A Time-Optimal Trajectory Algorithm Based on Accessibility Analysis

Minxiu Kong[1,2], Han Deng[1,2], Xiaopeng Liu[1,2]
[1] School of Mechatronics Engineering, Harbin Institute of Technology
[2] State Key Laboratory of Robotics and System
Harbin, China
hanniel_deng@163.com

2021 5th International Conference on Robotics and Automation Sciences (ICRAS) | 978-1-6654-1226-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRAS52289.2021.9476280

***Abstract*—In order to make industrial robots move faster, more accurately and more safely along a predetermined path [1], a time-optimal trajectory algorithm based on accessibility analysis is designed in this paper. This algorithm adopts a widely used path-velocity decomposition framework [2], introduces path parameters, changes the velocity and acceleration of the trajectory by changing the path parameters, and then completes the planning in Cartesian space. This paper discusses the problem of the number of interpolation points in the path [3], and puts forward the principle of coarsening the interpolation points according to the length of the path, and then supplementing the interpolation points according to the curvature and the planned acceleration. Make interpolation point distribution more reasonable. Then, the positive and negative search algorithm is proposed to determine the deceleration point, which greatly improves the efficiency compared with the traditional iterative search algorithm [4] and the method of finding the switching point [5]. Finally, the program of the straight line, arc, and spline curve is written to verify the effect of the algorithm.**

Keywords*—*robot dynamics, accessibility analysis, time-optimal, interpolation, trajectory planning

I. Introduction

Traditional trajectory planning is based on kinematics constraints and does not consider dynamic constraints. Therefore, there are two problems in the traditional trajectory planning. The first is that the planning speed may exceed the restricted area, and the second is that the driving performance of the joint motor can not be fully brought into play. For Cartesian coordinate robots and machine tools, the traditional trajectory planning is time-optimal, but for robots, the trajectory is not time-optimal. The corresponding constraints of the robot are different at different interpolation points, so it is necessary to analyze the constraints of each interpolation point [6], find the optimal value under the corresponding constraints, and finally the optimal trajectory is a curve. The time-optimal trajectory can be regarded as three segments, the starting segment is accelerated at the maximum acceleration of each point, the middle segment is close to the upper limit of the feasible region for limit operation, and the end segment is decelerated with the minimum acceleration.

In order to overcome the shortcomings of traditional trajectory planning, a time-optimal trajectory algorithm based on accessibility analysis is proposed in this paper, considering the dynamic constraints. The specific algorithm is shown in Fig. 1.The whole algorithm flow consists of two parts, a total of 8 steps. The first part is path parameterization, including the determination of path and interpolation point correlation parameters, which is the basis of the whole trajectory planning. the second part is to generate trajectories, including the determination of reachable sets, the determination of feasible domains, the adjustment of interpolation points, filtering, isochronous interpolation, and the output of joint parameters.

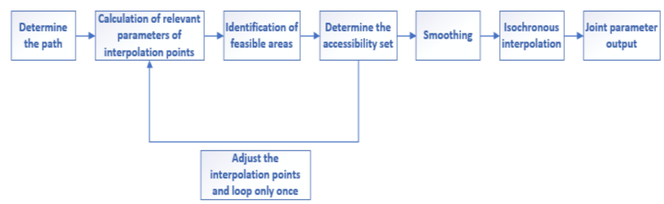


Fig. 1. Algorithm flow chart

II. Path Parameterization

A. Determine the Path

Path determination can be determined by path planning or manual teaching, both of which are involved in motion planning. However, compared with the two, the path planning is more complex, but the manual teaching is relatively simple, and the algorithm focuses on the trajectory planning, so the teaching method is adopted. If the path is a straight line, two points need to be given, if the path is an arc, three points need to be given, and if the path is a spline curve, multiple points need to be given.

B. Determination of Relevant Parameters of Interpolation Points

The determination of interpolation point correlation parameters is the basis of the whole algorithm, including the number of interpolation points, Cartesian spatial correlation parameters (position, attitude, line / angular velocity, line / angular acceleration), joint spatial correlation parameters (joint angle, joint velocity, joint acceleration) and kinetic equation coefficient [3].

The number of interpolation points is not only related to the calculation efficiency, but also to the accuracy of path shape. There are no absolute conclusions and principles on how to determine the number of interpolation points and how much is appropriate. At present, the general view is that the interpolation points should not be obtained too much or too little, to ensure that the calculation accuracy is sufficient, at the same time, the number of points is as few as possible, and the calculation efficiency is improved. Therefore, at the beginning of planning, only the path does not have the acceleration and curvature of each point. First, the interpolation points are coarsened according to the principle of equal distance. The number of interpolation points is:

$$N_1 = \left[k_1 \times L + 0.5 \right] \tag{1}$$

where: k_1 is coefficient; L is path length.

Taking the arc as an example, the spatial three-dimensional diagram after isometric interpolation is shown in Fig. 2.

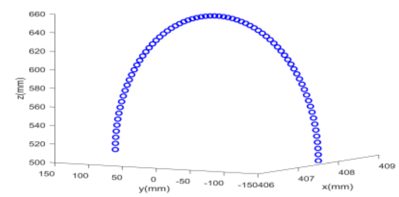


Fig. 2. Spatial three-dimensional after coarsening interpolation points

After the number of interpolation points is preliminarily determined, the Cartesian spatial correlation parameters, joint spatial correlation parameters and kinetic equation coefficients are calculated for straight line, arc and spline curve paths, respectively. Due to the limited space, the calculation process is not explained in detail here.

III. Generate Trajectories

A. Determine the Feasible Areas

The so-called feasible region is the range of parameter velocity corresponding to each interpolation point at the end of the robot. The minimum and maximum values of parameter velocity are required to obtain a feasible domain. The corresponding constraints of the robot are also different at different positions and attitudes. constraints include cartesian spatial constraints, and joint spatial constraints. cartesian spatial constraints include spatial line/angle velocity constraints, spatial line/angle acceleration constraints [7]. Joint spatial constraints include joint torque constraints, joint velocity constraints and joint acceleration constraints. These constraints can be transformed into formulas containing parameter velocity and parameter acceleration. Considering these constraints, the maximum value of the velocity square of each interpolation point is obtained point by point. By connecting these numerical points, a fluctuating curve can be obtained, as shown in Fig. 3. This curve corresponds to the upper limit of the feasible domain, and the lower limit is 0.

Fig. 3. Feasible regional map

B. Determine the Accessibility Set

The feasible region of the upper part is only a range. to know the actual parameter velocity of each interpolation point, it is necessary to solve the reachable set when the constraint is satisfied. as shown in Fig. 4, to obtain the time-optimal trajectory, the starting stage (section ab) needs to be accelerated with maximum parameter acceleration, the intermediate stage (section bc) needs to run close to the upper bound of the feasible domain, and the end-stage (section cd) decelerates with minimum parameter acceleration.

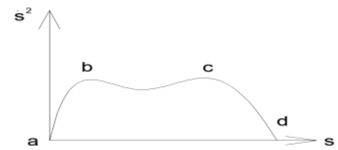


Fig. 4. Schematic diagram of time-optimal trajectory

At the beginning of the motion, the speed is 0; at the end of the motion, the speed is also 0. When the starting segment is connected to the middle segment, when the end segment begins to slow down, and how the middle segment is connected to the end segment, need to be clear when solving the reachable set. The traditional algorithm determines the deceleration point through multiple iterative searches and then determines the reachable set. In this paper, the deceleration point is determined quickly by the forward-reverse search method, and the reachable set can be determined by two calculations. First, the forward calculation is carried out. Considering the dynamic and kinematic constraints, the acceleration is:

$$\begin{cases} u_1 = \dfrac{\tau_{\max}}{|a_1|} + \dfrac{-a_3 - a_2 \cdot \dot{s}^2}{a_1} \\ u_2 = \dfrac{\dot{s}_{\max i}{}^2 - \dot{s}_{i-1}{}^2}{2\Delta s} \end{cases} \tag{2}$$

where: a_1, a_2, a_3 is kinetic equation coefficient; $\dot{s}_{\max i}{}^2$ is the square of the maximum parameter velocity corresponding to the latter interpolation point; $\tau_{\max}$ is maximum joint torque.

So, from the kinematics formula of physics, the square of the velocity of the latter interpolation point is:

$$\dot{s}_i{}^2 = \dot{s}_{i-1}{}^2 + 2u \cdot \Delta s \tag{3}$$

where: u take the smaller value in (2).

Because the initial velocity is 0, the velocity of the second interpolation point can be obtained from formula 3, and so on, until the velocity of the last point is obtained. Connect the obtained values, as shown in Fig. 5. Obviously, the final velocity obtained from the forward calculation is not 0. Therefore, there is also a need for the reverse search.

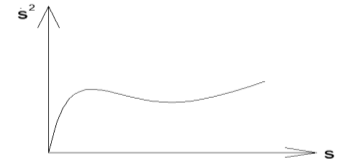


Fig. 5. Schematic diagram of trajectory obtained by the forward search

In reverse calculation, considering the dynamic and kinematic constraints, the acceleration of the parameters is as follows:

$$\begin{cases} u_3 = k\left(\dfrac{\tau_{\min}}{|a_1|} + \dfrac{-a_3 - a_2 \cdot \dot{s}^2}{a_1}\right) \\ u_4 = \dfrac{\dot{s}_{\max(i-1)}{}^2 - \dot{s}_i{}^2}{2\triangle s} \end{cases} \tag{4}$$

where: $\dot{s}_{\max(i-1)}{}^2$ is the square of the maximum parameter velocity corresponding to the previous interpolation point; $\tau_{\min}$ is minimum joint torque; k is adjustment coefficient, the range is 0-1, the smaller the value of the k, the more gentle the last tail deceleration.

So, from the kinematics formula of physics, the square of the velocity of the previous interpolation point is:

$$\dot{s}_{i\text{-}1}{}^2 = \dot{s}_i{}^2 + 2u \cdot \Delta s \tag{5}$$

where: u take the larger value in (4).

Because the velocity at the end is 0, the velocity of the penultimate interpolation point can be obtained from formula 5, and so on, until the velocity of the first point is obtained. Connect the obtained values, as shown in Fig. 6.

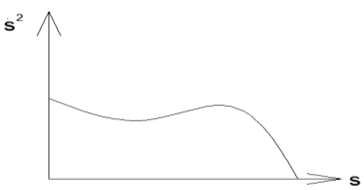


Fig. 6. Diagram of trajectory obtained by reverse search

Considering the results of the forward and reverse calculation, the starting and middle segments are calculated by the forward search, and the last segment is calculated by the reverse search, and the optimal trajectory of time is obtained, similar to Fig. 4.

C. Supplementary Interpolation

Taking the arc as an example, after the previous calculation, the feasible region and reachable set diagram are shown in Fig. 7.

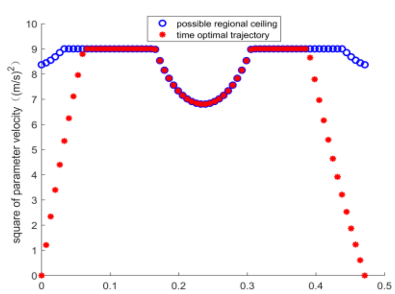


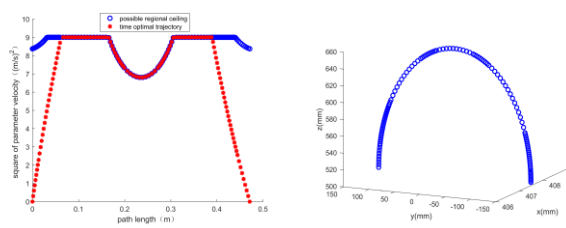
Fig. 7. Schematic illustration of feasible areas and accessible sets

As can be seen from the above figure, most of the interpolation points are concentrated in the middle segment, and the interpolation points of the starting and end segments are few. Obviously, this is not reasonable, because the acceleration of the starting and end segments is large and the velocity changes rapidly, the interpolation points should be taken more, the acceleration of the middle segment is small, and the interpolation points should be taken less. Also, for spline curves, the larger the curvature, the more dense the distribution of interpolation points should be, and the smaller the curvature, the sparse distribution of interpolation points should be. Therefore, the existing interpolation points should be supplemented according to the curvature and acceleration of each interpolation point. For straight lines and arcs, the curvature of each point is the same, focusing on acceleration. The number of additional interpolation points is:

$$N_2 = \left[k_2 \times U + 0.5\right] \tag{6}$$

where: k_2 is coefficient; U is parameter acceleration.

For spline curves, both acceleration and curvature need to be considered, and the formula is similar. Supplementary interpolation is performed on Fig. 7, and the resulting image is shown in Fig. 8.



(a) (b)

Fig. 8. Image obtained by fine-tuning interpolation points: (a) feasible area and accessible set image; (b) spatial 3D map.

D. Smoothing

There are sharp points in some places, especially at the junction of three segments. In addition, consider reducing the impact of the robot at the initial and final points. Therefore, it is necessary to filter the output parameter speed to make the image smooth. Because the position of each interpolation point is calculated, these positions can not be changed when filtering. Therefore, the filtering of this algorithm can not use interpolation or fitting function, but can only adjust the speed of existing interpolation points. In this case, the applicable algorithms are moving average filtering and second-order low-pass filtering. Among them, the moving average filtering method is relatively simple, fixed one queue length (the queue length in this paper is 4), and the average processing of multiple points is carried out. In order to achieve better results, this paper improves the moving average filtering method. In addition, because the final velocity is 0, and if the whole segment adopts the same filtering method, the velocity of the last point after filtering is certainly not 0. Therefore, each path is divided into two sections before and after processing. First, find the deceleration point c, then the front and rear segments are handled in different ways.

E. Isochronous Interpolation

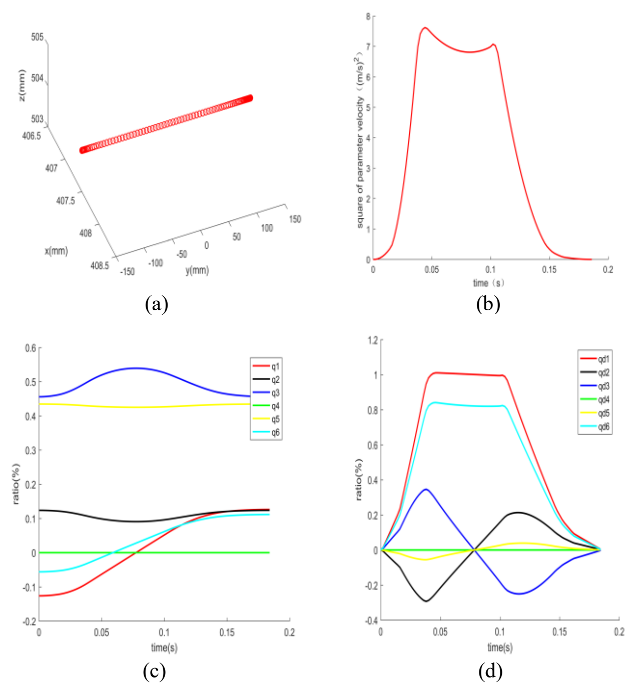
There are two common ways of interpolation, one is an isometric interpolation, the other is an isochronous interpolation. In the beginning, the specific information of velocity and acceleration of each interpolation point is not known, so equidistant interpolation is used first. After the reachable set is obtained by the planning algorithm, the specific information of velocity and acceleration of each interpolation point is obtained. However, the velocity and acceleration obtained here are s with path parameters, not real velocity and acceleration. At this time, we calculate the path position s sequence value of each cycle according to the interpolation period.

F. Joint Parameter Output

After the path position of each interpolation point is obtained, the angle value of each joint is obtained after the inverse solution. Further using the difference method, the velocity value and acceleration value of each joint can be obtained.

IV. SIMULATION EXPERIMENT OF TIME-OPTIMAL TRAJECTORY PLANNING

Since complex paths are composed of simple paths such as straight lines, arcs, and spline curves, so this paper only verifies the simple path. First, the line path is verified. The QJR6S-1 robot is chosen in this paper. Given two teaching points, the starting pose is (407.5, -150, 504, 180, 0, 0), and the final pose is (407.5, 150, 504, 180, 0, 20). Among them, the first three coordinates represent the position in millimeters, and the last three coordinates represent the attitude with Euler angles in degrees. Set the constraint value of the related parameters and complete the related calculation according to the algorithm idea. The resulting simulation image is shown in Fig. 9.



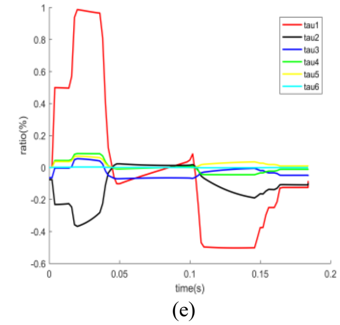


Fig. 9. Simulation image of single line: (a) 3D spatial diagram; (b) time-optimal trajectory; (c) articular angle ratio map; (d) joint velocity ratio map; (e) joint moment ratio map.

Then, the arc path is verified. Giving three teaching points, the initial pose is (407.5, -150, 504, 180, 0, 0), the middle pose is (407.5, 0, 654, 180, 0, 10), and the terminal pose is (407.5, 150, 504, 180, 0, 20). The simulation image is shown in Fig. 10.

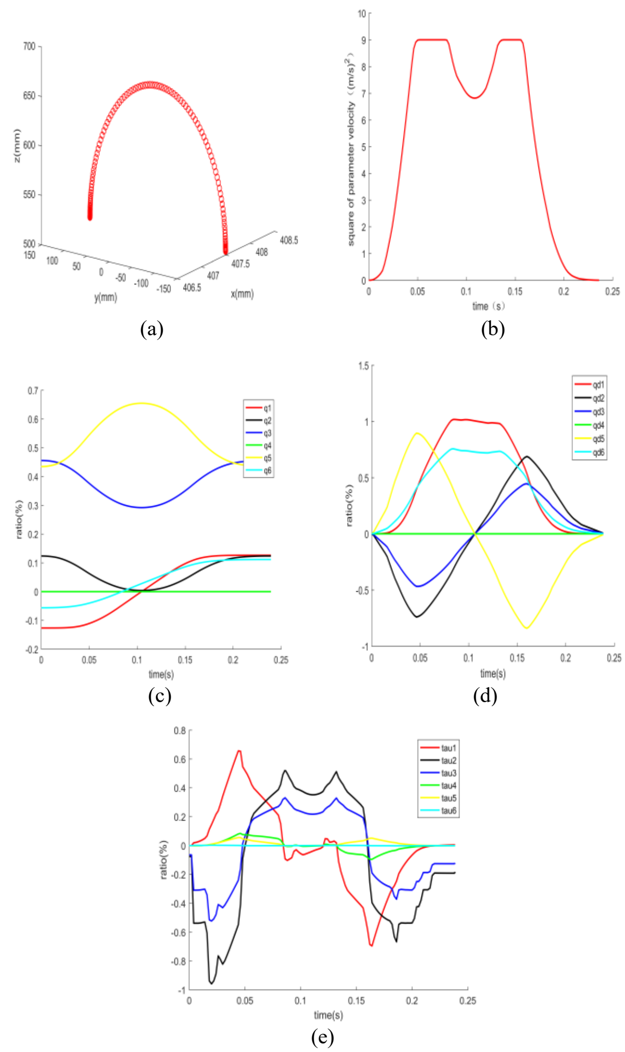


Fig. 10. Simulation image of arc: (a) 3D spatial diagram; (b) time-optimal trajectory; (c) articular angle ratio map of each joint; (d) joint velocity ratio map of each joint; (e) joint moment ratio map of each joint.

Finally, the spline curve path is verified. Giving three teaching points, the initial pose is (407.5, -150, 504, 180, 0, 0), the middle pose is (407.5, 0, 504, 180, 0, 10), and the terminal pose is (407.5, 0, 354, 180, 0, 20). The simulation image is shown in Fig. 11.

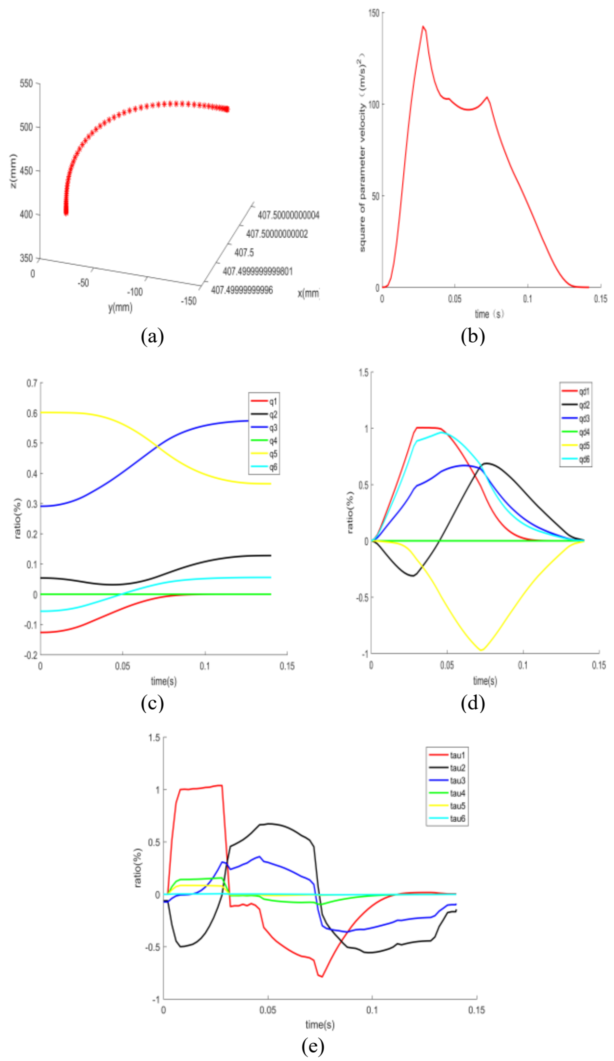


Fig. 11. Simulation image of spline curve: (a) 3D spatial diagram; (b) time-optimal trajectory; (c) articular angle ratio map of each joint; (d) joint velocity ratio map of each joint; (e) joint moment ratio map of each joint.

As can be seen from Fig. 9-11(b), Fig. 10(b) and Fig. 11(b), the time-optimal trajectory is relatively smooth, the starting section accelerates with maximum acceleration, the middle section runs close to the upper limit of the feasible region, and the deceleration section decelerates in an asymmetric manner with acceleration. Reduce the vibration at the end of the robot motion. Compared with the traditional kinematics trajectory planning, this algorithm has great advantages in time. As can be seen from Fig. 9(c-e), Fig. 10(c-e) and Fig. 11(c-e), the ratio values of joint parameters are between -1 and 1, so the joint parameters are running in a safe range. In summary, the feasibility and superiority of the algorithm are tested.

V. CONCLUSIONS

In this paper, the value of the number of interpolation points is discussed, and the corresponding principles are put forward to make the distribution of interpolation points more reasonable. In addition, a positive and negative search algorithm is proposed to determine the reachable set, which improves computational efficiency. Then, the program is written for simulation experiments, and the feasibility and superiority of the time-optimal trajectory algorithm are verified based on the simple path. Subsequently, we will continue to optimize the time-optimal trajectory algorithm to further verify whether it is feasible on complex paths. In addition, it is necessary to explain whether the trajectory obtained by this algorithm or the expected trajectory is ideal. There are still differences between the actual trajectory and the ideal trajectory, and the subsequent position detection and research are needed, which is a key aspect of trajectory planning [8].

ACKNOWLEDGMENT

This work is supported by the National Key R&D Projects of China (Project Number: 2018YFB1308100).

REFERENCES

[1] Member, Croft, E. A. "Trajectory Planning for Robots: the Challenges of Industrial Considerations.".

[2] KAMAL KENT, ZUCKER SW. Toward efficient trajectory planning: the path-velocity decomposition. The International journal of robotics research. 1986; 5(3):72-89. Accessed April 8, 2021.

[3] R. Wang, M. Kong, Y. Cheng and M. Castellani, "Time-Optimal Trajectory Planning Based on Dynamics for Industrial Robot," 2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE), Dalian, China, 2020, pp. 53-57, doi: 10.1109/CACRE50138.2020.9229917.

[4] Bobrow, J. E., S. Dubowsky, and J. S. Gibson. "Time-Optimal Control of Robotic Manipulators Along Specified Paths." Int. J. of Robotic Research 4.3(1985):3-17.

[5] Kang, Shin, McKay. "Minimum-time control of robotic manipulators with geometric path constraints." Automatic Control IEEE Transactions on (1985).

[6] H. Pham and Q. Pham, "A New Approach to Time-Optimal Path Parameterization Based on Reachability Analysis," in IEEE Transactions on Robotics, vol. 34, no. 3, pp. 645-659, June 2018, doi: 10.1109/TRO.2018.2819195.

[7] F Debrouwer e, Loock W V, Pipeleers G. "Convex Time-Optimal Robot Path Following with Cartesian Acceleration and Inertial Force and Torque Constraints." Proceedings of the Institution of Mechanical Engineers Part I Journal of Systems & Control Engineering 227.I10(2013):724-732.

[8] N. Thakur and C. Y. Han, "Multimodal Approaches for Indoor Localization for Ambient Assisted Living in Smart Homes," Information, vol. 12, no. 3, p. 114, Mar. 2021.