Automatic Parallel Parking via Trajectory Optimization

Liangchun Xu

Trajectory optimization can be applied into designing automatic parallel parking system. In this paper, a trapezoid direct collocation transcription method is implemented to solve parallel parking and straight line reverse problems.

Keywords: parallel parking, straight line reverse, trajectory optimization

1. Problem Statement

This final project primarily focuses on automatic parallel parking. A trapezoid direct collocation transcription method is implemented to solve parallel parking and straight line reverse problems. Both probelms are solved using the same transcription method. Simplified assumptions are made to model the automatic parallel parking system as follows

1. With only one availible spot between two used lots, the start and end locations of the car waiting to be parked are pre-determined;
2. All cars and parking lots are modeled as retangular. The car size is set as $4.726m \times 2.022m$, the parking lot size is set as $6.096m \times 2.591m$.
3. The road is flat, so the car only has 3 DoFs (x, y, θ), which are its 2D positions and rotation, respectively.

Car reversing along a straight line is an indispensable skill for drivers. Especially when the initial position of a car is disturbed, the driver need to adjust the pose of the car and reverse it straight along the curb. In this report, car reversing is used to test the transcription method before the automatic parallel parking implementation, which assures the transcription method's correctness.

2. Transcription Method

Trapezoid direct collocation method is used as the transcription method in this report, which approximates the control trajectory and the system dynamics as piecewise linear splines. The state trajectory then becomes piecewise quadratic splines, and the knot points of the spline are coincident with the collocation points.

All constraints and objective functions are composed of boudary parts and path parts. In transcription method function there are interface to load those constraints and costs from specific problems. The defect constraints for trapezoid method are

$$\vec{C}_k = \vec{x}_k + \frac{h_k}{2}(f(\vec{x}_k, \vec{u}_k) + f(\vec{x}_{k+1}, \vec{u}_{k+1})) - \vec{x}_{k+1} \tag{1}$$

Accordingly, the objecive function is

$$J = \sum_{k=0}^{N-1} \frac{h_k}{2}(g_k + g_{k+1}) \tag{2}$$

The transcription method is general purpose. In particular, for parallel parking and car reversing problem, it minimize the sum of the duration and the control effort (squared sum of control inputs).

3. Dynamical Model

A nonholonomic dynamical model of the car used in this report can be captured in [1]

$$\begin{aligned} \dot{x} &= u_1 \cos(\theta) \\ \dot{y} &= u_1 \sin(\theta) \\ \dot{\theta} &= \frac{\tan(\phi)}{l} u_1 \\ \dot{\phi} &= u_2 \end{aligned} \tag{3}$$

States are chosen to be $X = [x, y, \theta, \phi]^T$, which denotes the position and the orientation of the robot respectively. The control inputs $U = [u_1, u_2]$ are linear and angular steering velocities of the car. l is the length of wheelbase, which is $2.820m$ in the implementation.

4. Straight Line Reverse

The straight line reverse problem is to reverse a disturbed car in one lot with initial states

$$(x_0, y_0, \theta_0, \phi_0) = \left(-1.425m, 15.240m, \frac{6\pi}{13}, 0 \right) \tag{4}$$

into anther empty lot with the final state, which is parallel to the right curb

$$(x_F, y_F, \theta_F, \phi_F) = \left(-1.296m, 3.048m, \frac{\pi}{2}, 0 \right) \tag{5}$$

Two limits are set to bound states and controls as follows

$$\begin{aligned} \phi &\in \left[-\frac{\pi}{4}, \frac{\pi}{4} \right] \\ u_1 &\in [-2m/s, 0m/s] \end{aligned} \tag{6}$$

which means the maximum steering angle of this car is less than 45°; the maximum reverse velocity of the car is less than $2m/s$.

Besides the boundary constraints, the path constaints here are all four vertices of the car should lie left to the curb. If (x, y, θ) are intemediate states, the four vertices can be calculated as

$$\begin{pmatrix} x_a & x_b & x_c & x_d \\ y_a & y_b & y_c & y_d \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} -\frac{l_c}{2} & -\frac{l_c}{2} & -\frac{l_c}{2} & -\frac{l_c}{2} \\ -\frac{w_c}{2} & -\frac{w_c}{2} & -\frac{w_c}{2} & -\frac{w_c}{2} \end{pmatrix} + \begin{pmatrix} x & x & x & x \\ y & y & y & y \end{pmatrix}$$

where l_c, w_c are the length and the width of the car. The constraints are set to be

$$(\; x_a \;\; x_b \;\; x_c \;\; x_d \;)^T < \vec{0} \tag{7}$$

Objective function is to minimize the process duration and the control effort together

$$\begin{aligned} g_k &= g(t_k, x_k, u_k) \\ &= \text{sum}(u_k^T u_k) \qquad (8) \\ J &= J_{\text{path}} + J_{\text{bnd}} \\ &= \sum_{k=0}^{N-1} \frac{h_k}{2}(g_k + g_{k+1}) + (t_F - t_0) \qquad (9) \end{aligned}$$

Under these setting, the final simulation results are shown as follows. The four lines in different colors in Figure 1 show the tracks of four vertices of the car.

Figure 1. Straight line reverse

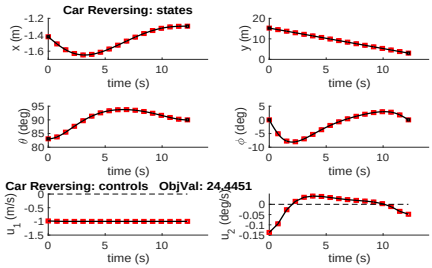

Figure 2. Straight line reverse: states and controls

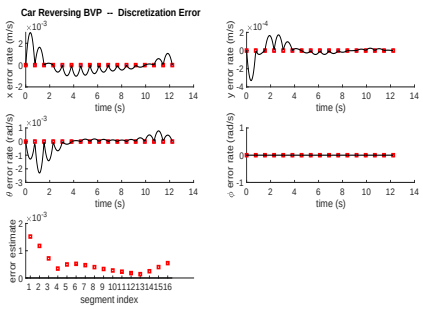

Figure 3. Straight line reverse: error plot

The fmincon solver converges easily, and the state and control solutions are smooth with small errors. It should be noted that the steering angle error rate is 0. The reason is the trapezoid method uses linear control asssumption, while according to the car's dynamic model indicates the derivative of the steering angle is this linear control input alone. This causes the zero error rate for steering angle.

5. Parallel Parking

After testing the transcription method with simple straight line reverse problem, we switch to more challenging parallel parking problem. The objective function here used is the same as in (9). The initial and final states are differently defined as

$$(x_0, y_0, \theta_0, \phi_0) = \left(-4.613m, 13.145m, \frac{\pi}{2}, -\frac{\pi}{4}\right) \tag{10}$$

$$(x_F, y_F, \theta_F, \phi_F) = \left(-4.613m, 13.145m, \frac{\pi}{2}, 0\right) \tag{11}$$

The initial steering angle is set to be the maximum steering angle which is in favor of starting the parallel parking. The path constraints are set to avoid one big circle inside the top lot and one small ellipse on the edge of the top lot, which is shown in Figure 4. The constraints are applied to the top two vertices

$$\begin{aligned} r_c &= \frac{w_{lot}}{2} \\ (x_d + r_c)^2 + (y_d - (2l_{lot} + r_c))^2 &> r_c^2 \end{aligned} \tag{12}$$

where l_{lot}, w_{lot} are the lot lenght and the lot width respectively. The ellipse constraints are

$$\begin{aligned} r_a &= \frac{w_{lot}}{2} \\ r_b &= \frac{l_{lot}}{50} \\ \frac{(x_d + r_c)^2}{r_a^2} + \frac{(y_d - 2l_{lot})^2}{r_b^2} &> 1 \end{aligned} \tag{13}$$

$$\frac{(x_c + r_c)^2}{r_a^2} + \frac{(y_c - 2l_{lot})^2}{r_b^2} > 1 \tag{14}$$

Two top vertices of the car are both bounded by the ellipse because the car can move forward inside the parking bay, which could cause collision if not bounded. Other constraints include

$$x_d < 0 \tag{15}$$

$$x_a < 0 \tag{16}$$

$$y_a > l_{lot} \tag{17}$$

$$y_b > l_{lot} \tag{18}$$

The velocity bound is changed to $u_1 \in [-2m/s, 2m/s]$ in case the car need positive velocity inside the parking bay. Under these settings, the final results are shown as below. The fmincon solver is extremely picky about the initialization. So I used several results of trapzoid method without path constraints to initialize the final solver. Still, the final solver takes about 400 iterations to reach a local mininal, which satisfied all the constraints, but from the error plot, the errors are a little bit larger than expected. It might be improved in the future work.

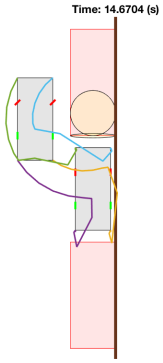

Figure 4. Parallel parking

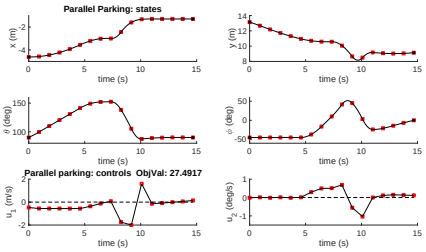

Figure 5. Parallel parking: states and controls

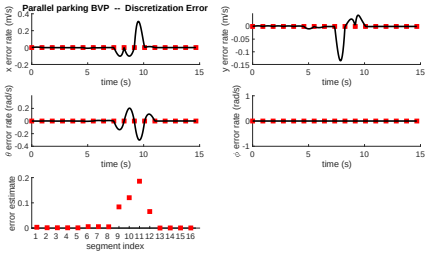

Figure 6. Parallel parking: error plot

APPENDIX

Aside from satisfying the basic requirements, the advanced features of this project are

- Constraints not found in homework problems for this course;
- Use a dynamical system that is not found in the code library for this course;
- Implement a tracking controller and forward simulation using the solution to one or both of your optimization problems.

BIBLIOGRAPHY

[1] Bernhard Müller. *Two-step Trajectory Planning for Automatic Parking*. Shaker, 2009.