دانشگاه سمنان

دانشکده مهندسی مکانیک

Semnan University
Faculty of Mechanical Engineering

دانشکده مهندسی مکانیک

درس رباتیک پیشرفته

ADVANCED ROBOTICS

Chapter 4 – Trajectory Planning

Class Lecture

❑ ***CONTENTS:***

- Chapter 1: Introduction
- Chapter 2: Kinematics
- Chapter 3: Differential Kinematics and Statics
- → Chapter 4: **Trajectory Planning**
- Chapter 5: Actuators and Sensors
- Chapter 6: Control Architecture
- Chapter 7: Dynamics
- Chapter 8: Motion Control

4. Trajectory Planning

- Trajectory planning:
  - Generate the reference inputs to the motion control system which ensures that the manipulator executes the planned trajectories
  - Generating a time sequence of the values attained by an interpolating function (typically a polynomial) of the desired trajectory
  - Some techniques for trajectory generation:
    - ✓ *Point-to-point motion*
    - ✓ *Motion through a sequence of points*

4.1 Path and Trajectory

- The difference between a path and a trajectory:
  - A path denotes the locus of points in the joint space, or in the operational space, which the manipulator has to follow in the execution of the assigned motion (a pure geometric description of motion).
  - A trajectory is a path on which a timing law is specified, for instance in terms of velocities and/or accelerations at each point.

- The inputs to a trajectory planning algorithm:
  - Path description, Path constraints, Constraints imposed by manipulator dynamics
- The outputs of a trajectory planning algorithm:
  - The end-effector trajectories in terms of a time sequence of the values attained by position, velocity and acceleration

4.2 Joint Space Trajectories

- To plan a trajectory in the joint space, the values of the joint variables have to be determined first from the end-effector position and orientation.

- It is then necessary to resort to an inverse kinematics algorithm, if planning is done off-line, or to directly measure the above variables, if planning is done by the teaching-by-showing technique.

- The planning algorithm generates a function *q(t)* interpolating the given vectors of joint variables at each point, in respect of the imposed constraints.

4.2 Joint Space Trajectories

- Joint space trajectory planning algorithm features:
  - The generated trajectories should be not very demanding from a computational viewpoint
  - The joint positions and velocities should be continuous functions of time (continuity of accelerations may be imposed, too)
  - Undesirable effects should be minimized, e.g. nonsmooth trajectories interpolating a sequence of points on a path.
- At first, only the initial and final points on the path and the traveling time are specified (point-to-point).
- Then intermediate points along the path are specified (motion through a sequence of points).
- Without loss of generality, the single joint variable $q(t)$ is considered.

4.2.1 Point-to-Point Motion

- In point-to-point motion, the manipulator has to move from an initial to a final joint configuration in a given time t_f.

- In this case, the actual end-effector path is of no concern.

- The algorithm should generate a trajectory which is also capable of optimizing some performance index.

4.2.1 Point-to-Point Motion

- The analysis of an incremental motion problem:
  - I : the moment of inertia
  - q_i : initial value
  - q_f : final value
  - t_f : final time
  - τ : torque supplied by motor

- A solution which minimizes the energy dissipated in the motor:

$$I\dot{\omega} = \tau \quad \rightarrow \quad \int_o^{t_f} \omega(t)dt = q_f - q_i$$

- *Performance Index:*

$$\int_0^{t_f} \tau^2(t)dt \quad \rightarrow \quad \omega(t) = at^2 + bt + c$$

4.2.1 Point-to-Point Motion

- The *cubic polynomial*

$$q(t) = a_3t^3 + a_2t^2 + a_1t + a_0$$

$$\longrightarrow \dot{q}(t) = 3a_3t^2 + 2a_2t + a_1$$

$$\longrightarrow \ddot{q}(t) = 6a_3t + 2a_2$$

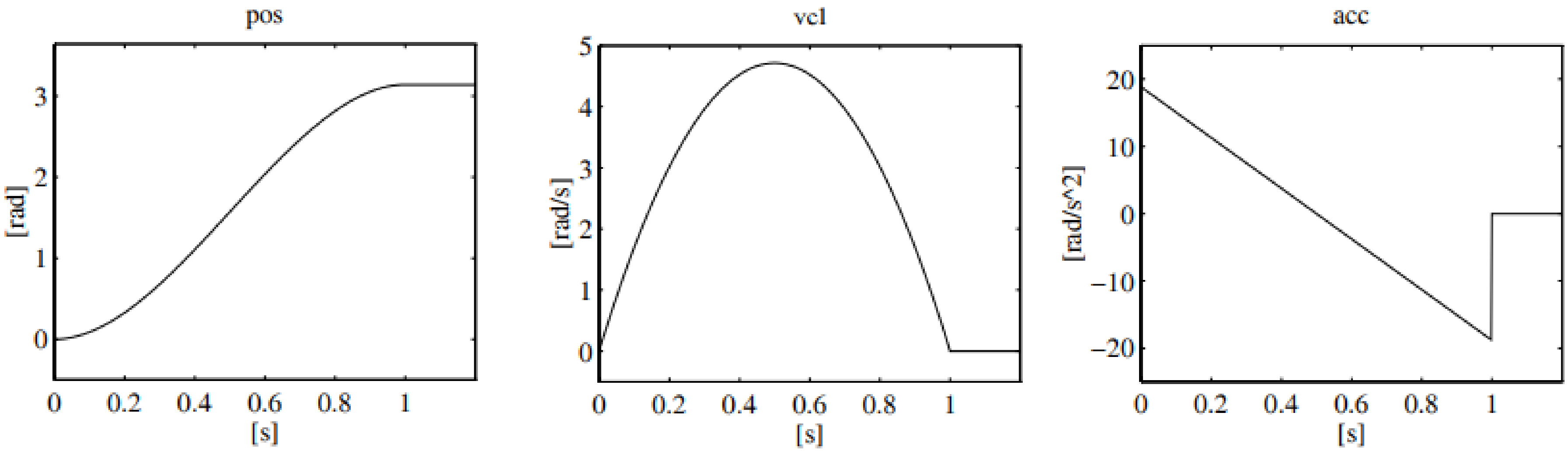
- Determination of a specific trajectory

$$\begin{aligned} a_0 &= q_i \\ a_1 &= \dot{q}_i \\ a_3t_f^3 + a_2t_f^2 + a_1t_f + a_0 &= q_f \\ 3a_3t_f^2 + 2a_2t_f + a_1 &= \dot{q}_f \end{aligned}$$

4.2.1 Point-to-Point Motion

- Time history of position, velocity and acceleration with a cubic polynomial timing law

$$q(t) = a_3t^3 + a_2t^2 + a_1t + a_0$$



4.2.1 Point-to-Point Motion

- If it is desired to assign also the initial and final values of acceleration, six constraints have to be satisfied:

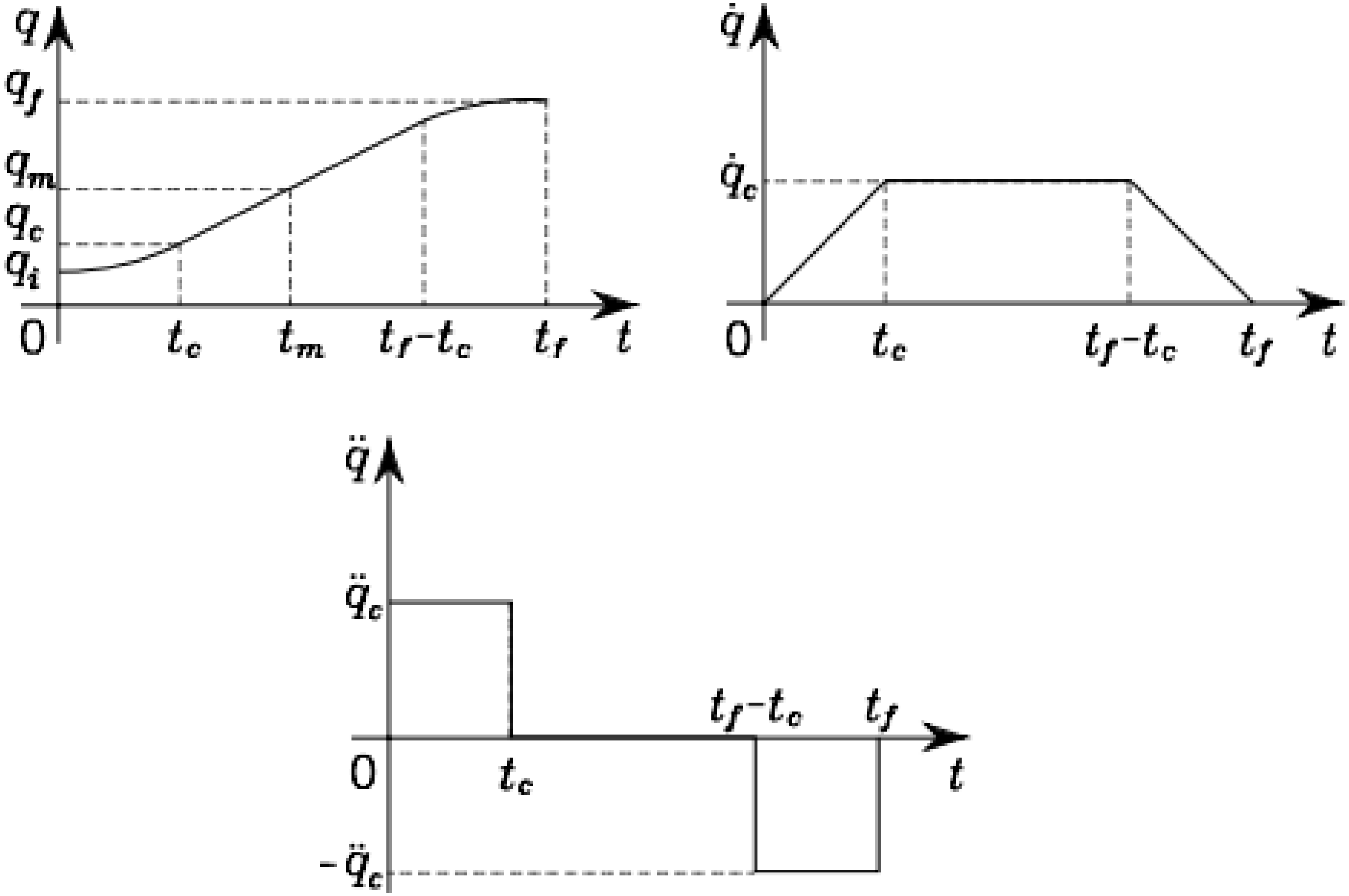
$$q(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

- An alternative approach with timing laws of blended polynomial type is frequently adopted in industrial practice, which allows a direct verification of whether the resulting velocities and accelerations can be supported by the physical mechanical manipulator.

- In this case, a trapezoidal velocity profile is assigned, which imposes a constant acceleration in the start phase, a cruise velocity, and a constant deceleration in the arrival phase.

4.2.1 Point-to-Point Motion

- *Trapezoidal velocity profile:* Characterization of a timing law



4.2.1 Point-to-Point Motion

- The sequence of polynomials

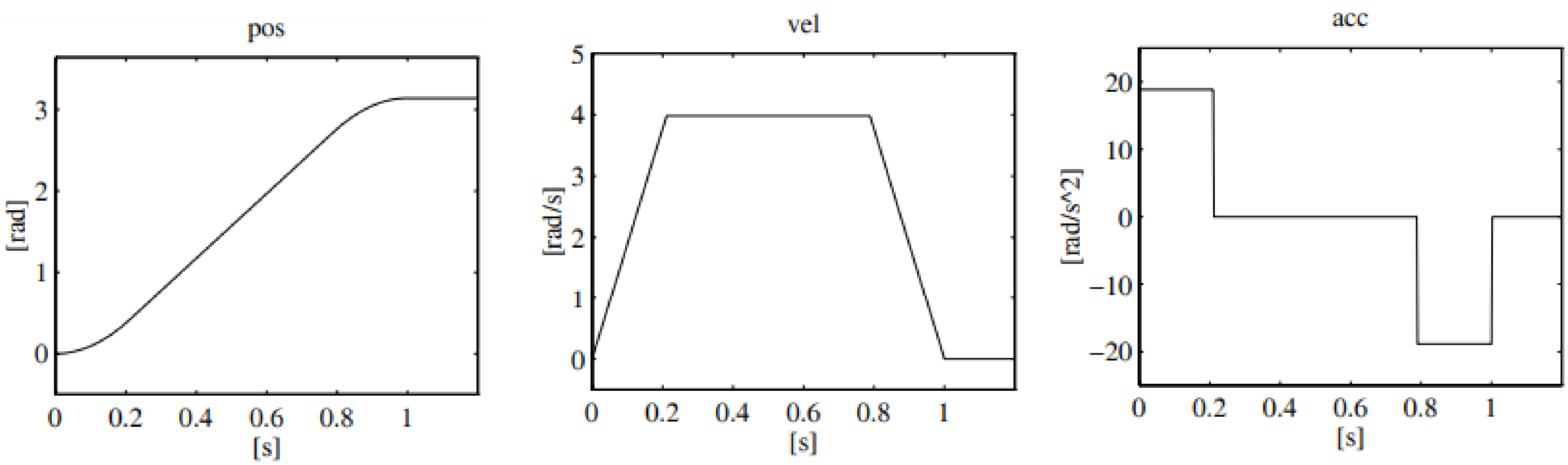
$$\rightarrow \quad t_c = \frac{t_f}{2} - \frac{1}{2}\sqrt{\frac{t_f^2 \ddot{q}_c - 4(q_f - q_i)}{\ddot{q}_c}}$$

$$\rightarrow \quad q(t) = \begin{cases} q_i + \frac{1}{2}\ddot{q}_c t^2 & 0 \le t \le t_c \\ q_i + \ddot{q}_c t_c (t - t_c/2) & t_c < t \le t_f - t_c \\ q_f - \frac{1}{2}\ddot{q}_c (t_f - t)^2 & t_f - t_c < t \le t_f \end{cases}$$

- The adoption of a trapezoidal velocity profile results in a worse performance index compared to the cubic polynomial.

4.2.1 Point-to-Point Motion

- Time history of position, velocity and acceleration with a trapezoidal velocity profile timing law



4.2.2 Motion Through a Sequence of Points

- In several applications, the path is described in terms of a number of points greater than two.

- For instance, even for the simple point-to-point motion of a pick-and-place task, it may be worth assigning two intermediate points between the initial point and the final point.

- For more complex applications, it may be convenient to assign a sequence of points so as to guarantee better monitoring on the executed trajectories.

- Therefore, the problem is to generate a trajectory when N points, termed path points, are specified and have to be reached by the manipulator at certain instants of time.

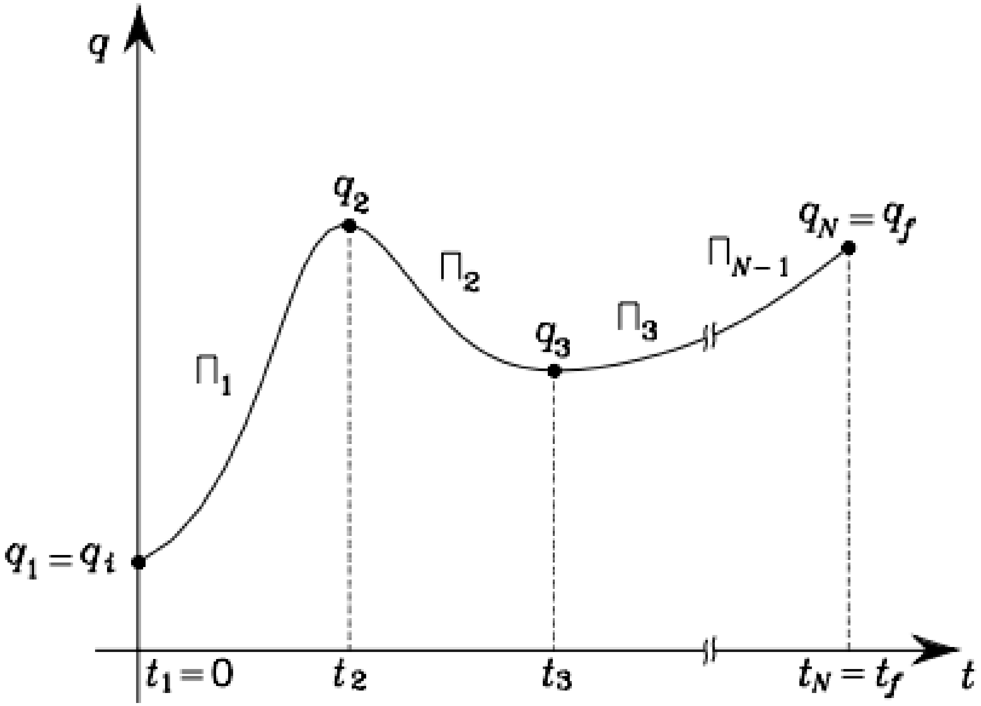
4.2.2 Motion Through a Sequence of Points

- For each joint variable there are N constraints, and then one might want to use an (N −1)-order polynomial. However:
  - It is not possible to assign the initial and final velocities.
  - As the order of a polynomial increases, its oscillatory behaviour increases, and this may lead to trajectories which are not natural for the manipulator.
  - Numerical accuracy for computation of polynomial coefficients decreases as order increases.
  - The resulting system of constraint equations is heavy to solve.
  - Polynomial coefficients depend on all the assigned points; thus, if it is desired to change a point, all of them have to be recomputed.
- These drawbacks can be overcome if a suitable number of low-order interpolating polynomials, continuous at the path points, are considered in place of a single high-order polynomial.

4.2.2 Motion Through a Sequence of Points

- The interpolating polynomial of lowest order is the cubic polynomial, since it allows the imposition of continuity of velocities at the path points

$$\Pi_k(t_k) = q_k$$
$$\Pi_k(t_{k+1}) = q_{k+1}$$
$$\dot{\Pi}_k(t_k) = \dot{q}_k$$
$$\dot{\Pi}_k(t_{k+1}) = \dot{q}_{k+1}$$

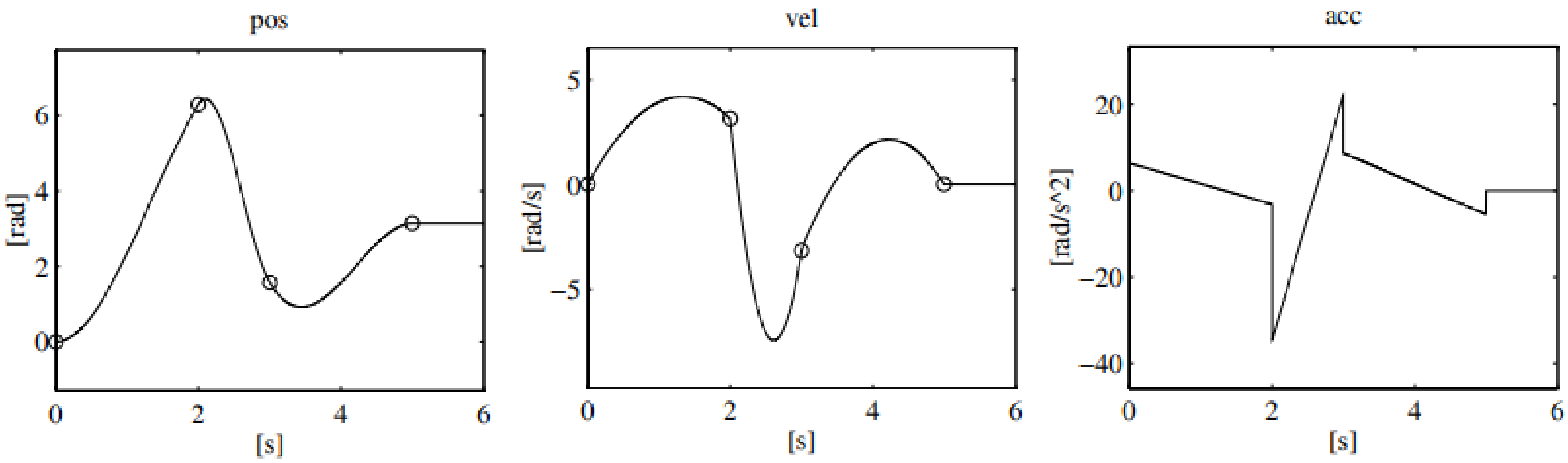


- Continuity of velocity:

$$\dot{\Pi}_k(t_{k+1}) = \dot{\Pi}_{k+1}(t_{k+1})$$

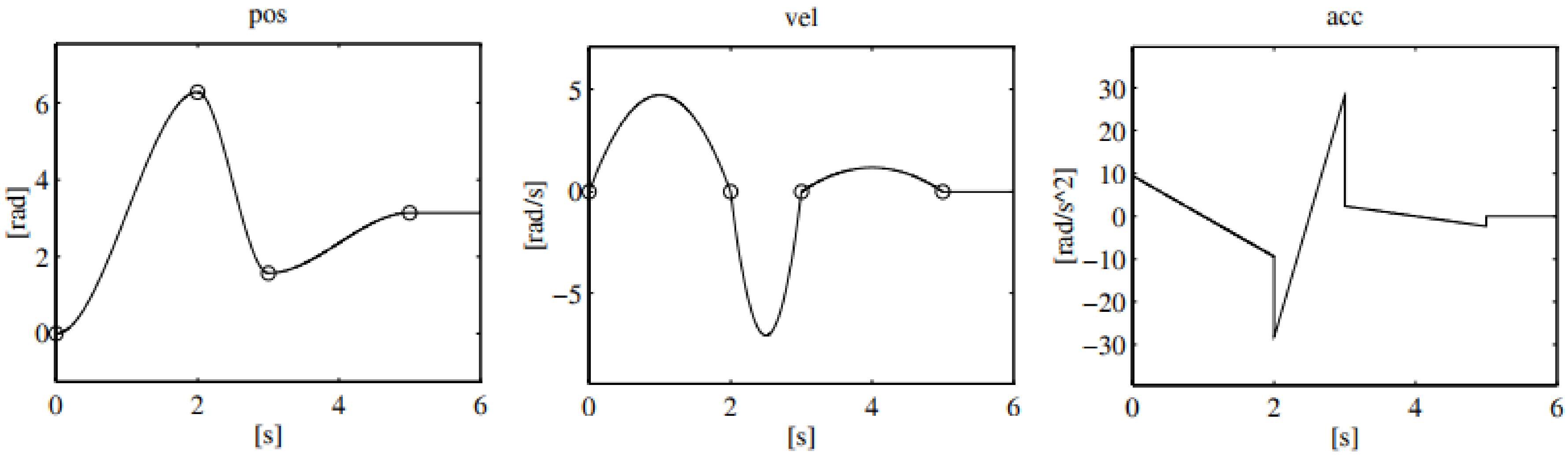
4.2.2 Motion Through a Sequence of Points

- Time history of position, velocity and acceleration with a timing law of interpolating polynomials with velocity constraints at path points



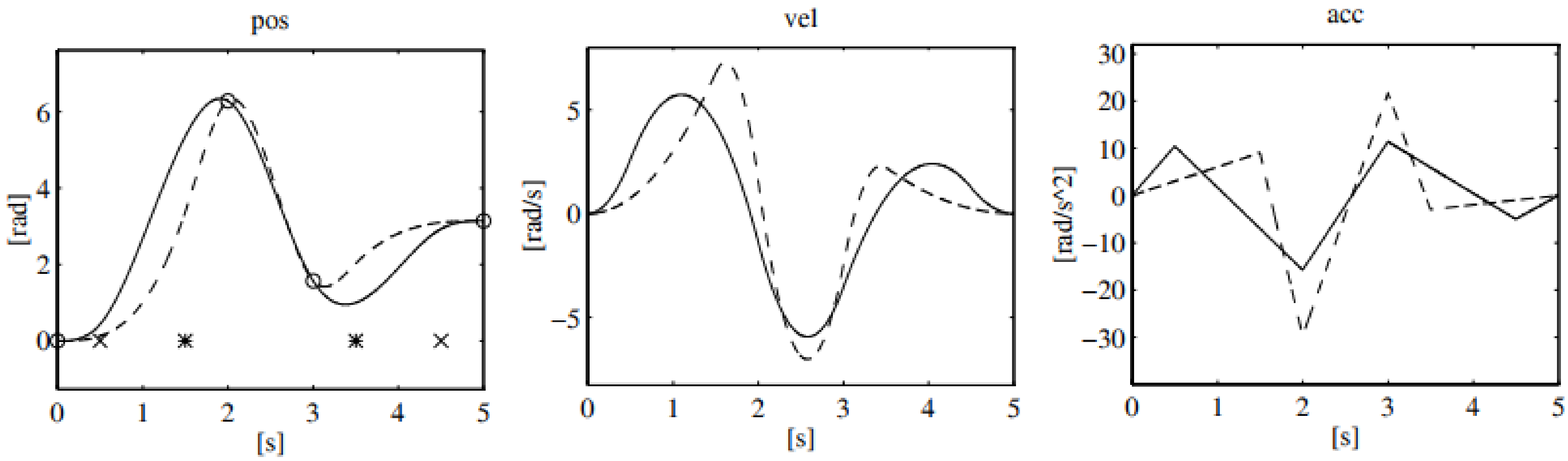
4.2.2 Motion Through a Sequence of Points

- Interpolating polynomials with computed velocities at path points



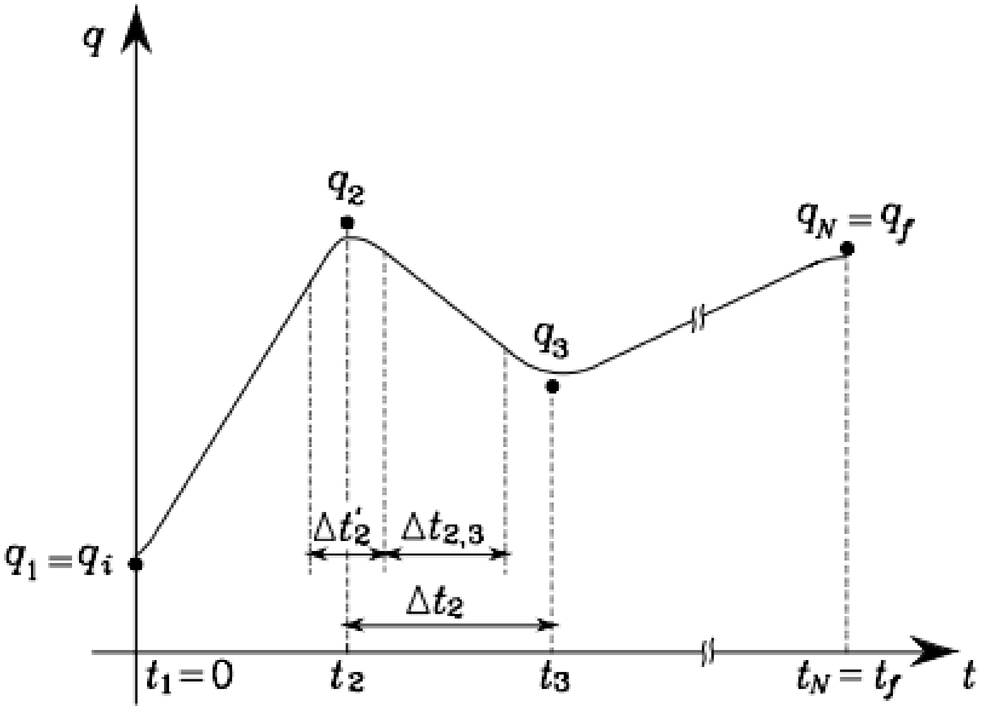
4.2.2 Motion Through a Sequence of Points

- Interpolating polynomials with continuous accelerations at path points (splines)
  - Introducing *virtual points:*
    - ✓ Continuity constraints on position, velocity and acceleration can be imposed, without specifying the actual positions



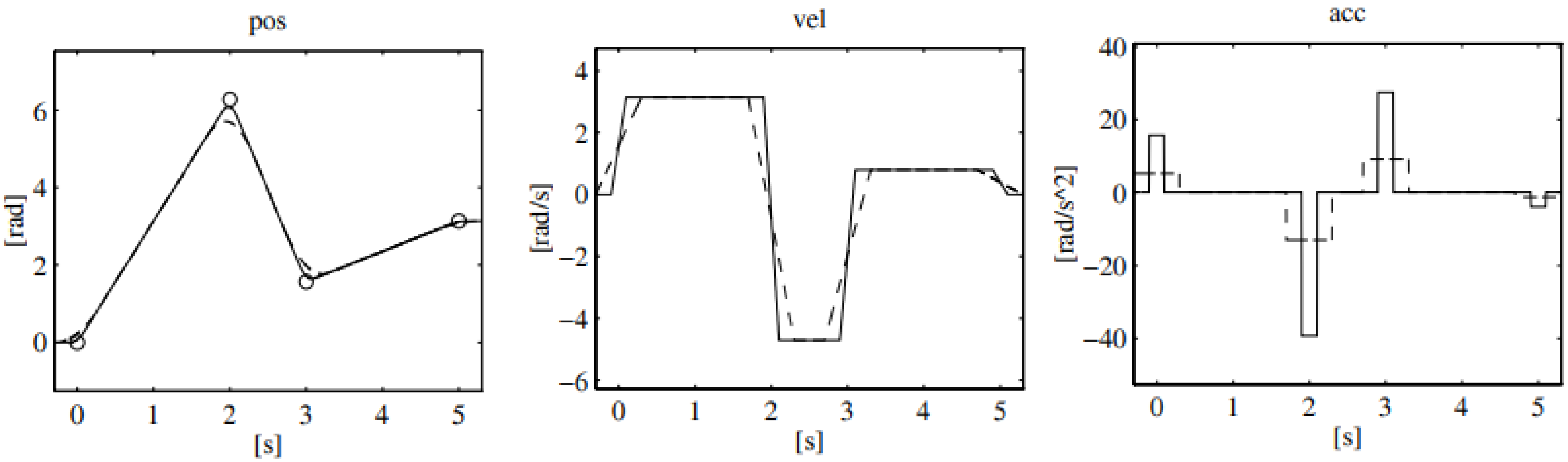
4.2.2 Motion Through a Sequence of Points

- Interpolating linear polynomials with parabolic blends



4.2.2 Motion Through a Sequence of Points

- Interpolating linear polynomials with parabolic blends



4.3 Operational Space Trajectories

- A joint space trajectory planning algorithm generates a time sequence of values for the joint variables *q(t)* so that the manipulator is taken from the initial to the final configuration.

- The resulting end-effector motion is not easily predictable, in view of the nonlinear effects introduced by direct kinematics.

- Whenever it is desired that the end-effector motion follows a geometrically specified path in the operational space, it is necessary to plan trajectory execution directly in the same space.