Received: 4 June 2026 | Revised: 26 July 2026 | Accepted: 27 October 2026 | Published online: 20 November 2026

RESEARCH ARTICLE

Journal of Climbing and Walking Robots
2026, Vol. 00(00) 1–14
DOI: 10.47852/bonviewJCWR620210678

BON VIEW PUBLISHING

Energy Minimization of Spatial 3 DOF Manipulator for Smooth Optimal Trajectory Planning Using Meta-Heuristics

Ahmed H. Shahein[1], Mohammed Ghazy[1] and Atef A. Ata[1,2,*]

[1] *Department of Engineering Mathematics and Physics, Alexandria University, Egypt*

[2] *Mechatronics Engineering Program, Zewail City of Science, Technology and Innovation, Egypt*

Abstract: In recent decades, meta-heuristic and evolutionary algorithms have been widely applied for optimal trajectory planning through imitating natural intelligence to solve constrained optimization problems. Among these approaches, adaptive particle swarm optimization (APSO) and genetic algorithm (GA) are employed in this study. In this research, we optimize the smooth trajectory of fourth-degree polynomial for minimum energy consumption for spatial manipulator using two meta-heuristics techniques, APSO and GA, and examine the different configurations of the manipulator. Results show that the elbow-up configuration is significantly more energy-efficient than the elbow-down configuration due to smoother motion and lower torque requirements. In addition to the total mechanical energy required along the trajectory, the energy loss per cycle is presented as another metric criterion for comparison between both configurations. Both APSO and GA achieve nearly identical optimal energy values. However, the APSO algorithm overcomes the GA algorithm regarding the convergence speed and computational time for the same level of precision. The smooth trajectory reduces jerk and abrupt vibrations and ensures larger actuator life.

Keywords: energy minimization, Adaptive Particle Swarm Optimization (APSO), meta-heuristics, robot dynamics, genetic algorithm

1. Introduction

Robot manipulators are characterized with high multi-input and multi-output nonlinear systems and coupled with high time varying dynamic features. Due to these reasons, robotic research is one of the most interesting and challenging areas. Designing the optimal trajectory relies on many criteria such as the shortest distance between start and target points or minimum energy consumption or minimum time to reach the goal point which satisfies the robot's dynamics and kinematics constraints. Optimal trajectory planning based on minimum energy consumption is a vital issue due to increasing energy demand, limited energy resources, and rising political crises. Therefore, many researchers have devoted their efforts in order to fulfill this objective.

2. Literature Review

Field and Stepanenko [1] introduced iterative dynamic programming to minimize energy while handling actuator and time constraints and reducing computational complexity. Xian, et al. [2] employed spline functions for robot manipulator trajectory planning in order to provide smooth and continuous paths with lower energy consumption and actuator stress. However, their approach requires careful parameter selection and additional computational cost. Ata [3] applied generalized pattern search to optimize polynomial trajectories for redundant manipulators, demonstrating effective energy reduction. Ata, et al. [4] proposed an optimal trajectory through GA algorithm for a 3 DOF planar manipulator that minimizes energy consumption. The approach ensures smooth joint motion, improving efficiency, and satisfying kinematic constraints with acceptable accuracy. Gregory, et al. [5] formulated the problem as an optimal control task with holonomic constraints, ensuring dynamically feasible solutions via variational methods. In order to reduce energy consumption through optimal control, a Hamiltonian formulation to construct energy optimal trajectories for a servomotor system was proposed by Wang, et al. [6]. However, this approach requires precise system modeling and increases computing complexity. Pellicciari, et al. [7] proposed an optimization method for pick and place industrial robots to reduce energy consumption through trajectory modification, their approach achieved significant energy savings while maintaining task performance. Liu, et al. [8] proposed a multi-objective particle swarm optimization (MOPSO) algorithm focused on optimizing energy consumption,

***Corresponding author:** Atef A. Ata, Department of Engineering Mathematics and Physics, Alexandria University, Egypt and Mechatronics Engineering Program, Zewail City of Science, Technology and Innovation, Egypt. Email: atef.ata@zewailcity.edu.eg

© The Author(s) 2026. Published by BON VIEW PUBLISHING. This is an open access article under the CC BY License (https://creativecommons.org/licenses/by/4.0/).

joint velocity, joint jerk, and obstacle avoidance for redundant robotic manipulators. However, a trade-off exists between energy efficiency and motion smoothness. Ye, et al. [9] developed an enhanced non-dominated sorting Dung Beetle Optimization algorithm for robotic manipulator trajectory planning, considering energy consumption, execution time, and mean jerk. Quintic *B*-spline interpolation and a Fréchet distance-based selection strategy were used to maintain trajectory smoothness. Nevertheless, additional decision-making strategy is required to select the most suitable solution from the Pareto optimal set. Lorenz, et al. [10] provided an energy-efficient trajectory planning framework using optimization techniques to reduce actuator energy consumption for robot manipulators. Sehgal, et al. [11] proposed a GA framework for parameter optimization for robotic manipulation tasks. The method improved system performance and indirectly reduced energy consumption through optimized motion control. Nonoyama, et al. [12] presented an energy-efficient robot configuration and motion planning framework utilizing genetic algorithm (GA) and particle swarm optimization (PSO). They optimize PID-based trajectories for dual arm manipulators to reduce energy and execution time under kinematic limits. Massaro, et al. [13] formulated minimum time trajectory planning as an optimum control problem that generates dynamically feasible trajectories. Their approach satisfies actuator constraints although the complexity rises with nonlinear dynamics for multi degree of freedom robotic manipulator. Soori, et al. [14] explored optimization techniques for reducing energy consumption in industrial robots. They compared trajectory planning, control algorithms, and hardware approaches by highlighting energy saving while considering trade-off between precision and complexity. Akbari et al. [15] presented an evolutionary optimization approach for robotic arm motion focusing on minimizing energy and execution time by tuning velocity and acceleration profiles. They used PSO, achieving significant efficiency improvement in industrial tasks. Oosterwyck, et al. [16] optimized the point-to-point motion profiles to minimize energy consumption, resulting in optimal trajectories that reduce actuator effort while considering motion time, smoothness, and computing complexity in robotic systems. Zhou et al. [17] proposed an improved Sparrow Search Algorithm (ISSA) to minimize robot energy consumption using joint torque and angular velocity, with seventh-degree *B*-spline trajectories. ISSA overcomes the standard SSA and PSO in energy consumption and computational time. However, generalization on different manipulator configurations and more complex tasks is limited. Shrivastava [18] presented a comparative analysis of trajectory planning schemes for robotic manipulators, including spline-based, optimization, and heuristic methods. The study discusses the trade-offs between energy efficiency, smoothness, time optimality, and computing complexity in motion selection algorithms. Elgohr, et al. [19] introduced a trajectory optimization method for a 6-DOF robotic arm using Whale GA. The method focused on minimizing energy consumption while improving trajectory smoothness and quality. Yang, et al. [20] demonstrated the energy consumption mapping and the NSGA-II algorithm to optimize robot joint energy. This approach applied multi-objective that optimize the trade-off of increasing computational cost and the energy consumption in practical systems with high complexity. Xiao, et al. [21] proposed a heuristic multi-objective particle swarm optimization (HMOPSO) algorithm for energy-efficient trajectory optimization of planetary manipulators under kinematic and dynamic constraints. The method improved convergence and Pareto solution diversity. However, the study is mainly focused on planetary manipulators, which limits its direct applicability to conventional industrial manipulators. Fan et al. [22] proposed an enhanced NSGA-II algorithm for multi-objective robotic arm trajectory optimization considering efficiency and energy consumption. The method improved convergence and Pareto solutions. However, its computational efficiency remains insufficient for real-time motion planning, limiting practical industrial applications. Ueno and Tachiya [23] proposed a heuristic-learning approach for optimizing industrial robot trajectories to reduce motor energy consumption and peak current. A neural network learned trajectories generated by the heuristic algorithm, enabling faster optimization for varying paths. However, the approach depends on previously generated training trajectories, potentially limiting generalization to significantly different robot paths. Zhang et al. [24] proposed a multi-objective Parrot Optimization (MOPO) algorithm for robotic arm trajectory planning, optimizing time, energy, and impact using quintic interpolation. The method improved trajectory performance and reduced energy consumption. However, validation was limited to one UR5 trajectory, requiring further testing with different and complex trajectories. Wei et al. [25] demonstrated an improved multi-objective Golden Jackal Optimization algorithm to minimize robotic arm energy consumption and execution time. Physical experiments demonstrated reduced energy use and smoother trajectories, and on the other hand, the evaluation of the algorithm's robustness to variable loads and uncertain environments wasn't examined. Li and Liu [26] proposed a multi-objective optimization combining PSO, GA, and breadth-first search to minimize energy consumption and end-effector error while avoiding obstacles in six DOF robotic arms. Despite this, the study does not sufficiently investigate the algorithm's computational complexity and scalability in complex and cluttered environments.

However, few studies have offered comparative analyses among different meta-heuristics algorithms for manipulator motion planning. Also, examining the different configurations of the manipulator and its effect on energy consumption is not considered in most of the papers. Many authors deal with planner robot rather than spatial robot. Our objective is to focus on designing an efficient fourth-order polynomial trajectory planning technique for a 3 DOF spatial manipulator. Section one includes the introduction and section two presents the literature review. Section three demonstrates the problem formulation, while section four presents problem definition and the algorithm. Section five covers the numerical simulation and the discussion and section six includes conclusions and future work

3. Problem Formulation

3.1. Mathematical model

3.1.1.kinematical model

Consider a three-degree-of freedom articulated robot consisting of a twist joint followed by two revolute joints (RRR), as illustrated in Figure 1. The robot model was developed using SolidWorks, and the coordinate frames were assigned according to the Denavit–Hartenberg (DH) technique. The corresponding kinematic parameters are summarized in Table 1.

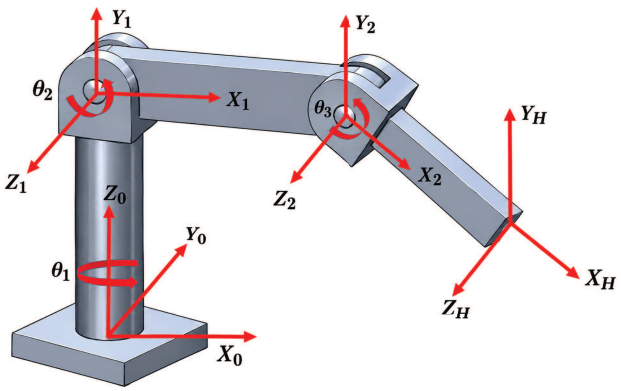


Figure 1
Solid work model and the coordinate frames assigned to the robot

The matrix of each transformation can be calculated through:

$$A_1 = \begin{bmatrix} C\theta_1 & 0 & S\theta_1 & 0 \\ S\theta_1 & 0 & -C\theta_1 & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} A_2 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & L_2C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & L_2S\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} C\theta_3 & -S\theta_3 & 0 & L_3C\theta_3 \\ S\theta_3 & C\theta_3 & 0 & L_3S\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The total transformation matrix from end effector to the robot's base is given by:

$${}^{o}TH = \begin{bmatrix} C\theta_{23}C\theta_1 & -S\theta_{23}C\theta_1 & S\theta_1 & C\theta_1[L_3C\theta_{23} + L_2C\theta_2] \\ C\theta_{23}S\theta_1 & -S\theta_{23}S\theta_1 & -C\theta_1 & S\theta_1[L_3C\theta_{23} + L_2C\theta_2] \\ S\theta_{23} & +C\theta_{23} & 0 & L_1 + L_3S\theta_{23} + L_2S\theta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Where $C\theta_1$ is cosine θ_1, $S\theta_1$ is sine θ_1, $C\theta_{23} = \cos(\theta_2 + \theta_3)$ and $S\theta_{23} = \sin(\theta_2 + \theta_3)$. The total transformation matrix (2) is useful because it describes the complete pose of the robot end effector relative to the base frame. It combines both position and orientation in a single matrix.

Inverse kinematics refers to the computational procedure used to determine the joint variables required for the end effector to attain a desired position and orientation within the robot workspace. This process involves solving a set of nonlinear and often complex equations that define the kinematic relationships of the system for more details the reader is refereed to [27]. Solutions to these equations can be obtained but their uniqueness is not always guaranteed due to system constraints or redundancy. In this work, an analytical inverse kinematics solution is derived and used in calculating the boundary conditions including initial and the final joints angle.

The inverse kinematics formulation serves as a criterion for workspace feasibility. A real solution implies that the target lies inside the reachable workspace, whereas the absence of a solution indicates that it is outside.

3.1.2. dynamic model

The total kinetic energy K and the potential energy P of the robot are given by [28]:

$$K = \sum_{i=1}^{n=3} \frac{1}{2} m_i v_i^2 + \frac{1}{2} I_i \omega_i^2, \; P = \sum_{i=1}^{n=3} m_i g h_i(q) \quad (3)$$

The equations of motion in compact form using Lagrange equation are given by:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \left(\frac{\partial L}{\partial q_i}\right) = Q_i \quad (4)$$

Where the Lagrangian is $L = K - P$, q_i represents the generalized coordinate and Q_i is the generalized force. The equation of motion that gives the torque requirements is given as [29]:

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_{fr}(\dot{q}_i) \quad (5)$$

Where $M(q)$ represents the inertia matrix, $C(q,\dot{q})$ represents the centrifugal and Coriolis terms, $G(q)$ represents the gravitational term while $\tau_{fr}(\dot{q}_i)$ represents the frictional joints torque. The dynamic model of the 3 DoF manipulator can be arranged in matrix format as following.

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} M_{11} M_{12} M_{13} \\ M_{21} M_{22} M_{23} \\ M_{31} M_{32} M_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} C_{11} C_{12} C_{13} \\ C_{21} C_{22} C_{23} \\ C_{31} C_{32} C_{33} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \\ \dot{\theta}_3^2 \end{bmatrix}$$

$$+ \begin{bmatrix} R_{11} R_{12} R_{13} \\ R_{21} R_{22} R_{23} \\ R_{31} R_{32} R_{33} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1\dot{\theta}_2 \\ \dot{\theta}_1\dot{\theta}_3 \\ \dot{\theta}_2\dot{\theta}_3 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} g + \begin{bmatrix} b_1\dot{\theta}_1 \\ b_2\dot{\theta}_2 \\ b_3\dot{\theta}_3 \end{bmatrix} \quad (6)$$

Where the elements of the matrices M, C, R, G are shown in the appendix, and we neglect the effect of joint friction, so the last term in the equations is zero.

Table 1
Robot's Parameters of DH.

Link No	Joint angle θ	Link offset d	Link length a	Twist angle α	Transformation T
1	θ_1	L_1	0	90 °	${}^{0}T_1$
2	θ_2	0	L_2	0	${}^{1}T_2$
3	θ_3	0	L_3	0	${}^{2}T_3$

3.2. Adaptive particle swarm algorithm

In order to maintain population diversity and prevent stagnation in local minima and to accelerate up convergence and create a balance between exploration and exploitation behavior, the Adaptive Particle Swarm Optimization (APSO) was suggested. The APSO is one of the biologically inspired algorithms that is believed to be simple to use for locating the best solution in the search space which follows the simple bird rules:

1) It tries not to come too close to other birds (Separation).
2) It steers toward the average direction of other birds (Alignment).
3) It tries to fit the "average position" between other birds with no wide gaps in the flock (Cohesion)

The updated velocity and position equations as described in [30] are:

$$V_{i,n}^{k+1} = \omega\, V_{i,n}^{k} + \alpha_1\, rand(0,\,1)(gbest_{i,n} - x_{i,n}) + \alpha_2\, rand(0,\,1)(pbest_{i,n} - x_{i,n}) \tag{7}$$

$$X_{i,n}^{k+1} = X_{i,n}^{k} + V_{i,n}^{k+1}\,\Delta t \tag{8}$$

Where ω is the inertia weight, the coefficient α_1 represents the global component, while α_2 corresponds to the personal component. The selection of the adaptive parameters is vital issue to compromise between exploitation and exploration.

$$\omega_i = \omega_{max} - \left(\frac{(\omega_{max} - \omega_{min})}{i_{max}}\right) i \tag{9}$$

Where $\omega_{max} = 0.8\, and\, \omega_{min} = 0.4$ and i is the iteration number, i_{max} represents the total number of iterations, i^{th} is the particle number [30].

In order to ensure that the sum of both coefficients α_1 and α_2 equals 4, the coefficient α_1 increases linearly from 0.5 to 3.5, while α_2 decreases from 3.5 to 0.5. Initially, ω is 0.8 and eventually converges to 0.4. Figure 2 shows the flowchart of the APSO.

3.3. Genetic algorithm

The GA is a population based meta-heuristic optimization technique inspired by natural selection and evolutionary principles and originally developed by John Holland [31] in 1970s based on Darwin's theory. In GA, each candidate solution is represented as a chromosome composed of genes and a population of such solutions evolves iteratively toward optimality. The algorithm operates through a sequence of main processes: initialization where a diverse set of solutions is generated then evaluation where each solution is evaluated using an objective function. Then, the selection step where fitter individuals are chosen to form mating pool followed by the recombination step where genetic operators such as crossover and mutation are applied. This step generates new offspring and finally replacement where new solutions update the population using strategies like elitism or full regeneration. These steps are repeated until a stopping criterion is met. The balance between exploration "diversity" and exploitation "selection of fit individuals" is crucial to ensure convergence toward high-quality solutions without stagnation. Figure 3 shows the flowchart of the GA.

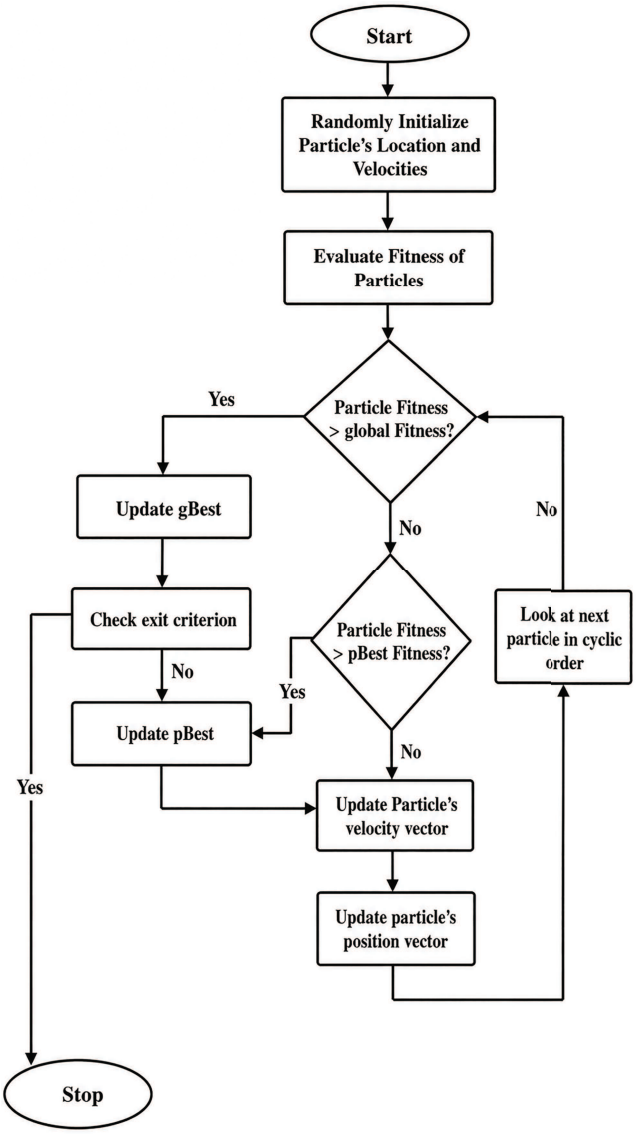


Figure 2
The flowchart of APSO

3.4. Fourth-order polynomial trajectory planning

The fourth-order polynomial scheme allows us to generate a smooth trajectory under certain constraints, where there are five constraints to be satisfied for each joint. The joint angle of the 3 DOF manipulator can be modeled as fourth-order polynomial as follows:

$$\theta(t)_i = a_{4i}t^4 + a_{3i}t^3 + a_{2i}t^2 + a_{1i}t + a_{oi} \tag{10}$$

Where i represents the angle index $i = 1 \rightarrow 3$ and the constants can be determined based on the initial and final conditions of the position and the velocity and based on the objective function to be minimized. The constant a_{oi} always refers to the initial joint angle while a_{1i} can be determined based on the initial velocity which is equal to zero in rest-to-rest motion which is preferable in the industrial manipulator. The coefficient a_{2i} is optimized based on the cost function while the coefficients a_{3i} *and* a_{4i} are determined based on a_{2i} as it represents the initial angular acceleration of each link of the manipulator. The initial conditions allow us to calculate the coefficients a_{1i}, a_{oi} while the final conditions will lead

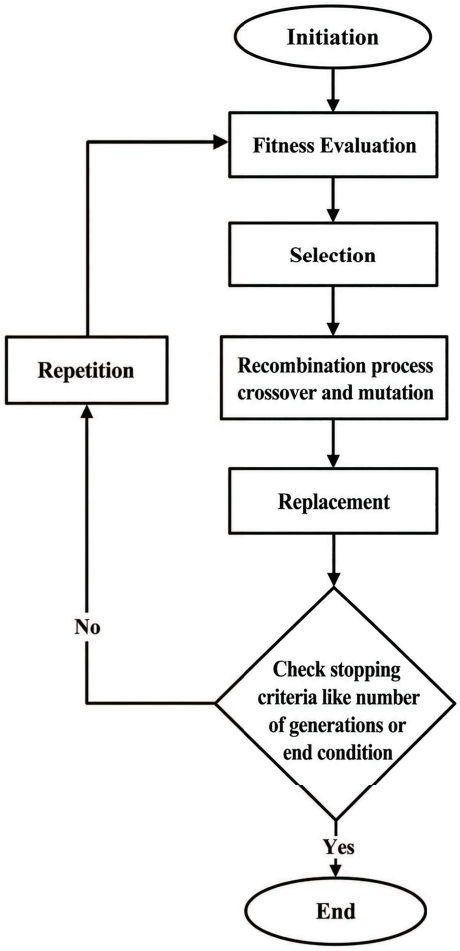


Figure 3
The flowchart of GA

to 3 unknowns (a_{4i}, a_{3i}, a_{2i}) in two equations. The coefficients (a_{4i}, a_{3i}) will be function in a_{2i}, and once we optimize a_{2i}, the value of a_{3i} *and* a_{4i} are obtained. So we have three design variables (a_{21}, a_{22}, a_{23}) for the three joints. The flowcharts for both APSO and GA are shown in Figures 2 and 3.

3.5. Mathematical model of the DC servo motor

Figure 4 shows the motor resistance and damping for the mathematical model of the DC servo motor [32]. The criteria of the energy loss per cycle can be used to perform another relevant comparison between the two configurations. The energy loss per cycle will be necessary to take into account for optimality in industrial applications.

The motor torque T_M is related to the armature current I_a by a constant factor:

$$T_M = K_T I_a \tag{11}$$

Where K_T is the motor constant. The back emf voltage V_b is related to the rotational velocity ω by:

$$V_b = K_b \dot{\theta} \tag{12}$$

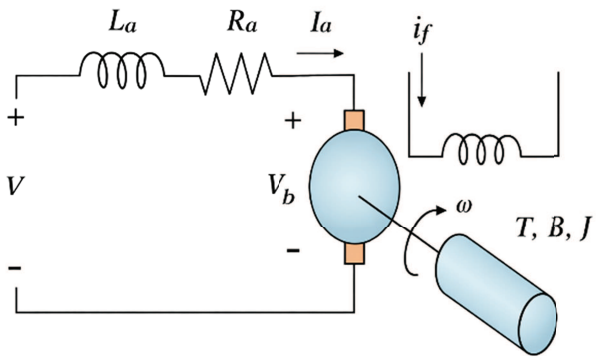


Figure 4
DC Motor modeling

The torque-speed characteristics are controlled by changing the voltage signal. By using Newton's law and Kirchhoff's law, the torque of the DC servo motor is given by:

$$J_m \frac{d\dot{\theta}}{dt} + B\dot{\theta} = K_T I_a - T_d \tag{13}$$

Where J_m is the inertia of the motor, B is the damping coefficient, and T_d is the disturbance torque and the torque load is T_L so that $T_d = \frac{T_L}{N}$. and N is motor gear ratio.

It is common in motion planning to assume that the damping coefficient $B = 0$, while the armature current is given by:

$$L_a \frac{dI_a}{dt} + R_a I_a = V_a - K_b \dot{\theta} \tag{14}$$

The energy loss per cycle E is defined as the energy loss in the motor due to the drawn current, which is reflected as heat dissipation in the resistors.

$$E = \int_0^{t_f} R_a (I_a)^2 \, dt = R_a \int_0^{t_f} \left(\frac{J_m \ddot{\theta} + T_d}{K_T} \right)^2 dt \tag{15}$$

Where the time per cycle is t_f. By assuming T_d is constant and expanding the term in bracket, one can get:

$$E = \frac{R_a (J_m)^2}{K_T^2} \int_0^{t_f} \ddot{\theta}^2 \, dt + \frac{2 R_a J_m T_d}{K_T^2} \int_0^{t_f} \ddot{\theta} \, dt + \frac{R_a T_d^2}{K_T^2} \int_0^{t_f} dt \tag{16}$$

Due to rest-to-rest motion, the second term in (16) yields to zero while the first term depends on the profile of the acceleration and the last term depends on the disturbance torque. Since the disturbance torque is constant, the dissipation energy is independent on it and the only term that will be considered in the energy dissipation is the first term and R_a, K_T are constants for all cases.

$$E = \frac{R_a (J_m)^2}{K_T^2} \int_0^{t_f} \ddot{\theta}^2 \, dt \tag{17}$$

4. Numerical Analysis

It is required to drive the end effector of the manipulator to move from starting point (0.4, 0.4, 0.2) to a goal point (0.15, 0.75, 0.65) using the fourth-order polynomial scheme for all joint angles under the constrain of minimum energy consumption. The total execution time for trajectory t_f is taken 5 seconds. There are infinite number of solutions that satisfy the fourth-order polynomial as shown in Figure 5. Our first objective is to optimize the coefficients of the 4th order polynomial of the three joint angles using APSO and GAs to select the trajectory which satisfies minimum energy consumption. The second objective is to compare between the two possible configurations elbow-up and elbow-down as they are the most common configurations using the energy loss per cycle. After the optimal trajectory is generated by the optimization, the energy loss per cycle is presented as another metric criterion for comparison between both configurations by integrating the electric model of the motor and the optimized polynomials.

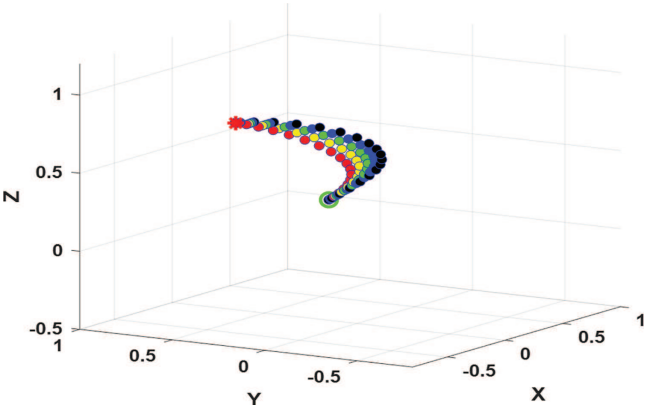


Figure 5
Different trajectories which satisfy fourth-order polynomial

The parameters employed in the simulation are listed in Table 2. The robot's workspace is strongly influenced by the robot's physical configuration, particularly a link length, joint type, and allowable joint range. The algorithm starts by defining the parameters of the manipulator, specifying the coordinates of the start and goal points. Then, the inverse kinematics is employed to calculate the joint angles for the start and goal positions due to selected configuration. After that the optimization algorithm is assigned to optimize the initial acceleration for the three joints. These accelerations are bounded by [–859.44, 859.44] deg/s^2. Moreover, the algorithm has to solve the inverse kinematic model to calculate the first three coefficients coupled in the 4th order polynomial.

The algorithm always checks that every point generated is a part of the workspace. Furthermore, discretization process is assigned to plot the final results. Finally, the kinematics properties are plotted through the operating time. Figure 6 summarized the hierarchy of the proposed algorithm.

The initial joint angles are calculated through inverse kinematics for elbow-up as:

$$\theta_{1oup} = 45 \text{ deg}, \theta_{2oup} = 16.71 \text{ deg}, \theta_{3oup} = -89.30 \text{ deg}$$

Table 2
Robot parameters

Symbol	Parameter	Value	Unit
m_1	The mass of first link	1	kg
m_2	The mass of second link	1	kg
m_3	The mass of third link	1	kg
L_1	The length of first link	0.45	m
L_2	The length of second link	0.45	m
L_3	The length of third link	0.45	m
I_{1Act}	Inertia of first actuator	0.1	kg.m^2
I_{2Act}	Inertia of second actuator	0.1	kg.m^2
I_{3Act}	Inertia of third actuator	0.1	kg.m^2
$I_1 = I_2 = I_3$	The mass moment of inertia of each link	0.016875	kg.m^2

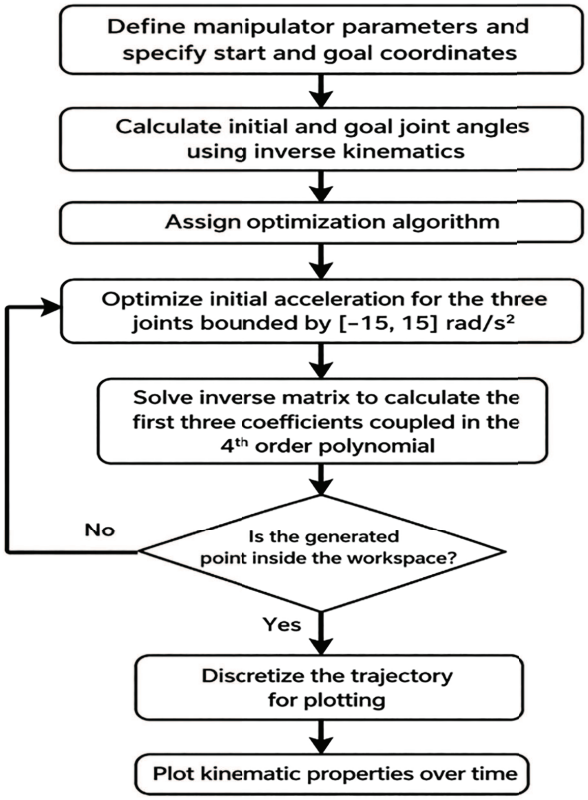


Figure 6
The hierarchy of the proposed algorithm

The final joint angles are calculated through inverse kinematics for elbow-up as:

$$\theta_{1fup} = 78.69 \text{ deg}, \theta_{2fup} = 43.20 \text{ deg}, \theta_{3fup} = -57.10 \text{ deg}$$

While in the case of elbow-down configuration the initial joint angles are calculated through inverse kinematics as:

$$\theta_{1odown} = 45 \text{ deg}, \theta_{2odown} = -72.58 \text{deg}, \theta_{3odown} = 89.30 \text{ deg}$$

The final joint angles are calculated through inverse kinematics for elbow-down as:

$$\theta_{1fdown} = 78.69 \text{ deg}, \theta_{2fdown} = -13.89 \text{ deg}, \theta_{3fdown} = 57.10 \text{ deg}$$

The total mechanical energy is the time integration of the instantaneous power of each joint angle. The total mechanical energy required throughout the trajectory was calculated through equation (18), and the objective function is defined as follows:

$$objective\ function = Mechanical\ Energy = \int_0^{tf} \sum_{i=1}^{3} |T_i \dot{q}_i| \, dt \tag{18}$$

5. Numerical Results and Discussion

The numerical simulations used to evaluate the effectiveness of the proposed framework for trajectory planning which integrate APSO or GA with fourth-order polynomial. Robotics Tool Box for MATLAB was used for the simulations of the robot configurations [33]. For the APSO, the number of particles is 35 according to sensitivity analysis to get the selected accuracy. In GA, the population size is 35 and the cross-over percent is 0.8, meaning that the cross over procedure will provide 80% of the next generation. Each parent has a 0.01 mutation rate, and the stopping criterion is satisfied when the fitness function does not improve within a tolerance of 10^{-6}. The maximum number of generations is taken as 200. The initial populations for both algorithms are taken randomly. All the optimized coefficients for both configurations are listed in Tables 3–6.

Figure 7 presents the optimum 3D trajectory of the manipulator end effector for both configurations within the workspace. Figure 7 shows that both trajectories successfully connect the initial and target positions in three-dimensional space. However, there is a clear distinction in how each configuration approaches the goal.

Figure 8 illustrates the spatial path followed by the end effector as it moves from the initial to the final position. The trajectory appears smooth and continuous, with no abrupt deviations, indicating that the optimized polynomial successfully ensures a feasible and efficient path. The motion is relatively direct, reflecting better workspace utilization and reduced unnecessary movement.

The joint angle profiles show smooth and gradual transition over time for all three joints. This behavior confirms that

Table 3
The optimized coefficients of the polynomial for elbow-up configuration using APSO

θ_1	$a_{41} = 0.0005$	$a_{31} = -0.0145$	$a_{21} = 0.0833$	$a_{11} = 0$	$a_{01} = 0.7854$
θ_2	$a_{42} = 0.0024$	$a_{32} = -0.0322$	$a_{22} = 0.1175$	$a_{12} = 0$	$a_{02} = 0.2916$
θ_3	$a_{43} = -0.0030$	$a_{33} = 0.0211$	$a_{23} = -0.0079$	$a_{13} = 0$	$a_{03} = -1.5585$

Table 4
The optimized coefficients of the polynomial for elbow-up configuration using GA

θ_1	$a_{41} = 0.0005$	$a_{31} = -0.0145$	$a_{21} = 0.0833$	$a_{11} = 0$	$a_{01} = 0.7854$
θ_2	$a_{42} = 0.0025$	$a_{32} = -0.0322$	$a_{22} = 0.1175$	$a_{12} = 0$	$a_{02} = 0.2916$
θ_3	$a_{43} = -0.0030$	$a_{33} = 0.0211$	$a_{23} = -0.0079$	$a_{13} = 0$	$a_{03} = -1.5585$

Table 5
The optimized coefficients of the polynomial for elbow-down configuration using APSO

θ_1	$a_{41} = 0.0045$	$a_{31} = -0.0541$	$a_{21} = 0.1822$	$a_{11} = 0$	$a_{01} = 0.7854$
θ_2	$a_{42} = 0.0055$	$a_{32} = -0.0715$	$a_{22} = 0.2608$	$a_{12} = 0$	$a_{02} = -1.2668$
θ_3	$a_{43} = 0.0257$	$a_{33} = -0.2479$	$a_{23} = 0.5748$	$a_{13} = 0$	$a_{03} = 1.5585$

Table 6
The optimized coefficients of the polynomial for elbow-down configuration using GA

θ_1	$a_{41} = 0.0044$	$a_{31} = -0.0536$	$a_{21} = 0.1811$	$a_{11} = 0$	$a_{01} = 0.7854$
θ_2	$a_{42} = 0.0058$	$a_{32} = -0.0746$	$a_{22} = 0.2684$	$a_{12} = 0$	$a_{02} = -1.2668$
θ_3	$a_{43} = 0.0254$	$a_{33} = -0.2448$	$a_{23} = 0.5671$	$a_{13} = 0$	$a_{03} = 1.5585$

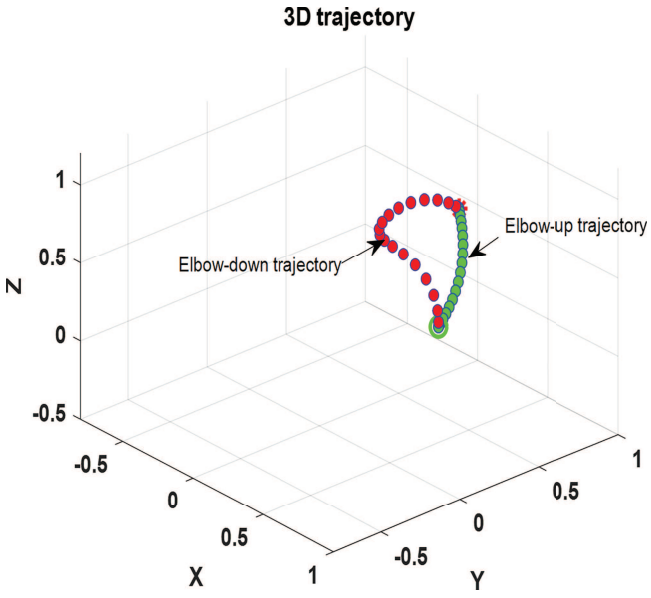


Figure 7
The 3D trajectory of the manipulator end effector for both configurations

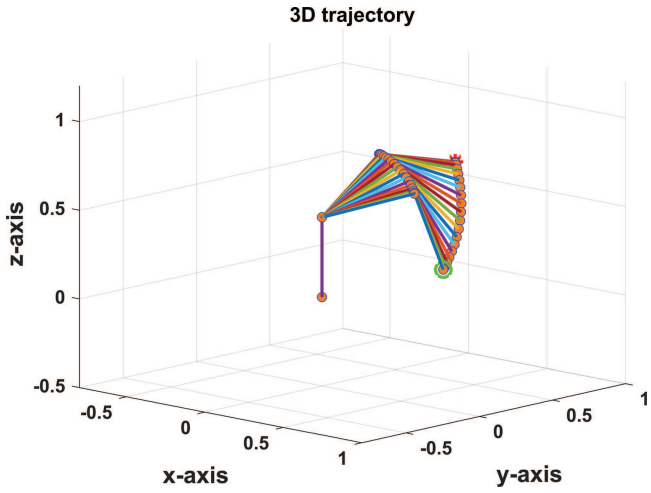


Figure 8
3D trajectory of manipulator with elbow-up configuration using APSO

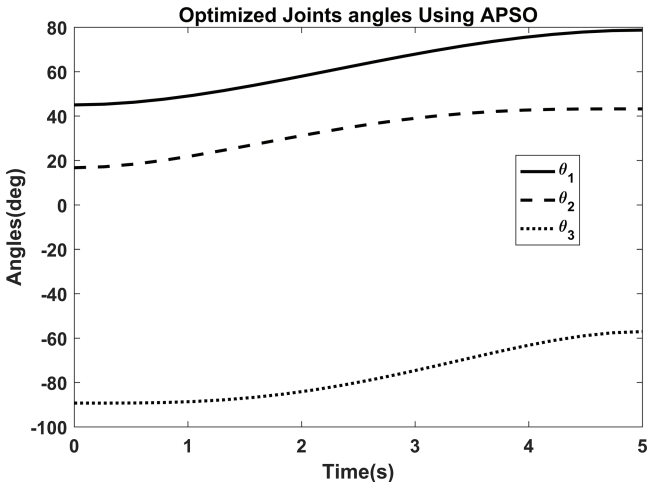


Figure 9
The joint angles for elbow-up using APSO

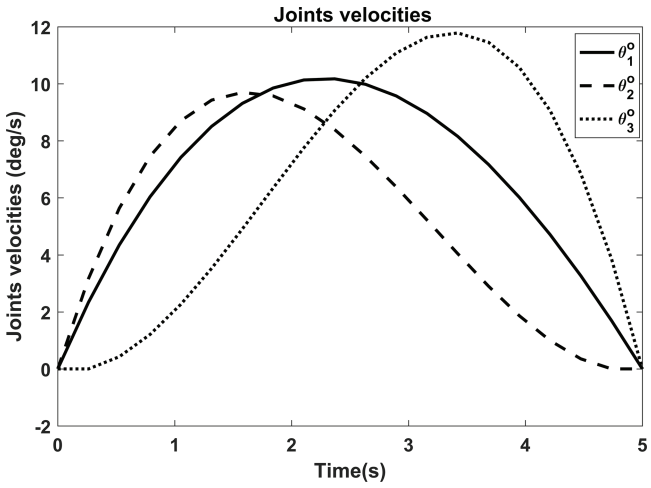


Figure 10
The joint velocities for elbow-up using APSO

the fourth-order polynomial provides continuity in position and avoids sudden changes as shown in Figure 9. The velocity curves as depicted in Figure 10 demonstrate continuous and bounded behavior throughout the motion. The absence of sharp peaks indicates that the manipulator operates within acceptable dynamic limits, which is essential for maintaining control accuracy. Figure 11 shows the acceleration profiles change smoothly with time, without sudden spikes. This reflects a well-optimized trajectory that minimizes dynamic shocks and reduces mechanical stress on the system components. The torques required to drive each joint during motion are shown in Figure 12. The torque profiles are smooth and moderate in magnitude, indicating efficient energy usage. It can be noticed that the torque required from the first joint is smaller compared with the second and the third joint as it acts in the z axis parallel to the weight while the second joint has to deal with the weight of the successive links.

In Figure 13, the convergence curve shows a rapid decrease in the objective function during the initial iterations, followed by gradual stabilization. This behavior indicates that APSO efficiently finds the optimal solution with stable convergence characteristics.

In the elbow-down configuration as shown in Figure 14, the trajectory is still smooth and continuous. However, it has more curves and less direct compared to the elbow-up case. This increases the path complexity and leads to less efficient motion which contributes to higher energy consumption. Despite that the joint angles vary smoothly over time similar to the elbow-up case. The range of joint angles in the elbow-down configuration is larger compared to elbow-up as shown in Figure 15. The velocity profiles remain continuous but have slightly higher magnitudes compared to the elbow-up configuration as depicted in Figure 16. Figure 17 presents the acceleration profiles of the three joints. Although the acceleration curves exhibit smooth behavior, their peaks are comparatively higher. This implies that the system is under more dynamic demand, which could result in increased mechanical stress and energy consumption. The torque profiles as

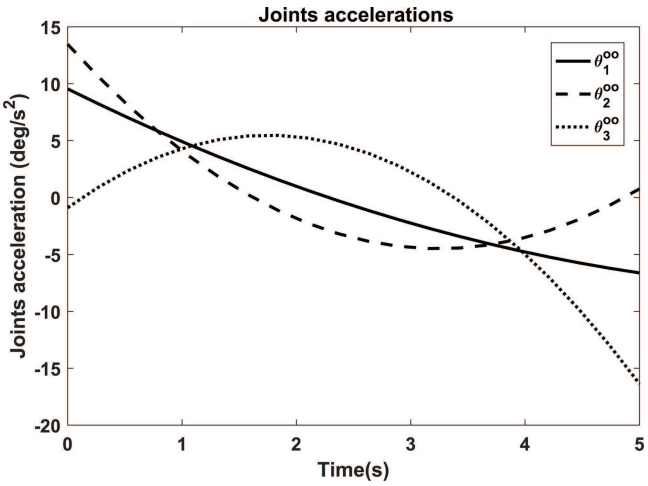


Figure 11
The joint accelerations for elbow-up using APSO

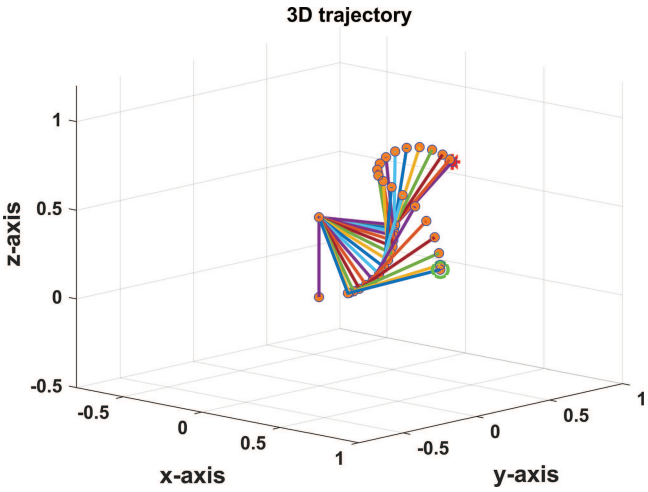


Figure 14
3D trajectory of manipulator with elbow-down configuration using APSO

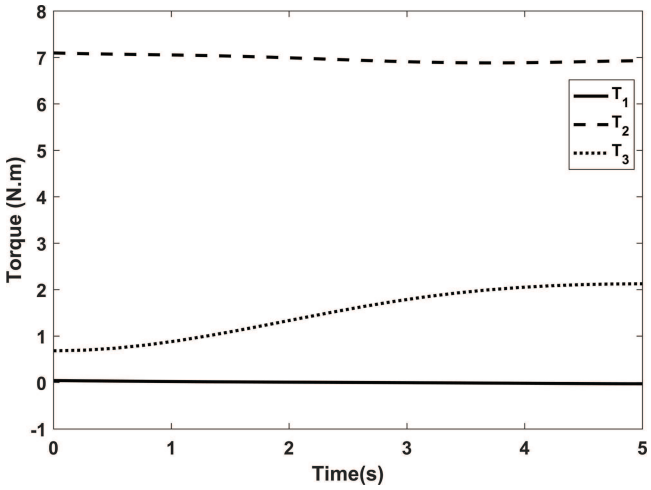


Figure 12
The torques required to drive the joints for elbow-up using APSO

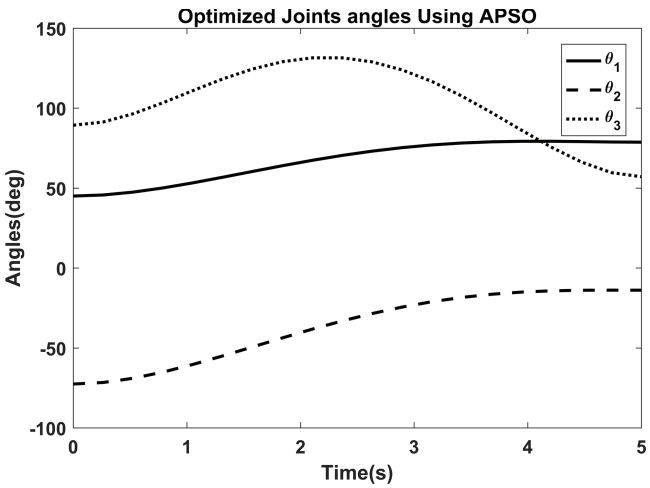


Figure 15
The joint angles for elbow-down using APSO

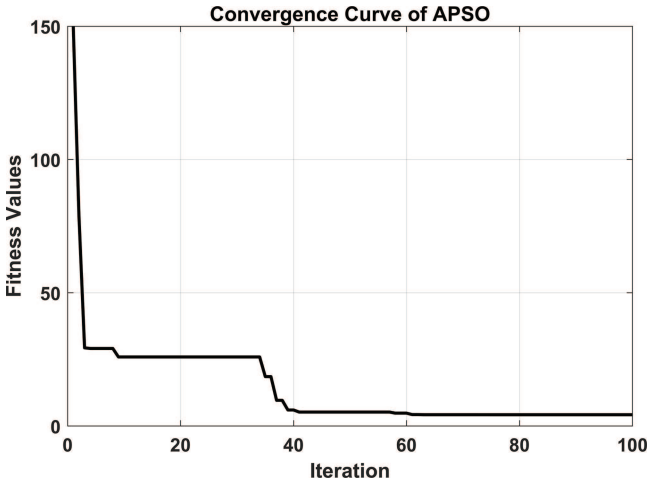


Figure 13
Convergence curve of APSO for elbow-up configuration

shown in Figure 18 are noticeably higher than the torques in the elbow-up configuration. This increase explains the higher energy consumption, as more effort is required to drive the joints during the trajectory.

The convergence behavior remains stable and similar to the elbow-up as shown in Figure 19, with rapid initial improvement followed by gradual convergence. This confirms the robustness of APSO regardless of the configuration. Figures 20, 21 present the convergence curve of the GA algorithm. The GA takes 100 generations for both configurations to reach to the optimal solution and satisfy the stability. Table 4 summarizes the computed total energy consumption for each configuration and optimization method. The results clearly show that the energy required is strongly dependent on the selected configuration. Also, Table 7 provides a comparative analysis between APSO and GA. It is observed that the elbow-up configuration achieves the lowest energy consumption 4.1986 (J). The elbow-down configuration requires significantly higher energy 7.0745 (J). The energy required

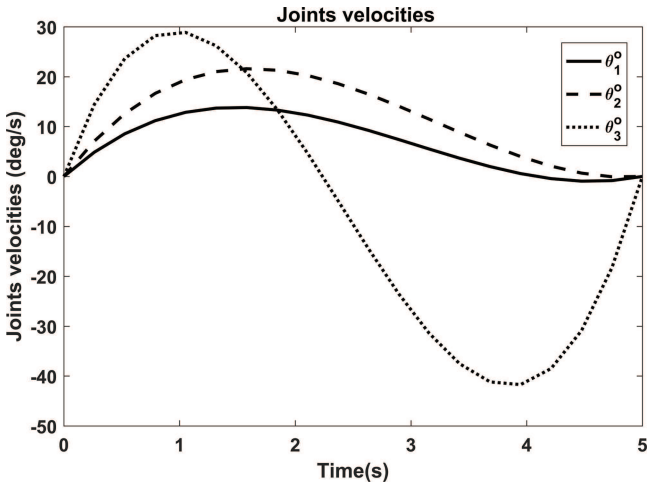


Figure 16
The joint velocities for elbow-down using APSO

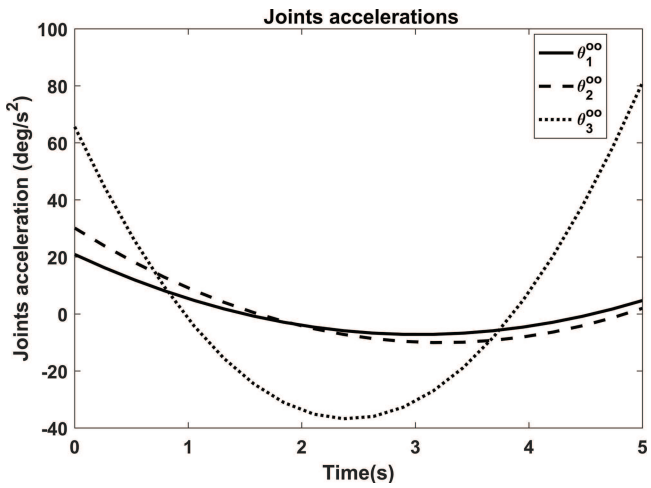


Figure 17
The joint accelerations for elbow-down using APSO

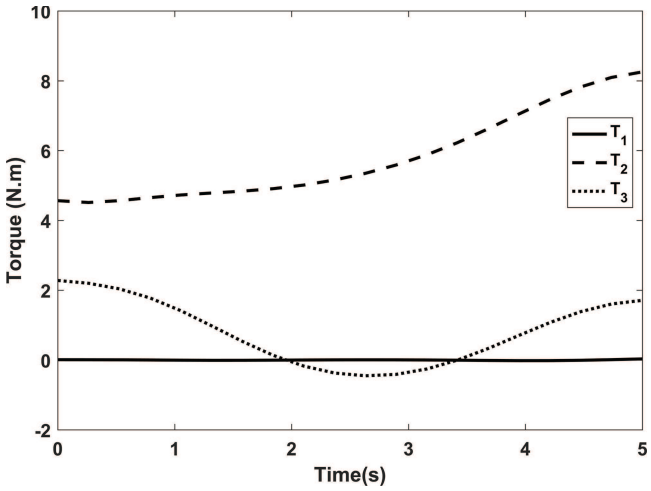


Figure 18
The torques required to drive the joints for elbow-down using APSO

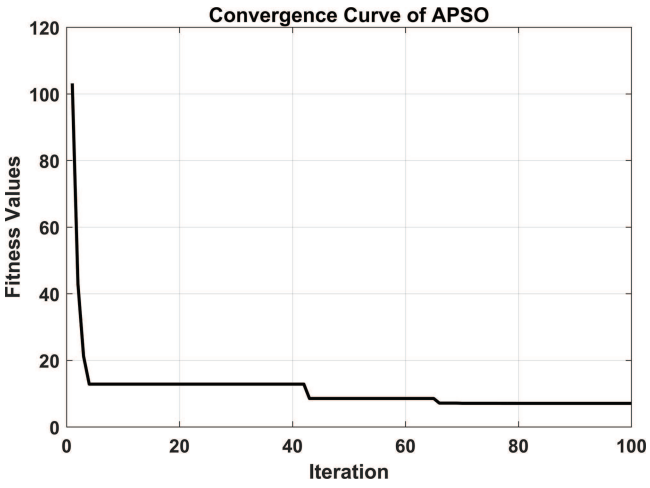


Figure 19
Convergence curve of APSO for elbow-down configuration

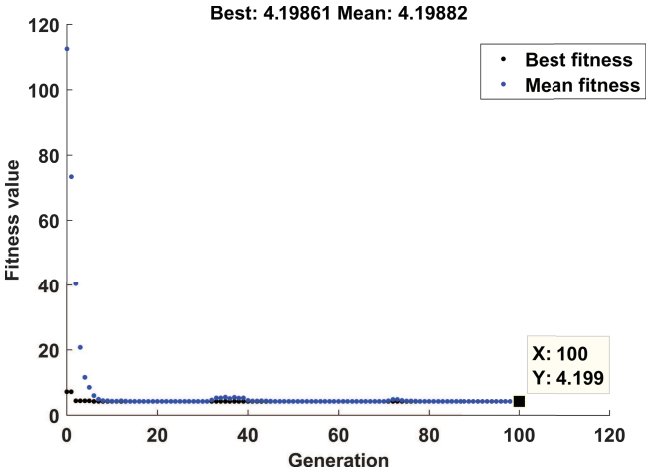


Figure 20
Convergence curve of GA for elbow-up configuration

from the elbow-up is less than elbow-down by 40% as shown in table 7. Both APSO and GA yield nearly identical energy values confirming the accuracy of the obtained solutions. The statistical results demonstrate that APSO provides superior computational efficiency compared to GA. The average execution time of APSO was 16.230 s, compared to 21.258 s for GA, which represents 23.65% reduction in computational time as illustrated in Table 8. It can be noted that the standard deviation is very small which means good repeatability with small variability. As a result, the coefficient of variation is small which means low variability and good consistency. Overall, APSO demonstrates a clear advantage in computational speed, while both algorithms exhibit relatively low variability, with coefficient of variation values below 6%.

The expressions given for the energy loss per cycle provide a physical meaningful way to compare the efficiency of the two configurations independent of specific operating conditions.

The small coefficient (0.1164) of the energy loss per Cycle E reflects that the elbow-up configuration operates with relatively low energy losses. This is mainly due to smoother motion, lower torque requirements, and reduced resistive dissipation in the motor, while the coefficient is significantly larger (1.9239)

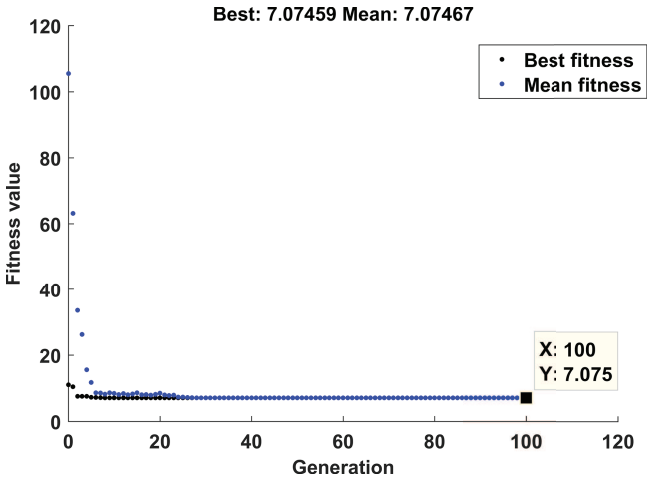


Figure 21
Convergence curve of GA for elbow-down configuration

Table 7
Comparison of the total mechanical energy along the trajectory for both configurations

Configuration	Total Energy (J)
Elbow-up "APSO"	4.1986
Elbow-up "GA"	4.1986
Elbow-down "APSO"	7.0745
Elbow-down "GA"	7.0746

Table 8
Comparison between the APSO and GA algorithms

Algorithm	Average Time per 25 runs (s)	Standard deviation	Coefficient of variation
Adaptive Particle Swarm Optimization (APSO)	16.230	0.962	5.93
GA	21.258	1.144	5.38

in the elbow-down configuration. This implies that for the same motor parameters, the elbow-down configuration consumes $\frac{1.9239}{0.1164} = 16.54$ times more energy per cycle than the elbow-up configuration under the same system parameters. The comparison is summarized in table 9.

The joint friction may slightly underestimate absolute energy values, while the relative comparison between configurations remains valid, since the joint velocities in the elbow-down configuration are higher than the elbow-up configuration as shown in Figures 10 and 16. It should be noticed that energy is the integration of the instant power and was done through trapezoidal rule of integration.

Table 9
Comparison between the two configurations in term of energy loss per cycle

Configuration	Energy loss per cycle (J)
Elbow-up	$E_{Elbow-up} = 0.1164 * \frac{R * \mathrm{J}_m^2}{\mathrm{K}_\mathrm{T}^2}$
Elbow-down	$E_{Elbow-down} = 1.9239 * \frac{R * \mathrm{J}_m^2}{\mathrm{K}_\mathrm{T}^2}$

6. Conclusions

This study demonstrates that optimal trajectory planning using a fourth-order polynomial combined with meta-heuristic optimization techniques is highly effective in minimizing energy consumption for a 3 DOF manipulator. The fourth-order polynomial generates smooth trajectory which reduces the jerk phenomena and increases the time life of the motors. The results clearly indicate that the elbow-up configuration is more energy-efficient by 40 % than the elbow-down configuration due to its smoother trajectory and lower torque requirements. Although both APSO and GA achieve nearly identical optimal energy values, APSO is superior to the GA in terms of faster convergence and lower computational time. Moreover, the analysis of energy consumption per cycle highlights its importance in real-world industrial applications, where repeated operations magnify the impact of energy efficiency. Selecting an optimal configuration and optimization algorithm can therefore lead to significant long-term energy savings. We found that the elbow-down configuration consumes approximately 16.5 times more energy loss per cycle than the elbow-up configuration. Overall, the combination of proper trajectory design configuration selection and efficient optimization algorithms plays a vital role in enhancing the performance, efficiency and reliability of robotic systems. The optimal solution will be affected by the joints' physical limits, and it should be considered as extra constraints in the optimization process.

7. Future Works

A few things should be considered in the future research, like the examination of how joint friction influences performance behavior. The fitness function could be extended to address obstacle avoidance. Comparison of the APSO to other meta-heuristics like the Squirrel Search Algorithm and Ant Colony Optimization could be performed. Also, increasing the degree of polynomial of the trajectory to satisfy certain constraints like jerk condition can be investigated.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data are available from the corresponding author upon reasonable request.

Author Contribution Statement

Ahmed H. Shahein: Methodology, Software, Formal analysis, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Mohammed Ghazy:** Conceptualization, Validation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration. **Atef A. Ata:** Conceptualization, Validation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration.

References

[1] Field, G., & Stepanenko, Y. (1996). Iterative dynamic programming: An approach to minimum energy trajectory planning for robotic manipulators. In *Proceedings of IEEE International Conference on Robotics and Automation*, *3*, 2755–2760. https://doi.org/10.1109/ROBOT.1996.506579

[2] Xu, X. R., Wang, X. G., & Qin, F. (2000). Trajectory planning of robot manipulators by using spline function approach. In *Proceedings of the 3rd World Congress on Intelligent Control and Automation*, *2*, 1215–1219. https://doi.org/10.1109/WCICA.2000.863436

[3] Ata, A. A. (2007). Energy minimization of redundant manipulators using generalized pattern search. *Journal of Information and Optimization Sciences*, *28*(5), 775–790. https://doi.org/10.1080/02522667.2007.10699773

[4] Ata, A. A., Faris, W. F., & Sa'adeh, M. Y. (2007). Optimal trajectory selection for a three DOF manipulator with minimum energy consumption. *International Journal of Applied Engineering Research*, *2*(1), 45–63.

[5] Gregory, J., Olivares, A., & Staffetti, E. (2012). Energy-optimal trajectory planning for robot manipulators with holonomic constraints. *Systems & Control Letters*, *61*(2), 279–291. https://doi.org/10.1016/j.sysconle.2011.11.005

[6] Wang, Y., Ueda, K., & Bortoff, S. A. (2013). A Hamiltonian approach to compute an energy efficient trajectory for a servomotor system. *Automatica*, *49*(12), 3550–3561. https://doi.org/10.1016/j.automatica.2013.09.019

[7] Pellicciari, M., Berselli, G., Leali, F., & Vergnano, A. (2013). A method for reducing the energy consumption of pick-and-place industrial robots. *Mechatronics*, *23*(3), 326–334. https://doi.org/10.1016/j.mechatronics.2013.01.013

[8] Liu, Y., Li, X., Jiang, P., Du, Z., Wu, Z., Sun, B., & Huang, X. (2022). Evolutionary multi-objective trajectory optimization for a redundant robot in Cartesian space considering obstacle avoidance. *Mechanical Sciences*, *13*(1), 41–53. https://doi.org/10.5194/ms-13-41-2022

[9] Ye, J., Hao, L., & Cheng, H. (2024). Multi-objective optimal trajectory planning for robot manipulator attention to end-effector path limitation. *Robotica*, *42*(6), 1761–1780. https://doi.org/10.1017/S0263574724000481

[10] Lorenz, M., Paris, J., Schöler, F., Barreto, J. P., Mannheim, T., Hüsing, M., & Corves, B. (2017). Energy-efficient trajectory planning for robot manipulators. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 58189*, V05BT08A067. https://doi.org/10.1115/DETC2017-67198

[11] Sehgal, A., Ward, N., La, H., & Louis, S. (2022). Automatic parameter optimization using genetic algorithm in deep reinforcement learning for robotic manipulation tasks. *arXiv Preprint:2204.03656*.

[12] Nonoyama, K., Liu, Z., Fujiwara, T., Alam, M. M., & Nishi, T. (2022). Energy-efficient robot configuration and motion planning using genetic algorithm and particle swarm optimization. *Energies*, *15*(6), 2074. https://doi.org/10.3390/en15062074

[13] Massaro, M., Lovato, S., Bottin, M., & Rosati, G. (2023). An optimal control approach to the minimum-time trajectory planning of robotic manipulators. *Robotics*, *12*(3), 64. https://doi.org/10.3390/robotics12030064

[14] Soori, M., Arezoo, B., & Dastres, R. (2023). Optimization of energy consumption in industrial robots, a review. *Cognitive Robotics*, *3*, 142–157. https://doi.org/10.1016/j.cogr.2023.05.003

[15] Akbari, A., Mozaffari, S., Singh, R., Ahmadi, M., & Alirezaee, S. (2023). Movement optimization of robotic arms for energy and time reduction using evolutionary algorithms. In *2023 International Symposium on Signals, Circuits and Systems,* 1–4. https://doi.org/10.1109/ISSCS58449.2023.10190949

[16] Van Oosterwyck, N., Vanbecelaere, F., Knaepkens, F., Monte, M., Stockman, K., Cuyt, A., & Derammelaere, S. (2024). Energy optimal point-to-point motion profile optimization. *Mechanics Based Design of Structures and Machines*, *52*(1), 239–256.

[17] Zhou, Y., Han, G., Wei, Z., Huang, Z., Chen, X., & Wu, J. (2024). Optimal trajectory planning of robot energy consumption based on improved sparrow search algorithm. *Measurement and Control*, *57*(7), 1014–1021. https://doi.org/10.1177/00202940231220080

[18] Shrivastava, A. (2025). Exploring optimal motion strategies: A comprehensive study of various trajectory planning schemes for trajectory selection of robotic manipulator. *Journal of The Institution of Engineers (India): Series C*, *106*(2), 691–710. https://doi.org/10.1007/s40032-025-01171-2

[19] Elgohr, A. T., Khater, H. A., & Mousa, M. A. (2025). Trajectory optimization for 6 DOF robotic arm using WOA, GA, and novel WGA techniques. *Results in Engineering*, *25*, 104511. https://doi.org/10.1016/j.rineng.2025.104511

[20] Yang, D., Wei, X., & Han, M. (2025). Research on energy consumption optimization strategies of robot joints based on NSGA-II and energy consumption mapping. *Robotics*, *14*(10), 138. https://doi.org/10.3390/robotics14100138

[21] Xiao, G., Ma, Y., & Li, Y. (2025). Energy-efficient trajectory optimization for planetary surface manipulators via heuristic multi-objective particle swarm optimization algorithm. *Robotica*, *43*(4), 1411–1432. https://doi.org/10.1017/S026357472500027X

[22] Fan, Y., Peng, Y., & Liu, J. (2025). Advanced multi-objective trajectory planning for robotic arms using a multi-strategy enhanced NSGA-II algorithm. *PLoS ONE*, *20*(5), e0324567. https://doi.org/10.1371/journal.pone.0324567

[23] Ueno, Y., & Tachiya, H. (2025). Study on trajectory optimization of robot by heuristic and learning. *International Journal of Automation Technology*, *19*(3), 315–325. https://doi.org/10.20965/ijat.2025.p0315

[24] Zhang, M., Liu, J., Wu, Y., Hou, T., & Li, T. (2025). Multi-objective trajectory planning for robotic arms based on MOPO algorithm. *Electronics*, *14*(12), 2371. https://doi.org/10.3390/electronics14122371

[25] Wei, J., Wang, J., Yang, M., & Feng, B. (2026). Trajectory planning method for a robotic arm based on an improved

multi-objective golden jackal optimization algorithm. *Sensors*, *26*(9), 2696. https://doi.org/10.3390/s26092696

[26] Li, T., & Liu, J. (2026). Multi-objective optimization of energy consumption and obstacle avoidance path planning for robotic arms. *Experimental Technology and Management*, *43*(3), 80–90. https://doi.org/10.16791/j.cnki.sjg.2026.03.011

[27] Espinosa, E. C., Prieto, A. F., Osorio, S. A. M., & Guardado, P. F. (2017). Modelos cinemático y dinámico de un robot de cuatro grados de libertad [Kinematic and dynamic models of a four-degree-of-freedom robot]. *Revista Científica de Ingeniería Electrónica, Automática y Comunicaciones*, *38*(3), 56–75.

[28] Niku, S. B. (2020). *Introduction to robotics: Analysis, control, applications*. USA: Wiley & Sons.

[29] Ashagrie, A., Salau, A. O., & Weldcherkos, T. (2021). Modeling and control of a 3-DOF articulated robotic manipulator using self-tuning fuzzy sliding mode controller. *Cogent Engineering*, *8*(1), 1950105. https://doi.org/10.1080/23311916.2021.1950105

[30] Shahein, A. H., Ghazy, M., & Ata, A. A. (2026). Optimal trajectory planning of 6 DOF manipulator using meta-heuristics integrated with artificial potential field theory. *Alexandria Engineering Journal*, *134*, 263–274. https://doi.org/10.1016/j.aej.2025.12.018

[31] Holland, J. H. (1992). Genetic algorithms. *Scientific American*, *267*(1), 66–73. https://www.jstor.org/stable/24939139

[32] Solihin, M. I., Kamal, M. A. S., & Legowo, A. (2008). Objective function selection of GA-based PID control optimization for automatic gantry crane. In *2008 International Conference on Computer and Communication Engineering*, 883–887. https://doi.org/10.1109/ICCCE.2008.4580732

[33] Corke, P. (2021). *Robotics and control: Fundamental algorithms in MATLAB®*. Germany: Springer Nature.

How to Cite: Shahein, A. H., Ghazy, M., & Ata, A. A. (2026). Energy Minimization of Spatial 3 DOF Manipulator for Smooth Optimal Trajectory Planning Using Meta-Heuristics. *Journal of Climbing and Walking Robots*. https://doi.org/10.47852/bonviewJCWR620210678

Appendix

$$M_{11} = I_{1Act} + \frac{m_1 L_1^2}{3} + \frac{m_2 L_2^2}{6} + \frac{m_3 L_2^2}{2} + \frac{m_3 L_3^2}{6} + \frac{m_3 L_3^2 c(2\theta_2 + 2\theta_3)}{6} + \frac{m_2 L_2^2 c(2\theta_2)}{6} + \frac{m_3 L_2^2 c(2\theta_2)}{2} + \frac{m_3 L_2 L_3 c(\theta_3)}{2} + \frac{m_3 L_2 L_3 c(2\theta_2 + \theta_3)}{2}$$

$$M_{12} = M_{13} = M_{21} = M_{31} = 0$$

$$M_{22} = I_{2Act} + \frac{m_2 L_2^2}{3} + m_3 L_2^2 + \frac{m_3 L_3^2}{3} + m_3 L_2 L_3 \, c(\theta_3)$$

$$M_{23} = \frac{m_3 L_3^2}{3} + m_3 L_2 L_3 \frac{c(\theta_3)}{2}$$

$$M_{32} = m_3 L_3 \left[\frac{L_3}{3} + \frac{L_2}{2} c(\theta_3) \right]$$

$$M_{33} = \left[I_{3Act} + \frac{m_3 L_3^2}{3} \right]$$

$$C_{11} = C_{12} = C_{13} = C_{22} = C_{33} = 0$$

$$C_{21} = m_2 L_2^2 \frac{s(2\theta_2)}{6} + m_3 L_2^2 \frac{s(2\theta_2)}{2} + \frac{m_3 L_3^2 s(2\theta_2 + 2\theta_3)}{6} + \frac{m_3 L_2 L_3 s(2\theta_2 + \theta_3)}{2}$$

$$C_{23} = -m_3 L_3 L_2 \frac{s(\theta_3)}{2}$$

$$C_{31} = m_3 L_3 \left[\frac{L_3}{6} s(2\theta_2 + 2\theta_3) + \frac{L_2}{4} s(\theta_3) + \frac{L_2}{4} s(2\theta_2 + \theta_3) \right]$$

$$C_{32} = m_3 L_3 L_2 \frac{s(\theta_3)}{2}$$

$$R_{11} = -\frac{m_3 L_3^2 s(2\theta_2 + 2\theta_3)}{3} - \frac{m_2 L_2^2 s(2\theta_2)}{3} - m_3 L_2^2 s(2\theta_2) - s(2\theta_2 + \theta_3)$$

$$R_{12} = -\frac{m_3 L_3^2 s(2\theta_2 + 2\theta_3)}{3} - \frac{m_3 L_2 L_3 s(\theta_3)}{2} - \frac{m_3 L_2 L_3 s(2\theta_2 + \theta_3)}{2}$$

$$R_{13} = R_{21} = R_{22} = R_{31} = R_{32} = R_{33} = 0$$

$$R_{23} = -m_3 L_3 L_2 \, s(\theta_3)$$

$$G_1 = 0$$

$$G_2 = \frac{m_3 L_3}{2} \, c(\theta_2 + \theta_3) + m_3 L_2 \, c(\theta_2) + \frac{m_2 L_2}{2} \, c(\theta_2)$$

$$G_3 = \frac{m_3 L_3 \, c(\theta_2 + \theta_3)}{2}$$