第7讲 机器人轨迹规划

谢 斌
xiebin@csu.edu.cn
中南大学人工智能与机器人实验室
http://airl.csu.edu.cn

Review

- 机器人视觉概述
- 数字图像获取与处理
- 相机标定
- 手眼标定
- 目标特征提取与跟踪

焊接是制造业的重要工序

2017年约有**300万**焊工；

45岁以上占**70%**；

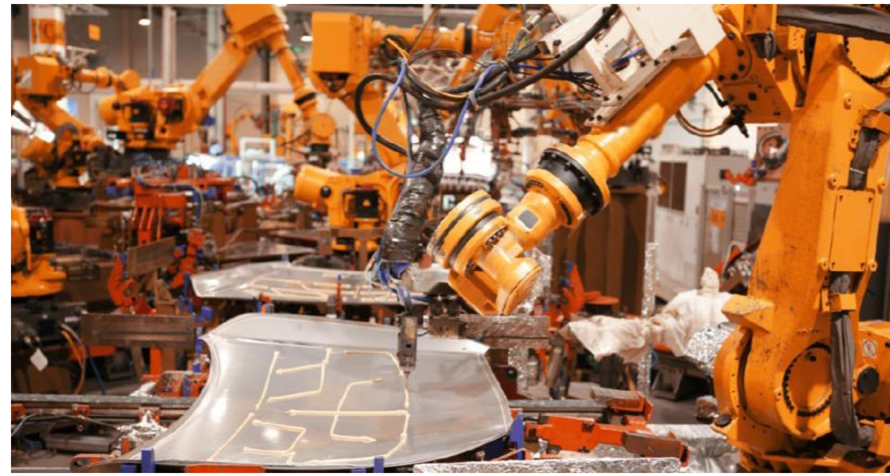
人社部公布的

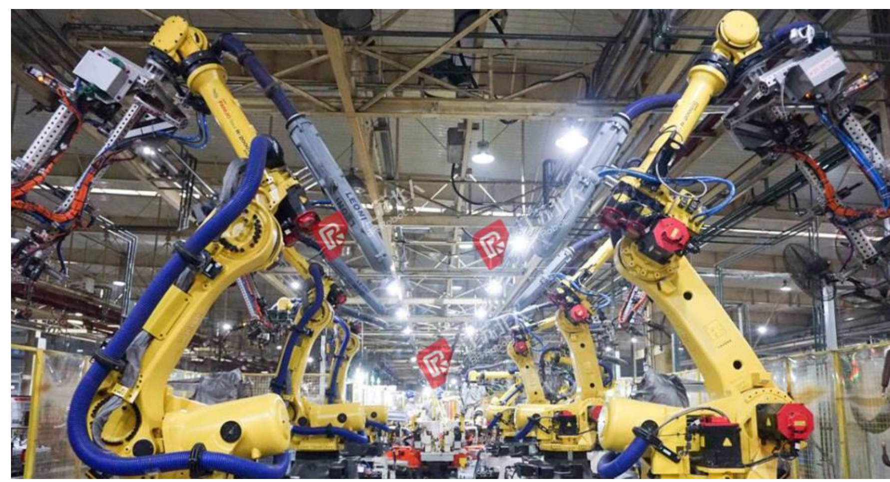
2021年第三季度全国招聘

最缺工的100个职业排行

焊工缺口排名前十

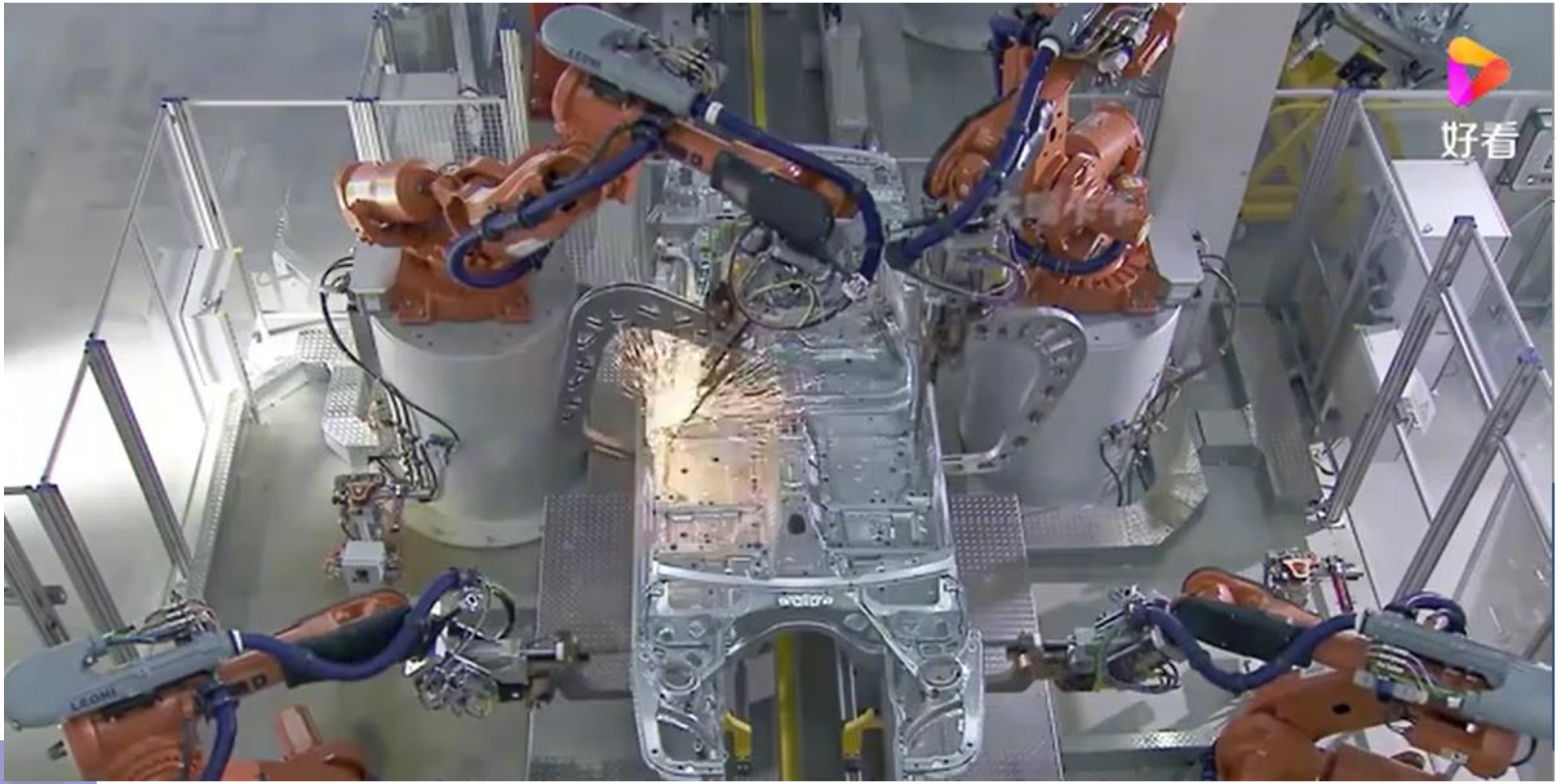
焊接是制造业的重要工序





全自动焊接机器人

焊接机器人如何在焊点间移动？



学习目标

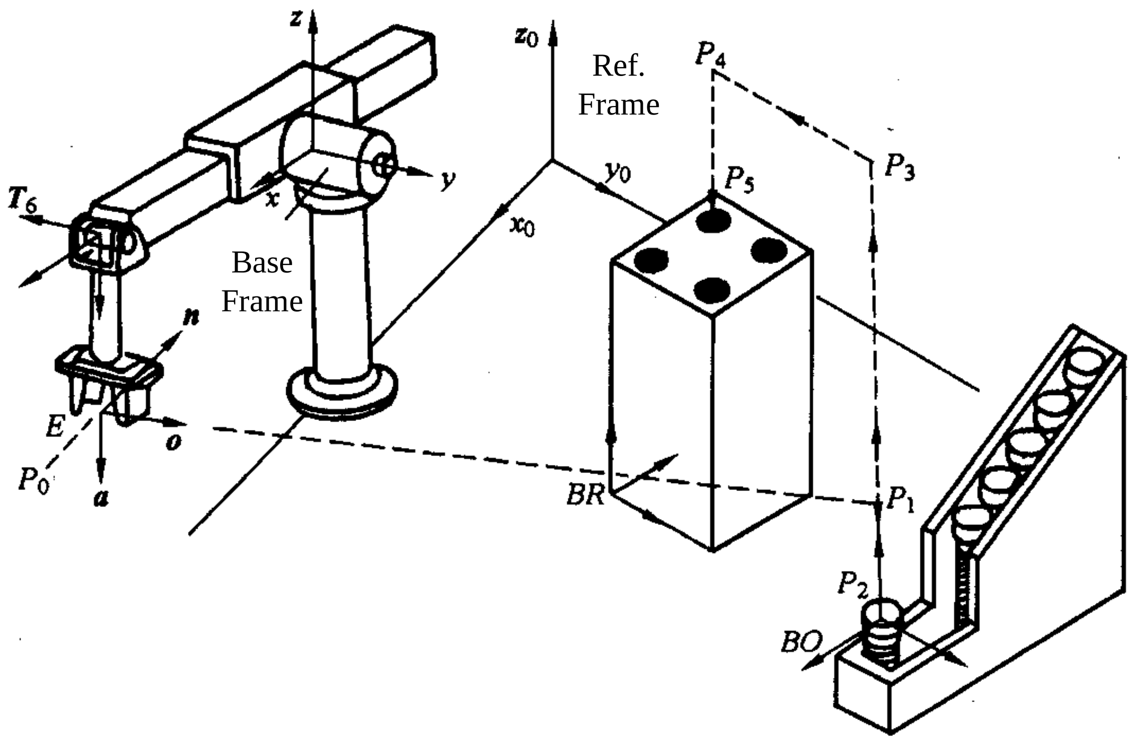
1、能够阐述轨迹规划的概念和目的；

2、能够掌握关节空间轨迹规划的三次多项式插值方法；

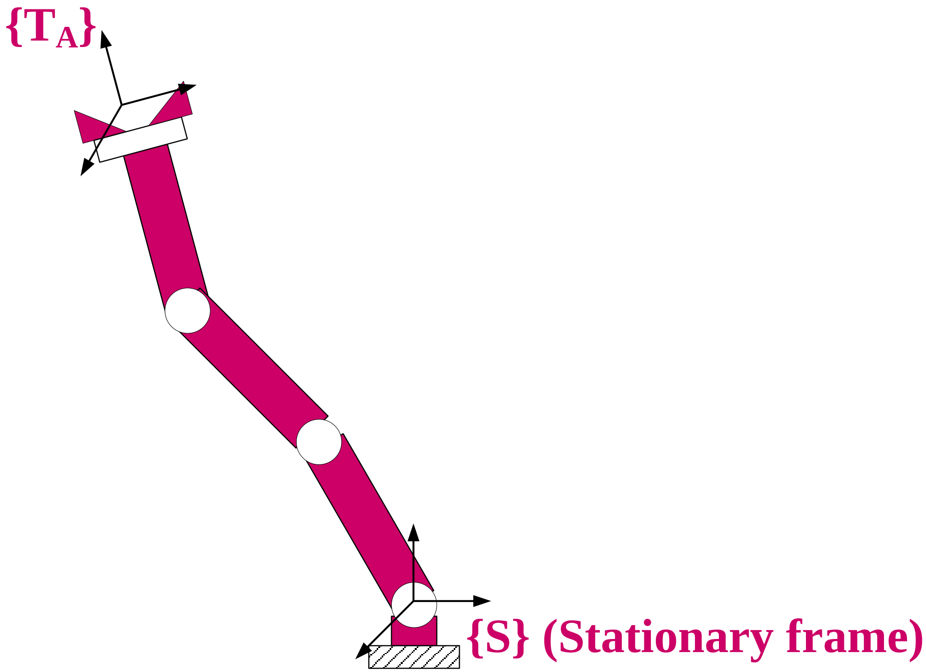
3、能够辨别两类轨迹规划方法，并比较其优缺点。

- **Basic Problem**:

  Move the manipulator arm from some initial position to some desired final position (May be going through some via points).

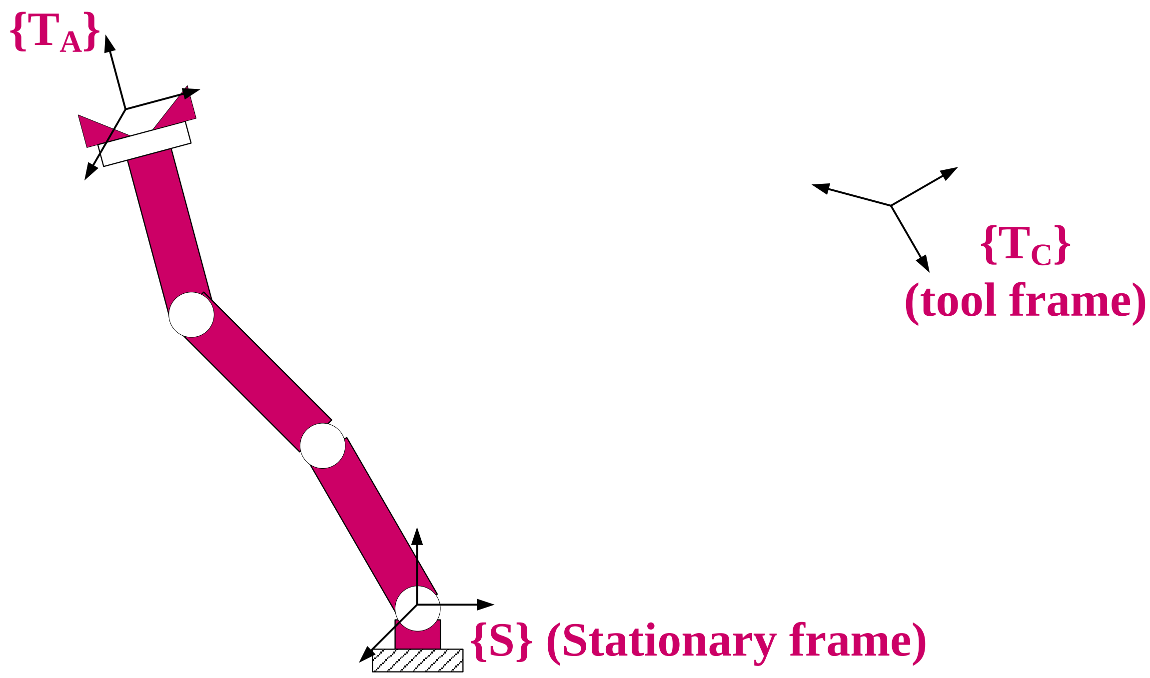


- **Trajectory** :



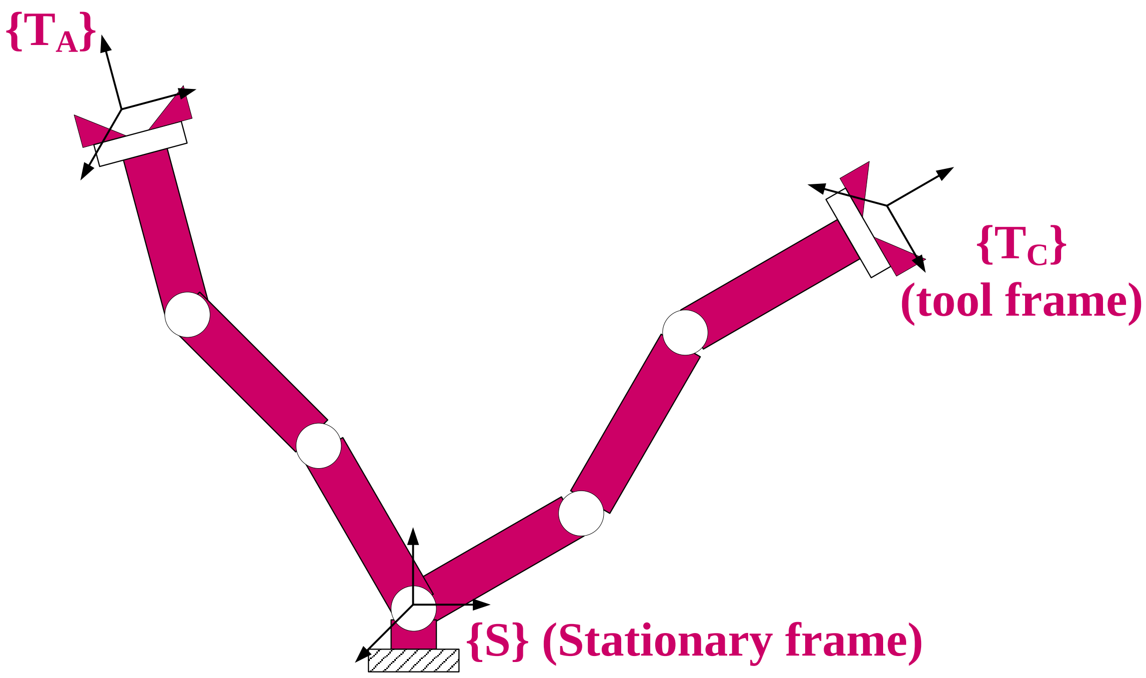
7.1 General Considerations

- **Trajectory** :



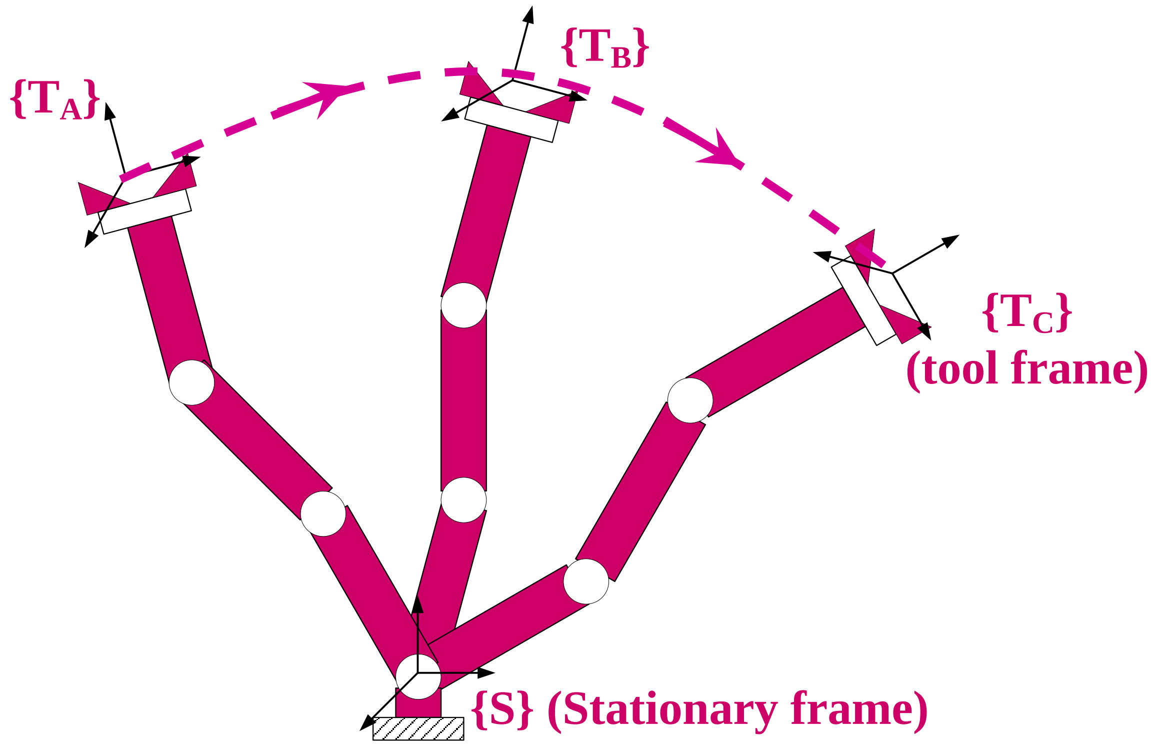
7.1 General Considerations

- **Trajectory** :

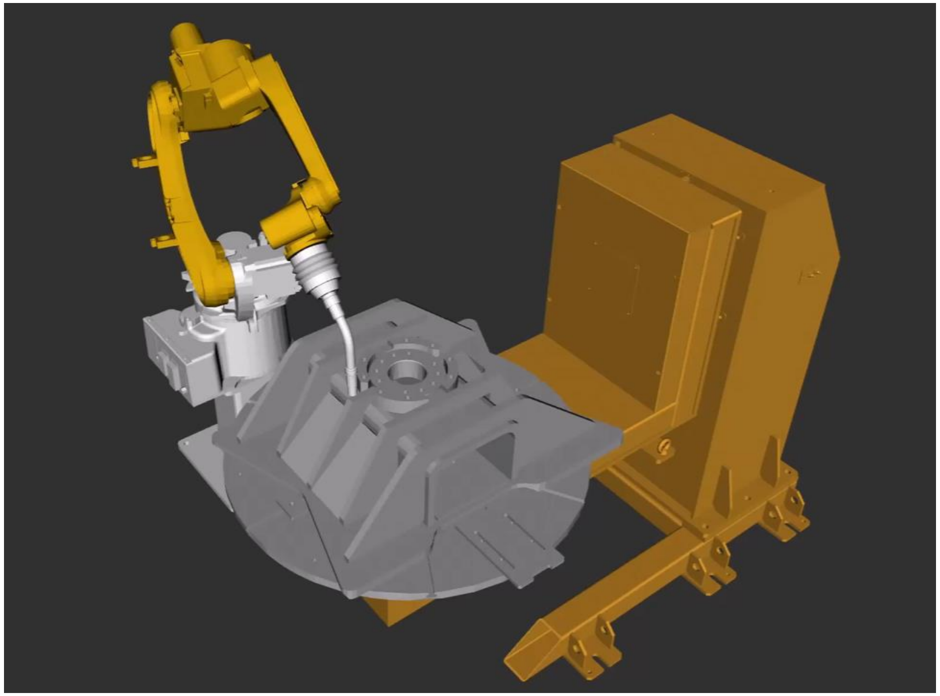


7.1 General Considerations

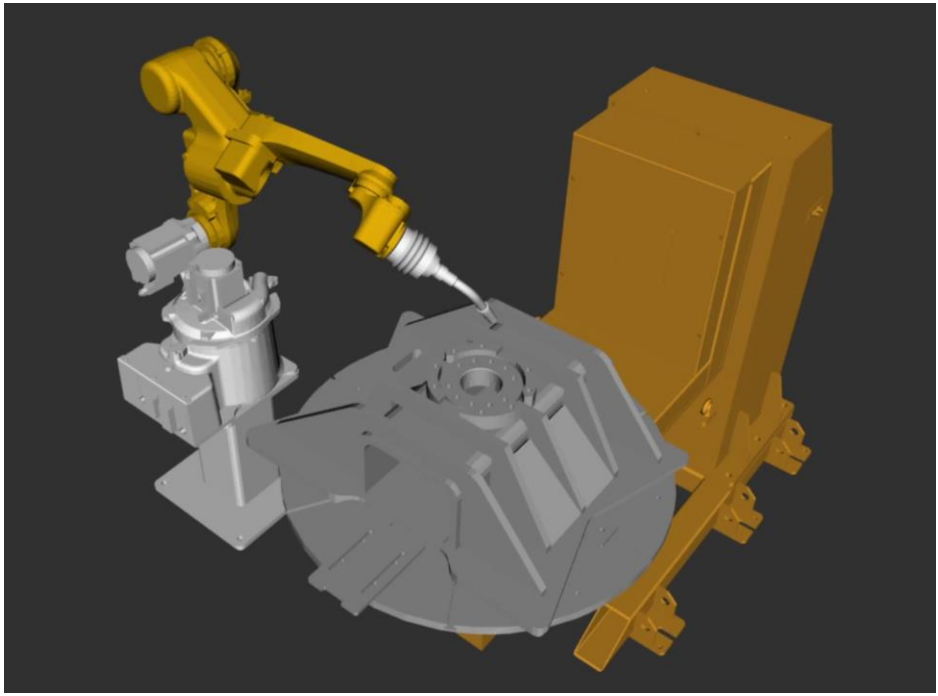
- **Trajectory** : Time history of position, velocity and acceleration for each DOF
- **Path points** : Initial, final and via points
- **Constraints**: Spatial, temporal, smooth



轨迹规划的两种类型



点焊



连续焊

关节空间轨迹规划

笛卡尔空间轨迹规划

共同要求：必须满足约束、连续且平滑，使得机械臂运动平稳

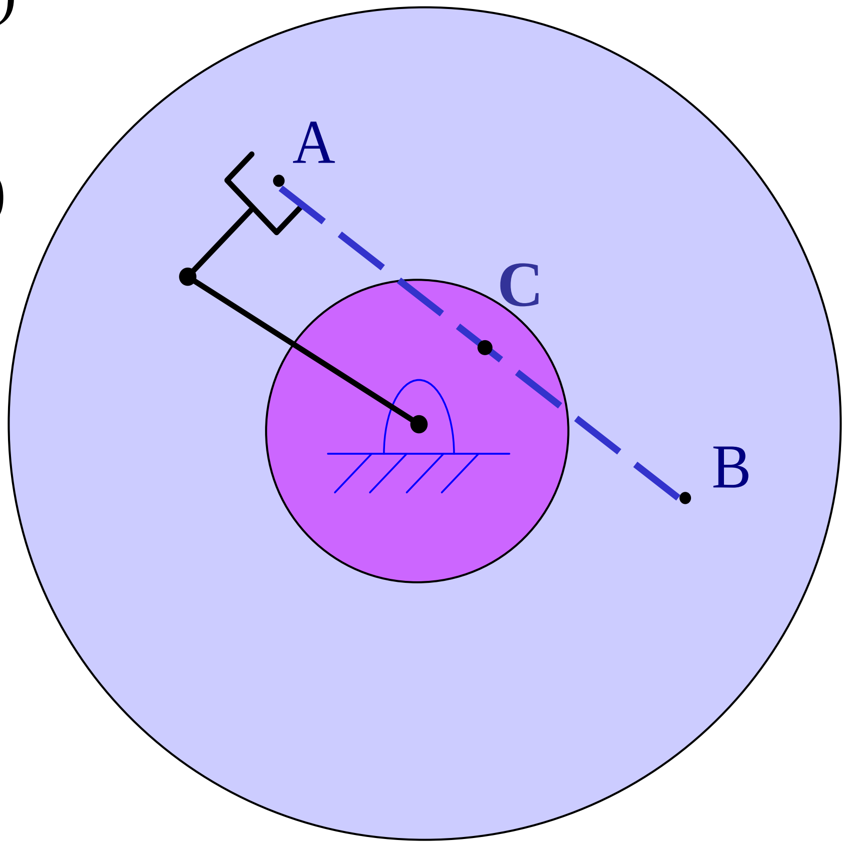
General Considerations - Solution Space

Solution Space	Calculations	Singularity Problems	Track a shape
Joint space	Less		
Cartesian space	More		

General Considerations - Solution Space

- Cartesian planning difficulties:
  - Initial (A) and Goal (B)

    Points are reachable.
  - Intermediate points (C) is unreachable.



General Considerations - Solution Space

Solution Space	Calculations	Singularity Problems	Track a shape
Joint space	Less	No	Hard
Cartesian space	More	Yes	Easy

Question: Which method is more often used?

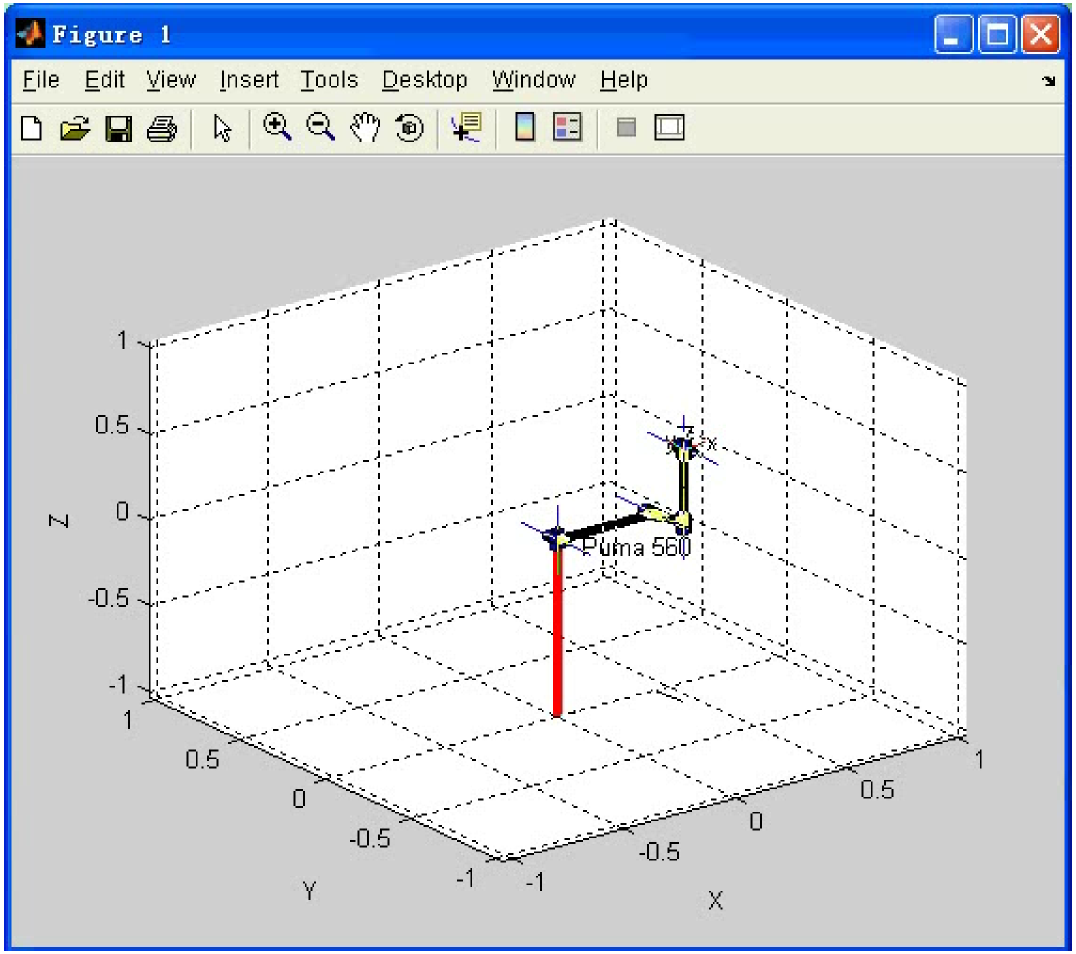
Trajectory planning in joint space is more often used!

Ch.7 Trajectory Planning of Robots

7.2 Interpolation of Joint Trajectories

Move from

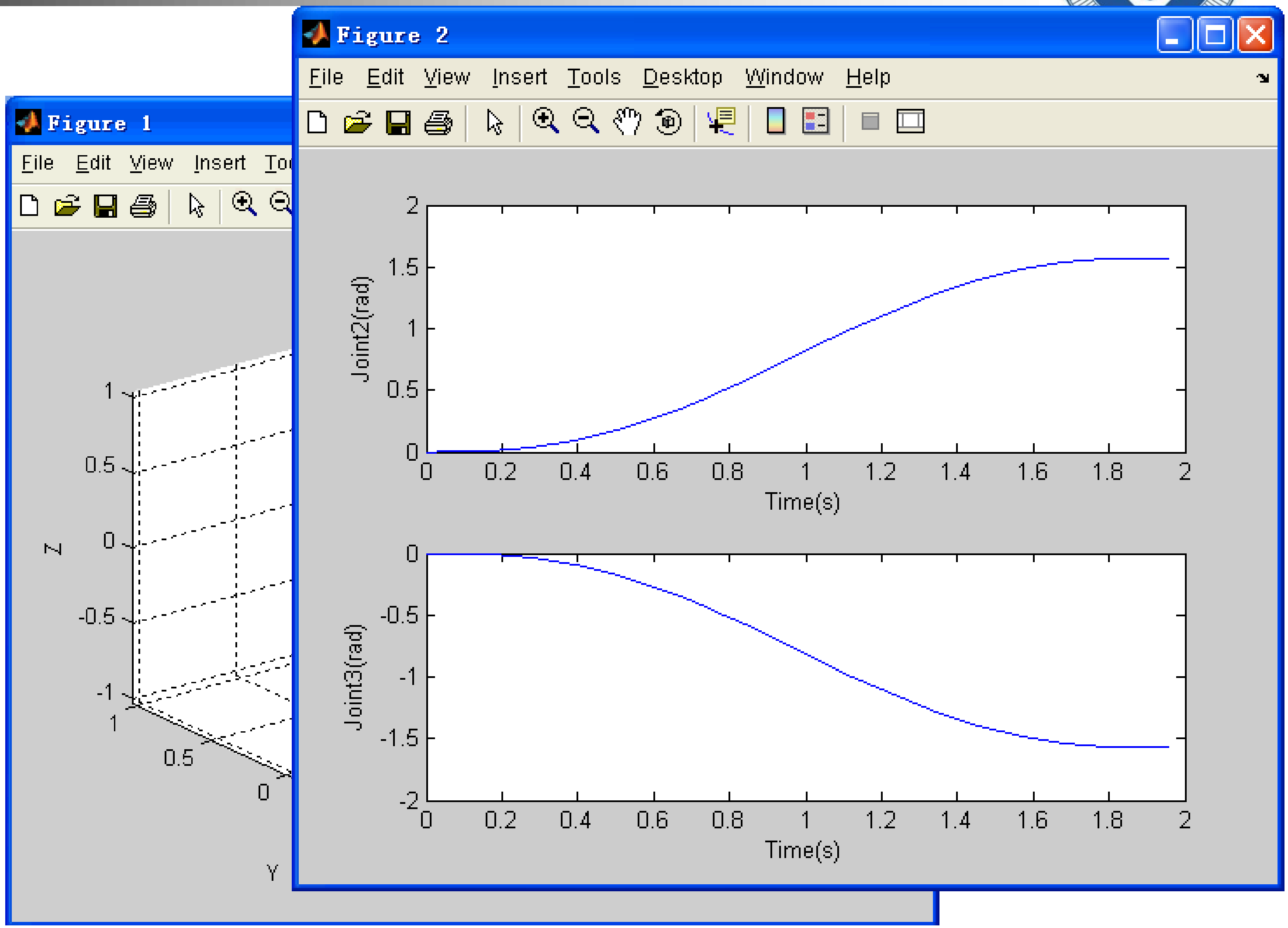
qz to *qr*



7.2 Interpolation of Joint Trajectories

Move from

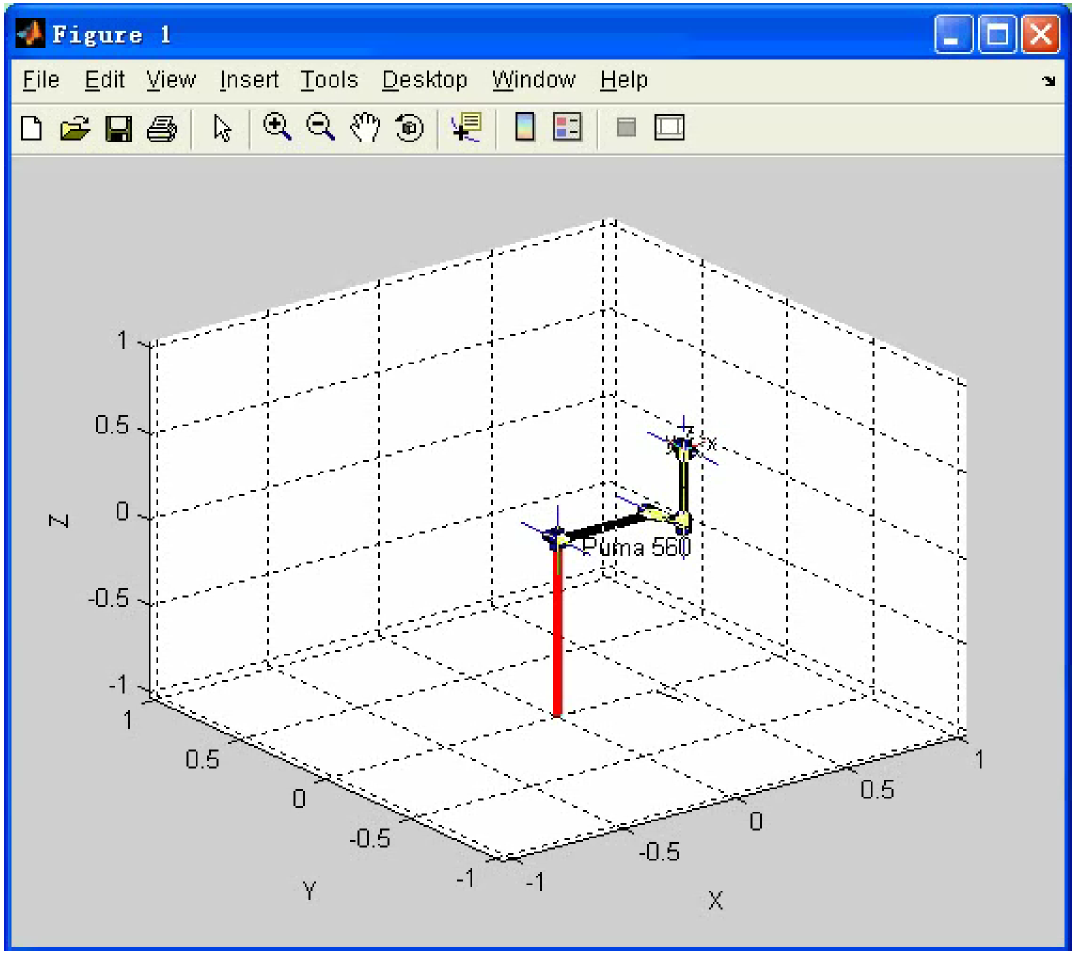
qz to *qr*



7.2 Interpolation of Joint Trajectories

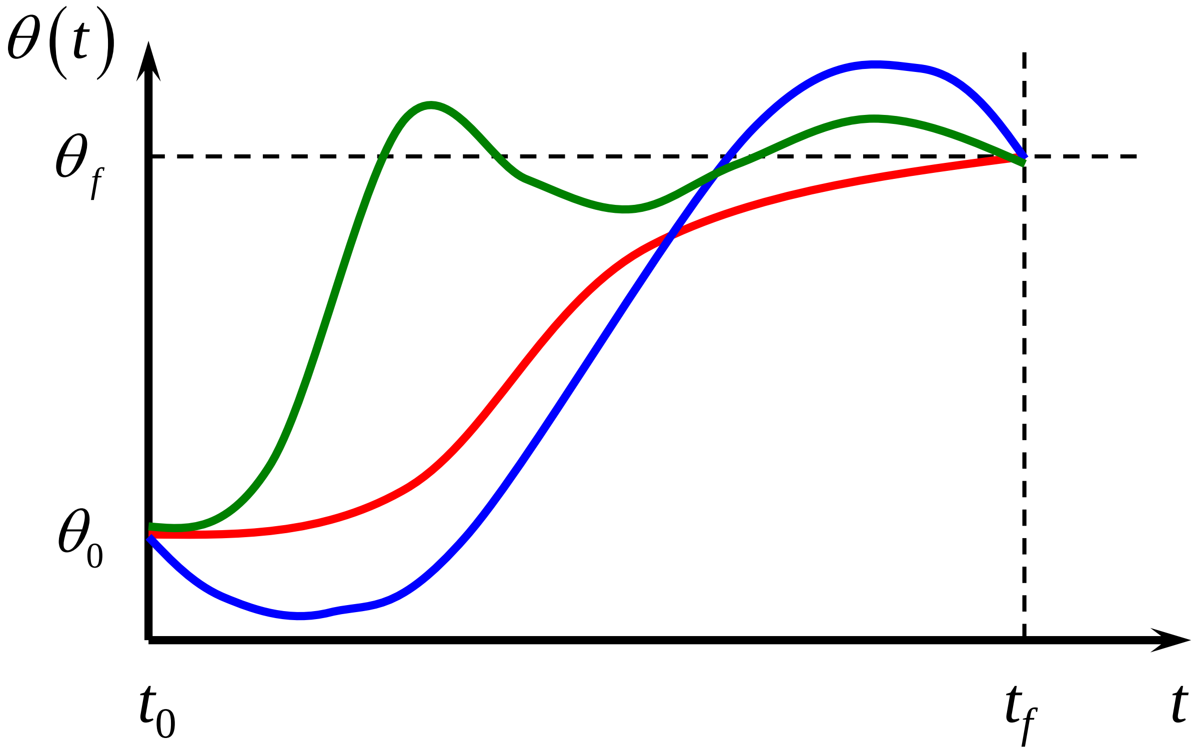
Move from

qz to *qr*



Joint-Space Schemes

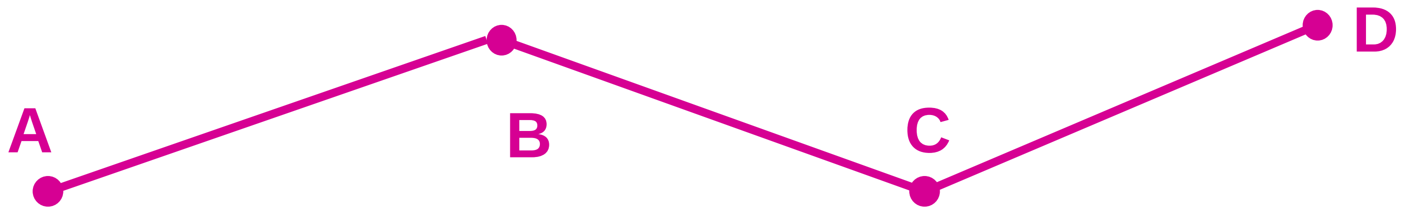
Several possible path shapes for a single joint



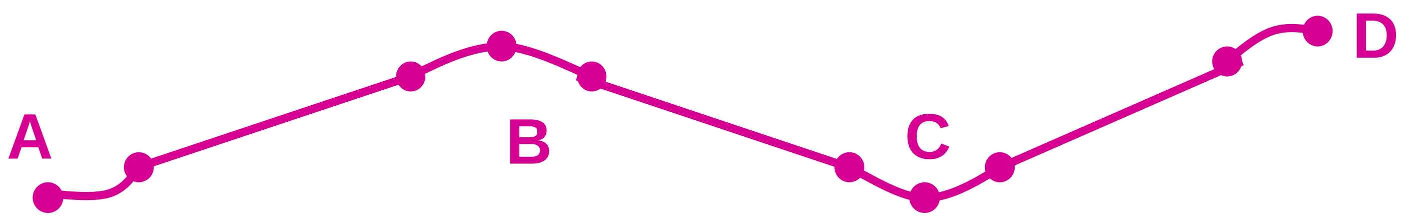
Note: Choice of interpolation function is **not unique**!

Candidate curves

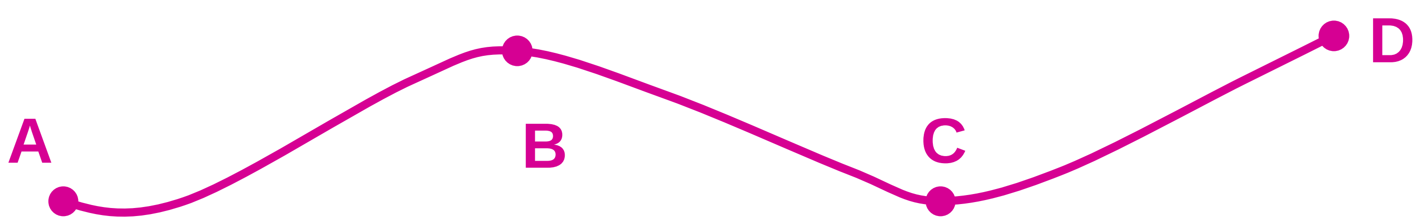
- straight line (discontinuous velocity at path points)



- straight line with blends



- cubic polynomials (splines)

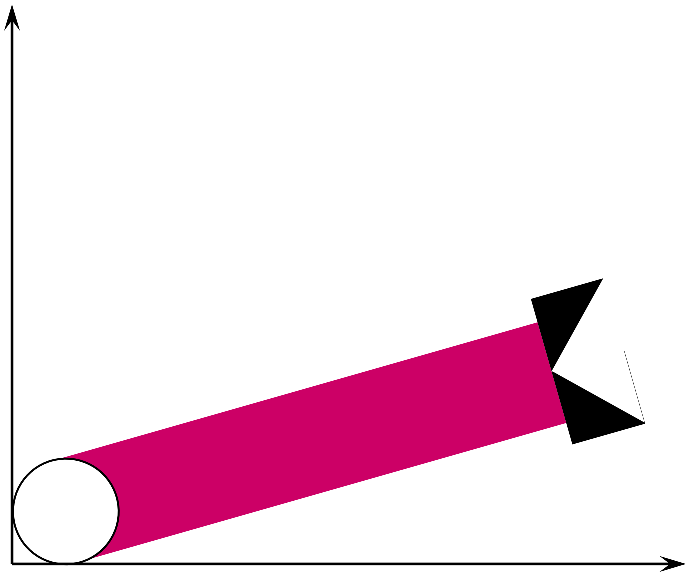


- higher order polynomials (quintic,…) or other curves

7.2.1 Cubic Polynomials

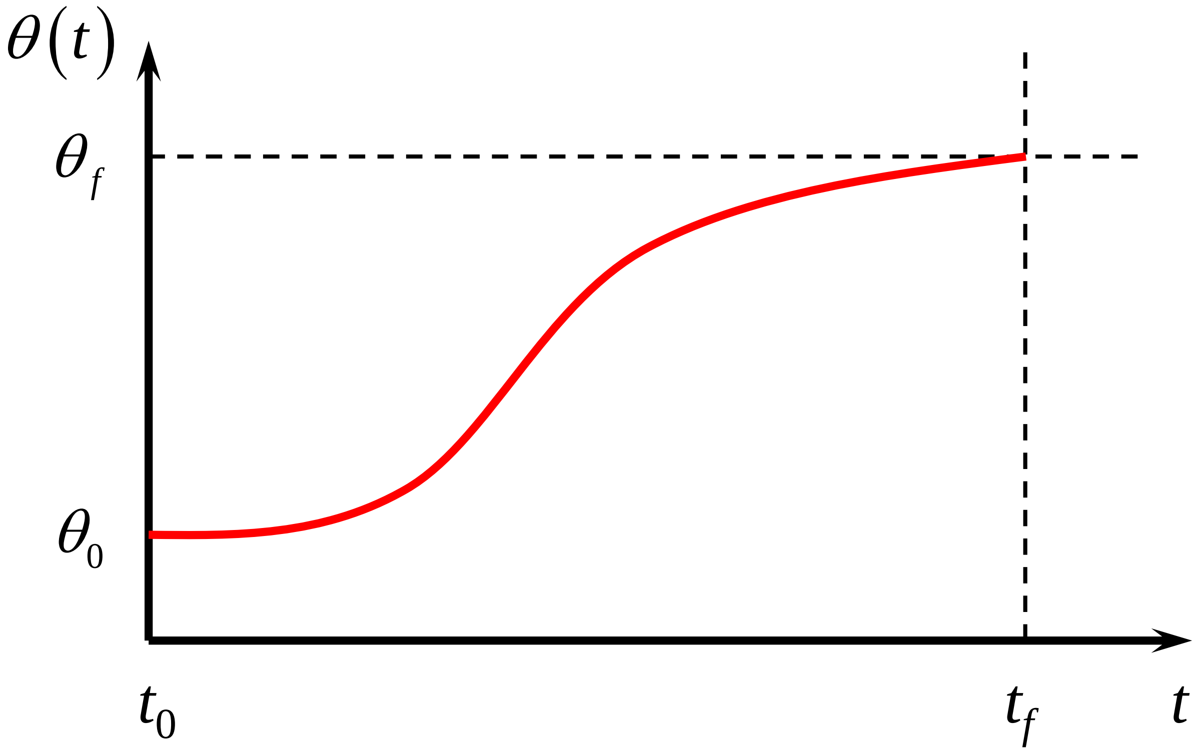
- **Question.** How to move a rotary single-link robot from $\theta = 15$ degrees to $\theta = 75$ degrees in 3 seconds in a cubic polynomials manner? Bring the manipulator starts and ends at rest.

Revolute joint



7.2.1 Cubic Polynomials

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

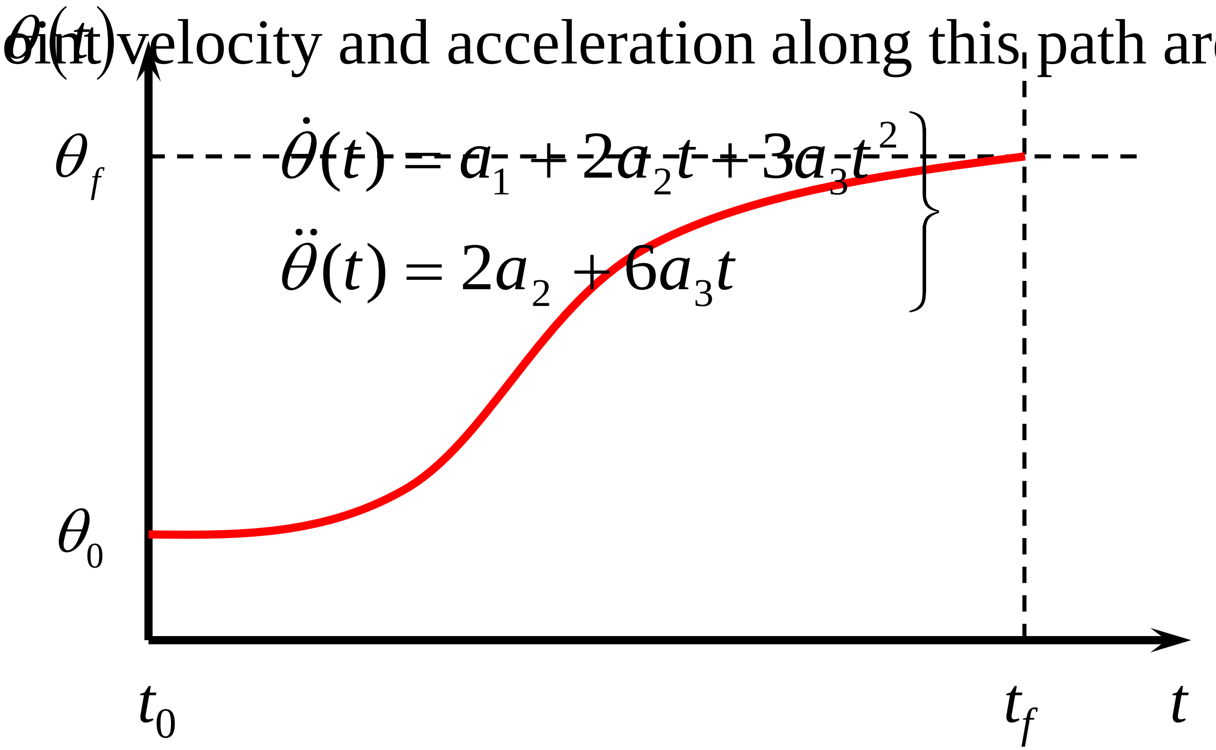


7.2.1 Cubic Polynomials

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

- The joint velocity and acceleration along this path are:

$$\left.\begin{aligned}\dot{\theta}(t) &= a_1 + 2a_2 t + 3a_3 t^2 \\ \ddot{\theta}(t) &= 2a_2 + 6a_3 t\end{aligned}\right\}$$



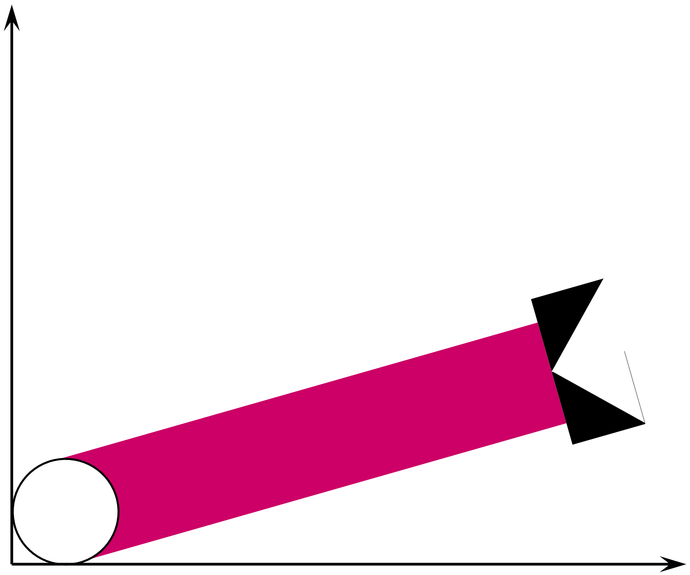
7.2.1 Cubic Polynomials

- **Question.** How to move a rotary single-link robot from $\theta = 15$ degrees to $\theta = 75$ degrees in 3 seconds in a cubic polynomials manner? Bring the manipulator starts and ends at rest.

Initial Conditions:

$$\left.\begin{aligned}\theta(t_0) &= \theta_0 \\ \theta(t_f) &= \theta_f\end{aligned}\right\}$$

$$\left.\begin{aligned}\dot{\theta}(t_0) &= 0 \\ \dot{\theta}(t_f) &= 0\end{aligned}\right\}$$



7.2.1 Cubic Polynomials

- **Question.** How to move a rotary single-link robot from $\theta = 15$ degrees to $\theta = 75$ degrees in 3 seconds in a cubic polynomials manner? Bring the manipulator starts and ends at rest.

Initial Conditions:

$$\left.\begin{aligned}\theta(t_0) &= \theta_0 \\ \theta(t_f) &= \theta_f\end{aligned}\right\} \qquad \left.\begin{aligned}\dot{\theta}(t_0) &= 0 \\ \dot{\theta}(t_f) &= 0\end{aligned}\right\} \quad \Rightarrow \quad \left.\begin{aligned}\theta_0 &= a_0 \\ \theta_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \\ 0 &= a_1 \\ 0 &= a_1 + 2a_2 t_f + 3a_3 t_f^2\end{aligned}\right\}$$

7.2.1 Cubic Polynomials

- **Question.** How to move a rotary single-link robot from $\theta = 15$ degrees to $\theta = 75$ degrees in 3 seconds in a cubic polynomials manner? Bring the manipulator starts and ends at rest.

Initial Conditions:

$$\left.\begin{aligned}\theta(t_0) &= \theta_0 \\ \theta(t_f) &= \theta_f\end{aligned}\right\}$$

$$\left.\begin{aligned}\dot{\theta}(t_0) &= 0 \\ \dot{\theta}(t_f) &= 0\end{aligned}\right\}$$

$$\Rightarrow$$

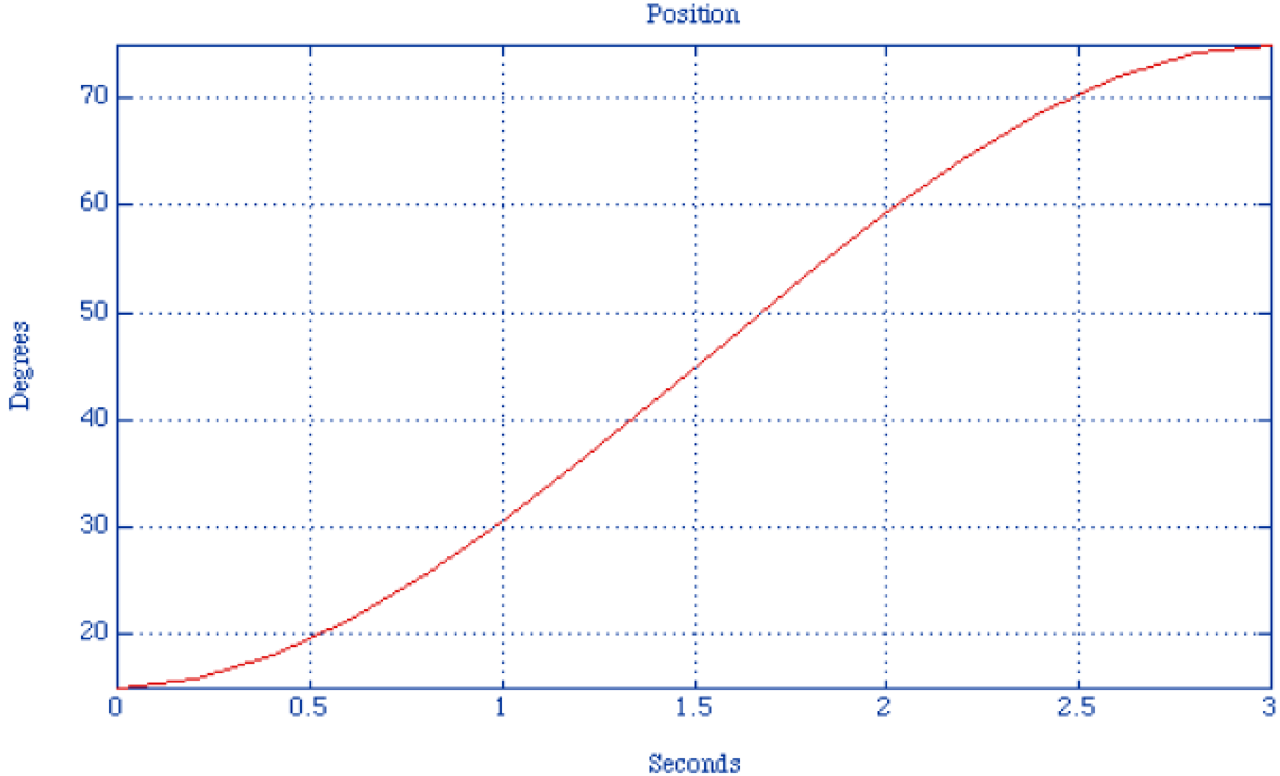
Solution:

$$\left.\begin{aligned}a_0 &= \theta_0 \\ a_1 &= 0 \\ a_2 &= \frac{3}{t_f^2}(\theta_f - \theta_0) \\ a_3 &= -\frac{2}{t_f^3}(\theta_f - \theta_0)\end{aligned}\right\}$$

7.2.1 Cubic Polynomials

- Solution: Plugging $\theta_0 = 15,\ \theta_f = 75,\ t_f = 3$ in, we find

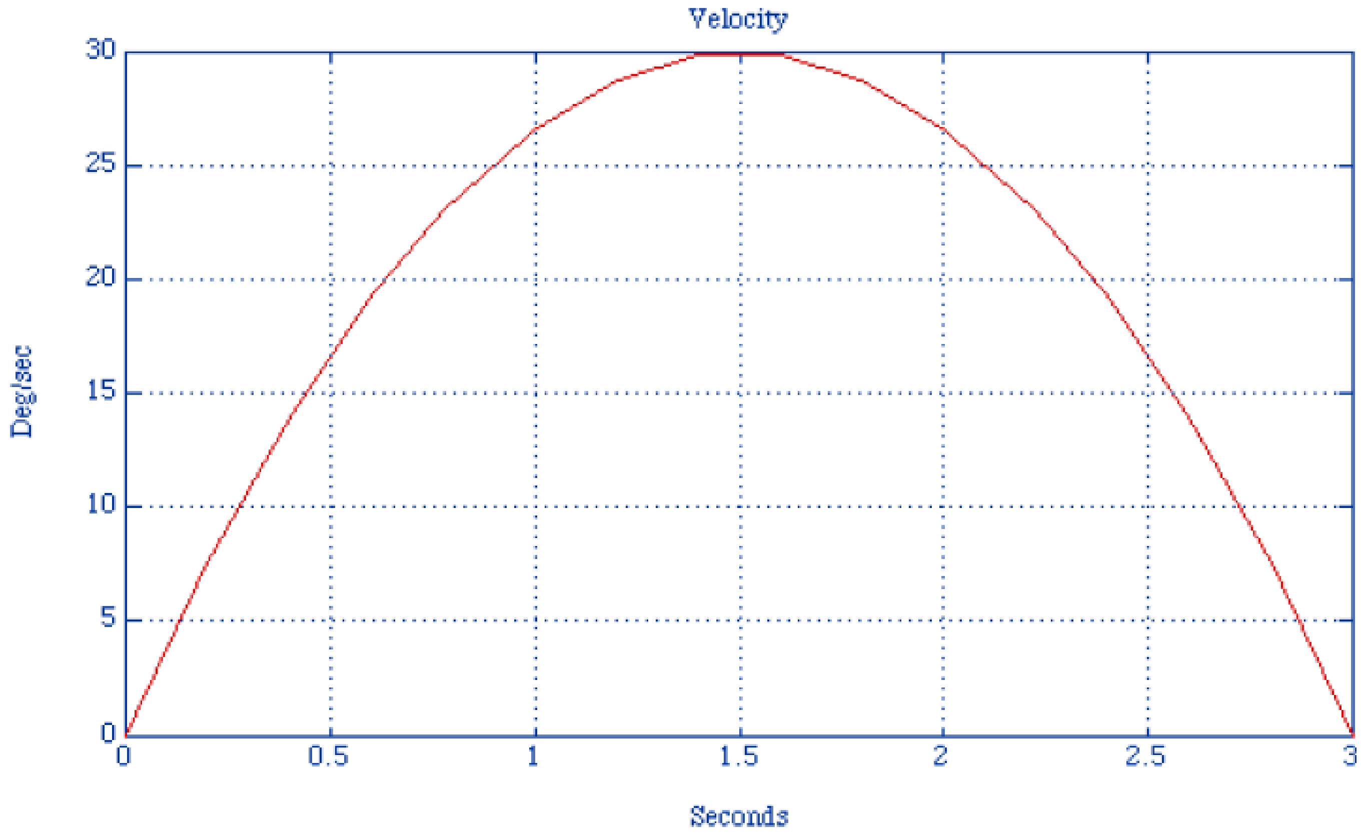
$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 = 15 + 20t^2 - 4.44t^3$$



Starts at 15 degrees and ends at 75 degrees!

7.2.1 Cubic Polynomials

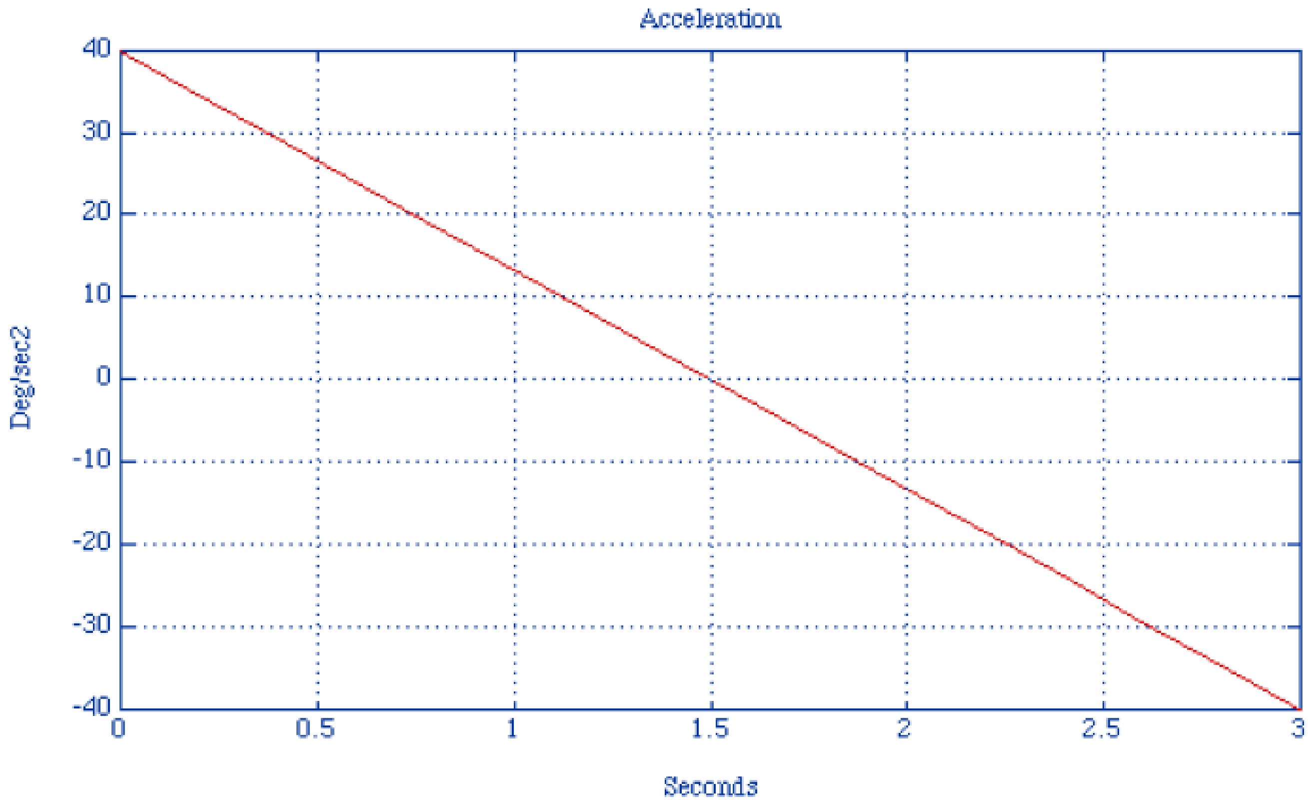
- Solution: $\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 = 40t - 13.33t^2$



Starts and ends at rest!

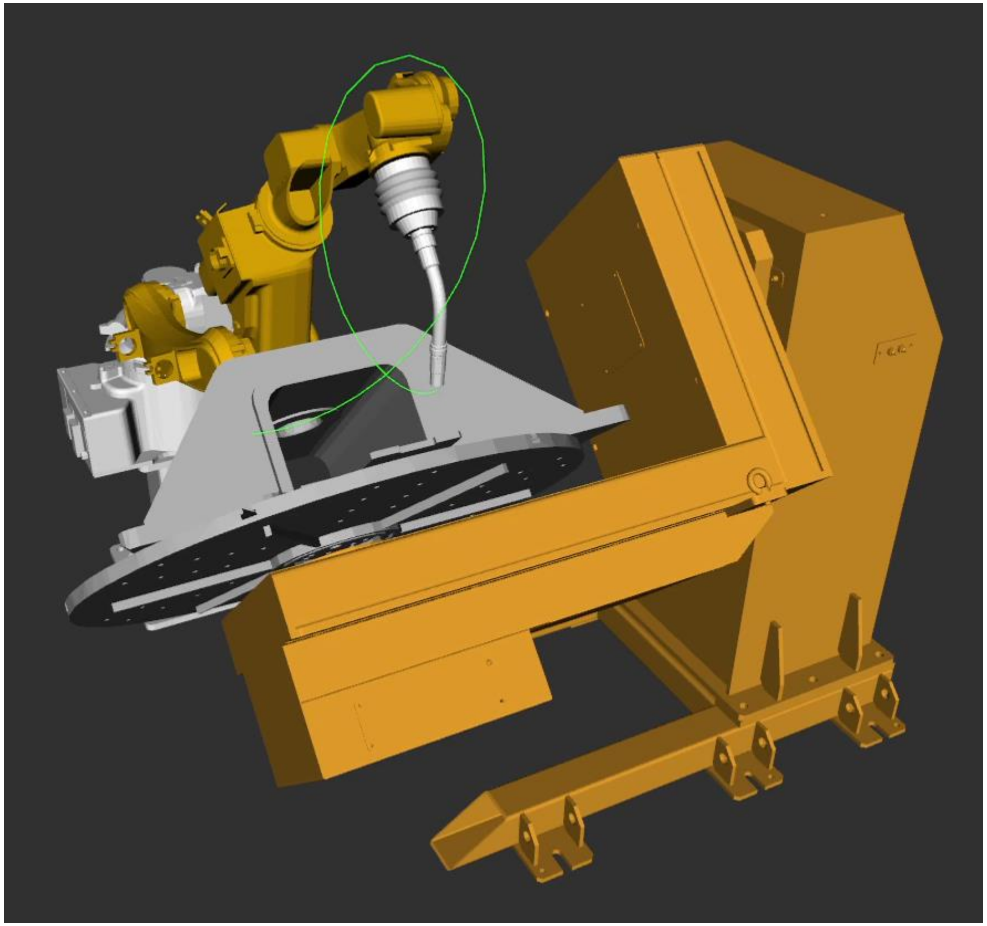
7.2.1 Cubic Polynomials

- Solution: $\ddot{\theta}(t) = 2a_2 + 6a_3 t = 40 - 26.66t$



Acceleration profile is linear!

关节空间轨迹规划的应用



7.2.2 Cubic polynomials with via points
过路径点的三次多项式插值

- If we come to rest at each point

  use formula from previous slide

- or continuous motion (no stops)

  need velocities at intermediate points:

Initial Conditions:

$$\left.\begin{aligned}\dot\theta(0)&=\dot\theta_0\\ \dot\theta(t_f)&=\dot\theta_f\end{aligned}\right\}$$

$$\left.\begin{aligned}\theta_0&=a_0\\ \theta_f&=a_0+a_1t_f+a_2t_f^2+a_3t_f^3\\ \dot\theta_0&=a_1\\ \dot\theta_f&=a_1+2a_2t_f+3a_3t_f^2\end{aligned}\right\}$$

Solutions:

$$\left.\begin{aligned}a_0&=\theta_0\\ a_1&=\dot\theta_0\\ a_2&=\frac{3}{t_f^2}(\theta_f-\theta_0)-\frac{2}{t_f}\dot\theta_0-\frac{1}{t_f}\dot\theta_f\\ a_3&=-\frac{2}{t_f^3}(\theta_f-\theta_0)+\frac{1}{t_f^2}(\dot\theta_0+\dot\theta_f)\end{aligned}\right\}$$

7.2.2 Cubic polynomials with via points

- How to specify velocity at the via points:
  - The user specifies the desired velocity at each via point in terms of a Cartesian linear and angular velocity of the tool frame at that instant.
  - The system automatically chooses the velocities at the via points by applying a suitable heuristic in either Cartesian space or joint space (average of 2 sides etc.).
  - The system automatically chooses the velocities at the via points in such a way as to cause the acceleration at the via points to be continuous.

7.2.3 Higher-order polynomials

高阶多项式插值

- Higher order polynomials are sometimes used for path segments. For example, if we wish to be able to specify the position, velocity, and acceleration at the beginning and end of a path segment, a **quintic polynomial** is required:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \qquad (7.10)$$

7.2.3 Higher-order polynomials

- Where the constraints are given as:

$$\left.\begin{aligned}
\theta_0 &= a_0 \\
\theta_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \\
\dot{\theta}_0 &= a_1 \\
\dot{\theta}_f &= a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \\
\ddot{\theta}_0 &= 2a_2 \\
\ddot{\theta}_f &= 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3
\end{aligned}\right\} \quad (7.11)$$

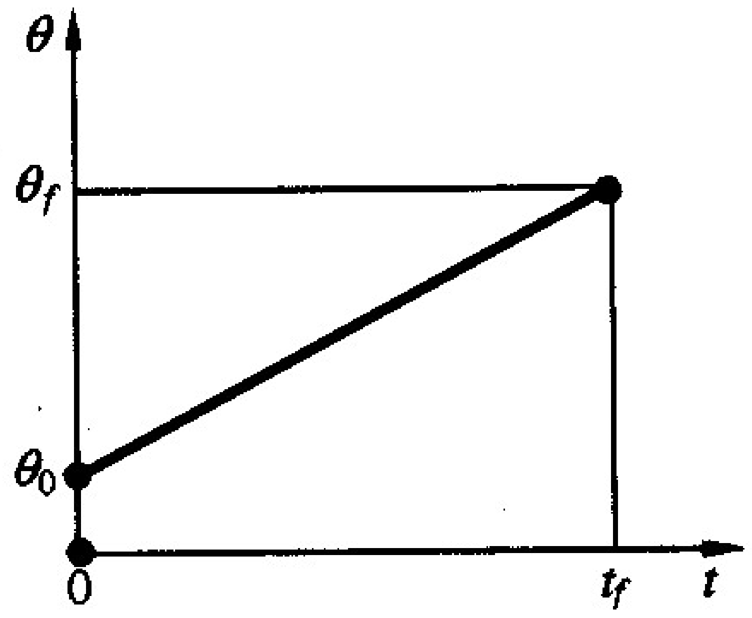
7.2.3 Higher-order polynomials

- Solution to these equations:

$$\left.\begin{aligned}
a_0 &= \theta_0 \\
a_1 &= \dot{\theta}_0 \\
a_2 &= \frac{\ddot{\theta}_0}{2} \\
a_3 &= \frac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^3} \\
a_4 &= \frac{30\theta_0 - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_0)t_f + (3\ddot{\theta}_0 - 2\ddot{\theta}_f)t_f^2}{2t_f^4} \\
a_5 &= \frac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)t_f - (\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^5}
\end{aligned}\right\} \qquad (7.12)$$

7.2.4 Linear function with parabolic blends
用抛物线过渡的线性插值

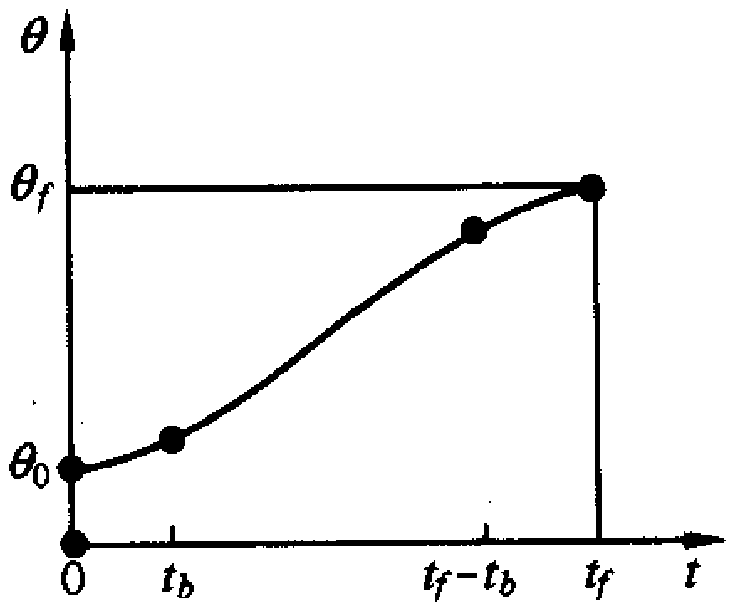
- Linear interpolation (Straight line):
- Note: Although the motion of each joint in this scheme is linear, the end-effector in general does not move in a straight line in space.



Discontinuous velocity - can not be controlled!

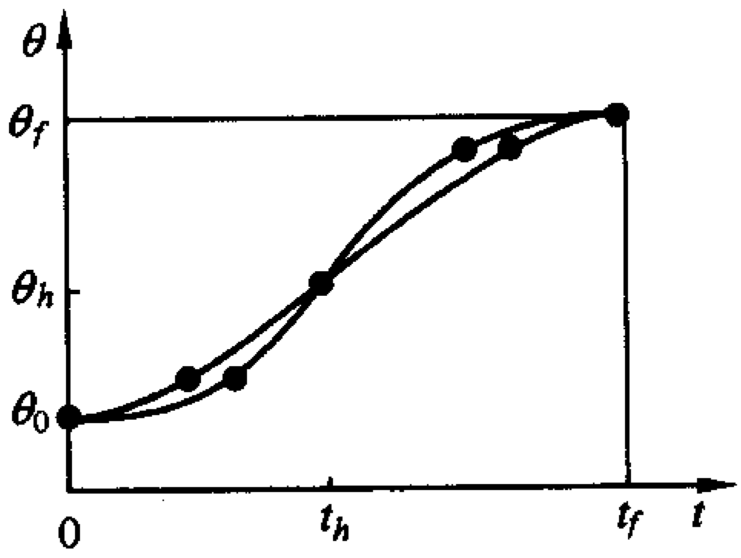
7.2.4 Linear function with parabolic blends

- To create a smooth path with continous position and velocity, we start with the linear function but add a parabolic blend region at each path point.
- Constant acceleration is used during the blend portion to change velocity smoothly.



7.2.4 Linear function with parabolic blends

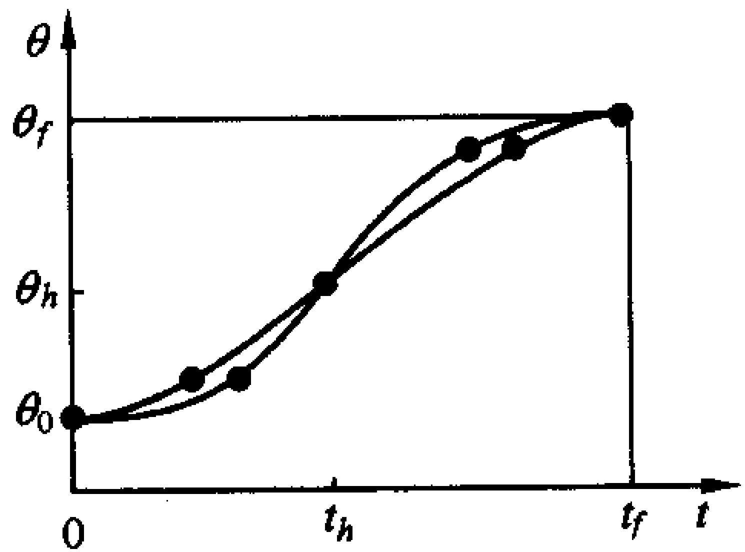
- Assume that the parabolic blends both have the same duration, and therefore the same constant acceleration (modulo a sign).
- There are many solutions to the problem-but the answer is always symmetric about the halfway point.



7.2.4 Linear function with parabolic blends

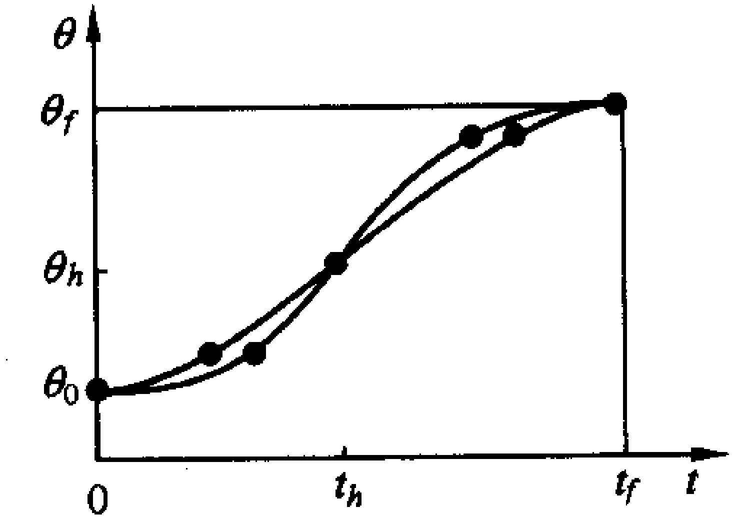
- The velocity at the end of the blend region must equal the velocity of the linear section:

$$\ddot{\theta} t_b = \dot{\theta}_{t_b} = \frac{\theta_h - \theta_b}{t_h - t_b} \qquad (7.13)$$



7.2.4 Linear function with parabolic blends

- Let $t=2t_h$,



$$\ddot{\theta} \qquad (7.15)$$

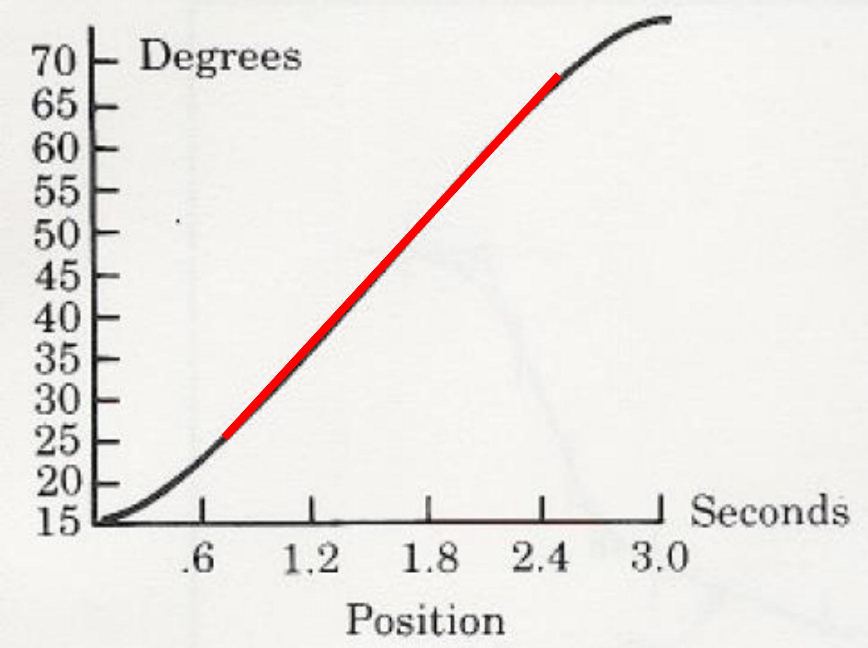
$$t_b = \qquad (7.16)$$

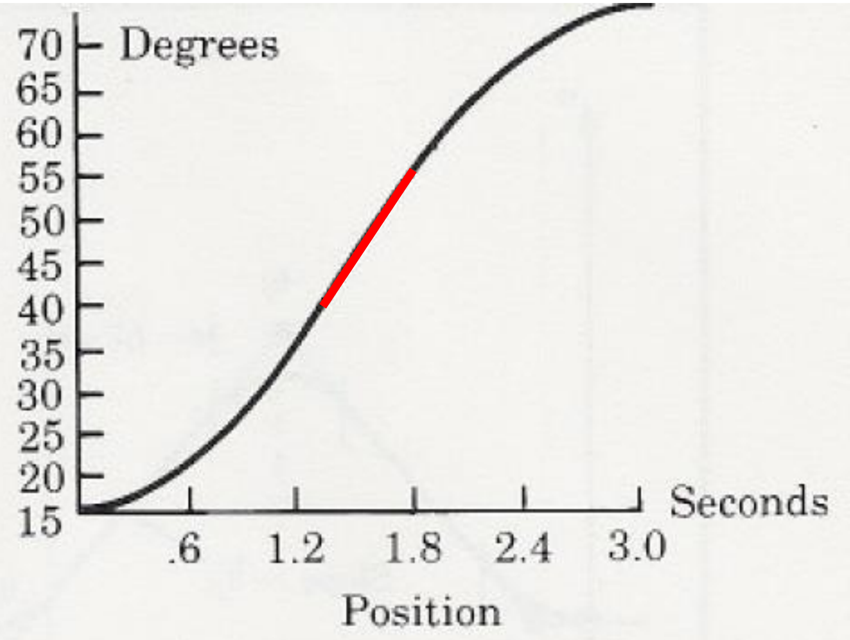
- The acceleration chosen must be sufficiently high, to ensure the existence of a solution:

$$\ddot{\theta} \geq \frac{4(\theta_f - \theta_0)}{t^2} \qquad (7.17)$$

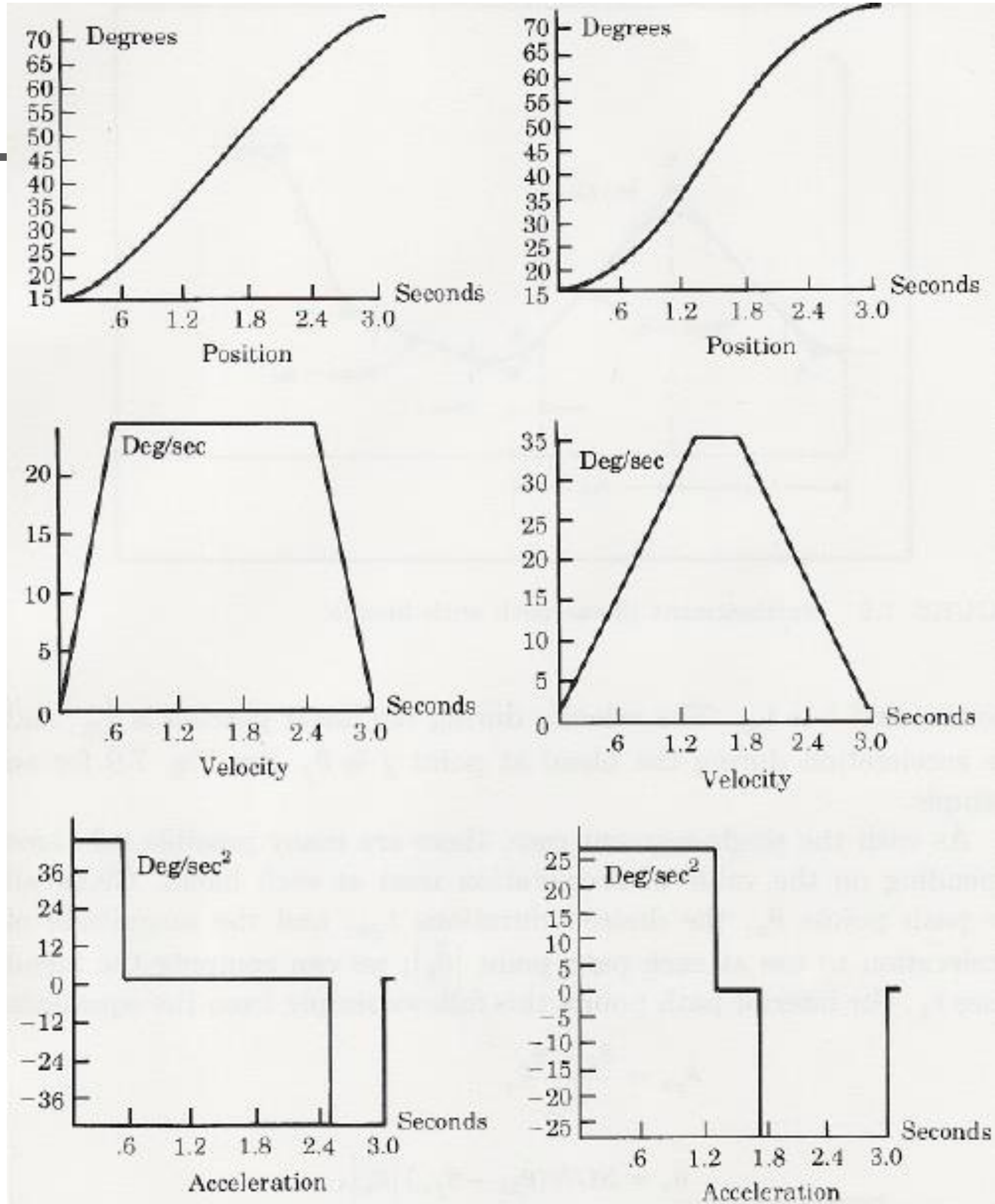
7.2.4 Linear function with parabolic blends

- Which one has a higher acceleration during the blends?





$$t_b = \frac{t}{2} - \frac{\sqrt{\ddot{\theta}^2 t^2 - 4\ddot{\theta}(\theta_f - \theta_0)}}{2\ddot{\theta}}$$

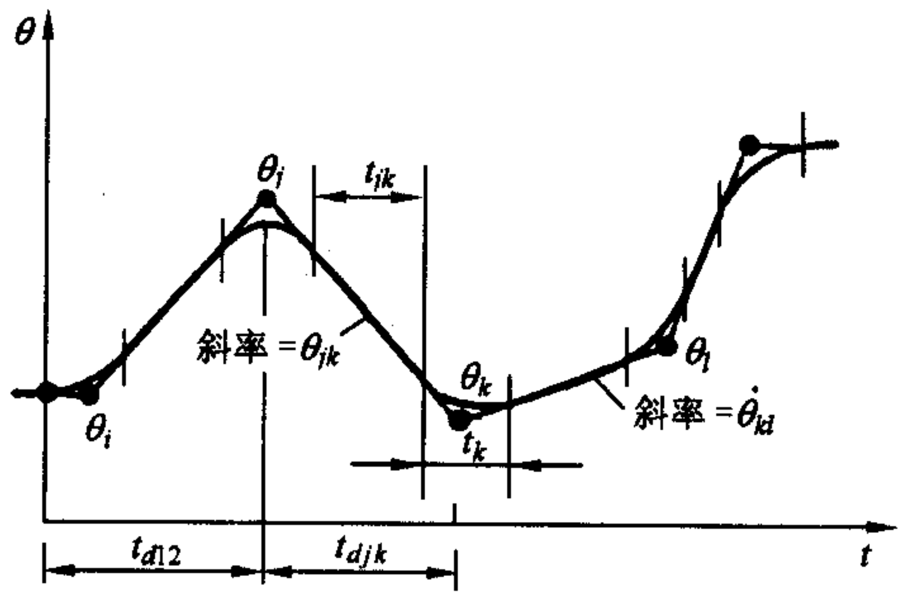


Degrees
Seconds
Position
Degrees
Seconds
Position
Deg/sec
Seconds
Velocity
Deg/sec
Seconds
Velocity
Deg/sec²
Seconds
Acceleration
Deg/sec²
Seconds
Acceleration

7.2.5 Linear function with parabolic blends for a path with via points

过路径点的用抛物线过渡的线性插值

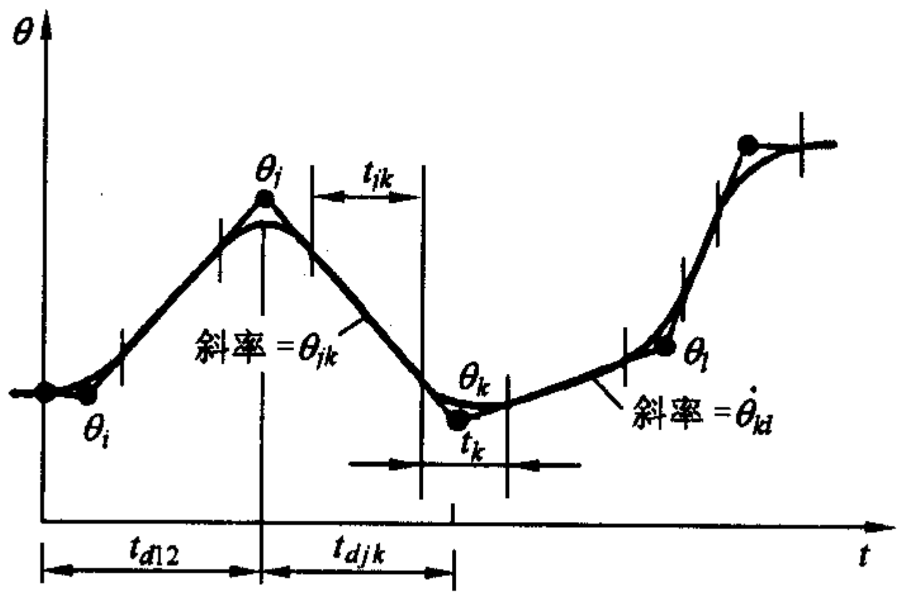
- Below shows a set of joint space via points for some joints. Linear functions connect the via points, and parabolic blend regions are added around each via point.



Multi-segment linear path with blends.

7.2.5 Linear function with parabolic blends for a path with via points

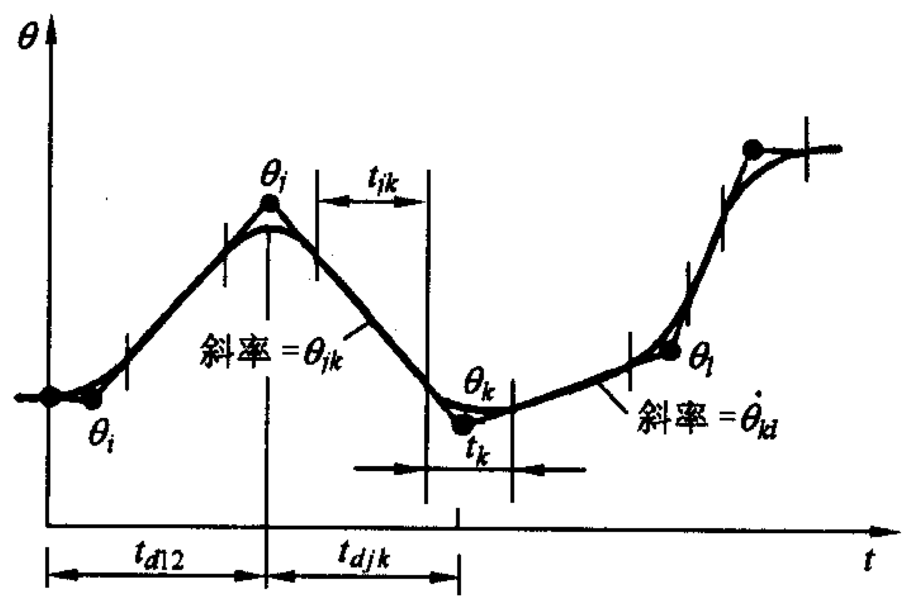
- Given:
  - positions
  - desired time durations
  - the magnitudes of the accelerations
- Compute:
  - blends times
  - straight segment times
  - slopes (velocities)
  - signed accelerations



7.2.5 Linear function with parabolic blends for a path with via points

- Inside segment:

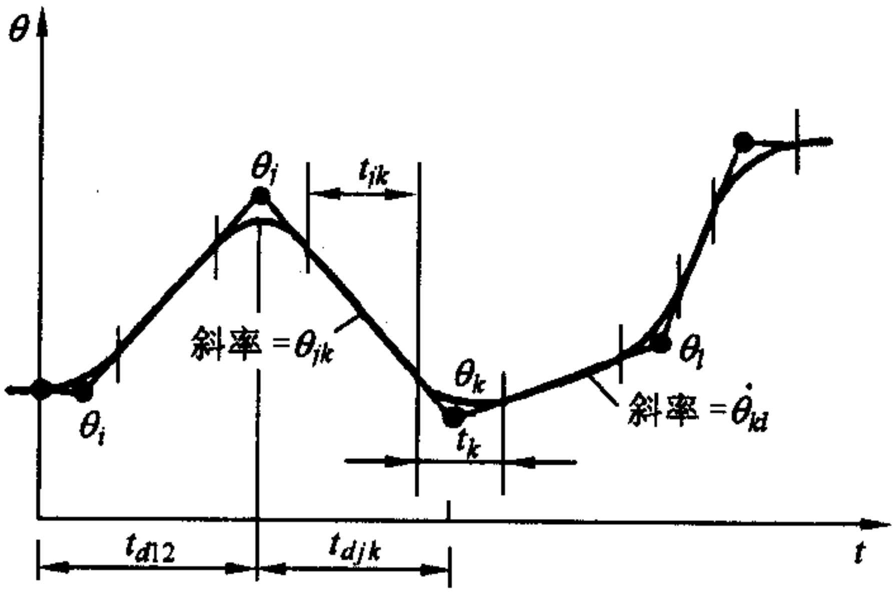
$$
\left.
\begin{aligned}
\dot{\theta}_{jk} &= \frac{\theta_k - \theta_j}{t_{djk}} \\
\ddot{\theta}_k &= \operatorname{sgn}(\dot{\theta}_{kl} - \dot{\theta}_{jk})\left|\ddot{\theta}_k\right| \\
t_k &= \frac{\dot{\theta}_{kl} - \dot{\theta}_{jk}}{\ddot{\theta}_k} \\
t_{jk} &= t_{djk} - \frac{1}{2}t_j - \frac{1}{2}t_k
\end{aligned}
\right\}
$$



7.2.5 Linear function with parabolic blends for a path with via points

- First segment:

$$
\left.
\begin{aligned}
&\ddot{\theta}_1 = \mathrm{sgn}(\dot{\theta}_2 - \dot{\theta}_1)\left|\ddot{\theta}_1\right| \\
&t_1 = t_{d12} - \sqrt{t_{d12}^2 - \frac{2(\theta_2 - \theta_1)}{\ddot{\theta}_1}} \\
&\dot{\theta}_{12} = \frac{\theta_2 - \theta_1}{t_{d12} - \frac{1}{2}t_1} \\
&t_{12} = t_{d12} - t_1 - \frac{1}{2}t_2
\end{aligned}
\right\}
$$



7.2.5 Linear function with parabolic blends for a path with via points

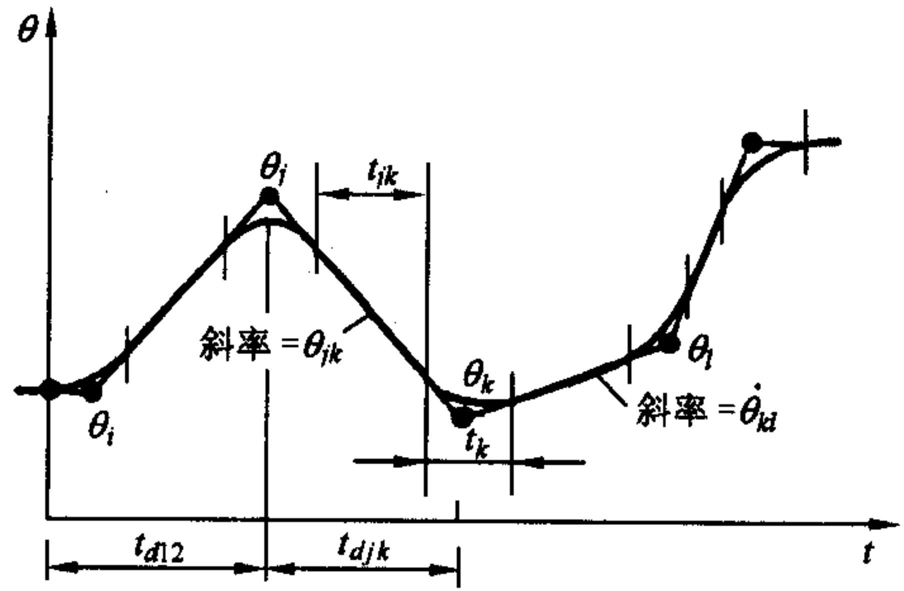
- Last segment:

$$\ddot{\theta}_n = \operatorname{sgn}(\dot{\theta}_{n-1} - \dot{\theta}_n)\left|\ddot{\theta}_n\right|$$

$$t_n = t_{d(n-1)n} - \sqrt{t_{d(n-1)n}^2 + \frac{2(\theta_n - \theta_{n-1})}{\ddot{\theta}_n}}$$

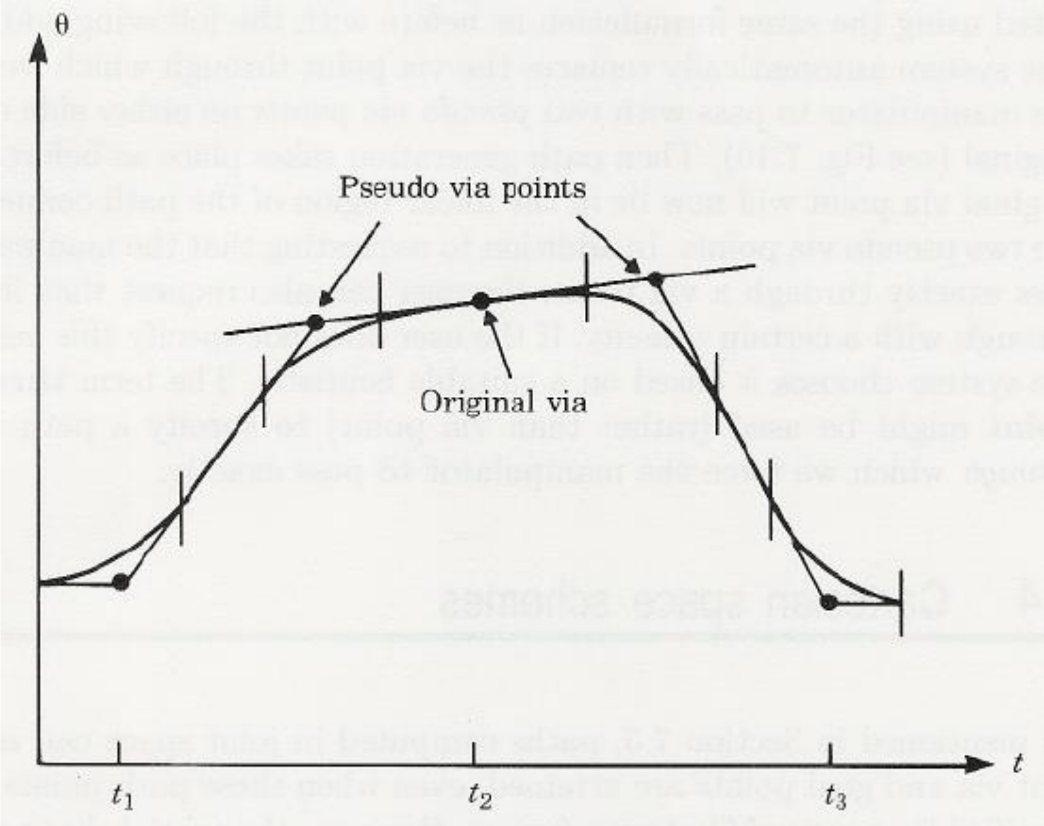
$$\dot{\theta}_{(n-1)n} = \frac{\theta_n - \theta_{n-1}}{t_{d(n-1)n} - \frac{1}{2}t_n}$$

$$t_{(n-1)n} = t_{d(n-1)n} - t_n - \frac{1}{2}t_{n-1}$$



7.2.5 Linear function with parabolic blends for a path with via points

- To go through the actual via points:
  - Introduce "Pseudo Via Points"
  - Use sufficiently high acceleration



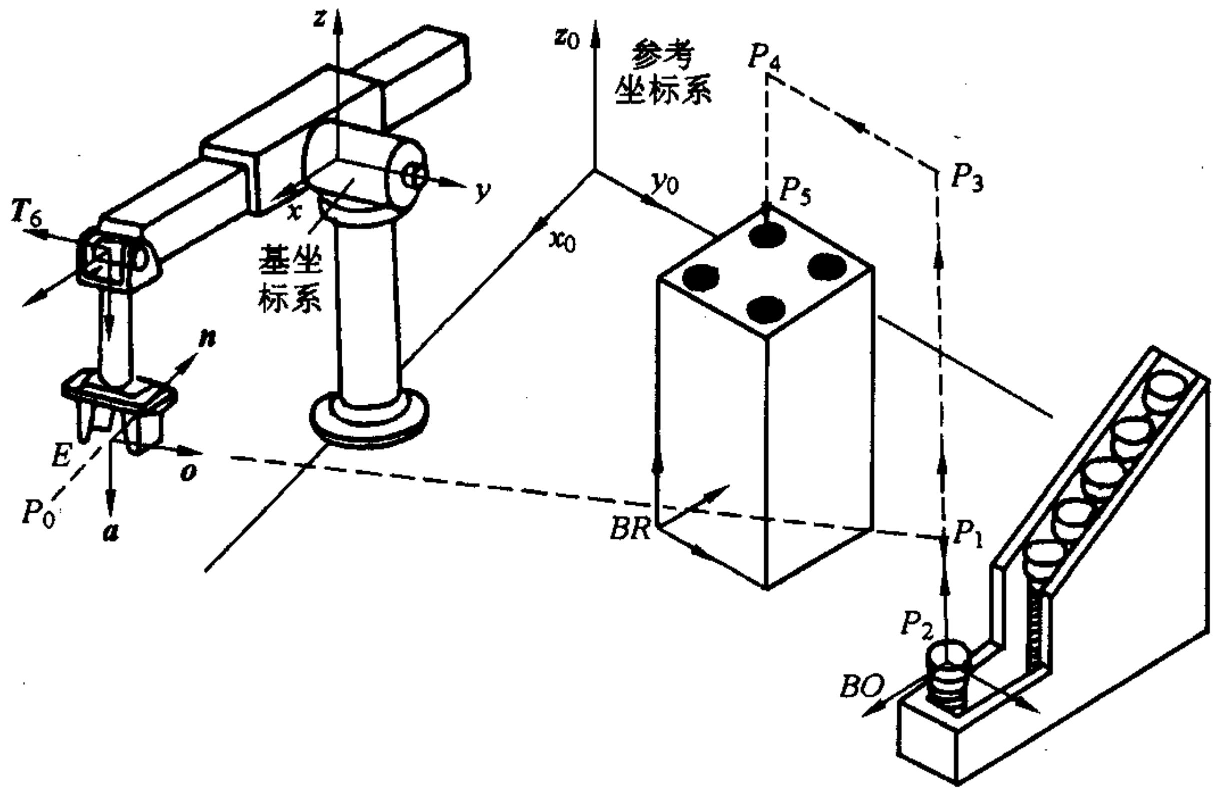
Ch.7 Trajectory Planning of Robots

7.3 Cartesian-Space Schemes

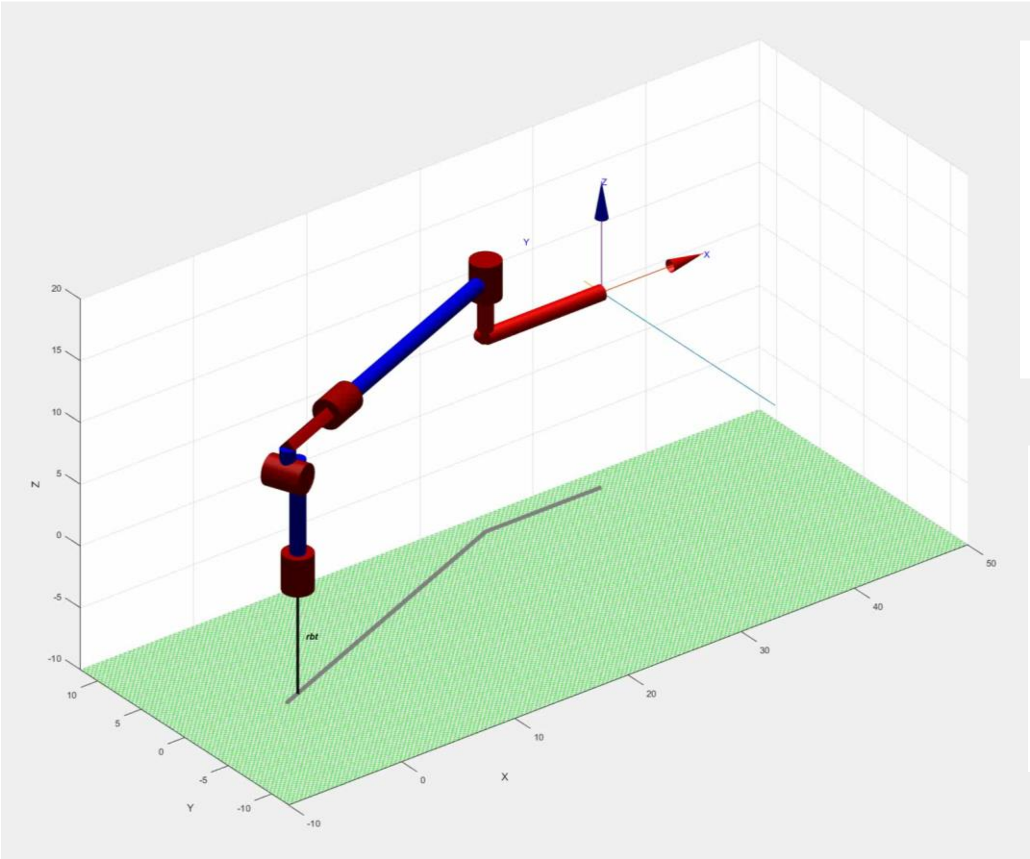
- When path shapes are described in terms of functions of Cartesian position and orientation, we can also specify the spatial shape of the path between path points.
- The most common path shape is a straight line; but circular, sinusoidal, or other path shapes could be used.
- Cartesian schemes are more computationally expensive to execute since at run time, inverse kinematics must be solved at the path update rate.

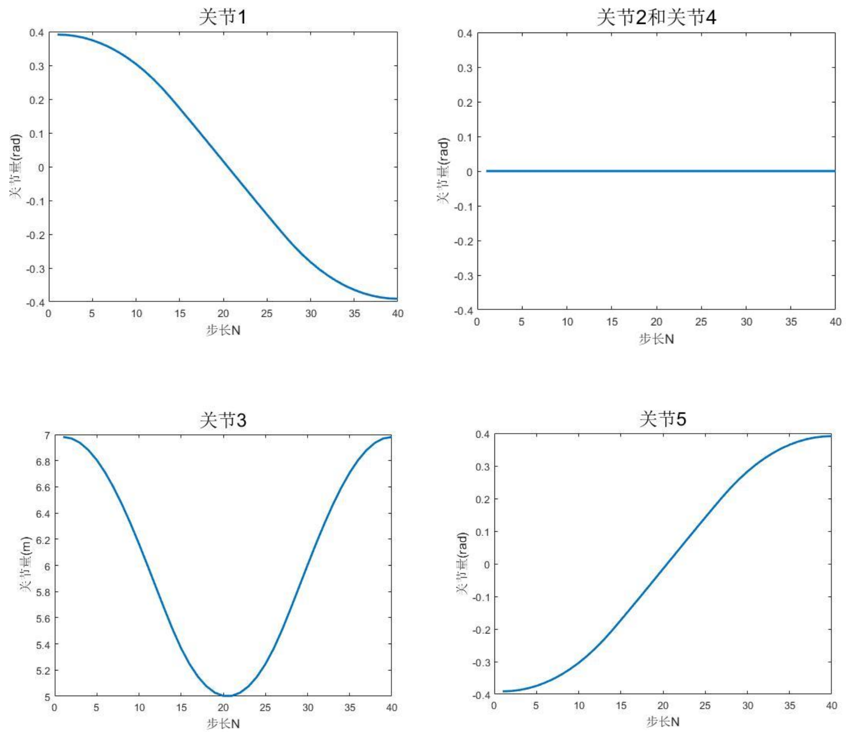
7.3 Cartesian-Space Schemes

- Description of a task

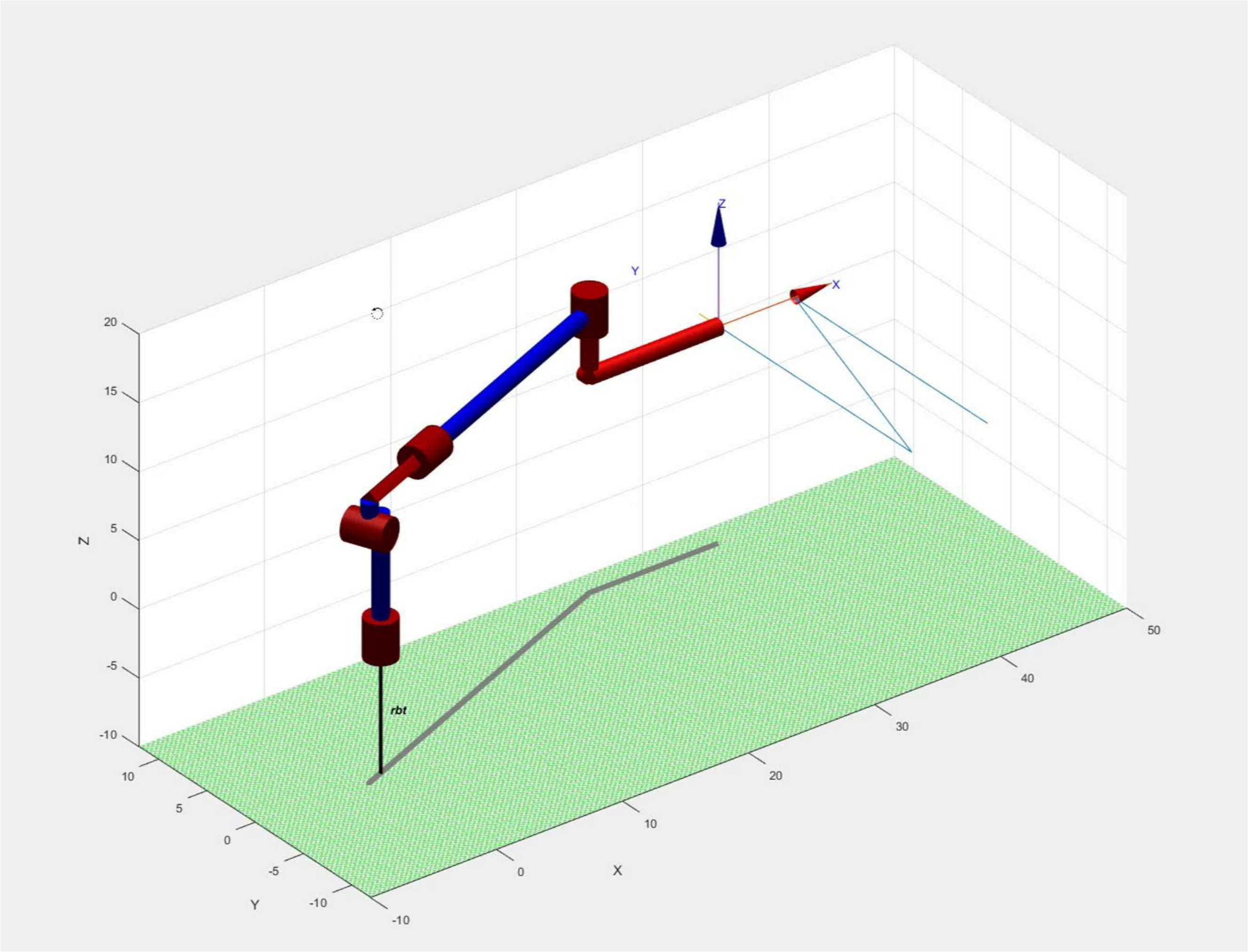


7.3 Cartesian-Space Schemes





7.3 Cartesian-Space Schemes



7.3 Cartesian-Space Schemes

若已知起始点坐标$P_1(x_1,y_1,z_1)$，终止点坐标$P_2(x_2,y_2,z_2)$

可求得该直线路径总长度为：

$$S=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}$$

此外，若末端执行器速度为V，插补周期为ΔT

则理论差值步数为：

$$N=\frac{S}{V\Delta T}$$

7.3 Cartesian-Space Schemes

所以每次插值各个方向上的增量为：

$$\begin{cases} \Delta x = \dfrac{x_2 - x_1}{N} \\ \Delta y = \dfrac{y_2 - y_1}{N} \\ \Delta z = \dfrac{z_2 - z_1}{N} \end{cases}$$

则每个插值点坐标为：

$$\begin{cases} x_{i+1} = x_i + \Delta x \\ y_{i+1} = y_i + \Delta y \\ z_{i+1} = z_i + \Delta z \end{cases}$$

其中i为正整数，取值范围为$[1, N]$

7.3 Cartesian-Space Schemes

求出每个插值点的坐标后，结合姿态信息即可求得每个路径点的位姿:

$$P_i = \begin{bmatrix} \boldsymbol{n}_i & \boldsymbol{o}_i & \boldsymbol{a}_i & \boldsymbol{p}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_{ix} & o_{ix} & a_{ix} & p_{ix} \\ n_{iy} & o_{iy} & a_{iy} & p_{iy} \\ n_{iz} & o_{iz} & a_{iz} & p_{iz} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

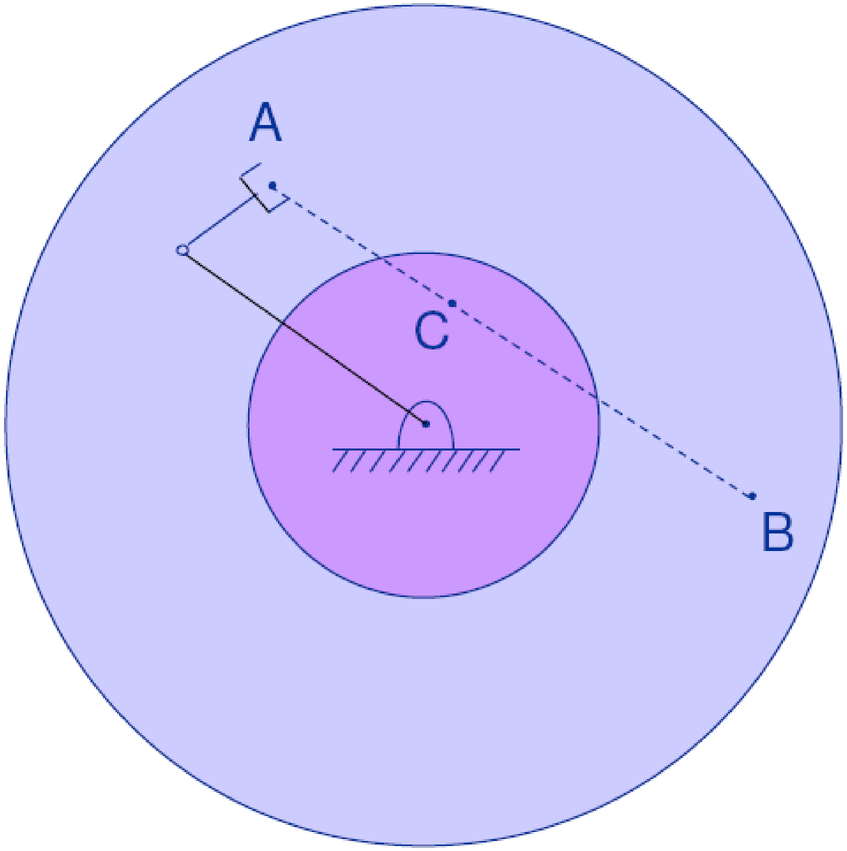
通过逆运动学求解可得到每个关节的变量序列：

$$\theta_{\mathrm{i}} = [\vartheta_1, \vartheta_2 \ldots \vartheta_n]$$

7.3 Cartesian-Space Schemes

- Cartesian planning difficulties (1/3):

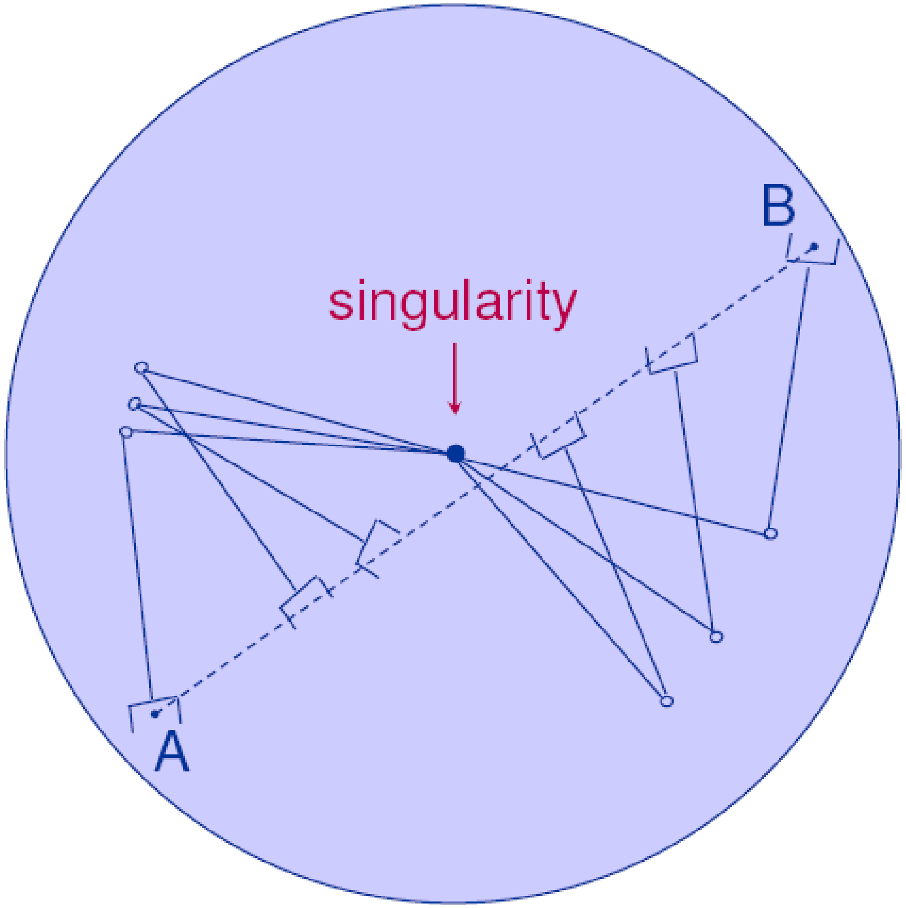
Initial (A) and Goal (B) Points are reachable, but intermediate points (C) unreachable.



7.3 Cartesian-Space Schemes

- Cartesian planning difficulties (2/3):

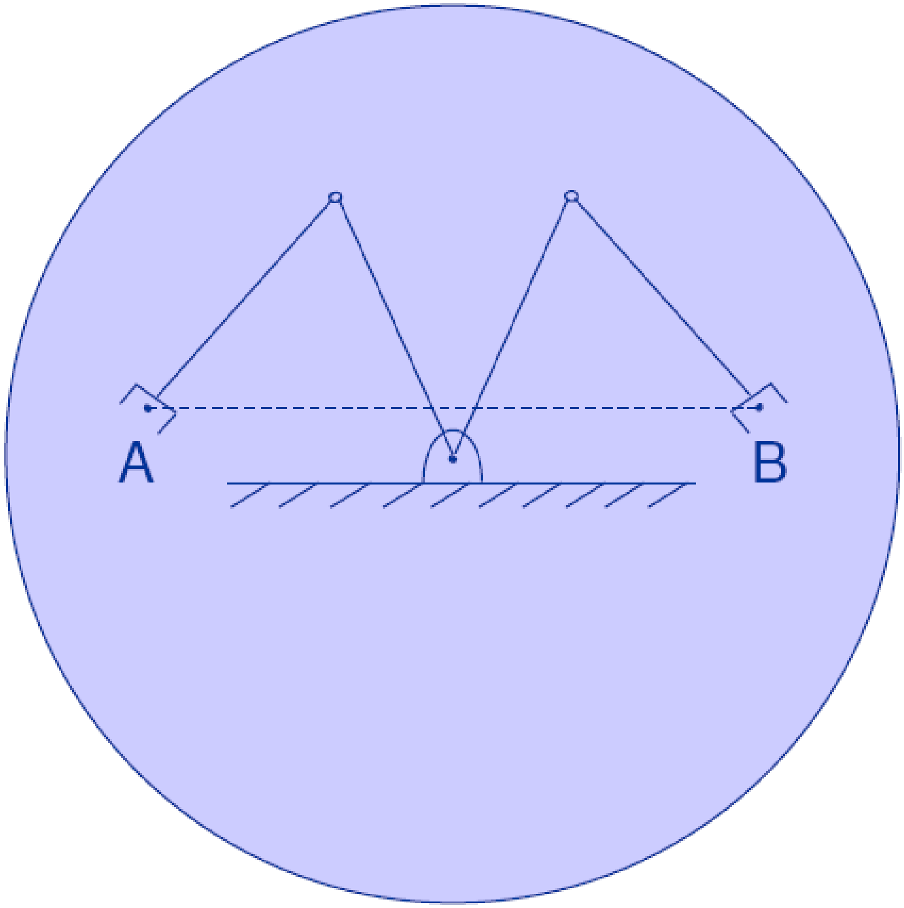
Approaching singularities some joint velocities go to ∞ causing deviation from the path.



7.3 Cartesian-Space Schemes

- Cartesian planning difficulties (3/3):

Start point (A) and goal point (B) are reachable in different joint space solutions (The middle points are reachable from below.)



Ch.7 Trajectory Planning of Robots

7.4 Path Generation at Real-Time

- At run time the path generator routine constructs the trajectory, usually in terms of $\theta, \dot{\theta}$, and $\ddot{\theta}$, and feeds this information to the manipulator's control system.
- This path generator computes the trajectory at the path update rate.

7.4.1 Generation of joint space paths

- In the case of cubic splines, the path generator simply computes (7.3) and (7.4) as *t* is advanced. When the end of one segment is reached, a new set of cubic coefficients is recalled, *t* is set back to zero, and the generation continues.

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \qquad (7.3)$$

$$\left.\begin{aligned} \dot{\theta}(t) &= a_1 + 2a_2 t + 3a_3 t^2 \\ \ddot{\theta}(t) &= 2a_2 + 6a_3 t \end{aligned}\right\} \qquad (7.4)$$

7.4.1 Generation of joint space paths

- In the case of linear splines with parabolic blends, the value of time, *t*, is checked on each update to determine whether we are currently in the linear or the blend portion of the segment.
- In the linear portion, the trajectory for each joint is calculated as

$$\left.\begin{aligned} \theta &= \theta_j + \dot{\theta}_{jk} t \\ \dot{\theta} &= \dot{\theta}_{jk} \\ \ddot{\theta} &= 0 \end{aligned}\right\} \qquad (7.41)$$

7.4.1 Generation of joint space paths

- In the case of linear splines with parabolic blends, the value of time, *t*, is checked on each update to determine whether we are currently in the linear or the blend portion of the segment.
- In the blend region, the trajectory for each joint is calculated as

$$\left.\begin{aligned} \theta &= \theta_j + \dot{\theta}_{jk}(t - t_{inb}) + \frac{1}{2}\ddot{\theta}_k t_{inb}^2 \\ \dot{\theta} &= \dot{\theta}_{jk} + \ddot{\theta}_k t_{inb} \\ \ddot{\theta} &= \ddot{\theta}_k \end{aligned}\right\} \qquad (7.42)$$

where $t_{inb} = t - \left(\frac{1}{2}t_j + t_{jk}\right)$

7.4.2 Generation of Cartesian space paths

- In the case of linear spline with parabolic blends path. Rewrite (7.41) and (7.42) with the symbol X representing a component of the Cartesian position and orientation vector.
- In the linear portion of the segment, each degree of freedom in X is calculated as

$$\left.\begin{aligned} x &= x_j + \dot{x}_{jk} t \\ \dot{x} &= \dot{x}_{jk} \\ \ddot{x} &= 0 \end{aligned}\right\} \qquad (7.44)$$

7.4.2 Generation of Cartesian space paths

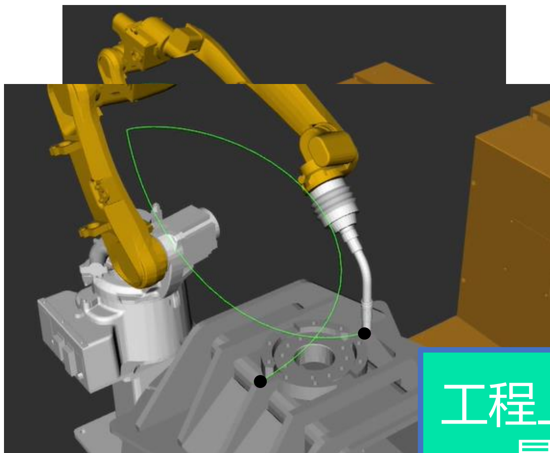
- In the case of linear spline with parabolic blends path. Rewrite (7.41) and (7.42) with the symbol X representing a component of the Cartesian position and orientation vector.
- In the blend region, the trajectory for each degree of freedom is calculated as

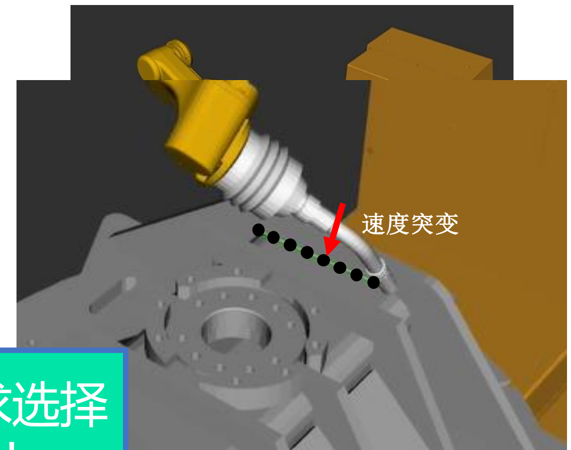
$$\left.\begin{aligned} t_{inb} &= t - \left(\frac{1}{2}t_j + t_{jk}\right) \\ x &= x_j + \dot{x}_{jk}\left(t - t_{inb}\right) + \frac{1}{2}\ddot{x}_k t^2{}_{inb} \\ \dot{x} &= \dot{x}_{jk} + \ddot{x}_k t_{inb} \\ \ddot{x} &= \ddot{x}_k \end{aligned}\right\} \qquad (7.45)$$

7.4.2 Generation of Cartesian space paths

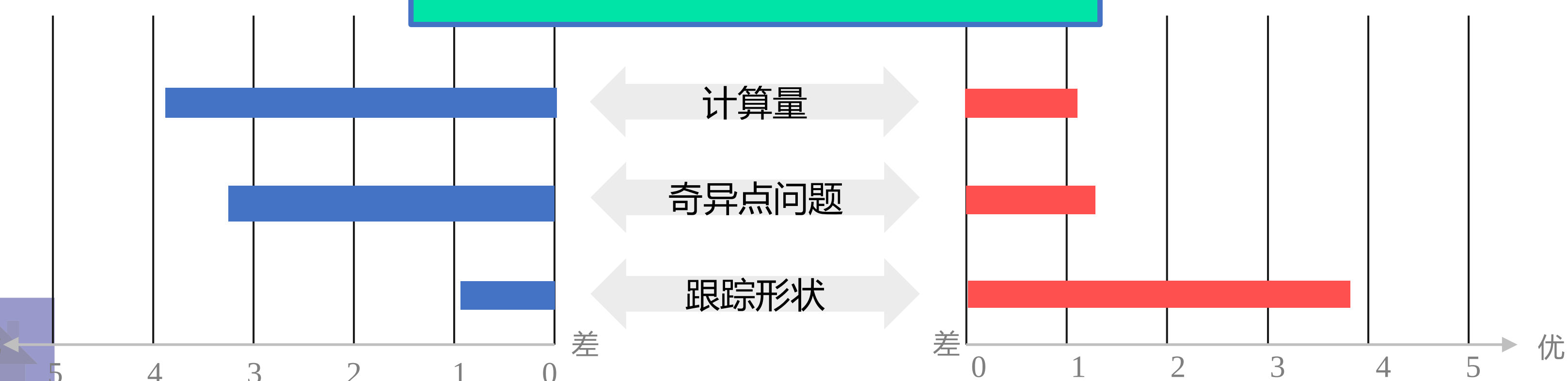
- Finally, this Cartesian trajectory (X, $\dot{X}$, and $\ddot{X}$) must be converted into **equivalent joint space** quantities.
- A complete analytical solution to this problem would use:
  - inverse kinematics to calculate joint positions,
  - inverse Jacobian for velocities,
  - inverse Jacobian plus its derivative for accelerations.

轨迹规划的两种类型对比





工程上要根据实际作业需求选择最合适的轨迹规划方法！



7.5 Summary

- General Considerations in Robot Trajectory Planning
- Joint-Space Schemes
  - Cubic polynomials
  - Cubic polynomials for a path with via points
  - Higher-order polynomials
  - Linear function with parabolic blends
  - Linear function with parabolic blends for a path with via points
- Cartesian-Space Schemes
  - Track of any desired shape
  - More expensive at run time
  - Discontinuity problems
- Real Time Generation of Planning Trajectories

谢 谢！