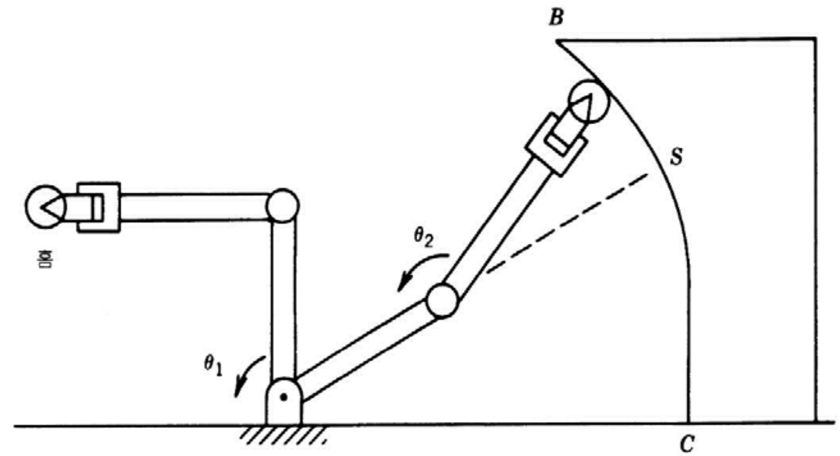
Chapter 5

Trajectory Planning

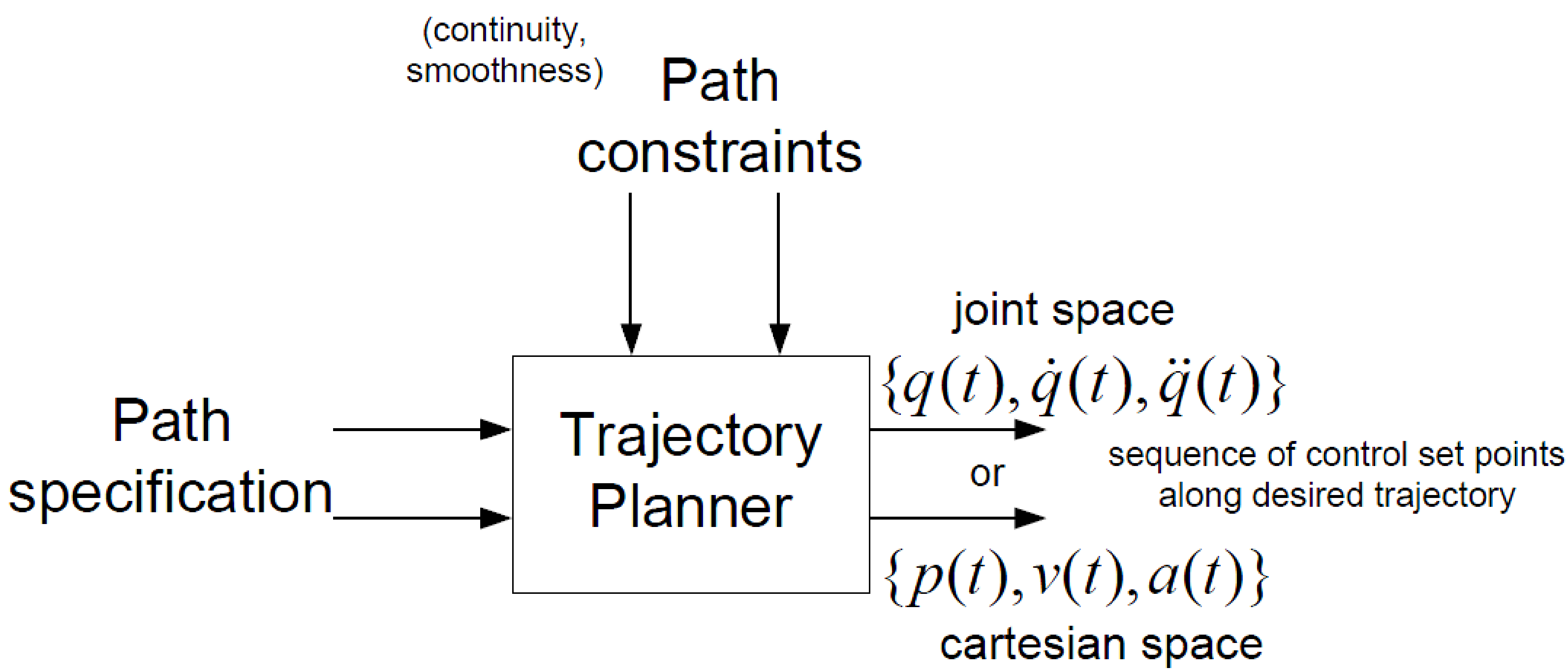
5.1 INTRODUCTION



In this chapters…….

- Path and trajectory planning relates to the way that a robot is moved from one location to another in a controlled manner.
- The sequence of movements for a controlled movement between motion segments, in straight-line motions or in sequential motions.
- It requires the use of both kinematics and dynamics of robots.

5.1 INTRODUCTION



Consideration

- In order to determine the robot trajectory, consider the location, velocity, position of desired end-effector as well as kinematic constraints condition of robot and dynamic characteristic

- kinematic constraints condition : each joint's range of motion, work space's limitation, singularity problem

- Dynamic characteristic : joint torque's limitation, maximum velocity, acceleration

As shown in Fig. 5.A1, the basic problem is to move the manipulator from an initial position to some desired final position.

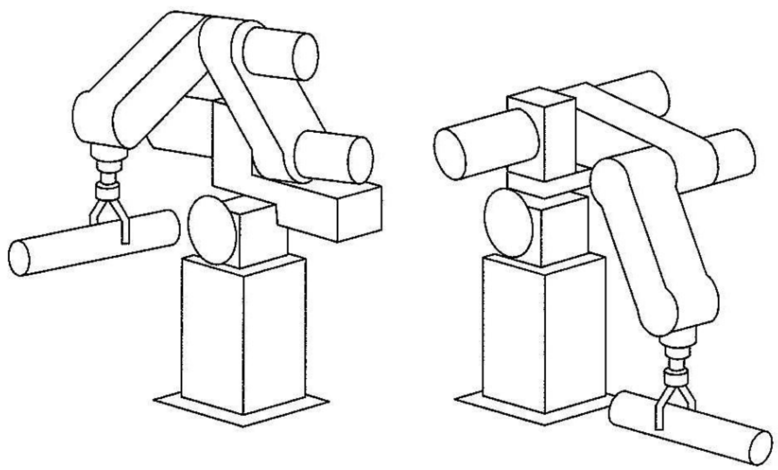


FIGURE 5.A1 : In executing a trajectory, a manipulator moves from its initial position to a desired goal position in a smooth manner.

In general, this motion involves both a change in orientation and a change in position of the tool relative to the station.

5.2 PATH VS. TRAJECTORY

- **Path:** A sequence of robot configurations in a particular order without regard to the timing of these configurations.
- **Trajectory:** It is concerned about when each part of the path must be attained, thus specifying timing.

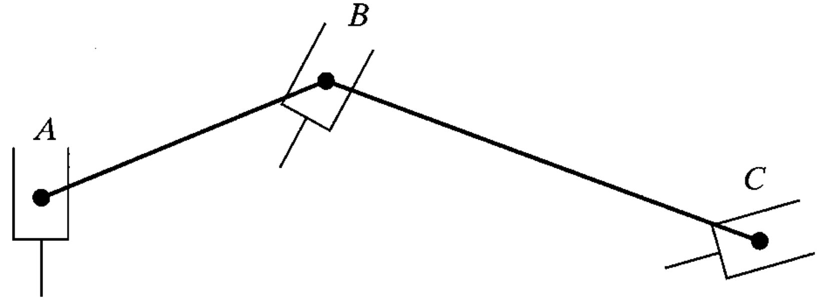


Fig. 5.1 Sequential robot movements in a path.

5.3 JOINT-SPACE VS. CARTESIAN-SPCAE DESCRIPTIONS

- Joint-space description:
  - The description of the motion to be made by the robot by its joint values.
  - The motion between the two points is unpredictable.
- Cartesian space description:
  - The motion between the two points is known at all times and controllable.
  - It is easy to visualize the trajectory, but it is difficult to ensure that singularity.

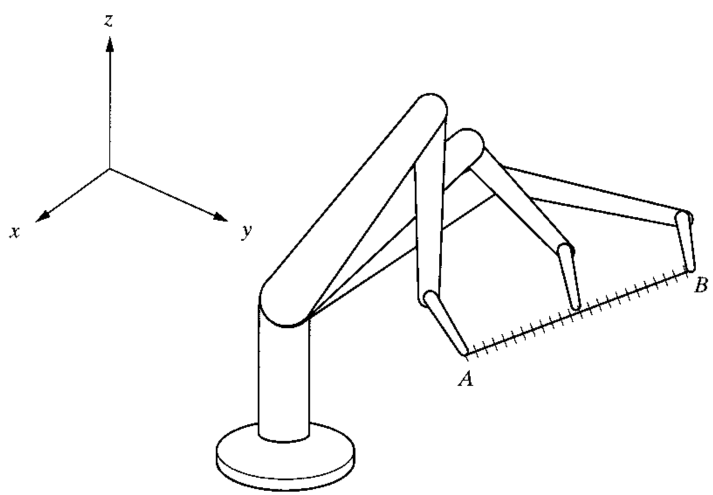


Fig. 5.2 Sequential motions of a robot to follow a straight line.

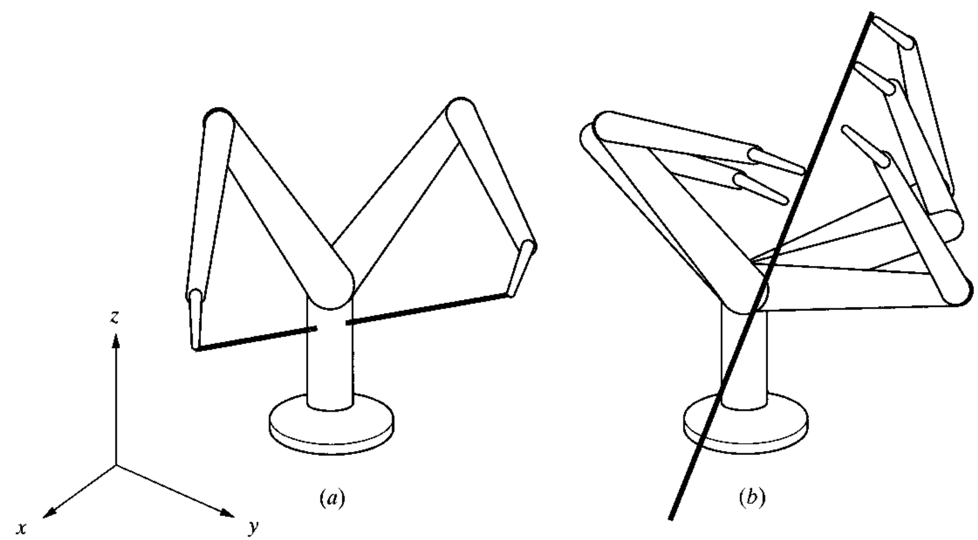
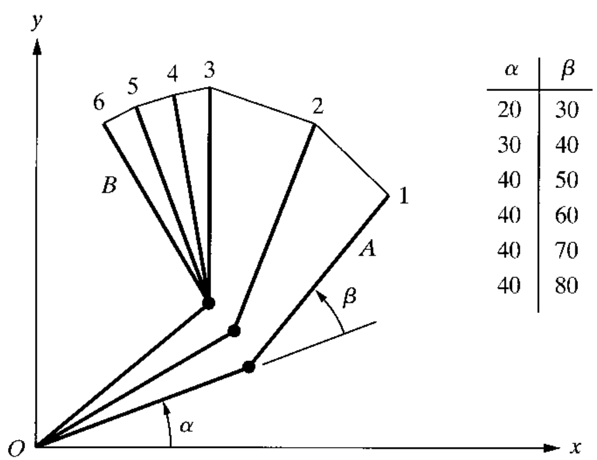


Fig. 5.3 Cartesian-space trajectory (a) The trajectory specified in Cartesian coordinates may force the robot to run into itself, and (b) the trajectory may requires a sudden change in the joint angles.

5.4 BASICS OF TRAJECTORY PLANNING

- Let's consider a simple 2 degree of freedom robot.
- We desire to move the robot from Point A to Point B.
- Let's assume that both joints of the robot can move at the maximum rate of 10 degree/sec.



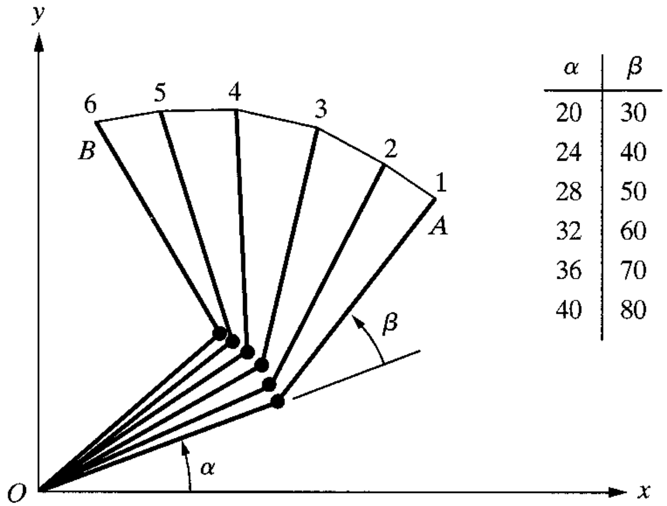
α	β
20	30
30	40
40	50
40	60
40	70
40	80

Fig. 5.4 Joint-space nonnormalized movements of a robot with two degrees of freedom.

- **Move the robot from A to B, to run both joints at their maximum angular velocities.**
- **After 2 [sec], the lower link will have finished its motion, while the upper link continues for another 3 [sec].**
- **The path is irregular and the distances traveled by the robot's end are not uniform.**

5.4 BASICS OF TRAJECTORY PLANNING

- Let's assume that the motions of both joints are normalized by a <u>common factor</u> such that the joint with smaller motion will move proportionally slower and the both joints will start and stop their motion simultaneously.



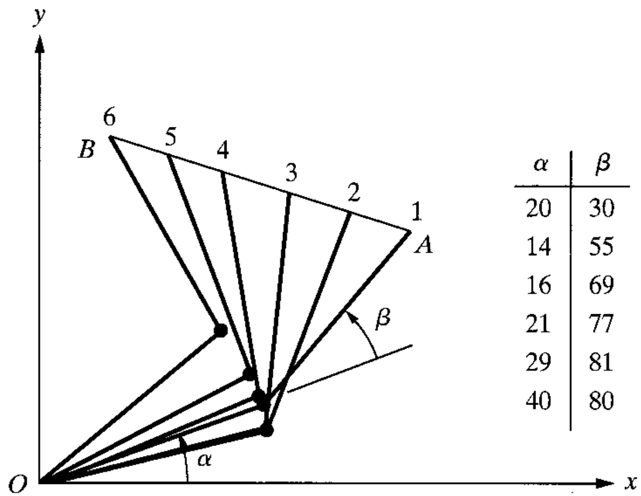
α	β
20	30
24	40
28	50
32	60
36	70
40	80

Fig. 5.5 Joint-space, normalized movements of a robot with two degrees of freedom.

- **Both joints move at different speeds, but move continuously together.**
- **The resulting trajectory will be different.**

5.4 BASICS OF TRAJECTORY PLANNING

- Let's assume that the robot's hand follows a known path between points A and B in a straight line.
- The simplest solution would be to draw a line between points A and B, so called interpolation.



α	β
20	30
14	55
16	69
21	77
29	81
40	80

Fig. 5.6 Cartesian-space movements of a two-degree-of-freedom robot.

- **Divide the line into five segments and solve for necessary angles α and β at each point.**
- **The joint angles are not uniformly changing.**

5.4 BASICS OF TRAJECTORY PLANNING

- Let's assume that the robot's hand follows a known path between point A and B with straight line.
- The simplest solution would be to draw a line between points A and B, so called interpolation.

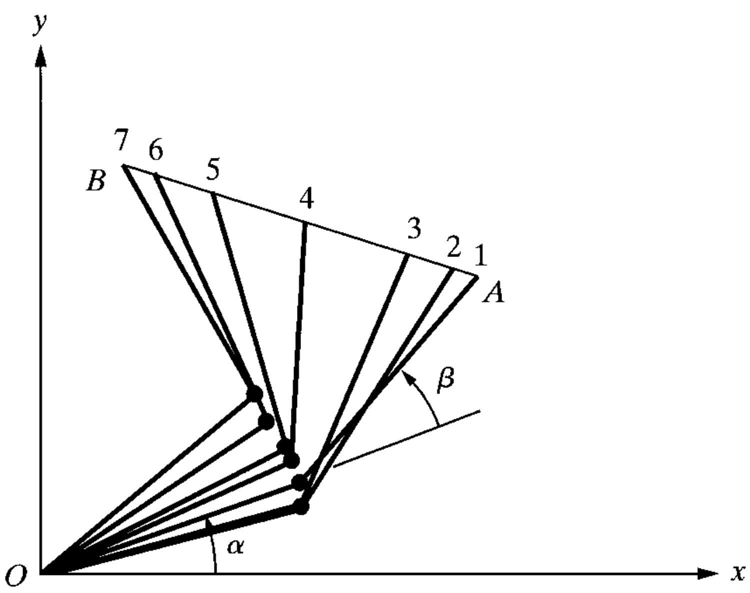


Fig. 5.7 Trajectory planning with an acceleration-deceleration regiment.

- **It is assumed that the robot's actuators are strong enough to provide large forces necessary to accelerate and decelerate the joints as needed.**
- **Divide the segments differently.**
  - **The arm moves at smaller segments as we speed up at the beginning.**
  - **Go at a constant cruising rate.**
  - **Decelerate with smaller segments as approaching point B.**

5.4 BASICS OF TRAJECTORY PLANNING

- Next level of trajectory planning is between multiple points for continuous movements.
- Stop-and-go motion create jerky motions with unnecessary stops.
- **Blend the two portions of the motion at point B.**
- **Specify two via point D and E before and after point B**

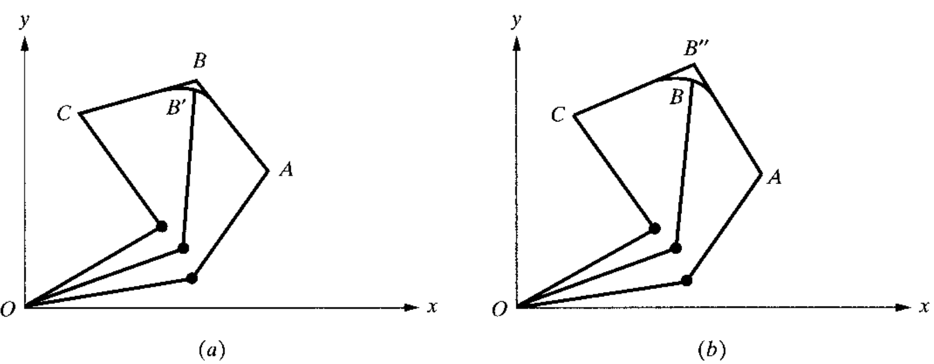


Fig. 5.8 Blending of different motion segments in a path.

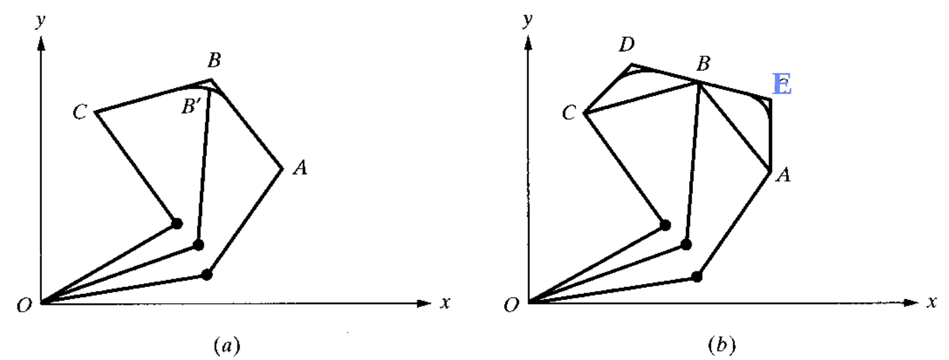


Fig. 5.9 An alternative scheme for ensuring that the robot will go through a specified point during blending of motion segments. Two via points D and E are picked such that point B will fall on the straight-line section of the segment ensuring that the robot will pass through point B.

Figure 5.A2 shows one possibility where $\ddot{\theta}$ was chosen quite high. In this case we quickly accelerate, then coast at constant velocity, and then decelerate.

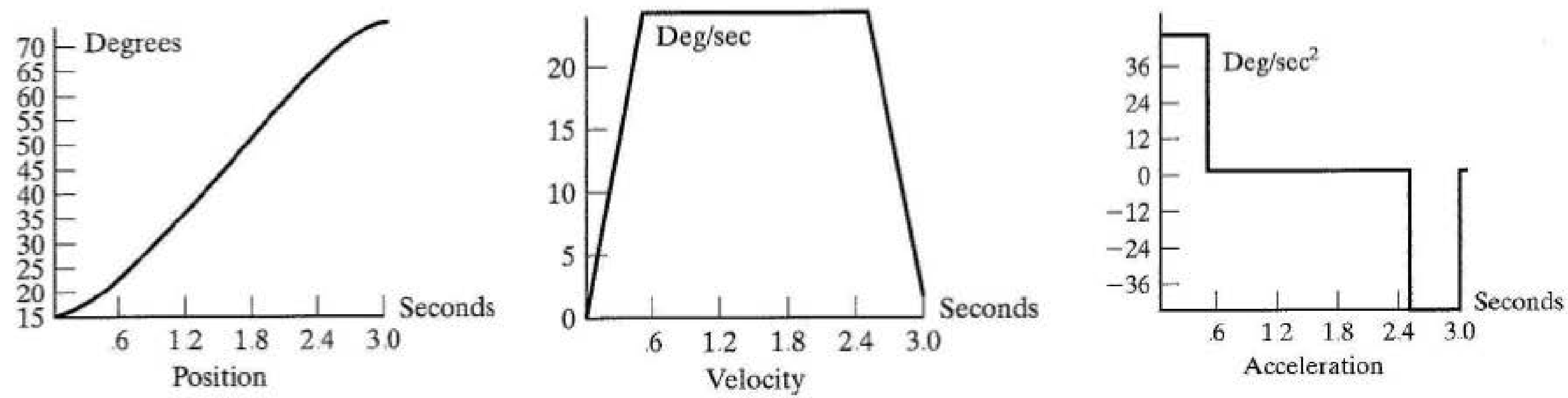


FIGURE 5.A2 : Position, velocity, and acceleration profiles for linear interpolation with parabolic blends. The set of curves on the left are based on a higher acceleration during the blends than is that on the right.

5.5 JOINT-SPACE TRAJECTORY PLANNING

5.5.1 Third-Order Polynomial Trajectory Planning

- How the motions of a robot can be planned in joint-space with controlled characteristics.
- Polynomials of different orders
- Linear functions with parabolic blends

- **The initial location and orientation of the robot is known, and using the inverse kinematic equations, we find the final joint angles for the desired position and orientation.**

$$\theta(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3$$

$$\theta(t_i) = \theta_i$$
$$\theta(t_f) = \theta_f$$
$$\dot{\theta}(t_i) = 0$$
$$\dot{\theta}(t_f) = 0$$

- **Initial Condition**

$$\dot{\theta}(t) = c_1 + 2c_2 t + 3c_3 t^2$$

- **First derivative of the polynomial of equation**

$$\theta(t_i) = c_0 = \theta_i$$
$$\theta(t_f) = c_0 + c_1 t_f + c_2 t_f^2 + c_3 t_f^3$$
$$\dot{\theta}(t_i) = c_1 = 0$$
$$\dot{\theta}(t_f) = c_1 + 2c_2 t_f + 3c_3 t_f^2 = 0$$

- **Substituting the initial and final conditions**

5.5 JOINT-SPACE TRAJECTORY PLANNING

5.5.2 Fifth-Order Polynomial Trajectory Planning

- Specify the initial and ending accelerations for a segment.
- To use a fifth-order polynomial for planning a trajectory, the total number of boundary conditions is 6.

- **Calculation of the coefficients of a fifth-order polynomial with position, velocity and an acceleration boundary conditions can be possible with below equations.**

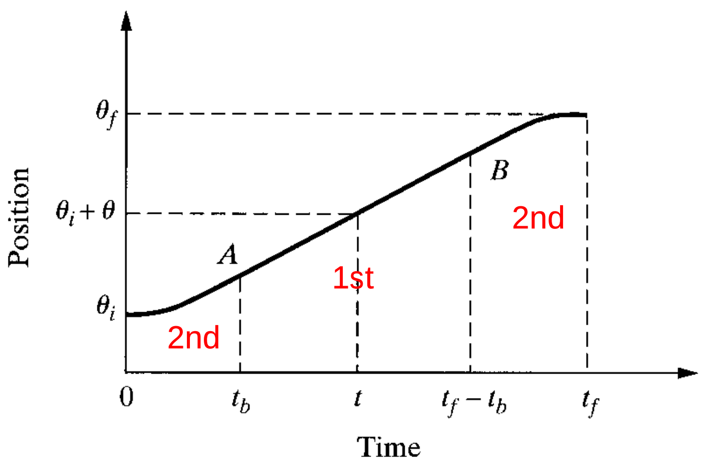
$$\theta(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5$$

$$\dot{\theta}(t) = c_1 + 2c_2 t + 3c_3 t^2 + 4c_4 t^3 + 5c_5 t^4$$

$$\ddot{\theta}(t) = 2c_2 + 6c_3 t + 12c_4 t^2 + 20c_5 t^3$$

5.5 JOINT-SPACE TRAJECTORY PLANNING

- Linear segment can be blended with parabolic sections at the beginning and the end of the motion segment, creating continuous position and velocity.
- Acceleration is constant for the parabolic sections, yielding a continuous velocity at the common points A and B.



$$\theta(t) = c_0 + c_1 t + \frac{1}{2} c_2 t^2 \qquad \Rightarrow \qquad \theta(t) = \theta_i + \frac{1}{2} c_2 t^2$$

$$\dot{\theta}(t) = c_1 + c_2 t \qquad \Rightarrow \qquad \dot{\theta}(t) = c_2 t$$

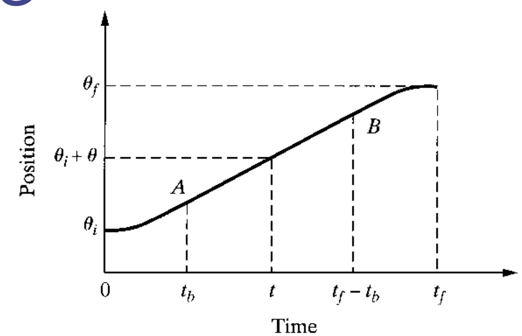
$$\ddot{\theta}(t) = c_2 \qquad \Rightarrow \qquad \ddot{\theta}(t) = c_2$$

Fig. 5.13 Scheme for linear segments with parabolic blends.

5.5 JOINT-SPACE TRAJECTORY PLANNING

$$\theta(t=0)=\theta_i=c_0$$
$$\dot{\theta}(t=0)=0=c_1$$
$$\ddot{\theta}(t)=c_2$$

$$\begin{cases} c_0=\theta_i \\ c_1=0 \\ c_2=\ddot{\theta} \end{cases}$$



Initial velocity 0, a constant known joint velocity ω in the linear portion, final velocity 0

$$\theta_A=\theta_i+\frac{1}{2}c_2t_b^{\ 2}$$
$$\dot{\theta}_A=c_2t_b=\omega$$
$$\theta_B=\theta_A+\omega((t_f-t_b)-t_b))=\theta_A+\omega(t_f-2t_b)$$
$$\dot{\theta}_B=\dot{\theta}_A=\omega$$

Constant joint velocity

$$\dot{\theta}_A=c_2t_b=\omega$$
$$\dot{\theta}_f=0$$

$$c_2=\frac{\omega}{t_b}$$
$$\theta_f=\theta_B+(\theta_A-\theta_i)=2\theta_A+\omega(t_f-2t_b)-\theta_i$$
$$=2\left(\theta_i+\frac{1}{2}c_2t_b^{\ 2}\right)+\omega(t_f-2t_b)-\theta_i$$
$$=\theta_i+c_2t_b^{\ 2}+\omega(t_f-2t_b)$$
$$=\theta_i+\left(\frac{\omega}{t_b}\right)t_b^{\ 2}+\omega(t_f-2t_b)$$
$$\therefore t_b=\frac{\theta_i-\theta_f+\omega t_f}{\omega}$$

5.5 JOINT-SPACE TRAJECTORY PLANNING

Linear function with parabolic blends for a path with via points

θ

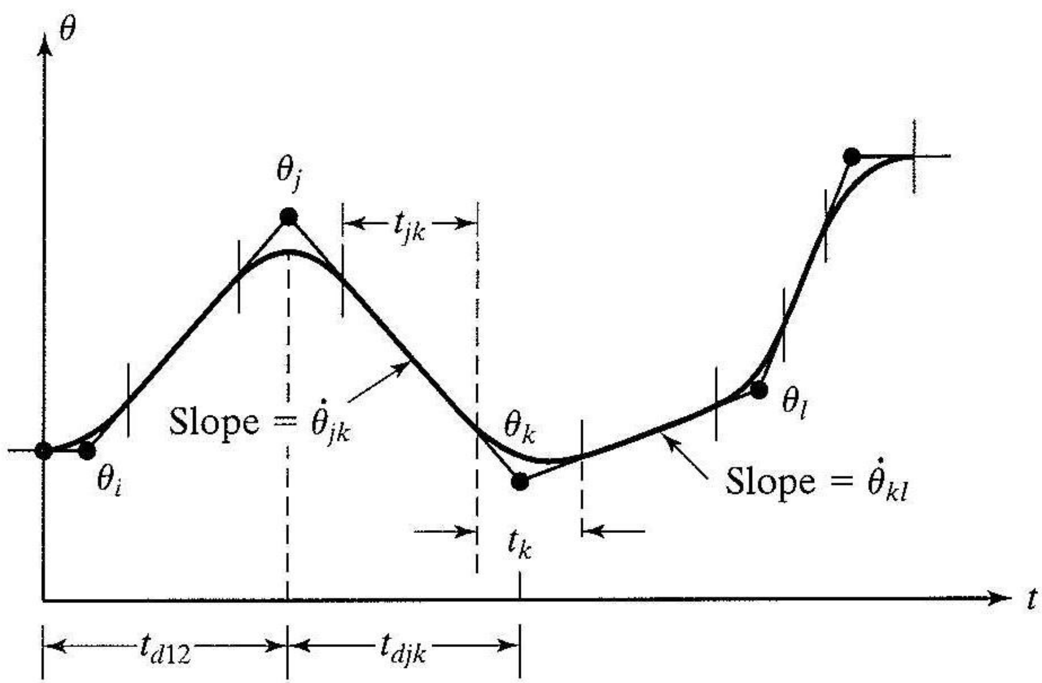


FIGURE 1.9 : Multisegment linear path with blends

5.5 JOINT-SPACE TRAJECTORY PLANNING

5.5.4 Linear Segments with Parabolic Blends and Via Points

- The position of the robot at time t_0 is known and using the inverse kinematic equations of the robot, the joint angles at via points and at the end of the motion can be found.
- To blend the motion segments together, the boundary conditions of each point to calculate the coefficients of the parabolic segments is used.
- Maximum allowable accelerations should not be exceeded.

5.5 JOINT-SPACE TRAJECTORY PLANNING

5.5.5 Higher Order Trajectories

- Incorporating the initial and final boundary conditions together with this information enables us to use higher order polynomials in the below form, so that the trajectory will pass through all specified points.

$$\theta(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + \cdots\cdots + c_{n-1} t^{n-1} + c_n t^n$$

- It requires extensive calculation for each joint and higher order polynomials.

- Combinations of lower order polynomials for different segments of the trajectory and blending together to satisfy all required boundary conditions is required.

  - 3-5-3 trajectory
  - 4-3-4 trajectory

5.6 CARTESIAN-SPACE TRAJECTORIES

- Cartesian-space trajectories relate to the motions of a robot relative to the Cartesian reference frame.

- In Cartesian-space, the joint values must be repeatedly calculated through the inverse kinematic equations of the robot.

- Computer Loop Algorithm

(1) Increment the time by $t=t+\Delta t$.

(2) Calculate the position and orientation of the hand based on the selected function for the trajectory.

(3) Calculate the joint values for the position and orientation through the inverse kinematic equations of the robot.

(4) Send the joint information to the controller.

(5) Go to the beginning of the loop

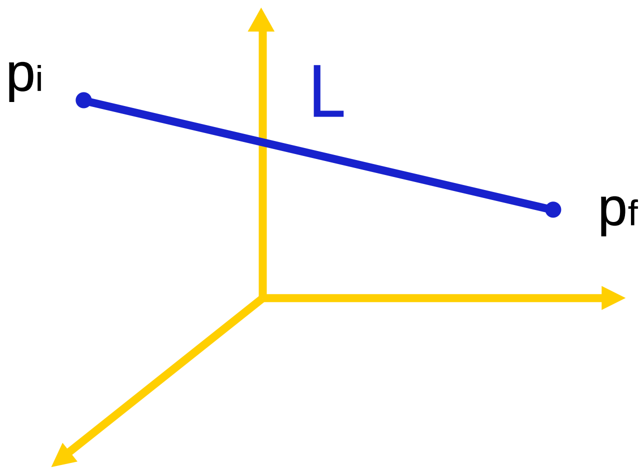
5.6 CARTESIAN-SPACE TRAJECTORIES

- If possible, we still prefer to plan Cartesian trajectories separately for position and orientation

- The number of knots to be interpolated in the Cartesian space is typically low (e.g., 2 knots for a PTP motion, 3 if a “via point” is added): use simple interpolating paths, such as straight lines...

5.6 CARTESIAN-SPACE TRAJECTORIES

- Planning a linear Cartesian path(position only)



GIVEN

$p_i,\ p_f,\ \mathrm{v}_{\max},\ \mathrm{a}_{\max}$

$v_i,\ \mathrm{v}_f\ (\text{typically} = 0)$

$$\mathrm{L} = \left\| p_f - p_i \right\|$$

path parameterization

$$p(s) = p_i + s(p_f - p_i)$$

$$\frac{p_f - p_i}{\left\| p_f - p_i \right\|}$$ = unit vector of line direction cosines

5.6 CARTESIAN-SPACE TRAJECTORIES

- Planning a linear Cartesian path(position only)

$$s \in [0,1]$$

setting s = σ/L, σ ∈ [0,L] is the arc length (gives the current length of the path)

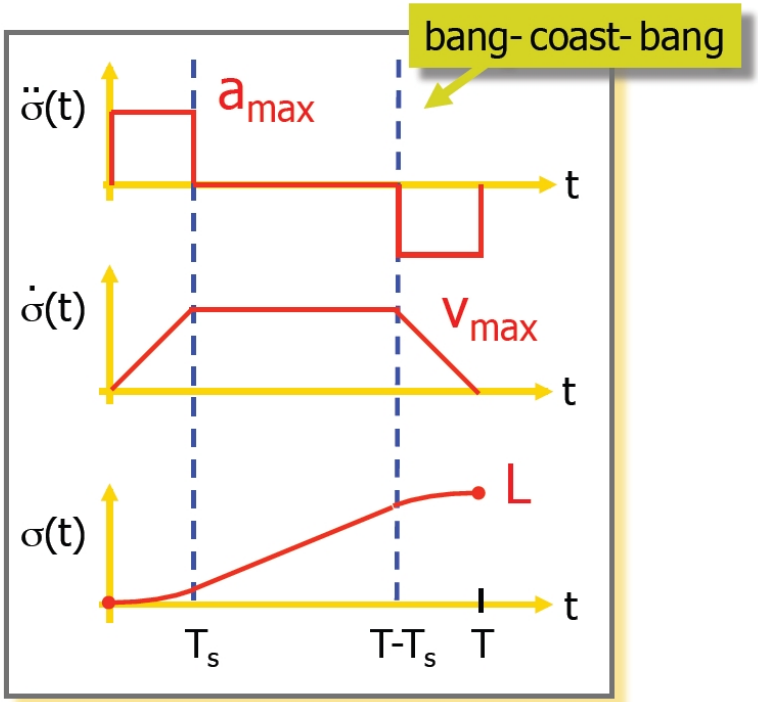
$$p(s) = p_i + s(p_f - p_i)$$

$$\dot{p}(s) = \frac{dp}{ds}\dot{s} = (p_f - p_i)\dot{s} = \frac{p_f - p_i}{L}\dot{\sigma}$$

$$\ddot{p}(s) = \frac{d^2 p}{ds^2}\dot{s}^2 + \frac{dp}{ds}\ddot{s} = (p_f - p_i)\ddot{s} = \frac{p_f - p_i}{L}\ddot{\sigma}$$

5.6 CARTESIAN-SPACE TRAJECTORIES

- Timing law with trapezoidal speed -1



$$given^* : L,\ v_{max},\ a_{max}$$
$$\text{find} : T_s,\ T$$

$$V_{max}(T - T_s) = L \leftarrow$$ = area of the Speed profile

$$T_s = \frac{v_{max}}{a_{max}}$$

$$T = \frac{La_{max} + v_{max}^2}{a_{max} v_{max}} \Leftarrow T - T_s = \frac{L}{v_{max}}$$

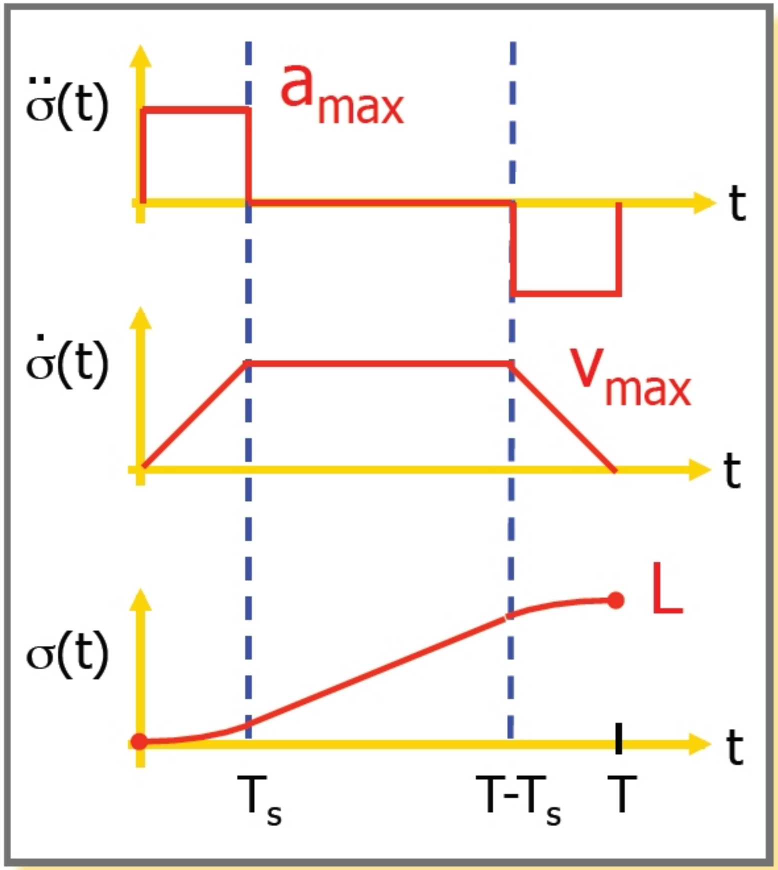
a "coast" phase exists iff: $L > v_{max}^2 / a_{max}$

* = other input data combinations are possible (see textbook)

5.6 CARTESIAN-SPACE TRAJECTORIES

- Timing law with trapezoidal speed -2

$$T_s = \frac{v_{\max}}{a_{\max}}$$

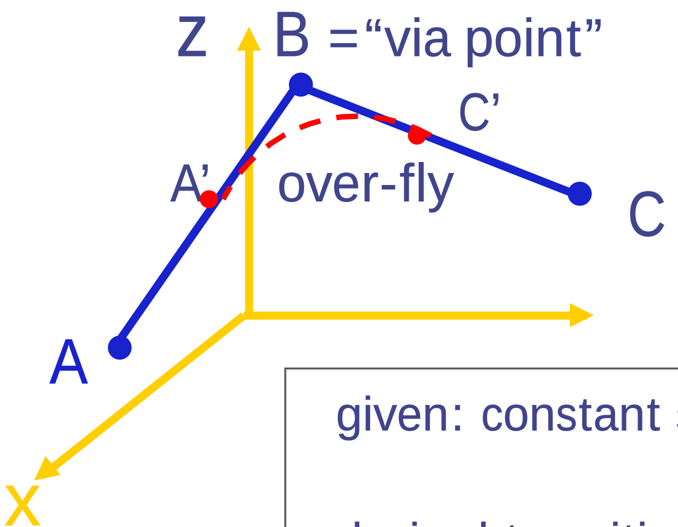


$$\sigma(t) = \begin{cases} a_{\max} t^2 / 2 & t \in [0, T_s] \\ v_{\max} t - \dfrac{v_{\max}^{\;2}}{2a_{\max}} & t \in [T_s, T - T_s] \\ \Leftarrow \int v_{\max} d\tau = v_{\max}\tau + c & \\ c = a_{\max} T_s^{\;2}/2, \quad \tau = t - T_s & \\ -a_{\max}(t-T)^2/2 + v_{\max}T - \dfrac{v_{\max}^{\;2}}{a_{\max}} & \\ & t \in [T\text{-}T_s, T] \end{cases}$$

can be used also in the joint space!

5.6 CARTESIAN-SPACE TRAJECTORIES

- Concatenation of linear paths



$$\frac{B-A}{\|B-A\|} = K_{AB}$$

$$\frac{C-B}{\|C-B\|} = K_{BC}$$

unit vectors of direction cosines

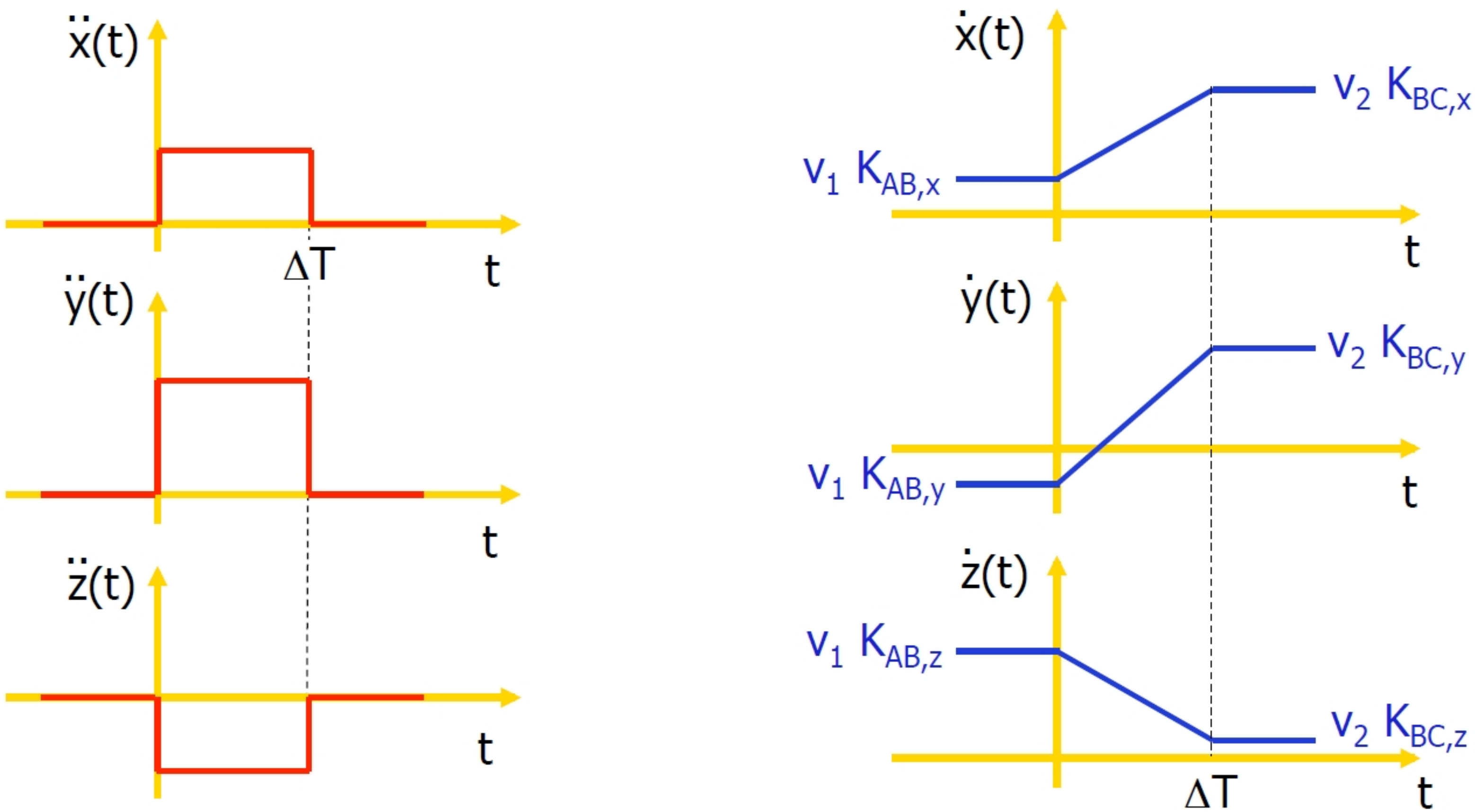
given: constant speeds v_1 on linear path AB
v_2 on linear path BC
desired transition: with constant acceleration for a time ΔT

$$p(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} \qquad t \in [0, \Delta T] (\text{transition starts at t = 0})$$

note: during over-fly, the path remains always in the plane specified by the two lines intersecting at B (in essence, it is a planar problem)

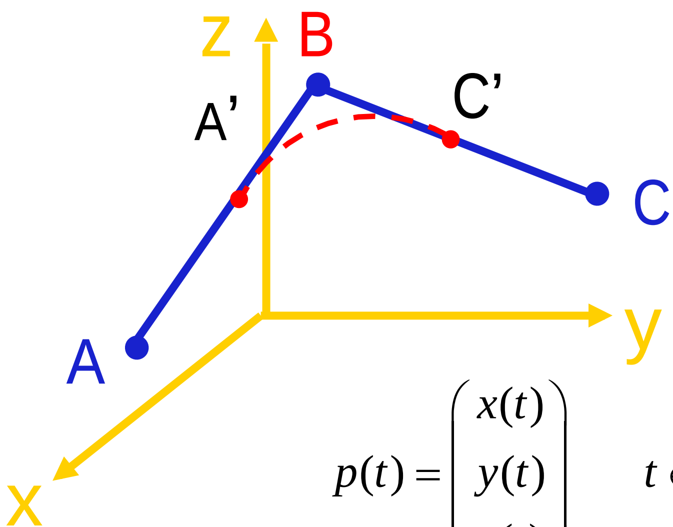
5.6 CARTESIAN-SPACE TRAJECTORIES

- Time profiles on components



5.6 CARTESIAN-SPACE TRAJECTORIES

- Timing law during transition



$$\frac{B-A}{\|B-A\|} = K_{AB}$$

$$\frac{C-B}{\|C-B\|} = K_{BC}$$

unit vectors of direction cosines

$$p(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} \qquad t \in [0, \Delta T] \text{(transition starts at t = 0)}$$

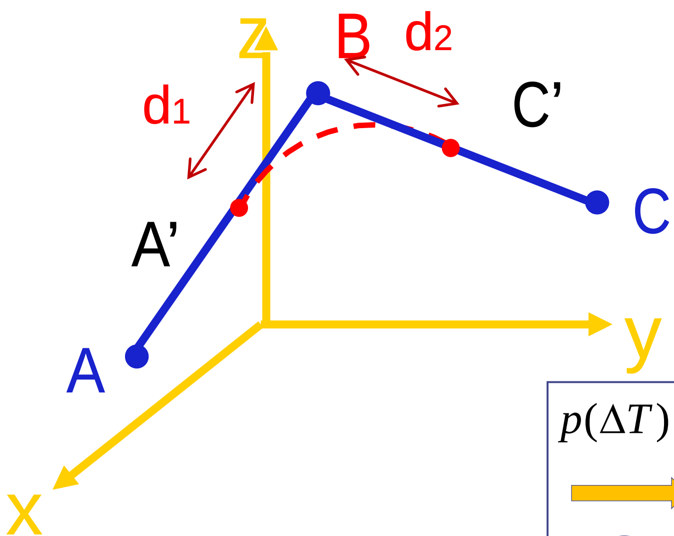
$$\boxed{\ddot{p}(t) = (1/\Delta T)(v_2 K_{BC} - v_1 K_{AB})} \xrightarrow{\int} \boxed{\dot{p}(t) = v_1 K_{AB} + (t/\Delta T)(v_2 K_{BC} - v_1 K_{AB})}$$

$$\xrightarrow{\int} \boxed{p(t) = A' + v_1 K_{AB} t + (t^2 / 2\Delta T)(v_2 K_{BC} - v_1 K_{AB})}$$

thus, we obtain a parabolic blending
(see textbook for this same approach in the joint space)

5.6 CARTESIAN-SPACE TRAJECTORIES

- Solution(various options)



$$\boxed{\begin{aligned} B - A' &= d_1 K_{AB} \\ C' - B &= d_2 K_{BC} \end{aligned}} \quad ①$$

$$\boxed{p(t) = A' + v_1 K_{AB} t + (t^2 / 2\Delta T)(v_2 K_{BC} - v_1 K_{AB})}$$

$$p(\Delta T) = A' + (\Delta T / 2)(v_1 K_{AB} + v_2 K_{BC}) = C'$$

$$\Rightarrow -B + A' + (\Delta T / 2)(v_1 K_{AB} + v_2 K_{BC}) = C' - B$$

$$\xrightarrow{①} d_1 K_{AB} + d_2 K_{BC} = (\Delta T / 2)(v_1 K_{AB} + v_2 K_{BC})$$

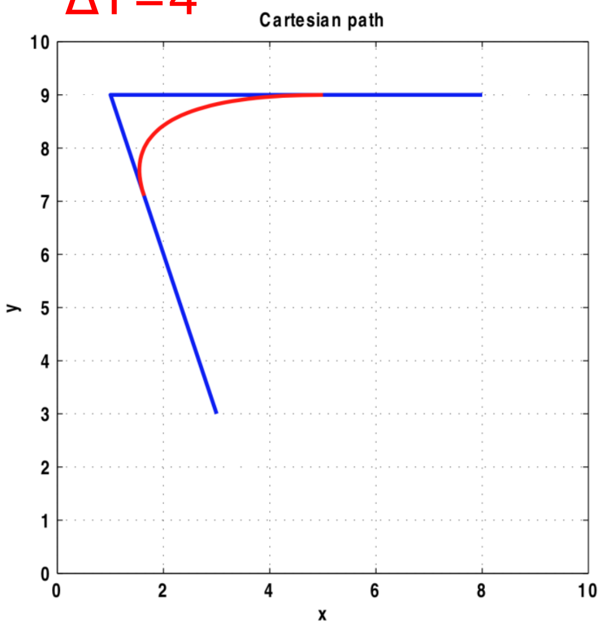
$$\Rightarrow \boxed{d_1 = v_1 \Delta T / 2} \quad \boxed{d_2 = v_2 \Delta T / 2}$$

by choosing, e.g., d_1 (namely A') $\Rightarrow$ $\Delta T = 2 d_1 / v_1 \Rightarrow d_2 = d_1 v_2 / v_1$

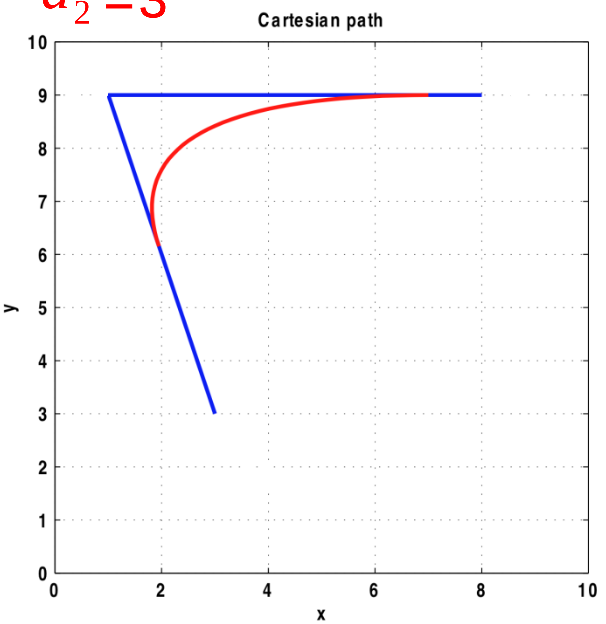
5.6 CARTESIAN-SPACE TRAJECTORIES

- A numerical example
  - transition from A=(3,3) to C=(8,9) via B=(1,9), with speed from v_1 =1 to v_2 =2
  - exploiting two options for solution (resulting in different paths!)
    - assign transition time: ΔT=4 (actually, re-centered for t ∈ [-ΔT/2, ΔT/2])
    - assign distance from B for departing: d_2 =3 (assign d_2 for landing is handled similarly)

ΔT=4

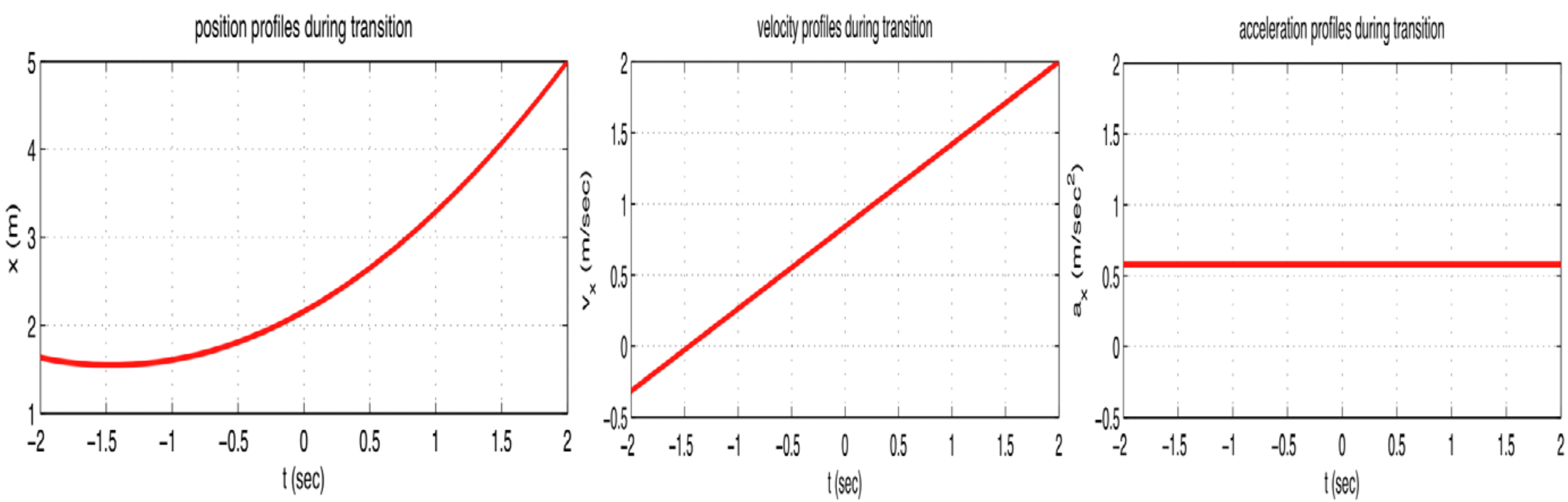


d_2 =3



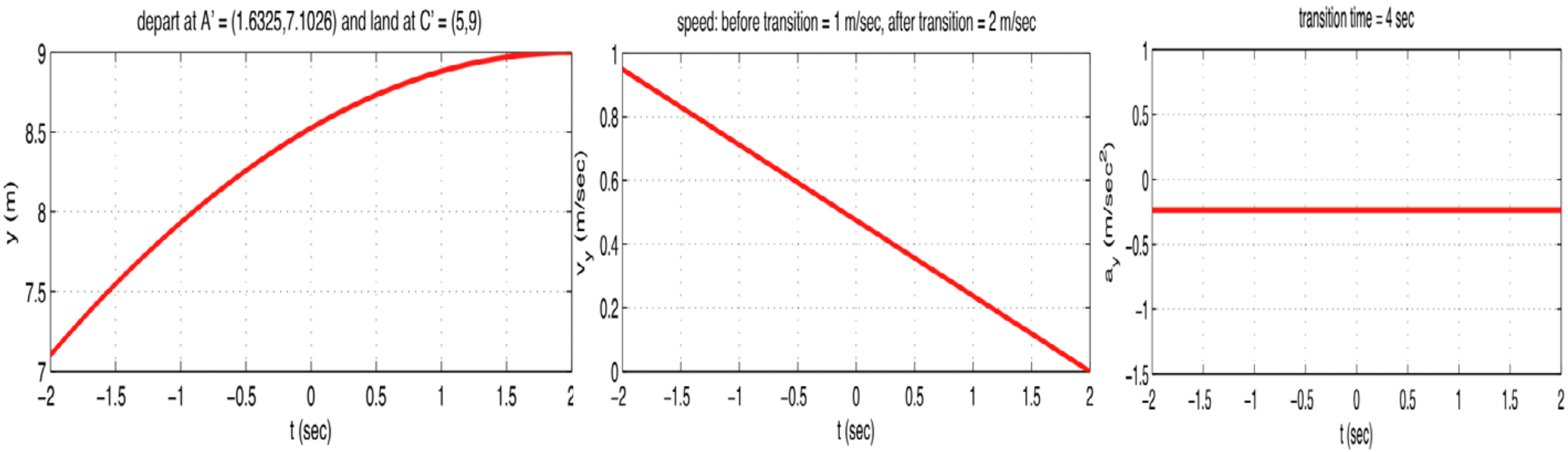
5.6 CARTESIAN-SPACE TRAJECTORIES

- A numerical example(cont'd)
  - first option: $\Delta T=4$ (resulting in $d_1=2$, $d_2=4$)



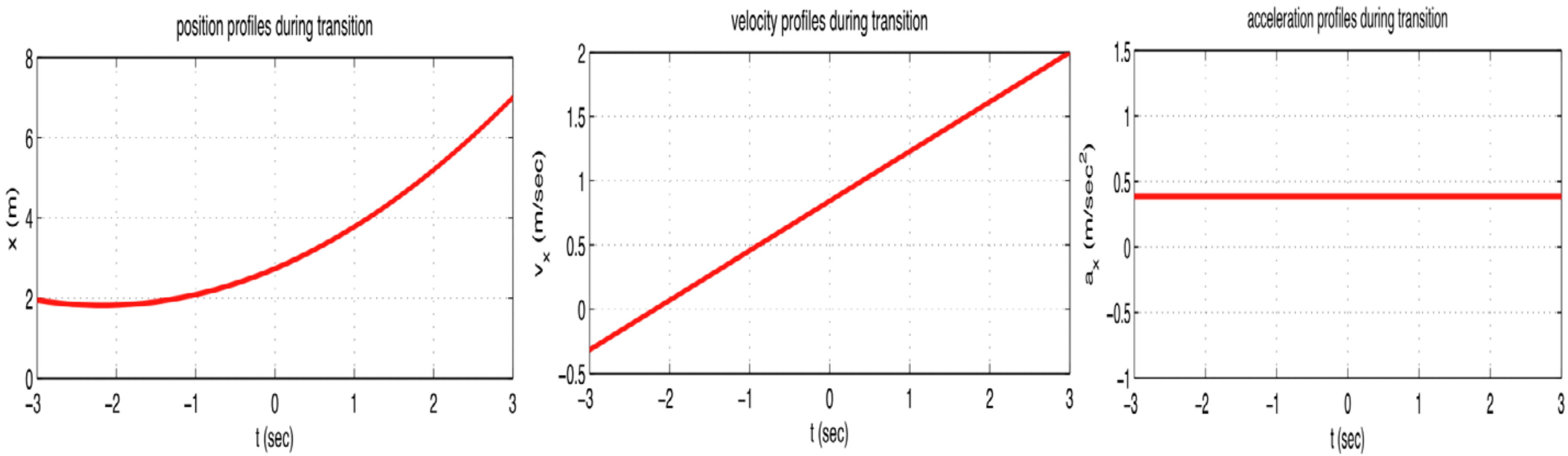
5.6 CARTESIAN-SPACE TRAJECTORIES

- A numerical example(cont'd)
  - first option: ΔT=4 (resulting in $d_1=2$, $d_2=4$)



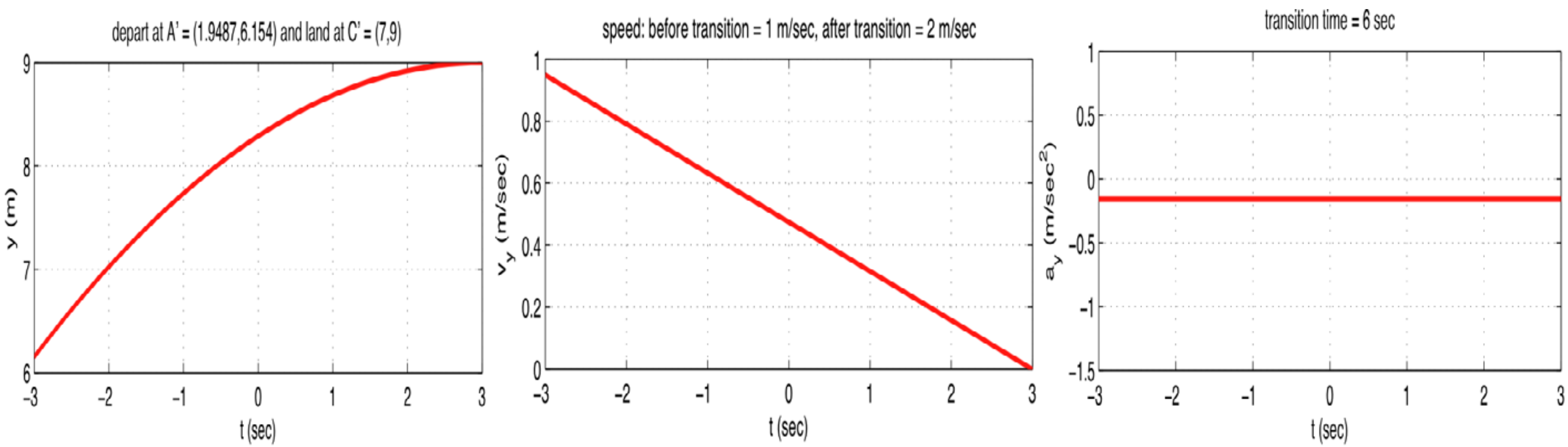
5.6 CARTESIAN-SPACE TRAJECTORIES

- A numerical example(cont'd)
  - second option: d_1 =3 (resulting in ΔT=6, d_2 =6)



5.6 CARTESIAN-SPACE TRAJECTORIES

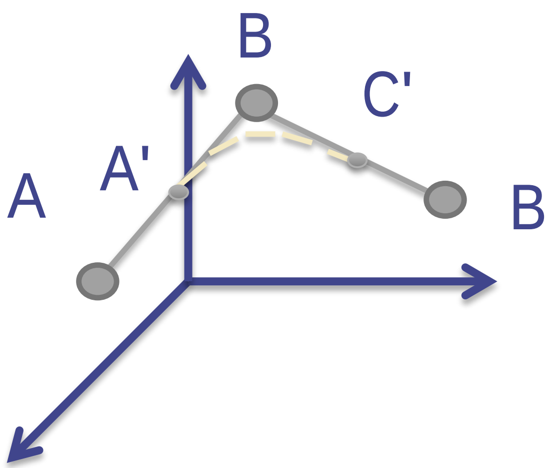
- A numerical example(cont'd)
  - second option: d_1 =3 (resulting in ΔT=6, d_2 =6)



actually, the same vel/acc profiles only with a different time scale

5.6 CARTESIAN-SPACE TRAJECTORIES

- Alternative solution
  - imposing acceleration



$$\ddot{p}(t) = (1/\Delta T)(v_2 K_{BC} - v_1 K_{AB})$$

$$v_1 = v_2 = v_{\max} \,(\,for\ simplicity)$$

$$\left\| \ddot{p}(t) = a_{\max} \right\|$$

$$\Delta T = (v_{\max} / a_{\max}) \left\| K_{BC} - K_{AB} \right\|$$

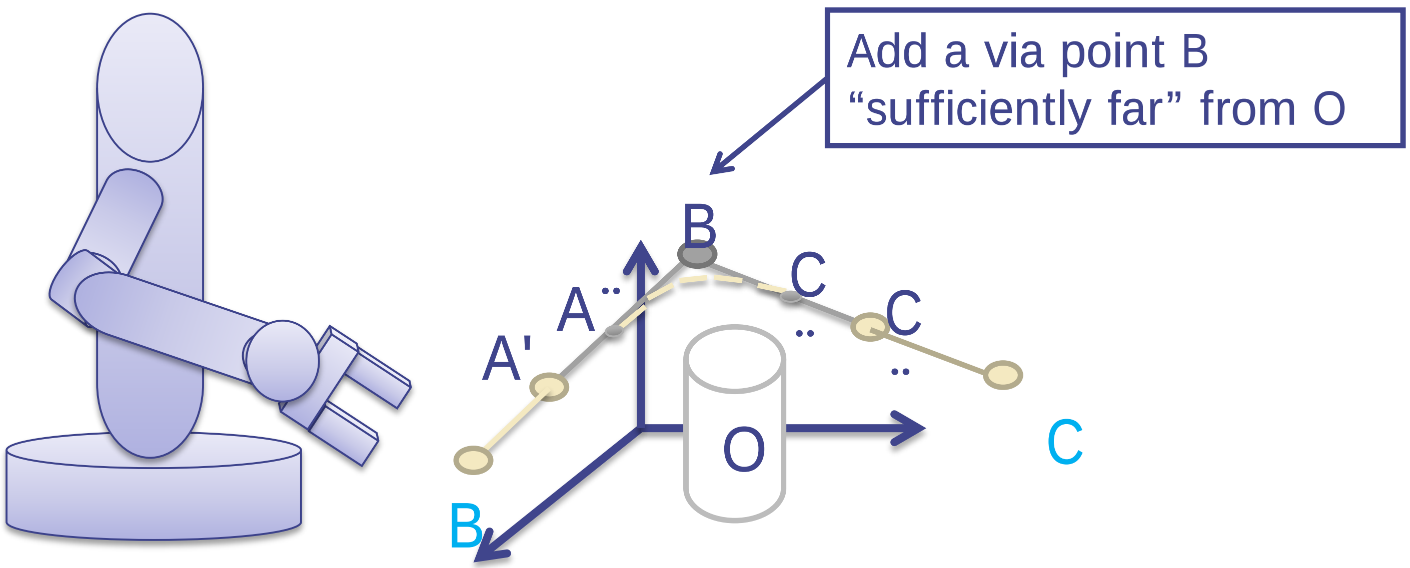
$$= (v_{\max} / a_{\max}) \sqrt{2(1 - K_{BC,x} K_{AB,x} - K_{bc,y} K_{AB,y-} K_{BC,z} K_{AB,z})}$$

$$then,\ d_1 = d_2 = v_{\max} \Delta T / 2$$

5.6 CARTESIAN-SPACE TRAJECTORIES

- Application example

plan a Cartesian trajectory from A to C (rest-to-rest)
that avoids the obstacle O, with $a \le a_{\max}$ *and* $v \le v_{\max}$



$$on\ \overline{AA'} \to a_{\max};\quad on\ \overline{A'B}\ and\ \overline{BC'} \to v_{\max}\quad on\ \overline{C'C} \to -a_{\max'}$$

$$+over - fly between\ A''\ e\ C'';$$

5.6 CARTESIAN-SPACE TRAJECTORIES

- Actual manipulator operates by joint coordinates. Beforehand we have to decide to represent trajectory as Cartesian coordinates or Joint coordinates.

- It can be expressed in the following 3 ways.

5.6 CARTESIAN-SPACE TRAJECTORIES

Method 1

- Within each sampling period, conversion of Cartesian coordinates and joint coordinates
  - To calculate the error between desired path and actual path, current location points are converted to the corresponding points on the Cartesian coordinates by using the Jacobian transformation.
  - Errors of joint space use inverse Jacobian.

- Disadvantages : Calculation is complicated. Therefore, controller's sampling frequency is limited.

5.6 CARTESIAN-SPACE TRAJECTORIES

Method 2

- The trajectory displayed at the joint coordinate system.
  - The constraints on the manipulator's maximum permissible speed and acceleration is considered. These constraints is displayed at the reference coordinate system or the generalized coordination of joint space. The path notation that is placed on another space can represent the relationship, using the kinematic equations.

5.6 CARTESIAN-SPACE TRAJECTORIES

Method 3

- The function, such as the polynomial, is used to represent the path that connect the two adjacent trajectories in the joint space.
  - These function can control manipulator to be flexible and be adjusted to meet the consecutive conditions.

5.7 DESCRIPTION OF PATHS WITH A ROBOT PROGRAMMING LANGUAGE

We will illustrate how various types of paths that we have discussed in this chapter might be specified in a robot language.

In these examples, we use the syntax of **AL**, a robot programming language developed at Stanford University.

5.7 DESCRIPTION OF PATHS WITH A ROBOT PROGRAMMING LANGUAGE

The symbols A, B, C, and D stand for variables of type "frame" in the AL-language examples below. These frames specify path points that we will assume have been taught or textually described to the system. Assume that the manipulator begins in position A.

To move the manipulator in joint-space mode along linear-parabolic-blend paths, we could say

```
move ARM to C with duration = 3*seconds;
```

5.7 DESCRIPTION OF PATHS WITH A ROBOT PROGRAMMING LANGUAGE

To move to the same position and orientation in a straight line we could say

```
move ARM to C linearly with duration = 3*seconds;
```

where the clause "linearly" denotes that Cartesian straight-line motion is to be used. If duration is not important, the user can omit this specification, and the system will use a default velocity- that is,

```
move ARM to C;
```

5.7 DESCRIPTION OF PATHS WITH A ROBOT PROGRAMMING LANGUAGE

A via point can be added, and we can write

move ARM to C via B;

or a whole set of via points might be specified by

move ARM to C via B,A,D;

Note that in

move ARM to C via B with duration = 6*seconds;

the duration is given for the entire motion.

5.7 DESCRIPTION OF PATHS WITH A ROBOT PROGRAMMING LANGUAGE

The system decides how to split this duration between the two segments. It is possible in AL to specify the duration of a single segment- for example, by

move ARM to C via B where duration = 3*seconds;

The first segment which leads to point B will have a duration of 3 seconds.

5.8 PLANNING PATHS WHEN USING THE DYNAMIC MODEL

Usually, when paths are planned, we use a default or a maximum acceleration at each blend point. Actually, the amount of acceleration that the manipulator is capable of at any instant is a function of the dynamics of the arm and the actuator limits.

Most actuators are not characterized by a fixed maximum torque or acceleration, but rather by a torque-speed curve.

5.8 PLANNING PATHS WHEN USING THE DYNAMIC MODEL

When we plan a path assuming there is a maximum acceleration at each joint or along each degree of freedom, we are making a tremendous simplification. In order to be careful not to exceed the actual capabilities of the device, this maximum acceleration must be chosen conservatively.

We might ask the following question: Given a desired spatial path of the end-effector, find the timing information (which turns a description of a spatial path into a trajectory) such that the manipulator reaches the goal point in minimum time. Such problems have been solved by numerical means. The solution takes the rigid body dynamics into account as well as actuator speed-torque constraint curves.