Admissible Trajectory Planning for Wheeled Mobile Robot

Nelli Korotkova
Saint-Petersburg State University
7-9, Universitetskaya nab.,
St. Petersburg, 199034, Russia
nelly_kor@mail.ru

Evgeny Veremey
Saint-Petersburg State University
7-9, Universitetskaya nab.,
St. Petersburg, 199034, Russia
e_veremey@mail.ru

ABSTRACT

In this paper the problem of planning a trajectory for driving a mobile robot from some initial position to an arbitrary goal that satisfies limitations of control signals is considered. For this purpose continuous-curvature turns are used within the Dubins's scheme. It is also offered to use presented trajectory planning algorithm along with a stabilizing feedback.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*control theory*; I.2.9 [**Artificial Intelligence**]: Robotics

General Terms

Theory

Keywords

Mobile robot, trajectory planning

1. INTRODUCTION

Wheeled mobile robots are subjects to nonholonomic constraints. It means that they need to perform some specific maneuver in order to reach an arbitrary configuration. Planning an admissible trajectory is a common approach to solving the task of finding such maneuver. Some physical constraints are also present and apply further limitations on movement of the system.

The first goal of this paper is to design a method of planning a path considering all constraints including acceleration bounds. Particularly the problem of meeting linear acceleration bounds is solved by adding straight segments to a path. This problem is not addressed in previous works. The second goal is to describe a way of building corresponding admissible control signals. Thus we implicitly associate path as a geometric curve with a timing law and make it a trajectory. Also the application of the described method along with a stabilizing feedback is presented.

IWAIT '15, Oct. 8 – 10, 2015, Aizu-Wakamatsu, Japan.

2. PROBLEM STATEMENT

A unicycle-type mobile robot can be represented with a following system:

$$\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{pmatrix} = \begin{pmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}v\\ \omega\end{pmatrix} \tag{1}$$

Its position is described with three values where (x, y) are plane coordinates and θ is orientation. v and ω are inputs, linear and angular velocities respectively. Due to mentioned above physical constraints their values are limited as well as values of their derivatives, linear and angular accelerations:

$$\begin{gathered} |v| \leqslant v_{\max},\ |\omega| \leqslant \omega_{\max},\\ |\dot{v}| = |a| \leqslant a_{\max},\ |\dot{\omega}| = |\varepsilon| \leqslant \varepsilon_{\max} \end{gathered} \tag{2}$$

The aim is to build control signals able to drive a robot from an initial position $q_s = (x_s,\ y_s,\ \theta_s)^T$ to an arbitrary goal q_g.

3. TRAJECTORY PLANNING

Let us solve the trajectory planning task by planning a path as a plane geometric curve and simultaneously building admissible control laws that realize motion along this path.

As it is commonly done, let us simplify the path planning task by assuming that the robot moves with a constant maximum speed the whole way. In order to satisfy linear acceleration limit, segments that allow robot to gain speed and to slow down must be added to the beginning and ending of the path. These can be line segments of length $S = v_{\max}^2/2a_{\max}$. Then the rest of the path must be built to connect shifted configurations

$$\begin{aligned} q_s' &= q_s + (S\cos\theta_s,\ S\sin\theta_s,\ 0)^T\,,\\ q_g' &= q_g - (S\cos\theta_g,\ S\sin\theta_g,\ 0)^T\,. \end{aligned}$$

The control signal for a linear velocity is then a trapezoid function

$$v(t) = \begin{cases} a_{\max}t, & 0 \leqslant t < \tau,\\ v_{\max}, & t \leqslant T-\tau,\\ v_{\max} - a_{\max}t, & T-\tau < t \leqslant T, \end{cases}$$

where T is total time of motion and $\tau = v_{\max}/a_{\max}$.

3.1 Dubins's Paths

A method of planning a path that connects two specified configurations using straight lines and circle arcs of minimum radius assuming that robot moves with a constant speed was proposed in [1]. It has been proved that the shortest path belongs to one of two families: CSC which stands for paths that consist of an arc, a tangent straight line and another arc; CCC which denotes paths combined of three arcs; or is a subpath of any of these. Considering that a curve segment can correspond to different rotations, left or right turn, these define a set of six paths.

The problem of these paths is that in the intermediate points between consequent segments of such path an angular acceleration is required to be infinite. Thus these paths fail to satisfy the angular acceleration limit.

3.2 Continuous-Curvature Turns

This problem was solved in [2] by offering continuous-curvature turns that provide a transitional phase between a straight line and a circle arc or two arcs with different direction of rotation. The main idea is to use clothoid arcs with a sharpness less or equal to the maximum sharpness $\sigma_{\max}$ for this purpose. $\sigma_{\max}$ and radius of circular arcs $r_{\min}$ depend on limitations (2):

$$r_{\min} = \frac{v_{\max}}{\omega_{\max}},\ \sigma_{\max} = \frac{\varepsilon_{\max}}{v_{\max}^2}. \tag{3}$$

The set of positions that are reachable from some initial position is a circle of radius r. At each point of such circle an angle between orientation of the robot and tangent to the circle is equal to a constant value μ. r and μ can be found using (3) and equations from [2]. These circles are called CC circles. For any initial position q four CC circles can be defined: $C_l^+(q)$, $C_r^+(q)$, $C_l^-(q)$, $C_l^-(q)$. Here sign "+" or "−" denotes direction of motion, forward or backward, r or l denotes direction of rotation. CC circles with "−" can be also interpreted as sets of configurations from which q is reachable by forward CC turns. Centers (x_Ω, y_Ω) of CC circles can be found as following

$$\begin{aligned} C_l^\pm(q): \quad & x_\Omega = x \pm r\sin(\mu \mp \theta), \\ & y_\Omega = y + r\cos(\mu \mp \theta); \\ C_r^\pm(q): \quad & x_\Omega = x \pm r\sin(\mu \pm \theta), \\ & y_\Omega = y - r\cos(\mu \pm \theta). \end{aligned} \tag{4}$$

CC turns can be used within Durbins's scheme by substituting regular circles representing turns with CC circles. An algorithm is to use (4) to find centers of $C_l^+(q_s)$, $C_r^+(q_s)$ for the initial position and $C_l^-(q_g)$, $C_r^-(q_g)$ for the goal and then connect them with tangent lines or circles considering directions of rotations.

The section of control signal for an angular velocity corresponding to a CC turn of angle δ is a trapezoid or triangular function. Let $\delta_{\min}$ be the angle turning by which requires a CC turn combined only of two symmetrical clothoid arcs of maximum sharpness.

Then if $\delta > \delta_{\min}$

$$\omega(t) = \begin{cases} \varepsilon_{\max} t, & t_0 \leqslant t < t_0 + \tau \\ \omega_{\max}, & t_0 + \tau \leqslant t \leqslant t_1 - \tau \\ \omega_{\max} - \varepsilon_{\max} t, & t_1 - \tau < t \leqslant t_1 \end{cases}$$

where t_0 and t_1 are moments of time of beginning and ending a CC turn respectively and $\tau = \omega_{\max}/\varepsilon_{\max}$.

If $\delta \leqslant \delta_{\min}$

$$\omega(t) = \begin{cases} \varepsilon t, & t_0 \leqslant t < \frac{t_0+t_1}{2} \\ -\varepsilon t, & \frac{t_0+t_1}{2} \leqslant t \leqslant t_1 \end{cases}$$

where $\varepsilon = \sigma v_{\max}$, $\sigma \leqslant \sigma_{\max}$.

Let us note that for building control signals we actually need to know only turning angles and lengths of line segments. The length of CC turn curve must be also known to find a total length of a path in order to determine the shortest one. This value can be computed knowing turning angle δ.

3.3 CSC-type Paths

Here and further let us assume that two centers $\Omega_1 = (x_1, y_1)$ and $\Omega_2 = (x_2, y_2)$ of appropriate CC circles are known.

In the case of CC circles tangency between such circle and a line is quite different from a classical tangency. μ-tangency, conditions of existence of those lines and ways to find lengths of line segments are described in [2]. So let us mention here only a way to determine turning angles for a first and a second turns.

For an external μ-tangent which is associated with left-straight-left (LSL) and right-straight-right (RSR) paths orientation of the robot while moving along a line segment is

$$\theta = \alpha = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right).$$

For an internal μ-tangent associated with a LSR path

$$\theta = \alpha - \beta + \frac{\pi}{2},$$

for a RSL path

$$\theta = \alpha - \beta,$$

where $\beta = \arcsin(2r\cos\mu/\rho)$, ρ is distance between centers.

Using θ_s, θ and θ_g and considering direction of rotation, angles of first and second turns can be obtained.

3.4 CCC-type Paths

For paths consisting of three curve segments, RLR or LRL, center of a CC circle tangent to both known circles must be found. In this case tangency between circles is classical though orientation of a robot is not aligned with a tangent line in the point of tangency.

For a pair of circles two options of a tangent circle are available. Their centers $\Omega = (x, y)$ can be found as following

$$x = x_1 + \cos\gamma,\ y = y_1 + \sin\gamma,\ \gamma = \alpha \pm \arccos(\rho/(4r)).$$

θ at the intermediate point between two circles with centers $\Omega_1' = (x_1', y_1')$ and $\Omega_2' = (x_2', y_2')$ is

$$\theta = \alpha' - \mu + \frac{\pi}{2},\ \alpha' = \arctan\left(\frac{y_2' - y_1'}{x_2' - x_1'}\right)$$

for a LR transition and

$$\theta = \alpha' + \mu - \frac{\pi}{2}$$

for a RL transition.

Again we can obtain angles of each of three turns using θ_s, θ_g and robot's orientations at the intermediate points.

3.5 Backward Motion

Also the possibility of backward motion is considered. In order to build paths for moving backwards it is convenient to virtually change orientations of q_s and q_g to opposite by adding π to corresponding values. Then the same computations must be performed considering that rotation directions are changed to opposite. Finally, signs of both control laws must be changed to opposite.

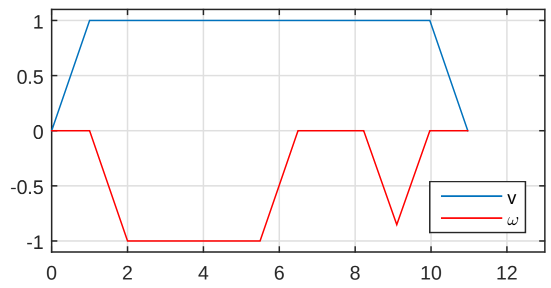

Figure 1: Control signals.

4. FEEDBACK

In order to provide stability of the motion path planning is used along with the tracking control offered in [3]. This approach supposes using a reference object described by model (1). Previously built control laws are acting as inputs to the reference object and its output is a trajectory function $q_r(t) = (x_r(t),\ y_r(t),\ \theta_r(t))$. Then control laws that go to robot's input are following:

$$v(t) = v_r(t) + C_2 \{[x_r(t - x(t))] \cos\theta + [y_r(t) - y(t)] \sin\theta\},$$
$$\omega(t) = \omega_r(t) + C_2 [\theta_r(t) - \theta(t)].$$

In [3] it is proved that if $C_1 > 0$, $C_2 > 0$ then a closed-loop system is exponentially stable.

5. EXPERIMENTAL RESULTS

To demonstrate the performance of proposed algorithm the Simulink model of the closed-loop system has been made. The robot is modeled with presence of disturbance and measurement noise.

The model is also utilized for solving the problem of choosing C_1 and C_2 that provide the best performance. The maximum of mean squared error for each degree of freedom in N moments of time is chosen as motion quality measure:

$$e(C_1, C_2) = \max_{\xi = x,y,\theta} \left[\frac{1}{N} \sum_{i=0}^{N-1} \left(\xi_r\left(\frac{T}{N} i\right) - \xi\left(\frac{T}{N} i\right) \right)^2 \right]$$

where $q_r(t)$ is reference trajectory and $q(t)$ is simulated trajectory of the robot.

Then the quality matrix Q is built for some grid of C_1 and C_2 values. The pair of coefficients that gives the least $e(C_1, C_2)$ is taken as an answer.

Quality matrices can be built for some set of trajectories, for example, a set that includes at least one of each type, forward and backward. Then coefficients are chosen using mean of all matrices Q for this set. The other way is to consider 12 sets, each containing trajectories of the same type, and get 12 pairs (C_1, C_2). Which values to actually use must be decided after the path planning stage when it is known path of which type has turned out to be optimal.

Now let us demonstrate performance of the system for specified initial and goal positions

$$q_s = (0,\ 0,\ 0)^T,\ q_g = (0,\ 3,\ \pi/3)^T$$

and limitations

$$v_{\max} = 1,\ \omega_{\max} = 1,\ a_{\max} = 1,\ \varepsilon_{\max} = 1.$$

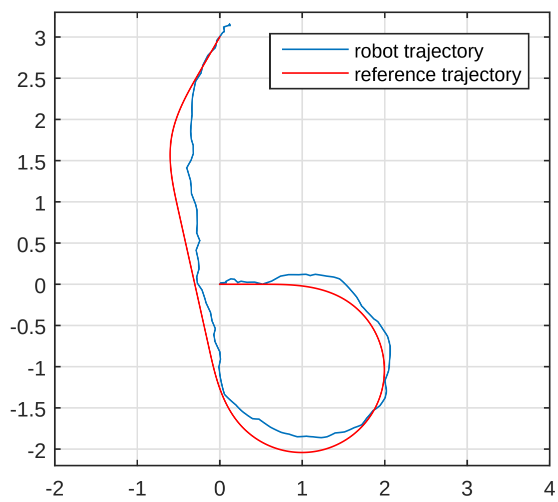

Figure 2: Robot trajectory and reference trajectory.

Values of feedback coefficients are taken

$$C_1 = 1.5,\ C_2 = 0.2.$$

Results are shown on figures 1 and 2. Robot stopped at position $q(t) = (0.04,\ 3.23,\ 1.14)^T$ and whole motion took $T = 10.97$ seconds.

6. CONCLUSIONS

Presented approach allows planning an admissible trajectory that is guaranteed to be realized by control laws that satisfy all considered constraints and building feedback control laws which drive robot to specified goal precisely enough in spite of presence of disturbance and measurement noise.

A subject to further research can be the task of creating more universal approach to choosing feedback law coefficients. Another direction is designing an algorithm to build a full set of continuous-curvature Reeds and Shepp's [4] paths which unlike Dubins's paths include cusps, points where direction of motion is changed. The main problem here is to satisfy linear acceleration limit at cusp points.

7. REFERENCES

[1] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics*, pages 497–516, 1957.

[2] T. Fraichard and A. Scheuer. From reeds and shepp's to continuous-curvature paths. *Robotics, IEEE Transactions on*, 20(6):1025–1035, 2004.

[3] E. Panteley, E. Lefeber, A. Loria, and H. Nijmeijer. Exponential tracking control of a mobile car using a cascaded approach. In *Proceedings of the IFAC workshop on motion control*, pages 221–226. Pergamon Grenoble, France, 1998.

[4] J. Reeds and L. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific journal of mathematics*, 145(2):367–393, 1990.