Nat. Volatiles & Essent. Oils, 2021; 8(5): 1583 - 1592

NVEO
Natural Volatiles & Essential Oils

An efficient Hybrid global optimizer for accurate modeling of manipulator dynamics in minimum-time trajectory-planning problems

[1]Dr. Anandakumar Haldorai, [2]Dr. Rudresh Deepak Shirwaikar, [3]Dr. L. Rama Parvathy, [4]Dr. S. Markkandan, [5]Dr. Kovendan AKP, [6]Dr.Ram Subbiah

1 Professor (Associate), Department of Computer science and engineering, Sri Eshwar College of Engineering, Coimbatore, Tamil Nadu, India. 641202 anandakumar.psgtech@gmail.com
2 Assistant Professor, Dept. of ISE, BMSIT&M, Doddaballapur Main Road, Avalahalli, Yelahanka, Bengaluru, Karnataka 560064. rudreshshirwaikar@gmail.com

3 Professor, CSE Department, Saveetha School of Engineering, SIMATS, Chennai-602105 ramaparvathyl.sse@saveetha.com

4 Department of ECE, SRM TRP Engineering College, Trichy, Tamilnadu. India markkandan.s@trp.srmtrichy.edu.in

5 Assistant professor, Department of ECE, St Joseph's College of engineering, OMR, Chennai 119. kpkvendan@gmail.com

6 Associate Professor, Mechanical engineering, Gokaraju Rangaraju institute of Engineering and Technology, Hyderabad. ram4msrm@gmail.com

Abstract

This research focuses on examining the minimum time trajectory planning problems for a robot. With the model of manipulator dynamics, we use a two-segment proportional-integration model that includes the squared jerk and trajectory integration to provide an overall execution time (including integration time) and the squared jerk segment as an approximation for the acceleration derivative. The augmented Lagrange multiplier (ALM) technique, which adds Lagrange multipliers, is used to help maximise the functional goal. The new hybrid global optimizer that has been created for each of the initial processes in the manipulator dynamics prevents a localised effective value from forming because a decade-long history of past effective values is saved and reused during the iterative analysis process, keeping the previous decade's effective value constant. Simulation findings show that the suggested method works for a three-DOF planar robot, and a three-degree-of-freedom (3-DOF) robot will allow for time-optimal trajectory planning.

Index keys- Hybrid global optimizer, Augmented Lagrange Multiplier, Minimum-time trajectory planning, manipulator dynamics.

1. Introduction

To accomplish motion, the torque actuator for the robot manipulator's electrical motor must be greater than the inertial torsion of the joint actuator movement [1, 2]. A proportionate application of energy to the actuator torque is required for the movement. Reduced actuator torque based on joint trajectory would save energy and extend the operating life of the robotic manipulator [3]. Artificial intelligence (AI) techniques, such as adaptive computing optimization approaches, may be used to reduce actuator torque if the joint torque optimization issue is defined correctly [4–6].

The parameters optimization and the technical development are the foundations of a robot's optimum trajectory planning. The minimal execution time, lowest energy, as well as least jerk seems to be the most important optimization requirements. In Cartesian coordinates, trajectory planning is simple, as well as the end-effector's trajectory with its orientation is visible. However, issues induced by kinematic singularities aren't considered in this technique [7–10]. These issues can be avoided by using shared space trajectory planning. Interpolating functions that fulfill kinematically as well as dynamic constraints yields joint trajectory conditions. In joint space, a systematic controller-based trajectory is easy to alter [11–13]. [14] presents an optimizing time trajectory planning technique with acceleration, velocity, and jerk restrictions. This optimization approach uses a hybrid optimization approach to tackle the issue by taking into account path specification, robot kinematics, as well as restrictions. The generated trajectory is ideal in terms of time, but it lacks smoothness. The minimum jerk trajectory's point-to-point route planning outcome is achieved utilising an analytic method. Experiments [15] support the minimum-time/jerk strategy. A simple analytical approach suggests linearizing the industrial robot's feature expression [16].

A multi-objective goal of the path planning method is introduced for robot navigation depending on particle swarm optimization (PSO) [17]. There have been two reference qualities regarded: the degree of danger as well as the path distance. The findings revealed that the suggested method is a feasible alternative for addressing the robot planning issue, but for in the case of robotic planar. A hybrid method named ACO-PSO is introduced [18]. To address the path planning issue, the Ant Colony Optimization (ACO) method is utilized to build the matching solution depending on Swarm Intelligence (SI). The research is done for path planning of robots in a two - dimensional space along with obstacles. Distinct hybrid optimization methods are utilized with restricted variables to create an optimum trajectory for a planar redundancy manipulator employing revolute joints using the least amount of electricity possible [19]. They examined using two trajectories that had the same beginning angular coordinates but distinct end locations. The results indicated that for every single trajectory, both hybrid approaches produced similar outcomes, but for the second trajectory path, the Genetic–Particle Swarm methodology performed better.

2. Related Works

The genetic algorithm (GA) is utilized to optimize robot manipulator routes in terms of joint torque for elementary robotic manipulators having 2 degrees of freedom (DOFs) [19], and a similar technique may be utilized for more sophisticated robotic manipulators. In point-to-point planning, GA is utilized to demonstrate the three-link (redundancy) robot arm. Within this situation, the study showed that GA is being an efficient optimization approach, suitable for multi-objective goal optimization. In a point-to-point and pick-and-place scenario, GA is used to improve joint angles of a manipulator system containing a three-link planar in terms of route controlling and accuracy. The article indicates how GA may be used to enhance search performance and approximations while dealing with complicated path control. To optimize a point-to-point route, it is being interpolated using the 4th polynomial in terms of energy used via the robotic manipulator as well as traveling time with an upper limit torque. In addition, the article contrasts the outcomes of a GA and a geometrical path planning technique. The path is planned using GA, then the findings demonstrate a reduction in energy use.

Wei et al. illustrate that evolutionary algorithms may be used to generate collective behaviors, particularly work allocation [20]. To solve the difficulties of bootstrap as well as deception, they suggested a two-step approach for job division and assignment. The findings demonstrate that the proposed technique outperforms the traditional evolutionary robotics method in terms of performance. Abu-Dakka et al. use an evaluation of variances testing to examine the effectiveness of a multi-populated GA for time and smoother trajectory optimization [21]. The researchers note that utilizing a multi-populated GA is legitimate and delivers rapid results. Larsen et al. demonstrate how to compute collision-free routes using evolutionary algorithms as well as quickly exploring random trees (RRT) with varied variables on a robotic system include KUKA R3100 as well as KUKA KR210 robots on a shared axis [22]. The results demonstrate the evolutionary algorithms outperform sampling approaches and also establishing that GA outcomes are superior to RRT findings. An end-to-end control rule for robotic swarms using a deep Q-learning Artificial Intelligence algorithm [23]. It is demonstrated that by employing such approaches, control policies may be developed with less computational resources, particularly in instances when solution spaces are vast. El Haiek et al. [24] explain how to use evolutionary algorithms to design the route of a 3-DOF spherical robot to minimize traveling time and joint moving length. Polynomial interpolation has been used to compute the route, which is point-to-point. According to the findings, utilizing a Genetic Algorithm, the robot may effectively discover an optimum, collision-free trajectory despite decreasing the trajectory's duration and length.

ROS is a set of software toolkits for creating robotic applications by piecing together relatively basic algorithms to create a modular solution to a complicated issue [25]. As [26] demonstrates, there exist open-source educational systems with robots of different degrees of independence that may offer academic teaching to interested people for free.

3. Proposed Methodology

3.1. Minimum-time Trajector Planning Problem:

Initially, the manipulator modeling includes the geometric path is supplied, and this optimum trajectory-planning concept has been provided. Thus, using inverse kinematics, the positioning points in the joint space manipulator corresponding to the discrete operational space namely Cartesian space may be acquired; then, using the cubic or polynomial splines technique, the joint trajectory manipulator of the robot has been generated by linking the appropriate position points. Since these resulting trajectories contain cubic splines, constant acceleration values are commonly used in the trajectory planning of the manipulator dynamics. Furthermore, unlike cubic splines, higher-order polynomials are capable of overcoming severe oscillations and then overshoot issues among every set of reference points. The combined trajectory-planning manipulator is implemented in this study using cubic splines. The accuracy of trajectory planning manipulator dynamics is primarily determined by the goal function. The following is the definition of the functional goal:

$$\min f(h) = K_T N \sum_{i=1}^{n-1} h_i + \alpha K_J \sum_{i=1}^{N} \sum_{i-1}^{n-1} \left[\frac{(\ddot{Q}_{j,i+1} - \ddot{Q}_{j,i})^2}{h_i} \right] \quad (1)$$

State that $\max\{|\dot{Q}_{j,i}(t_i)|, |\dot{Q}_{j,i}(t_i *)|, |\dot{Q}_{j,i}(t_{i+1})|\} - V_{jm} \leq 0$ (2)

$$\max\{|\ddot{Q}_{j,1}(t_1)|, |\ddot{Q}_{j,2}(t_2)|, |\ddot{Q}_{j,i}(t_n)|\} - A_{jm} \leq 0 \quad (3)$$

$$max \left| \frac{\ddot{Q}_{j,i}(t_{i+1}) - \ddot{Q}_{j,i}(t_i)}{h_i} \right| - J_{jm} \leq 0 \quad (4)$$

$$\sum_{i=1}^{n-1} h_i - T_m \leq 0 \qquad j = 1, \dots\dots, N \ \forall i = 1, \dots\dots, n-1 \quad (5)$$

From (3), $K_T + K_J = 1$ is achieved by changing the K_T value and K_J value to get the best overall execution time as well as jerk effect. The coefficient of elastic is included to equalize the impacts of the entire execution time as well as the proportionate to the integration of squared jerk since they could have a considerable variation in quantity grading. As a result, the minimum-time trajectory-planning issue of the manipulator dynamics under constraints may be addressed by searching for the best response for the period h_i utilizing (3).

The hybrid global optimizer algorithmic approach is nothing but the integration of Particle Swarm Optimization with Ant colony optimization (ACO) algorithmic approach. The relevant solution is established using ant colony optimization (ACO), and certain important algorithm stages are laid forth. The parameters of ACO are optimized using particle swarm optimization (PSO), and the parameters can be set self-adaptively. The Augmented Lagrange Multiplier (ALM) technique has several benefits over the penalty function, including numerical stability, then estimated efficiency.

3.2. Constrained Hybrid global Optimizer:

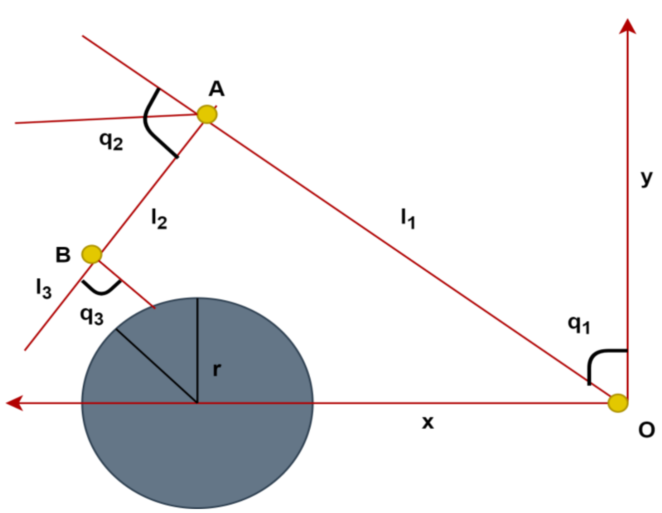

Figure 1: 3 degree-of-freedom (DOF) planar robot.

The hybrid global optimizer constriction factor χ is implemented to guarantee that the hybrid global optimizer converged as follows:

$$\chi = \frac{2}{\left|2-\varphi-\sqrt{\varphi^2-4\varphi}\right|} \tag{6}$$

The hybrid global optimizer's particle velocity and location are revised in the following way:

$$v_j^i(k+1) = \chi\left[v_j^i(k) + c_1 r_1\left[p_j^i(k) - x_j^i(k)\right] + c_2 r_2\left[g_j - x_j^i(k)\right]\right] \tag{7}$$

$$x_j^i(k+1) = x_j^i(k) + v_j^i(k+1) \quad 1 \leq i \leq n \,\forall 1 \leq j \leq N_{d,max}$$

Researchers frequently specify $\varphi = c_1 + c_2$ and put $c_1 = c_2 = 2.05$ in expression (6), φ resulting in 4.1 with χ = 0.729. From (7): k represents the number of repetitions; c_1 as well as c_2 are positive acceleration factors; r_1 as well as r_2 are two values distributed randomly in the interval of [0, 1]; $p_j^i(k)$ represents the individual most appropriate position; g_j represents the global appropriate position. The constrained hybrid global optimizer algorithm, which may achieve an effective balance among globalized and localized searching, is now known as the hybrid global optimizer.

3.3. Method: Augmented Lagrange Multiplier (ALM)

The ALM approach is familiarly otherwise known as the Lagrange multiplier method, but unlike the latter, the former requires fewer experiments to evaluate an acceptable beginning penalty factor as well as the dynamic condition penalty factor doesn't seem to be infinity; thus, the former outperforms the latter concerning numerical stability and computationally efficient. The optimization model may be stated as follows for generalized inequality constraints:

$\min f(X)$

Declare that, $g_j(X) \leq 0$

$$j = 1, \ldots, m \tag{8}$$

The following is a description of its ALM approach:

$$L(X, \lambda, r) = f(X) + \sum_{j=1}^{m} \lambda_j \varphi_j + r_j \varphi_j^2 \tag{9}$$

The independent variable is denoted by $X = (x_1, \ldots, x_n)$, the functional goal is denoted by (X), the inequality constraints are denoted by $g_j(X) \leq 0$ $(j = 1, \ldots, m)$, the Lagrange multiplier is denoted by λ_j, as well as the penalty factor is denoted by r_j. The following is a description of φ_j:

$$\varphi_j = max\left(g_j(x), \frac{-\lambda_j}{2r_j}\right) \qquad j = 1, \ldots, m \tag{10}$$

By addressing the initial constrained issue (8), the solution may be found using (9). The Lagrange multiplier has iteratively obtained as follows:

$$\lambda_j^{v+1} = \lambda_j^v + 2r_j\varphi_j \tag{11}$$

Here, γ is how often the Lagrange multiplier is upgraded. The expression for the revised penalty factor is described in the following:

$$r_i^{v+1} = \begin{cases} 2r_j^v & |g_j(X^v)| > |g_j(X^{v-1})| \wedge |g_j(X^v)| > \varepsilon_g \\ \frac{r_j^v}{2} |g_j(X^v)| > \varepsilon_g \\ r_j^v & else, \end{cases} \quad (12)$$

Here, ε_g stands for constraint error precision. The following are the termination criteria when $\gamma = 1$:

$$c = max\{max[0, g_j(X^v)] \le \varepsilon; j = 1, \dots, m \ \forall i = 1, \dots, v_{max}\} \quad (13)$$

If v_{max} is more than $\gamma = 2$, the termination criteria are given as follows:

$$|f(X^v) - f(X^{v-1})| \wedge c \le \varepsilon \quad (14)$$

Here, v_{max} represents the update maximum times and then ε signifies the convergence accuracy. If the penalty factor doesn't seem to exceed infinity, the ALM technique may ensure that perhaps the optimal solution will be discovered, something neither the interior nor the exterior penalty functioning method can promise.

3.4. Lagrange multiplied Algorithm for constrained hybrid global optimization:

Because the hybrid global optimizer and ALM each have their own set of benefits, the proposed algorithmic approach combines the hybrid global optimizer and the ALM to give a novel technique for addressing nonlinearly constrained issues that make use of both. Such a challenge is the minimum-time trajectory-planning modeling of manipulator dynamics. The proposed algorithm is made up of the stages below.

Step 1: Setting $k_{max}, N_{d,max}, v_{max}, n, r_j^0, \lambda_j^0, \varepsilon, \varepsilon_g, v = 1 \ and \ k = 0$ Convert the restricted issue (8) into an unconstrained issue (9).

Step 2: $N_{d,max}$ particles are created with randomly determined locations and velocities, as well as the related fitness values, are evaluated at every position.

Step 3: Examine the condition for termination (13). The process ends with the appropriate value $x^* = x_{swarm}^{best}$ whether they are happy; when they're not, the acquired appropriate value x* is stored. Setting k=1 then establish $N_{d,max}$ particles at randomized locations and velocities.

Step 4: Substitute one set of particles with the appropriate value acquired from the previous generation.

Step 5: Assess the proper fitness levels at each stage to improve the particles and determine the optimal p_{g-best}^k, use the formula (7).

Step 6: Examine the Lagrange multiplier and penalty factor revised for how many times. Whether it is fulfilled, verify the termination criteria (14); whether they are met, the appropriate value $x^* = x_{swarm}^{best}$

has been attained; If they aren't met, change the According to Lagrange multipliers and penalty factors (11) and (12), assign V = V+1. Continue to step 5 unless the revised amount of times both the penalty factor and the Lagrange multiplier are not fulfilled.

To prevent falling into the localized appropriate value, a novel hybrid global optimizer is produced in each beginning process in the proposed algorithmic approach, and the appropriate value can be discovered more efficiently and rapidly as its previous generation's appropriate value is stored and produced to the preceding generation during the iterative searching procedure.

1. Experimental Analysis

To optimize the robot manipulator's trajectory, it must first be inverted (inverse kinematics). Inverse kinematics, required for repetitive manipulator dynamics, is shown by using a control layout method. Also, using this method, less computer resources are required, which means it is ideal for running a redundant robot. We take use of the control layout approach's DLS technique to find the minimum-time robot's kinematics.

The inverse kinematics positional solution may then be found by integrating (11). To test the technique's accuracy and dependability, a robot runs on a trajectory-planning scenario using the proposed algorithmic approach. MATLAB R2014 has been used to compile the software. The testing item is a space robot that can control the positioning as well as the posture of its holding in the manipulator modeling platform; this is a 3-DOF robot planar like the one illustrated in Figure 2 and Figure 3 for joint 1. The task under testing is to follow a circular course in the plane. Joint angles such as q_1, q_2, as well as q_3 are included in Figure 1. Set l_1, l_2, as well as l_3 are 2,070 mm, 2,070 mm, and 430 mm seem to be the robot's geometric specifications. The control layout technique is used to calculate the inverse kinematics of the robot since it has a minimum-time degree of freedom. The manipulator end effector's starting position is (4000, 0) in mm, is proportional to the joint angles of (30°, -49.03°, -38.27°), the tracking circle's center as well as the radius r are (3600, 0) in mm and 400 (mm), correspondingly.

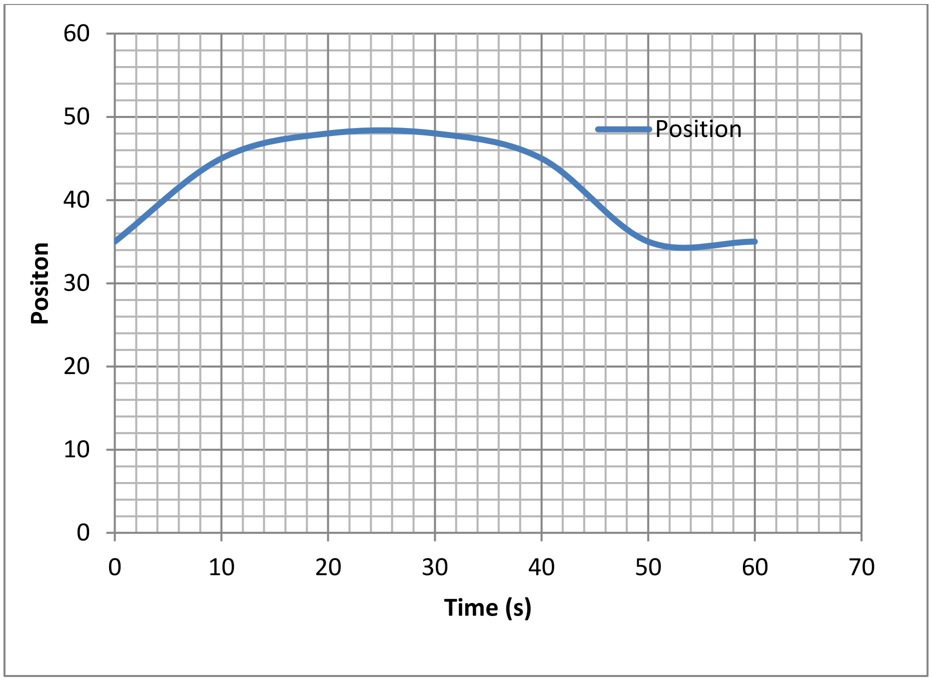

Figure 2: Position joint trajectory.

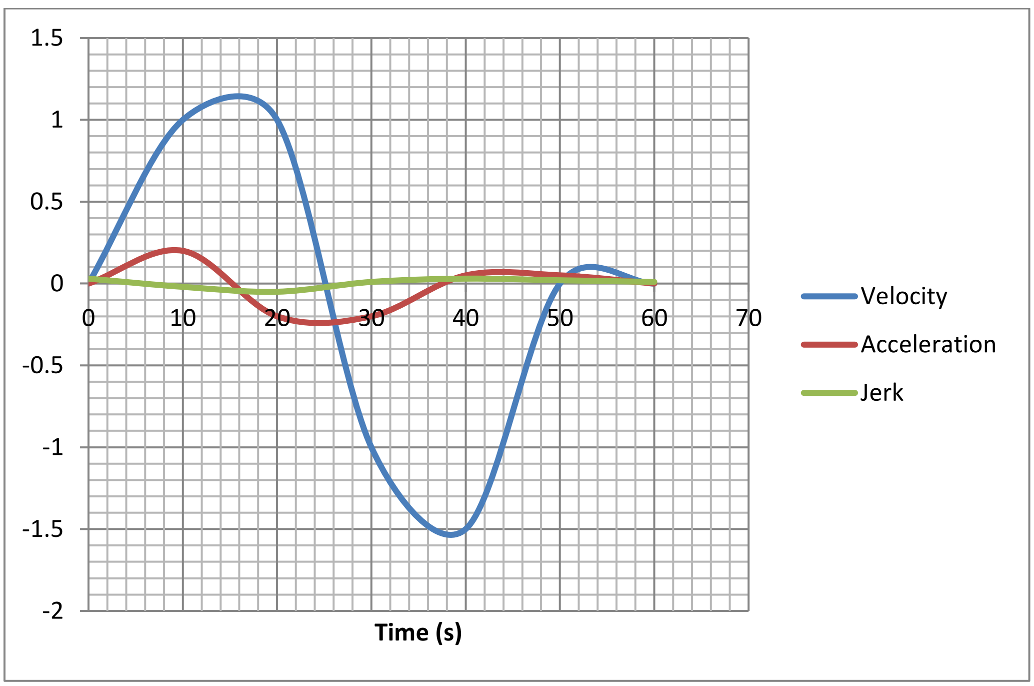

Figure 3: Trajectory-planning using optimizer.

1. Conclusion

This article presents a hybrid global optimizer algorithm as the path planner's recommended algorithmic strategy, which uses PSO and ACO in combination with ALM methods to address the trajectory planning issue during the simulation of robot dynamics. By having a fresh global optimizer come into existence each time in the beginning step, we prevent the possibility of converging into a locally optimal solution. To get the most relevant value, the method searches iteratively to preserve as many earlier values as possible and then passes on these values to the next generation. Following the strategy outlined, the functional goal employs both the overall time to complete the movement and the incorporation of the squared jerk into the trajectory of the manipulator dynamics. The two weighting coefficient values and the elastic coefficient value are then tweaked to meet criteria such as quick execution and smoother trajectory. A 3-DOF planar robot is used for the method. The simulation results show that the kinematics and execution time restrictions may be met via a minimal time trajectory. As a result, the suggested strategy has not been shown to be useful, and future research will instead attempt to evaluate the effectiveness of the approach under development.

REFERENCE

1. Pfeiffer F and Johanni R, "A concept for manipulator trajectory planning", IEEE J Robot Autom 1987; 3(2): 115–123.
2. Wang Y and Xu Q. Design and fabrication of a soft robotic manipulator driven by fiber-reinforced actuators. In: 2018 IEEE international conference on mechatronics, robotics and automation (ICMRA), Hefei, China, 18–21 May 2018, pp. 157–161. IEEE.
3. Sharma GS, Singh M and Singh T. Optimization of energy in robotic arm using genetic algorithm. Int J Comp Sci Technol 2011; 2(2): 315–317.
4. Suresh, P., Saravanakumar, U., Celestine Iwendi, Senthilkumar Mohan, Gautam Srivastav 2020 "Field-programmable gate arrays with low power vision system using dynamic switching" Computers & Electrical Engineering, Vol. 90, 2021, 106996.
5. Albert F, Koh S, Chen C, et al. Optimizing joint angles of robotic manipulator using genetic algorithm. In: International conference on computer engineering and applications IPC-SIT (ed Othman M), vol. 2. 2011, pp. 134–140. Singapore: IACSIT Press.
6. Nassehi A, Essink W and Barclay J. Evolutionary algorithms for generation and optimization of tool paths. CIRP Annals 2015; 64(1): 455–458.
7. C. Zheng, Y. Su, and P. C. M¨uller, "Simple online smooth trajectory generations for industrial systems," Mechatronics, vol. 19, no. 4, pp. 571–576, 2009.
8. S. Zhang, A. Maria Zanchettin, R. Villa, and S. Dai, "Real-time trajectory planning based on joint-decoupled optimization in human-robot interaction," Mechanism and Machine Theory, vol. 144, 2020.
9. Suresh. P., "Creation of optical chain in the focal region of high NA lens of tightly focused higher order Gaussian beam", Springer – Journal of Optics, Vol. 46, p. 225 – 230, 2017.
10. S. Chiaverini, B. Siciliano, and O. Egeland, "Experimental results on controlling a 6-DOF robot manipulator in the neighborhood of kinematic singularities," in Experimental Robotics III. Lecture Notes in Control and Information Sciences, T. Yoshikawa and F. Miyazaki, Eds., vol. 200, pp. 1–13, Springer, Berlin, Germany, 1994.
11. H. Wang, H. Wang, J. Huang, B. Zhao, and L. Quan, "Smooth point-to-point trajectory planning for industrial robots with kinematical constraints based on high-order polynomial curve," Mechanism and Machine Theory, vol. 139, pp. 284– 293, 2019.

12. G. Chen, B. Yuan, Q. Jia, Y. Fu, and J. Tan, “Trajectory optimization for inhibiting the joint parameter jump of a space manipulator with a load-carrying task,” Mechanism and Machine Theory, vol. 140, pp. 59–82, 2019.
13. X. Liu, C. Qiu, Q. Zeng, and A. Li, “Kinematics Analysis and Trajectory Planning of collaborative welding robot with multiple manipulators,” Procedia CIRP, vol. 81, pp. 1034– 1039, 2019.
14. J. Dong, P. M. Ferreira, and J. A. Stori, “Feed-rate optimization with jerk constraints for generating minimum-time trajectories,” International Journal of Machine Tools & Manufacture, vol. 47, no. 12-13, pp. 1941–1955, 2007.
15. V. Zanotto, A. Gasparetto, A. Lanzutti, P. Boscariol, and R. Vidoni, “Experimental validation of minimum time-jerk algorithms for industrial robots,” Journal of Intelligent & Robotic Systems, vol. 64, no. 2, pp. 197–219, 2011.
16. Y. Zhang, D.W. Gong, J.H. Zhang, Robot path planning in uncertain environment using multi-objective particle swarm optimization, Neurocomputing 103 (2013) 172–185.
17. C. Shi, Y. Bu, Z. Li, J. Tan, Solving path planning problem by an ACO-PSO hybrid algorithm, School of Mechanical and Electronical Engineering, Center South University, Changsha, 2007, Tech. Rep.
18. M.E. Mohamed, T.M. Khaled, A.A. Atef, Power optimization of planar redundant manipulator moving along constrained-end trajectory using hybrid techniques, AEJ 56 (4) (2017) 439–447.
19. Garg DP and Kumar M. Optimization techniques applied to multiple manipulators for path planning and torque minimization. Eng Appl Artif Intel 2002; 15(3–4): 241–252.
20. Wei Y, Hiraga M, Ohkura K, et al. Autonomous task allocation by artificial evolution for robotic swarms in complex tasks. Artif Life Robot 2019; 24(1): 127–134.
21. Abu-Dakka FJ, Assad IF, Alkhdour RM, et al. Statistical evaluation of an evolutionary algorithm for minimum time trajectory planning problem for industrial robots. Int J Adv Manuf Technol 2017; 89(1–4): 389–406.
22. Suresh P., Aanandha Saravanan K., Celestine Iwendi, Ebuka Ibeke, Gautam Srivastava, “An artificial intelligence based quorum system for the improvement of the lifespan of sensor networks,” IEEE sensors journal, Vol. 21, Issue: 15, pp. 17373 – 17385, 2021.
23. Wei Y, Nie X, Hiraga M, et al. Developing end-to-end control policies for robotic swarms using deep Q-learning. J Adv Comput Intel Intel Inform 2019; 23(5): 920–927.
24. El Haiek D, Aboulissane B, El Bakkali L, et al. Optimal trajectory planning for spherical robot using evolutionary algorithms. Procedia Manuf 2019; 32: 960–968.
25. Suresh, P., Ravikumar O, Hari Krishna Mahesh K, Sri Aashritha S, 2020 “Content Extraction Through Chatbots With Artificial Intelligence Techniques,” International Journal of Scientific & Technology Research, Vol. 9, Issue 02, pp. 1960-1963.
26. S. Manzoor, R. Ul Islam, A. Khalid, A. Samad, J. Iqbal, “An open-source multi-DOF articulated robotic educational platform for autonomous object manipulation”, Robotics and Computer-Integrated Manufacturing, 30, 2014, pp. 351–362.