Birlesik Dünya Arastırma BD CENTER Innovasyon ve Yayıncılık Merkezi

Global Journal of Computer Sciences: Theory and Research

Volume 12, Issue 1, (2022) 42-54

www.gjcs.eu

Time-optimal trajectory generation in joint space for 6R industrial serial robots using Cuckoo search algorithm

Oguzhan Karahan [a1]**,** Department of Electronics and Communications Engineering, Kocaeli University, Kocaeli, Turkey.

Hasan Karci [b], Department of Electronics and Communications Engineering, Kocaeli University, Kocaeli, Turkey.

Ali Tangel [c], Department of Electronics and Communications Engineering, Kocaeli University, Kocaeli, Turkey.

Suggested Citation:

Karahan, O., Karcı, H. & Tangel, A. (2022). Time-optimal trajectory generation in joint space for 6R industrial serial robots using Cuckoo search algorithm. *Global Journal of ComputerSciences: Theory and Research*. *12*(1), 42-54. https://doi.org/10.18844/gjcs.v12i1.7447

Received from January 07, 2022; revised from March 25, 2022; accepted from April 13, 2022.

Selection and peer-review under the responsibility of Assist. Prof. Dr. Ezgi Pelin Yıldız, Kafkas University.

Abstract

The trajectory planning problem in industrial robotic applications has recently attracted the great attention of many researchers. In this paper, an optimal trajectory planning approach is proposed based on optimal time by utilizing the interpolation spline method. The method including a combination of cubic spline and 7th order polynomial is used for generating the trajectory in joint space for robot manipulators. Cuckoo Search (CS) optimization algorithm is chosen to optimize the joint trajectories based on objective, including minimizing total traveling time along the whole trajectory. The spline method has been applied to the PUMA robot for optimizing the joint trajectories with the CS algorithm based on the objective. With the trajectory planning method, the joint velocities, accelerations, and jerks along the whole trajectory optimized by CS meet the requirements of the kinematic constraints in the case of the objective. Simulation results validated that the used trajectory planning method based on the proposed algorithm is very effective in comparison with the same methods based on the algorithms proposed by earlier authors.

Keywords: CS; industrial robots; interpolation; spline methods; trajectory planning.

* ADDRESS FOR CORRESPONDENCE: Oguzhan KARAHAN*, Department of Electronics and Communications Engineering, Kocaeli University, Kocaeli, Turkey.
E-mail: oguzhan.karahan@kocaeli.edu.tr

1. Introduction

The use of industrial robots in production systems with the development of automation technology has grown tremendously over the past few decades [1]. Especially, the robotic manipulators in industrial assembly lines and production systems are widely used today due to their ability to achieve particular tasks with improved speed, reliability, and quality. In these industrial activities, trajectory planning for industrial robots working in industrial environments is a very important issue. Accordingly, the trajectory planning problem in industrial robotic applications has recently attracted the great attention of many researchers.

The capability of trajectory planning algorithms is restricted by the time needed for carrying out the trajectory and also by the physical limitations of the robot [1]. Most trajectory planning algorithms have proposed an objective function consisting of execution time, actuator effort, and the absolute value of the jerk for performing certain criteria such as minimum execution time, minimum jerk, and minimum kinetic energy consumed by the robot [2]. The concept of minimum-time trajectory was firstly introduced [3-5].

To enhance manufacturing productivity and movement stability, trajectory planning with a short execution time and smooth profile is required. Spline functions have a major role in improving smooth trajectory to ensure velocity continuity, acceleration continuity, or jerk continuity in the motion of industrial robots. Lin, Chang, and Luh [6] constructed joint trajectories for industrial robots using cubic splines based on minimum time trajectory through the polyhedron search method. Aribowo and Terashima [7] proposed a combination of input shaping with cubic spline optimization for generating the minimum time trajectory of the robot arm. Sequential Quadratic Programming (SQP) was used for solving the constrained trajectory planning optimization problem outlined in that paper.

Gasparetto and Zanotto [8] suggested a technique based on the B-spline for realizing the jerk continuity in the joint trajectory planning by minimizing both time and jerk during optimization with the SQP. In Gasparetto, & Zanotto, [9], the same authors analyzed the results obtained from running the SQP algorithm based on an objective function comprising a weighted sum of time and jerk using cubic splines and fifth-order B-splines. Liu, Lai, and Wu [10] presented a smooth trajectory planning approach using a combination of the planning with multi-degree splines in Cartesian space and multi-degree B-splines in joint space for robot manipulators with kinematic constraints. In that paper, for solving the minimum execution time problem during optimal trajectory planning, the SQP method was used. Chen and Chen [5] used the B-spline curve by fitting this curve through an approximation method and then to plan the motion trajectory, the intermediate points were interpolated by using the S-Curve feed-rate profile for the trajectories of the tooltip and tool axis of a 6-DOF robot manipulator.

On the other hand, for the 6-DOF robot manipulator, Perumaal and Jawahar [11] presented an automated trajectory planner to obtain a smooth and minimum-time trajectory based on the synchronized trigonometric S-Curve trajectory technique with jerk constraints. Moreover, in that paper, suitable examples and comparisons with the cubic spline-based trajectory were demonstrated. With fifth-order B-spline or quantic polynomial spline proposed in [12,13], continuous motion in the position, velocity, acceleration, and jerk is provided. But, for obtaining a smooth trajectory, the least six coefficients must be solved. Accordingly, computation complexity may arise. To overcome this problem, Boryga and Graboś [14] designed a trajectory planning mode with a high-degree polynomial for serial link manipulators. In that paper, the acceleration profile of the tool-tip was planned to utilize the determination of only one polynomial coefficient using the property of the root multiplicity.

1.1. Related studies

Recently, interpolation functions have been utilized for generating a trajectory under kinematical or dynamical constraints in joint space for industrial robots. In the literature on the trajectory planning problem, the optimal motion planning for robotic manipulators can be considered as an optimization problem applied to the nature-inspired optimization algorithms using an objective function for minimizing execution (traveling) time. Several natures inspired optimization algorithms developed by researchers have been successfully applied to the trajectory optimization for industrial robots.

Machmudah et al. [15] employed the GA and Particle Swarm Optimization (PSO) algorithms by minimizing the total traveling time and the torque under kinematic and dynamic limitations to find the feasible joint trajectory with the high degree polynomial curve for the robotic arm in the obstacle environment. For a space robotic system composed of the 6-DOF spacecraft and the 7-DOF redundant manipulator, Yang et al. [16] employed PSO to find the optimal solutions using the execution time criterion to construct the fifth order Bèzier curve in joint space under the velocity and acceleration boundaries. Kucuk [17] developed a minimum-time smooth motion trajectory in joint space by combining cubic spline with the 7th order polynomial for serial and parallel manipulators. Savsani et al. [18] employed the TLBO and ABC algorithms by minimizing traveling time, traveling distance, and total joint Cartesian lengths for planning point-to-point trajectory in joint space with high order polynomial spline for a 3-DOF robotic arm.

In Savsani et al., [19], the same authors presented a comparative study of the proposed heuristic solution approaches for planning the trajectories in joint space based on high order polynomial spline by minimizing three different objective functions for a 3-DOF robotic arm. Wang et al. [20] presented an optimal joint trajectory planning method using the DE algorithm with specific objectives under kinematic limitations through the Bèzier curve for a 7-DOF kinematically redundant manipulator. Ficarella et al. [21] proposed the robot's optimum trajectory planning in joint space via the five-order polynomial function for a 6-DOF robot. In that paper, for the robot optimum trajectory planning, the hybrid real code population based on incremental learning and DE was used based on the two objective functions of minimizing trajectory time and jerk under kinematic constraints of all joints.

1.2. Purpose of study

Motivated by the above-mentioned studies, the central objective of this work is to present a comparative study of the S-curve profiles based on interpolation splines, i.e., the combination of cubic spline and 7th order polynomial model for point-to-point movements. For obtaining the optimal trajectory from initial to intermediate positions and from intermediate to final position in joint space under kinematic constraints, the CS-based optimization algorithm is employed by minimizing the objective function, namely, the traveling time objective function. Considering the concept of planning the trajectory optimization addressed in the aforementioned works, the main contributions of this study are: (I) to provide an independent comparison of the used interpolation methods for solving optimal joint trajectory planning problems, (II) to present the values of maximum joint velocities, accelerations, jerks generated by the used methods for each joint trajectory, (III) to demonstrate the maximal values of all kinematic variables of each joint along the path segments according to the combination of cubic spline and 7th order polynomial, (IV) to compare the obtained results with that of the other optimization algorithms proposed in the literature based on the same interpolation method for each joint along the path from initial to the final point.

2. Materials and Methods

Trajectory planning for industrial robots can be specified either in Cartesian space or joint space. The trajectory in Cartesian space can be constructed based on a set of consecutive intermediate points between the start and endpoints. In view of the kinematic and dynamic constraints imposed on the robot joints, complex modeling and heavy mathematical computations appear in Cartesian space, but trajectory planning in joint space is not the case. Therefore, planning a trajectory in joint space is preferred rather than trajectory planning in Cartesian space. By solving the inverse kinematic problem related to via-points determined in the Cartesian space, the corresponding counterparts in joint space can be found. Thus, a suitable trajectory is generated between two consecutive points under kinematic constraints such as displacement, velocity, acceleration, and jerk.

In joint trajectory planning between the start position and the final position, the motion constraints specified in the datasheet of the robot manipulator must be considered for each joint. Kinematic constraints for each joint in trajectory planning are defined as:

$$|\theta_j(t)| \le \theta_j^{max}, \; j = 1, \dots, N$$
$$|\dot{\theta}_j(t)| \le \dot{\theta}_j^{max}, \; j = 1, \dots, N$$
$$|\ddot{\theta}_j(t)| \le \ddot{\theta}_j^{max}, \; j = 1, \dots, N$$
$$|\dddot{\theta}_j(t)| \le \dddot{\theta}_j^{max}, \; j = 1, \dots, N \qquad (1)$$

In trajectory planning, it is expected that a smooth motion trajectory is generated by minimizing mechanical vibration and traveling time for achieving greater productivity with applying simultaneously kinematic limitations. The optimization problem is solved under kinematic constraints given above by minimizing the equation defined as follows:

$$f = \sum_{i=1}^{n} h_i \qquad (2)$$

As can be observed from the objective function given above, the total duration of the trajectory is aimed to be minimized by function f. The symbol explanations aforementioned equations above are presented in Table 1.

In this work, trajectory planning is applied between any pair of consecutive intermediate points based on the minimization of the objective function given above. Also, to apply the optimization algorithm, splin functions such as cubic spline, and high order polynomial spline for the interpolation are chosen.

TABLE I
Meaning of symbols in the optimization formulations.

Symbol	**Meaning**	**Symbol**	**Meaning**
N	Number of robot joints	$\dot{\theta}_j(t)$	Velocity of the *j^th^* joint
n	Number of segments	$\ddot{\theta}_j(t)$	Acceleration of the *j^th^* joint
h_i	Time interval between the i^{th} via-point and $(i+1)^{th}$ via-point	$\dddot{\theta}_j(t)$	Jerk of the j^{th} joint
t_f	Total traveling time of the trajectory	θ_j^{max}	Displacement constraint for the j^{th} joint
K_t	Weight of the term proportional to the traveling time	$\dot{\theta}_j^{max}$	Velocity constraint for the j^{th} joint
K_j	Weight of the term proportional to the jerk	$\ddot{\theta}_j^{max}$	Acceleration constraint for the j^{th} joint
$\theta_j(t)$	Displacement of the j^{th} joint	$\dddot{\theta}_j^{max}$	Jerk constraint for the j^{th} joint

Symbol Meaning Symbol Meaning

N Number of robot joints $\theta\dot{j}\ (t)$ Velocity of the jth joint

n Number of segments $\theta\ddot{j}\ (t)$ Acceleration of the jth joint

hi Time interval between the ith via-point and (i + 1) th via-point $\theta\dddot{}j(t)$ Jerk of the jth joint

tf Total traveling time of the trajectory θmax

j Displacement constraint for the jth joint

Kt Weight of the term proportional to the traveling time $\dot{\theta} max$

j Velocity constraint for the jth joint

Kj Weight of the term proportional to the jerk $\ddot{\theta} max$

j Acceleration constraint for the jth joint

$\theta(t)$ Displacement of the jth joint $\theta\dddot{}max$

j Jerk constraint for the jth joint

The trajectory planning of the robot manipulator is usually based on the movement of the end effector in Cartesian space. Accordingly, from the points specified in Cartesian space, the displacements in Joint space can be calculated by using inverse kinematics. Consequently, the joint displacement calculated by the inverse kinematics constitutes the trajectory of the motion between given points within the determined time. For obtaining smooth and continuous movements between these via points in the Joint space of the robot manipulator, the appropriate interpolation splines are used. For this study, the spline functions widely used in trajectory planning are outlined in the following sections.

2.1. Analysis

2.1.1. Cubic splines

For trajectory interpolation, the cubic spline is frequently used in trajectory planning methods due to its easy and fast mathematical calculation. Also, it can produce continuous velocity and acceleration at every via-point [22]. In Cartesian space, intermediate points are kinematically defined between start and endpoints. Utilizing inverse kinematics, these nodes defined in Cartesian space are converted into Joint space. In Figure 1, the motion trajectory consisting of multiple segments is planned with the cubic spline for each joint. The trajectory in Joint space with n cubic segments has n+1 prespecified joint angles [23]. In Figure 1, the generic component θi represents the angle value of the jth joint at the ith via-point and hi denotes the length of time interval ti+1 − ti.

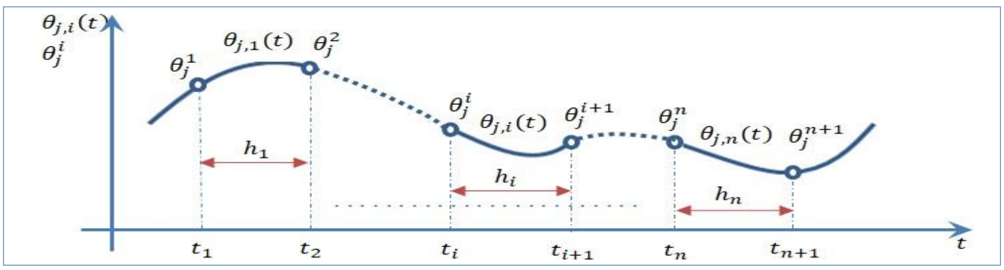


Fig. 1. Cubic spline interpolation of the via-points in the joint trajectory.

The cubic spline piecewise polynomials can be given as follows:

$\theta j, i(t) = aj, i\,(t - ti)3 + bj, i\,(t - ti)2 + cj, i\,(t - ti) + dj, i \quad j = 1,2, \ldots, N\ and\ i = 1,2, \ldots, n + 1$ (3)

where the ith cubic polynomial function for the jth joint is shown by $\theta j,\ i(t)$ with the time interval hi. The coefficients of the cubic polynomial function are a, bi, ci and di. By taking the first and the second derivatives of the cubic polynomial function, the velocity and acceleration of the joint jth at each via-point can be obtained.

2.1.2. Higher-order polynomial splines

In trajectory planning, it is possible to generate a smooth path and improve the movement stability by interpolating the trajectory utilizing a reasonable order polynomial function. The smooth trajectory can be achieved by obtaining the continuity of acceleration and jerk. At the beginning and ending points, the cubic polynomial curve provides continuous displacement, velocity, and acceleration; but this curve does not support the continuity of the jerk. Hence, this case may cause vibrations, especially at the initial and the rest points. Accordingly, the accurate trajectory may be reduced by this discontinuous jerk. At the beginning and end of the trajectory, zero velocity, acceleration, and jerk are desired especially by the robot designers for obtaining a faster and more accurate trajectory. It is possible to specify zero jerk at the start and end points of the trajectory by using 5th, 7th, and 9th order polynomials. As examined in the study in [14], among these polynomials, minimum jerk trajectory in terms of linear and angular motion is provided with the 7th order polynomial.

The 7th order polynomial used to describe the joint displacement profile can be written as

$(t) = a7(t - t0)7 + a6(t - t0)6 + a5(t - t0)5 + a4(t - t0)4 + a3(t - t0)3 + a2(t - t0)2 + a1(t - t0)$

$+ a0$ (4)

The profiles of velocity, acceleration, and jerk of the robot joints are obtained utilizing the first, second, and third derivative of the above equation, respectively with eight limits specified as $\theta(t0) = \theta 0$, $\theta(tf) = \theta f$, $\theta(t0) = \theta 0$,

$\theta(tf) = \theta f$, $\theta(t0) = \theta 0$, $\theta(tf) = \theta f$, $\dddot{\theta}(t0) = \dddot{\theta} 0$, $\dddot{\theta}(tf) = \dddot{\theta} f$. The initial time, position, velocity, acceleration, and jerk are denoted by $t0$, $\theta 0$, $\theta 0$, $\theta 0$, and $\dddot{\theta} 0$ while tf, θf, θf, θf and $\dddot{\theta} f$ illustrate the final time, position, velocity, acceleration, and jerk, respectively. By combining these profiles of the joints with the eight constraints, eight equations can be yielded with eight unknown coefficients.

2.1.3. Cuckoo Search Algorithm

Yang and Deb [16] improved a new population-based metaheuristic optimization technique known as the cuckoo search algorithm which is inspired by the brood parasitic breeding strategy of certain species of cuckoos by laying their eggs in the nests of other host birds.

In the CS optimization algorithm, each nest is considered a suitable solution candidate, and the optimum nest is selected as the best nest. There are three important rules of this newly developed algorithm. These can be stated as follows [20].

- Each cuckoo randomly selects one of the nests in its environment and lays one egg there.
- The best nests with high-quality eggs will be carried over to the next generation.

- The number of host nests in the selected environment is fixed and the eggs released by the cuckoo may be recognized by the host, with probability (pa) between 0 and 1. In such a case, the host bird may throw the foreign eggs out of the nest or leave the existing nest to establish a new nest.

The cuckoo finds the new nest with a general random walk using the L vy flight law. The Lévy flight process is basically a random walk that is derived from the L vy distribution with infinite variance and infinite mean (Yang, & Deb, 2010). According to the Lévy flight, let xn−1 be the current solution for kth cuckoo, then new solution xn is presented as given

xn = xn−1+a⊗ $lev(\lambda)$, k = 1,2, …, n

{ k k

$lev(\lambda) = t^{-\lambda}, (1 < \lambda \leq 3)$ (5)

where $\alpha > 0$ is the step size for the scale of the problem, t presents the current iteration. The Lévy flight, one of the most outstanding features of cuckoo search, produces a new candidate solution by a random walk. The Lévy flight represented by $lev(\lambda)$ is the main parameter of the CS algorithm and is used for both local search and global search.

Summarizing the CS algorithm, the initial population is generated using random numbers. The error involved in each population is then specified. Some quality solutions obtained at each iteration are stored for further processing. Then, by using Lévy's flight, the rest is updated.

2.2. Procedure

In this section, optimal trajectory planning for a 6-DOF industrial robot is performed by using the aforementioned trajectory planning method through the implementation of the CS optimization algorithm. The optimization algorithm and the planning method have been simulated in MATLAB R2018b executing on an Intel Core i5 laptop with 8250U at 3.40 GHz.

The path to be followed by the end-effector of the robot in Cartesian Space is shown in Figure 2. The flow chart of the optimal trajectory planning is shown in Figure 3. In this chart, the optimal trajectory planning consists of three basic sections: the input section for robots where initial and final configurations are obtained using inverse kinematics, the optimization section where the optimization algorithm and spline method are applied using the objective function, and the last section where the position, velocity, acceleration and jerk profiles are calculated for each joint.

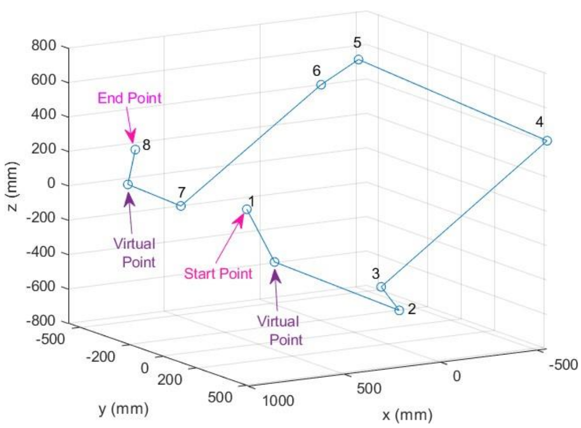


Fig 2. Path to be followed by end-effector of the robot in Cartesian Space.

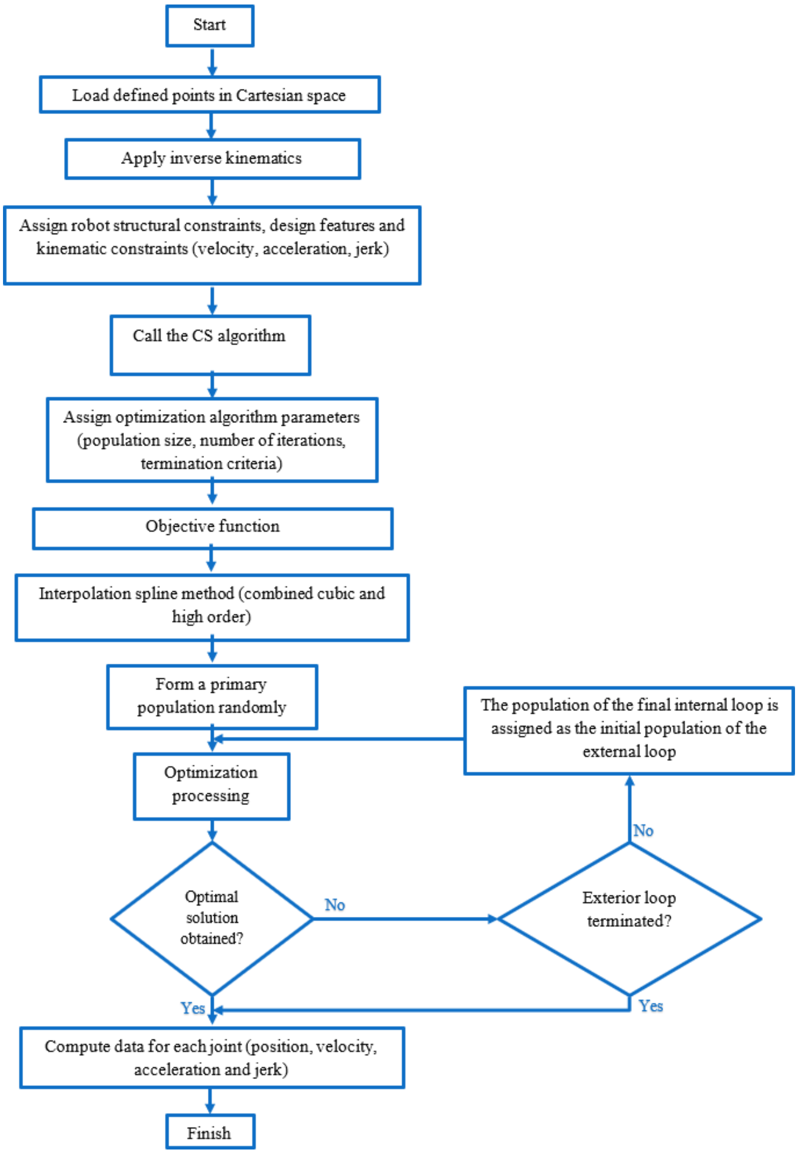


Fig. 3. Flowchart of the optimal trajectory planning for a robotic arm.

TABLE II
The joint via-points for trajectory planning.

Via-Points	Joints (°)					
	1	2	3	4	5	6
1	10	15	45	5	10	6
Virtual						
2	60	25	180	20	30	40
3	75	30	200	60	-40	80
4	130	-45	120	110	-60	70
5	110	-55	15	20	10	-10
6	100	-70	-10	60	50	10
7	-10	-10	100	-100	-40	30
Virtual						
8	-50	10	50	-30	10	20

TABLE III
Kinematical constraints of the robot joints

Joints No.	**1**	**2**	**3**	**4**	**5**	**6**
Velocity ($°/s$)	100	95	100	150	130	110
Acceleration ($°/s^2$)	60	60	75	70	90	80
Jerk ($°/s^3$)	60	66	85	70	75	70

Consequently, according to the selection of interpolation spline technique and objective function, the profiles of the position, velocity, acceleration, and jerk for each joint are obtained with the chosen optimization algorithm between the initial and intermediate positions, and also intermediate and final positions defined in the robot trajectory.

3. Results

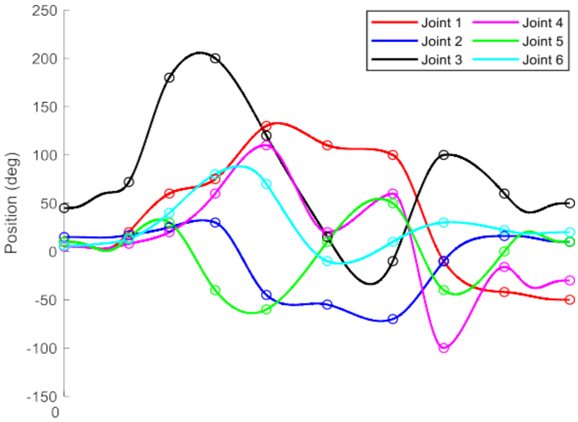
The parameters of the optimization algorithm are selected as follows: the number of nests is chosen as 20. The number of iterations is selected as 150. The Discovery rate of foreign eggs is set to 0.25.

TABLE IV
Time intervals of time-optimal trajectory planning approach.

Time interval (The interpolation methods)	**Interval no.**								
	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9
Cubic spline with 7th order polynomial (s)	4.9452	3.0933	3.4937	3.9156	4.6478	4.9945	3.8972	4.3261	4.9980

To verify the effectiveness of the proposed approach, the simulation tests are implemented according to the time-optimal using the aforementioned spline method. The lower and upper bounds of time interval (hi) are set to 3 s and 5 s, respectively. After optimization with CS based on the time-optimal by using the interpolation methods, the displacements, velocities, accelerations, and jerks of each joint are obtained as in Figure 4. Table 4 gives the values of the consecutive time intervals (hi) optimized with the proposed algorithm for time-optimal trajectory planning based on the used interpolation method.

As can be observed from Figure 4, a motion that has the velocity, acceleration, and jerk of zero at the start and end points of the trajectory is obtained based on the combination of cubic and 7th order polynomial. On the other hand, jerk continuous profiles are realized only beginning and end of the trajectory.



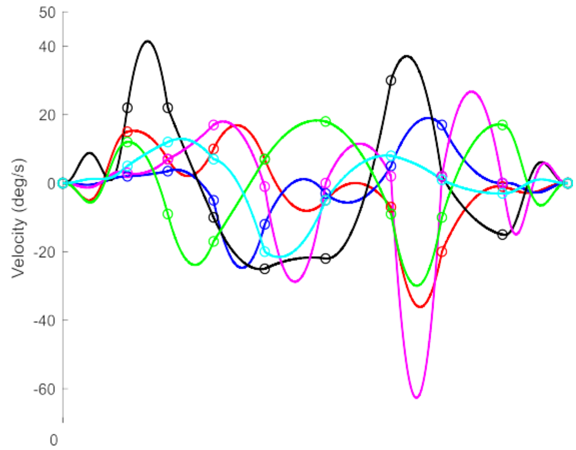
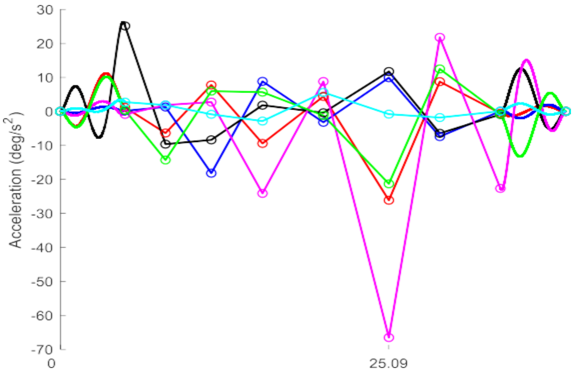
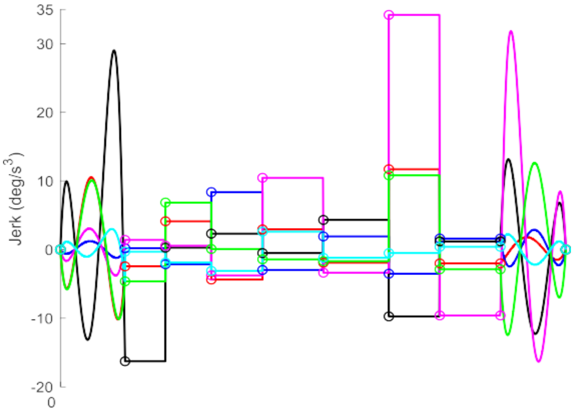


Figure 4. Resulting in joint positions (a), velocities (b), accelerations (c), and jerks (d) based on the combination of cubic spline and 7th order polynomial for time-optimal trajectory planning.





(c) (d)

Fig. 4. (continued).

TABLE V

Maximum joint kinematic values obtained from optimization algorithms for time-optimal trajectory planning.

Algorithm	Method		Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Average
Proposed	Cubic + 7th order polynomial	V_{max}	16.92	18.99	41.43	26.72	18.30	12.93	**22.55**
		A_{max}	11.20	9.91	26.29	21.78	12.49	5.60	**14.54**
		J_{max}	11.72	8.38	29.04	34.20	12.65	3.01	**16.50**
Kucuk (2018)	Cubic + 7th order polynomial	V_{max}	28.36	39.49	54.53	30.03	39.22	22.13	35.62
		A_{max}	20.29	32.27	29.95	27.78	36.18	16.44	27.15
		J_{max}	17.22	24.15	29.05	24.81	29.04	11.06	22.55
Gasparetto and Zanotto (2008)	Cubic	V_{max}	25.89	20.59	45.29	45.84	29.05	22.57	31.54
		A_{max}	21.34	15.23	42.75	32.46	24.29	35.54	28.60
		J_{max}	14.22	30.16	37.50	43.56	46.03	21.52	32.17
Simon and Isik (1991)	Trigonometric	V_{max}	23.93	22.88	45.80	37.33	30.38	19.70	30.00
		A_{max}	21.34	15.23	42.75	32.46	24.29	35.54	28.60
		J_{max}	32.71	20.79	57.46	65.15	28.94	56.95	43.67

TABLE VI

Mean kinematic values obtained from optimization algorithms for time-optimal trajectory planning

Algorithm	Method		Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Average
Proposed	Cubic + 7th order polynomial	V_{mean}	8.32	5.70	15.65	13.28	11.16	6.20	10.05
		A_{mean}	4.22	2.99	6.89	8.83	5.88	2.42	5.20
		J_{mean}	3.95	2.51	5.75	8.76	4.53	1.40	4.48
Kucuk (2018)	Cubic + 7th order polynomial	V_{mean}	10.27	8.14	17.59	16.05	12.77	7.87	12.11
		A_{mean}	4.32	4.88	8.63	8.78	7.16	3.80	6.26
		J_{mean}	4.94	4.94	6.79	7.10	6.63	2.29	5.49
Gasparetto and Zanotto (2008)	Cubic	V_{mean}	13.03	9.05	23.45	21.25	16.11	9.65	15.43
		A_{mean}	7.25	6.85	12.77	10.00	12.33	6.89	11.02

		J_{mean}	8.33	8.95	14.67	28.76	16.70	9.01	14.40
Simon and Isik (1991)	Trigonometric	V_{mean}	9.84	5.64	13.42	11.89	9.21	6.12	9.35
		A_{mean}	6.29	4.88	9.94	9.57	6.61	6.16	7.24
		J_{mean}	8.61	6.24	16.01	15.38	8.39	7.58	10.37

4. Discussion

The results obtained from the proposed CS algorithm are compared to the proposed optimization techniques in [13] [23,24] based on the used interpolation methods and the used objective functions. Also, the comparison results are comprehensively given in the tables in terms of the maximum kinematic values and their average values.

From Table 5 it is noticeable that the maximum kinematic values of each joint based on the proposed algorithm are almost much lower as compared with those yielded by the algorithms [13] [23,24]. Also, the average values of the maximum velocities, accelerations, and jerks of all joints are obtained lower compared to those obtained from previous studies. As a result, in the case of using the proposed algorithm and the combination of cubic spline and 7th order polynomial, better results are provided as compared with that of Gasparetto, & Zanotto, [13], Kucuk, [23] Simon, & Isik, [24].

From Table 6, the mean joint velocities, accelerations, and jerks yielded by the proposed algorithm as well as their average values are much lower compared with those of Gasparetto, & Zanotto, [13], Kucuk, [23] Simon, & Isik, [24]. based on each interpolation method. CS can perform better compared with those algorithms proposed by Gasparetto, & Zanotto, [13], Kucuk, [23] Simon, & Isik, [24] in terms of time optimization.

In this paper, the commonly used interpolation method for the trajectory generation of industrial robots hasbeen adopted to obtain a time-optimal trajectory. All method has been tested on PUMA robot manipulator by using both the proposed CS algorithm and the algorithms suggested by Gasparetto, & Zanotto, [13], Kucuk, [23] Simon, & Isik, [24]. with the aforementioned objective function. Also, the results from the proposed algorithm have been compared with those by executing the algorithms provided in Gasparetto, & Zanotto, [13], Kucuk, [23] Simon, & Isik, [24].for each joint.

5. Conclusion

The conclusions can be summarized as follows:

(1) As can be observed from all results, the kinematic values of each joint generated by the interpolation method using the proposed algorithm meet the kinematic limits of the joint velocity, acceleration, and jerk. This is crucial for dynamic trajectory planning.

(2) At the start and end points, smooth motion trajectories with the zero jerk for each joint can be obtained in case of using the interpolation method such as the combination of cubic spline and 7th order polynomial. This smooth motion can introduce small errors while the robot is tracking the trajectory. Accordingly, it is useful to improve control for accurate positioning.

(3) CS optimization algorithm outperforms the algorithms proposed by previous works in terms of obtainingthe mean kinematic values for both times optimal and optimal trajectory planning.

(4) All of the average values of the maximum kinematic variables for each joint based on the aforementioned methods have been obtained lower as compared with that of previous works in case of time optimal.

(5) Having considered the used before interpolation methods, the combination of the cubic spline and 7th order polynomial provides a continuous and smooth path at the initial and final via points.

(6) The simulation results have shown that the used trajectory planning methods with the proposed CSalgorithm are very effective and suitable for optimal time trajectory planning.

References

[1] Rubio, F., Llopis-Albert, C., Valero, F., & Suñer, J. L. (2016). Industrial robot efficient trajectory generation without collision through the evolution of the optimal trajectory. Robotics and Autonomous Systems, 86, 106-112. doi: 10.1016/j.robot.2016.09.008.

[2] Bobrow, J. E. (1988). Optimal robot plant planning using the minimum-time criterion. *IEEE Journal on Robotics and Automation*, 4(4), 443-450. doi:10.1109/56.811.

[3] Bobrow, J. E., Dubowsky, S., & Gibson, J. S. (1985). Time-Optimal Control of Robotic Manipulators Along Specified Paths. *The International Journal of Robotics Research*, 4(3), 3-17. doi:10.1177/027836498500400301.

[4] Shin, K., & McKay, N. (1985). Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, 30(6), 531-541. doi:10.1109 / TAC.1985.1104009.

[5] Chen, C.S., & Chen, S.K. (2019). Synchronization of tool tip trajectory and attitude based on the surface characteristics of workpiece for 6-DOF robot manipulator. Robotics and Computer-Integrated Manufacturing, 59, 13-27. doi: 10.1016/j.rcim.2019.01.016.

[6] Lin, C.S., Chang, P.R., & Luh, J.Y.S. (1983). Formulation and optimization of cubic polynomial joint trajectories for industrial robots. IEEE Transactions on Automatic Control, 28(12), 1066-1073. doi: 10.1109/TAC.1983.1103181.

[7] Aribowo, W., & Terashima, K. (2014). Cubic Spline Trajectory Planning and Vibration Suppression of Semiconductor Wafer Transfer Robot Arm. International Journal of Automation Technology,8(2), 265-274. doi: 10.20965/ijat. 2014.p0265.

[8] Gasparetto, A., & Zanotto, V. (2008). A technique for time-jerk optimal planning of robot trajectories. Robotics and Computer- Integrated Manufacturing, 24(3), 415-426. doi: 10.1016/j.rcim.2007.04.001.

[9] Gasparetto, A, & Zanotto, V. (2010). Optimal trajectory planning for industrial robots. Advances in Engineering Software, 41(4), 548-556. doi: 10.1016/j.advengsoft.2009.11.001.

[10] Liu, H., Lai, X., & Wu, W. (2013). Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints. Robotics and Computer-Integrated Manufacturing, 29(2), 309-317. doi: 10.1016/j.rcim.2012.08.002.

[11] Perumaal, S.S., & Jawahar, N. (2013). Automated Trajectory Planner of Industrial Robot for Pick-and-Place Task. International Journal of Advanced Robotic Systems, 10(2), 1-17. doi:10.5772/53940.

[12] Liu, L., Chen, C., Zhao, X., & Li, Y. (2016). Smooth trajectory planning for a parallel manipulator with joint friction and jerk constraints. International Journal of Control Automation System, 14, 1022-1036. doi:10.1007/s12555-014-0495-4.

[13] Gasparetto, A, & Zanotto, V. (2007). A new method for smooth trajectory planning of robot manipulators. Mechanism and Machine Theory, 42(4), 455-471. doi: 10.1016/j.mechmachtheory.2006.04.002.

[14] Boryga, M., & Graboś, A. (2009). Planning of manipulator motion trajectory with higher-degree polynomials use. Mechanism and Machine Theory, 44(7), 1400-1419. doi: 10.1016/j.mechmachtheory.2008.11.003.

[15] Machmudah, A., Parman, S., Zainuddin, A., & Chacko, S. (2013). Polynomial joint angle arm robot motion planning in complex geometrical obstacles. Applied Soft Computing, 13(2), 1099-1109.doi: 10.1016/j.asoc.2012.09.025.

[16] Yang, X. S., & Deb, S. (2009). Cuckoo search via Levy flights. In 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC) (pp. 210–214). IEEE. doi:10.1109/NABIC.2009.5393690.

[17] Kucuk, S. (2018). Optimal trajectory generation algorithm for serial and parallel manipulators. Robotics and Computer- Integrated Manufacturing, 48, 219-232. doi: 10.1016/j.rcim.2017.04.006.

[18] Savsani, P., Jhala, R. L., & Savsani, V. J. (2013). Optimized trajectory planning of a robotic arm using teaching-learning-based optimization (TLBO) and artificial bee colony (ABC) optimization techniques. 2013 IEEE International Systems Conference (SysCon). doi:10.1109/syscon.2013.6549910

[19] Savsani, P., Jhala, R. L., & Savsani, V. J. (2016). Comparative Study of Different Metaheuristics for the Trajectory Planning of a Robotic Arm. IEEE Systems Journal, 10(2), 697–708. doi:10.1109/jsyst.2014.2342292.

[20] Yang, X. S., & Deb, S. (2010). Engineering optimisation by cuckoo search. *International Journal of Mathematical Modellingand Numerical Optimization*, 1(4), 330–343. arXiv:1005.2908.

[21] Ficarella, E., Lamberti, L., & Degertekin, S. O. (2019). Mechanical identification of materials and structures with optical methods and metaheuristic optimization. *Materials*, *12*(13), 2133.

[22] Kolter, J.Z., & Andrew, Y.N. (2009). Task-space trajectories via cubic spline optimization. In IEEE international conference on robotics and automation (pp. 1675–82). IEEE. doi:10.1109/ROBOT.2009.5152554.

[23] Kucuk, S. (2016). Maximal dexterous trajectory generation and cubic spline optimization for fully planar parallel manipulators. Computer and Electrical Engineering,56, 634-647. doi: 10.1016/j.compeleceng.2016.07.012.

[24] Simon, D., & Isik, C. (1991). Optimal trigonometric robot trajectories. Robotica, 9(4), 379–386. doi:10.1017/S0263574700000552