A finite and fast time-optimal trajectory planning based on velocity sinking method for robots along specified path

DOI : 10.36909/jer.ICCSCT.19477

Wenhu Nan*

Key Laboratory of heavy duty flexible robot in mechanical industry, LanZhou University of Technology, LanZhou Gansu, 730050, China.

* Corresponding Author: nanwenhu@163.com

ABSTRACT

Time-optimal trajectory planning is used in many areas of industrial robots and has a wide range of applications. This paper presents a finite and fast computation method for calculating time-optimal robot trajectories along specified geometric paths, namely the velocity sinking method. The velocity sinking method differs from those proposed in the existing literature which solve uncertain optimization problem. Firstly, the maximum velocity in multiple state constraints is calculated using a path parameterization method. Then, the pseudo maximum velocity trajectory is calculated using a numerical integration method. Finally, the time-optimal trajectory planning under path constraints is determined using an iterative velocity sinking algorithm. The experiment demonstrates the effectiveness and efficiency of this method in industrial robotic settings.

Keywords: Time-optimal trajectory ; Industrial robots ; Velocity sinking method ; Numerical integration.

INTRODUCTION

Time-optimal trajectory planning is of significant importance in many areas of robotics and automation, from industrial to mobile and service robotics. The theory of time-optimal trajectory planning under geometric constraints is considered to be essentially solved (Pham et al., 2018), however the efficiency and adaptability of algorithm under dynamic constraints still need to increase in practice(Barnett et al., 2020). Bobrow proposed the decoupled approach which converts multi-dimension time-optimal problems of robots to a two-dimension dynamic system (Bobrow et al., 1985). Based on this transformation, there are two methods to solve time-optimal trajectory planning problems under geometric constraints.

The first methods are indirect methods, which mainly consist of the Pontryagin Maximum Principle and Dynamic programming (Shin et al., 1985, Rojas-Quintero et al., 2022, Shin et al., 1986 & Singh et al., 1987). The time optimal trajectory solved by using the Pontryagin Maximum Principle is considered a “bang-bang” trajectory type in the plane, and can be calculated by successive integration of the maximum and minimum acceleration. Theoretically, this approach is the quickest algorithm, as it adopts the bang-bang structure of

the optimization algorithm. However, this method is difficult to use due to the involved programming difficulties and the robustness issues associated with the dynamic singularities (Kunz et al., 2012 & Pham, Q.C 2014).
The second approach consists of direct methods, which are convex optimization (Verscheure et al., 2009, Kingston et al., 2018 & Debrouwere et al., 2013) and types of swarm intelligence algorithms(Huang et al., 2018, Ertenlice et al., 2018 & Tharwat et al., 2018). Convex optimization methods discretize the *s*-axis into segments and subsequently convert the original problem into a convex optimization problem with $O(N)$ variables, $O(N)$ equality and inequality constraints. The swarm intelligence algorithm translates the time-optimal trajectory planning problem into a multi-variation optimization problem, this is similar to convex optimization methods in that both methods are able to take into account more general constraints and objective functions, such as energy or torque rate, leading to less aggressive use of the actuators. The direct method has the limitations of producing suboptimal results and has low computation efficiency.

Here, we introduce a finite, fast and widely applicable technique called Velocity Sinking Method that extends the work of Bobrow in several important respects. First, rather than use a line-search technique to find the acceleration switching points, this is very significantly since the time optimal trajectory planning problem is particularly sensitive to numerical instability near the switching points. Second, a customized numerical velocity sinking approach is used to adjust velocity. The rest of paper is organized as follows. In Section 2, the time-optimal trajectory planning problem is formulated. Then, the velocity sinking algorithm is given in Section 3. In Section 4, the experiments with real robot are used to test the performance of the velocity sinking method by comparison with other time-optimal trajectory planning methods. Conclusions and future research directions are summarized in Section 5.

PROBLEM STATEMENT USING PATH PARAMETERIZATION

The formulation of the problem is described below and the discretized path parameterization reformulation of time-optimal trajectory planning problems is described. Consider a *N-DOF* robot with dynamics equation (Zhang et al., 2021):

$$\mathbf{U} = \mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + sign(\dot{\mathbf{q}})\mathbf{F}(\mathbf{q}) + \mathbf{G}(\mathbf{q}) \tag{1}$$

where $\mathbf{D}$ is the $\mathrm{n}\times\mathrm{n}$ matrix of the robot system, $\mathbf{C}$ is the $\mathrm{n}\times\mathrm{n}$ mass matrix containing the centripetal and Coriolis terms, $\mathbf{F}$ is a $\mathrm{n}\times\mathrm{n}$ matrix of Coulomb friction torques, which can be joint angle dependent, while $\mathbf{G}$ denotes the vector accounting for gravity and other joint angle dependent torques.

Consider a path $s(t)$ in work space coordinates, the joint space coordinates $\mathbf{q}(s) \in R^n$ as a function of a scalar path coordinate $s(t):[0,T] \to [s_s, s_e]$ can be solved by robot inverse kinematics. This paper considers time-optimal path tracking problem and it is assumed that $\dot{s}(t) \geq 0$ everywhere. For the given path, differentiating $\mathbf{q}(s(t))$ with respect to *t* using the chain rule results in:

$$\dot{\mathbf{q}}(s) = \mathbf{q}'(s)\dot{s} \tag{2}$$
$$\ddot{\mathbf{q}}(s) = \mathbf{q}''(s)\dot{s}^2 + \mathbf{q}'(s)\ddot{s} \tag{3}$$

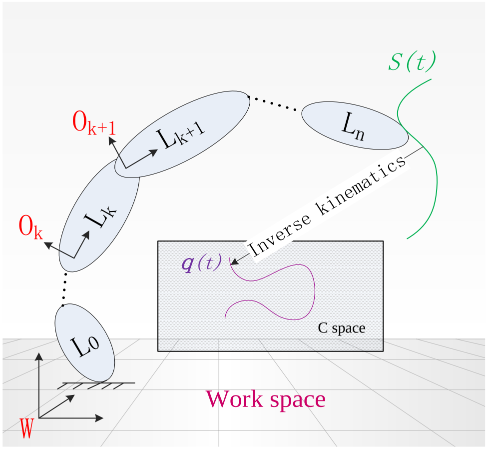

Figure 1. A schematic of time-optimal trajectory planning with path constraint.

Where dots denote differentiations with respect to the time parameter *t*, and $\mathbf{q}'(s) = d(\mathbf{q})/ds$, $\mathbf{q}''(s) = d^2(\mathbf{q})/ds^2$. Substituting (2) and (3) into (1), the industrial robot dynamics equation (3) can be transformed into:

$$\mathbf{u} = \mathbf{d}(s)\ddot{s} + \mathbf{c}(s)\dot{s}^2 + \mathbf{g}(s) + \mathbf{w}(s,\dot{s}) \tag{4}$$

Where:

$$\mathbf{d}(s)\text{=}\mathbf{D}(\mathbf{q}(s))\mathbf{q}'(s) \tag{5}$$
$$\mathbf{c}(s) = \mathbf{D}(\mathbf{q}(s))\mathbf{q}''(s) + \mathbf{q}'^T(s)\mathbf{C}(\mathbf{q}(s))\mathbf{q}'(s)\dot{s}^2 \tag{6}$$
$$\mathbf{g}(s) = \mathbf{G}(\mathbf{q}(s)) \tag{7}$$

Equation (4) constitutes an abstracted expression, which is convenient for computer implementation. It evaluates the *N* dimensional vectors $\mathbf{d}$, $\mathbf{c}$, and $\mathbf{g}$ along the path and feeds these vectors as inputs to the optimization algorithm. The complex dissipative force function $\mathbf{w}(s,\dot{s})$ is commonly replaced by linear models or ignored, similar to previously reported work(Zhang et al., 2021), the robot dynamics model was simplified to solve practical problems, and therefore the equation (4) is simplified to:

$$\mathbf{u} = \mathbf{d}(s)\ddot{s} + \mathbf{c}(s)\dot{s}^2 + \mathbf{g}(s) \tag{8}$$

Where: $\mathbf{u}$、$\mathbf{d}(s)$、$\mathbf{c}(s)$ and $\mathbf{g}(s)$ are the vectors, it can be seen from equation (8) that the robot dynamics equation is a linear differential equation about the $\ddot{s}$ and $\dot{s}^2$ on the fixed path s. Equations (3) and (8) are compared and have the same formulation, therefore:

$$\mathbf{h}(s)\text{=}\mathbf{a}(s)\ddot{s}\text{+}\mathbf{b}(s)\dot{s}^2 \tag{9}$$

VELOCITY SINGING METHOD

The scheme for the velocity sinking method of time-optimal trajectory is shown in Figure 2. The new method is based on the shooting method (Kunz et al., 2012), but it differs in that it is not necessary to solve arcs and switch points and is a finite way for time-optimal trajectory planning.

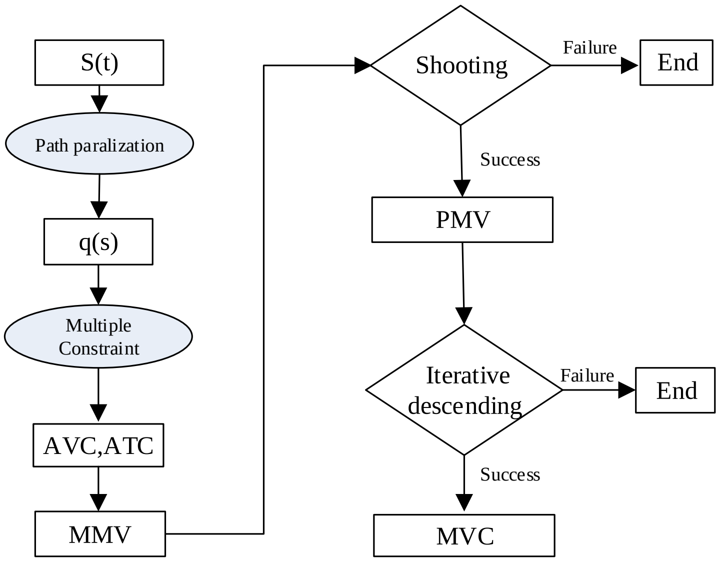

Figure 2. The scheme of VS method.

The scheme primarily consists of four parts. First, the path parameterization method makes the multi-dimensional state space of a robot reduce to a two-dimensional state space. Second, the maximum velocity in multiple state constraints (MMV) is calculated using the intersection of the AVC and the ATC. Third, the pseudo maximum velocity trajectory (PMV) is calculated using the shooting method based on MMV. Finally, an iterative descending approach translates PMV to MVC, which gives the optimal velocity curve. By using this hybrid strategy, the optimized trajectory can be improved. The scheme is valid for all time-optimal trajectory planning along specified paths.

1.1 The maximum velocity curve under multiple state constraints

The actuator velocity constraints are transformed into path velocity constraints by kinematic models in robots. Therefore, the velocity constraints are represented as：

$$|\dot{\mathbf{q}}(s)| \le \dot{\mathbf{q}}_c \tag{10}$$

Where: dots denote differentiations with respect to the time parameter t , and $\dot{\mathbf{q}}_c$ is the actuator constraint velocity. The path function $s(t)$ increases monotonically, the $\dot{s}$ is non-negative, substituting (2) into (10) results in:

$$\dot{s} \le \dot{\mathbf{q}}_c \big/ |\mathbf{q}'(s)| \tag{11}$$

The maximum velocity curve with actuator velocity constraints (AVC) can be represented as:

$$\mathrm{AVC}(s) = \min(\dot{s} \le \dot{\mathbf{q}}_c \big/ |\mathbf{q}'(s)|, s \in [s_s, s_e]) \tag{12}$$

According to the equation (12), the workspace maximum speed curve determined by actuator velocity constraints can be calculated, which is a necessary constraint condition.

To observe joint acceleration equation (3) and dynamics equation (8), torque and joint acceleration are the matrix equation about $\ddot{s}$ and $\dot{s}^2$, and the equations can be unified as:

$$\mathbf{a}(s)\ddot{s}+\mathbf{b}(s)\dot{s}^2 \le \mathbf{h}(s) \tag{13}$$

Where: $\mathbf{h}(s)$ denotes the constraint of joint torque or joint acceleration, $\mathbf{a}(s)$ and $\mathbf{b}(s)$ are the coefficients of equation (3) or equation (8), the $\mathbf{h}(s)$ is the joint extreme torque or acceleration.

In order to guarantee joint acceleration and torque constraints, the scalars s、$\dot{s}$、$\ddot{s}$ should satisfy the inequality (13). Therefore, given the path coordinate s and path velocity $\dot{s}$, the path accelerations $\ddot{s}$ satisfies the following two inequalities：

(1) If $a_i(s) > 0$, the inequality (13) can be written as：

$$\ddot{s} \le \left(h_i(s) - b_i(s)\dot{s}^2\right) \big/ a_i(s) \tag{14}$$

(2) If $a_i(s) < 0$, the inequality (13) can be written as：

$$\ddot{s} \ge \left(h_i(s) - b_i(s)\dot{s}^2\right) \big/ a_i(s) \tag{15}$$

Where: $i \in [1, N]$, with N being the degree of the robot. Solving (14) or (15) results in:

$$\ddot{s} \in \{[\underline{\ddot{s}}_i, \bar{\ddot{s}}_i] \bigcap [\underline{\ddot{s}}_{i+1}, \bar{\ddot{s}}_{i+1}], i = 1, 2\cdots, N-1\} \tag{16}$$

Where: the allowable acceleration $\ddot{s}$ is the union of minimum acceleration $\underline{\ddot{s}}_i$ and maximum acceleration $\bar{\ddot{s}}_i$. Note that the equation (16) does not have a meaningful acceleration due to the velocity in the boundary conditions, resulting in a limited velocity determined by acceleration and torque constraints. The scanning method shown in Table 1 was used to calculate the maximum velocity curve ATC constrained by acceleration and torque constrants. The central idea of the scanning method is to divide up the estimated velocities at *s*, and filter out the maximum speed according to the necessary constraint conditions as equation (16).

Table 1. The pseudo code of scanning algorithm.

Scan (ATC)
1. initialize $\dot{s}(n,m) = \text{linspace}(0, \dot{s}_{\max}(n), m)$
2. ***for*** i=1:n
for j=1:m
if $\bar{\ddot{s}}(s(i), \dot{s}(i,j) \le \underline{\ddot{s}}(s(i), \dot{s}(i,j) + \delta$
3. $y(i) \longleftarrow \dot{s}(i,j)$
4.   ***end***
5.   ***end***
6. ***end***
7. ATC $\longleftarrow y$
8. ATC

The necessary maximum speed curve ATC constrained by acceleration and torque in the phase plane can be calculated using the scanning method. Solving the union of maximum velocity curve of joint acceleration and torque constraints (ATC) and maximum velocity curve with actuator velocity constraints (AVC), the maximum velocity curve in multiple state constraints (MMV) was developed. Since the maximum velocity curve in working space is non-negative, the maximum velocity curve in the plane $(s, \dot{s})$ can be represented by:

$$\mathrm{MMV} = \min\{\mathrm{MVC}(s), \mathrm{ATC}(s), s \in [s_s, s_e]\} \tag{17}$$

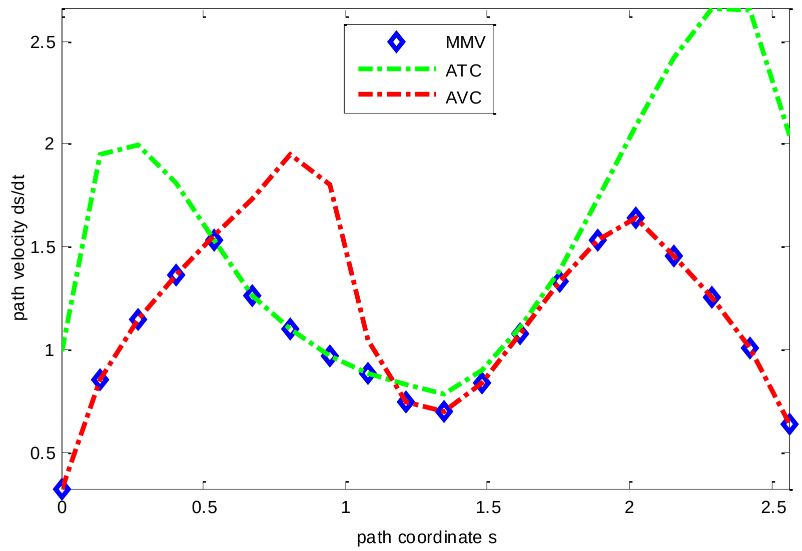

Figure 3. The schematic of three kinds of constraints speed curve.

For example, the blue diamond dash curve in Figure 3 is the MMV, the green dash curve is the ATC and the red dash line is the AVC. If the robot state is MMV, there exists at least one saturated actuator torque or actuator velocity. The MMV curve is also a trajectory of non-zero starting velocity, therefore the shooting algorithm was used to solve this.

1.2 The pseudo maximum velocity trajectory (PMV)

The maximum velocity in MMV is defined by the limited velocity curve. Therefore, using the numerical integration method, the velocity values must be maintained under MMV. The algorithm to determine the time-optimal parameterization curve of PMV, starting and ending with the zero velocities, proposes a velocity curve using forwards and backwards integrations, to get point to point trajectory as curve V1 and V2 in Figure 4.

The maximum velocity in a point to point trajectory (PMV) is the concatenation of the intersected forward profile V1, the backward profile V2 and MMV, as shown Figure 4.

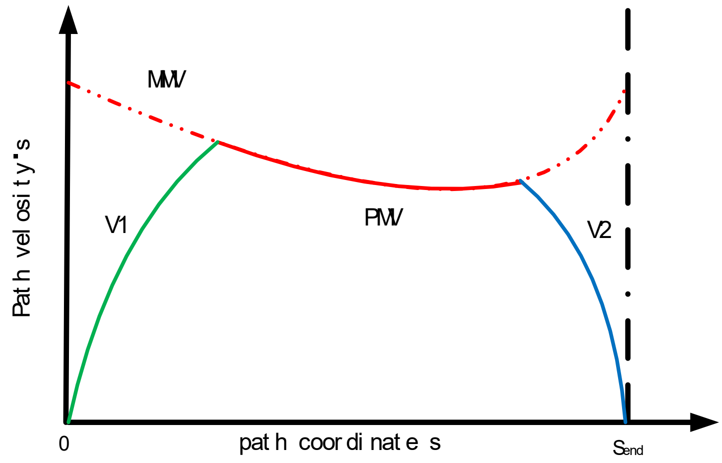

Figure 4. The constituent parts of PMV curve.

Using the successive scanning method described above and numerical integration method, the maximum velocity curve PMV can be calculated using multiple state constraints. The curve is

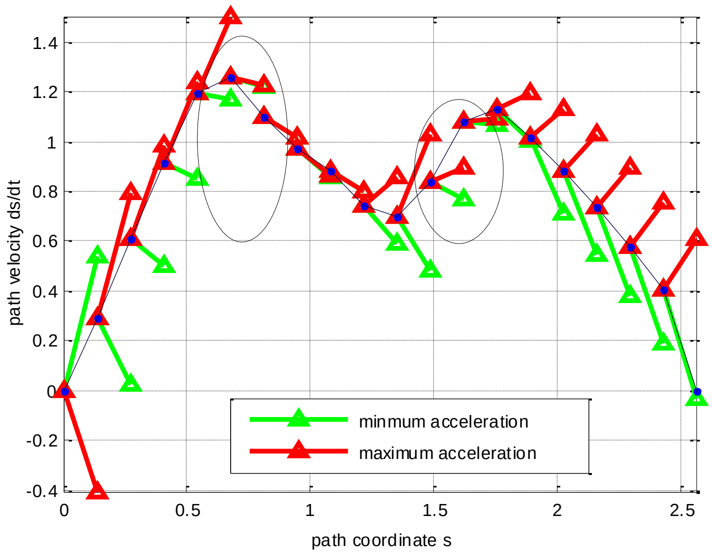

Figure 5. The velocity distribution diagram.

only necessary for real maximum velocity curve of time-optimal trajectory because the coupled robot dynamics determines that the MMV does not satisfy acceleration and torque constraints. For example, it can be found that the curve in an ellipse does not satisfy the conditions that the velocity curve is in acceleration scissors as showed in Figure 5.

1.3The iterative descending approach for velocity vector

Based on the Pontryagin Maximum Principle, numerical integrating possesses a bang-bang structure of torque inputs, thus the core of numerical integrating is the computation of switch points. Dubowsky proposed that finding multiple switching points is the most difficult part of the minimum-time problems. However, this paper proposes a method in which the time-optimal trajectory can be calculated without considering switch points, by decreasing the pseudo maximum velocity trajectory (PMV) gradually and determining the maximum velocity curve MVC in which acceleration tangent is in the acceleration scissors (see Figure 6).

The pseudo maximum velocity trajectory (PMV) is derived by calculating the maximum velocity curve (MVC) with the iterative descending method:

Step 1: According to equation (17), calculate the discrete maximum acceleration $A_{\max}$ and minimum acceleration $A_{\min}$ of the path velocity curve PMV;

Step 2: Using the directional differentiation to calculate the acceleration of PMV, which is denoted by A_p;

Step 3: Corresponding to every path coordinate *s*, find the element of acceleration vector which meet $A_p > A_{\max}$, and formulate the child vector $A_{pm} \subset A_p$ as shown by the small ellipse in Figure 5, find the element of vector $A_p < A_{\min}$, and formulate the child vector $A_{pl} \subset A_p$ as shown by the large ellipse in Figure 5.

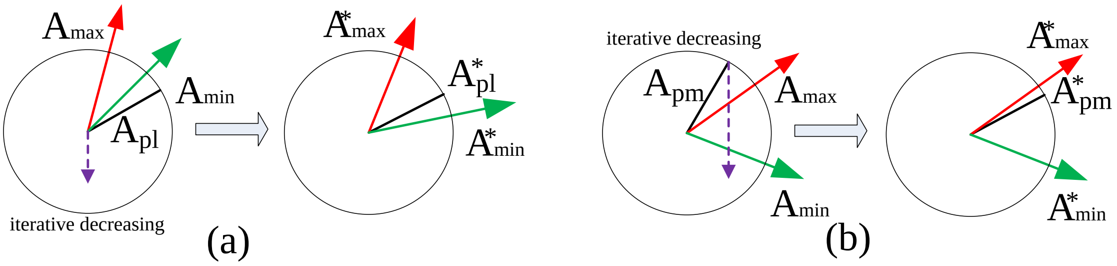

Figure 6. The velocity distribution diagram after velocity sinking approach The velocity profile in the Velocity Sinking approach ((a) represents the actual acceleration is less than the minimum constraint acceleration, (b) represents the actual acceleration is greater than the maximum constraint acceleration).

Step 4: For the acceleration vector A_{pl}, as shown in Figure 6(a), the starting point of velocity vector curve is slightly down, resulting in a new acceleration vector A_{pl}^*.

Step 5: For the acceleration vector A_{pm}, as shown in Figure 6(b), the ending point of velocity vector curve is slightly down, resulting in a new acceleration vector A_{pm}^*.

Step 6: The A_{pl}^* and A_{pm}^* constitute the new A_p. Then, repeat steps (3) to (5). If the A_{pl} and A_{pm} are all empty, it indicates that the all velocity vectors are in the acceleration scissor mouth, the velocity curve denoted by MVC must be the optimal time trajectory under multiple constraints as shown in Figure 7.

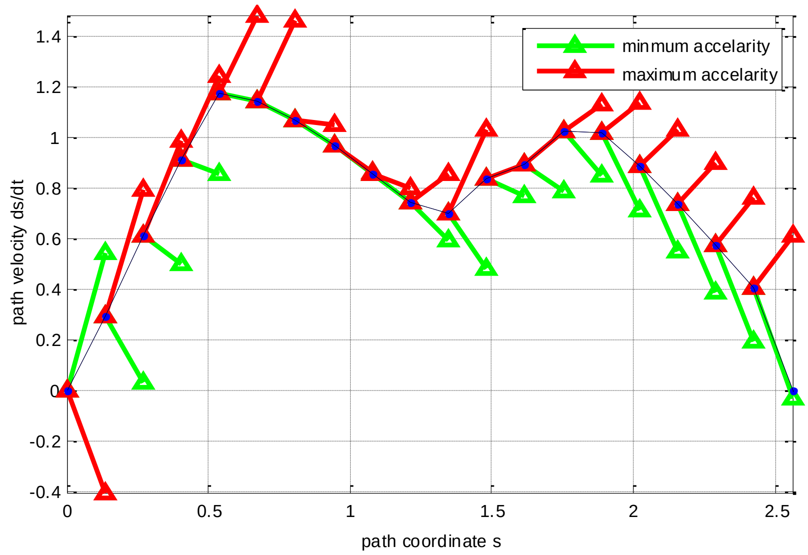

Figure 7. The velocity distribution diagram after velocity sinking approach.

After the above six steps were completed, the time-optimal trajectory curve denoted by MVC was obtained, and the proposed method to transform the N-dimensional space optimization to a deterministic problem, which is an easy and intuitive way to solve time-optimal problems and has great practical value. The following experiments test the efficiency and quality of the proposed VS method.

EXPERIMENT

The effectiveness and feasibility of the proposed VS algorithm was tested, and the presented algorithm was applied to a six-axis robot controlled by a standard industrial PC with a 2.59 GHz processor, as shown in Fig.8. For limited motor torques, a full dynamic model with identified parameters was used. The velocity and acceleration limits were taken from the manufacturer's data sheet. Linear and circular paths were studied experimentally by comparing with conventional algorithms used for the time-optimal trajectory planning problems.

1.1 Effectiveness analysis of proposed algorithm

1.1.1 Linear path simulation and experimental verification

Linear path tracking (see Figure 8), starting position coordinates (1.4507, 0.2558, 0.4899) meter, and end position coordinates (-0.3334, -0.9160, -0.1551) meter were used to calculate inverse kinematic transformation of straight line in Cartesian space with position of the tool center. A MATLAB-implementation of the presented algorithm was used to compare the results with the genetic and SQP algorithms (Abd Elrehim et al., 2019 & Liu et al., 2013). The resulting path velocity profiles are shown in Figure 9.

Figure 8. The straight path schematic of industrial robot.

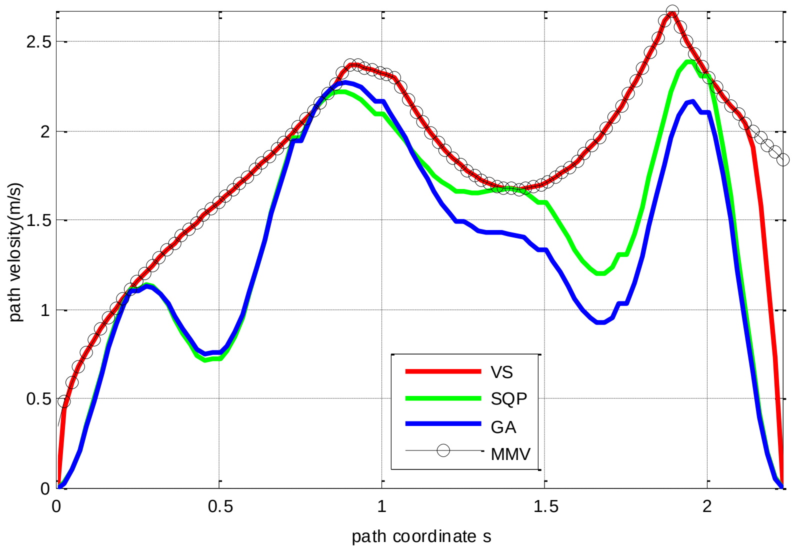

Figure 9. The path velocity of different planning method.

As shown in Figure 9, the comparison of three methods indicates that the three velocity profiles were all under the MMV denoted by the blank circle curve. The curves of the GA and SQP algorithms were smoothly, the two curves had more oscillations than the curve of the VS method. This suggests that the trajectory based on the VS algorithm takes less time compared to the other two trajectories.

Figure 10 shows the joint acceleration results of the three algorithms, the limit acceleration of three axis was (36.7544,27.4534,32.3333) rad/s2, the accelerations of joint 1 and joint 2 were less than its limit acceleration with all three algorithms. However, the acceleration of joint 3 with VS algorithm equals its limit acceleration in some parts of the curve, indicating that the acceleration using the VS algorithm was greater than that of the other two algorithms.

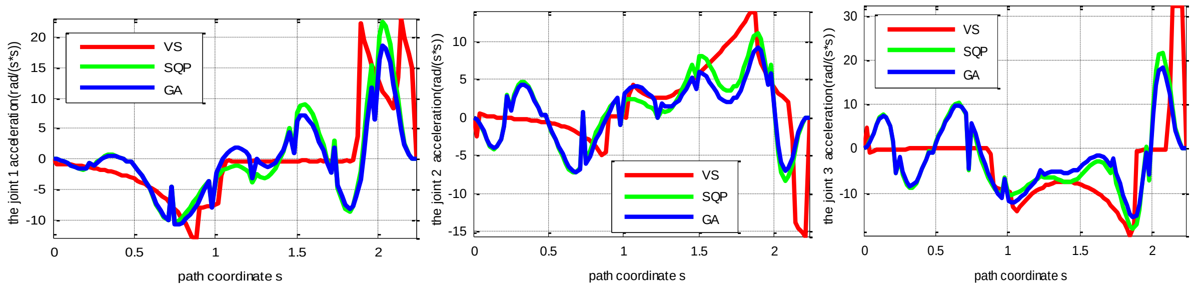

Figure 10. Joint acceleration diagram.

As shown in Figure 11, the torque curve of the VS algorithm changed significantly, but the torque variation with genetic algorithms (GA) and sequential quadratic programming algorithm (SQP) was relatively stable, suggesting that the VS method had the"bang-bang" features for multiple constraints. The acceleration curves of the three algorithms were all below the limit acceleration curve, which was 400 N.m. Therefore, the torque of the VS method satisfied the constraint condition and was compliant with planning requirements.

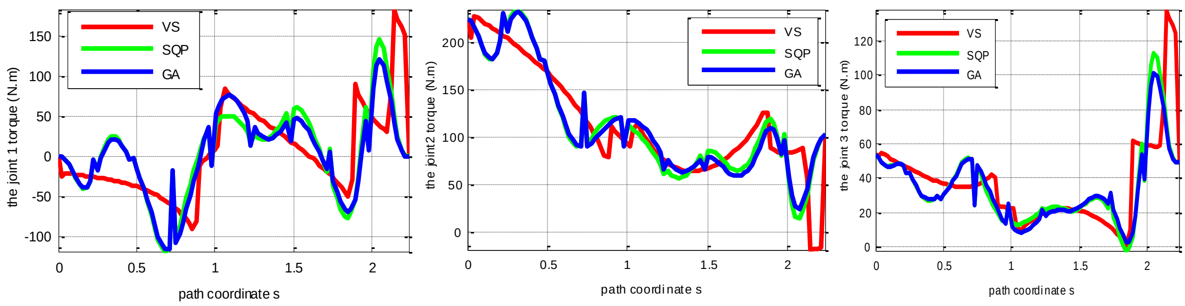

Figure 11. Joint pseudo torque comparison chart.

The optimal time of the three algorithms is shown in Figure 12. The optimal values of the VS algorithm at different positions were less than that of the other two algorithms. This was due

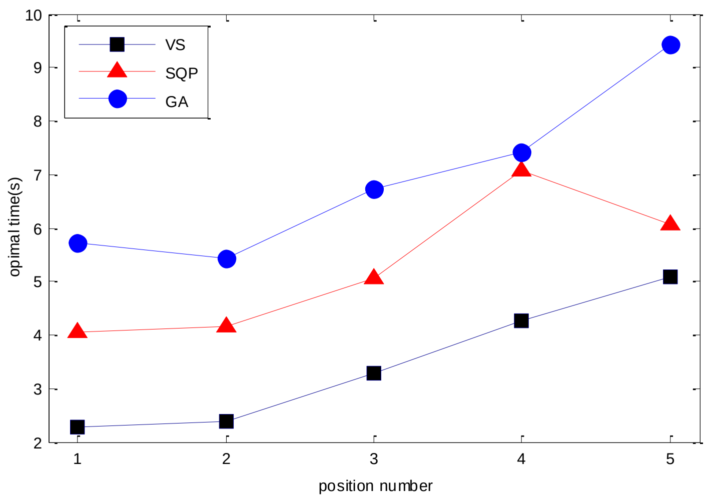

Figure 12. The optimal time comparison.

to the finite optimization used for the VS algorithm which was derived from the multiple trajectory constraints, while the other two algorithms were susceptible to local minima. It was difficult to obtain a global optimum. Therefore, the proposed method had a minimum time, and therefore better compared to the other two algorithms for the optimization results.

1.1.2 Circular path simulation and experimental verification

This example is a time-optimal trajectory planning problem for a circular path located on a horizontal surface, whose center coordinates were (0.80, 0.10, 0.36) meters and the radius was 0.08 meters. The trajectory of the VS methods is shown in Figure 13. The green curves are the iterative sequences of the velocity profile using the VS method, the red curve represents the time-optimal trajectory profile, and the black circle curve represents the pseudo maximum velocity trajectory (PMV). The results indicate that the PMV turns into MVC after 7 iterations and the minimum time velocity curve MVC was determined. A detailed comparison is shown in Figure 14.

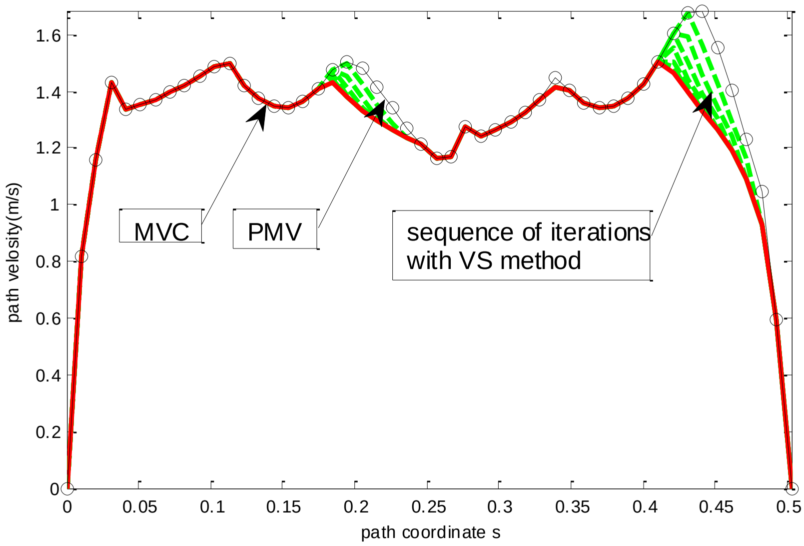

Figure 13. The schematic of VS method.

The results in Figure 14 show the performance of different planning methods. The results suggest that VS had the more efficient velocity curve compared to GA and SQP, possibly due

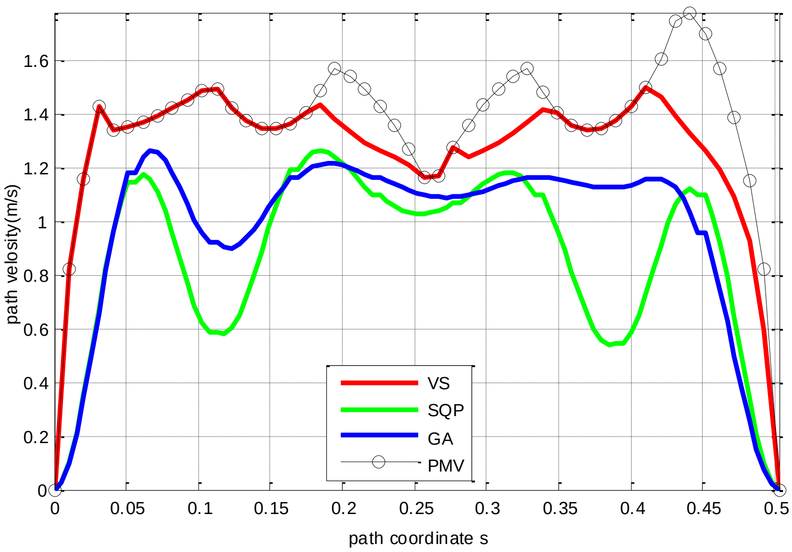

Figure 14. Performance of different methods in time-optimal trajectory planning.

to the VS method having a finite optimization algorithm, but the SQP and GA were stochastic optimistic algorithms which may fall into local minimum. The proposed method can get better optimum results compared to the other two algorithms.

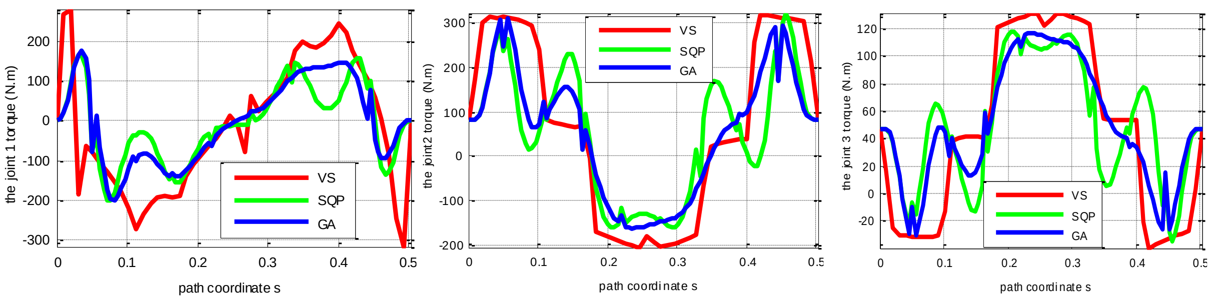

Figure 15. Comparison of joint torque.

Figure 15 shows a comparison chart of joint torques. The torques calculated using the three algorithms were all in scope of the joint limit torque, which was 400 Nm. The torque variation of VS was flatter than that of SQP and GA for joint 2 and joint 3. However, the joint 1 torque using the VS method had a small jitter, which was due to the sudden change in acceleration. Figure 14 indicates that VS algorithm satisfies torque constraints for computing time-optimal trajectory planning.

1.2 Efficiency analysis of proposed algorithm

Trajectory planning time is an important indicator in time-optimal planning, improving computational efficiency. With the development of computing and storage technology, the VS algorithm can be applied to industrial robots. Table 3 shows the computational efficiency of the VS algorithm, sequential quadratic programming algorithm (SQP) and genetic algorithm (GA).

Table 2. The Comparison of optimization algorithms.

Item	VS	SQP	GA
CPU time for line path (s)	0.24	0.75	1.71
CPU time for circle path (s)	0.87	1.32	1.85
Num. points	50	50	50

The VS is the presented algorithm, SQP is a gradient optimization method and GA is an evolutionary algorithm. Num. points are the number of discrete points. CPU time is the planning time for the whole path using MATLAB. Although computation times have to be interpreted carefully, the comparison suggests that the proposed method could be efficient compared to the other two algorithms for line path and circle path. Furthermore, the circular path consumes more CPU time when compared to the line path. Therefore, the VS algorithm has the advantages of being time efficient and calculating the optimum time for a time-optimal trajectory plan problem.

CONCLUSION

This paper presents a velocity sinking (VS) method, a finite approach for time-optimal trajectory planning along specified paths with dynamics constraint. The pseudo maximum velocity trajectory was established on the path coordinate, which was calculated using the constraint equation and shooting method. To determine the time-optimal trajectory, VS used a phase plane method to outline the distribution features of acceleration. Moreover, discrete velocity vector violating the acceleration principle were ajusted to conform to the acceleration laws, which could result in the velocity distribution being closer to the real maximum velocity goal. Comparisons with other time-optimal trajectory planning methods showed statistically significant improvements in optimum and computation efficiency.

Future work should include the investigation of the issue of dynamic singularities using the VS method to expend algorithm adaptability. The VS method would also benefit from dynamic singularities. The application of the VS method in industrial robots involving handling tasks, as well as mobile robots involving time-optimal goal could be all advantageous.

ACKNOWLEDGEMENTS

Support of the Natural Science Foundation of Gansu Province under Grant 17JR5RA111 is gratefully acknowledged. The authors would like to thank the editor for their important contribution to this work. The authors would also like to thank the anonymous reviewers for their very useful comments.

REFERENCES

Pham, H. & Pham, Q. C. 2018. A new approach to time-optimal path parameterization based on reachability analysis. IEEE Transactions on Robotics. 34(3)645-659.

Barnett, E. & Gosselin, C. 2020. A bisection algorithm for time-optimal trajectory planning along fully specified paths. IEEE Transactions on Robotics. 37(1)131-145.

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.

Shin, K. & Mckay, N. 1985. Minimum-time control of robotic manipulators with geometric path constraints. IEEE Transactions on Automatic Control. 30(6)531–541.

Rojas-Quintero, J.A., Dubois, F. & Ramírez-De-Ávila, H.C. 2022. Riemannian Formulation of Pontryagin's Maximum Principle for the Optimal Control of Robotic Manipulators. Mathematics. 10(7)1117-1128.

Shin, K. & McKay, N. 1986. A dynamic programming approach to trajectory planning of robotic manipulators. IEEE Transactions on Automatic Control. 31(6)491-500.

Singh, S. & Leu, M. 1987. Optimal trajectory generation for robotic manipulators using dynamic programming. Journal of Dynamic Systems, Measurement, and Control. 109(2)88-96.

Pham, Q.C. 2014. A general, fast, and robust implementation of the time-optimal path parameterization algorithm. IEEE Transactions on Robotics. 30(6) 1533-1540.

Verscheure, D., Demeulenaere, B., Swevers J., Schutter, J.D. & Diehl, M. 2009. Time-optimal path tracking for robots: A convex optimization approach. IEEE Transactions on Automatic Control. 54(10) 2318-2327.

Kingston, Z., Moll, M. & Kavraki, L.E. 2018. Sampling-based methods for motion planning with constraints. Annual review of control, robotics, and autonomous systems 1(1)159-185.

Debrouwere, F., Loock, W.V., Pipeleers G , Dinh, Q.T., Diehl, M., Schutter, J.D. & Jan Swevers, J. 2013. Time-Optimal Path Following for Robots With Convex–Concave Constraints Using Sequential Convex Programming. IEEE Transactions on Robotics. 29(6)1485-1495.

Jin, Q.B. & Huang, J. 2018. The solar-tracking optimal trajectory planning research based on minimum energy consumption in SSPS. Aerospace Science & Technology. 76(5)272-279.

Ertenlice, O. & Kalayci, C.B. 2017. A survey of swarm intelligence for portfolio optimization: Algorithms and applications. Swarm & Evolutionary Computation. 33(1)1-17.

Tharwat, A., Elhoseny, M., Hassanien, A. E., Gabel, T. & Kumar, A. 2019. Intelligent Bézier curve-based path planning model using Chaotic Particle Swarm Optimization algorithm. Cluster Computing. 22(4)1-22.

Zhang, L., Wang, Y., Zhao, X., Zhao, P. & He, L. 2021. Time-optimal trajectory planning of serial manipulator based on adaptive cuckoo search algorithm. Journal of Mechanical Science and Technology. 35 (7)3171-3181.

Zhang, T., Zhang, M. & Zou, Y. 2021. Time-optimal and smooth trajectory planning for robot manipulators. International Journal of Control, Automation and Systems. 19(1)521-531.

Abd Elrehim, M. Z. & Eid, M. A. 2019. Sayed M G. Structural optimization of concrete arch bridges using Genetic Algorithms. Ain Shams Engineering Journal. 10(3) 507-516.

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.