IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 14, 2019, accepted September 24, 2019, date of publication October 7, 2019,
date of current version October 21, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2945824

Online Time-Optimal Trajectory Planning for Robotic Manipulators Using Adaptive Elite Genetic Algorithm With Singularity Avoidance

YI LIU, CHEN GUO, AND YONGPENG WENG
College of Marine Electrical Engineering, Dalian Maritime University, Dalian 116026, China

Corresponding author: Chen Guo (dmuguoc@126.com)

This work was supported in part by the fundamental research funds for the Central Universities under Grant 3132019106 and Grant 3132019318.

ABSTRACT In this paper, a new method of online planning high smooth and time-optimal trajectory for robotic manipulators that applies an adaptive elite genetic algorithm with singularity avoidance (AEGA-SA) is presented. The strategy is designed as a combination of the time-optimal trajectory planning with quintic polynomial in Cartesian space. For improving optimization performance, elitist group and adaptive adjustment mechanisms are used based on genetic algorithm (GA) framework. In the meantime, GA is combined with singularity avoidance mechanism to avoid the singularities appearing in the trajectory, improves the recognition capability of optimum solution. Experimental results show that, the proposed approach is more effective and better performance than the original GA and its variants, with ensuring a both smooth and efficiency performance for the robotic manipulators.

INDEX TERMS Robotic manipulators, online trajectory planning, singularity avoidance, minimum-time optimization, quintic polynomial.

I. INTRODUCTION

Trajectory planning is a very popular and valuable benchmark problem in robotic manipulators [1]–[3]. Trajectory can be planned either in Cartesian space or in joint space to make the motion of the robot smooth and continuous. Trajectory planning in Cartesian space can be planned online without interpolation for joints which is more intuitive and accurate than planning in joint space [4]–[6]. However, large amount of kinematic calculation, singularities of the joints are following. With the development of computer technology, the real-time calculation of kinematics is improved [7]. Considering the requirement to increase productivity in automatic production line, this paper aims at online time-optimal trajectory planning in Cartesian space.

Several kinds of curves such as trigonometric spline [8], polynomial [9], B-spline [10] and cubic spline [11] are used for trajectory generation in literature. For online time-optimal trajectory planning in Cartesian space, polynomial directly maps the dynamic relationship between the end effector and the joints of the robotic manipulators [12]. Moreover, polynomial provides simpler transformational relation between Cartesian and joint space which reduces the cost of the computation resources.

Optimal trajectory planning is to generate a trajectory from an initial pose to a final pose that satisfy certain kinematic and dynamic constraints. Many optimization algorithms have been applied for optimizing trajectory such as genetic algorithm (GA) [13], [14], particle swarm optimization (PSO) [15]–[17] and sequential quadratic programming (SQP) [18], [19]. Online time-optimal trajectory planning in Cartesian space needs to consider constrained optimization as well as singularity avoidance. The singularity results in sudden change of some joints speed. Optimization algorithms have to extend the runtime to pass the singularity for meeting angular velocity constraint, which affects the algorithms' identification of the optimal time.

The trajectory curves of the robot joints are time-discrete. Compared with the existing optimization algorithms, GA is more accurate to optimize discrete objects using binary

The associate editor coordinating the review of this manuscript and approving it for publication was Christopher H. T. Lee.

encoding format. Moreover, GA is more expandable for framing object function in unique domain [20], [21]. Because of these kinds of characteristics, GA can integrate with singularity avoidance strategy and increase the ability to identify the optimal time.

The GA has drawbacks such as precocious phenomenon, atavism phenomenon, and searching dullness. Precocious phenomenon means that GA falls into local optimization prematurely, and is difficult to jump to the global optimization. Atavism phenomenon means that newly generated individuals universally have worse fitness. Searching dullness slows down the convergence rate and makes the temporary optimal solution fluctuate when GA almost gets the optimal solution. The above-mentioned drawbacks need to be solved to improve the efficiency and accuracy of GA.

In view of that above considerations, this paper presents an adaptive elite genetic algorithm with singularity avoidance (AEGA-SA) for online time-optimal trajectory planning in Cartesian space. The algorithm takes into account the kinematical constraints in joint space which consists of bounds on the joint velocity, acceleration and jerk. The rest of this paper is organized as follows. In section 2, the singularity separation for Jacobian matrix is introduced. In section 3, the singularity avoidance method based on singularity separation is formulated. In section 4, methods for improving GA is introduced and AEGA-SA is proposed. In section 5, the application of AEGA-SA to the online time-optimal trajectory planning is illustrated. In section 6, the proposed algorithm is executed on a 6-DOF industrial robot. The results of online trajectory planning in Cartesian space among AEGA-SA, the original GA and its variants are compared. In section 7, conclusions are presented.

II. SINGULARITIES SEPARATION FOR JACOBIAN MATRIX

The singularity of kinematics is an inherent property of the manipulators. As shown in Fig.1, the 6-DOF (degrees of freedom, DOF) serial mechanism can be separated into two substructures of 3-DOF and considered with clarity. In order to conveniently discuss, some symbols and variables are defined as follows:

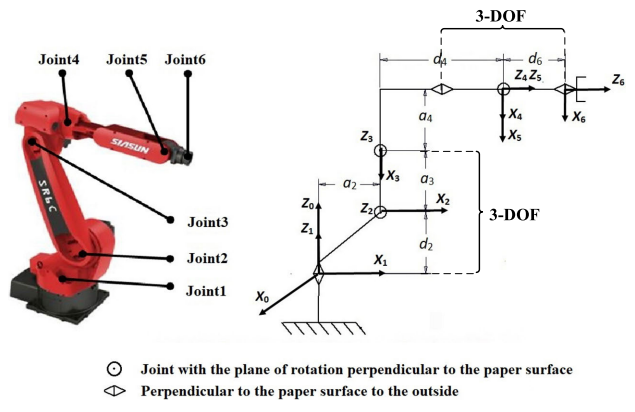


FIGURE 1. 6-DOF robot and its link coordinates.

·$\{T\}$: T coordinate system, T means any coordinate system of the joints.
·${}_{i}^{i-1}\boldsymbol{T}$: Transformation matrix from joint i-1 to joint i.
·$[\boldsymbol{n}, \boldsymbol{o}, \boldsymbol{a}]$: Pose vector of the end effector.
·$\boldsymbol{J}(\theta)$: 6×6 Jacobian matrix.
·${}^{T}\boldsymbol{J}(\theta)$: 6×6 Jacobian matrix based on $\{T\}$.
·${}^{\mathrm{e}}\boldsymbol{J}(\theta)$: 6×6 Jacobian matrix based on $\{6\}$.
·${}_{6}^{0}\boldsymbol{R}$: 3×3 pose matrix forward kinematics.
·$[{}^{\mathrm{e}}\boldsymbol{v}, {}^{\mathrm{e}}\boldsymbol{w}]^{\mathrm{T}}$: Description of the velocity vector for the end effector based on $\{6\}$.
·$[v_x, v_y, v_z]^{\mathrm{T}}$, $[w_x, w_y, w_z]^{\mathrm{T}}$: Linear and angular velocity of the end effector based on $\{0\}$.

$\boldsymbol{J}(\theta)$ transmit velocity from joints to end effector shown as (1). The first 3 lines and the last 3 lines undertake the transmission for linear velocity and angular velocity of the end effector.

$$\begin{bmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} {}_{0}^{\mathrm{e}}\boldsymbol{v} \\ {}_{0}^{\mathrm{e}}\boldsymbol{w} \end{bmatrix} = \boldsymbol{J}(\theta)\dot{\boldsymbol{\theta}} = \boldsymbol{J}(\theta) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{bmatrix} = \boldsymbol{J}(\theta) \begin{bmatrix} \dot{\theta}_{\mathrm{u}} \\ \dot{\theta}_{\mathrm{d}} \end{bmatrix} \tag{1}$$

${}^{T}\boldsymbol{J}(\theta)$ has 6 columns in total, ${}^{T}\boldsymbol{J}_i(\theta)$ is determined by ${}_{6}^{i-1}\boldsymbol{T}$. By using differential transformation method, the ${}^{T}\boldsymbol{J}_i(\theta)$ is expressed as:

$${}^{T}\boldsymbol{J}_i(\theta) = \begin{bmatrix} ({}_{6}^{i-1}\boldsymbol{p} \times {}_{6}^{i-1}\boldsymbol{n})_z \\ ({}_{6}^{i-1}\boldsymbol{p} \times {}_{6}^{i-1}\boldsymbol{o})_z \\ ({}_{6}^{i-1}\boldsymbol{p} \times {}_{6}^{i-1}\boldsymbol{a})_z \\ {}_{6}^{i-1}\boldsymbol{n}_z \\ {}_{6}^{i-1}\boldsymbol{o}_z \\ {}_{6}^{i-1}\boldsymbol{a}_z \end{bmatrix} \tag{2}$$

$\boldsymbol{J}(\theta)$ is based on the base coordinate system, therefore, ${}^{T}\boldsymbol{J}(\theta)$ and $\boldsymbol{J}(\theta)$ have the following transformation relation:

$$\boldsymbol{J}(\theta) = \begin{bmatrix} {}_{6}^{0}\boldsymbol{R} & \boldsymbol{0} \\ \boldsymbol{0} & {}_{6}^{0}\boldsymbol{R} \end{bmatrix} {}^{T}\boldsymbol{J}(\theta) = \begin{bmatrix} \boldsymbol{J}_{11} & \boldsymbol{J}_{12} \\ \boldsymbol{J}_{21} & \boldsymbol{J}_{22} \end{bmatrix} \tag{3}$$

According to (1), with regard to $[{}^{\mathrm{e}}\boldsymbol{v}, {}^{\mathrm{e}}\boldsymbol{w}]^{\mathrm{T}}$ and ${}^{\mathrm{e}}\boldsymbol{J}(\theta)$, we have:

$$\begin{bmatrix} {}^{\mathrm{e}}\boldsymbol{v} \\ {}^{\mathrm{e}}\boldsymbol{w} \end{bmatrix} = {}^{\mathrm{e}}\boldsymbol{J}(\theta)\dot{\boldsymbol{\theta}} = \begin{bmatrix} {}^{\mathrm{e}}\boldsymbol{J}_{11} & \boldsymbol{0} \\ {}^{\mathrm{e}}\boldsymbol{J}_{21} & {}^{\mathrm{e}}\boldsymbol{J}_{22} \end{bmatrix} \dot{\boldsymbol{\theta}} \tag{4}$$

According to (3) and (4), $[\dot{\theta}_u, \dot{\theta}_d]^{\mathrm{T}}$ can transform into $[{}^{\mathrm{e}}\boldsymbol{v}, {}^{\mathrm{e}}\boldsymbol{w}]^{\mathrm{T}}$ using block Jacobian matrix:

$$\begin{aligned} \begin{bmatrix} \dot{\boldsymbol{\theta}}_{\mathrm{u}} \\ \dot{\boldsymbol{\theta}}_{\mathrm{d}} \end{bmatrix} &= [{}^{\mathrm{e}}\boldsymbol{J}(\theta)]^{-1} \begin{bmatrix} {}^{\mathrm{e}}\boldsymbol{v} \\ {}^{\mathrm{e}}\boldsymbol{w} \end{bmatrix} \\ &= \begin{bmatrix} [\boldsymbol{J}_{11}]^{-1} & \boldsymbol{0} \\ -[\boldsymbol{J}_{11}]^{-1}\boldsymbol{J}_{21}[\boldsymbol{J}_{22}]^{-1} & [\boldsymbol{J}_{22}]^{-1} \end{bmatrix} \begin{bmatrix} {}_{0}^{\mathrm{e}}\boldsymbol{v} \\ {}_{0}^{\mathrm{e}}\boldsymbol{w} \end{bmatrix} \end{aligned} \tag{5}$$

With:

$$\begin{cases} \dot{\boldsymbol{\theta}}_{\mathrm{u}} = [\boldsymbol{J}_{11}]^{-1} {}_{0}^{\mathrm{e}}\boldsymbol{v} \\ \dot{\boldsymbol{\theta}}_{\mathrm{d}} = [\boldsymbol{J}_{22}]^{-1} [{}_{0}^{\mathrm{e}}\boldsymbol{w} - \boldsymbol{J}_{21}\dot{\boldsymbol{\theta}}_{\mathrm{u}}] \end{cases} \tag{6}$$

When any path point in Cartesian space make $|\boldsymbol{J}_{11}|$ or $|\boldsymbol{J}_{22}|$ be 0, singularity condition is satisfied, and singularity value appears. Therefore, $|\boldsymbol{J}_{11}| = 0$ is called first half singularity and $|\boldsymbol{J}_{22}| = 0$ is called latter part singularity. The separation of the singularity parameters from the Jacobian inverse has less computation and the location is more accurate.

III. SINGULARITIES AVOIDANCE METHOD BASED ON SINGULARITIES SEPARATION AND CYCLIC DAMPED COEFFICIENT

In the trajectory, all $[\theta_1(t), \ldots, \theta_6(t)]^{\mathrm{T}}$ of the same instant are substituted in $|\boldsymbol{J}_{11}|$ and $|\boldsymbol{J}_{22}|$ to judge the singularity condition and the occurrence location. The detected singularity area needs to be avoided under reasonable means to ensure normal operation during the online trajectory planning.

The $[\theta_1(t_s), \ldots, \theta_6(t_s)]^{\mathrm{T}}$ vectors are found from searching the whole trajectory which satisfy the singularity condition using (6). The time within $[t_s\text{-}t_a, t_s\text{-}t_b]$ is called singularity time range. The bounds t_a and t_b limit the time area where the velocity of a certain joint exceed the kinematical constraints. The position curve of the joints is discrete in operation of the motion controller. $\dot{\theta}_i$ can be computed by the difference equation:

$$\dot{\theta}_i(t+\Delta t) = \frac{\theta_i(t+\Delta t) - \theta_i(t)}{\Delta t}, \quad i = 1, 2, \ldots, 6 \tag{7}$$

The maximum difference between $\theta_i(t + \Delta t)$ and $\theta_i(t)$ has a magnitude of 10^{-3}. According to the kinematical constraints, difference threshold $\varphi_e(\varphi_e = 0.0042\text{rad/s}, \Delta t = 4\text{ms})$ is set for $|\theta_i(t + \Delta t)\text{-}\theta_i(t)|$. If $|\theta_i(t + \Delta t)\text{-}\theta_i(t)|$ exceeds φ_e, the damped coefficient λ is introduced for k cycles of running down until $|\theta_i(t + \Delta t)\text{-}\theta_i(t)| < \varphi_e$:

$$\begin{cases} \dot{\theta}_i(t+\Delta t)' = \lambda^k \dot{\theta}_i(t+\Delta t) \\ \lambda = 1 - \dfrac{\varphi_e}{|\theta_i(t+\Delta t) - \theta_i(t)|} \end{cases} \tag{8}$$

After singularity avoidance, the joints velocity of the singularity area changed into $\theta(t)$', then joint i position curve $\theta_i(t)'$ can be obtain by integral:

$$\theta_i(t+\Delta t)' = \theta_i(t) + \int_t^{t+\Delta t} \dot{\theta}_i(t+\Delta t)' dt \tag{9}$$

Finally, $\theta_i(t)'$ substitutes $\theta_i(t)$ at same time and the singular value can be avoided in actual motion. In MATLAB, the end effector of the simulative robot move form pose state [0.24, 0.2,0.9, 0;0.7, −0.6, −0.03, 0; 0.6, 0.7, −0.3, 0; 0.8, 0.8, 0.19,1]$^{\mathrm{T}}$ to [0.54, −0.5, 0.62, 0; −0.74, −0.66, 0.05, 0;0.38, −0.49, −0.7, 0; 0.7, −0.71, 0.17,1]$^{\mathrm{T}}$. The runtime is 12s. Trapezoidal speed mode for the end effector is used. The acceleration and deceleration time are 2s with peak speed 0.2m/s. This trajectory of joint 4 and 6 avoids singularity area by using singularities avoidance method based on cyclic damped reciprocal. Before and after singularities avoidance, the velocity curves of the joints are shown as Fig.2.

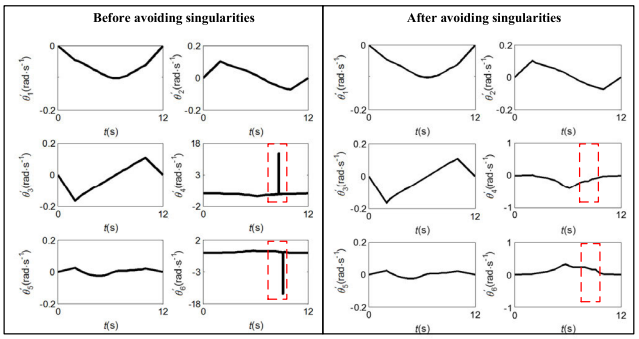


FIGURE 2. Comparison between before and after avoiding singularities of velocity curves of the joints.

IV. ADAPTIVE ELITE GENETIC ALGORITHM COMBINING WITH SINGULARITY AVOIDANCE

Selection operator of GA screens the population and retains more excellent individuals based on roulette method. Roulette method is overwhelmingly dependent on probability transfer mechanism and lead precocious phenomenon, atavism phenomenon. Integrating elite population strategy with selection operator can speed up the convergence of GA and improve the evolutionary direction of the population. Firstly, the individuals p_i of initial population $\boldsymbol{P}'$ were sorted in fitness order. Secondly, $\boldsymbol{P}'$ was divide into k equal sub-populations such as $\boldsymbol{P}'_1, \boldsymbol{P}'_2, \ldots, \boldsymbol{P}'_k$. Finally, $\boldsymbol{P}'_m$ with the strongest fitness substitutes for the one with the worst fitness. Specific action mechanism of the mentioned strategy is shown as Fig.3. Improved selection operator speeds up the convergence of GA.

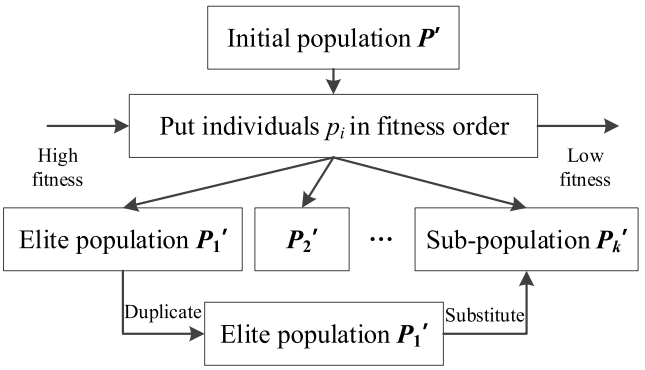


FIGURE 3. Action mechanism of selection operator with elite population strategy integrated.

Crossover operator is the main pathway for developing new individuals. Mutation operator plays a decisive role to local search of GA. In the early stages of the algorithm, the population need to evolve in its entirety. In the later stages, GA has no use for the large-scale changes of the individuals. Instead, several individuals mutate to approximate the optimal solution in local search mode. Therefore, the convergence of GA depends upon the probability of crossover and mutation operator in optimization process.

For reflecting the average fitness level of the population and the discrete fitness distribution of the individuals, fitness

expectation $E(f_i)$ and fitness variance $D(f_i)$ are introduced as:

$$E(f_i) = f_{avg} = \frac{\sum_{i}^{M} f_i}{M} \tag{10}$$

$$D(f_i) = E(f_i^2) - E^2(f_i) = \frac{\sum_{i}^{M} f_i^2}{M} - f_{avg}^2 \tag{11}$$

where, f_i denotes the fitness of the individuals. The mean fitness of the population is expressed as f_{avg}. Crossover probability p_c and mutation probability p_m are redefine as:

$$p_c = \frac{1}{1 + e^{-\frac{h_1}{v}}} - 0.1, \quad h_1 \in R + \tag{12}$$

$$p_m = \frac{h_2}{5(1 + e^{\frac{1}{v}})}, \quad h_2 \in (0, 1) \tag{13}$$

With:

$$v = \frac{E(f_i) + 1}{D(f_i)} \tag{14}$$

where v is adaptive parameter. In the early stages of population evolution, because of the low fitness of the individuals, p_c is increased and p_m is reduced to enhance the ability of global search. In the later stages, local search around several individuals makes the population have activity and does not affect the general trend of convergence.

If the end effector goes through singularity area, the singularities make angular velocity of the joints great change. GA has to prolong the runtime of the trajectory to satisfy the dynamic and static thresholds of the joints which affect GA to identify the optimal runtime of the trajectory. Before selecting an individual (the runtime of the trajectory) for trajectory planning, the proposed singularities avoidance mechanism can be used to eliminate the distractions, and then trajectory optimization and fitness calculation are carried out.

V. TIME-OPTIMAL TRAJECTORY PLANNING USING AEGA-SA

The optimal time trajectory optimization first needs to select the trajectory curve and establish the fitness function. Trajectory curve determines the smoothness and continuity of the robot action. The fitness function is an important indicator for judging individuals.

According to the waypoints $\{\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_m\}$, the corresponding joints vector $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_m\}$ can be obtained. Under kinematical constraints such as velocity, acceleration and so on, the time-optimal trajectory planning problem can be defined as:

$$\min t_e = \sum_{i=1}^{m} \min t_i \tag{15}$$

$$\text{Subject to}: \begin{cases} \left|\dot{\theta}_{ki}(t)\right| \le W_{i\max}, \\ \quad k = 1, 2, \ldots, 6, i = 1, 2, \ldots, m \\ \left|\ddot{\theta}_{ki}(t)\right| \le A_{i\max}, \\ \quad k = 1, 2, \ldots, 6, i = 1, 2, \ldots, m \\ |\theta_{ki}(t)| \le Jerk_{i\max}, \\ \quad k = 1, 2, \ldots, 6, i = 1, 2, \ldots, m \\ \quad t \in [t_0, t_e] \end{cases} \tag{16}$$

For straight through neighboring waypoints by the robot, there are starting time t_0 and runtime t_e. Quintic polynomial can satisfy the continuity of the joints' jerk curves without artificial interpolation for the neighboring waypoints. The trajectory curve model in Cartesian space is shown as:

$$\begin{bmatrix} s(t_0) \\ v(t_0) \\ a(t_0) \\ s(t_e) \\ v(t_e) \\ a(t_e) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & t_e & t_e^2 & t_e^3 & t_e^4 & t_e^5 \\ 0 & 1 & 2t_e & 3t_e^2 & 4t_e^3 & 5t_e^4 \\ 0 & 0 & 2 & 6t_e & 12t_e^2 & 20t_e^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \tag{17}$$

In (17), parameters $a_0, a_1, \ldots, a_5$ belong to the real. The trajectory relation between joint space and Cartesian space is kinematics. The trajectory curve of joint i with respect to time is given by:

$$\frac{d\theta_i}{dt} = \frac{\partial \theta_i}{\partial p_x}\frac{\partial p_x}{\partial t} = \frac{d\theta_i}{ds}\frac{ds}{dt}, \quad i = 1, 2 \ldots 6 \tag{18}$$

It has been recognized in (18) that the dynamic feature of the trajectory in joint space is consistent with Cartesian space. Therefore, quintic polynomial adopted on end effector trajectory makes the joints have the same smoothness and continuity of motion state.

The objective function is defined as the runtime t. The fitness value needs to consider the kinematical constraints such as overrun of position, velocity, acceleration and jerk of the joints. The fitness f_m of individual t_i can be computed by the formula:

$$f_m = \frac{\mu_4 f(\theta_i)}{\mu_0 f(t_i) + \mu_1 \sum_{i=1}^{6} f(w_i) + \mu_2 \sum_{i=1}^{6} f(a_i) + \mu_3 \sum_{i=1}^{6} f(j_i)} \tag{19}$$

where, $f(\theta_i), f(w_i), f(a_i)$ and $f(j_i)$ are respectively the function of over-limit judging for the position, angular velocity, acceleration and jerk of joint i. The values of the joints are 1 within the kinematical constraints, otherwise are 0. $f(t_i)$ is the object function for the ith path. $\mu_0, \mu_1, \ldots, \mu_4$ is the weighting factors for balancing the effect of each function in the AEGA-SA.

The AEGA-SA algorithm enhances the exploration capacities of the original GA. It can be used to minimize the total traveling time of the robotic manipulators. The detailed processes are described as follows:

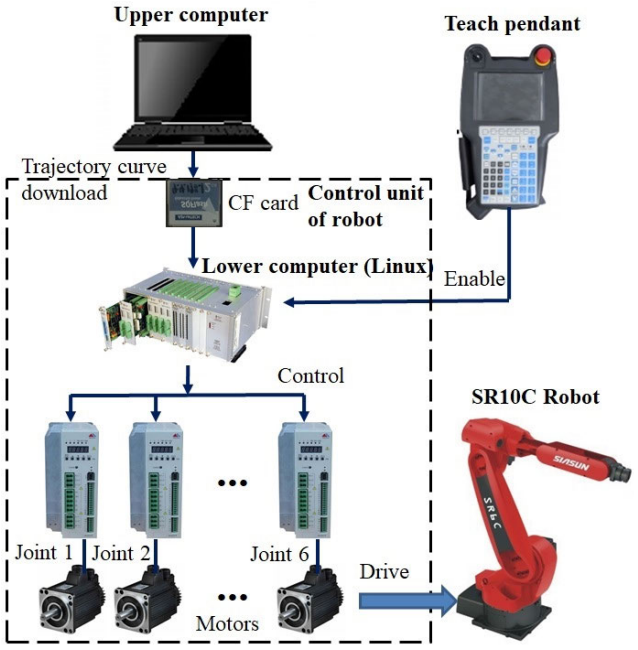


FIGURE 4. Experimental platform for trajectory planning.

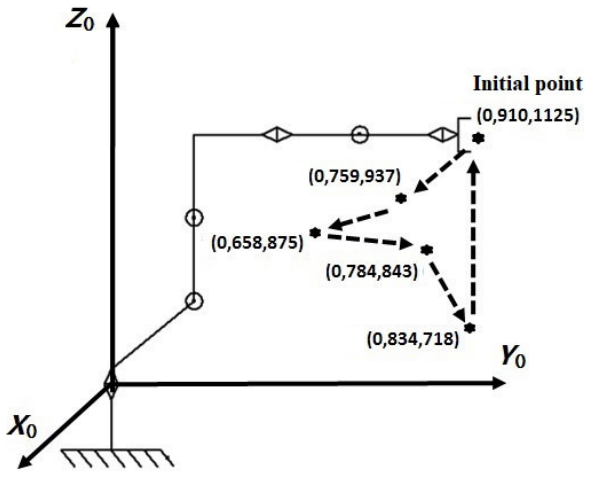


FIGURE 5. Working paths in the operating space.

Step 1: Optimization parameters are initialized and one population $\boldsymbol{P}_i$ with n individuals (t_e) is obtained randomly. There are k waypoints, and there will be k paths. Under the first segment of the paths, an individual $p_m(m = 1,2, \ldots, n)$ is extracted for judging singularities. The normal path can be obtained by singularities avoidance method. The trajectory of the end effector and joints can be obtained using p_m, (17), and inverse kinematics equation. If all the joints' curves satisfy the kinematical constraints, p remain unchanged, else, p_m is increased. After investigation of all the individuals, new population $\boldsymbol{P}_i^{'}$ is produced.

Step 2: Circulation utilization of the improved selection, crossover, and mutation operators prompted the population to evolve until reach the number of iterations. The individual

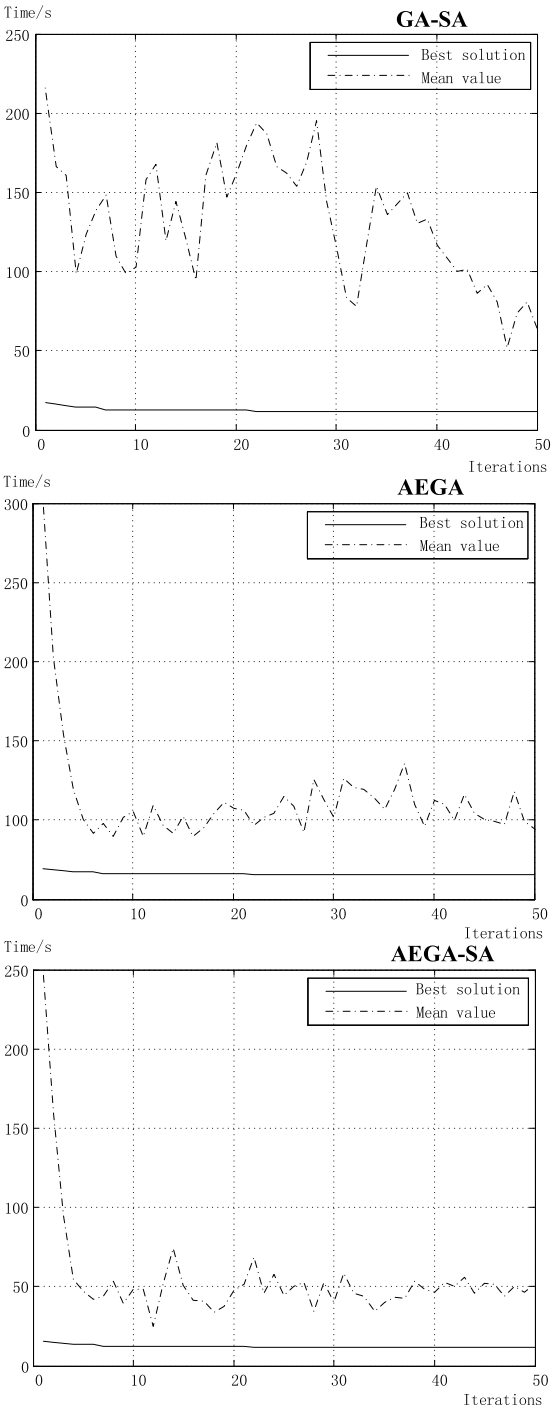


FIGURE 6. Convergence plots for trajectory planning by the various methods.

with the highest fitness is selected for $t_e^l(l = 1, 2, \ldots, k)$ of the first segment. After improving the trajectory of all the paths, T_e is the accumulation of t_e^l shown as:

$$\min T = T_e = \sum_{i=1}^{k} \min t_e^l, t_e^l \in [T_{li}, T_{ui}] \tag{20}$$

The upper bound T_{li} and the lower bound T_{ui} is determined by the optimization parameters.

Step 3: The robot bottom control use servo controller. Therefore, the trajectory curves of the joints are in the form of time-discrete. There are adjacent waypoints $s(t_{i+1})$ and $s(t_i)$ in the task space. The waypoints defer to robotic work precision d_m of the end effector:

$$|s(t_{i+1}) - s(t_i)| < d_m, \quad i = 1, 2, \ldots, n-1 \tag{21}$$

The minimum interval can be computed by the following formula:

$$\Delta t_i = t_{i+1} - t_i = \frac{|s(t_{i+1}) - s(t_i)|}{\bar{v}_i}, \quad i = 1, 2, \ldots, p \tag{22}$$

$\bar{v}_i$ denotes average velocity through the adjacent waypoints. The minimum Δt_i is 4ms. Through differential operation of the joints' trajectory curves with discrete time $\boldsymbol{\theta}(t)$, we can obtain the velocity curves $\dot{\theta(t)}$ and the other dynamic curves.

VI. EXPERIMENTAL RESULTS

To verify the effectiveness of the AEGA-SA algorithm proposed in section 5, experiments are implemented on SIASUN SR10C platform for trajectory planning shown in Fig.4. The platform is composed of a 6-DOF robotic manipulator and control units. The D-H parameters and kinematical constraints of the manipulator are listed in Tables 1 and 2, respectively.

TABLE 1. D-H parameters of robotic manipulator.

Link	$\boldsymbol{d}_i$(mm)	$\boldsymbol{a}_i$ (mm)	$\boldsymbol{\theta}_i$ (°)	$\boldsymbol{\alpha}_i$(°)
1	0	0	90	0
2	420	160	0	90
3	0	575	-90	0
4	645	130	0	-90
5	0	0	0	90
6	105	0	0	-90

TABLE 2. Kinematical constraints.

Joint	Position (°)	Velocity (rad./s)	Acceleration (rad./s^2)	Jerk (rad./s^3)
1	[-180,180]	1.74	1.05	1.05
2	[-80,60]	1.66	1.05	1.15
3	[-137.5,150]	1.74	1.31	1.48
4	[-360,360]	2.62	1.22	1.22
5	[-130,130]	2.27	1.57	1.31
6	[-360,360]	1.92	1.40	1.22

In the experiments, the end effector of the robotic manipulator will move along several linear paths that pass through random five points shown in Fig.5 sequentially. As formulated before, a quintic polynomial is used to plan the trajectory of the end effector and the joints. AEGA-SA algorithm ensures the high efficiency, smoothness and continuity of the trajectory.

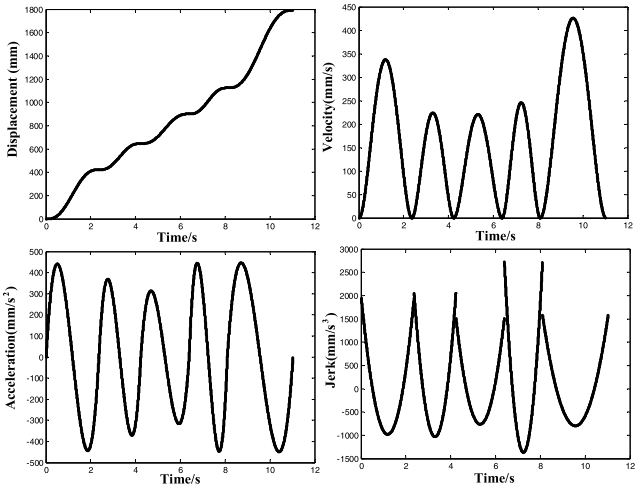


FIGURE 7. Dynamic curves of the end effector in the quintic polynomial trajectory.

TABLE 3. Comparative results for trajectory planning by the various methods.

Methods	BS (s)	CT (s)	CI
GA-SA	11.088	521.4706	46
AEGA	34.224	362.1284	29
AEGA-SA	10.944	442.6367	27

AEGA-SA, adaptive elite genetic algorithm (AEGA), and genetic algorithm combining with singularity avoidance (GA-SA) algorithms proposed in this paper are applied to generate the trajectory with minimum runtime while satisfying the kinematical constrains. The parameters of the algorithms are set as follows: searching range is set (0s, 100s), The population number is set to 300 and the numerical value of individuals in each group are random. The maximum number of iterations is set to 50. The generation gap is set to 0.9. The individuals are encoded in binary format.

The trajectory planning programs based on the aforementioned algorithms are executed on an upper computer that has an Intel core i7 2.2 GHz processor and is an important component of the robotic control unit. The upper computer download data to CF card by TCP/IP. The lower computer in control unit of robot is enabled into trajectory planning mode by teach pendant and receives the planning results via CF card. AEGA-SA is compared with the aforementioned methods for the best solution (BS) of T_e, computing time (CT) and the closing iteration (CI). The comparative results of all the methods are listed in Table3. It is observed from Table3 that the best solution obtained by AEGA-SA is better than others. Singularities rob AEGA of its capacity to best solution. Moreover, it is also observed from the results that the singularity avoidance method makes computing time longer, especially the GA-SA algorithm with slow convergence.

In addition, the convergence rate of all the methods is compared. The convergence plots shown in Fig.6 are obtained

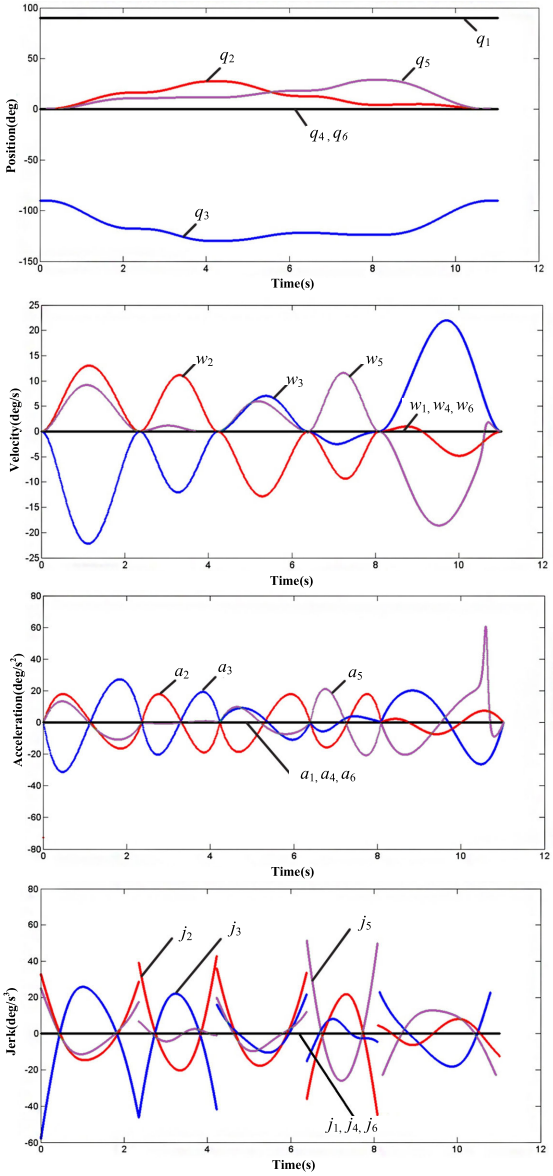


FIGURE 8. Dynamic curves of the joints after singularity avoidance.

by best solutions and mean values of the objective function for each generation. It can be seen from Fig.6 that the convergence of AEGA-SA is better than the other algorithms without precocious phenomenon, atavism phenomenon, and searching dullness.

The best trajectory planning results obtained by AEGA-SA are depicted from Fig.7 and Fig.8. The figures show the trajectories of minimum time and their derivatives in Cartesian and joint space. The results satisfy the kinematics constraints. The planning results are simulated in inventor for showing the visible trajectory of the end effector. Then, the planning results are transmitted to the lower computer. The lower computer controls the end effector to maneuver along the planning trajectory which moves along the path. This process is shown in Fig.9. During this process, the robotic manipulator operates smoothly.

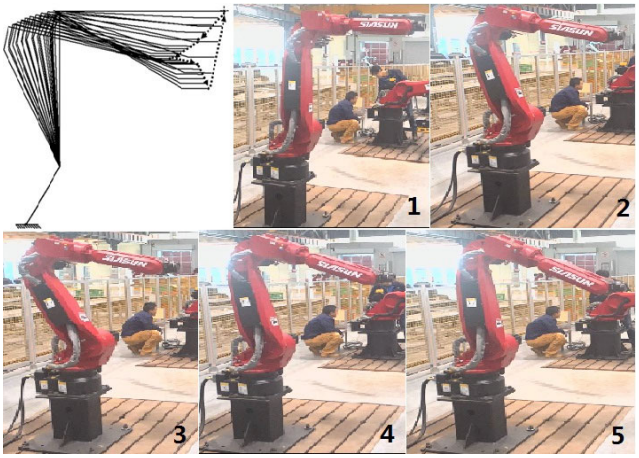


FIGURE 9. Experiment of online time-optimal trajectory planning.

VII. CONCLUSION

This paper proposes an AEGA-SA algorithm for online time-optimal trajectory planning of robotic manipulators. Based on GA framework, elitist group strategy is proposed to speed up the convergence in search for the optimal time. Adaptive adjustment crossover and mutation operators are introduced into GA to increase search activity. Singularity avoidance is combined to avoid the singularities appearing in the online Cartesian space trajectory planning, improves the recognition capability of the best runtime. The trajectories based on quintic polynomial are generated for a 6-DOF robotic manipulator using GA-SA, AEGA, AEGA-SA, respectively. The objective function is the minimum runtime with the kinematics constrains (position, velocity, acceleration and jerk). The experimental results show that the AEGA-SA algorithm successfully obtains better solution than the other two algorithms, and has stronger universality for most of robotic manipulators. In the future work, the AEGA-SA algorithm will be applied to energy optimization and other kinds of optimization problems.

REFERENCES

[1] C. Ji, M. Kong, and R. Li, "Time-energy optimal trajectory planning for variable stiffness actuated robot," *IEEE Access*, vol. 7, pp. 14366–14377, 2019.

[2] X. Lv, Z. Yu, M. Liu, G. Zhang, and L. Zhang, "Direct trajectory planning method based on IEPSO and fuzzy rewards and punishment theory for multi-degree-of freedom manipulators," *IEEE Access*, vol. 7, pp. 20452–20461, 2019.

[3] X. Zhang, Y. Fang, and N. Sun, "Minimum-time trajectory planning for underactuated overhead crane systems with state and control constraints," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 6915–6925, Dec. 2014.

[4] C. Zheng, Y. Su, and P. Müller, "Simple online smooth trajectory generations for industrial systems," *Mechatronics*, vol. 19, no. 4, pp. 571–576, 2009.

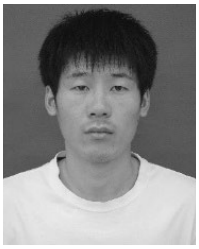
[5] B. Xian, M. S. De Queiroz, D. Dawson, and I. Walker, "Task-space tracking control of robot manipulators via quaternion feedback," *IEEE Trans. Robot. Autom.*, vol. 20, no. 1, pp. 160–167, Feb. 2004.

[6] M. K. Özgören, "Topological analysis of 6-joint serial manipulators and their inverse kinematic solutions," *Mech. Mach. Theory*, vol. 37, no. 5, pp. 511–547, 2002.

[7] T. T. Su, L. Cheng, Y. K. Wang, X. Liang, J. Zheng, and H. Zhang, "Time-optimal trajectory planning for delta robot based on quintic pythagorean-hodograph curves," *IEEE Access*, vol. 6, pp. 28530–28539, 2018.
[8] D. Simon and C. Isik, "Optimal trigonometric robot joint trajectories," *Robotica*, vol. 9, no. 4, pp. 379–386, 1991.
[9] Y. Liu, M. Cong, H. Dong, and D. Liu, "Reinforcement learning and EGA-based trajectory planning for dual robots," *Int. J. Robot. Automat.*, vol. 33, no. 4, pp. 367–378, 2018.
[10] A. Gasparetto and V. Zanotto, "A new method for smooth trajectory planning of robot manipulators," *Mech. Mach. Theory*, vol. 42, no. 4, pp. 455–471, 2007.
[11] S. Kucuk, "Optimal trajectory generation algorithm for serial and parallel manipulators," *Robot. Comput.-Integr. Manuf.*, vol. 48, pp. 219–232, Dec. 2017.
[12] M. Boryga and A. Graboś, "Planning of manipulator motion trajectory with higher-degree polynomials use," *Mech. Mach. Theory*, vol. 44, no. 7, pp. 1400–1419, 2009.
[13] R. Roy, M. Mahadevappa, and C. S. Kumar "Trajectory path planning of EEG controlled robotic arm using GA," *Procedia Comput. Sci.*, vol. 84, pp. 147–151, Dec. 2016.
[14] L. Tian and C. Collins, "An effective robot trajectory planning method using a genetic algorithm," *Mechatronics*, vol. 14, no. 5, pp. 455–470, Jun. 2004.
[15] H.-I. Lin, "A fast and unified method to find a minimum-jerk robot joint trajectory using particle swarm optimization," *J. Intell. Robotic Syst.*, vol. 75, nos. 3–4, pp. 379–392, 2014.
[16] M. Wang, J. Luo, and U. Walter, "Trajectory planning of free-floating space robot using particle swarm optimization (PSO)," *Acta Astronautica*, vol. 112, pp. 77–88, Jul./Aug. 2015.
[17] S. Kucuk, "Energy minimization for 3-RRR fully planar parallel manipulator using particle swarm optimization," *Mech. Mach. Theory*, vol. 62, pp. 129–149, Apr. 2013.
[18] Z. Qingzhen, G. Chen, G. Fei, and R. Zhang, "Reentry trajectory planning optimization based on sequential quadratic programming," in *Proc. 2nd Int. Symp. Syst. Control Aerosp. Astronaut.*, New York, NY. USA, vol. 1, Dec. 2008, pp. 1–5.
[19] A. Gasparetto and V. Zanotto, "A technique for time-jerk optimal planning of robot trajectories," *Robot. Comput.-Integr. Manuf.*, vol. 24, no. 3, pp. 415–426, 2008.
[20] A. S. Saif-Eddine, M. M. El-Beheiry, and A. K. El-Kharbotly, "An improved genetic algorithm for optimizing total supply chain cost in inventory location routing problem," *Ain Shams Eng. J.*, vol. 10, no. 1, pp. 63–76, 2019.
[21] K. Singh, and S. Sundar, "A hybrid steady-state genetic algorithm for the min-degree constrained minimum spanning tree problem," *Eur. J. Oper. Res.*, vol. 276, no. 1, pp. 88–105, 2019.

YI LIU received the B.S., M.S., and Ph.D. degrees from the School of Mechanical Engineering, Dalian University of Technology, Dalian, China, in 2003, 2011, and 2018, respectively, and the second B.S. degree from the School of Foreign Languages, in 2007. He is currently with the School of Marine Electrical Engineering, Dalian Maritime University, Dalian. His current research interests include deep learning, machine learning, mechanized welding, optimal control, motion planning, trajectory planning, and robotics.

CHEN GUO was born in Liaoning, China, in 1956. He received the B.S. degree from the Department of Automatic Control, Chongqing University, China, in 1982, and the M.S. and Ph.D. degrees both in marine engineering automation from Dalian Maritime University, China, in 1985 and 1992, respectively. He currently works as a Professor and the Director of the Institute of Ship Automation and Simulator, School of Marine Electrical Engineering, Dalian Maritime University, China. His current research interests include intelligent control, soft computing, and marine system automation and simulation.



YONGPENG WENG was born in Yingkou, Liaoning, China, in 1986. He received the B.S. degree in electrical engineering and automation from Liaoning University, in 2010, and the M.S. and Ph.D. degrees in control theory and control engineering from the School of Information Science and Engineering, Northeastern University. He is currently with the School of Marine Electrical Engineering, Dalian Maritime University, Dalian, China. His research interests include data-driven adaptive control, sliding mode control, modeling of complex industry process and intelligent control, unmanned vehicles, autonomous robotics, and autonomous control.