EE5110/5062/6110: Special Topics in Automation & Control—Autonomous systems

Motion planning (III)

Sunan Huang

Temasek Laboratories

National University of Singapore

— Oct 6. 2021 —

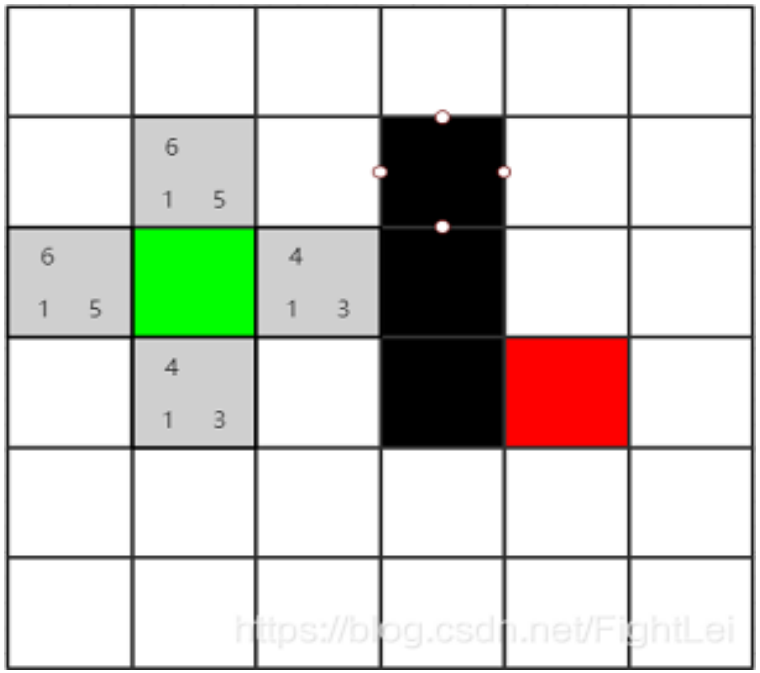
NUS National University of Singapore

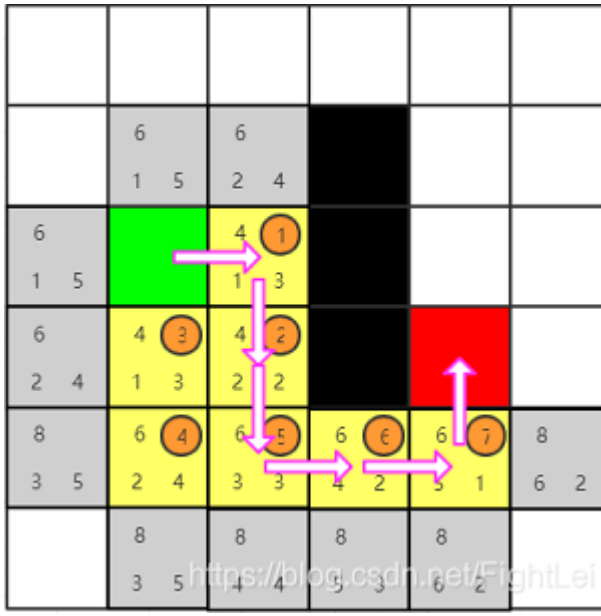
Review:

- **A* finds the shortest path between starting point A and ending point B without colliding with static obstacles**

An example in MATLAB

Rule:right->down->left->up





Can A* handle moving obstacles?

Learning Objective

- Understand the trajectory generation—polynomial method
- Can integrate this trajectory with controller.

NUS National University of Singapore

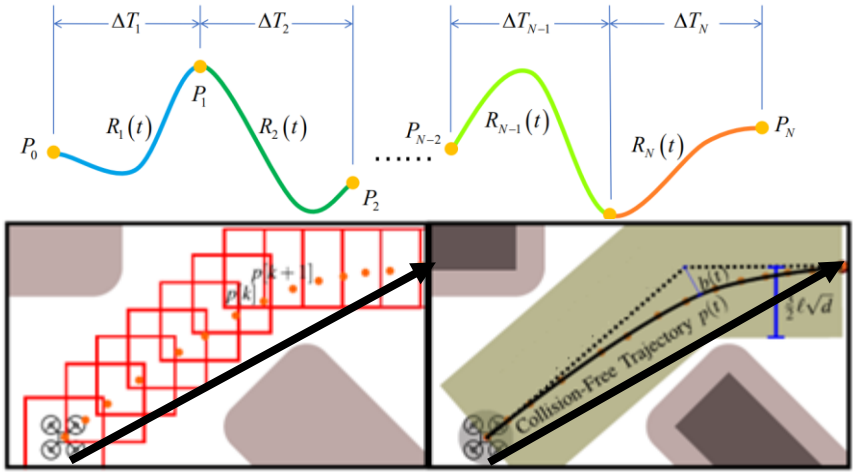
Topics To Be Covered

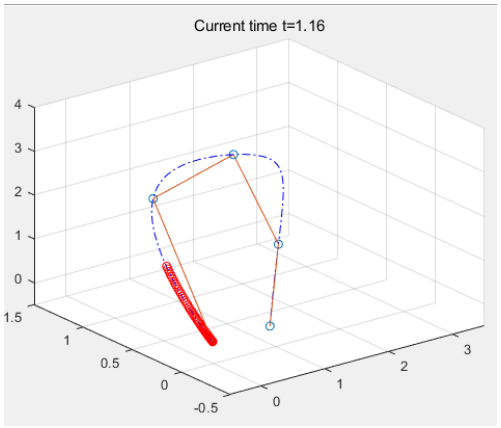
1. **Concept of trajectory generation**
2. **Several typical trajectories**
3. ***Trajectory represented by polynomial**

1.Background

Can we move from point A to point B directly even it is free?

- A*
- D*

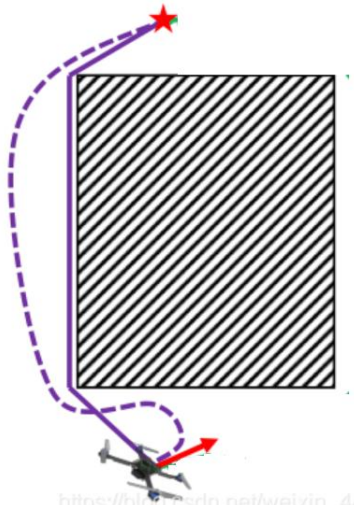


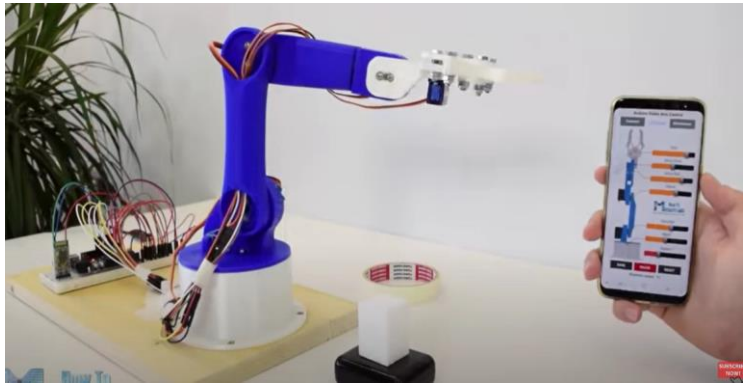


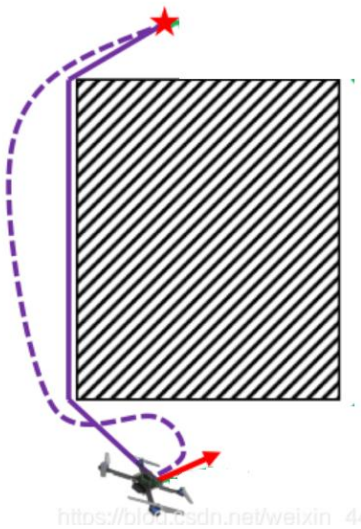
Fusion work of path-finding algorithm

Optimization of path: trajectory(smooth)

Following trajectory







Path vs. Trajectory

Path: is an ordered locus of points in the space (either joint or operational), witch the manipulator should follow.
It is a pure geometric description of motion.

Trajectory: a path on which timing law is specified, e.g., velocities and accelerations in each point.

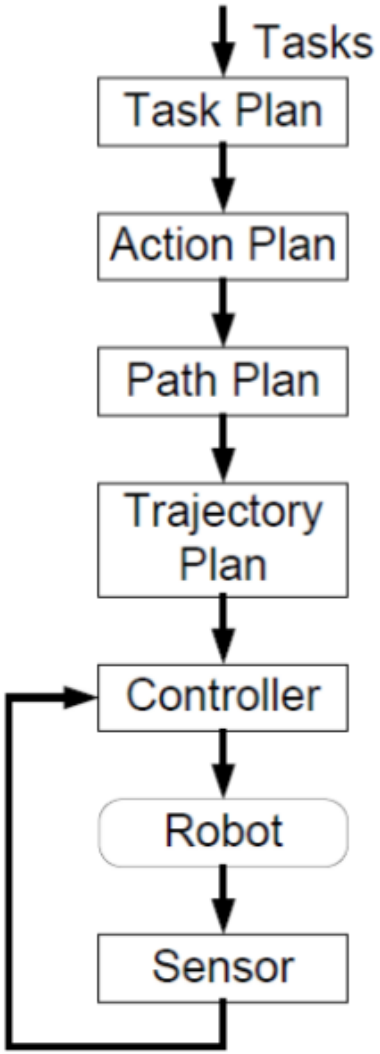
Robot Motion Planning

Path planning:

- Geometric path
- Issues: optimal path.

Trajectory:

- Interpolate or approximate the desired path by a class of polynomial functions.
- Generate a sequence of time-based "control set points" for the control of manipulator from the initial configuration to its destination.



NUS National University of Singapore

Roughly, a trajectory is a specification of the robot position (configuration) as a function of time.

Trajectory Generation Problem

- **Trajectory: θ(s(t)) or θ(t)**
- **Velocity and acceleration along a trajectory θ(s(t)):**

$$\dot{\theta} = \frac{d\theta}{ds}\dot{s}; \qquad \ddot{\theta} = \frac{d\theta}{ds}\ddot{s} + \frac{d^2\theta}{ds^2}\dot{s}^2$$

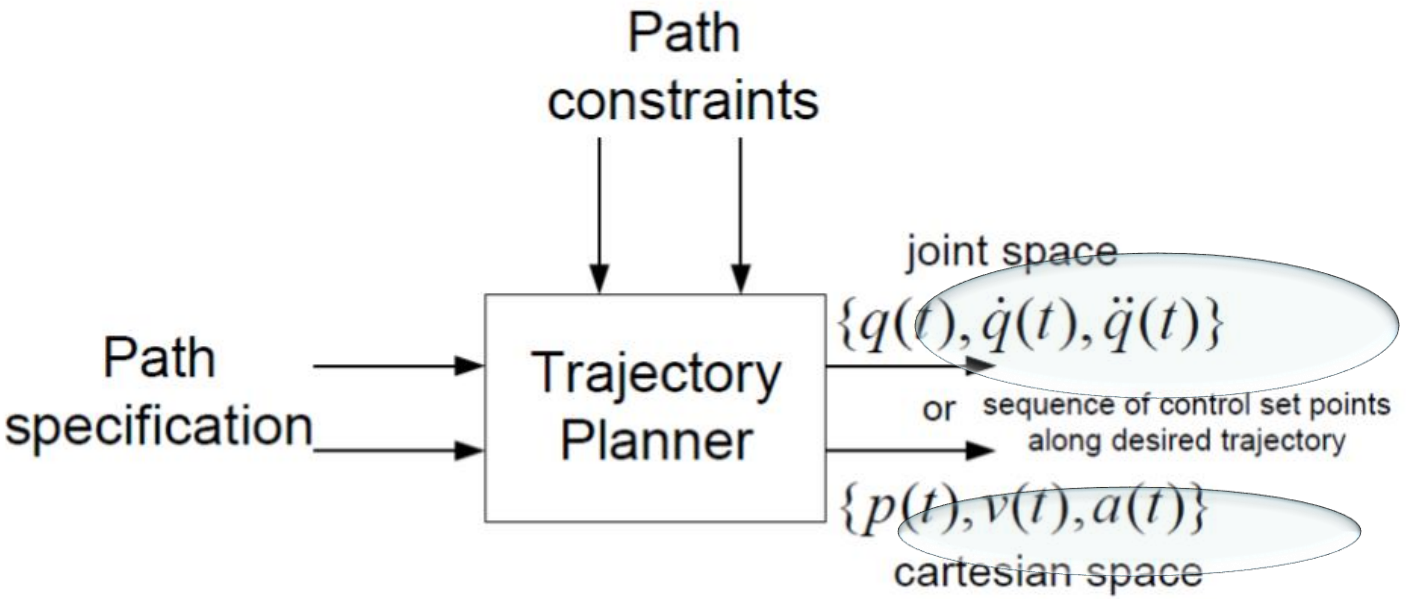
- **Jerk and snap along a trajectory θ(s(t)):**
- $\dddot{\theta}$ *and* $\ddddot{\theta}$

- **Trajectory Generation: Construct a trajectory (path + time scaling) so that the robot reaches a sequence of points in a given time**

- **Trajectory should be sufficiently smooth and respect limits on joint variables, velocities, accelerations, or torques**

NUS National University of Singapore

Trajectory Planning

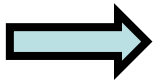


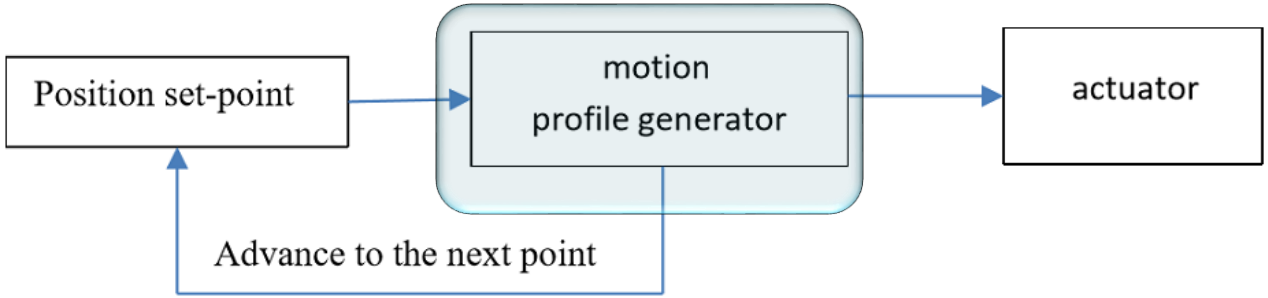
- Path Profile
- Velocity Profile
- Acceleration Profile

NUS National University of Singapore

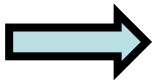
Implementation of the motion profiles generator

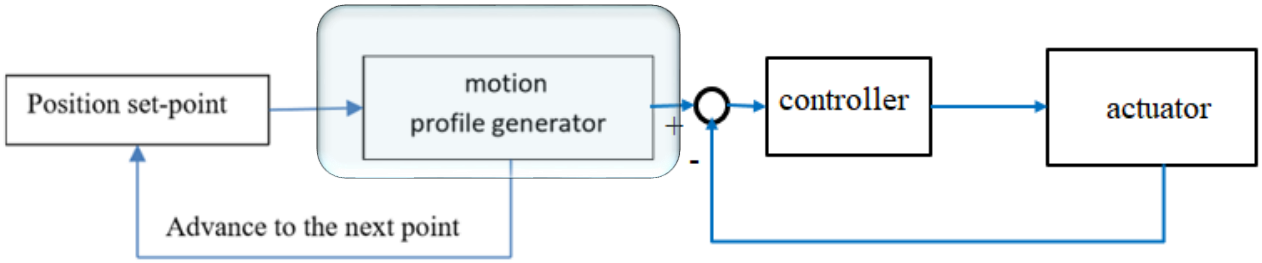
Feedforward





Desired





NUS National University of Singapore

2. Several typical motion profiles

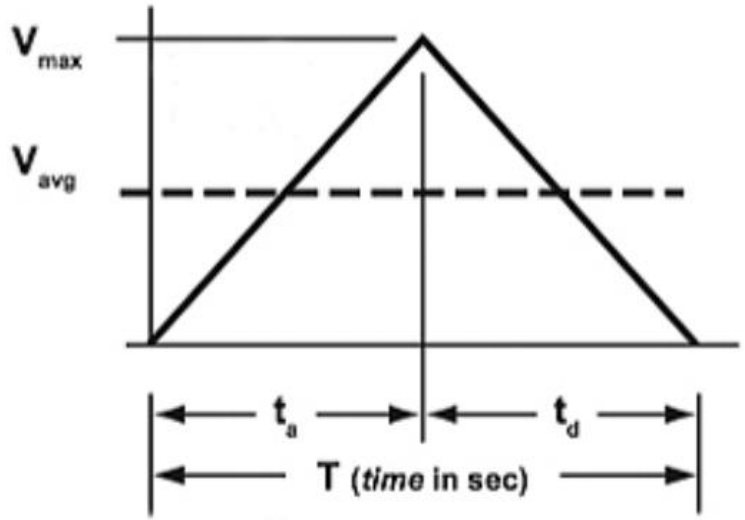
The type of motion profile required for an application depends on the purpose of the movement.

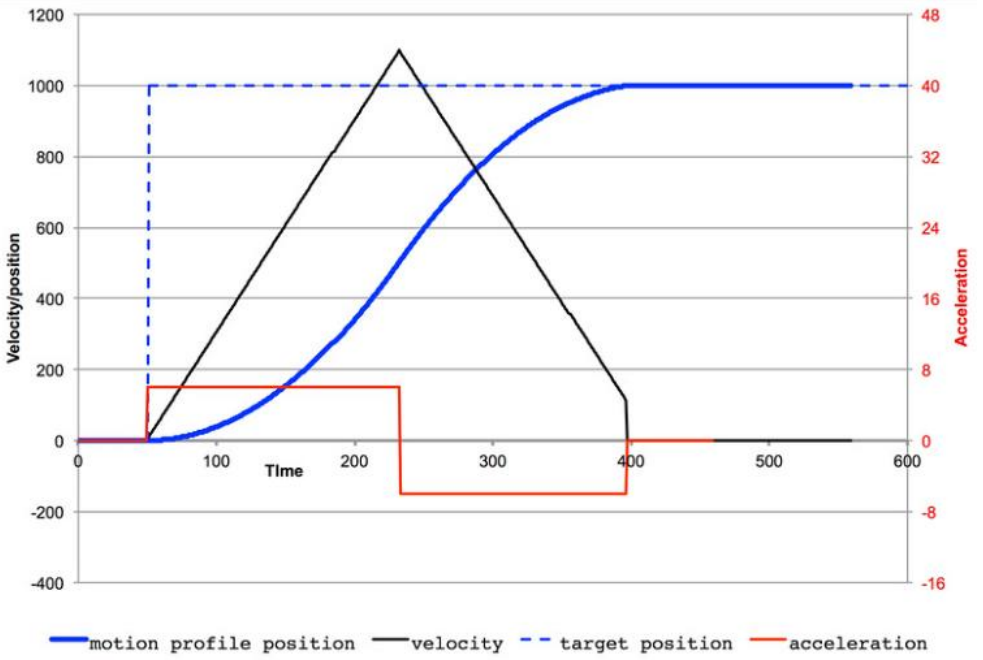
We consider

- **Triangular profile**
- **Trapezoidal motion profiles:**
- **S-curve**

2.1.Triangular motion profiles: For quick, point-to-point movements

The basic premise of a triangular move profile is to accelerate to a maximum velocity, then immediately decelerate, with acceleration and deceleration being equal in terms of both time and distance. In other words, if you want to move from here to there in the fastest time, you would use a triangular move profile. Determining any of the move variables—time, velocity, acceleration, or distance—is based on the geometry of the triangle. Below are some common equations for triangular move profiles.





Not very smooth, but allows for easy incorporation of joint velocity and acceleration limits.

NUS National University of Singapore

Triangular motion profiles:

$t \in [0, t_a]$: constant acceleration phase $\ddot{s} = a$

$$v = at$$

$$p = \frac{v_0 + v}{2} t = \frac{a}{2} t^2$$

$t \in (t_a, t_{a+d}]$: constant deceleration phase $\ddot{s} = -a$

Objective: find position profile, velocity profile and calculate acceleration.

Not very smooth, but allows for easy incorporation of joint velocity and acceleration limits.

Limitations:

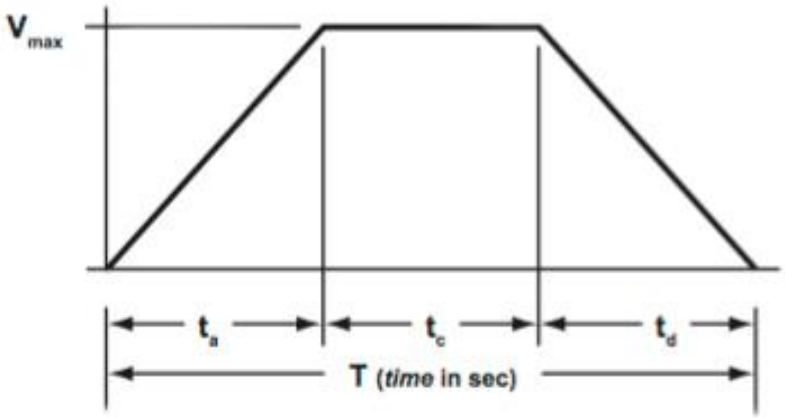
1) overall fastest option for getting from one point to the other,

2) Not smooth

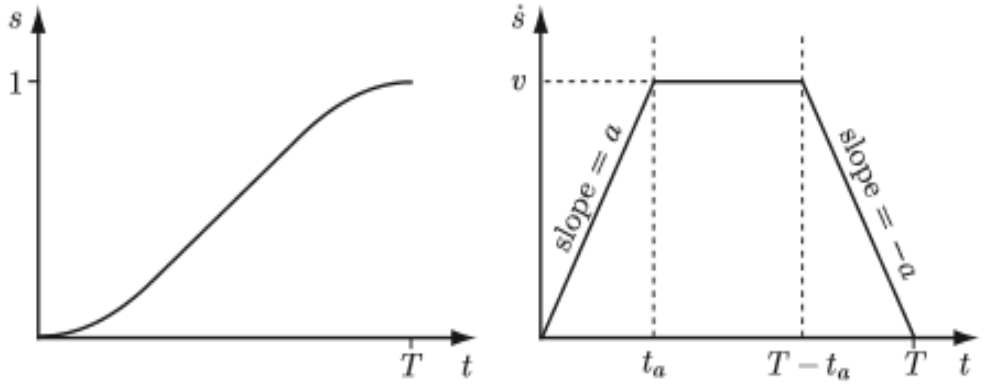
NUS National University of Singapore

2.2Trapezoidal move profile

A trapezoidal move profile is used when the application needs to accelerate to a maximum velocity and then travel at that velocity for a specified time or distance. Some common processes that use trapezoidal move profiles are machining, dispensing, and painting.



The simplest form of trapezoidal move profile, and the one used in the examples below, is the 1/3, 1/3, 1/3 profile. In this case, 1/3 of the time is used for accelerating, 1/3 is used for constant velocity, and 1/3 is used for decelerating. But if you understand the geometry of the move profile, which is essentially two triangles and a rectangle, you can calculate the necessary parameters regardless of whether the time segments are equal or not.



- Trapezoidal time scaling:
  - $t \in [0, t_a]$: constant acceleration phase $\ddot{s} = a$ $\quad v = at \quad p = \frac{a}{2}t^2$
  - $t \in (t_a, T - 2t_a]$: constant velocity phase $\dot{s} = v$ $\quad p = vt + \frac{a}{2}t_a^2$
  - $t \in (T - 2t_a, T]$: constant deceleration phase $\ddot{s} = -a$
- Not very smooth, but allows for easy incorporation of joint velocity and acceleration limits.

NUS National University of Singapore

Advantages:

For applications that benefit from a period of constant velocity, a trapezoidal move profile offers the advantage of having a lower maximum velocity than a triangular profile.

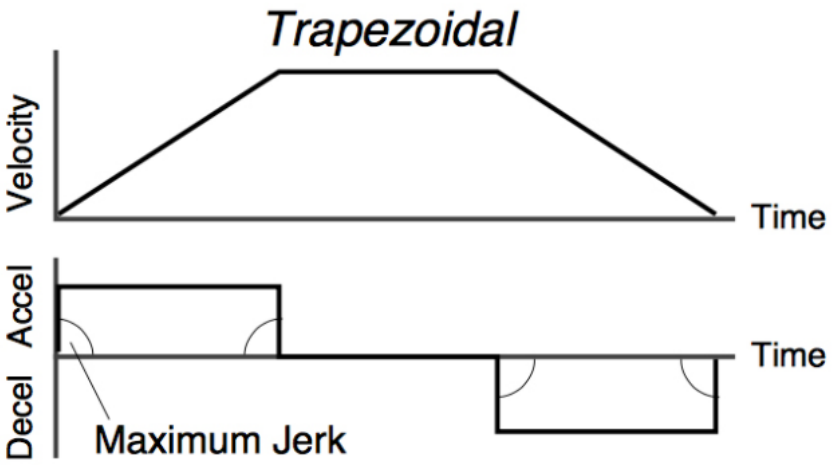
Limitations:

Trapezoidal profile causes discontinuities in acceleration at t=0,ta,ta+tc and T.

NUS National University of Singapore

2.3.Adding curves produces smoother motion

Despite underpinning the majority of motion control applications, neither triangular nor trapezoidal move profiles are ideal for motion systems due to a phenomenon known as “jerk.” Jerk is the rate of change of acceleration, and for trapezoidal and triangular move profiles, the initial acceleration and final deceleration occur instantly, meaning jerk is (theoretically) infinite

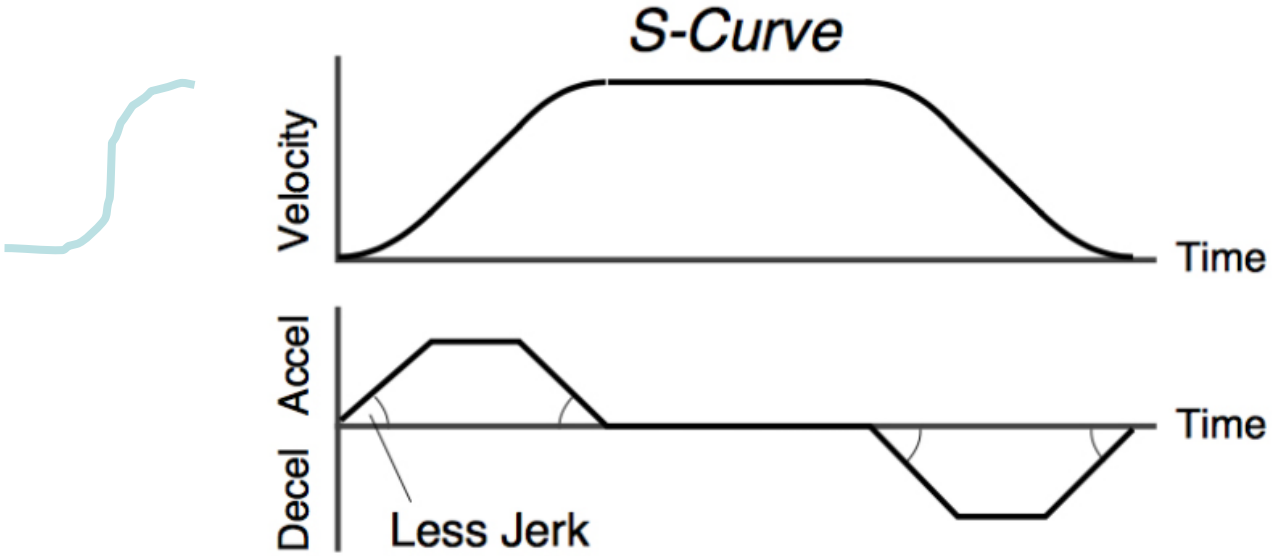


Standard triangular and trapezoidal (shown above) motion profiles require instant acceleration, which leads to (theoretically) infinite jerk.

Jerk is especially problematic for systems that require smooth, accurate motion because it causes vibrations that can reduce positioning accuracy and extend settling time.

NUS National University of Singapore

To reduce jerk, the beginnings and ends of the acceleration and deceleration phases of the move are smoothed into an “S” shape. This limits the rate of change of acceleration and deceleration (jerk) and produces smoother motion and more accurate positioning.



Smoothing the beginning and end of the acceleration and deceleration phases of motion – known as an S-curve motion profile – allows acceleration to increase and decrease over time, which reduces jerk.

An improved version is the S-curve. It consists of 7 phases.

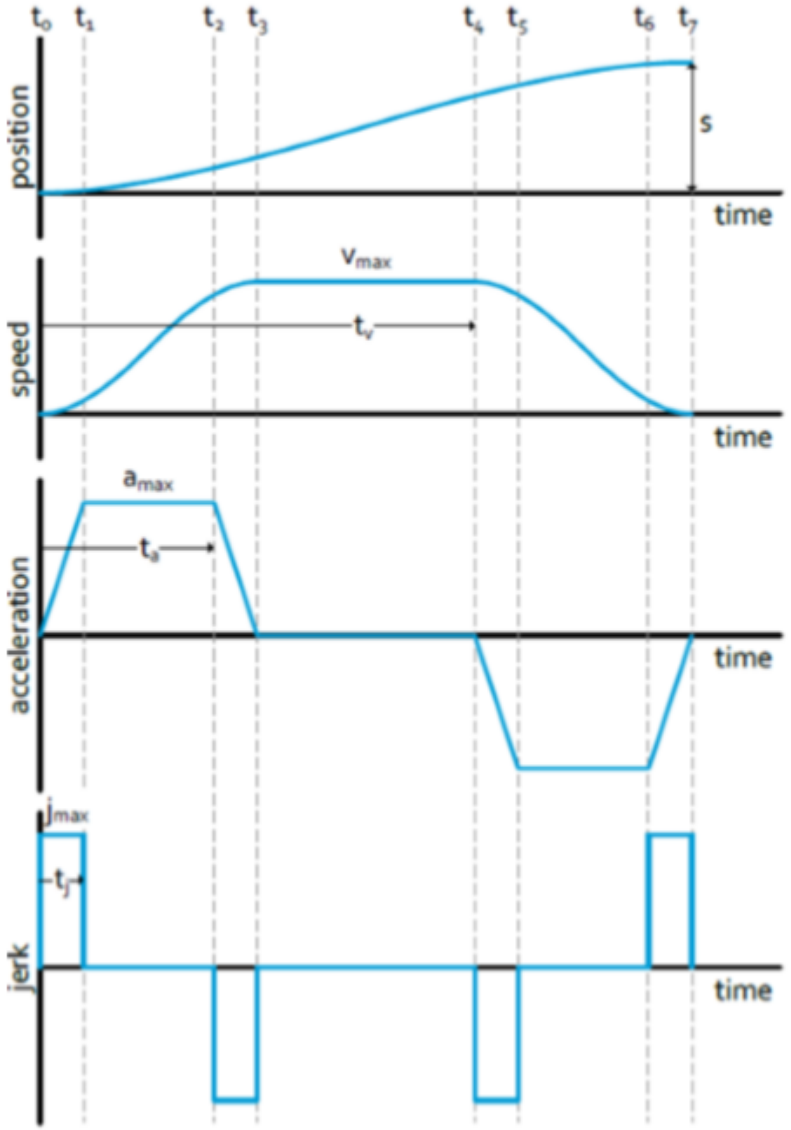
(1) constant jerk d3s/dt3 = J; (2) constant acceleration s = a;
(3) constant negative jerk −J;(4) coasting at constant v;
(5) constant negative jerk; (6) constant deceleration;
(7) constant positive jerk

S-curve is divided into seven segments. We use time segments to represent the curve: t1, t2,t3,t4,t5,t6,t7.

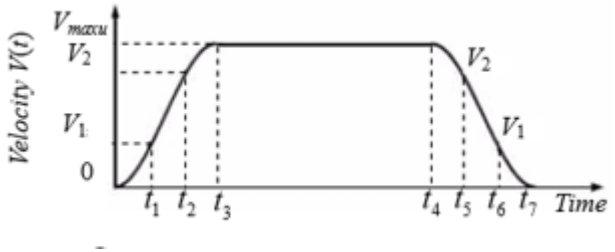
We have to give several parameters before giving the S-curve.

1) Displacement Xmax
2) Maximum speed v_{max}
3) Maximum acceleration a_{max}
4) Maximum jerk j_{max}
5) Seven time periods

Objective: plot position X, speed V and acceleration A profiles of each time segment.



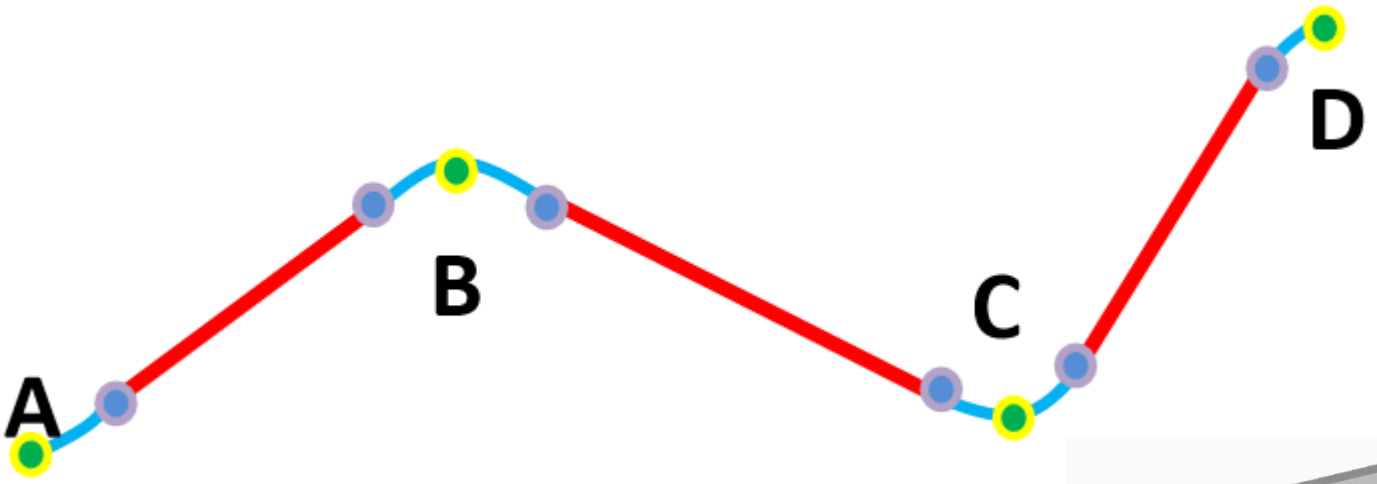
$$v_3 = v_{max}$$



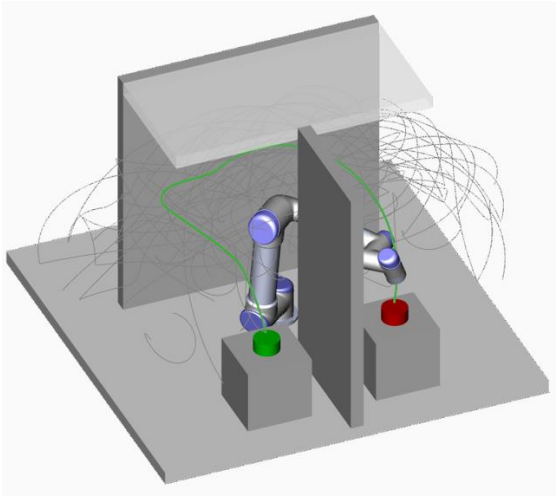
Motion parameters							
	jerk	acceleration	velocity		position		
$t_0 \ldots t_1$	j_{max}	$j_{max} \cdot (t - t_0)$	$\frac{1}{2} \cdot j_{max} \cdot (t - t_0)^2$		$\frac{1}{6} \cdot j_{max} \cdot (t - t_0)^3$		
$t_1 \ldots t_2$	0	$a_1 = a_2$	$v_1 + a_1 \cdot (t - t_1)$		$p_1 + v_1 \cdot (t - t_1) + \frac{1}{2} \cdot a_1 \cdot (t - t_1)^2$		
$t_2 \ldots t_3$	$-j_{max}$	$a_2 - j_{max} \cdot (t - t_2)$	$v_2 + a_2 \cdot (t - t_2) + \frac{1}{2} \cdot -j_{max} \cdot (t - t_2)^2$		$p_2 + v_2 \cdot (t - t_2) + \frac{1}{2} \cdot a_2 \cdot (t - t_2)^2 + \frac{1}{6} \cdot -j_{max} \cdot (t - t_2)^3$		
$t_3 \ldots t_4$	0	0	$v_3 = v_4$		$p_3 + v_3 \cdot (t - t_3)$		
$t_4 \ldots t_5$	$-j_{max}$	$-j_{max} \cdot (t - t_4)$	$v_4 + \frac{1}{2} \cdot -j_{max} \cdot (t - t_4)^2$		$p_4 + v_4 \cdot (t - t_4) + \frac{1}{6} \cdot -j_{max} \cdot (t - t_4)^3$		
$t_5 \ldots t_6$	0	$a_5 = a_6$	$v_5 - a_{max} \cdot (t - t_5)$		$p_5 + v_5 \cdot (t - t_5) + \frac{1}{2} \cdot a_5 \cdot (t - t_5)^2$		
$t_6 \ldots t_7$	j_{max}	$a_6 + j_{max} \cdot (t - t_6)$	$v_6 + a_6 \cdot (t - t_6) + \frac{1}{2} \cdot j_{max} \cdot (t - t_6)^2$		$p_6 + v_6 \cdot (t - t_6) + \frac{1}{2} \cdot a_6 \cdot (t - t_6)^2 + \frac{1}{6} \cdot j_{max} \cdot (t - t_6)^3$		
$t_1 = t_j$		$t_2 = t_a$	$t_3 = t_a + t_j$	$t_4 = t_v$	$t_5 = t_v + t_j$	$t_6 = t_v + t_a$	$t_7 = t_v + t_j + t_a$

NUS National University of Singapore

S-curve can generate a trajectory profile as follows

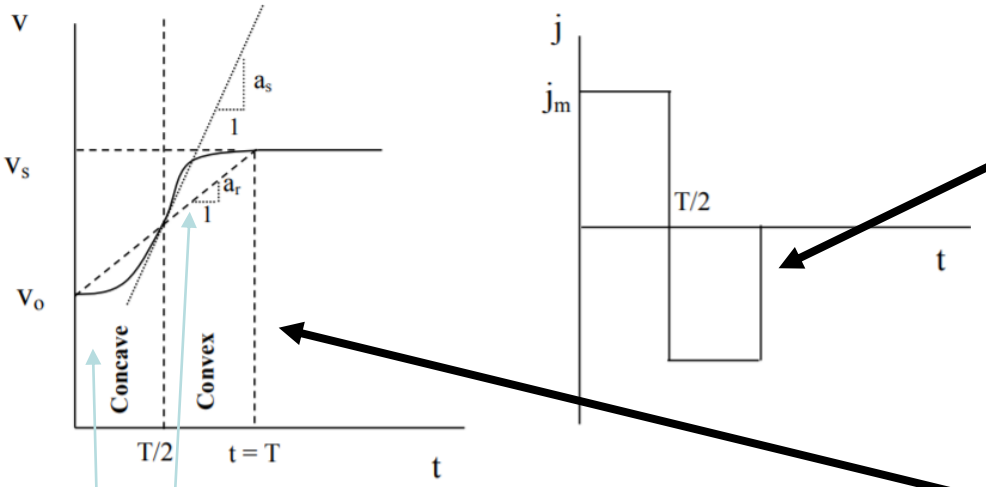


Limitations: Complex expression; cannot handle other constraints, for example, snap and environment



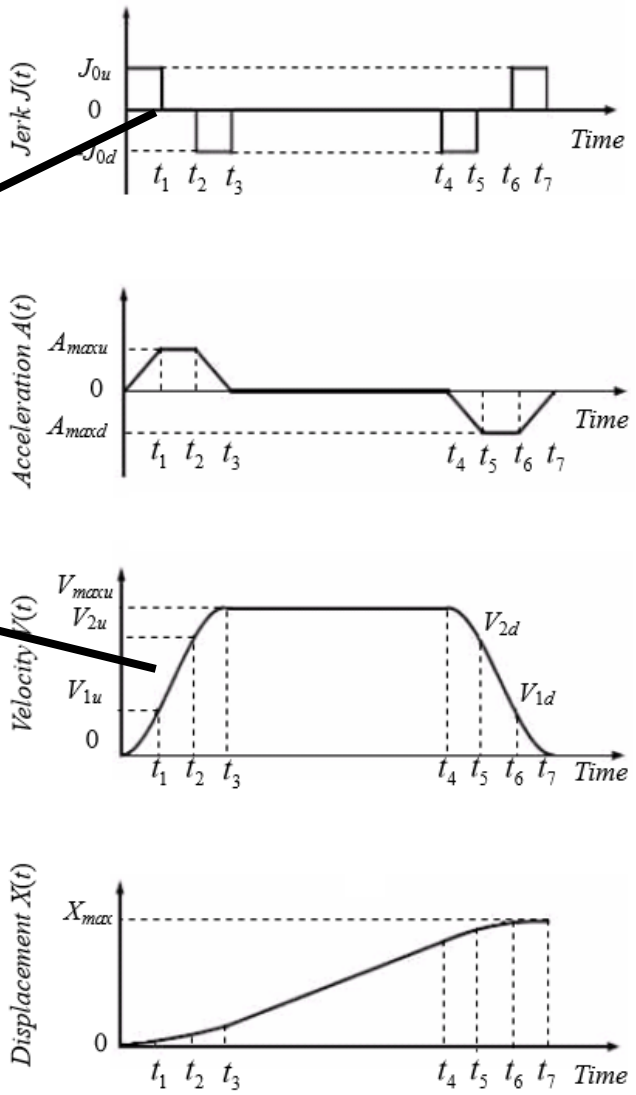
NUS National University of Singapore

3. Trajectory represented by polynomial



S-curve profile

$$v(t) = c_o + c_1 t + c_2 t^2$$



One polynomial or multiple polynomials?

3.1 Trajectory represented by one polynomial

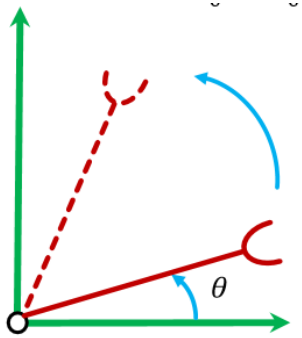
Can we use one polynomial to represent position, velocity and acceleration profiles?

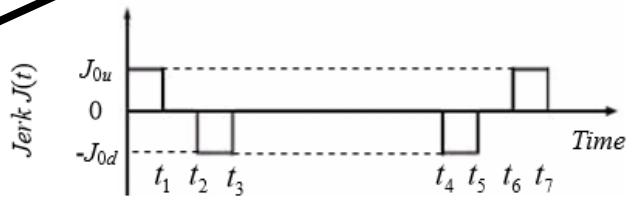
$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

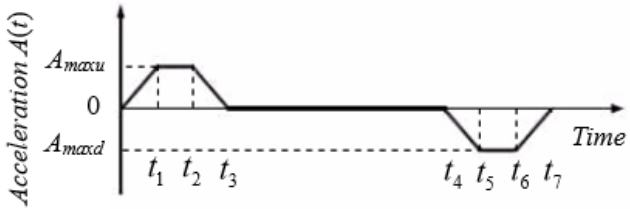
Joint velocity and acceleration along this path are

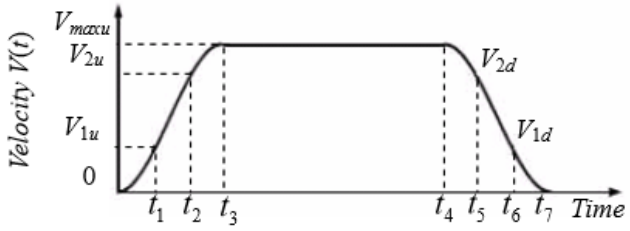
$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2$$

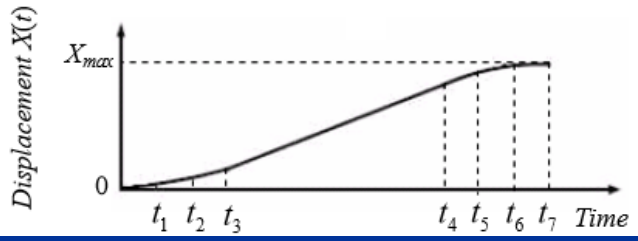
$$\ddot{\theta}(t) = 2a_2 + 6a_3 t$$











NUS National University of Singapore

Question: how do we get coefficients?

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

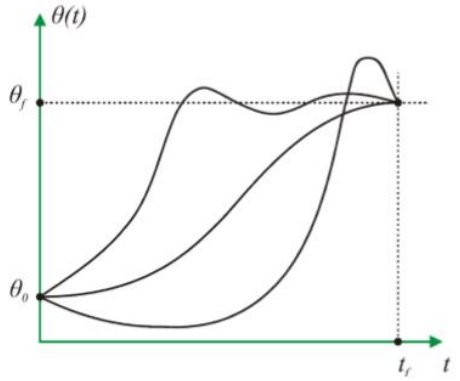
$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2$$

Initial and terminal conditions

- Initial value $\theta(0) = \theta_0$,
- Final value $\theta(T) = \theta_f$,
- Initial velocity $\dot{\theta}(0) = 0$,
- Final velocity $\dot{\theta}(T) = 0$.

Boundary conditions:

	Position	Velocity
$t = 0$	a	0
$t = T$	b	0



These four constraints can be satisfied by a polynomial of at least third degree (four coefficients)

Example:

A single-link robot with a rotary joint is motionless at $\theta = 15$ degrees. It is desired to move the joint in a smooth manner to $\theta = 75$ degrees in 3 seconds. Find the coefficients of a cubic that accomplishes this motion and brings the manipulator to rest at the goal. Plot the position, velocity, and acceleration of the joint as a function of time.

$$\theta(0) = 15.0 \quad , \quad T=3 \quad , \quad \theta(3) = 75$$

Boundary conditions:

	Position	Velocity
$t = 0$	a	0
$t = T$	b	0

$$\theta(0) = a_0 = 15.0$$

$$\theta(3) = a_0 + a_1 3 + a_2 3^2 + a_3 3^3 = 75 \Rightarrow 9a_2 + 27a_3 = 60$$

$$\dot{\theta}(0) = a_1 = 0$$

$$\dot{\theta}(3) = a_1 + 2a_2 3 + 3a_3 3^2 = 0 \Rightarrow a_2 = -\frac{9}{2} a_3$$

$$\begin{bmatrix} a \\ b \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ T^3 & T^2 & T & 1 \\ 0 & 0 & 1 & 0 \\ 3T^2 & 2T & 1 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}$$

$$a_0 = 15.0, \quad a_1 = 0.0, \quad a_2 = 20.0, \; a_3 = -4.44.$$

$$\theta(t) = 15.0 + 20.0t^2 - 4.44t^3,$$

$$\dot{\theta}(t) = 40.0t - 13.33t^2,$$

$$\ddot{\theta}(t) = 40.0 - 26.66t.$$

⇩

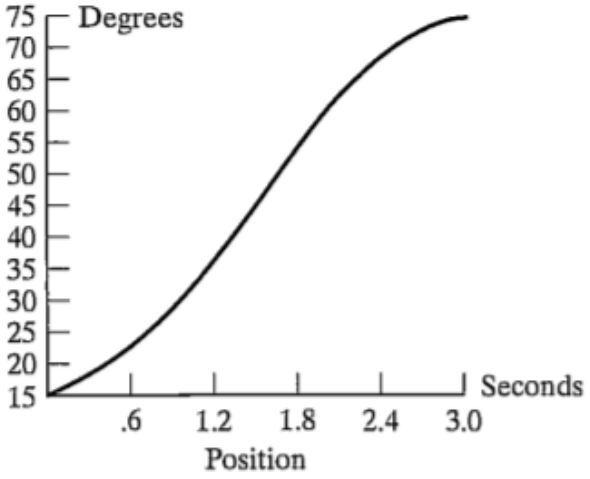
MATLAB ?

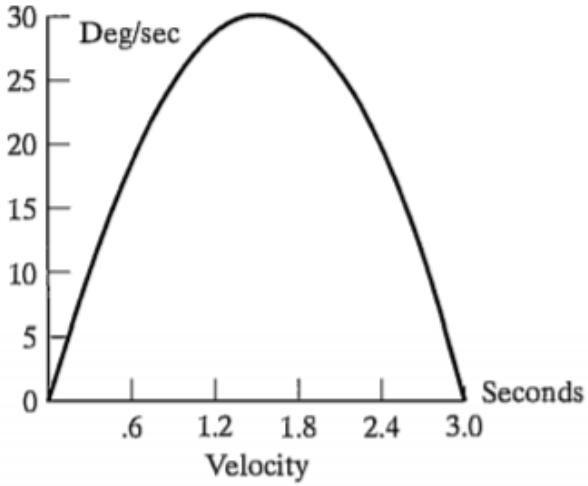
NUS National University of Singapore

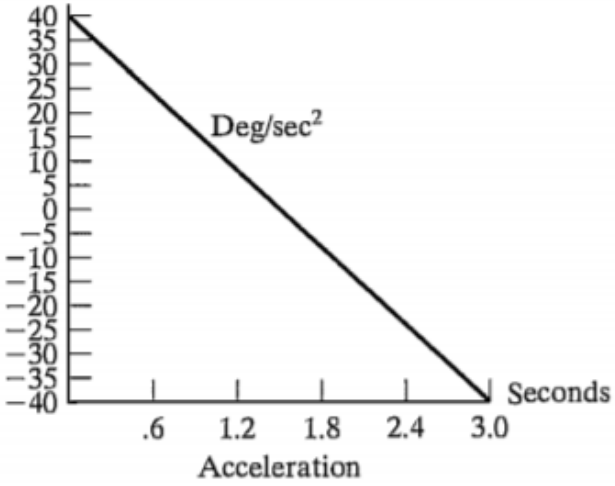
$$\theta(t) = 15.0 + 20.0t^2 - 4.44t^3,$$

$$\dot{\theta}(t) = 40.0t - 13.33t^2,$$

$$\ddot{\theta}(t) = 40.0 - 26.66t.$$



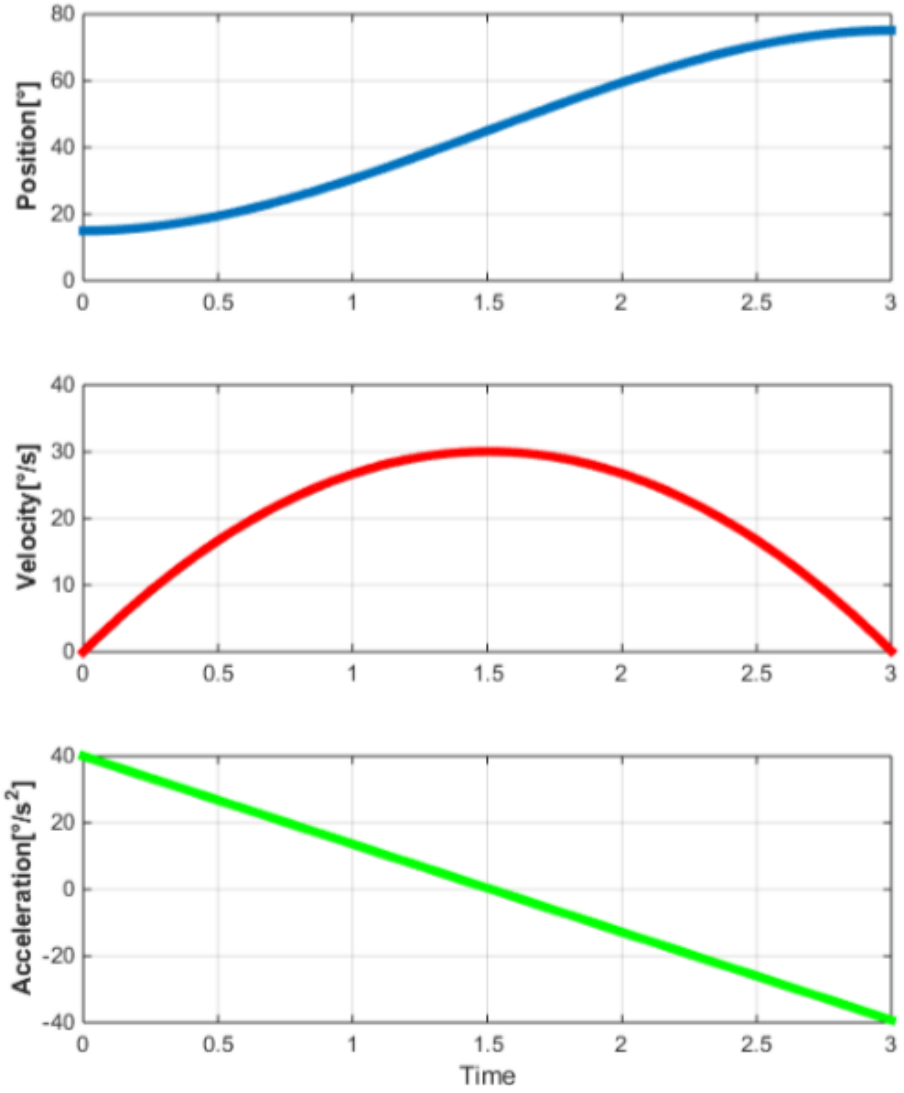




NUS National University of Singapore

MATLAB program

```
2
3 -     t = 0:0.01:3;
4
5 -     theta_t = 15 + 20.*t.*t - 4.44.*t.*t.*t;
6 -     theta_d_t = 40.*t - 13.33.*t.*t;
7 -     theta_2d_t = 40 - 26.6.*t;
8
9 -     subplot(3,1,1);
10 -    plot(t,theta_t); grid on;
11 -    ylabel('Position[°]');
12
13 -    subplot(3,1,2);
14 -    plot(t,theta_d_t); grid on;
15 -    ylabel('Velocity[°/s]');
16
17 -    subplot(3,1,3);
18 -    plot(t,theta_2d_t); grid on;
19 -    ylabel('Acceleration[°/s^2]');
20
21
```



NUS National University of Singapore

Summary:

For a third-order polynomial, we have four coefficients. Thus, we need to have four boundary conditions, i.e., four equations for getting the values of coefficients.

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \qquad \dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2$$

Boundary conditions:

	Position	Velocity
$t = 0$	a	0
$t = T$	b	0

List equations

$$\begin{bmatrix} a \\ b \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ T^3 & T^2 & T & 1 \\ 0 & 0 & 1 & 0 \\ 3T^2 & 2T & 1 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} \Rightarrow \; ?$$

NUS National University of Singapore

Question: can we use a polynomial more than 3rd order degree?

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5,$$

Now we have six coefficients.
But the four boundary constraints are not enough for obtaining all coefficients. In this case, we need to have more boundary conditions.

Can we use a simpler one, for example, first order ?

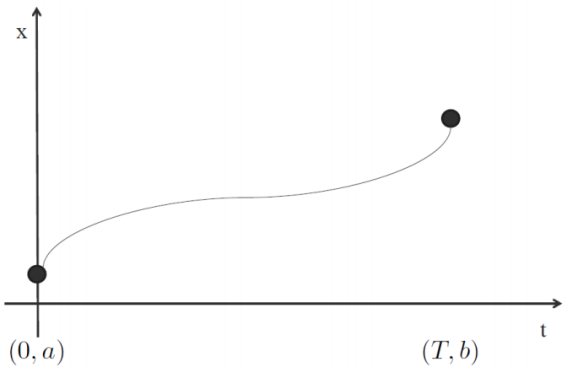
Higher order polynomial model $x(t) = p_0 + p_1 t + p_2 t^2 + \cdots + p_n t^n = \sum_{i=0}^{n} p_i t^i$

Position $x(t) = p_0 + p_1 t + p_2 t^2 + p_3 t^3 + p_4 t^4 + p_5 t^5 = \sum_{i=0}^{5} p_i t^i$

Speed $\dot{x}(t) = p_1 + 2\,p_2 t + 3\,p_3 t^2 + 4 p_4 t^3 + 5\,p_5 t^4$

Acceleration $\ddot{x}(t) = 2\,p_2 + 6\,p_3 t + 12 p_4 t^2 + 20\,p_5 t^3$

How many parameters?



Boundary condition

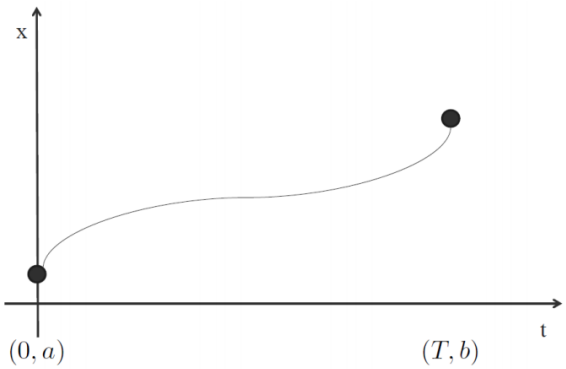
	Position	*Velocity*	*Acceleration*
$t = 0$	a	0	0
$t = T$	b	0	0

$$\begin{bmatrix} a \\ b \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ T^5 & T^4 & T^3 & T^2 & T & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5T^4 & 4T^3 & 3T^2 & 2T & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 20T^3 & 12T^2 & 6T & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_5 \\ p_4 \\ p_3 \\ p_2 \\ p_1 \\ p_0 \end{bmatrix}$$

Six coefficients have six boundary conditions. We have six linear equations.

Higher order polynomial model

$$x(t) = p_0 + p_1 t + p_2 t^2 + \cdots + p_n t^n = \sum_{i=0}^{n} p_i t^i \quad (\boldsymbol{n = 5})$$



Boundary condition

	Position	***Velocity***	***Acceleration***
$t = 0$	a	v_0	a_0
$t = T$	b	v_T	a_T

$$\Updownarrow$$

$$\begin{bmatrix} a \\ b \\ v_0 \\ v_T \\ a_0 \\ a_T \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ T^5 & T^4 & T^3 & T^2 & T & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5T^4 & 4T^3 & 3T^2 & 2T & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 20T^3 & 12T^2 & 6T & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_5 \\ p_4 \\ p_3 \\ p_2 \\ p_1 \\ p_0 \end{bmatrix}$$

Summary:

For 5th order polynomial, we have boundary conditions

1. Initial Position
2. Final Position
3. Initial Velocity
4. Final Velocity
5. Initial Acceleration
6. Final Acceleration

Continuous, smooth and comfortable
How about 7th order polynomial?

Higher order polynomial model

$$x(t) = p_0 + p_1 t + p_2 t^2 + \cdots + p_n t^n = \sum_{i=0}^{n} p_i t^i$$

where there are n+1 coefficients. Define coefficient vector

$$p = [p_0, p_1, \ldots, p_n]^T$$

Then, position trajectory can be expressed as

$$p(t) = [1, t, t^2, \ldots, t^n] \cdot p$$

We also obtain velocity, acceleration , jerk and snap trajectories

$$v(t) = p'(t) = [0, 1, 2t, 3t^2, 4t^3, \ldots, nt^{n-1}] \cdot p$$

$$a(t) = p''(t) = [0, 0, 2, 6t, 12t^2, \ldots, n(n-1)t^{n-2}] \cdot p$$

$$jerk(t) = p^{(3)}(t) = [0, 0, 0, 6, 24t, \ldots, \frac{n!}{(n-3!)} t^{n-3}] \cdot p$$

$$snap(t) = p^{(4)}(t) = [0, 0, 0, 0, 24, \ldots, \frac{n!}{(n-4!)} t^{n-4}] \cdot p$$

NUS National University of Singapore

More coefficients, more boundary conditions.

For example, for n+1 coefficients, we need to have n+1 boundary conditions，i.e., equation constraints

	position	speed	acceleration	jerk	snap	……
t=0	a	v_0	a_0	j_0	s_0	…….
t=T	b	v_T	a_T	j_T	s_T	……..

Remember to give the time first!!!

Total distance/average speed

NUS National University of Singapore

Calculate parameters and conditions, depending on the polynomial you selected

- **Initial and terminal points: position,velocity, acceleration, and jerk, i.e., pvaj**

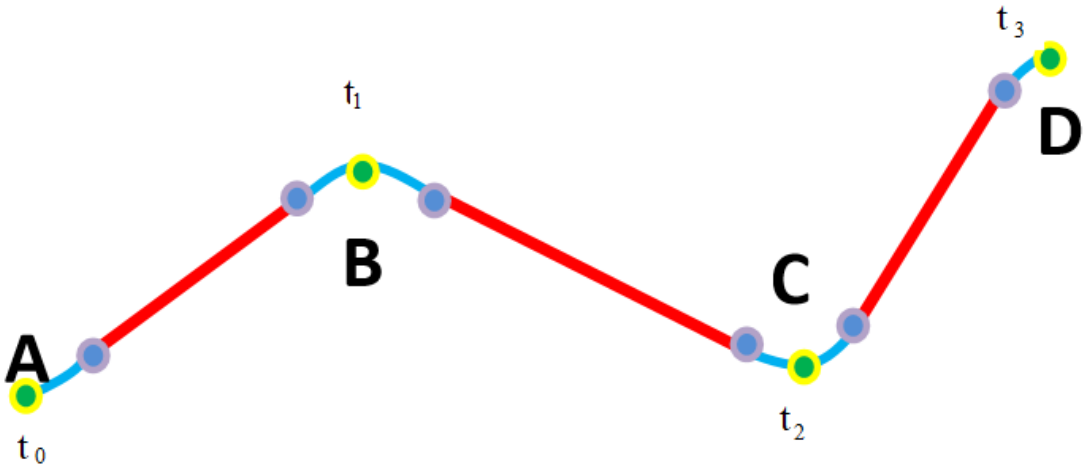
4(initial)+4(terminal)

NUS National University of Singapore

3.2 Multiple polynomials

From A->B, we can use one polynomial. How about A→B→C→D? Can we use one polynomial to express it?

Usually, we use multiple polynomials

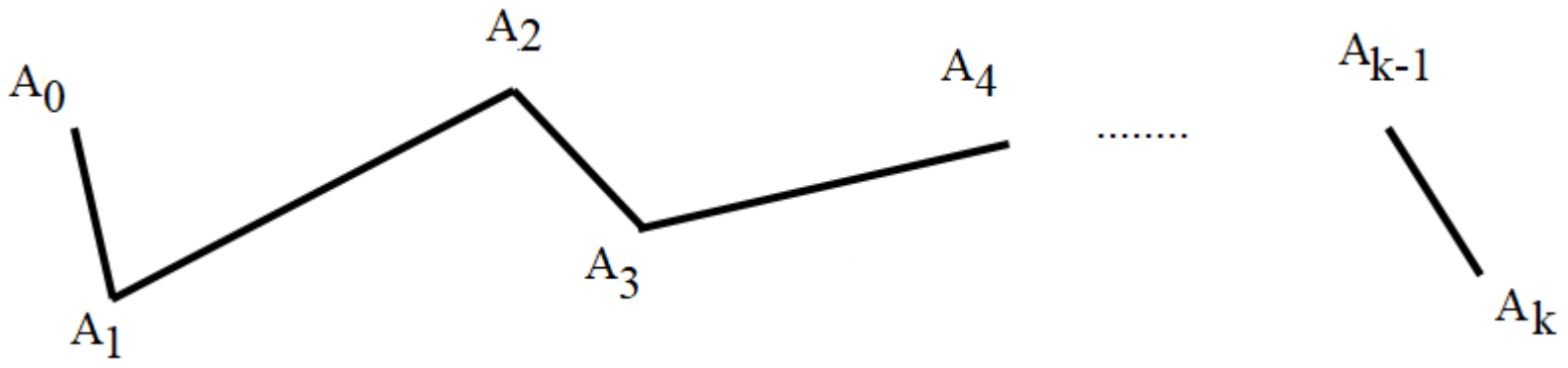


$$p(t) = \begin{cases} [1, t, t^2, \ldots, t^n] \cdot p_1 & t_0 \leq t < t_1 \\ [1, t, t^2, \ldots, t^n] \cdot p_2 & t_1 \leq t < t_2 \\ [1, t, t^2, \ldots, t^n] \cdot p_3 & t_2 \leq t < t_3 \end{cases}$$

NUS National University of Singapore

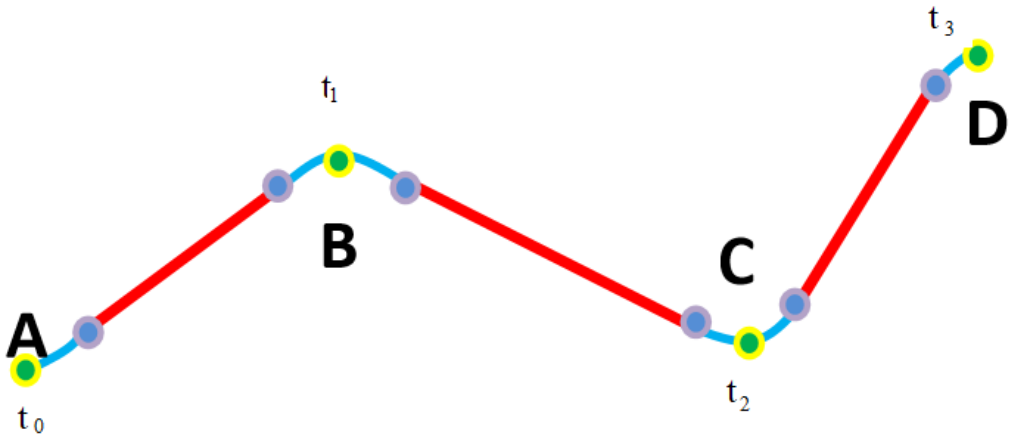
For k segments, we have k polynomials

$$p(t) = \begin{cases} [1, t, t^2, \ldots, t^n] \cdot p_1 & t_0 \leq t < t_1 \\ [1, t, t^2, \ldots, t^n] \cdot p_2 & t_1 \leq t < t_2 \\ \ldots & \\ [1, t, t^2, \ldots, t^n] \cdot p_k & t_{k-1} \leq t < t_k \end{cases}$$



NUS National University of Singapore

You may argue with me. Why we need to learn the multiple polynomials?



A->B B→C, C→D, use the same boundary condition v=0, a=0.

Sometimes, this may take longer time to reach target or exceed the limits.

NUS National University of Singapore

1)Time intervals: we know the time interval at each segment.

2)Boundary conditions

- **Initial values**
- **Terminal values**
- **For the relationship between two adjacent segments, the continuity of position, velocity, acceleration, and jerk can also be formulated as equality constraint equation, i.e., pvaj must be equal**

For simplicity, all polynomials should have the same order.

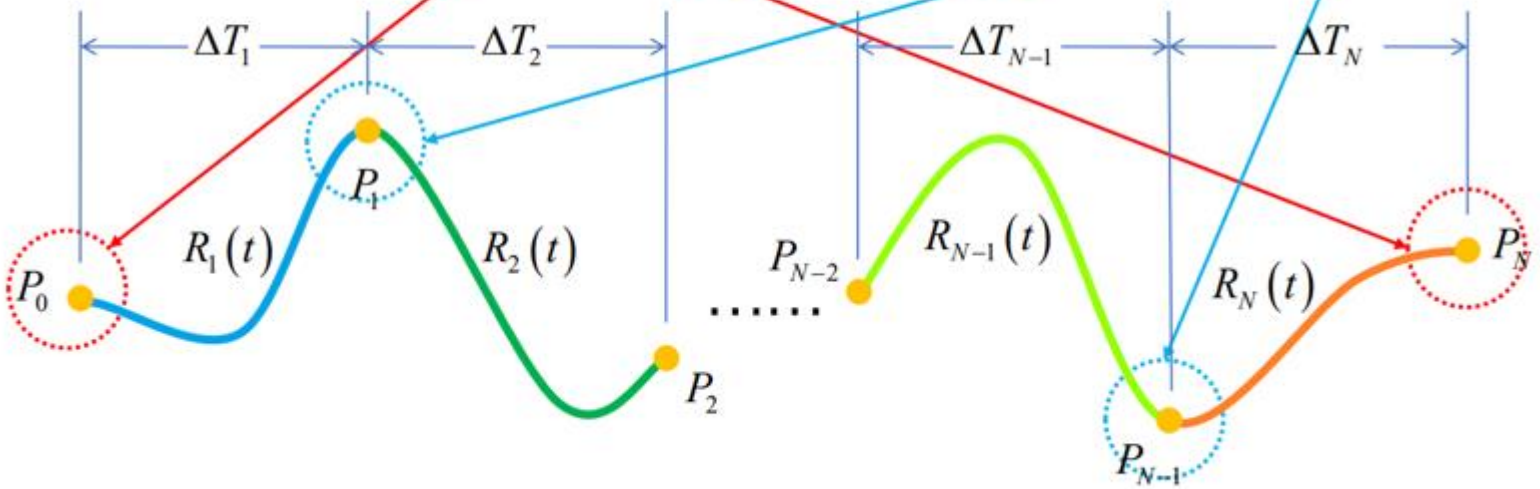
Initial values
Terminal values

For the relationship between 2 adjacent segments, the continuity of position, velocity and acceleration can also be formulated as equality constraint equation.

- Derivative constraints:

$$\begin{cases} f_j^{(k)}(T_0) = x_{0,j}^{(k)} \\ f_j^{(k)}(T_j) = x_{T,j}^{(k)} \end{cases}$$

- Continuity constraints: $f_j^{(k)}(T_j) = f_{j+1}^{(k)}(T_j)$



For example, continuity constraints:

$$[1, t_1, t_1^2, \ldots, t_1^n] \cdot p_1 = [1, t_1, t_1^2, \ldots, t_1^n] \cdot p_2 \qquad t = t_1$$

$$[1, t_1, t_1^2, \ldots, t_1^n] \cdot p_1 - [1, t_1, t_1^2, \ldots, t_1^n] \cdot p_2 = 0 \qquad t = t_1$$

NUS National University of Singapore

- **Derivative constraints**

$$\text{position} : [1, t_0, t_0^2, \ldots, t_0^n, \underbrace{0 \ldots 0}_{(k-1)(n+1)}]p = p_0$$

$$\text{velocity} : [0, 1, 2t_0, \ldots, nt_0^{n-1}, \underbrace{0 \ldots 0}_{(k-1)(n+1)}]p = v_0$$

$$\text{acceleration} : [0, 0, 2, \ldots, n(n-1)t_0^{n-2}, \underbrace{0 \ldots 0}_{(k-1)(n+1)}]p = a_0$$

$$\text{Final position} : [1, t_\text{T}, t_\text{T}^2, \ldots, t_\text{T}^n, \underbrace{0 \ldots 0}_{(k-1)(n+1)}]p = p_\text{T}$$

$$\text{Final velocity} : [0, 1, 2t_\text{T}, \ldots, nt_\text{T}^{n-1}, \underbrace{0 \ldots 0}_{(k-1)(n+1)}]p = v_\text{T}$$

$$\text{Final acceleration} : [0, 0, 2, \ldots, n(n-1)t_\text{T}^{n-2}, \underbrace{0 \ldots 0}_{(k-1)(n+1)}]p = a_\text{T}$$

- **Continuity constraints**

$$[\underbrace{0 \ldots 0}_{(i-1)(n+1)}, 1, t_i, t_i^2, \ldots, t_i^n, -1, -t_i, -t_i^2, \ldots, -t_i^n, \underbrace{0 \ldots 0}_{(k-i-1)(n+1)}]p = 0$$

NUS National University of Singapore

Combining equation constraints

$$
\begin{bmatrix}
1, t_0, t_0^2, \cdots, t_0^n, 0_{(k-1)(n-1)} \\
0,1,2t_i, \cdots, nt_i^{n-1}, 0_{(k-1)(n-1)} \\
0,0,2, \cdots, n(n-1)t_i^{n-2}, 0_{(k-1)(n-1)} \\
\vdots \\
0_{1(i-1)(n+1)}, 1, t_i, t_i^2, \cdots, t_i^n, 0_{(k-i)(n+1)} \\
\vdots \\
0_{(k-1)(n+1)}, 1, t_k, t_k^2, \cdots, t_k^n \\
\vdots \\
0_{(k-1)(n+1)}, 1, t_k, t_k^2, \cdots, t_k^n \\
0_{(k-1)(n+1)}, 0,1, 2t_k, \cdots, nt_k^{n-1} \\
0_{(k-1)(n+1)}, 0,0,2, \cdots, n(n-1)t_k^{n-2} \\
0_{(i-1)(n+1)}, 1, t_i, t_i^2, \cdots, t_i^n, -1, -t_i, -t_i^2, \cdots, -t_i^n, 0_{(k-i-1)(n+1)} \\
0_{(i-1)(n+1)}, 0,1,2t_i, \cdots, nt_i^{n-1}, 0, -1, -2t_i, \cdots, -nt_i^{n-1}, 0_{(k-i-1)(n+1)} \\
0_{(i-1)(n+1)+2}, 2, \cdots, \frac{n!}{(n-2)!}t_i^{n-2}, 0,0, -2, \cdots, -\frac{n!}{(n-2)!}t_i^{n-2}, 0_{(k-i-1)(n+1)}
\end{bmatrix}
p = \begin{bmatrix} p_0 \\ v_0 \\ a_0 \\ \vdots \\ p_i \\ \vdots \\ p_k \\ v_k \\ a_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}
$$

Derivative constraints

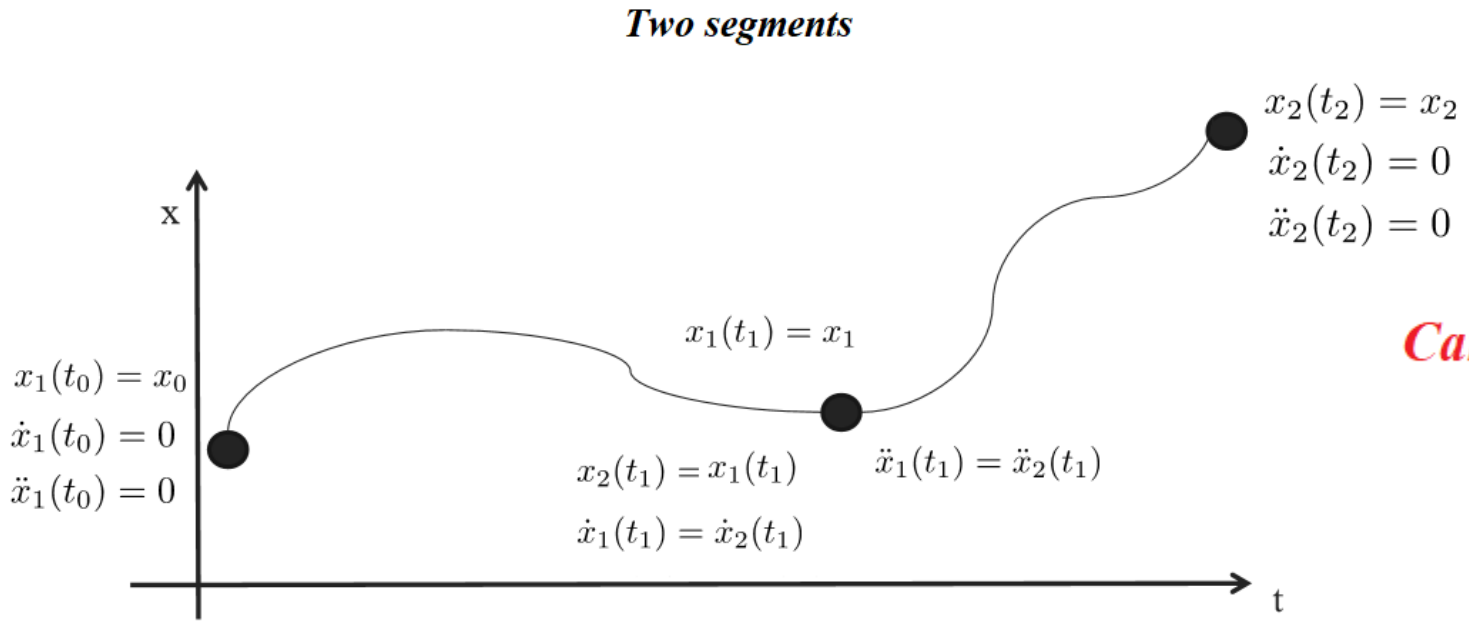
Continuity constraints

$$A_{eq}p = b_{eq}$$

Estimate number of equation constraints

- **Equation constrains=3 (initial pva) +k-1 (middle p) +3(terminal pva) +3(k-1) (middle continuity pva) =4k+2, where k represents polynomials (k segments)**
- **This implies that we can solve 4k+2 parameters.**

k=2 →equation constraints=3(initial)+3(terminal)+1(k-1)+3x1=10



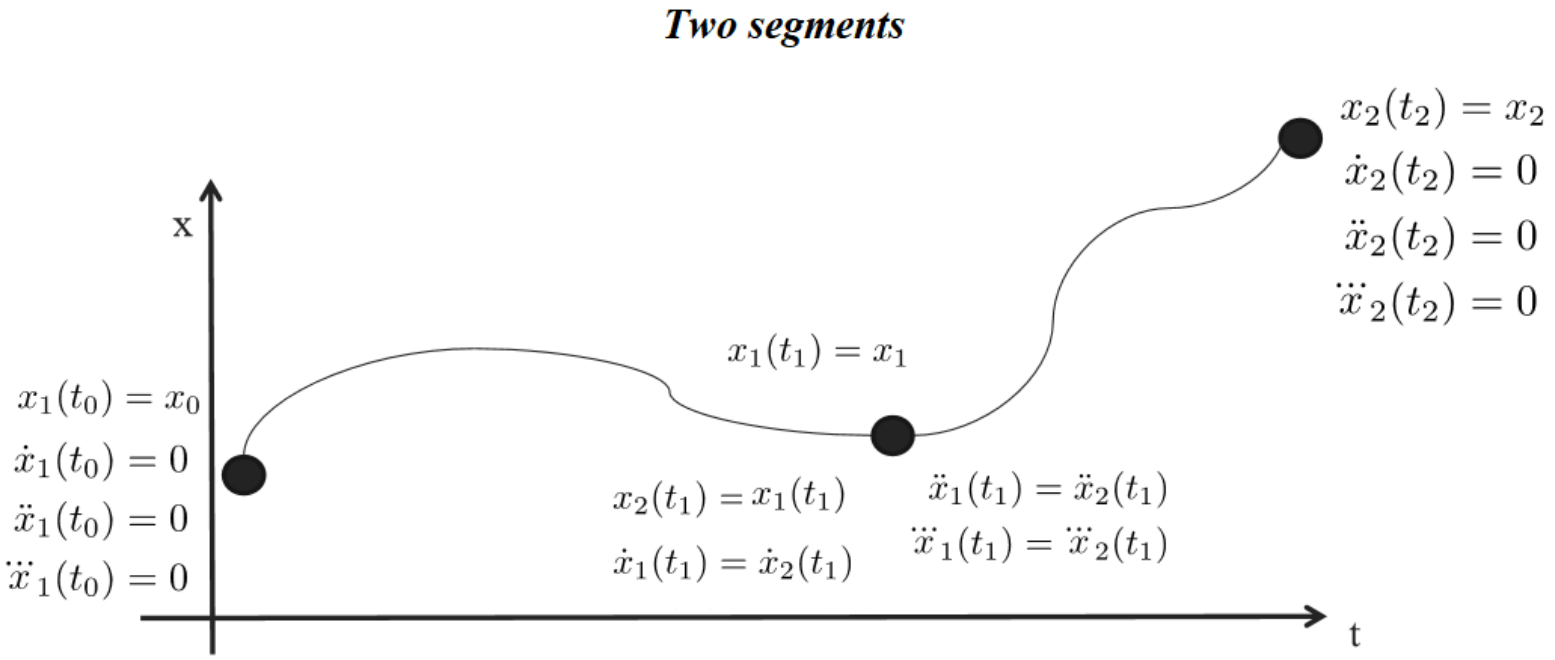
Can we express like

$$x_1(t_1) = x_1$$

$$x_2(t_1) = x_1$$

If we consider pvaj, how many number of equation constraints?

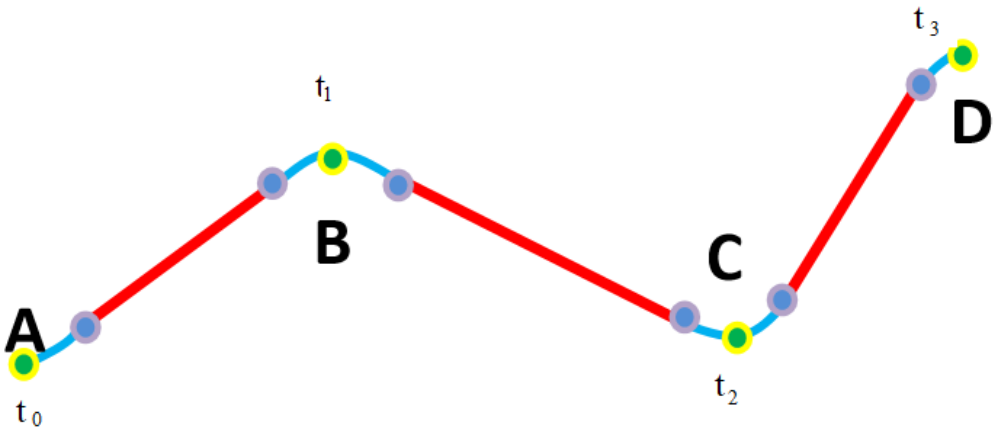
- **equation constraints=4+4+1+4=13**



NUS National University of Singapore

Several issues in polynomial:

- Time is allocated according to the path distance. It is obtained by totDist/avgSped
- Order number (we should have the same order number in all polynomials)
- Boundary conditions



NUS National University of Singapore

Limitations:

- Boundary conditions are set to zero. It is too conservative. Sometimes this causes the velocity or acceleration exceeding the limits.
- You have to check if the velocity and acceleration or jerk exceed the limits.

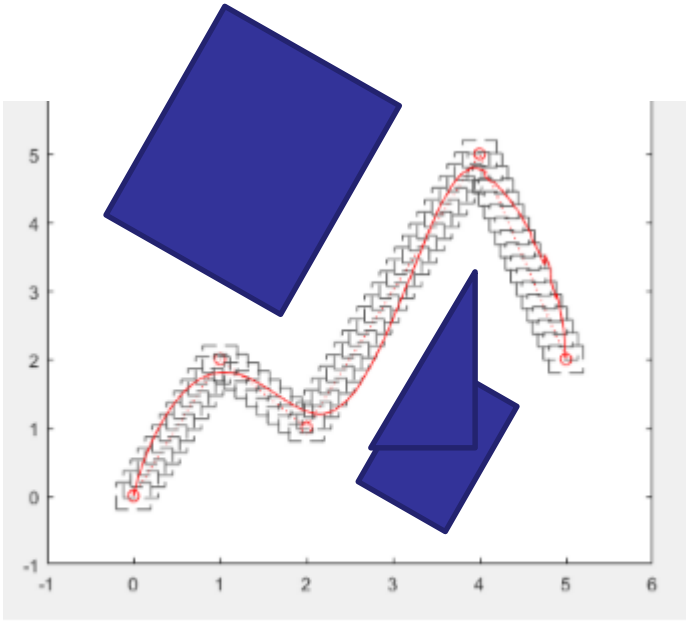
3.3 Trajectory optimization

- **More constraints,**

position and velocity

$$x_{min} \leq x \leq x_{max}$$

$$v_{min} \leq v \leq v_{max}$$



Minimum snap result with corridor

- **Objective optimizations**

$$minimum\ snap: \quad \min f(p) = \min \int (p^{(4)}(t))^2 dt$$

$$minimum\ jerk: \quad \min f(p) = \min \int (p^{(3)}(t))^2 dt$$

$$minimum\ acce: \quad \min f(p) = \min \int (p^{(2)}(t))^2 dt$$

?

NUS National University of Singapore

Key point:

The problem is formulated as following

$$\min f(p)$$ ⟸ Acceleration , jerk or snap

$$s.t. \quad A_{eq}p = b_{eq},$$ ⟸ Equation constraints

$$A_{ieq}p \leq b_{ieq}$$ ⟸ Inequality constraints

This is quadratic programming (QP)

Boundary conditions ⟹ Equation constraints

$$x_{min} \leq x \leq x_{max}$$
$$v_{min} \leq v \leq v_{max}$$
⟹ Inequality constraints

In this module,

- We will not concentrate on optimization method. This is beyond this module.
- The main purpose of QP is to use it in trajectory generation.

NUS National University of Singapore

A QP example

This example demonstrates a specific QP problem:

$$\min f(x) = 3{x_1}^2 + {x_2}^2 + 2x_1x_2 + x_1 + 6x_2 + 2$$
$$\text{s.t. } 2x_1 + 3x_2 \geq 4$$
$$x_1, x_2 \geq 0$$

Rewrite the constraints.

$$g_1 = -2x_1 - 3x_2 + 4 \leq 0$$
$$g_2 = -x_1 \leq 0$$
$$g_3 = -x_2 \leq 0$$

For example, we consider to minimizing jerk, where 5th order polynomial

$s(t) \in [0, t_f]$ ⟶ $s(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \ldots + \alpha_n t^n + \ldots$

Jerk $= \dddot{s}(t)$

$s(t) = \sum_{n=0}^{\infty} \alpha_n t^n$

Total (squared) Jerk $= \int_0^{t_f} \dddot{s}(t)^2 dt$

Using THIS form of s

Minimize this!

And incorporating THIS realization

$\frac{d^m s}{dt^m} = 0 \quad \forall m \geq 6$

$\alpha_6, \alpha_7, \ldots = 0$

MINIMUM 1D JERK TRAJECTORIES

$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$

https://blog.csdn.net/luteresa

NUS National University of Singapore

Minimum Snap (usually, 7th order)

$$\min \int_0^T (p^{(4)}(t))^2 \mathrm{d}t$$

$$= \min \sum_{i=1}^{k} \int_{t_{i-1}}^{t_i} (p^{(4)}(t))^2 \mathrm{d}t$$

$$= \min \sum_{i=1}^{k} \int_{t_{i-1}}^{t_i} ([0,0,0,0,24,\ldots,\frac{n!}{(n-4)!}t^{n-4}] \cdot p)^T [0,0,0,0,24,\ldots,$$

$$\frac{n!}{(n-4)!}t^{n-4}] \cdot p \, \mathrm{d}t$$

$$= \min \sum_{i=1}^{k} p^T \int_{t_{i-1}}^{t_i} [0,0,0,0,24,\ldots,\frac{n!}{(n-4)!}t^{n-4}]^T [0,0,0,0,24,\ldots,$$

$$\frac{n!}{(n-4)!}t^{n-4}] \, \mathrm{d}t \, p$$

NUS National University of Singapore

Minimum Snap

$$\min \int_0^T (p^{(4)}(t))^2 \, \mathrm{d}t \quad = \min \sum_{i=1}^{k} p^T Q_i p$$

where

$$Q_i = \int_{t_{i-1}}^{t_i} [0,0,0,0,24,\ldots,\frac{n!}{(n-4!)}t^{n-4}]^T [0,0,0,0,24,\ldots,\frac{n!}{(n-4!)}t^{n-4}] \, \mathrm{d}t$$

$$= \begin{bmatrix} 0_{4\times 4} & 0_{4\times(n-3)} \\ 0_{(n-3)\times 4} & \frac{r!}{(r-4)!}\frac{c!}{(c-4)!}\frac{1}{(r-4)+(c-4)+1}(t_i^{(r+c-7)} - t_{i-1}^{(r+c-7)}) \end{bmatrix}$$

$$Q = \begin{bmatrix} Q_1 & & & \\ & Q_2 & & \\ & & \ddots & \\ & & & Q_k \end{bmatrix}$$

NUS National University of Singapore

Minimum Snap

$$\min \int_0^T (p^{(4)}(t))^2 \, \mathrm{d}t \quad = \quad \min p^T Q p$$

$$Q = \begin{bmatrix} Q_1 & & & \\ & Q_2 & & \\ & & \ddots & \\ & & & Q_k \end{bmatrix}$$

Equation Constraints:

$$A_{eq} p = b_{eq}$$

NUS National University of Singapore

For inequality constraints, it is more complex.

$$A_{ieq}p \leq b_{ieq}$$

We give one example

position constraint

$$\boldsymbol{x_{min}} \leq [1, t, t^2, \ldots, t^n, \underbrace{0 \ldots 0}_{(k-1)(n+1)}]p \leq \boldsymbol{x_{max}}$$

$$[1, t, t^2, \ldots, t^n, \underbrace{0 \ldots 0}_{(k-1)(n+1)}]p \leq \boldsymbol{x_{max}}$$

$$-[1, t, t^2, \ldots, t^n, \underbrace{0 \ldots 0}_{(k-1)(n+1)}]p \leq -\boldsymbol{x_{min}}$$

$$\Rightarrow \begin{bmatrix} 1, & t, & t^2, \ldots, & t^n, & 0 \ldots 0 \\ -1, & -t, & -t^2, \ldots, & -t^n, & 0 \ldots 0 \end{bmatrix} p \leq \begin{bmatrix} \boldsymbol{x_{max}} \\ -\boldsymbol{x_{min}} \end{bmatrix}$$

NUS National University of Singapore

Example:

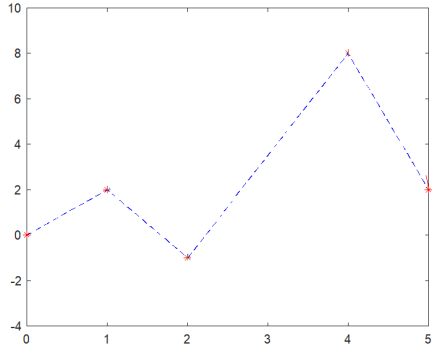
A vehicle has obtained five Waypoints from A*, `Waypts(x,y) = [(0,0);(1,2);(2,-1);(4,8);(5,2)].` It is desire to move from the initial point (0,0) to the goal (5,2) in 5s. Please use minimum snap to find the polynomial and plot position, velocity and acceleration profiles. Boundary conditions

```
Initial: v0 = [0,0];a0 = [0,0];
Terminal:v1 = [0,0];a1 = [0,0];
```

We use 7th order polynomial and the following optimization

$$\min \int_0^T (p^{(4)}(t))^2 \mathrm{d}t$$

$$A_{eq}p = b_{eq}$$



We use 7th order polynomials for each axis for each segment.

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 + a_6 t^6 + a_7 t^7$$

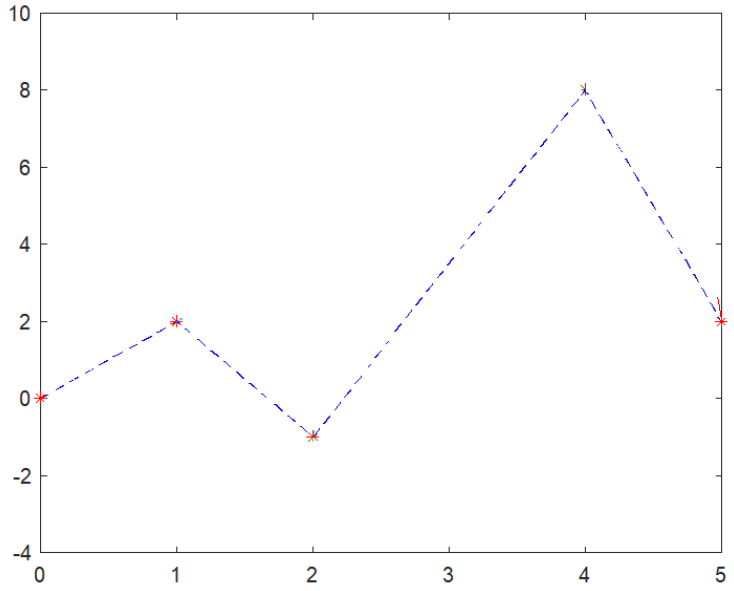
$$y(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5 + b_6 t^6 + b_7 t^7$$

To minimize the following cost

$$\min \int_0^T (p^{(4)}(t))^2 \, dt$$

$$= \min \sum_{i=1}^{4} \int_{t_{i-1}}^{t_i} ([0,0,0,0,24,\ldots,\frac{7!}{(7-4)!}t^{7-4}] \cdot p)^T [0,0,0,0,24,\ldots, \frac{7!}{(7-4)!}t^{7-4}] \cdot p \, dt$$

$$A_{eq} p = b_{eq}$$



NUS National University of Singapore

Time intervals:

- 0 0.5401 1.3039 3.5308 5.0000

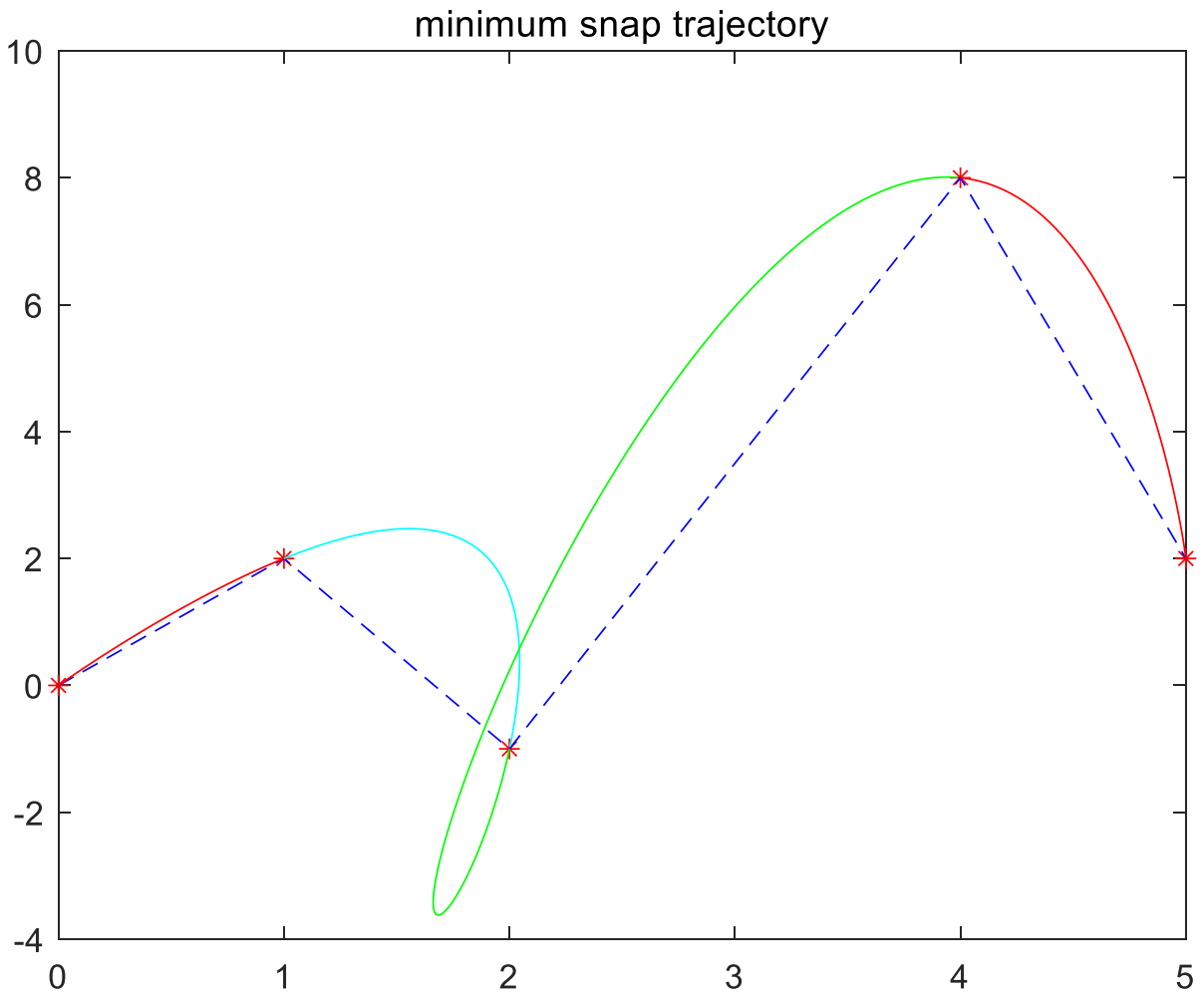
Equation constraints:
6 boundary condictions,i.e.,pva

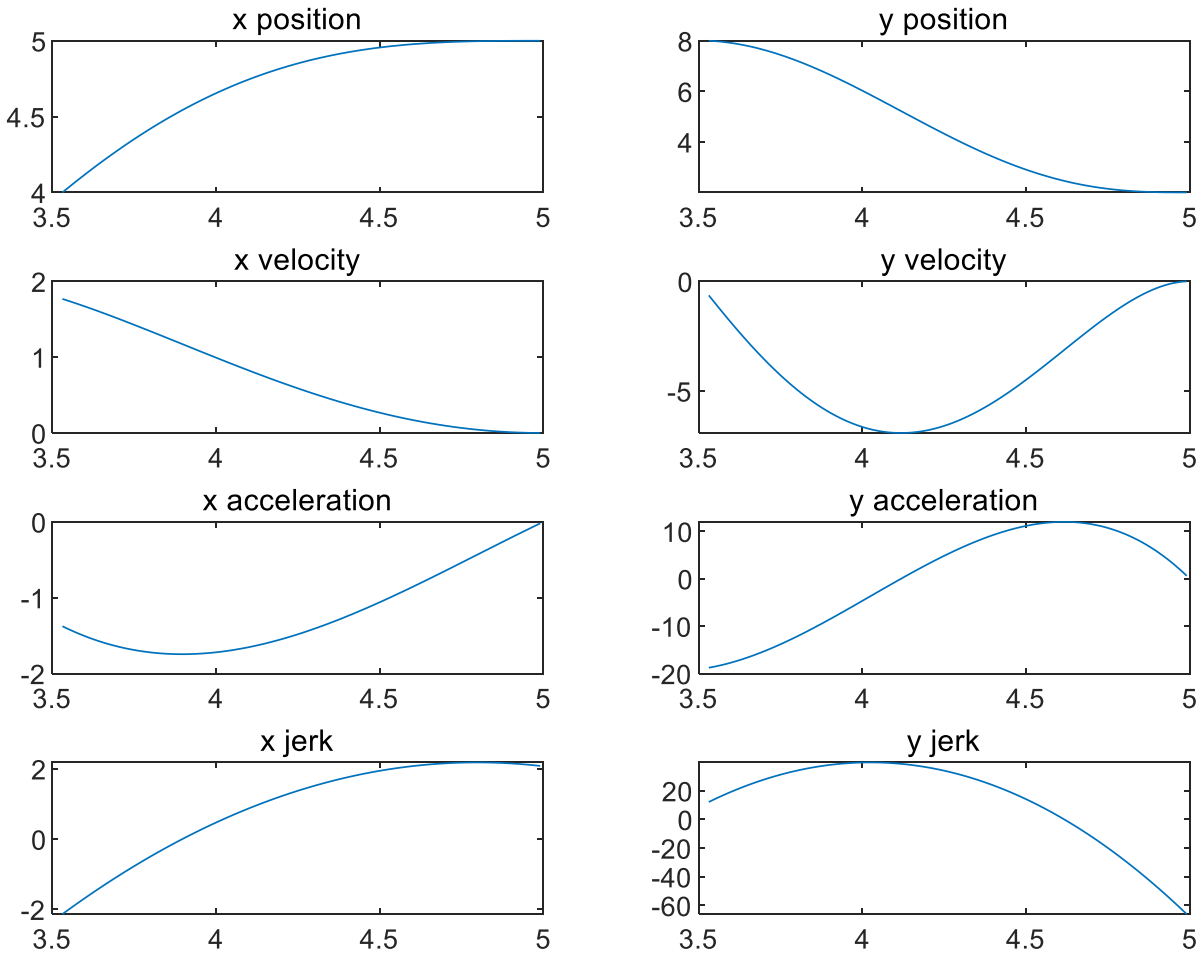
Quadratic programming.
We use **MATLAB quadprog**

$$\min_x \frac{1}{2} x^T H x + f^T x \text{ such that } \begin{cases} A \cdot x \leq b, \\ Aeq \cdot x = beq, \end{cases}$$

f=0, A=[ ], b=[ ], since we don't have inequality constraints

```
x = quadprog(H,f,A,b,Aeq,beq)
```


minimum snap trajectory
10
8
6
4
2
0
-2
-4
0
1
2
3
4
5


x position
5
4.5
4
3.5
4
4.5
5
y position
8
6
4
3.5
4
4.5
5
x velocity
2
1
0
3.5
4
4.5
5
y velocity
0
-5
3.5
4
4.5
5
x acceleration
0
-1
-2
3.5
4
4.5
5
y acceleration
10
0
-10
-20
3.5
4
4.5
5
x jerk
2
0
-2
3.5
4
4.5
5
y jerk
20
0
-20
-40
-60
3.5
4
4.5
5

Comparison between polynomial method 3.2 and 3.3

- Method 3.2 has only equation constraints
- Method 3.3 includes 3.2's equation constraints and it adds cost function and inequality constraints

How do we select constraints ?
In general,

- Max Velocity (wrt car capabilities and speed limit)
- Min Velocity
- Max Acceleration
- Min Acceleration
- Steering Angle

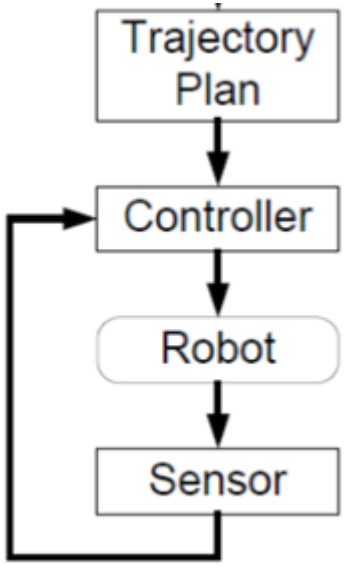
- Cannot exceed max speed of vehicle
- Minimum velocity → usually shouldn't be negative (no backing up)
- Lateral acceleration needs to be checked to avoid rollover / skid
- Longitudinal acceleration needs to be checked with powertrain capabilities
- Corresponds to max braking force
- Cannot exceed max / min steering

NUS National University of Singapore

How do we use trajectories?

- We have trajectories

$$p(t)_d,\ \dot{p}_d(t),\ \ddot{p}_d(t)$$



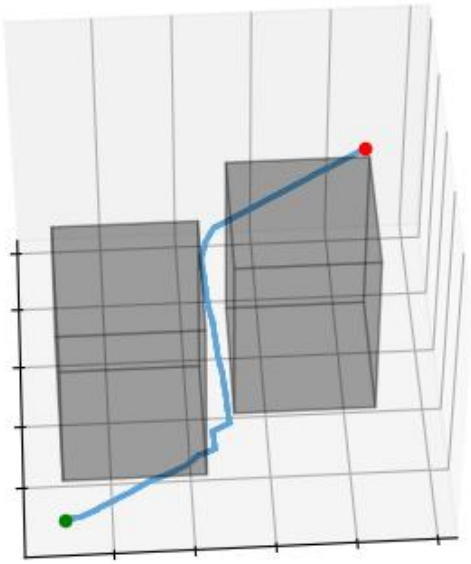
- Control=feedforward+feedback

$$\ddot{p}_d(t) + k_p(p_d - p_{\text{meas}}) + k_d(\dot{p}_d - v_{\text{meas}})$$

NUS National University of Singapore

One example

Lab environment
2m*2m*1.5m

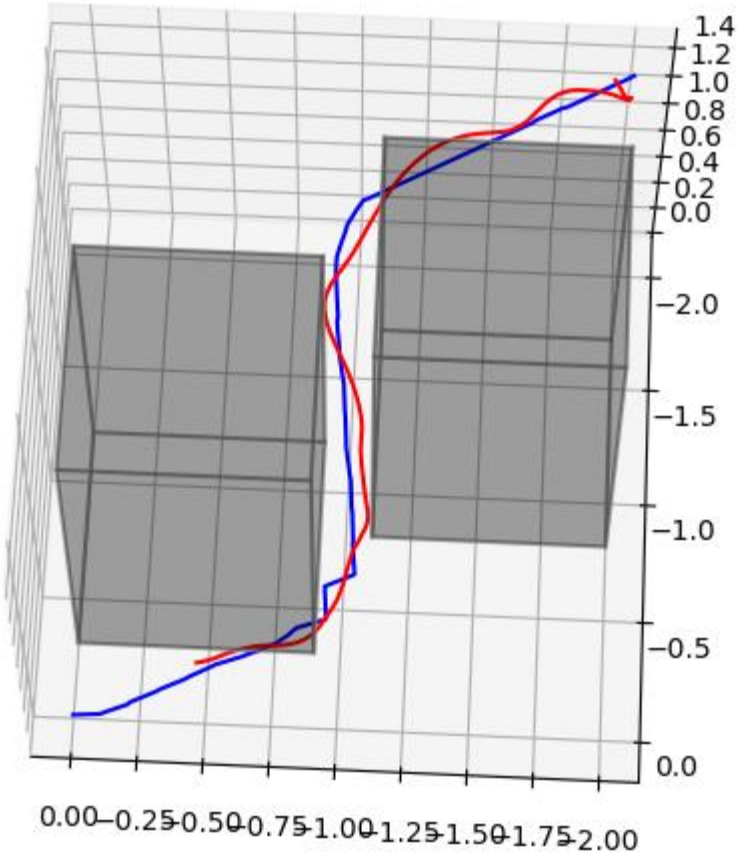


Map size: 256*256*256
Level:3

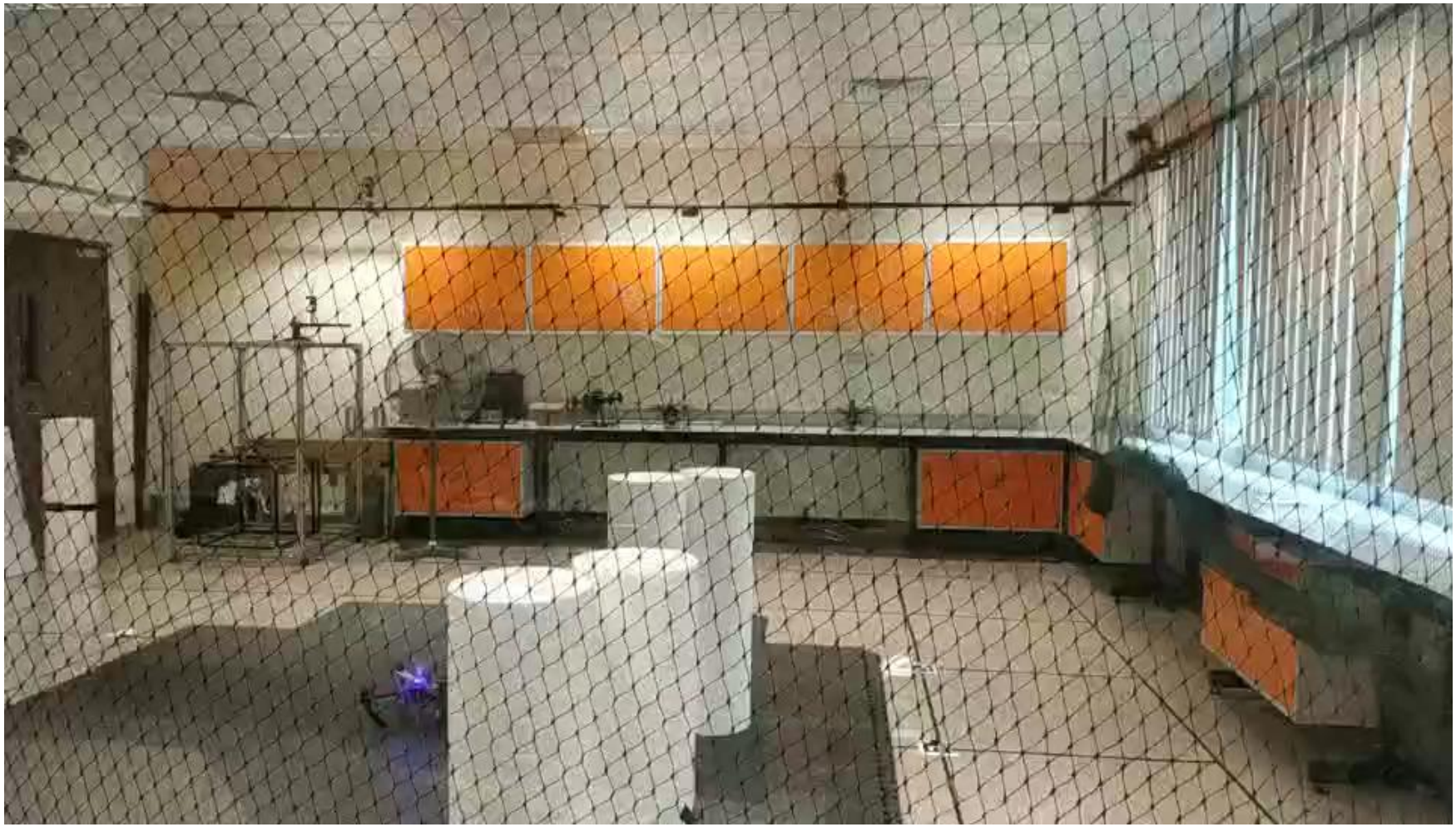
NUS National University of Singapore

Desired and actual trajectories

Blue:desired Red:actual



NUS National University of Singapore



NUS National University of Singapore

Summary

- We use polynomial for generating trajectories, position, velocity and acceleration for two points.
- For multiple points, we can use multiple polynomials to generate trajectories.
- Several methods have been discussed. It is hard to say which method is better. This depends on your applications.

Limitations:

- Computational load is high, especially in multiple waypoints.

Motion planning

NUS National University of Singapore

Element	Method	Key point	Remark
C -space	Grid map	Vehicle → point	
		Obstacles with adding vehicle's radius	
Path planning	A*	F=G+H, H --heuristic function	
	Sampling-based search (PRM)	Generate random sampling points	
		Remove points or links colliding with obstacles	
		A* search	
Trajectory generation	Polynomial trajectory without optimization	Define n order → n+1 parameters	
		Define boundary conditions such that we have n+1 equations	
	Polynomial trajectory optimization	Define n order →n+1 parameters	
		Define boundary and continuity conditions → equations	
		Define constraint conditions → inequality constraints	
		QP	

NUS National University of Singapore

References

- Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki, and S. Thrun. Principles of Robot Motion: Theory, Algorithms, and Implementations. MIT Press, Boston, MA, 2005.
- Jean-Claude Latombe, Robot Motion Planning
- John J. Craig, Introduction to Robotics

NUS National University of Singapore

Thank you!