scientific reports

OPEN

Optimized sliding mode controller for trajectory tracking of flexible joints three-link manipulator with noise in input and output

Muhammad I. Azeez[1]✉, A. M. M. Abdelhaleem[1], S. Elnaggar[1], Kamal A. F. Moustafa[2] & Khaled R. Atia[1]

The aim of this study is to enhance the performance of a nonlinear three-rigid-link maneuver (RLM) in terms of trajectory tracking, disturbance and noise cancellation, and adaptability to joint flexibility. To achieve this, an optimized sliding mode controller with a proportional integral derivative surface (SMC-PID) is employed for maneuver control. An improved artificial bee colony algorithm with multi-elite guidance (MGABC) is utilized to obtain optimal values for the sliding surface and switching mode gain and attain the best performance for the robot maneuver system. The selection of the MGABC algorithm is based on its efficient exploration and exploitation techniques. The performance of the optimized SMC-PID robotic system is compared against other optimization algorithms found in existing literature, including Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Artificial Bee Colony (ABC), Ant Lion Optimizer (ALO), and Grey Wolf Optimizer (GWO). The implemented controller effectively reduces the tracking error to 0.00691 radians, eliminates chattering phenomena in the control effort, and demonstrates robustness against disturbances and noise. The controller ensures that the objective function (OBJF) is minimized, with 0.954% increase in OBJF under low disturbance and noise conditions and 14.55% under severe disturbance and noise conditions. Moreover, the optimized controller exhibits resilience to variations in payload mass analysis, with the percentage increase in OBJF values ranging from 5.726% under low uncertainty conditions to 18.887% under severe uncertainty conditions. Flexible-link maneuvers (FLM) offer advantages such as improved safety and increased operating speeds in real-world applications. In this study, we investigated the impact of joint flexibility on the performance of the FLM system. Our proposed controller demonstrated superior tracking performance, characterized by minimal vibrations in the movement of the end effector.

Robotic maneuvers have gained significant prominence in the contemporary industrial landscape, contributing to improved product quality, enhanced productivity, increased accuracy, heightened speed, and enhanced flexibility within work environments. These maneuvers are increasingly employed in hazardous, monotonous, or repetitive industrial processes, despite the complex and demanding nature of industrial applications. Material picking and placing operations are among the fundamental applications of mechanical robotic maneuvers. Furthermore, industrial maneuvers find utility in welding, assembling, manufacturing, painting, and various operations within the automotive industry, as well as in the handling of radioactive and biohazardous materials during robotically assisted surgery. The imperative for enhanced efficiency and high-quality manufacturing has driven the pursuit of more robust and sophisticated robotic maneuvers[1–4].

The mathematical modeling is known to present considerable difficulties and time constraints, especially when dealing with complex systems featuring strong interdependencies and multiple inputs and outputs. Recent studies have conducted a thorough and comprehensive examination of diverse mathematical modeling approaches, aiming to shed light on their strengths and limitations[2]. Moreover, a new simulation technique for physical modeling was compared to traditional mathematical methods for dynamic modeling of multi-DOF robotic

[1]Mechanical Design and Production Engineering Department, Zagazig University, Zagazig 44519, Egypt. [2]Industrial Engineering Department, Zagazig University, Zagazig 44519, Egypt. ✉email: mohibrahiem@eng.zu.edu.eg; moazezian97@gmail.com

manipulators, offering valuable insights into the distinctive mathematical characteristics associated with the implementation of Simscape software[1,5–9].

Simscape Multibody models, also known as SimMechanics models, have gained popularity due to their ease of access, simulation capabilities, and control feasibility[1,8]. The utilization of simulation software provides several advantages, such as gaining insights into the behavior of multi-degree-of-freedom (DOF) robotic maneuver systems in a simulated environment and circumventing the complexities associated with mathematical formulations. Additionally, SimMechanics offers valuable tools for analyzing the flexibility of joints and links, which can be challenging to achieve using traditional mathematical derivation methods. However, a literature review reveals that this specific area requires further attention, as limited research has been conducted in this domain.

Alandoli et al.[1] investigated the modeling and control of Flexible Link Manipulators (FLMs) using the Finite Element Method (FEM), and the dynamic equation was derived using Euler–Lagrange's equation. To validate the accuracy of the mathematical model, a Simscape model was developed, and the open-loop responses of both models were observed to exhibit similar and concurrent increases. This validation process confirms the reliability of the Simscape model for the FLM system.

The validation of the system modeling was performed by designing a 3DOF rigid-link articulated manipulator in Simscape and comparing it with the analytically modeled system[7]. The outputs obtained from both simulations exhibited complete similarity, indicating a high level of agreement between the Simscape model and the analytically modeled system.

The primary goal of this study[10] was directed towards tackling the obstacles linked to dynamic modeling. of a 7-DOF hydraulically actuated tele-operated robotic manipulator. The objective was to create a robust simulator capable of accurately deriving the arm's dynamic and kinematic characteristics. To achieve this, a numerical model was developed using Simulink, incorporating the SimMechanics and Simscape toolboxes. Notably, the model accounted for uncertain parameters, allowing for a more comprehensive and realistic representation of the system's behavior.

In this study[6], the primary emphasis was on utilizing Sim-Mechanics software as a valuable tool for modeling multi-DOF robotic manipulators in a manner that does not require explicit mathematical equations or relations. The Sim-Mechanics block model was employed to analyze variations in reaction forces and reaction torques acting on both 2DOF and 3DOF planar robotic manipulators.

Controlling robot manipulators is an intriguing field of study due to the complex dynamics involved. The dynamical analysis of robotic models aims to understand the relationship between the positions of the robotic arm and the joint torques applied by the actuators. Achieving accurate and reliable control is challenging due to the inherent coupling relations and nonlinear dynamics of the system. Conventional control techniques that rely on the dynamics of the robotic system struggle to address model uncertainties, including dynamic parameters (such as inertia and payload conditions), dynamic effects (such as complex nonlinear frictions), and unmodeled dynamics[11,12].

The sliding mode control (SMC) technique has gained significant popularity in uncertain systems due to its numerous benefits, including robustness, order reduction, ease of implementation, strong resistance to external disturbances, low sensitivity to changes in system parameters, and design simplicity. To apply SMC design, a sliding surface must first be established to ensure the desired convergence property. Subsequently, an SMC is constructed to guide the system states towards the selected position, unaffected by uncertainty or disturbances. The underlying concept of SMC is to maintain the control signal on the sliding surface by inducing it to move in that direction[11,13–16].

However, the discontinuous nature of the SMC law can lead to a chattering effect, characterized by high-frequency oscillations of the controlled variable when the system state reaches the sliding surface. This chattering effect can have detrimental consequences on actuator control and excite unwanted unmodeled dynamics[13,14].

Recent research has been dedicated to the investigation of proportional integral derivative sliding mode controllers (SMC-PIDs) as robust nonlinear controllers. In the context of a three-phase active power filter, an adaptive dynamic global sliding mode controller with a proportional integral derivative (PID) sliding surface has been employed. This controller aims to achieve overall robustness, minimize steady-state error, and enhance system response speed[17].

To optimize the control parameters of the fuzzy controller in the Adaptive PID Fuzzy Sliding Mode Control (APIDFSMC) for handling uncertain systems, a GA-based adaptive PID fuzzy sliding mode control strategy (APIDFSMC-GA) has been introduced[18].

For uncertain nonlinear systems, an adaptive robust controller has been developed using a proportional-integral-derivative-type nonsingular fast terminal sliding mode control. This controller aims to provide essential characteristics such as rapid transient response, finite-time convergence, negligible steady-state error, and chattering cancellation[19].

In the control of a nonlinear 2-DOF system using a sliding mode controller (SMC) with a PID surface, various optimization techniques have been explored. These techniques include the Antlion Optimization Algorithm (ALO), Sine Cosine Algorithm (SCA), Grey Wolf Optimizer (GWO), and Whale Optimizer Algorithm (WOA)[14].

The selection of an optimal sliding surface and switching mode controller's gain is a crucial aspect in enhancing the performance and efficiency of a sliding mode controller (SMC) with a proportional integral derivative (PID) surface. Traditional optimization methods often fail to deliver satisfactory results when optimizing these parameters. In the realm of scientific research and technical applications, there has been a notable surge in the utilization of evolutionary algorithms (EAs) as efficient and effective optimization approaches. This rise in popularity is primarily due to the fact that modern practical optimization problems tend to be non-convex, discontinuous, or even non-differentiable, posing significant challenges for traditional optimization techniques such as gradient-based approaches[20–23].

Swarm-based optimization algorithms (SOAs) employ natural processes to guide the search for optimal solutions. Unlike simple optimization algorithms such as stochastic process and hill climbing, SOAs utilize a population of solutions instead of a single solution per iteration. This distinction is significant, as a population of solutions is processed and created at each iteration[24].

Evolutionary algorithms encompass various unique paradigms, including Jellyfish Search Optimization (JSO), Glowworm Swarm Optimization (GSO), White Whale Optimization (WHO), Particle Swarm Optimization (HSO), Grey Wolf Optimization (GWO), Genetic Algorithm (GA), and Artificial Bee Colony (ABC) algorithm.

Among these, the Artificial Bee Colony (ABC) algorithm stands out as a relatively recent swarm intelligence-based optimization paradigm, known for its simplicity and effective performance. The basic ABC approach, initially proposed by Karaboga[20], mimics the foraging behavior of a bee colony to maximize honey collection from the search space and identify the optimal food source. The ABC algorithm has exhibited superior performance compared to other well-known evolutionary algorithms, including FA, ACO, and CSO, across various common objective functions[25–28].

The ABC algorithm comprises three types of bees: employed, onlooker, and scouting bees. Employed bees search for food at specific locations, gather it, and return to the hive. They perform a waggle dance, which conveys information about the location, proximity, and fitness of a flower patch, enabling efficient communication within the colony. This information facilitates the selection of promising flower patches without the need for external guides[24,29]. Food supplies are distributed among neighbors in each iteration, generating novel solutions that are evaluated using the fitness function. Follower bees are dispatched to more promising regions, while abandoned food sources are replaced. This enables the colony to collect food swiftly and effectively[20,30]. However, like other evolutionary algorithms, the ABC algorithm faces challenges related to early convergence and slow convergence rates when tackling complex optimization problems[23,26,31]. The exploration technique equation in the basic ABC algorithm performs well in searching for new food sources but exhibits limitations during exploitation, as highlighted by various studies in the ABC community[28,32,33]. Consequently, there has been significant focus on maximizing exploitation while enhancing the performance of the ABC algorithm. Balancing exploitation and exploration is crucial for evolutionary algorithms, as these two aspects are inherently antagonistic. Achieving a proper balance between the two is challenging but essential for boosting the performance of the ABC algorithm[23,26,31].

The MGABC algorithm refers to an enhanced version of the Artificial Bee Colony (ABC) algorithm that incorporates multi-elite guidance to improve its performance[28]. The enhancements introduced in MGABC involve two key modifications. Firstly, the neighborhood search method has been enhanced to improve the search process within the algorithm. Secondly, two new food search methods have been developed for the exploration and exploitation stages of the algorithm. These methods rely on a selected set of superior food sources, known as the elite solutions. The inclusion of this group of elite solutions aims to maximize exploitation while ensuring that exploration is not compromised, leveraging the available information to strike a balance between the two objectives.

The present study focuses on utilizing the Lagrangian method and the Simscape multibody toolbox in MATLAB to develop a model for a 3-DOF rigid link maneuver and investigate the effectiveness of the proposed model in a representative multi-degree-of-freedom system, while keeping the complexity manageable for a comprehensive analysis. In order to validate the model, the open-loop system responses of both the mathematical and Simscape models are examined in section "Modelling the three-link robotic maneuver". The proposed approach of using a sliding mode controller (SMC) with a proportional integral derivative (PID) sliding surface is presented in section "Sliding mode controller with PID surface". Section "Metaheuristic optimization algorithm" elaborates on the enhanced artificial bee colony with multi-elite guidance (MGABC) metaheuristic optimization technique employed in the study. The optimization process using MGABC to enhance the overall robustness, stability, and tracking performance of the robotic system is discussed in section "Controller optimization for trajectory tracking". A comprehensive performance analysis is conducted in section "Detailed simulated performance evaluation" to evaluate the proposed controller's capability in mitigating disturbances and noise, as well as its robustness to variations in payload mass. Additionally, the adaptability of the controller to the flexibility of the three joints of the maneuver is examined. Finally, section "Conclusion" provides a summary of the concluding remarks of the study.

Throughout the entire study, bold lowercase letters are used to represent vectors, while uppercase letters are used for matrices.

Modelling the three-link robotic maneuver

The decision to select a 3 DOF robotic maneuver for our study was driven by the objective of examining and showcasing the efficacy of the proposed model in a representative multi-DOF system. This choice was made while considering the need to maintain a manageable level of complexity for a comprehensive analysis. It is important to note that the first 3 angles of a robotic manipulator are often considered the most critical DOFs in industrial robots. This is because the remaining DOFs are typically incorporated in the end-effector, which generally exhibits less dynamics complexity compared to the base structure of the manipulator.

The purpose of the modeling is to develop a mathematical representation of the system that accurately describes its behavior and dynamics. The model will be derived using established mathematical methods and principles, such as the Lagrangian method, which considers the kinetic and potential energies of the links. The model will be designed to capture the complexity of the system and its interactions, incorporating various variables and assumptions, as necessary.

As the complexity of a problem increases, the need for variables, assumptions, and iterations also increases, resulting in prolonged computing times. However, Simscape Multibody, which is seamlessly integrated with

MATLAB and Simulink, offers a potent solution for rapidly modeling, simulating, and analyzing systems. The tool is capable of accurately simulating both processes and systems, allowing for the analysis and prediction of system behavior. The model is constructed by arranging blocks from the Simulink library on the diagram canvas, endowing Simulink with an inherent versatility that makes it suitable for a wide range of applications. Additionally, Simulink serves as a valuable tool for testing and verifying models, enabling the simulation of a model's behavior under diverse conditions. This functionality can be exploited to evaluate the limits of a model and locate potential errors. The subsequent subsections will also encompass the Simscape modeling and validation of the mathematical model using the Simscape model.

Mathematical modelling. Let us consider the three rigid-link planar robotic maneuver model shown in Fig. 1 and the mathematical model of the robotic maneuver with three rotating rigid links can be described by a second order nonlinear ordinary differential equations as shown in Eq. (1) by applying Lagrange's principle[3,12,34].

In the given system, $\boldsymbol{\theta}(\boldsymbol{t}), \dot{\boldsymbol{\theta}}(\boldsymbol{t})$, and $\ddot{\boldsymbol{\theta}}(\boldsymbol{t})$ represent the vectors of joint angles, joint velocities, and joint accelerations, respectively, for the three links. The 3×3 matrix represents the mass inertia matrix, which is symmetric and positive definite. $\boldsymbol{C}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right)$ and $\boldsymbol{G}(\boldsymbol{\theta})$ represent the vectors of centrifugal and Coriolis forces, and gravitational forces, respectively. Additionally, $\boldsymbol{\tau}(\boldsymbol{t})$ represents the vectors of torques applied at each joint of the three links.

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} C_1\left(\theta, \dot{\theta}\right) \\ C_2\left(\theta, \dot{\theta}\right) \\ C_3\left(\theta, \dot{\theta}\right) \end{bmatrix} + \begin{bmatrix} G_1(\theta) \\ G_2(\theta) \\ G_3(\theta) \end{bmatrix} = \begin{bmatrix} \tau_1(t) \\ \tau_2(t) \\ \tau_3(t) \end{bmatrix} \tag{1}$$

where

$$\begin{aligned} m_{11} = {} & l_1^2(0.25m_1 + m_2 + m_3) + l_2^2(0.25m_2 + m_3) + 0.25l_3^2m_3 + l_1l_2\ cos(\theta_2)(m_2 + 2m_3) \\ & + l_2l_3m_3\ cos(\theta_3) + l_1l_3m_3\ cos(\theta_2 + \theta_3) + I_1 + I_2 + I_3 \end{aligned} \tag{2}$$

$$\begin{aligned} m_{12} = {} & l_2^2(0.25m_2 + m_3) + 0.25l_3^2m_3 + l_1l_2\ cos(\theta_2)(m_3 + 0.5m_2) + l_2l_3m_3\ cos(\theta_3) \\ & + 0.5l_1l_3m_3\ cos(\theta_2 + \theta_3) + I_2 + I_3 \end{aligned} \tag{3}$$

$$m_{13} = 0.25l_3^2m_3 + 0.5l_2l_3m_3\ cos(\theta_3) + 0.5l_1l_3m_3\ cos(\theta_2 + \theta_3) + I_3 \tag{4}$$

$$\begin{aligned} m_{21} = {} & l_2^2(0.25m_2 + m_3) + 0.25l_3^2m_3 + l_1l_2\ cos(\theta_2)(m_3 + 0.5m_2) + l_2l_3m_3\ cos(\theta_3) \\ & + 0.5l_1l_3m_3\ cos(\theta_2 + \theta_3) + I_2 + I_3 \end{aligned} \tag{5}$$

$$m_{22} = l_2^2(0.25m_2 + m_3) + 0.25l_3^2m_3 + l_2l_3m_3\ cos(\theta_3) + I_2 + I_3 \tag{6}$$

$$m_{23} = 0.25l_3^2m_3 + 0.5l_2l_3m_3\ cos(\theta_3) + I_3 \tag{7}$$

$$m_{31} = 0.25l_3^2m_3 + 0.5l_2l_3m_3\ cos(\theta_3) + 0.5l_1l_3m_3\ cos(\theta_2 + \theta_3) + I_3 \tag{8}$$

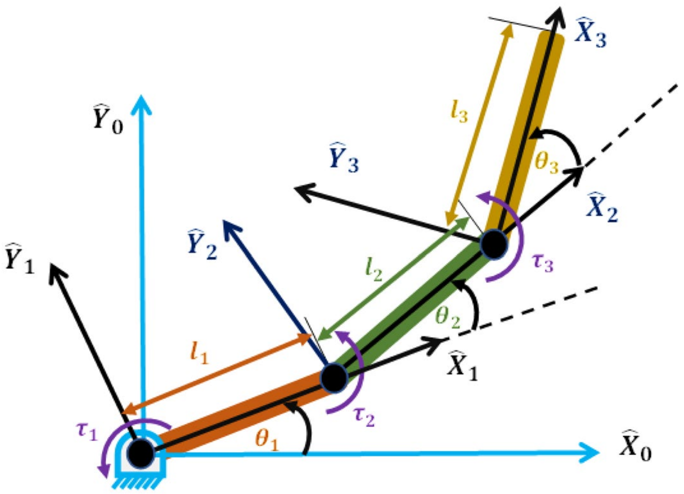


Figure 1. The three DOF robotic maneuver model.

$$m_{32} = 0.25 l_3^2 m_3 + 0.5 l_2 l_3 m_3\ cos(\theta_3) + I_3 \tag{9}$$

$$m_{33} = 0.25 l_3^2 m_3 + I_3 \tag{10}$$

$$\begin{aligned} C_1\left(\theta,\dot{\theta}\right) = {} & \left(\dot{\theta}_2\right)^2(-0.5 l_1 l_2 m_2\ sin(\theta_2) - l_1 l_2 m_3\ sin(\theta_2) - 0.5 l_1 l_3 m_3\ sin(\theta_2+\theta_3)) \\ & - (\dot{\theta}_3)^2(0.5 l_2 l_3 m_3\ sin(\theta_3) - 0.5 l_1 l_3 m_3\ sin(\theta_2+\theta_3)) \\ & + \dot{\theta}_1\dot{\theta}_2(-l_1 l_2 m_2\ sin(\theta_2) - 2 l_1 l_2 m_3\ sin(\theta_2) - l_1 l_3 m_3\ sin(\theta_2+\theta_3)) \\ & + \dot{\theta}_2\dot{\theta}_3(-l_1 l_3 m_3\ sin(\theta_2+\theta_3) - l_2 l_3 m_3\ sin(\theta_3)) \\ & + \dot{\theta}_1\dot{\theta}_3(-l_2 l_3 m_3\ sin(\theta_3) - l_1 l_3 m_3\ sin(\theta_2+\theta_3)) \end{aligned} \tag{11}$$

$$\begin{aligned} C_2\left(\theta,\dot{\theta}\right) = {} & \left(\dot{\theta}_1\right)^2(0.5 l_1 l_2 m_2\ sin(\theta_2) + l_1 l_2 m_3\ sin(\theta_2) + 0.5 l_1 l_3 m_3\ sin(\theta_2+\theta_3)) \\ & - 0.5(\dot{\theta}_3)^2(l_2 l_3 m_3\ sin(\theta_3)) - \dot{\theta}_1\dot{\theta}_3(l_2 l_3 m_3\ sin(\theta_3)) - \dot{\theta}_2\dot{\theta}_3(l_2 l_3 m_3\ sin(\theta_3)) \end{aligned} \tag{12}$$

$$\begin{aligned} C_3\left(\theta,\dot{\theta}\right) = {} & \left(\dot{\theta}_1\right)^2(0.5 l_2 l_3 m_3\ sin(\theta_3) + 0.5 l_1 l_3 m_3\ sin(\theta_2+\theta_3)) + 0.5\left(\dot{\theta}_2\right)^2(l_2 l_3 m_3\ sin(\theta_3)) \\ & + \dot{\theta}_1\dot{\theta}_2(l_2 l_3 m_3\ sin(\theta_3)) \end{aligned} \tag{13}$$

$$\begin{aligned} G_1(\theta) = {} & m_3 g(l_1\ cos(\theta_1) + 0.5 l_3\ cos(\theta_1+\theta_2+\theta_3) + l_2\ cos(\theta_1+\theta_2)) \\ & + m_2 g(l_1\ cos(\theta_1) + 0.5 l_2\ cos(\theta_1+\theta_2)) + 0.5 m_1 g l_1\ cos(\theta_1) \end{aligned} \tag{14}$$

$$G_2(\theta) = m_3 g(0.5 l_3\ cos(\theta_1+\theta_2+\theta_3) + l_2\ cos(\theta_1+\theta_2)) + 0.5 m_2 g l_2\ cos(\theta_1+\theta_2) \tag{15}$$

$$G_3(\theta) = 0.5 m_3 g l_3\ cos(\theta_1+\theta_2+\theta_3) \tag{16}$$

Simscape multibody model.

Within the Simulink® environment, Simscape™ enables the quick construction of physical model systems. Simscape allows for the creation of physical component models that are based on physical connections and easily interact with block diagrams and other modelling paradigms. Simscape add-on products give users access to more sophisticated parts and analysis tools. Simscape aids in the creation of control systems and the testing of system performance. Using MATLAB variables and expressions, we can parameterize our models, and Simulink can be used to create the control schemes for our physical system.

The Simscape model, which represents the entire rigid link maneuver (RLM) system is shown in Fig. 2. All the physical parameters of the components that are generated with the MATLAB Simscape toolbox are listed in Table 1.

For the purpose of validating the mathematical model, a comparison between the open loop performance of the mathematical model and the Simscape model is developed when a step input signal is applied to the first joint as shown in Fig. 3.

The response of the mathematical model compared with the response of the Simscape model is shown in Fig. 4. It is clearly obvious that the two model are identical, hence we can use the Simscape model in the remaining of this study and utilize its great facilities.

While our study focused on a 3 DOF system, the proposed model can be effectively applied to other DOF systems as well. The methodology employed in our approach can be extended to systems with higher degrees of freedom, with appropriate modifications and considerations for the increased complexity. However, it is important to note that the applicability of the proposed model to different DOF systems may depend on various

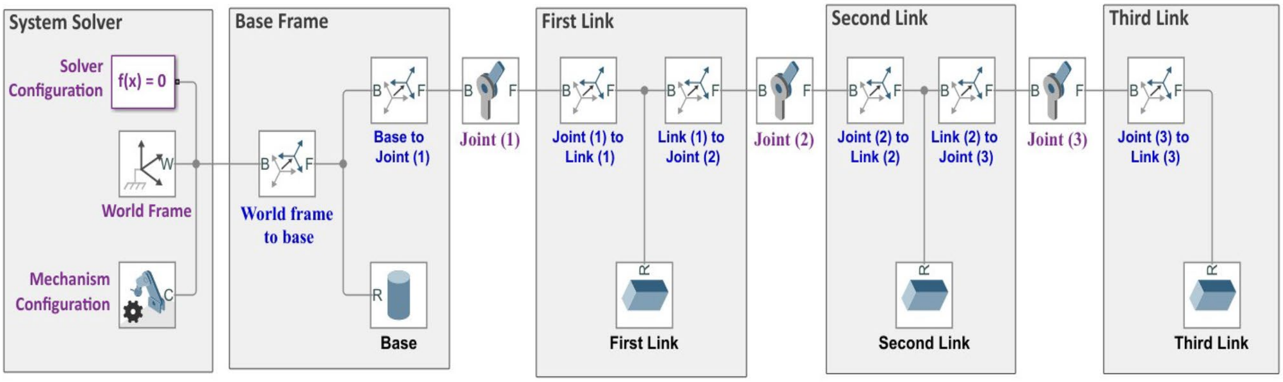


Figure 2. Simscape multibody model of the 3-DOF robotic maneuver.

Parameters	Link 1	Link 2	Link 3
Mass	1 kg	1 kg	1 kg
I_{zz}	0.0033 kg m^2	0.0033 kg m^2	0.0033 kg m^2
Length	0.2 m	0.2 m	0.2 m
Cross-section area	2.2 × 2.2 cm^2	2.2 × 2.2 cm^2	2.2 × 2.2 cm^2

Table 1. The physical parameters of the links.

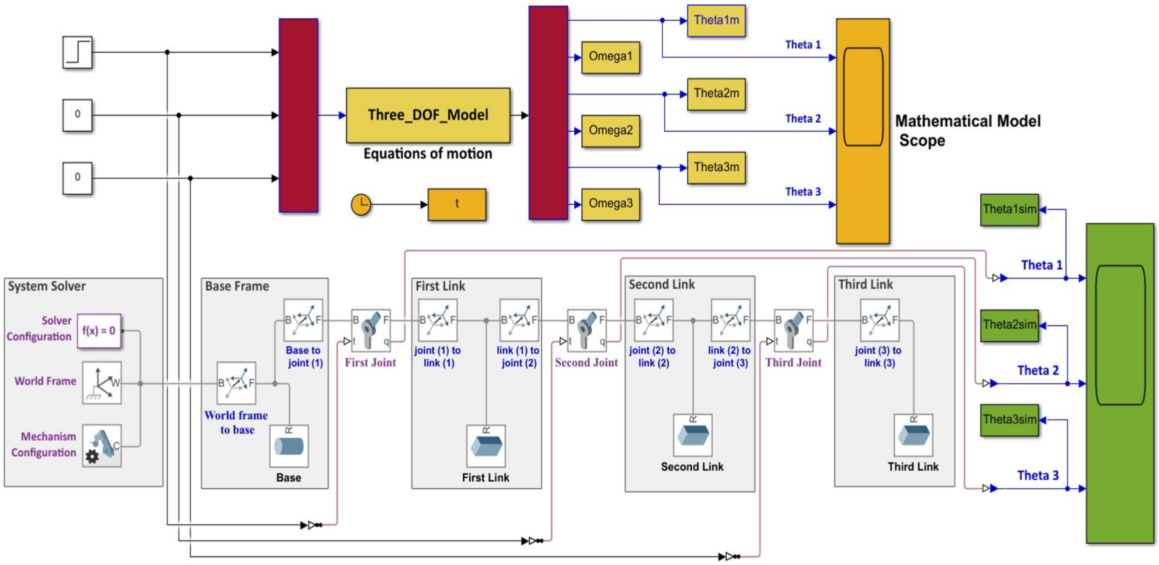


Figure 3. Step input signal applied to the first joint of the mathematical and Simscape models.

factors, such as system dynamics, mechanical constraints, and control objectives. Therefore, further investigation and adaptation of the model may be necessary to ensure its effectiveness in specific DOF configurations.

Sliding mode controller with PID surface

Before designing the sliding mode controller, it is essential to determine the appropriate sliding surface function. Subsequently, an equivalent controller and switching mode controller need to be constructed based on the system's states. The fundamental principle underlying this type of control is to ensure that the system's representative point, regardless of its initial conditions, resides on a hypersurface within the phase space. This hypersurface represents a collection of static relationships among the state variables, allowing for robust control of the system”s evolution. There are two components in the sliding mode control as described in Eq. (17).

$$\boldsymbol{\tau}(\boldsymbol{t}) = \boldsymbol{\tau}_{\boldsymbol{eq}}(\boldsymbol{t}) + \boldsymbol{\tau}_{\boldsymbol{sm}}(\boldsymbol{t}) \tag{17}$$

where $\boldsymbol{\tau}_{\boldsymbol{eq}}(\boldsymbol{t})$ is the continuous component and called equivalent controller and $\boldsymbol{\tau}_{\boldsymbol{sm}}(\boldsymbol{t})$ is the discontinuous component called switching mode controller.

The PID sliding surface for the SMC of each link can be specified with Eq. (18).

$$\boldsymbol{s}(\boldsymbol{t}) = K_p\boldsymbol{e}(\boldsymbol{t}) + K_I \int_0^t \boldsymbol{e}(\boldsymbol{t})dt + K_d\dot{\boldsymbol{e}}(\boldsymbol{t}) \tag{18}$$

The diagonal matrices K_p, K_I and K_d represent the proportional, integral, and derivative gains of the PID surface, respectively. Additionally, the error signal $\boldsymbol{e}(\boldsymbol{t})$ is defined as the difference between the desired angular position $\boldsymbol{\theta}_{\boldsymbol{d}}(\boldsymbol{t})$ and the measured angular position $\boldsymbol{\theta}(\boldsymbol{t})$. Similarly, the derivative of the error signal $\dot{\boldsymbol{e}}(\boldsymbol{t})$ is obtained by subtracting the measured angular speed $\dot{\boldsymbol{\theta}}(\boldsymbol{t})$ from the desired angular speed $\dot{\boldsymbol{\theta}}_{\boldsymbol{d}}(\boldsymbol{t})$. These variables, namely $\boldsymbol{\theta}_{\boldsymbol{d}}(\boldsymbol{t})$, $\boldsymbol{\theta}(\boldsymbol{t})$, $\dot{\boldsymbol{\theta}}_{\boldsymbol{d}}(\boldsymbol{t})$, and $\dot{\boldsymbol{\theta}}(\boldsymbol{t})$, correspond to the vectors of desired angular position, measured angular position, desired angular speed, and measured angular speed of the robotic maneuver, respectively.

To calculate the continuous equivalent controller $\boldsymbol{\tau}_{\boldsymbol{eq}}(\boldsymbol{t})$ it is necessary that $\dot{\boldsymbol{s}}(\boldsymbol{t}) = 0$, so that we have the following:

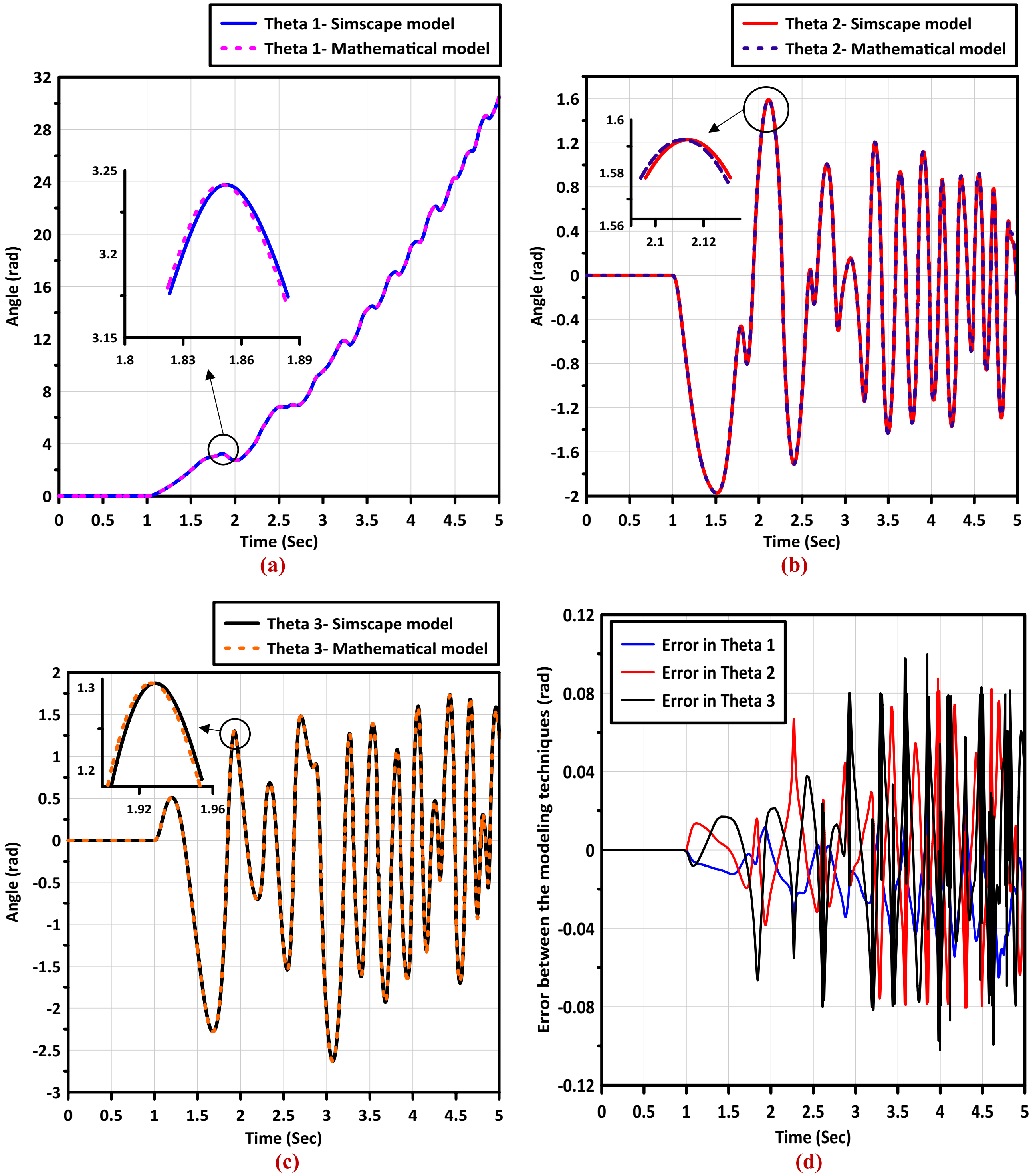


Figure 4. The response of both the mathematical and the Simscape models for: (**a**) the first link, (**b**) the second link, (**c**) the third link, and (**d**) the error in angles between the Simscape and mathematical models.

$$
\begin{aligned}
&K_p\dot{\boldsymbol{e}}(\boldsymbol{t}) + K_d\ddot{\boldsymbol{e}}(\boldsymbol{t}) + K_I\boldsymbol{e}(\boldsymbol{t}) = 0 \\
&\quad \ddot{\boldsymbol{e}}(\boldsymbol{t}) = -K_d^{-1}\left(K_p\dot{\boldsymbol{e}}(\boldsymbol{t}) + K_I\boldsymbol{e}(\boldsymbol{t})\right) = \ddot{\boldsymbol{\theta}}_{\boldsymbol{d}}(\boldsymbol{t}) - \ddot{\boldsymbol{\theta}}(\boldsymbol{t}) \\
&\quad \ddot{\boldsymbol{\theta}}(\boldsymbol{t}) = \ddot{\boldsymbol{\theta}}_{\boldsymbol{d}}(\boldsymbol{t}) + K_d^{-1}\left(K_p\dot{\boldsymbol{e}}(\boldsymbol{t}) + K_I\boldsymbol{e}(\boldsymbol{t})\right) = M(\theta)^{-1}\left(\boldsymbol{\tau}_{\boldsymbol{eq}}(\boldsymbol{t}) - \boldsymbol{C}\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right) - \boldsymbol{G}(\boldsymbol{\theta})\right) \\
&\boldsymbol{\tau}_{\boldsymbol{eq}}(\boldsymbol{t}) = M(\theta)\left(\ddot{\boldsymbol{\theta}}_{\boldsymbol{d}}(\boldsymbol{t}) + K_d^{-1}\left(K_p\dot{\boldsymbol{e}}(\boldsymbol{t}) + K_I\boldsymbol{e}(\boldsymbol{t})\right)\right) + \boldsymbol{C}\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right) + \boldsymbol{G}(\boldsymbol{\theta})
\end{aligned}
\tag{19}
$$

The switching mode control law $\boldsymbol{\tau}_{\boldsymbol{sm}}(\boldsymbol{t})$ is chosen to be of the following form:

$$\boldsymbol{\tau}_{\boldsymbol{sm}}(\boldsymbol{t}) = M(\theta)\lambda\mathrm{sign}(\boldsymbol{s}(\boldsymbol{t})) \tag{20}$$

where λ is a diagonal matrix of positive elements λ_{ii}.

The Lyapunov function is defined as following to construct the discontinuous switching mode controller and ensure the stability of the system[35].

$$V = \frac{1}{2}\boldsymbol{s}^{\boldsymbol{T}}(\boldsymbol{t})\boldsymbol{s}(\boldsymbol{t}) \tag{21}$$

By differentiating the Lyapunov function with respect to time along the trajectory of the system and utilizing Eqs. (19) and (20), the following expression is obtained.

$$\begin{aligned}\dot{V} &= \boldsymbol{s}^{\boldsymbol{T}}(\boldsymbol{t})\dot{\boldsymbol{s}}(\boldsymbol{t}) = \boldsymbol{s}^{\boldsymbol{T}}(\boldsymbol{t})\left(K_p\dot{\boldsymbol{e}}(\boldsymbol{t}) + K_d\ddot{\boldsymbol{e}}(\boldsymbol{t}) + K_I\boldsymbol{e}(\boldsymbol{t})\right)\\ &= \boldsymbol{s}^{\boldsymbol{T}}(\boldsymbol{t})\left(K_p\dot{\boldsymbol{e}}(\boldsymbol{t}) + K_I\boldsymbol{e}(\boldsymbol{t}) + K_d\left(\ddot{\boldsymbol{\theta}}_{\boldsymbol{d}}(\boldsymbol{t}) - M(\theta)^{-1}\left(\boldsymbol{\tau}_{\boldsymbol{eq}}(\boldsymbol{t}) + \boldsymbol{\tau}_{\boldsymbol{sm}}(\boldsymbol{t}) - \boldsymbol{C}\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right) - \boldsymbol{G}(\boldsymbol{\theta})\right)\right)\right)\\ &= -\boldsymbol{s}^{\boldsymbol{T}}(\boldsymbol{t})\ K_d\lambda\mathrm{sign}(\boldsymbol{s}(\boldsymbol{t}))\\ &\leq -\lambda_m\{K_d\lambda\boldsymbol{s}(t)\}\end{aligned} \tag{22}$$

where $\lambda_m\{\cdot\}$ is the minimum eigen value of the product $K_d\lambda$.

According to the last equation, the system trajectories will be pushed toward the sliding surface in finite time[36]. Since, the signum function causes chattering, we will replace it with the saturated vector function $\boldsymbol{tanh}(\boldsymbol{s}(\boldsymbol{t}))$. Therefore, the new control law will be written as following:

$$\boldsymbol{\tau}_{\boldsymbol{sm}}(\boldsymbol{t}) = M(\theta)\lambda\tanh\left(\frac{\boldsymbol{s}(\boldsymbol{t})}{\delta}\right) \tag{23}$$

where δ is a small positive gain, in this study δ is chosen to be 0.1.

Finally, the total control law is given by the following equation:

$$\boldsymbol{\tau}(\boldsymbol{t}) = M(\theta)\left(\ddot{\boldsymbol{\theta}}_{\boldsymbol{d}}(\boldsymbol{t}) + K_d^{-1}\left(K_p\dot{\boldsymbol{e}}(\boldsymbol{t}) + K_I\boldsymbol{e}(\boldsymbol{t})\right) + \lambda\tanh\left(\frac{\boldsymbol{s}(\boldsymbol{t})}{\delta}\right)\right) + \boldsymbol{C}\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right) + \boldsymbol{G}(\boldsymbol{\theta}) \tag{24}$$

The elements of the diagonal matrices $k_{p_{ii}}$, $k_{I_{ii}}$, $k_{d_{ii}}$ and λ_{ii} are the parameters to be tuned with the metaheuristic optimization algorithms.

Metaheuristic optimization algorithm

Three distinct bees make up the honeybee colony in the bee colony: employed bees, onlooker bees, and scouter bees. Each kind of bee is responsible for a particular duty. The employed bees might be regarded of as playing an exploratory role because they are in charge of searching for nutrient-rich food sources over the entire search space. It should be noted that the number of food sources in the search region equals the total number of bees employed. The employed bees will share nectar information with the onlooker bees at the hive when they have finished their search in order to decrease the time spent in searching for new food sources in the following generation. The shared information comprises the richness of the nectar and the distance traversed to reach the food source. Each position of the food source represents a potential solution to the optimization problem, with the nectar's richness indicating its fitness value. Leveraging the acquired knowledge, the onlooker bees aim to identify new, high-quality food sources in the vicinity of selected sources. If a food supply has a high fitness value, it is more likely to be chosen. The onlookers play the exploitation rule, and there are exactly as many onlookers as there are employed bees, to guarantee the diversity of the solution space and prevent the algorithm from becoming stuck at a local optimum, scouter bees are added to each generation. If a food source cannot be improved after a certain number of trials, it is abandoned. The employed bee associated with the abandoned solution will then transform into a scouter bee and start scanning the entire search region for a new food source[21,23,26–28].

The following subsections describe the five techniques that make up the detailed procedures of the enhanced artificial bee colony with multi-elite guidance (MGABC)[28]. The pseudocode of the MGABC optimization algorithm is described in Algorithm (1).

Population Initialization.

The algorithm initiates by stochastically generating food source sites (SN), which symbolize potential solutions in the search space. These sites can be perceived as the environment of the hive, where the search for solutions takes place. Within the range of the lower and upper boundaries of the parameters, the initial food sources $X_i = \left(x_{i,1}, x_{i,2}, \ldots, x_{i,k}\right)$ are produced randomly using Eq. (25)[20,21], where $i \in (1, 2, \ldots, SN)$, $k \in (1, 2, \ldots, S)$ and S denotes the dimension size of the parameters to be optimized.

$$x_{i,k} = x_k^{min} + rand(0,1) \times \left(x_k^{max} - x_k^{min}\right) \tag{25}$$

where x_k^{min} and x_k^{max} are the lower and upper bounds of the Sth dimensions, respectively.

Exploration technique.

The solution search equation for employed bees should prioritize robust exploration, as their role is to discover new solutions throughout the entire search space. The enhanced solution exploratory equation described in Eq. (26) is therefore utilised by the exploratory bees, which comes from the CABC method proposed by Gao[37].

$$v_{i,k} = x_{r_1,k} + \phi_{i,k} \times \left(x_{r_1,k} - x_{r_2,k} \right) \tag{26}$$

where X_{r1} and X_{r2} are two distinct food sources that were randomly chosen from the initial population and are both different from X_k, in the range of $[-1, 1]$, $\phi_{i,k}$ is a uniformly distributed random number[21,23,26].

Exploitation technique. The onlooker bees should focus on exploitation rather than exploration because it is their duty to undergo comprehensive searches for possible promising food sources in the area in order to generate new offspring. As a result, Eq. (27)[28] lists the new solution search method based on multiple elite solutions for the onlooker bees.

$$v_{i,k} = \left\{ \begin{array}{cc} x_{e,k} + \phi_{i,k} \times \left(x_{e,k} - x_{i,k} \right) & if \ rand(0,1) \leq MR \\ x_{i,k} & otherwise \end{array} \right\} \tag{27}$$

where $X_e = \left(x_{1,k}, x_{2,k}, \ldots, x_{G_e,k} \right)$ is one of the most promising elite solutions from the current population that was selected at random from the elite group. $Ge = q.SN$ denotes the size of the elite group. The control parameter MR, as employed by Akay and Karaboga[38], serves the purpose of regulating the number of dimensions that can be transferred from the superior solution X_e to the new solution V_i.

Scouting technique. In each cycle, the algorithm checks to see whether there is any exhausted source that may be abandoned once all employed bees and onlooker bees have finished their searches. The counters that were updated throughout search are utilised to determine whether a source should be abandoned[23,37]. If the value of the counter surpasses the predefined "limit" the source associated with this counter is considered depleted and abandoned within the context of the ABC algorithm[21,23,36]. If X_i represents the abandoned food source, then the scout bee generates a new random food source according to Eq. (25).

Enhanced neighborhood search operator technique. The modified operator that performs additional exploration and aids in locating better alternatives or even the best solution, if one solution is unfortunate became trapped by one of the local optima is listed in Eq. (28)[22,23,27,28,38]

$$TX_i = r_1 \cdot X_i + r_2 \cdot X_{e1} + r_3 \cdot (X_{e2} - X_{e3}) \tag{28}$$

where X_{e1}, X_{e2}, and X_{e3} are three food sources chosen at random from the elite group, and they must be distinct from X_i. Table 2 lists the various MGABC design parameters design parameters that are utilised to maximize controller gains.

Controller optimization for trajectory tracking

All of the simulations provided in this study were run in MATLAB/SIMULINK with the ODE45 solver using 0.001 s sampling time and the simulation time is taken to be 5 s on a personal computer with an Intel® Core™ i7-10750H CPU running at 2.60 GHz, 16 GB of RAM, and a 64-bit Win 11 operating system. Equations (29)–(31) list the desired trajectories. Figure 5 illustrates a schematic diagram depicting the fundamental tuning scheme of the SMC-PID control method.

Convergence of results is a key aspect in any optimization process. While there is no fixed iteration count to guarantee an optimal solution, we monitored the convergence behavior of the objective function values. Once we observed negligible deviations and attained a near-optimal solution, we decided to terminate the iterations. Furthermore, it is worth mentioning that each run of the optimization process yields global optimal solutions, although with varying iterations number. We aimed to assess the performance of various optimization algorithms for our particular problem, considering that the most efficient algorithm reported in previous publications may not necessarily be the most suitable for our specific problem domain. To ensure a comprehensive evaluation, we explored multiple optimization algorithms with different parameter settings, including the number of iterations.

In our work, we conducted experiments with different iteration values, specifically starting with 30, 40, 50 and 60 iterations. Through careful analysis, we observed that after 50 iterations, the differences in the objective function values became negligible, indicating that the optimal value was reached.

$$\theta_{d1} = \sin(2t) - 0.1 \tag{29}$$

$$\theta_{d2} = \cos(2t) - 0.1 \tag{30}$$

DP. No	The MGABC	
	Algorithm parameter	Value
1	Colony size, *SN*	20
2	Iterations, *T*	50
3	Modification rate, *MR*	0.5
4	Size of elite group,*Ge*	3

Table 2. The internal design parameters of MGABC optimization algorithm.

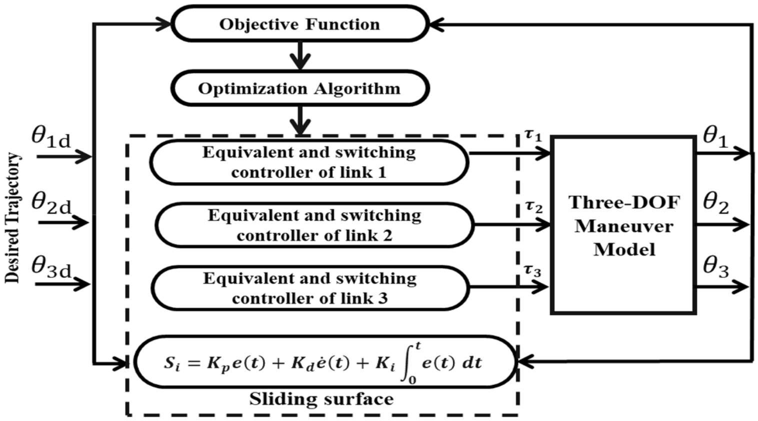

Objective Function
Optimization Algorithm
Desired Trajectory
θ_{1d}
θ_{2d}
θ_{3d}
Equivalent and switching controller of link 1
Equivalent and switching controller of link 2
Equivalent and switching controller of link 3
τ_1
τ_2
τ_3
Three-DOF Maneuver Model
θ_1
θ_2
θ_3
$S_i = K_p e(t) + K_d \dot{e}(t) + K_i \int_0^t e(t)\,dt$
Sliding surface

Figure 5. Schematic diagram of the tuning process.

$$\theta_{d3} = \cos(2t) - 0.1 \tag{31}$$

The integral time absolute error (ITAE) listed in Eq. (32) served as the objective function (OBJF) of the tuning process. The curves of OBJF versus iteration for MGABC and other optimization techniques from literature study such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Artificial Bee Colony (ABC), Ant Lion Optimizer (ALO) and Grey Wolf Optimizer (GWO) are depicted in Fig. 6. Additionally, it should be highlighted that optimal controlling parameters remained unaltered throughout the studies.

$$ITAE = \int_0^t t|\boldsymbol{e}(\boldsymbol{t})|dt \tag{32}$$

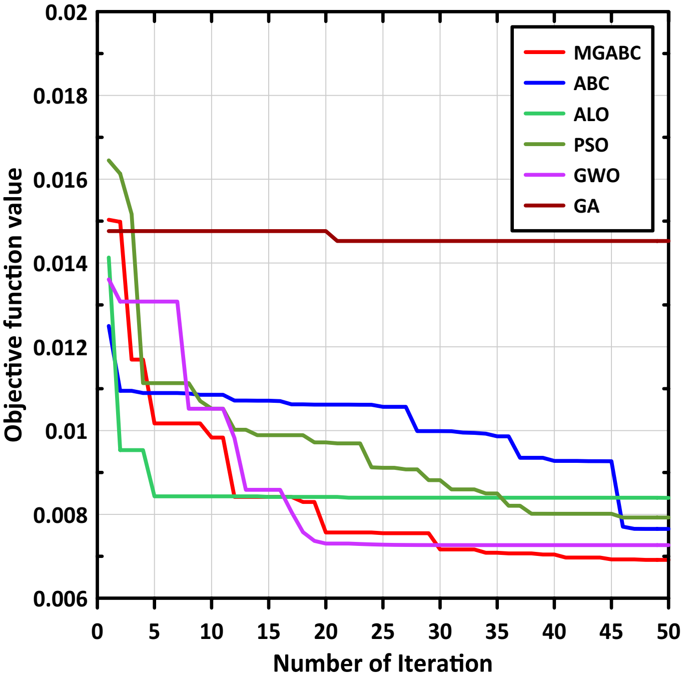


Figure 6. Objective function versus iteration curves.

Algorithm (1): MGABC Pseudo-code

```
Input: Food sources ($\boldsymbol{SN}$), Lower bound ($\mathbf{LB}$), Upper bound ($\mathbf{UB}$), Modification rate ($\mathbf{MR}$), Elite group size ($\mathbf{G_e}$), Max. no. of iterations ($\mathbf{It_{max}}$) and Counter limit.
Initialize population then select the elite solutions and store the group and fitness values.
While ($\mathrm{It} < \mathrm{It_{max}}$)
    /* Exploration Stage*/
    for $i = 1{:}SN$
        Generate new solution according to Eq. ($\mathbf{26}$) and evaluate their fitness values.
        if $f(V_i) \leq f(X_i)$
            $X_i = V_i\,, Counter\ (i) = 0$
            else
            $Counter(i) = Counter\ (i) + 1$
        end if
        Update the elite group
    end for
    /* Exploitation Stage*/
    for $i = 1{:}SN$
        Find new offspring according to Eq. ($\mathbf{27}$), check for bounds and evaluate fitness values,
        if $f(V_i) \leq f(X_i)$
            $X_i = V_i\,, Counter\ (i) = 0$
            else
            $Counter(i) = Counter\ (i) + 1$
        end if
        Update the elite group
    end for
    /* Scouter Bee Stage*/
    if $Counter(i) \geq limit$
        Replace exhausted solution with new solution using Eq. ($\mathbf{25}$) and update the population.
    end if
    /* Enhanced neighborhood search Stage*/
    for $i = 1{:}SN$
        Find a trial solution using Eq. ($\mathbf{28}$) and evaluate their fitness values.
        if $f(TX_i) \leq f(X_i)$
            $X_i = TX_i$
        end if
        Update the elite group
    end for
end while
Output: Print position and fitness value of the best solution
Stop
```

Table 3 displays the parameters of the robotic system model's upper and lower search spaces, together with the MGABC optimizer's tuned values. The lower bound values for the sliding surface and switching mode gain in the controller were specifically chosen to ensure the stability of Eq. (18). The process of selecting these values incorporated several factors, including the physical properties of the system such as inertia and weight of each link, as well as the system dynamics, desired performance, and stability criteria. These considerations were crucial in determining the lower bounds that would guarantee the stability of the system and ensure its proper functioning.

	Lower bound			Upper bound			Optimum		
Parameter	Link 1	Link 2	Link 3	Link 1	Link 2	Link 3	Link 1	Link 2	Link 3
k_p	5	5	5	50	50	50	50	50	50
k_I	0.1	0.1	0.1	10	10	10	9.973	10	9.330
k_d	0.1	0.1	0.1	1	1	1	0.206	0.100	0.109
λ	50	50	50	300	300	300	300	295.986	299.837

Table 3. The optimum parameter values of 3-DOF system model solved based on MGABC optimizer.

While the upper bound values were selected in a more approximate manner. While these values were not derived through a rigorous optimization process, they were carefully chosen to establish a reasonable range for the parameters without violating any operational constraints. The intent behind this selection was to encompass a broad range of feasible values that could be explored for experimentation and analysis purposes.

Table 4 presents the minimum, average, and maximum OBJFs obtained by the MGABC optimizer in comparison to other meta-heuristic algorithms in the literature where 10 separate runs of each algorithm were performed. It is notable that the MGABC algorithm produced the minimum results overall runs with the lowest standard deviation with the lowest OBJF values. Figure 7a shows the $X - Y$ plot of end-effector and Fig. 7b–d demonstrate the trajectory tracking of each link. As we can see the desired and actual trajectories are nearly identical.

Figure 8a–c shows the error in trajectories of links 1, 2 and 3, respectively. The controller exhibits superior tracking performance, as evidenced by the objective function (OBJF) values of links 1, 2, and 3. Which are 0.00215, 0.00126 and 0.00350 respectively. That illustrates how the MGABC method outperforms other algorithms in the literature in terms of result superiority and robustness.

Detailed simulated performance evaluation

In order to ensure robustness, a controller must effectively mitigate the impact of both measured and unmeasured noise and disturbance, as well as exhibit adaptability to uncertainties present in the system[3]. In this section, we extensively investigate the controller's ability to reject disturbance and noise, while also evaluating its robustness against uncertainties in the payload's mass. Furthermore, we examine the controller's adaptability to the flexibility exhibited by the joints in the system.

Robustness vs disturbance and noise.

The dynamic disturbance signal and noise, listed in Eqs. (33) and (34), are injected at the output of each controller and sensor of joint at the same time as shown in Fig. 9.

$$\tau_i = per \times rand \times \tau_i \tag{33}$$

$$N_i = per \times rand \times \left(\theta_{d_i} + \theta_i\right) \tag{34}$$

In the given context, the variable "*rand*" represents a randomly generated number within the inclusive range of $[0, 1]$. The symbols τ_i, θ_{d_i} and θ_i, where i takes values of 1, 2, and 3, correspond to the applied torques, desired angles, and measured angles, respectively. The systems performance is checked for varying percentage $\left(per\right)$ values from 1% to severe conditions of 5% percentage of disturbance and noise. The corresponding OBJFs values of the optimized controller with MGABC to the variation in percentage of disturbance and noise are listed in Table 5.

The controller is designed to minimize the objective function (OBJF), and its effectiveness is evident from the observed percentage increase in OBJF values. This increase ranges from 0.954% under conditions of low disturbance and noise to 14.55% under conditions of severe disturbance and noise. The controller's ability to mitigate the impact of disturbance and noise is demonstrated by the relatively small increase in OBJF values, indicating its robustness and effectiveness in maintaining desired performance even in challenging situations with significant disturbances and noise.

Robustness vs the variation in mass of payload.

The primary objective of a maneuvering system is to perform the task of gripping and manipulating objects with diverse masses using its end-effector. When there is a change in the mass of the end-effector, it introduces a new configuration to the system, necessitating the need for a robust controller. The robust controller is essential to mitigate the impact of variations in the end-effector mass and ensure stable and reliable performance[3].

In this study we utilized the great facilities of the Simscape multibody toolbox and easily added a new component which describes the end-effector tool as shown in Fig. 10. The variation in the mass of end-effector follows Eq. (35), and the corresponding OBJFs values of the optimized controller with MGABC to the variation in percentage of mass are listed in Table 6.

$$Uncertainty = per \times m \tag{35}$$

where m is the mass of the link i and in our study $m = 1$ Kg and the percentage (per) of uncertainty ranges from 10% to severe uncertainty conditions of 50% (i.e., from 0.1 to 0.5 Kg).

Algorithm	Minimum	Maximum	Mean	Standard deviation
ABC	0.007653312	0.00831256	0.008040319	0.000270904
PSO	0.007925914	0.008892364	0.008292557	0.000448724
GWO	0.007266579	0.010600297	0.008693191	0.00140179
ALO	0.008393979	0.00889129	0.008791828	0.000222404
GA	0.014525562	0.01920016	0.016967015	0.001764241
MGABC	0.006914077	0.007176508	0.007025913	0.00010948

Table 4. The optimum OBJF of the MGABC algorithm in comparison with other algorithms.

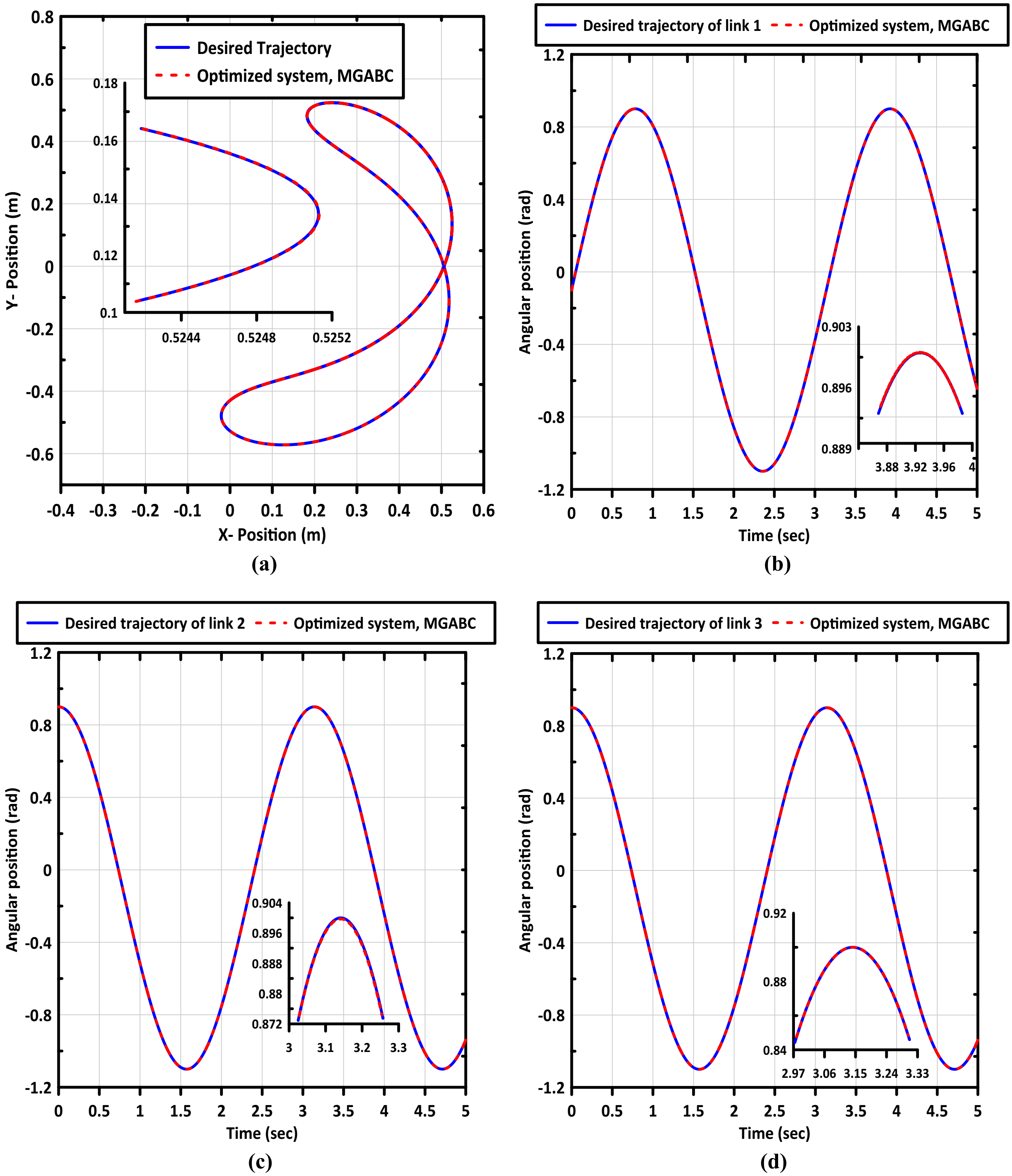


Figure 7. Shows (**a**) X − Y plot of end-effector and trajectory tracking curves of (**b**) link (1), (**c**) link (2), and (**d**) link (3).

The optimized controller utilizing the MGABC algorithm demonstrates resilience in the face of payload mass variations. Through analysis, it is observed that the OBJF values exhibit an inflation ranging from 5.726% under low uncertainty conditions to 18.887% under severe uncertainty conditions. This indicates that the controller maintains its performance and robustness even in challenging situations with significant uncertainties.

Adaptability to joint flexibility. Flexible link manipulators (FLMs) offer numerous advantages such as reduced material consumption, lower power requirements, lighter weight, fewer actuators, and improved

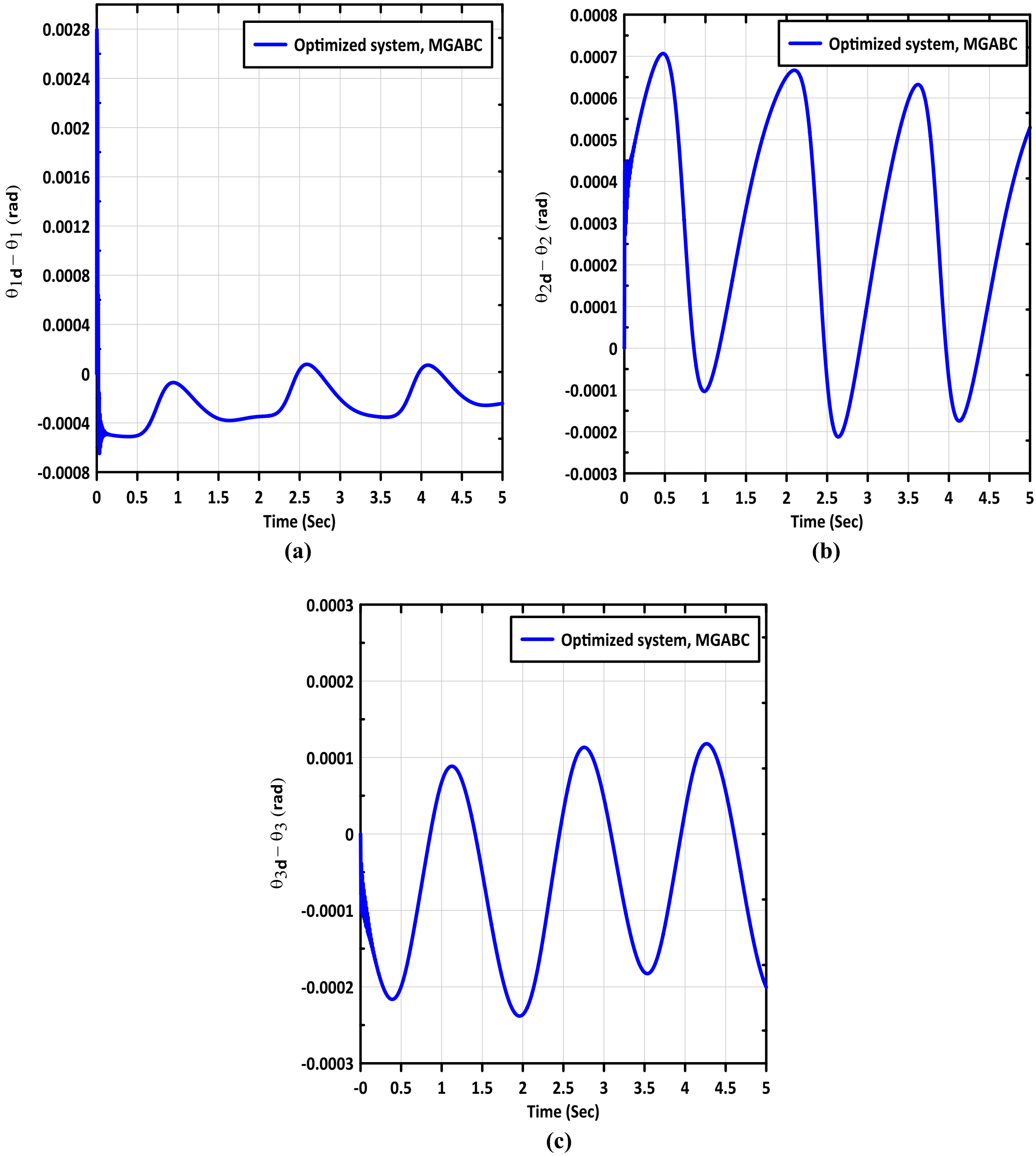


Figure 8. The error in signals of (**a**) link (1), (**b**) link (2), and (**c**) link (3).

maneuverability. Despite these benefits, the control of FLM systems presents several unresolved challenges that must be addressed before their widespread adoption in practical real-world applications[1,39]. In this section, we will investigate the capability of the optimized controller approach to handle the influence of joint flexibility. The Simscape multibody toolbox is employed to simulate the system, allowing us to model the three joints as flexible. The internal mechanics of these joints are characterized by a spring stiffness of 1 and an equivalent viscous damping of 0.001. This setup enables us to analyze the response of the controller in the presence of joint flexibility and evaluate its adaptability to such conditions. $X - Y$ plot of end-effector and trajectory tracking curves are shown in Fig. 11. The proposed controller shows superior tracking performance with little vibration in the movement of the end effector.

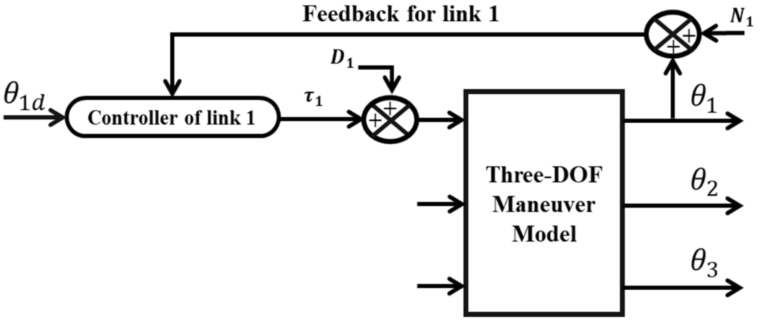


Figure 9. The robotic system in the presence of disturbance and noise.

	OBJFs values
Percentage	MGABC
1%	0.00698
2%	0.00715
3%	0.00733
4%	0.00761
5%	0.00792

Table 5. OBJFs values for variation in percentage of noise and disturbance.

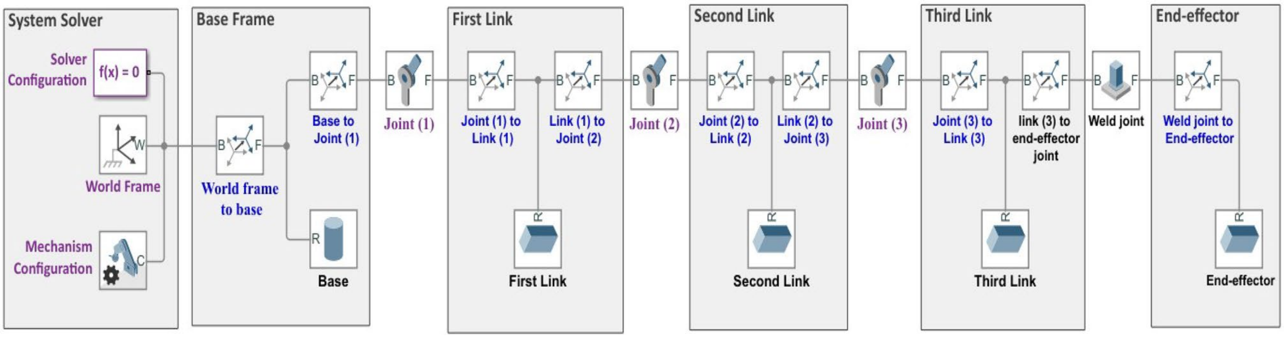


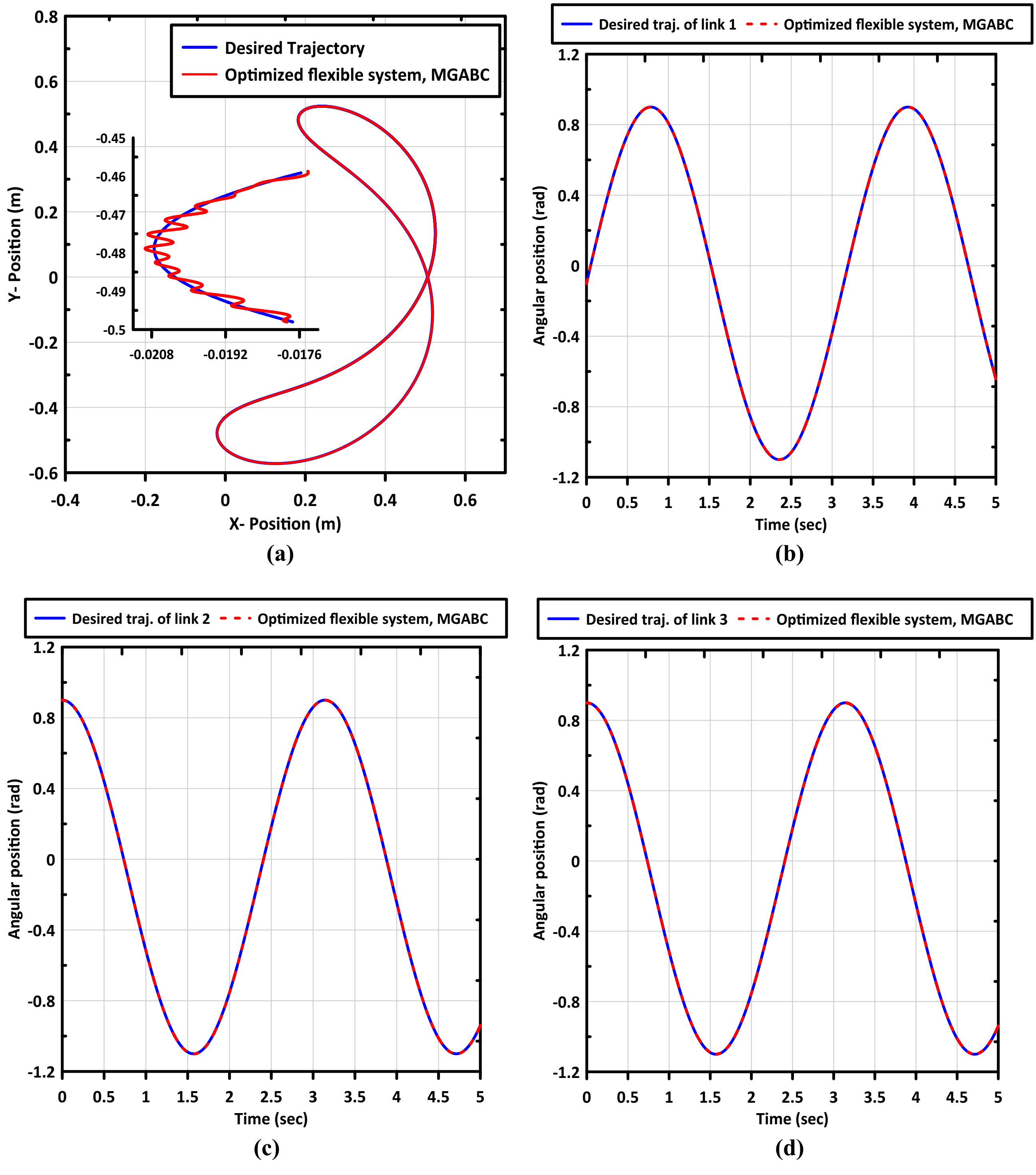
Figure 10. Simscape multibody model of the 3-DOF maneuver with end-effector.

	OBJFs values
Percentage	MGABC
10%	0.00731
20%	0.00749
30%	0.00789
40%	0.00802
50%	0.00822

Table 6. OBJFs values for variation in percentage of mass.

Conclusion

This paper highlights the significance of utilizing the MATLAB Simscape Multibody Toolbox as a powerful tool for modeling and controlling both rigid and flexible maneuvers. The optimization of a sliding mode controller with a proportional integral derivative (PID) surface is achieved by employing an enhanced artificial bee colony algorithm with multi-elite guidance (MGABC). The selection of the optimal sliding surface poses a challenge due to the vast search space, leading many existing optimization algorithms to become trapped in local optima and fail to identify the optimal parameters for the sliding surface and switching mode controller gain. The



(a)

(b)

(c)

(d)

Figure 11. Shows (**a**) X − Y plot of end-effector of the FLM and trajectory tracking curves of (**b**) link (1), (**c**) link (2), and (**d**) link (3).

MGABC algorithm is chosen for its exceptional exploratory and exploitative techniques, enabling it to find the best solution. The convergence analysis conducted in this study provides evidence of the MGABC algorithm's exceptional capability to overcome local minima through its proficient exploration and exploitation techniques. This algorithm demonstrates superiority over other algorithms mentioned in the existing literature in terms of its ability to navigate and escape from local minima solutions. The investigation focuses on disturbance and noise rejection, as well as robustness against uncertainty in payload mass. The proposed controller demonstrates the ability to mitigate the effects of disturbances and noises, maintaining the objective function (OBJF) at a minimum even in highly noisy and disturbed systems. The controller effectively minimizes the OBJF, with the percentage increase in OBJF values ranging from 0.954% under low disturbance and noise conditions to 14.55%

under severe disturbance and noise conditions. Additionally, the optimized controller exhibits resilience to variations in payload mass analysis, with the percentage increase in OBJF values ranging from 5.726% under low uncertainty conditions to 18.887% under severe uncertainty conditions. Furthermore, the adaptability of the controller to joint flexibility is evaluated, demonstrating superior tracking performance with minimal vibration in the movement of the end effector.

Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Received: 13 March 2023; Accepted: 16 July 2023
Published online: 02 August 2023

References

1. Alandoli, E. A., Lee, T. S., Lin, Y. J. & Vijayakumar, V. Dynamic model and intelligent optimal controller of flexible link manipulator system with payload uncertainty. *Arab. J. Sci. Eng.* **46**, 7423–7433 (2021).
2. Lee, T. S. & Alandoli, E. A. A critical review of modelling methods for flexible and rigid link manipulators. *J. Braz. Soc. Mech. Sci. Eng.* **42**, 10 (2020).
3. Kumar, J., Kumar, V. & Rana, K. P. S. Fractional-order self-tuned fuzzy PID controller for three-link robotic manipulator system. *Neural Comput. Appl.* **32**, 7235–7257 (2020).
4. Ayten, K. K. & Dumlu, A. Real-time implementation of sliding mode control technique for two-DOF industrial robotic arm. *J. Inst. Sci. Technol.* **8**, 77–85 (2018).
5. Schlotter, M. Multibody system simulation with SimMechanics. *Analysis* **2003**, 1–23 (2003).
6. Manjaree, S. & Thomas, M. Modeling of multi-DOF robotic manipulators using sim-mechanics software. *Indian J. Sci. Technol.* **9**, 48 (2017).
7. Sadegh-Lafmejani, H. & Zarabadipour, H. Modeling, simulation and position control of 3DOF articulated manipulator. *Indones. J. Electr. Eng. Inf.* **2**, 3 (2014).
8. Lee, T. S., Alandoli, E. A. & Vijayakumar, V. 2-DOF robot modelling by SimMechanics and PD-FL integrated controller for position control and trajectory tracking. *F1000Research* **10**, 1045 (2021).
9. Sarkhel, P., Banerjee, N. & Hui, N. B. Fuzzy logic-based tuning of PID controller to control flexible manipulators. *SN Appl. Sci.* **2**, 6 (2020).
10. West, C., Montazeri, A., Monk, S. D. & Taylor, C. J. A genetic algorithm approach for parameter optimization of a 7DOF robotic manipulator. *IFAC-PapersOnLine* **49**, 1261–1266 (2016).
11. Alandoli, E. A. & Lee, T. S. A critical review of control techniques for flexible and rigid link manipulators. *Robotica* **38**, 2239–2265. https://doi.org/10.1017/S0263574720000223 (2020).
12. Renfrew, A. Book review: Introduction to robotics: Mechanics and control. *Int. J. Electr. Eng. Educ.* **41**, 388–388 (2004).
13. Bendimrad, A., El Amrani, A. & El Amrani, B. Optimization of the performances of a two-joint robotic arm using sliding mode control. *Int. J. Automot. Mech. Eng.* **19**, 9634–9646 (2022).
14. Loucif, F. & Kechida, S. Optimization of sliding mode control with PID surface for robot manipulator by evolutionary algorithms. *Open Comput. Sci.* **10**, 369–407 (2020).
15. Zhang, J. *et al.* High-order sliding mode control for three-joint rigid manipulators based on an improved particle swarm optimization neural network. *Mathematics* **10**, 3418 (2022).
16. Pathak, R. R., Mohanty, S. & Sengupta, A. An optimization-based adaptive sliding mode control for an inverted pendulum. *IETE J. Res.* https://doi.org/10.1080/03772063.2022.2127941 (2022).
17. Chu, Y., Fei, J. & Hou, S. Dynamic global proportional integral derivative sliding mode control using radial basis function neural compensator for three-phase active power filter. *Trans. Inst. Meas. Control* **40**, 3549–3559 (2018).
18. Elawady, W. M., Lebda, S. M. & Sarhan, A. M. An optimized fuzzy continuous sliding mode controller combined with an adaptive proportional-integral-derivative control for uncertain systems. *Optim. Control Appl. Methods* **41**, 980–1000 (2020).
19. Goel, A. & Mobayen, S. Adaptive nonsingular proportional–integral–derivative-type terminal sliding mode tracker based on rapid reaching law for nonlinear systems. *JVC/J. Vib. Control* **27**, 2669–2685 (2021).
20. Karaboga, D. & Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **39**, 459–471 (2007).
21. Karaboga, D. & Basturk, B. On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput. J.* **8**, 687–697 (2008).
22. Cui, Y., Hu, W. & Rahmani, A. A reinforcement learning based artificial bee colony algorithm with application in robot path planning. *Expert Syst. Appl.* **203**, 1173 (2022).
23. Aslan, S., Badem, H. & Karaboga, D. Improved quick artificial bee colony (iqABC) algorithm for global optimization. *Soft Comput.* **23**, 13161–13182 (2019).
24. Pham, D. T. *et al.* The bees algorithm—a novel tool for complex optimisation problems. In *Intelligent Production Machines and Systems—2nd I*PROMS Virtual International Conference 3–14 July 2006* 454–459 (Elsevier Ltd, 2006). https://doi.org/10.1016/B978-008045157-2/50081-X.
25. Wang, H., Du, H., Cui, Q. & Song, H. Artificial bee colony algorithm based PID controller for steel stripe deviation control system. *Sci. Prog.* **105**, 1 (2022).
26. Celik, Y. An enhanced artificial bee colony algorithm based on fitness weighted search strategy. *Automatika* **62**, 300–310 (2021).
27. Zhou, X., Wang, H., Wang, M. & Wan, J. Enhancing the modified artificial bee colony algorithm with neighborhood search. *Soft Comput.* **21**, 2733–2743 (2017).
28. Zhou, X., Lu, J., Huang, J., Zhong, M. & Wang, M. Enhancing artificial bee colony algorithm with multi-elite guidance. *Inf. Sci. (N. Y.)* **543**, 242–258 (2021).
29. Gul, E. & Toprak, A. N. Contourlet and discrete cosine transform based quality guaranteed robust image watermarking method using artificial bee colony algorithm. *Expert Syst. Appl.* **212**, 118730 (2023).
30. Sahu, P., Singh, B. K. & Nirala, N. An improved feature selection approach using global best guided Gaussian artificial bee colony for EMG classification. *Biomed. Signal Process. Control* **80**, 104399 (2023).
31. Zhou, J. *et al.* An individual dependent multi-colony artificial bee colony algorithm. *Inf. Sci. (N. Y.)* **485**, 114–140 (2019).
32. Xue, Y., Jiang, J., Zhao, B. & Ma, T. A self-adaptive artificial bee colony algorithm based on global best for global optimization. *Soft Comput.* **22**, 2935–2952 (2018).
33. Cui, L. *et al.* A novel artificial bee colony algorithm with an adaptive population size for numerical function optimization. *Inf. Sci. (N. Y.)* **414**, 53–67 (2017).
34. Dudek, G. & Jenkin, M. *Inertial Sensing, GPS and Odometry* (Springer Handbooks, 2016). https://doi.org/10.1007/978-3-319-32552-1_29.

35. Slotine, J.-J. & Li, W. *Applied Nolinear Optimal Control* (Springer, 1991).
36. Mahmoodabadi, M. J. & Shahangian, M. M. A new multi-objective artificial bee colony algorithm for optimal adaptive robust controller design. *IETE J. Res.* **68**, 1251–1264 (2022).
37. Gao, W. F., Liu, S. Y. & Huang, L. L. A novel artificial bee colony algorithm based on modified search equation and orthogonal learning. *IEEE Trans. Cybern.* **43**, 1011–1024 (2013).
38. Akay, B. & Karaboga, D. A modified Artificial Bee Colony algorithm for real-parameter optimization. *Inf. Sci. (N. Y.)* **192**, 120–142 (2012).
39. Ahmad, M. A., Raja Ismail, R. M. T., Ramli, M. S. & Zawawi, M. A. Elastic joint control using non-collocated fuzzy and filtering scheme: A comparative assessment. In *AMS2010 Asia Model. Symp. 2010—4th Int. Conf. Math. Model. Comput. Simul.* 366–371 (2010). https://doi.org/10.1109/AMS.2010.80.

Author contributions

M.I.A.: SimMechanics modelling, writing-conceptualization and methodology, original draft preparation, and optimization developing.A.M.M.A.: writing-conceptualization and methodology, writing-original draft preparation, preparation of figures, writing—reviewing and editing and supervision.S.E.: Writing-conceptualization and methodology, preparation of figures and supervision, writing—reviewing and editing and supervision.K.A.F.moustafa: writing-conceptualization and methodology, original draft preparation, investigation, writing—reviewing and editing.K.R.A.: idea, mathematical modelling, preparation of figures, writing-conceptualization and methodology, writing—reviewing and editing and supervision.

Funding

Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB). Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.I.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.