Time-Optimal Trajectory Optimization of Serial Robotic Manipulator With Kinematic and Dynamic Limits Based On Improved Particle Swarm Optimization

Yu Yang (✉ 416271907@qq.com)

Beijing Jiaotong University <https://orcid.org/0000-0003-2481-0502>

Hongze Xu

Beijing Jiaotong University

Shaohua Li

Beijing Jiaotong University

Lingling Zhang

Beijing Jiaotong University

Xiuming Yao

Beijing Jiaotong University

Research Article

Keywords: 4-3-4 polynomial interpolation function, particle swarm optimization with cosine decreasing weight, time-optimal trajectory planning, kinematic and dynamic limits

Posted Date: July 22nd, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-710806/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at The International Journal of Advanced Manufacturing Technology on February 11th, 2022. See the published version at <https://doi.org/10.1007/s00170-022-08796-y>.

Time-optimal trajectory optimization of serial robotic manipulator with kinematic and dynamic limits based on improved particle swarm optimization

Yu Yang · Hong-ze Xu · Shao-hua Li · Ling-ling Zhang · Xiu-ming Yao

Received: date / Accepted: date

Abstract Effective motion control can achieve accurate and fast positioning and movement of industrial robotics and improve industrial productivity significantly. Time-optimal trajectory optimization (TO) is a great concern in the motion control of robotics and can improve motion efficiency by providing high-speed and reasonable motion references to the motion controller. In this study, a new time-optimal TO strategy, polynomial interpolation function (PIF) combined with improved particle swarm optimization (PSO) considering kinematic and dynamic limits, successfully optimizes the movement time of the PUMA 560 serial manipulator along a randomly assigned path. The 4-3-4 PIF is first used to generate the smooth and 3-order continuous movement trajectories of six joints in joint space. The PSO with cosine decreasing weight (CDW-PSO) algorithm further reduces the trajectories movement time considering the limits of the angular displacement, angular velocity, angular acceleration, angular jerk, and joint torque. Experimental results show that the CDW-PSO algorithm achieves a better convergence rate of 23 and a better fitness value of 2.46 compared with the PSO with constant weight and linearly decreasing weight algorithms. The CDW-PSO optimized movement time is reduced by 83.6% compared to the manually setting movement time value of 15. The proposed time-optimal TO strategy can be conducted easily and directly search for global optimal solutions without approximation of the limits. The opti-

mized trajectories could be incorporated in the motion controller of the actual manipulators due to considering the kinematic and dynamic limits.

Keywords 4-3-4 polynomial interpolation function · particle swarm optimization with cosine decreasing weight · time-optimal trajectory planning · kinematic and dynamic limits

1 Introduction

The operational stock of industrial robots in factories was in ascending trend over the last few years and achieved 2.7 million in 2019 around the world, an 12% increase compared to 2018 [1]. Robotic manipulators are the most popular industrial robots and used widely to perform repetitive tasks with high efficiency and precision in tedious, dirty, and dangerous industrial environments. Efficient motion control of the manipulators consists of trajectory planning and tracking, which can improve the execution efficiency and stability of the manipulators greatly. Specifically, trajectory planning can shorten movement time, improve tracking accuracy, and reduce wear of the manipulators by providing high-speed and smooth motion references to the motion controller. The trajectory planning consists of path generation (PG) and trajectory optimization (TO) and could be conducted offline generally. The PG can generate a raw path through or avoid some assigned position points in operating spaces for specified task requirements. The TO then convert the raw path to an optimized trajectory, which can reduce movement time and avoid unnecessary jitter and impact of the manipulators [2]. The conventional TO algorithms only have been used to generate a smooth and continuous movement trajectory, such as polynomial interpolating function (PIF) [3, 4], B-spline curve [5, 6], NURBS curve [7, 8], Bezier curve [9, 10]. However, the execution efficiency of the manipulators is not considered in these

This work was supported in part by the National Natural Science Foundation of China under Grant 61773052.

Yu Yang
School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, 100044, China
E-mail: 416271907@qq.com

Hong-ze Xu · Shao-hua Li · Ling-ling Zhang · Xiu-ming Yao
School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, 100044, China

algorithms. Time-optimal TO algorithms can be used to find the fastest movement trajectory while respecting kinematic or dynamic limits. The dynamic programming (DP) [11, 12], quadratic programming (QP) [13, 14], and convex optimization (CO) algorithms [2, 15] have been researched to optimize the movement time. However, the DP and QP algorithms might consume great memory storage and long computation time to find the optimal solution. Approximate conversions of the jerk limits are inevitable in the CO algorithms, otherwise, the convexity of the CO algorithms is not satisfied. The heuristic search techniques, such as the genetic [16, 17], cuckoo search [18], and particle swarm optimization (PSO) algorithms [5, 19], have been researched to directly search for optimal solutions without approximation of the jerk limits by effective adjustment of the search strategies. The PSO algorithm has advantages of fewer parameters, simpler structure, and easier realization compared with other algorithms. However, the slow convergence rate and trapping into the local optimum could occur in the basic PSO algorithm. In addition, many researchers do not consider all kinematic and dynamic limits using the PSO algorithm. The trajectories optimized by these algorithms could not be applied in the industry. Therefore, an improved PSO algorithm with the limits of the angular displacement, angular velocity, angular acceleration, angular jerk, and joint torque is proposed to optimize the movement time of trajectories. The performance of the algorithm is evaluated in this study. The finding of this study will build a time-optimal TO strategy by which the generated trajectories could be incorporated in the motion controller of the actual manipulator.

2 Kinematics and dynamics analysis of the serial manipulator

The PUMA 560 serial manipulator with six degrees of freedom is used to study (Fig. 1). The geometric and inertial parameters of the PUMA 560 are as shown in Table 1 [20]. The simulation model of the PUMA 560 based on the above parameters is built in the MATLAB platform using the Robotics Toolbox. The reachable workspace of the manipulator is then calculated based on the Monte Carlo method [21], which is used as the feasible region of the trajectory planning, as shown in Fig. 2.

According to the above kinematic parameters(θ_i , d_i , a_{i-1} , α_{i-1}), the rotation angles of the joints can be calculated using the Kinematics Toolbox in MATLAB when given a fixed position in the Cartesian space [4]. The dynamic equations of the manipulator based on the Lagrangian function are shown as follows.

$$\tau_i = \sum_{j=1}^n \mathbf{D}_{ij} \ddot{\mathbf{q}}_j^{(2)} + \sum_{j=1}^n \sum_{k=1}^n \mathbf{D}_{ijk} \dot{\mathbf{q}}_j^{(1)} \dot{\mathbf{q}}_k^{(1)} + \mathbf{D}_i$$

Fig. 1 PUMA 560 serial manipulator

Table 1 Geometric and inertial parameters of PUMA 560 [20]

Link N	1	2	3	4	5	6
$\theta_i(\text{rad})$	Q1	Q2	Q3	Q4	Q5	Q6
$d_i(\text{m})$	0	0	0.15	0.4318	0	0
$a_{i-1}(\text{m})$	0	0.4318	0.0203	0	0	0
$\alpha_{i-1}(\text{°})$	$\pi/2$	0	$-\pi/2$	$\pi/2$	$-\pi/2$	0
$m(\text{kg})$	0	17.4	4.8	0.82	0.34	0.09
$p_x(\text{m})$	0	-0.3638	-0.0203	0	0	0
$p_y(\text{m})$	0	0.006	-0.0141	0.19	0	0
$p_z(\text{m})$	0	0.2275	-0.0700	0	0	0.032
$I_{xx}(\text{kg} \cdot \text{m}^2)$	0	0.13	0.066	1.8e-3	0.3e-3	0.15e-3
$I_{yy}(\text{kg} \cdot \text{m}^2)$	0.35	0.524	0.086	1.3e-3	0.4e-3	0.15e-3
$I_{zz}(\text{kg} \cdot \text{m}^2)$	0	0.539	0.0125	1.8e-3	0.3e-3	0.04e-3

Fig. 2 Simulation model of PUMA 560 in MATLAB

$$\begin{aligned} \mathbf{D}_{ij} &= \sum_{p=\max(i,j,k)}^n \text{Tr}(\mathbf{U}_{pj} \mathbf{J}_p \mathbf{U}_{pi}^T) \\ \mathbf{D}_{ijk} &= \sum_{p=\max(i,j,k)}^n \text{Tr}(\mathbf{U}_{pjk} \mathbf{J}_p \mathbf{U}_{pi}^T) \\ \mathbf{D}_i &= \sum_{p=i}^n -m_p \mathbf{g}^T \mathbf{U}_{pi} \bar{\mathbf{r}}_p \\ \mathbf{U}_{ij} &= \frac{\partial^0 \mathbf{T}_i}{\partial \mathbf{q}_j} = \frac{\partial(\mathbf{A}_1 \mathbf{A}_2 \dots \mathbf{A}_j \dots \mathbf{A}_i)}{\partial \mathbf{q}_j} \end{aligned}$$

Table 2 Limits of PUMA 560 [20]

Joint N	1	2	3	4	5	6
$\theta_{\max}(\text{rad})$	π	$3\pi/4$	$3\pi/4$	π	$3\pi/4$	$3\pi/4$
$v_{\max}(\text{rad/s})$	8	10	10	5	5	5
$a_{\max}(\text{rad/s}^2)$	10	12	12	8	8	8
$J_{\max}(\text{rad/s}^3)$	30	40	40	20	20	20
$\tau_{\max}(\text{N}\cdot\text{m})$	97.6	186.4	89.4	24.2	20.1	21.3

$$\mathbf{U}_{ijk} = \frac{\partial \mathbf{U}_{ij}}{\partial \mathbf{q}_k}$$

$$\mathbf{J}_p = \int \bar{\mathbf{r}}_p \bar{\mathbf{r}}_p^T dm_p$$

$$\bar{\mathbf{r}}_p^T = [P_{p,x} \ P_{p,y} \ P_{p,z} \ 1]$$

Where, $\mathbf{q}_i^{(1)}$, $\mathbf{q}_i^{(2)}$, and τ_i are the angular velocity, angular acceleration, and joint torque of the i_{th} joint, respectively; $\mathbf{D}_{ij}$, $\mathbf{D}_{ijk}$, and $\mathbf{D}_i$ are the inertial system matrix, Coriolis and centripetal forces matrix, and gravity loading, respectively; $\mathbf{A}_i$ is the transformation matrix of the i_{th} link coordinate system to the $(i-1)_{th}$ link coordinate system; $\mathbf{J}_p$ is the inertia matrix consists of the moments of inertia and products of inertia; m_p is the mass of the p_{th} link; $\bar{\mathbf{r}}_p$ is the position vector of the center of m_p in the p_{th} link coordinate system; $\mathbf{g}^T$ is the acceleration vector due to gravity in the base coordinate system. The viscous damping and Coulomb friction are not considered for simplicity [20, 22].

The limits of the angular velocity, angular acceleration, angular jerk, and joint torque of the PUMA 560 are as shown in Table 2 [20].

3 Time-optimal TO with the kinematic and dynamic limits

3.1 4-3-4 PIF used in joint space

In this study, four positions are assigned randomly in the reachable region of the manipulator in the Cartesian coordinate system. Therefore, the end effector of the manipulator has to pass by above the positions. The 4-3-4 PIF is first used to generate the smooth and 3-order continuous movement trajectories of six joints in joint space. Four rotation angles of each joint are set as interpolation points in joint space. Three sections then are divided from the starting to the end interpolation point of each joint. The quartic, cubic and quartic polynomial functions are built on three sections, respectively, as shown in formula (1).

$$\begin{cases} \theta_{j1}(t) = a_{j14}t^4 + a_{j13}t^3 + a_{j12}t^2 + a_{j11}t + a_{j10}, \\ \quad \forall t \in [0, t_{j1}] \\ \theta_{j2}(t) = a_{j23}t^3 + a_{j22}t^2 + a_{j21}t + a_{j20}, \\ \quad \forall t \in [0, t_{j2}] \\ \theta_{j3}(t) = a_{j34}t^4 + a_{j33}t^3 + a_{j32}t^2 + a_{j31}t + a_{j30}, \\ \quad \forall t \in [0, t_{j3}] \end{cases} \quad (1)$$

Where, a_{jik} is the k_{th} coefficient of the i_{th} polynomial function of the j_{th} joint; $[0, t_{ji}]$ is the time domain of the i_{th} polynomial function of the j_{th} joint. To reduce the mechanical friction, the movement time in each section of each joint needs to be the same, namely $t_{ji} = t_i$. $\theta_{ji}(t)$ is the angular displacement of the i_{th} polynomial function of the j_{th} joint at time t .

According to the angular displacement of four interpolation points, zero angular velocity and angular acceleration of the starting and the end interpolation points, and continuity of the angular velocity, angular acceleration of intermediate interpolation points, the coefficients in the formula (1) can be obtained by γ , $\mathbf{B}$, and $\mathbf{A}$ as follows.

$$\begin{aligned} \mathbf{A}_1 &= [a_{14} \ a_{13} \ a_{12} \ a_{11} \ a_{10}] \\ \mathbf{A}_2 &= [a_{23} \ a_{22} \ a_{21} \ a_{20}] \\ \mathbf{A}_3 &= [a_{34} \ a_{33} \ a_{32} \ a_{31} \ a_{30}] \\ \mathbf{A}^T &= \gamma^{-1} \mathbf{B} = [\mathbf{A}_1 \ \mathbf{A}_2 \ \mathbf{A}_3] \end{aligned} \quad (2)$$

$$\gamma = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \mathbf{0}_{3 \times 5} \\ \mathbf{0}_{3 \times 5} & \gamma_{22} & \gamma_{23} \\ \mathbf{0}_{3 \times 5} & \mathbf{0}_{3 \times 4} & \gamma_{33} \\ \gamma_{41} & \mathbf{0}_{3 \times 4} & \mathbf{0}_{3 \times 5} \\ \mathbf{0}_{2 \times 5} & \gamma_{52} & \gamma_{53} \end{bmatrix}$$

$$\gamma_{11} = \begin{bmatrix} t_{j1}^4 & t_{j1}^3 & t_{j1}^2 & t_{j1} & 1 \\ 4t_{j1}^3 & 3t_{j1}^2 & 2t_{j1} & 1 & 0 \\ 12t_{j1}^2 & 6t_{j1} & 2 & 0 & 0 \end{bmatrix}$$

$$\gamma_{12} = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & -2 & 0 & 0 \end{bmatrix}$$

$$\gamma_{22} = \begin{bmatrix} t_{j2}^3 & t_{j2}^2 & t_{j2} & 1 \\ 3t_{j2}^2 & 2t_{j2} & 1 & 0 \\ 6t_{j2} & 2 & 0 & 0 \end{bmatrix}$$

$$\gamma_{23} = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -2 & 0 & 0 \end{bmatrix}$$

$$\gamma_{33} = \begin{bmatrix} t_{j3}^4 & t_{j3}^3 & t_{j3}^2 & t_{j3} & 1 \\ 4t_{j3}^3 & 3t_{j3}^2 & 2t_{j3} & 1 & 0 \\ 12t_{j3}^2 & 6t_{j3} & 2 & 0 & 0 \end{bmatrix}$$

$$\gamma_{41} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\gamma_{52} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\gamma_{53} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{B}^T = [\mathbf{0}_{6 \times 1} \ b_1 \ \mathbf{0}_{2 \times 1} \ b_2 \ \mathbf{0}_{2 \times 1} \ b_3 \ b_4]$$

$$b_1 = \theta_{j3}(t_{j3}), b_2 = \theta_{j1}(0)$$

$$b_3 = \theta_{j3}(0), b_4 = \theta_{j2}(0)$$

3.2 4-3-4 PIF combined with improved PSO used in joint space

In this study, an improved PSO algorithm considering kinematic and dynamic limits (Table 2) is proposed to optimize the movement time of the trajectories generated by the 4-3-4 PIF. The objective function and limits of joints are as follows:

$$\theta(t) = \min(t_1 + t_2 + t_3)$$

$$s.t. \begin{cases} \max\{|\theta_{j1}|, |\theta_{j2}|, |\theta_{j3}|\} < \theta_{j,\max} \\ \max\{|\theta_{j1}^{(1)}|, |\theta_{j2}^{(1)}|, |\theta_{j3}^{(1)}|\} < v_{j,\max} \\ \max\{|\theta_{j1}^{(2)}|, |\theta_{j2}^{(2)}|, |\theta_{j3}^{(2)}|\} < a_{j,\max} \\ \max\{|\theta_{j1}^{(3)}|, |\theta_{j2}^{(3)}|, |\theta_{j3}^{(3)}|\} < J_{j,\max} \\ \max\{|\tau_{j1}|, |\tau_{j2}|, |\tau_{j3}|\} < \tau_{j,\max} \end{cases} \quad (3)$$

Where, t_i is the time span of the i_{th} polynomial function of each joint. θ_{ji} , $\theta_{ji}^{(1)}$, $\theta_{ji}^{(2)}$, $\theta_{ji}^{(3)}$, and τ_{ji} are the angular displacement, angular velocity, angular acceleration, angular jerk, and joint torque of the j_{th} joint in the i_{th} section, respectively. $\theta_{j,\max}$, $v_{j,\max}$, $a_{j,\max}$, $J_{j,\max}$, and $\tau_{j,\max}$ are the limiting values of the j_{th} joint (Table 2).

The basic PSO algorithm is a stochastic global optimization technique and used widely in multidimensional space optimization problems [23–25]. In this study, an PSO with cosine decreasing weight (CDW-PSO) algorithm is proposed for overcoming the shortcomings of the basic PSO algorithm. The description of parameters and formulas of the CDW-PSO algorithm is as follows:

N and K are the number of iterations and particles respectively; $\mathbf{P}_i^k(t_{i1}^k, t_{i2}^k, t_{i3}^k)$ ($t_{in}^k \in [t_{\min}, t_{\max}]$) and $\mathbf{V}_i^k(v_{i1}^k, v_{i2}^k, v_{i3}^k)$ ($v_{in}^k \in [v_{\min}, v_{\max}]$) are the position and speed vectors of the i_{th} particle, respectively, at the k_{th} iteration; Γ_i^k is the fitness value of the i_{th} particle at the k_{th} iteration and calculated by $t_{i1}^k + t_{i2}^k + t_{i3}^k$; Γ_{iL}^k and Γ_G^k are the minimum fitness value of the i_{th} particle and the global minimum fitness value of all particles, respectively, during k iterations; $\mathbf{P}_{iL}^k$ and $\mathbf{P}_G^k$ are the corresponding position vectors of Γ_{iL}^k and Γ_G^k respectively; the dynamic inertia weight ω , learning factors $c1$ and $c2$ are expressed in formulas (4)–(5), respectively. In the CDW-PSO algorithm, ω descends in the form of the cosine function with the increase of the number of iterations. The descending speed of ω in the early stage of iterations is slower than in the later stage for achieving the better global searching ability early and stronger local convergence ability later. *rand* is the random value in the range of 0 to 1. The updating formulas of speed and position vectors are shown in (6) and (7) respectively.

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \cos\left(\frac{\pi k}{2K}\right) \quad (4)$$

$$c1 = c2 = \text{const} \quad (5)$$

Table 3 Joint angles of interpolation points

	Joint angle(rad)			
	A	B	C	D
Joint 1	0	0.2	1.6	2
Joint 2	-0.5	-0.2	0	0.2
Joint 3	0	0.2	0.5	0.7
Joint 4	-1	-0.5	0	0.7
Joint 5	0	-0.2	-0.7	-1
Joint 6	0	0.6	0.4	0.3

$$\mathbf{V}_i^{k+1} = \omega \mathbf{V}_i^k + c1 \text{rand}(\mathbf{P}_i^k - \mathbf{X}_i^k) + c2 \text{rand}(\mathbf{P}_g^k - \mathbf{X}_i^k) \quad (6)$$

$$\mathbf{P}_i^{k+1} = \mathbf{P}_i^k + \mathbf{V}_i^{k+1} \quad (7)$$

The procedure of the CDW-PSO algorithm is as follows:

First, N , K , $t_{\min}$, $t_{\max}$, $v_{\max}$, $v_{\min}$, ω , $\omega_{\max}$, $\omega_{\min}$, $c1$, $c2$, $\mathbf{P}_i^0$, and $\mathbf{V}_i^0$ are initialized;

Second, Γ_{iL}^0 ($i = 1, 2, \dots, N$), Γ_G^0 , $\mathbf{P}_{iL}^0$ and $\mathbf{P}_G^0$ are calculated.

Third, $\mathbf{P}_i^k$ and $\mathbf{V}_i^k$ are updated by the formulas (4)–(7).

Fourth, the coefficients in the formula (1) can be calculated according to the formula (2). Γ_i^k is set to a big large real value if the coefficients do not exist.

Fifth, the angular displacement, angular velocity, angular acceleration, angular jerk, and joint torque in trajectories of the joints can be obtained, when the coefficients exist. Γ_i^k is calculated by $t_{i1}^k + t_{i2}^k + t_{i3}^k$ if the formula (3) is satisfied, otherwise, Γ_i^k is set to a big large real value.

Sixth, Γ_{iL}^k , Γ_G^k are calculated.

Seventh, the corresponding $\mathbf{P}_{iL}^k$, and $\mathbf{P}_G^k$ are obtained and return to the third step until iteration end.

Eighth, the global minimum fitness value can be obtained when iterations end.

The flow chart of the CDW-PSO algorithm is as shown in Fig. 3.

4 Results and discussion

Four random positions (**A**, **B**, **C**, and **D**) are assigned in the reachable region (Fig. 2) in the Cartesian system and the corresponding joint angles can be calculated by using the inverse kinematic function of the Robotics Toolbox, as shown in Table 3. t_1 , t_2 , and t_3 are manually set to 5, which satisfies the formula (3) and will be further optimized by the CDW-PSO algorithm.

The basic PSO with const weight (CW-PSO) and linearly decreasing weight (LDW-PSO) algorithms are used as comparisons to illustrate the advantages of the CDW-PSO algorithm. N , K , $t_{j,\max}$, $t_{j,\min}$, $v_{j,\max}$, $v_{j,\min}$, $c1$, and $c2$ are set to 40, 100, 5, 0, 2, -2, 2, and 2, respectively, in three algorithms; ω is set to 1 in the CW-PSO algorithm; $\omega_{\max}$ and $\omega_{\min}$ are set to 0.9 and 0.4,

Fig. 3 Flow chart of CDW-PSO algorithm

respectively, in both the LDW-PSO and CDW-PSO algorithms.

Iteration processes of the above algorithms are as shown in Fig. 4 and Fig. 5. The global fitness values decrease and tend to be stable with the increase of the number of iterations. The CW-PSO, LDW-PSO, and CDW-PSO algorithms converge to 2.70, 2.50, and 2.46 at about the 57th, 37th, and 23rd iteration. The optimized movement time t_{all} based on the above algorithms is reduced by 82%, 83%, and 83.6%, respectively, compared to manually setting $t_{all} = 3t_{max} = 15$. In this study, the performance of the CDW-PSO algorithm is verified well and shown better than the CW-PSO and LDW-PSO algorithms in both convergence speed and final-global-optimal fitness value. The resulting angular displacement, angular velocity, angular acceleration, angular jerk, and joint torque of joints 1 to 6, based on the CDW-PSO algorithm, are within their limits obviously, as shown in Fig. 6.

5 Conclusion

A new time-optimal TO strategy, the 4-3-4 PIF combined with CDW-PSO algorithm considering kinematic and dynamic limits, successfully optimizes the tra-

Fig. 4 Iterations of CDW-PSO, LDW-PSO, CW-PSO algorithms

Fig. 5 Iterations of t1, t2, and t3 of CDW-PSO

jectory movement time of the PUMA 560 robotic manipulator. Compared with the CW and LDW, adopting the CDW in the PSO algorithm can achieve a better convergence rate of 23 and a better fitness value of 2.46. The optimized movement time based on the CDW-PSO algorithm is reduced by 83.6% compared to the manually setting movement time. The optimized trajectories could be incorporated in the motion controller of the actual manipulators due to considering all kinematic and dynamic limits. Further studies of multi-objective TO, such as minimum time-energy and time-jerk, should be considered for achieving the better motion performance of the manipulators.

Declarations

Funding

This work is supported in part by the National Natural Science Foundation of China under Grant 61773052.

Fig. 6 Angular displacement (a), angular velocity (b), angular acceleration (c), angular jerk (d), and joint torque (e) of joints 1 to 6

Conflicts of interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Availability of data and material

The datasets are presented in the present study.

Code availability

The algorithm is described in the present study.

Ethics approval

The article follows the guidelines of the Committee on Publication Ethics (COPE) and involves no studies on human or animal subjects.

Consent to participate

Not applicable.

Consent for publication

Not applicable.

References

1. IFR press room (2020) IFR presents world robotics report 2020. International Federation of Robotics. <https://ifr.org/ifr-press-releases/news/record-2.7-million-robots-work-in-factories-around-the-globe>. Accessed 24 Sep 2020
2. Ma JW, Gao S, Yan HT, Lv Q, Hu GQ (2021) A new approach to time-optimal trajectory planning with torque and jerk limits for robot. *ROBOT AUTON SYST* 140(3):103744–103753. <https://doi.org/10.1016/j.robot.2021.103744>
3. Macfarlane S, Croft EA (2003) Jerk-bounded manipulator trajectory planning: design for real-time applications. *IEEE T ROBOTIC AUTOM* 19(1):42–52. <https://doi.org/10.1109/TRA.2002.807548>
4. Xu W, Yuan SC, Cao YH (2014) Research on yinma palletizing robot motion feasibility based on matlab-robotics toolbox. *Appl Mech Mater* 543–547:1296–1300. <https://doi.org/10.4028/www.scientific.net/AMM.543-547.1296>
5. Guo TY, Li F, Huang K, Zhang FZ, Feng Q (2013) Application of optimal algorithm on trajectory planning of mechanical arm based on b-spline curve. *Appl Mech Mater* 376:253–256. <https://doi.org/10.4028/www.scientific.net/AMM.376.253>
6. Chen D, Li SQ, Wang JF, Feng Y, Liu Y (2019) A multi-objective trajectory planning method based on the improved immune clonal selection algorithm. *ROBOT CIM-INT MANUF* 59:431–442. <https://doi.org/10.1016/j.rcim.2019.04.016>
7. Zuo YH (2013) Study on nurbs surface tool trajectory planning of 6r engraving robot. *Adv Mat Res* 711:422–425. <https://doi.org/10.4028/www.scientific.net/AMR.711.422>
8. Liu Y, Shi L, Tian XC (2018) Weld seam fitting and welding torch trajectory planning based on NURBS in intersecting curve welding. *INT J ADV MANUF TECH* 95(1–4):2457–2471. <https://doi.org/10.1007/s00170-017-1374-y>
9. Chen W, Liu JJ, Tang Y, Huan J (2017) Trajectory optimization of spray painting robot for complex curved surface based on exponential mean bezier method. *MATH PROBL ENG*. <https://doi.org/10.1155/2017/4259869>
10. Ogbemhe J, Mpofu K, Tlale N (2019) Optimal trajectory scheme for robotic welding along complex joints using a hy-

- brid multi-objective genetic algorithm. *IEEE Access* 7:158753 - 158769. <https://doi.org/10.1109/ACCESS.2019.2950561>
11. Oberherber M, Gattringer H, Mueller A (2015) Successive dynamic programming and subsequent spline optimization for smooth time optimal robot path tracking. *MECH SCI* 6(2):245-254. <https://doi.org/10.5194/ms-6-245-2015>
12. Kaserer D, Gattringer H, Mueller A (2019) Nearly optimal path following with jerk and torque rate limits using dynamic programming. *IEEE T ROBOT* 35(2):521-528. <https://doi.org/10.1109/TRO.2018.2880120>
13. Zhang Q, Li SR, Guo JX (2012) Smooth time-optimal tool trajectory generation for CNC manufacturing systems. *J MANUF SYST* 31(3):280-287. <https://doi.org/10.1016/j.jmsy.2012.06.001>
14. Liu HS, Lai XB, Wu WX (2013) Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints. *ROBOT CIM-INT MANUF* 29(2):309-317. <https://doi.org/10.1016/j.rcim.2012.08.002>
15. Xie YE, Wu XD, Inamori T, Shi Z, Sun XZ, Cui HT (2019) Compensation of base disturbance using optimal trajectory planning of dual-manipulators in a space robot. *ADV SPACE RES* 63(3):1147-1160. <https://doi.org/10.1016/j.asr.2018.10.034>
16. Ramabalan S, Saravanan R, Balamurugan C (2009) Multi-objective dynamic optimal trajectory planning of robot manipulators in the presence of obstacles. *INT J ADV MANUF TECH* 41(5-6):580-594. <https://doi.org/10.1007/s00170-008-1506-5>
17. Marcos MDG, Machado JAT, Azevedo-Perdicoulis TP (2009) Trajectory planning of redundant manipulators using genetic algorithms. *COMMUN NONLINEAR SCI* 14(7):2858-2869. <https://doi.org/10.1016/j.cnsns.2008.10.014>
18. Wang WJ, Tao Q, Cao YT, Wang XH, Zhang X (2020) Robot time-optimal trajectory planning based on improved cuckoo search algorithm. *IEEE Access* 8:86923-86933. <https://doi.org/10.1109/ACCESS.2020.2992640>
19. Liu C, Cao GH, Qu YY, Cheng YM (2020) An improved pso algorithm for time-optimal trajectory planning of delta robot in intelligent packaging. *INT J ADV MANUF TECH* 107:1091-1099. <https://doi.org/10.1007/s00170-019-04421-7>
20. Chettibi T, Lehtihet HE, Haddad M, Hanchi S (2004) Minimum cost trajectory planning for industrial robots. *EUR J MECH A-SOLID* 23(4):703-715. <https://doi.org/10.1016/j.euromechsol.2004.02.006>
21. Chui C, Boyce M (1999) Monte Carlo Techniques. Springer Netherlands, Berlin.
22. Saramago SFP, Steffen V (1999) Dynamic optimization for the trajectory planning of robot manipulators in the presence of obstacles. *J BRAZ SOC MECH SCI* 21(3):372-383. <https://doi.org/10.1590/S0100-73861999000300001>
23. Luo YQ, Yuan XG, Liu YJ (2007) An improved pso algorithm for solving non-convex nlp/minlp problems with equality constraints. *COMPUT CHEM ENG* 31(3):153-162. <https://doi.org/10.1016/j.compchemeng.2006.05.016>
24. Liu CJ, Ding WF, Li Z, Yang CY (2016) Prediction of high-speed grinding temperature of titanium matrix composites using bp neural network based on pso algorithm. *INT J ADV MANUF TECH* 89:2277C2285. <https://doi.org/10.1007/s00170-016-9267-z>
25. Saad MS, Nor AM, Baharudin ME, Zakaria MZ and Aiman AF (2019) Optimization of surface roughness in FDM 3D printer using response surface methodology, particle swarm optimization, and symbiotic organism search algorithms. *INT J ADV MANUF TECH* 105(12):5121-5137. <https://doi.org/10.1007/s00170-019-04568-3>