Motion Planning and State Estimation in Robotics

Arun Kumar Singh

University of Tartu *arun.singh@ut.ee, aks1812@gmail.com*

October 19, 2021

Overview

Course Repo

Link to Course Repo
https://github.com/arunkumar-singh/Motion_Planning_Lecture_Codes

Global Trajectory Planning

- Till now we have considered only incremental trajectory planning. That is, we were only considered about planning a trajectory for say next 0.1s or less.
- Incremental trajectory planning is myopic and greedy. The overall trajectory quality is sub-optimal
- We now move to global trajectory planning. That is, we plan the trajectory taking into account all the costs and constraints for all time steps into the future
- We will do this by modeling trajectories as polynomials.

Recall Triple Integrator System

If the input jerk j_x, j_y is constant, we get the following.

$$\dddot{x} = \jmath_x, \dddot{y} = \jmath_y \tag{1}$$

$$\ddot{x}(t) = \ddot{x}_0 + \jmath_x(t - t_0), \ddot{y} = \ddot{y} + \jmath_y(t - t_0) \tag{2}$$

$$\dot{x}(t) = \dot{x}_0 + \ddot{x}_0(t - t_0) + \frac{1}{2}\jmath_x(t - t_0)^2, \dot{y}(t) = \dot{y}_0 + \ddot{y}_0(t - t_0) + \frac{1}{2}\jmath_y(t - t_0)^2 \tag{3}$$

$$x(t) = x_0 + \dot{x}_0 t + \frac{1}{2}\ddot{x}_0(t - t_0)^2 + \frac{1}{6}\jmath_x(t - t_0)^3 \tag{4}$$

$$y(t) = y_0 + \dot{y}_0 t + \frac{1}{2}\ddot{y}_0(t - t_0)^2 + \frac{1}{6}\jmath_y(t - t_0)^3 \tag{5}$$

What if the input is not constant?

Polynomial Trajectories

Let

$$\dddot{x}(t) = 6a_1 + 24a_2 t + 60a_3 t^2 + 120a_4 t^3 + 210a_5 t^4 \tag{6}$$

Integrating three times, we get the following, where $x_0, \dot{x}_0, \ddot{x}_0$ are the initial position, velocity and acceleration of the robot.

$$\ddot{x}(t) = \ddot{x}_0 + 6a_1 t + 12a_2 t^2 + 20a_3 t^3 + 30a_4 t^4 + 42a_5 t^5 \tag{7}$$

$$\dot{x}(t) = \dot{x}_0 + \ddot{x}_0 t + 3a_1 t^2 + 4a_2 t^3 + 5a_3 t^4 + 6a_4 t^5 + 7a_5 t^6 \tag{8}$$

$$x(t) = x_0 + \dot{x}_0 t + \frac{1}{2}\ddot{x}_0 t^2 + a_1 t^3 + a_2 t^4 + a_3 t^5 + a_4 t^6 + a_5 t^7 \tag{9}$$

We can write similar expression for y as well

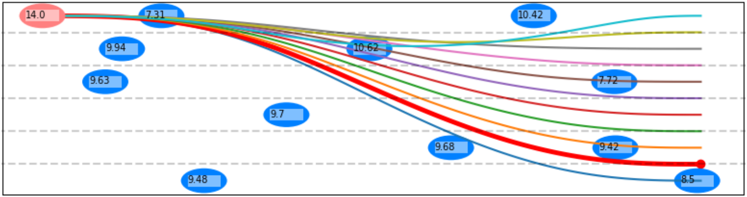
$$\ddot{y}(t) = \ddot{y}_0 + 6b_1 t + 12b_2 t^2 + 20b_3 t^3 + 30b_4 t^4 + 42b_5 t^5 \tag{10}$$

$$\dot{y}(t) = \dot{y}_0 + \ddot{y}_0 t + 3b_1 t^2 + 4b_2 t^3 + 5b_3 t^4 + 6b_4 t^5 + 7b_5 t^6 \tag{11}$$

$$y(t) = y_0 + \dot{y}_0 t + \frac{1}{2}\ddot{y}_0 t^2 + b_1 t^3 + b_2 t^4 + b_3 t^5 + b_4 t^6 + b_5 t^7 \tag{12}$$

Computing the right a's and b's

- Different a's and b's lead to different trajectories.
- Global trajectory planning can be thought as the problem of computing the right a's and b's for your problem setting.



(a)

Figure: Each trajectory corresponds to a different set of a's and b's

Polynomial Trajectories in Matrix Form

During numerical computations, we are mostly interested in knowing trajectory values at specific time instants like $t_0, t_1, t_2 \dots t_f$.

$$x(t_0) = x_0 \tag{13}$$

$$x(t_1) = x_0 + \dot{x}_0 t_1 + \frac{1}{2}\ddot{x}_0 t_1^2 + a_1 t_1^3 + a_2 t_1^4 + a_3 t_1^5 \tag{14}$$

$$x(t_2) = x_0 + \dot{x}_0 t_2 + \frac{1}{2}\ddot{x}_0 t_2^2 + a_1 t_2^3 + a_2 t_2^4 + a_3 t_2^5 \tag{15}$$

$$\dots\dots\dots\dots\dots\dots \tag{16}$$

$$x(t_f) = x_0 + \dot{x}_0 t_f + \frac{1}{2}\ddot{x}_0 t_f^2 + a_1 t_f^3 + a_2 t_f^4 + a_3 t_f^5 \tag{17}$$

Polynomial Trajectories in Matrix Form

In matrix form, we have

$$x(t) = \begin{bmatrix} x(t_1) \\ x(t_2) \\ x(t_3) \\ \dots \\ x(t_f) \end{bmatrix} = \mathbf{A}_x \widetilde{\mathbf{c}_x} + \mathbf{P}_x \mathbf{c}_x, \tag{18}$$

where,

$$\mathbf{A}_x = \begin{bmatrix} 1.0 & t_1 & 0.5t_1^2 \\ 1.0 & t_2 & 0.5t_2^2 \\ 1.0 & t_3 & 0.5t_3^2 \\ \dots\dots & & \\ 1.0 & t_f & 0.5t_f^2 \end{bmatrix}, \widetilde{\mathbf{c}_x} = \begin{bmatrix} x_0 \\ \dot{x}_0 \\ \ddot{x}_0 \end{bmatrix} \tag{19}$$

$$\mathbf{P}_x = \begin{bmatrix} t_1^3 & t_1^4 & t_1^5 \\ t_2^3 & t_2^4 & t_2^5 \\ t_3^3 & t_3^4 & t_3^5 \\ \dots\dots & & \\ t_f^3 & t_f^4 & t_f^5 \end{bmatrix}, \mathbf{c}_x = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{20}$$

Polynomial Trajectories in Matrix Form

Similarly, we get

$$\dot{x}(t) = \begin{bmatrix} \dot{x}(t_1) \\ \dot{x}(t_2) \\ \dot{x}(t_3) \\ \dots \\ \dot{x}(t_f) \end{bmatrix} = \dot{\mathbf{A}}_x \widetilde{\mathbf{c}_x} + \dot{\mathbf{P}}_x \mathbf{c}_x, \tag{21}$$

where,

$$\dot{\mathbf{A}}_x = \begin{bmatrix} 0.0 & 1.0 & t_1 \\ 0.0 & 1.0 & t_2 \\ 0.0 & 1.0 & t_3 \\ \dots\dots & & \\ 0.0 & 1.0 & t_4 \end{bmatrix}, \widetilde{\mathbf{c}_x} = \begin{bmatrix} x_0 \\ \dot{x}_0 \\ \ddot{x}_0 \end{bmatrix} \tag{22}$$

$$\dot{\mathbf{P}}_x = \begin{bmatrix} 3t_1^2 & 4t_1^3 & 5t_1^4 \\ 3t_2^2 & 4t_2^3 & 5t_2^4 \\ 3t_2^2 & 4t_2^3 & 5t_2^4 \\ \dots\dots & & \\ 3t_f^2 & 4t_f^3 & 5t_f^4 \end{bmatrix}, \mathbf{c}_x = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{23}$$

Polynomial Trajectories in Matrix Form

Similarly, we get

$$\ddot{x}(t) = \begin{bmatrix} \ddot{x}(t_1) \\ \ddot{x}(t_2) \\ \ddot{x}(t_3) \\ \dots \\ \ddot{x}(t_f) \end{bmatrix} = \ddot{\mathbf{A}}_x \widetilde{\mathbf{c}_x} + \ddot{\mathbf{P}}_x \mathbf{c}_x, \tag{24}$$

where,

$$\dot{\mathbf{A}}_x = \begin{bmatrix} 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 1.0 \\ \dots\dots & & \\ 0.0 & 0.0 & 1.0 \end{bmatrix}, \widetilde{\mathbf{c}}_x = \begin{bmatrix} x_0 \\ \dot{x}_0 \\ \ddot{x}_0 \end{bmatrix} \tag{25}$$

$$\ddot{\mathbf{P}}_x = \begin{bmatrix} 6t_1 & 12t_1^2 & 20t_1^4 \\ 6t_2 & 12t_2^2 & 20t_2^4 \\ 6t_3 & 12t_3^2 & 20t_3^4 \\ \dots\dots & & \\ 6t_f & 12t_f^2 & 20t_f^4 \end{bmatrix}, \mathbf{c}_x = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{26}$$

Point to Point Trajectory in Matrix Form

$$\overbrace{\begin{bmatrix} x(t_f) \\ \dot{x}(t_f) \\ \ddot{x}(t_f) \end{bmatrix}}^{\mathbf{s}_{end}} = \overbrace{\begin{bmatrix} {}^n\mathbf{A}_x \\ {}^n\dot{\mathbf{A}}_x \\ {}^n\ddot{\mathbf{A}}_x \end{bmatrix}}^{\mathbf{G}_{end}} \widetilde{\mathbf{c}}_x + \overbrace{\begin{bmatrix} {}^n\mathbf{P}_x \\ {}^n\dot{\mathbf{P}}_x \\ {}^n\ddot{\mathbf{P}}_x \end{bmatrix}}^{\mathbf{H}_{end}} \mathbf{c}_x \tag{27}$$

Solution is given by

$$\mathbf{c}_x = \mathbf{H}_{end}^{-1}(\mathbf{s}_{end} - \mathbf{G}_{end}\widetilde{\mathbf{c}}_x) \tag{28}$$

Solution for intermediate way-point trajectory

We need to append the rows corresponding to intermediate points. Suppose, the intermediate points happen at time t_i, t_j. So, we need to extract row i and j from matrix $\mathbf{A}_x$ and $\mathbf{P}_x$.

$$\overbrace{\begin{bmatrix}\mathbf{s}_{mid}\\ \mathbf{s}_{end}\end{bmatrix}}^{\mathbf{s}} = \overbrace{\begin{bmatrix}\mathbf{G}_{mid}\\ \mathbf{G}_{end}\end{bmatrix}}^{\mathbf{G}} \widetilde{\mathbf{c}}_x + \overbrace{\begin{bmatrix}\mathbf{H}_{mid}\\ \mathbf{H}_{end}\end{bmatrix}}^{\mathbf{H}} \mathbf{c}_x \tag{29}$$

$$\mathbf{G}_{mid} = \begin{bmatrix}{}^{i}\mathbf{A}_x\\ {}^{j}\mathbf{A}_x\end{bmatrix}, \mathbf{H}_{mid} = \begin{bmatrix}{}^{i}\mathbf{P}_x\\ {}^{j}\mathbf{P}_x\end{bmatrix}, \mathbf{s}_{mid} = \begin{bmatrix}x(t_1)\\ x(t_2)\end{bmatrix} \tag{30}$$

Solution is given by

$$\mathbf{c}_x = \mathbf{H}^{-1}(\mathbf{s} - \mathbf{G}\widetilde{\mathbf{c}}_x) \tag{31}$$

Introducing Optimality

- We want the robots to behave in a particular way and not just go from point A to point B
- The most intuitive way of describing robot behavior is in terms of cost functions. For example, move with a particular reference velocity as much as possible.
- Most trajectory planning problems are framed as optimization problems

Computing minimum of functions

$$\min f(s) = s^2 - 6s + 4 \tag{32}$$

To solve this minimization, we take the derivative with respect to x and equate it to zero

$$2s - 6 = 0 \tag{33}$$

Multi variable function

$$\min f(s_1, s_2) = 2s_1^2 + 2s_1 s_2 + 2s_2^2 - 6s_1 \tag{34}$$

We compute partial derivatives and equate them to zero

$$\bigtriangledown f_{s_1} = 4s_1 + 2s_2 - 6 = 0 \tag{35}$$

$$\bigtriangledown f_{s_2} = 2s_1 + 4s_2 = 0 \tag{36}$$

(37)

Multi-variable quadratic function in Matrix form

$$f(s_1, s_2) = 2s_1^2 + 2s_1 s_2 + 2s_2^2 - 6s_1$$

$$\Rightarrow f(s_1, s_2) = \frac{1}{2}\begin{bmatrix} s_1 \\ s_2 \end{bmatrix}^T \overbrace{\begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}}^{\mathbf{Q}} \overbrace{\begin{bmatrix} s_1 \\ s_2 \end{bmatrix}}^{\mathbf{s}} + \underbrace{\begin{bmatrix} -6 \\ 0 \end{bmatrix}^T}_{\mathbf{q}} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$$

$$f(s_1, s_2) = \frac{1}{2}\mathbf{s}^T\mathbf{Q}\mathbf{s} + \mathbf{q}^T\mathbf{s} \tag{38}$$

The derivative (or gradient) in matrix form is given by

$$\bigtriangledown f = \mathbf{Q}\mathbf{s} + \mathbf{q} = 0 \Rightarrow \mathbf{s} = -\mathbf{Q}^{-1}\mathbf{q} \tag{39}$$

Multiple Quadratic Costs

$$
\begin{aligned}
\frac{1}{2}\mathbf{s}^T\mathbf{Q}_1\mathbf{s} + \mathbf{q}_1^T\mathbf{s} + \frac{1}{2}\mathbf{s}^T\mathbf{Q}_2\mathbf{s} + \mathbf{q}_2^T\mathbf{s} \\
= \frac{1}{2}\mathbf{s}^T\mathbf{Q}\mathbf{s} + \mathbf{q}^T\mathbf{s} \\
\mathbf{Q} = \mathbf{Q}_1 + \mathbf{Q}_2, \mathbf{q} = \mathbf{q}_1 + \mathbf{q}_2
\end{aligned} \tag{40}
$$

The derivative (or gradient) in matrix form is given by

$$
\mathbf{Q}\mathbf{s} + \mathbf{q} = 0 \Rightarrow \mathbf{s} = -\mathbf{Q}^{-1}\mathbf{q} \tag{41}
$$

Least Squares Cost

Consider the following cost function where a_{ij}, d_i are given constants.

$$\min f(s_1, s_2) = \frac{1}{2}((a_{11}s_1 + a_{12}s_2 - d_1)^2 + (a_{21}s_1 + a_{22}s_2 - d_2)^2 + (a_{31}s_1 + a_{32}s_2 - d_3)^2) \tag{42}$$

$$\begin{aligned} f(s_1, s_2) &= \frac{1}{2} \| \overbrace{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}}^{\mathbf{A}} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} - \overbrace{\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}}^{\mathbf{b}} \|_2^2 \\ f(s_1, s_2) &= \mathbf{s}^T\mathbf{Q}\mathbf{s} + \mathbf{q}^T\mathbf{s} \\ \mathbf{Q} &= \mathbf{A}^T\mathbf{A}, \mathbf{q} = -\mathbf{A}^T\mathbf{b} \end{aligned} \tag{43}$$

Again, solution is given by

$$\mathbf{Q}\mathbf{s} + \mathbf{q} = 0 \Rightarrow \mathbf{s} = -\mathbf{Q}^{-1}\mathbf{q} \tag{44}$$

Constructing Cost Functions

Suppose, you want to choose s_1, s_2 such that their sum is as close as possible to 1 and s_2 is as close as possible to 0.5

Constructing Cost Functions

Suppose, you want to choose s_1, s_2 such that their sum is as close as possible to 1 and s_2 is as close as possible to 0.5

$$f(s_1, s_2) = (s_1 + s_2 - 1)^2 + (s_2 - 0.5)^2 \tag{45}$$

Constructing Cost Functions

Suppose, you want to choose s_1, s_2 such that their sum is as close as possible to 1 and magnitude of s_2 is as small as possible

Constructing Cost Functions

Suppose, you want to choose s_1, s_2 such that their sum is as close as possible to 1 and magnitude of s_2 is as small as possible

$$f(s_1, s_2) = (s_1 + s_2 - 1)^2 + (s_2)^2 \tag{46}$$