i\left(2 v_j \omega_j + r_j \alpha_j\right)\right) e^{i\theta_j} \tag{25}$$

Like position and velocity equations, the number of unknowns per loop can be a maximum of 2 among a and α. All the possible 4 cases for finding acceleration variables are,

Case-1 (a_p and α_p are unknowns)

$$a_p = -a_a \cos\left(\theta_a - \theta_p\right) \tag{26}$$

$$\alpha_p = -\frac{a_a \sin\left(\theta_k - \theta_p\right)}{r_p} \tag{27}$$

Case-2 (a_p and α_p are unknowns)

$$\alpha_q = -\frac{a_a \sin\left(\theta_a - \theta_p\right)}{r_q \sin\left(\theta_q + \frac{\pi}{2} - \theta_p\right)} \tag{28}$$

$$a_p = -\frac{a_a \sin\left(\theta_a - \theta_q - \frac{\pi}{2}\right)}{\sin\left(\theta_p - \frac{\pi}{2} - \theta_q\right)} \tag{29}$$

Case-3 (a_p and a_q are unknowns)

$$a_q = -\frac{a_a \sin\left(\theta_a - \theta_p\right)}{\sin\left(\theta_q - \theta_p\right)} \tag{30}$$

$$a_p = -\frac{a_a \sin\left(\theta_a - \theta_q\right)}{\sin\left(\theta_p - \theta_q\right)} \tag{31}$$

Case-4 (α_p and α_q are unknowns)

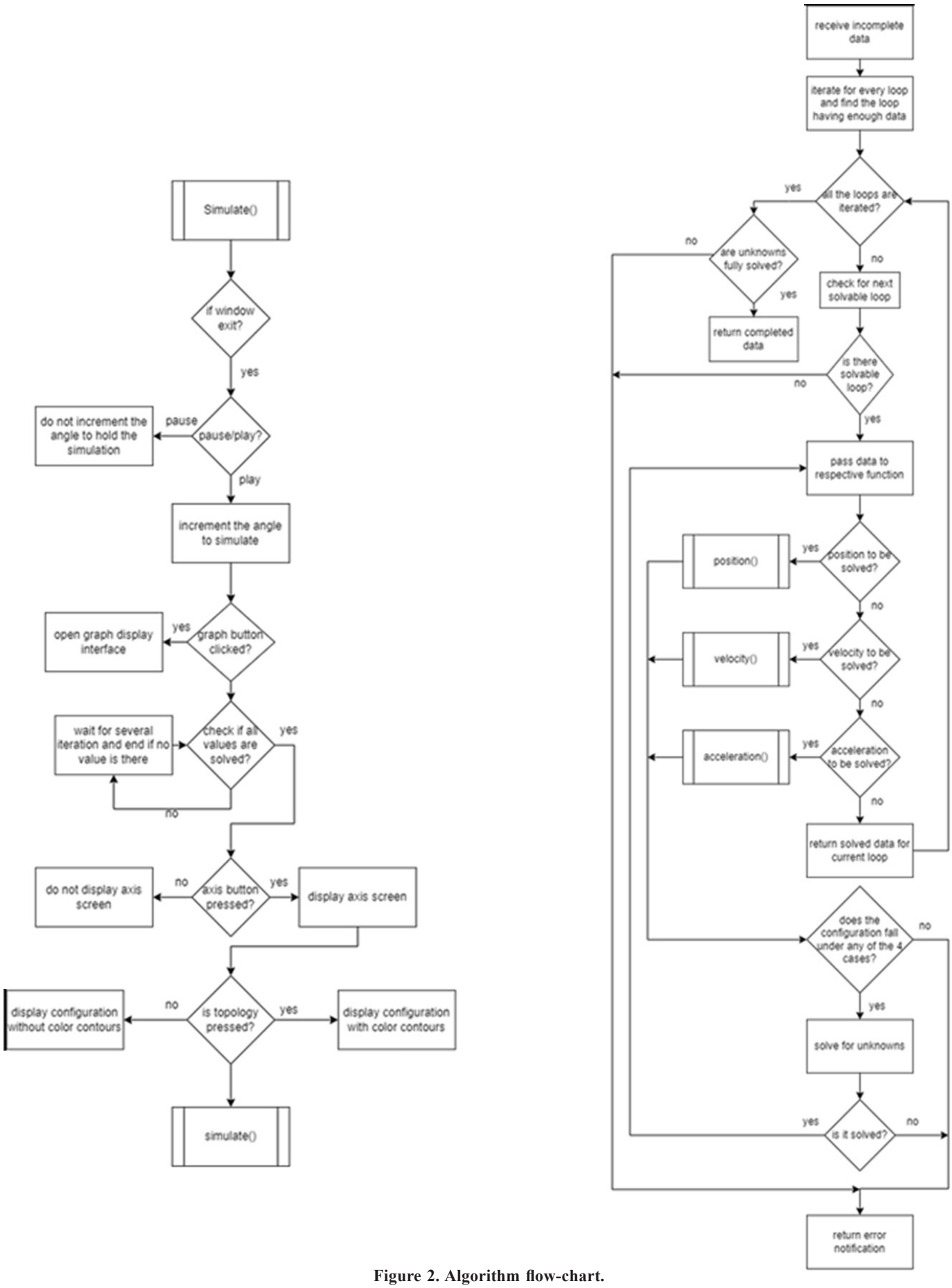


Figure 2. Algorithm flow-chart.

$$\alpha_q = -\frac{a_a \sin\left(\theta_a - \frac{\pi}{2} - \theta_p\right)}{r_q \sin\left(\theta_q - \theta_p\right)} \quad (32)$$

$$\alpha_p = -\frac{a_a \sin\left(\theta_a - \frac{\pi}{2} - \theta_q\right)}{r_p \sin\left(\theta_p - \theta_q\right)} \quad (33)$$

These are the general equation formulations for four scenarios for solving each loop's unknown accelerations. Once complete, each link in the loop is fully defined.

3. METHODOLOGY

The primary focus of this part is on the fundamental requirements that the loops must satisfy, along with a few default presumptions in the code that are used to assist the user when the simulator first launches. Further, this section focuses on implementing multi-loop mechanism, color contour, inverse mechanisms, multiple configurations, and problem-solving algorithms. All of the algorithms were programmed in Python, which was also used to create interactive user interfaces for displaying data and a sketching interface. Additionally, the process of identifying solutions is broken down into many functions, each dependent on the results of the others. The algorithm solves all necessary configurations and displays only the configuration required by the user.

3.1 Solution Method for Multi-Loop Mechanisms

This section describes the criteria the loop must meet for the simulator to solve the entire configuration, including position, velocity, and acceleration. The solution method for multi-loop mechanisms is also discussed, a notable feature of the proposed simulator. Additionally, default assumptions made in the simulator and a case study simulation of the crank-rocker mechanism using the proposed simulator and Ansys are presented.

3.1.1 Criteria to be Satisfied by the Loop

Loop represents a series of links where the magnitude of the assumed vectors is equal to the length of the links, and the direction forms an angle theta with the horizontal axis. When connected from head to tail, these vectors create a closed loop. The code used to build the simulator can solve any configuration with any number of links as long as the configuration satisfies certain assumptions, such as:

- It does not contain any open loops
- Each loop may contain an N-number of links, but each loop should have no more than two unknowns for length and angle
- The unknowns may pertain to the same or separate links within the loop
- For the configuration to be solvable, the loop should be similar to the solvable loop in Fig. 3(b).

The velocity and acceleration calculations are also subjected to the same restrictions, with a maximum of two unknowns (for velocity and acceleration), either for the same link or different links in the same loop. As the requirements are satisfied, the algorithm solves the complete configuration, from position to acceleration. An alert stating 'configuration cannot be solved' is notified if the abovementioned requirements are not satisfied during simulation since no combination of length and theta of each link can form a closed loop. Further, the values of the variables for subsequent simulation are modified by incrementing the angle of driving links depending on the angular velocity and acceleration at that instant. The mechanisms depicted in Fig. 3(b) illustrate the configurations that are solvable and non-solvable.

3.1.2 Default Assumptions Made in the Simulator

Certain assumptions were made while developing the algorithms for the simulator to simplify user input. By default, the simulator displays a four-bar system, and users can create their configurations by typing values. However, if a drawing interface gives the input, the code automatically changes all links except the last two to driver links. Additionally, the simulator ensures that closed loops have at least n-2 degrees of freedom among n links. Users can vary the initial angular acceleration and velocity of all driving links. If a three-link system is used, one of the links is variable in length, and the length input should be left blank for the code to determine its value.

3.1.3 Solution Algorithm

The algorithm for solving the equations described in section 2 is divided into three sections: position(), velocity(), and acceleration(). It checks the problem's feasibility and solves it accordingly. If there is only one loop, the algorithm runs only once. However, if there are multiple loops, the algorithm examines each loop until it finds one that meets the number of unknown variables requirements. It then solves the loop and saves the values in an array before locating the next solvable loop. Loops with feasible unknowns that share a common link with the solved loop are considered the following loop for solving. The algorithm continues until all loops are fully solved, and the array of unknowns is fully defined. If the criteria for the closed loop are not satisfied or any other rule is violated, an error is thrown back, warning that the problem cannot be solved. The process stops if any discrepancies exist. The flow chart in Fig. 2(a) and Fig. 2(b) illustrates this process.

3.1.4 Color Contours

In simulation visualization, the color contours feature played a crucial role in displaying the relative values of different parameters for each link. This feature is demonstrated in Fig. 3(a), which displays the color bands for four parameters: ω, α, a, and v, ranging from red (highest value) to blue (lowest value). The color contours were assigned to the links based on their parameter value ratio, with red denoting the most valued link and blue denoting the least valued link. For example, based on the difference values (4,8,7), the links (ground, 2, 3, 4) were assigned the colors (0-red, 3-orange, 10-green, 16-blue). The color contour feature was instrumental in visualizing the simulation data for different parameters and links, providing valuable insights into the simulation results.

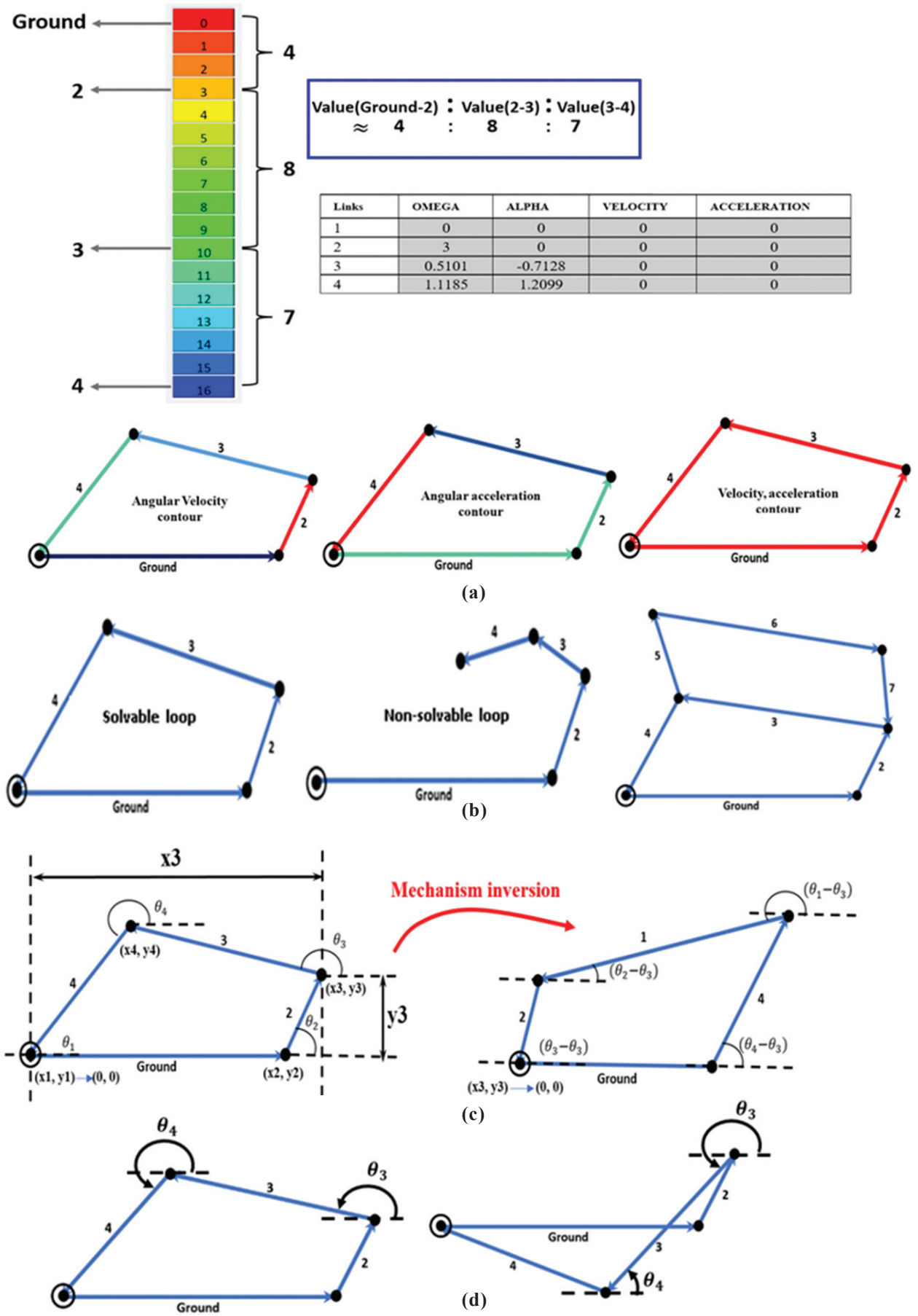


Figure 3. Features of simulator GUI, (a) Color contours with data values, (b) Solvable, non-solvable loop and multiple-loops, (c) Mechanism inversions, and (d) Multiple configurations.

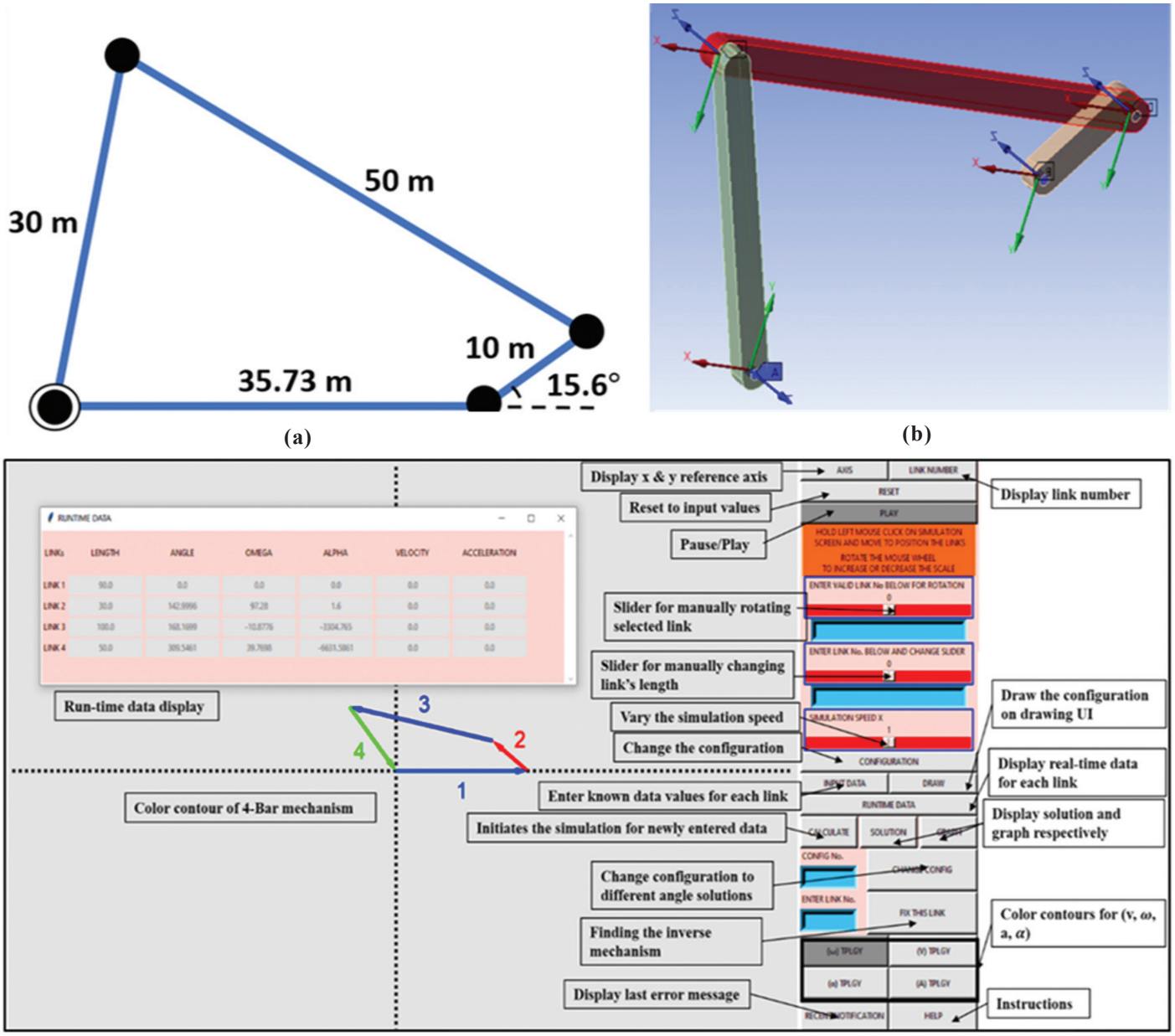


Figure 4. Case study validation: (a) Case study geometry, (b) Isometric view of the four bar assembly, and (c) Simulator GUI.

3.1.5 Multiple Loops

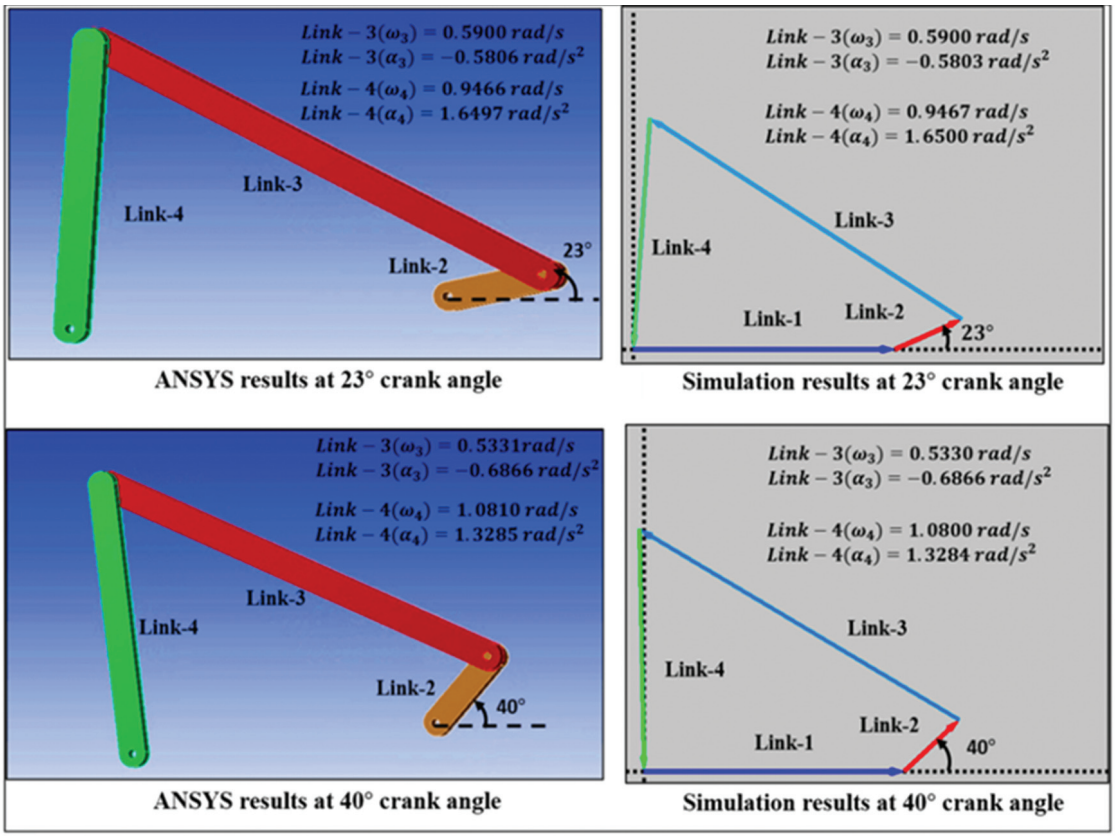
The solution method for multi-loop mechanisms is also addressed in this work, which is one of the highlighting features. Multiple loops are created when at least one link connects two loops, as demonstrated in Fig. 3(b). If a loop (say L1) has more unknowns than the algorithm can solve, the attached solvable loops (say L2, L3, and more) are solved first, based on specific criteria discussed in the section 3.1.1. These solvable loops provide information about undefined links in the loop (L1). The number of unknowns decreases upon solving other loops, enabling the present loop, which previously had many unknowns, to be solved using the known values from other solved loops. As shown in Fig. 3(b), when two loops share a common link, it becomes fully defined upon solving the loop with more known values (assuming it is the bottom loop). The top loop can now be fully resolved using this common link, provided all requirements are met.

Table 1. Ansys simulation settings

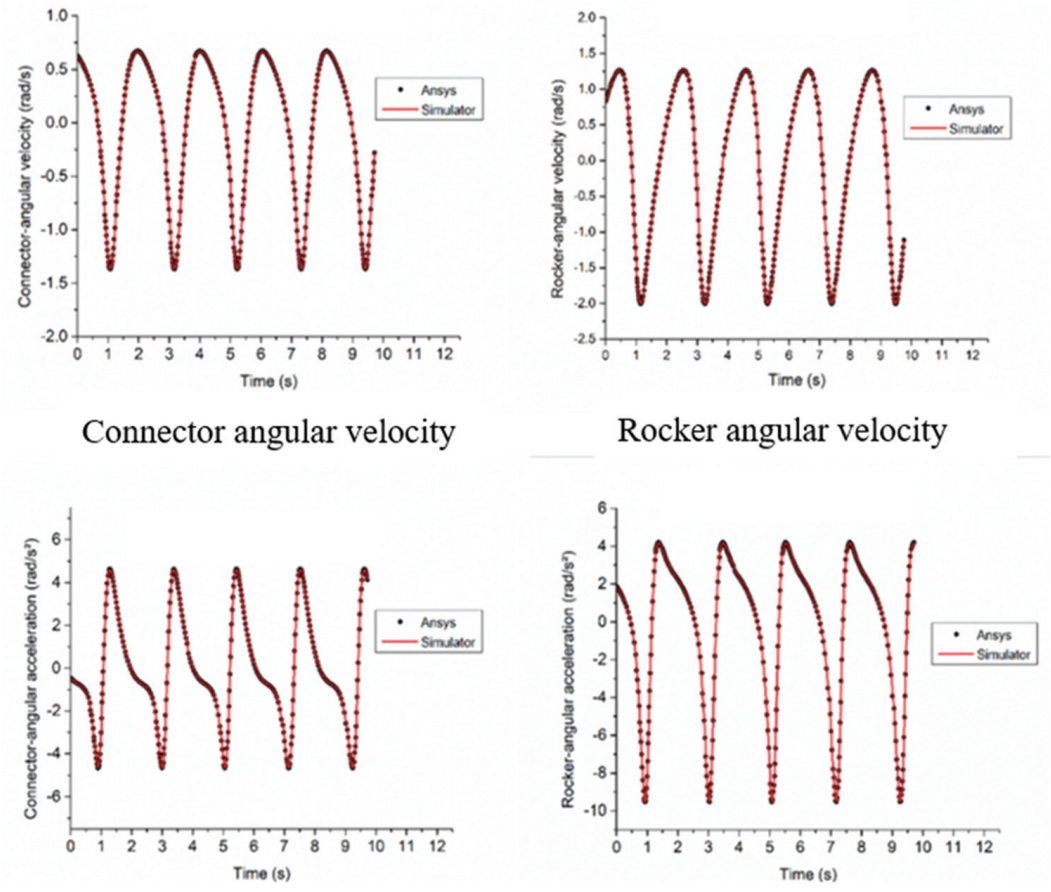
Step Controls	
Number of steps	1
Current step number	1
Step end time	200 s
Auto time stepping	On
Initial time step	10^{-2} s
Minimum time step	10^{-7} s
Maximum time step	0.05 s

3.1.6 Inverse Mechanisms

Different inverse mechanisms could be obtained by selecting various links as ground links based on their link number and subtracting their angle from each link in the



(a)



(b)

Figure 5. Ansys v/s Simulator results comparison for connector angular velocity, rocker angular velocity, connector angular acceleration and rocker angular acceleration: (a) ANSYS v/s simulations results for two different time stamps, and (b) ANSYS v/s simulation data value plots as a function of time.

configuration. Further, the link was moved to the screen's center to simulate its inverse mechanism. Different links were grounded to show different inverse mechanisms, and the typical simulation was repeated concerning the newly grounded link's altered frame of reference. An n^{th} loop link can be grounded by considering its angle about the x-axis and the coordinate of its base point. The complete mechanism was rotated and shifted to the new base point. Figure 3(c) depicts an example of a 4-bar mechanism in which the new inversion was created by shifting and rotating the links of the original/base configuration being inverted by link 3.

3.1.7 Multiple Configurations

In some cases, multiple configurations can be achieved for various unknowns. One way to achieve multiple configurations is to include the angle as an unknown. This can be done using inverse trigonometric relations, producing many angles for unknown values. These extra configurations can be helpful in certain situations, such as when the mechanism must adapt to different operating conditions.

The second configuration in Fig. 3(d), which results from the connector and rocker links' undetermined angles (θ_3 and θ_4), is an example of how multiple configurations can be achieved. In this case, the mechanism can have different connector and rocker link arrangements that result in different motion characteristics. Understanding these different configurations can be necessary for designing and analysing mechanisms to ensure they can perform their intended functions effectively.

3.2 Simulation and Comparison of a Crank-Rocker Mechanism Using the Proposed Simulator and Ansys

A case study simulation of the crank-rocker mechanism using the proposed simulator and Ansys is presented in this study. The mechanism's dimensions used in the study are shown in Fig. 4(a). The primary objective was to look at the patterns in the plots for the rocker and connector links' angular velocity and acceleration. Results of the proposed simulator and Ansys simulation were compared to ensure the accuracy of the algorithm.

Figure 4(a) illustrates the lengths of the links and their angles from the positive x-axis. The mechanism's response is examined by studying the trends of the plots for the rocker and connector links. The study employs the simulator GUI with a default 4-bar (along with its color contour of angular velocity) mechanism shown in Fig. 4(c), which can be modeled and refined by entering accurate values into user-provided fields. The simulation's accuracy is validated by comparing it with Ansys simulation results, using a 4-bar setup to model the crank-rocker mechanism, as shown in Fig. 4(b), with identical linkage dimensions and revolute joint constraints. The Ansys simulation measures rotational velocity and acceleration through probes during runtime.

4. RESULTS AND DISCUSSION

The current section deals with the results of the case study described in section 3.2, followed by the highlighting features of this work, such as the simulation of multiple loops, displaying the geometry, multiple configurations, and inverse mechanism along with angular position plots of links to show the behavior of the multi-loop mechanism.

4.1 Case Study Result Comparison

The geometric statistics of the ansys model include a total of 3 bodies, of which three are active hence containing three nodes and elements. Based on the step controls shown in Table 1, a rotational velocity of magnitude 0.04 rad/s was applied for simulation purposes for 200 sec. time stamps. The data values of velocity and acceleration variables at two particular time stamps having 23° and 40° crank angles are compared in the simulator and ANSYS software, as shown in Fig. 5(a).

The solver controls had program-controlled time integration types using position and velocity correction features of a pure kinematic kind. The assembly was done with the inertia matrix with a dropoff tolerance of $10^{-6,}$ along with relative assembly tolerance. The Non-linear controls had program control energy accuracy tolerance while storing the data values at all times under Output controls. The values for ten data samples are shown in the subsequent Tables. Table 2 shows ANSYS values, Table 3 shows simulator values

Table 2. ANSYS simulation data

Rocker-angular velocity (Z) (rad/s)	Rocker-angular acceleration (Z) (rad/s²)	Connector-angular velocity (Z) (rad/s)	Connector-angular acceleration (Z) (rad/s²)
0.8364	1.8473	0.6235	-0.4807
0.8547	1.8173	0.6186	-0.4984
0.8791	1.7757	0.6117	-0.5213
0.9099	1.7204	0.6022	-0.5490
0.9466	1.6497	0.5900	-0.5806
0.9881	1.5621	0.5745	-0.6147
1.0331	1.4558	0.5556	-0.6503
1.0810	1.3285	0.5331	-0.6866

Table 3. Algorithm simulation data

Rocker-angular velocity (Z) [rad/s]	Rocker-angular acceleration (Z) [rad/s²]	Connector-angular velocity (Z) [rad/s]	Connector-angular acceleration (Z) [rad/s²]
0.8362	1.8472	0.6237	-0.4805
0.8549	1.8168	0.6186	-0.4981
0.8791	1.7754	0.6115	-0.5211
0.9095	1.7213	0.6023	-0.5494
0.9467	1.6500	0.5900	-0.5803
0.9884	1.5630	0.5745	-0.6145
1.0332	1.4559	0.5555	-0.6501
1.0800	1.3284	0.5330	-0.6866

Table 4. Error percentage

Rocker-angular velocity (Z) (rad/s) (%)	Rocker-angular acceleration (Z) (rad/s²) (%)	Connector-angular velocity (Z) (rad/s) (%)	Connector-angular acceleration (Z) (rad/s²) (%)
0.0203	0.0081	0.0337	0.0562
0.0246	0.0303	0.0065	0.0542
0.0011	0.0163	0.0262	0.0365
0.0418	0.0529	0.0116	0.0601
0.0180	0.0176	0.0017	0.0603
0.0364	0.0551	0.0035	0.0423
0.0116	0.0089	0.0180	0.0323
0.0009	0.0090	0.0113	0.0058

and Table 4 shows error percentages between both values. The final data values produced by the simulator and the ANSYS simulation following the above case study showed good agreement with each other for the variables,

- Angular velocity of the connector link
- Angular velocity of rocker link
- Angular acceleration of connector link
- Angular acceleration of rocker link

As shown in Fig. 5(b). The plot shows apparent data overlapping, ensuring the simulator results are highly accurate.

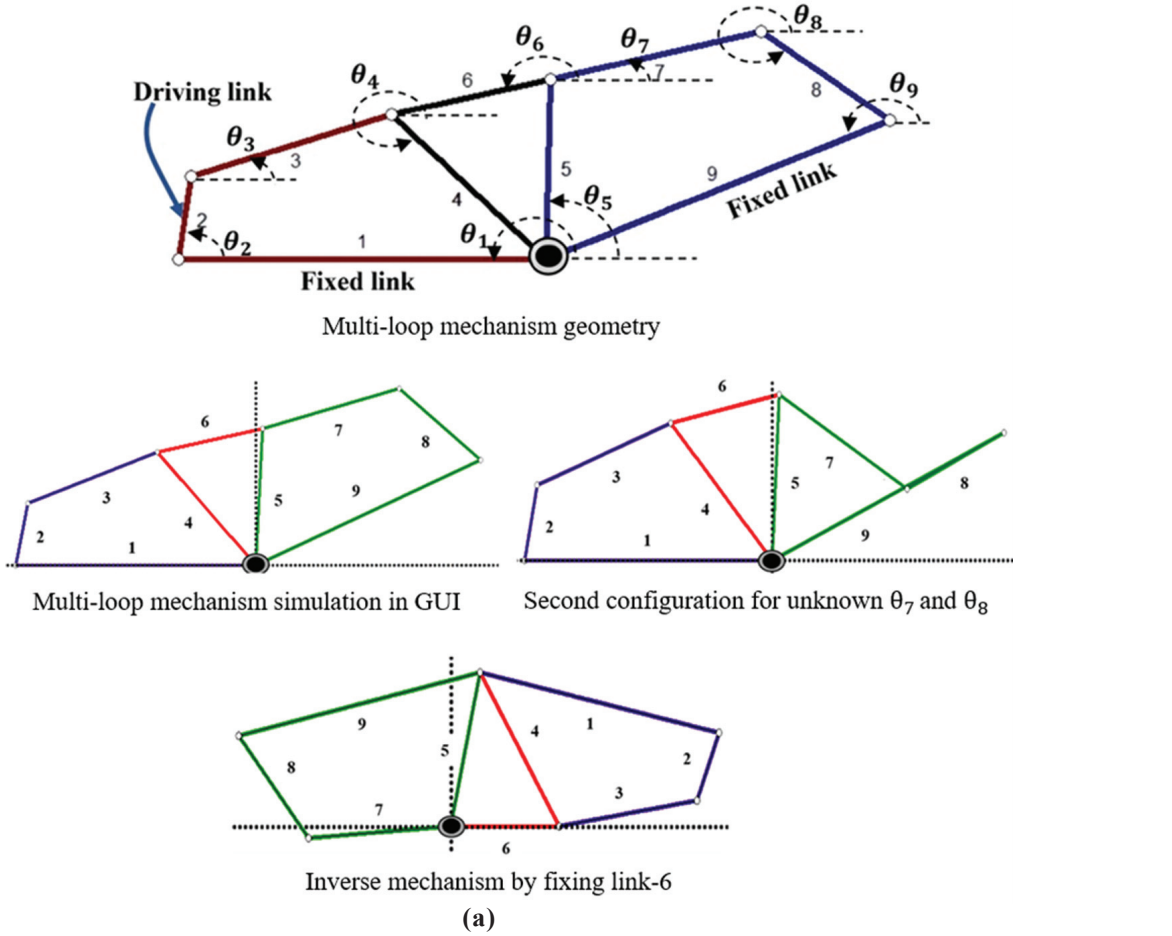
The results provided by the simulator proposed are accurate and reliable enough since the solution process is purely analytical and doesn't involve any approximation. As seen from Table 5, there are a few variations between the two values that errors in the rounding of the values and matching of time stamps may cause. Thus, this simulator offers quick and straightforward visualizations for the specified configuration.

Table 5. Angular position of links in multiple loop mechanism

Iterations	θ_2 (Deg)	θ_7 (Deg)	θ_8 (Deg)
1	98.34	1.92	342.37
2	89.74	8.84	331.64
3	81.15	12.86	325.25
4	72.55	15.72	320.74
5	63.96	17.84	317.47
6	55.37	19.37	315.19
7	46.77	20.34	313.76
8	38.18	20.79	313.12
9	29.58	20.70	313.24
10	20.99	20.10	314.11

4.2 Simulation of Multiple Loops

Simulation of multiple loop mechanisms can be beneficial in developing Remote Centre Mechanisms (RCM)[32], a minor-mobility mechanism involving a rotating part around a fixed point located distally from it. Due to its unique mechanical characteristics, this special mechanism finds its application in various fields. In surgery, the RCM mechanism is beneficial for minimally invasive procedures, where it is used as a wrist for surgical robots. Another implementation of the multi-loop mechanism described by Mohamed & Duffy[33] focused on a planar multi-loop mechanism while describing the first-order geometric influence coefficients in their work.



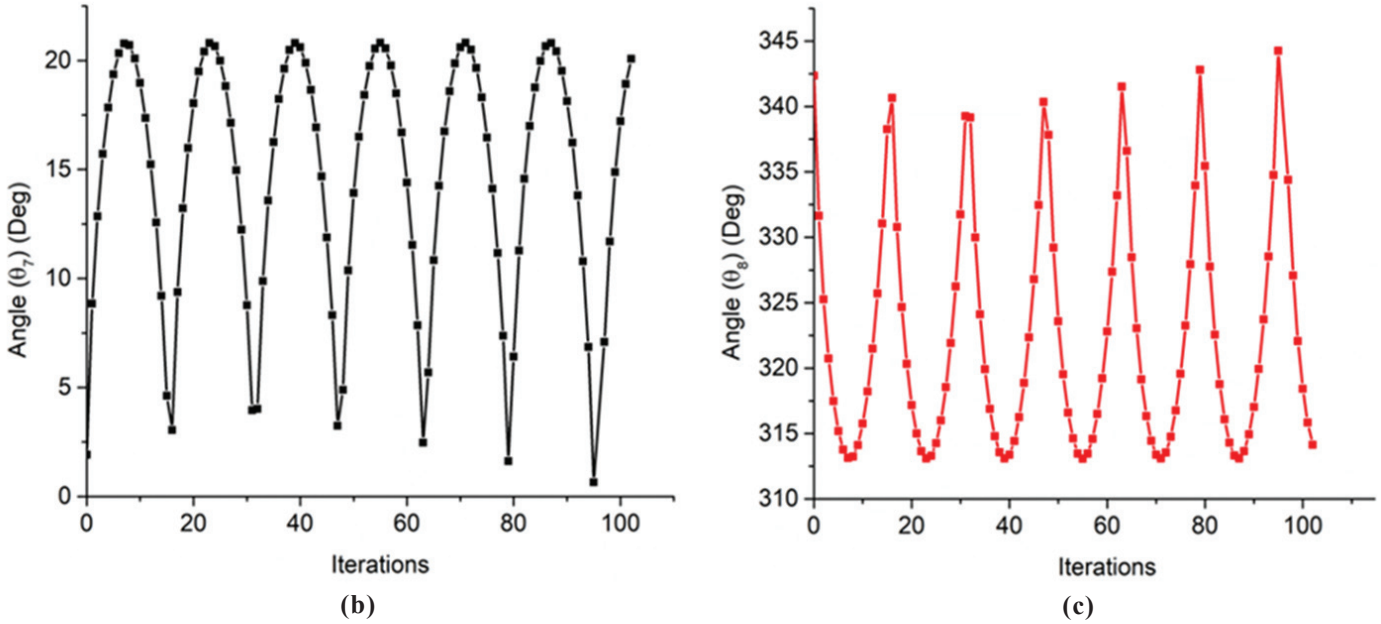


Figure 6. Multi-loop mechanism simulation: (a) Multi-loop mechanism with its second configuration and inverse mechanism, (b) Angular positions of link-7 (θ_7) vs iterations, and (c) Angular positions of link-8 (θ_8) vs iterations.

The planar multi-loop mechanism was reconstructed with custom dimensional values (that satisfy close loop property) for demonstration purposes in the current study, which is shown in Fig. 6(a) (link-1=314.4m, link-2=71.8m, link-3=179.2m, link-4=181.5m, link-5=154.1m, link-6=140m, link-7=184.1m, link-8=133.9m, link-9=317.1m) along with its other configuration and inverse mechanism generated by the simulator. Here links 1 and 9 are fixed. Link 2 is the driving link, and links 3, 4, 5, 6, 7, and 8 are driven. Since links 7 and 8 are the farthest from the driving link, their angle positions as a function of the

number of iterations are plotted in Fig. 6(b) and 6(c), and data values for ten iterations are shown in Table 5 to understand their behavior as solved from the simulator developed in this study. The results show that links 7 and 8 are in a rocker motion. Based on this analysis and simulation, the user can get various insights into the mechanism of interest, thereby aiding in making crucial design decisions.

5. CONCLUSION

In conclusion, this paper presents a novel approach for analyzing the kinematics of multiple looped N-bar closed-loop mechanisms using an iterative algorithm and a user-friendly GUI. The proposed methodology provides a cost-effective and efficient solution for modeling a wide range of generic mechanisms that require multiple interconnected loops. The algorithm utilises algebraic equations, such as the Euler form of complex numbers, to simulate and derive the unknowns necessary to characterise any closed-loop mechanism. The Python code implementation adapts to various scenarios by utilizing information on the mechanism's position, velocity, and acceleration variables. The simulation tool can display real-time color contour plots for linear and angular velocities and accelerations, simulate mechanisms with multiple loops and switch configurations, and find inverse mechanisms. Previous studies on mechanism design and synthesis were limited to specific mechanisms, and simulation tools and algorithm codes had limited built-in mechanisms, necessitating costly commercial software and high computational power. This study's proposed algorithm simplifies developing codes for analysing custom mechanisms, providing an accurate and effective solution. The case study conducted on a four-bar mechanism demonstrates the effectiveness of the simulator which can provide low-cost and user-friendly simulation results for a wide range of generic mechanisms involving multiple interconnected loops. Overall, the proposed methodology provides a cost-effective and efficient solution for analysing the kinematics of closed-loop mechanisms, enabling researchers to focus on designing and optimizing custom mechanisms.

REFERENCES

1. Khan, S.; Jamal, A.; Ali, S.; Horoub, M.M.; Albalasie, A. & Ali, S. Dynamic modeling and analysis of a four-bar mechanism for automobile applications. *In* 2nd Int. Conf. Electr. Commun. Comput. Eng. ICECCE 2020, 2020, 12-13. doi: 10.1109/ICECCE49384.2020.9179221
2. Myszka, D.H. Machines and mechanisms: Applied kinematic analysis. Pearson Prentice Hall, 2012.
3. Marques, F.; Roupa, I.; Silva, M.T.; Flores, P. & Lankarani, H.M. Examination and comparison of different methods to model closed loop kinematic chains using Lagrangian formulation with cut joint, clearance joint constraint and elastic joint approaches. *Mech. Mach. Theory*., 2021, **160**. doi: 10.1016/j.mechmachtheory.2021.104294
4. Ding, B.R.; Qu, X.H. & Chen, Y.L. Application and research of mechanical design optimization based on genetic algorithm kinematics simulation technology. *J. Intell. Fuzzy Syst.*, 2018, **34**(2), 871-878. doi: 10.3233/JIFS-169380
5. Müller, A. Semialgebraic regularisation of kinematic loop constraints in multibody system models. *J. Comput. Non-Linear Dyn.*, 2011, **6**(4). doi: 10.1115/1.4002998
6. Han, Z.; Yuan, S.; Li, X. & Zhou, J. Enhanced closed-loop systematic kinematics analysis of wheeled mobile robots. *Int. J. Adv. Robot Syst.*, 2019, **16**(4). doi: 10.1177/1729881419863242
7. Sünkün, S.; Parlak, B.O.; Yildirim, A. & Yavaşoğlu, H.A. Jansen Bağlantısının Kinematik Analizi için Araç Kutusu Tasarımı. *Comput. Sci.*, 2022. doi: 10.53070/bbd.1173829
8. Adorno, B.V. & Marques Marinho, M. DQ Robotics: A library for robot modeling and control in IEEE Robotics & Automation Magazine, 2021, **28**(3), 102-116. doi: 10.1109/MRA.2020.2997920
9. Dang, X.Z.; Zhou, L.S. & Liang, D. Modeling and simulation of kinematics for planar R-2RRP-RRR mechanism based on SimMechanics and adams. *Adv. Mater. Res.,* 2013, 753-757. doi: 10.4028/www.scientific.net/AMR.816-817.753
10. Wang, Y.; Lyu, C. & Liu, J. Kinematic analysis and verification of a new 5-dof parallel mechanism. *Appl. Sci.* (Switzerland), 2021, **11**(17). doi: 10.3390/app11178157
11. Wang, X.; Yuan, D.; Wang, X. & Wu, J. Kinematic analysis and virtual prototype simulation of the thrust mechanism for shield machine. *Appl. Sci.* (Switzerland), 2022, **12**(3). doi: 10.3390/app12031431
12. Yamamoto, T.; Iwatsuki, N. & Ikeda, I. Automated kinematic analysis of closed-loop planar link mechanisms. *Machines*, 2020, **8**(3). doi: 10.3390/MACHINES8030041
13. Bai, L.; Dong, Z.F. & Ge, X.S. The closed-loop kinematics modeling and numerical calculation of the parallel hexapod robot in space. *Adv. Mech. Eng*., 2017, **9**(2), 1-11. doi: 10.1177/1687814016688849
14. Bieze, T.M.; Largilliere, F.; Kruszewski, A.; Zhang, Z.; Merzouki, R. & Duriez, C. Finite element method-based kinematics and closed-loop control of soft, continuum manipulators. *Soft Robot*., 2018, **5**(3), 348-364. doi: 10.1089/soro.2017.0079
15. Bhattacherjee, R. & Saha, A. On a new six degree of freedom modelling method for homing missiles and its application for design/analysis. *Def. Sci. J*., 2013, **47**(3), 327-341. doi: 10.14429/dsj.47.3997
16. Virtual Labs - NITK SURATHKAL. (n.d.). https://www.vlab.co.in/participating-institute-nitk-surathkal. (Accessed on 5 January 2023).
17. Pennock, G.R. & Israr, A. Kinematic analysis and synthesis of an adjustable six-bar linkage, *Mech. Mach. Theory*., 2009, **44**(2), 306–323. doi: 10.1016/j.mechmachtheory.2008.04.007
18. Smaili, A. & Zeineddine, F. SoftLink: A matlab/simulink based code for the analysis, synthesis, optimisation and

simulation of mechanisms, *In* Proceedings of ASEE Annu. Conf., 2003, 10949–10959.
doi: 10.18260/1-2--11943
19. Schmidt, P.L. & Lax, P.A. Use of computer coding to teach design in a mechanics course, resulting in an implementation of a kinematic mechanism design tool using PYTHON. *In* Proceedings of ASEE Annu. Conf. Expo. 2018.
doi: 10.18260/1-2--31187
20. Lokesh,R.;Chittawadigi,R.G.&Saha,S.K.MechAnalyzer: 3D simulation software to teach kinematics of machines. *In* Proceedings of 2nd Int. 17th Natl. Conf. Mach. Mech. Ina., 2015, 1-9.
21. Simionescu, P.A. Simulation of repetitive mechanisms using modular kinematics, *Int. J. Mech. Robot. Syst*., 2020, **5**.
doi: 10.1504/ijmrs.2020.10034061
22. Neeraj Kumar, R.V. & Sreenivasulu, R. Inverse kinematics (IK) solution of a robotic manipulator using PYTHON. *J. Mechatron. Robot*., 2019, **3**(1), 542–551.
doi: 10.3844/jmrsp.2019.542.551
23. Sreenivasulu, R.; Chaitanya, G.; Kumar, G.V. & Devi, M.R., Inverse kinematic solution for five bar parallel linkage planar manipulator using python and optimisation by Taguchi method, *Int. J. Eng. Trends Technol*., 2021, **69**(5), 94–100.
doi: 10.14445/22315381/IJETT-V69I5P214
24. Simionescu, P. A. & Neagoe, M. Cam profiles generation as follower envelopes with MATLAB Programs. *In* Proceedings of the 2020 USCToMM Symposium on Mechanical Systems and Robotics, 2020, **83**.
doi: 10.1007/978-3-030-43929-3_3
25. Singh, A.; Kanchan, R.P. & Singh, S. A review paper on analysis and simulation of kinematics of 3R robot with the help of roboanalyzer. *Int. J. Eng. Res.*, 2016, **5**(4), 272–275.
doi: 10.17577/ijertv5is040535
26. Hussain, Z. & Siyab Khan, M. Introducing python programming for engineering scholars. *Int. J. Comput. Sci. Netw. Secur*., 2018, **18**(12), 26–33.
27. Patel, A.; Tannahill, B.; Bauerb, R. & Warkentin, A. Development of a kinematic simulation tool to study cylindrical plunge grinding. *In* Proceedings of the Canadian Society for Mechanical Engineering International Congress, 2020, **3**.
doi: 10.32393/csme.2020.113
28. Aswal, P.; Singh, R.; Kumar, R.; Bhatt, A. & Raj, T. Resolving the four – Bar link mechanism by kinamatics and revolving angle solution. *Int. J. Recent Technol. Eng.*, 2020, **8**(5), 1003–1009.
doi: 10.35940/ijrte.e6069.018520
29. Ma, N.; Yu, J.; Dong, X. & Axinte, D. Design and stiffness analysis of a class of 2-DoF tendon driven parallel kinematics mechanism. *Mech. Mach. Theory*, 2018, **129**, 202–217.
doi: 10.1016/j.mechmachtheory.2018.07.023
30. Sapietová, A.; Novák, P.; Sága, M.; Šulka, P. & Sapieta, M. Dynamic and stress analysis of a locking mechanism in the ansys workbench software environment. *Adv. Sci. Technol. Res. J*., 2019, **13**(1), 23–28.
doi: 10.12913/22998624/101601
31. ANSYS Fluent Theory Guide. ANSYS Inc.
32. Guochao, Bai.; Duanling, Li.; Shimin, Wei. & Qizheng, Liao. Kinematics and synthesis of a type of mechanisms with multiple remote centers of motion. Proceedings of the Institution of Mechanical Engineers, Part C: *J. Mech. Eng. Sci.,* 2014*,* **228**(18)*,* 3430-3440.
doi: 10.1177/0954406214527915
33. Mohamed, M.G. & Duffy, J. The rotational geometric influence coefficients of a planar multi-loop mechanism. *Appl. Math. Model*., 1999, **23**(2), 161–174.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of SOLVE, The Virtual Lab @ NITK (Grant number: No.F.16-35/2009-DL, facility provided by Centre for System Design (CSD): A Centre of Excellence (http://csd.nitk.ac.in/) at National Institute of Technology Karnataka, India

CONTRIBUTORS

Mr Rahul V.M. is a BTech undergraduate student from the National Institute of Technology Karnataka, Surathkal. His expertise includes machine learning, deep learning, FEA simulation, CFD simulation, and simulator development using coding. He has grown to have a strong interest in modeling and simulation. Research interests include: Creation of cutting-edge simulation tools and methodologies for the resolution of challenging engineering issues.
His contributions to the present study include: Development of the simulator, code for the simulator, methodology, and algorithms. Further, carry out FEA simulation in ANSYS to analyse and compare the results for accuracy.

Mr Sandesh Bhaktha B. obtained his MTech degree in Manufacturing Engineering and Technology from Manipal Institute of Technology, Manipal in 2017 and currently pursuing PhD from National Institute of Technology Karnataka, Surathkal. His research areas include: Automotive NVH, smart materials, finite element solutions and electrical mobility.
His contributions in the present study include: Design, execution of FEA simulations and analysis of results and discussions.

Dr Gangadharan K.V. obtained his PhD from IIT Madras. He is currently working as a Professor in the Department of Mechanical Engineering, National Institute of Technology Surathkal, Karnataka. His areas of research include: System design, vibration and its control, smart material and its applications in vibration control, dynamics, and finite element analysis, condition monitoring and experimental methods in vibration.
His contributions in the present study include: Project guidance, motivation, discussion regarding the results obtained, preparation of the manuscript.