Week 10 – Trajectory Planning

Advanced Robotic Systems – MANU2453

Dr Ehsan Asadi, School of Engineering
RMIT University, Victoria, Australia
Email: ehsan.asadi@rmit.edu.au

Lectures

Wk	Date	Lecture (NOTE: video recording)	Maths Difficulty	Hands-on Activity	Related Assessment
1	24/7	• Introduction to the Course • Spatial Descriptions & Transformations			
2	31/7	• Spatial Descriptions & Transformations • Robot Cell Design			Robot Cell Design Assignment
3	7/8	• Forward Kinematics • Inverse Kinematics			
4	14/8	• ABB Robot Programming via Teaching Pendant • ABB RobotStudio Offline Programming		ABB RobotStudio Offline Programming	Offline Programming Assignment
5	21/8	• Jacobians: Velocities and Static Forces			
6	28/8	• Manipulator Dynamics			
7	11/9	• Manipulator Dynamics		MATLAB Simulink Simulation	
8	18/9	• Robotic Vision		MATLAB Simulation	Robotic Vision Assignment
9	25/9	• Robotic Vision II		MATLAB Simulation	
10	2/10	• Trajectory Generation			
11	9/10	• Linear & Nonlinear Control		MATLAB Simulink Simulation	
12	16/10	• Introduction to I4.0 • Revision			Final Exam

Content

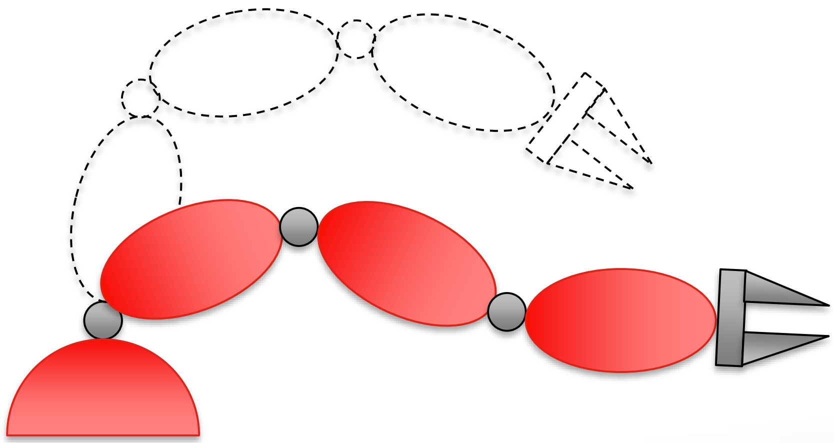
- Introduction
- Cartesian Space Schemes vs. Joint Space Schemes
- Cubic Polynomial
- Quintic Polynomial
- Linear Function with Parabolic Blends
- Matlab / Simulink Simulation

Content

- Introduction
- Cartesian Space Schemes vs. Joint Space Schemes
- Cubic Polynomial
- Quintic Polynomial
- Linear Function with Parabolic Blends
- Matlab / Simulink Simulation

Introduction

- In the past two weeks, you learnt how to find the target using vision. Today you will learn how to specify the path from current position to the target.
  - And next week, you will learn how to control the robot to follow this path.



- Here, we use the term "trajectory" in place of path.
  - Trajectory means a time history of position, velocity and acceleration for each degree of freedom.
  - For e.g. we may specify: at time 0, robot is at x=1,y=1; at time 0.1, robot should go to x=1.2, y=0.8; at time 0.2, x=1.3, y=0.7 etc.

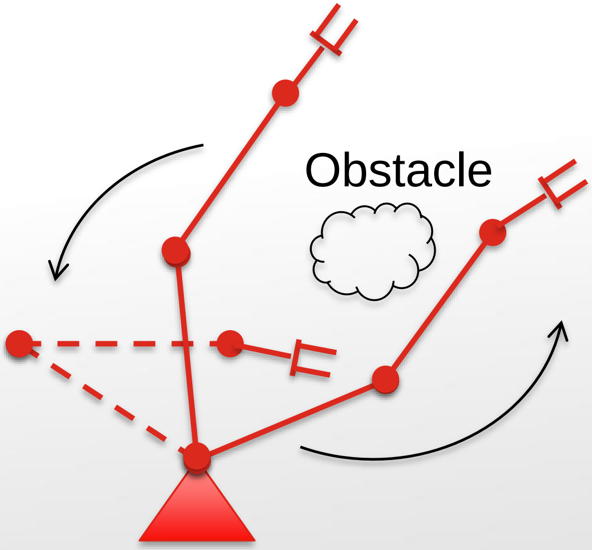
Introduction

- However, the above example of specifying point by point at different times is not convenient.
- Can we just specify the:
  - Desired goal position and orientation for the end-effector;
  - The time to reach goal position;
  - General shape of the path (straight line, polynomial etc.);
  - (Optional) Some intermediate points / "via points"
- and let the system figure out the trajectory?

- NOTE: The term "points" in this lecture should be interpreted as frames containing both position and orientation!

Introduction

- It is often desirable that the motion of the manipulator to be smooth.
  - Continuous function, with continuous first derivative.
  - It's even better if second derivative is continuous.
  - Reason: Rough and jerky motions causes vibrations due to resonance modes, as well as increases wear and tear.
- "Via points" are usually given for the purpose of collision avoidance.

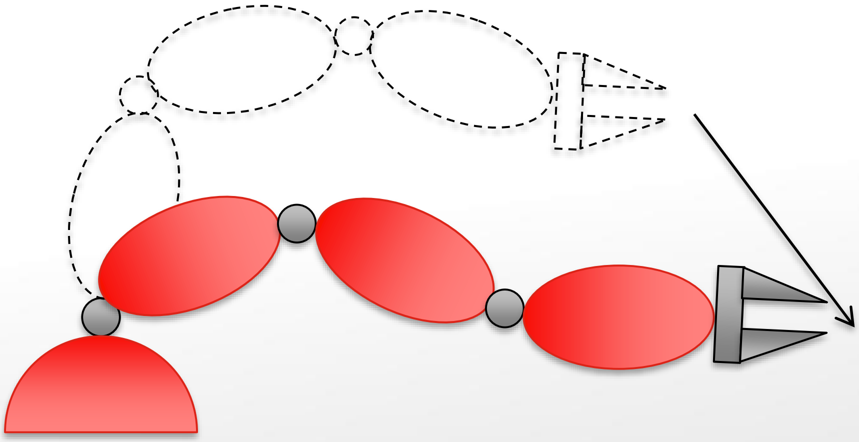


Content

- Introduction
- Cartesian Space Schemes vs. Joint Space Schemes
- Cubic Polynomial
- Quintic Polynomial
- Linear Function with Parabolic Blends
- Matlab / Simulink Simulation

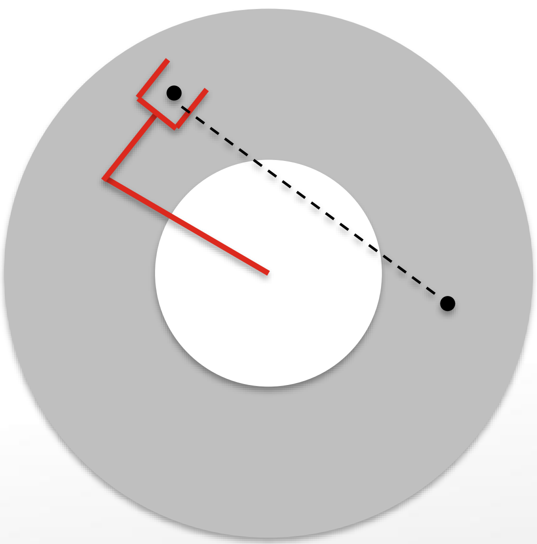
Cartesian Space Schemes

- Cartesian Space Schemes means specifying trajectory directly through position and orientation of the end-effector.
  - The advantage of Cartesian Space Schemes is that we can enforce certain shape of the trajectory (for e.g. straight line), or enforce orientation of the end-effector (for e.g. maintain same orientation throughout).

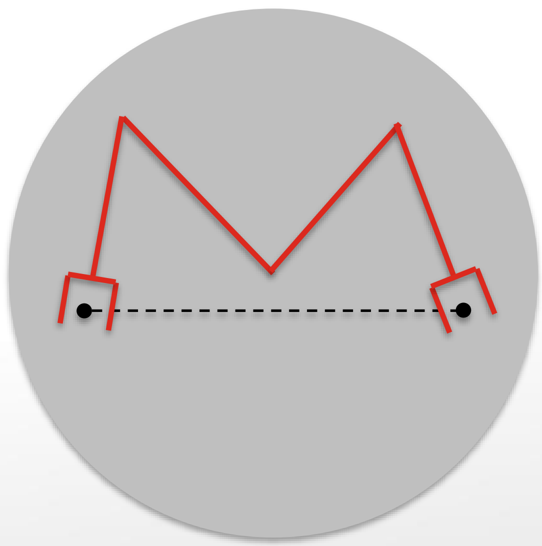


Cartesian Space Schemes

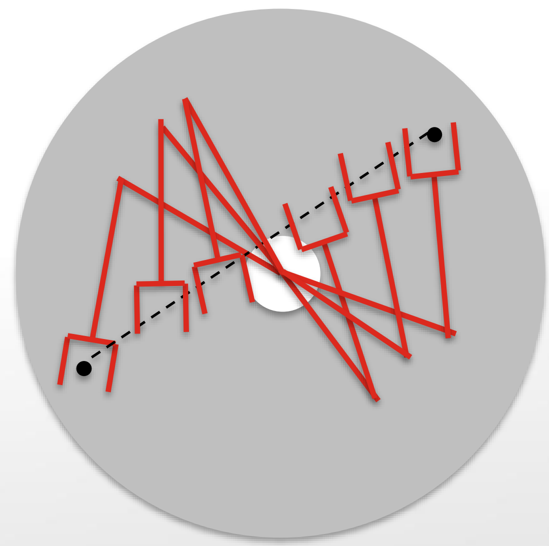
- However, Cartesian Space Schemes also have quite a few disadvantages:
  - Computationally expensive: After path is generated, inverse kinematics has to be solved at every time step (update rate) to calculate joint angles.
  - Prone to problems relating to workspace and singularities



Intermediate points not reachable



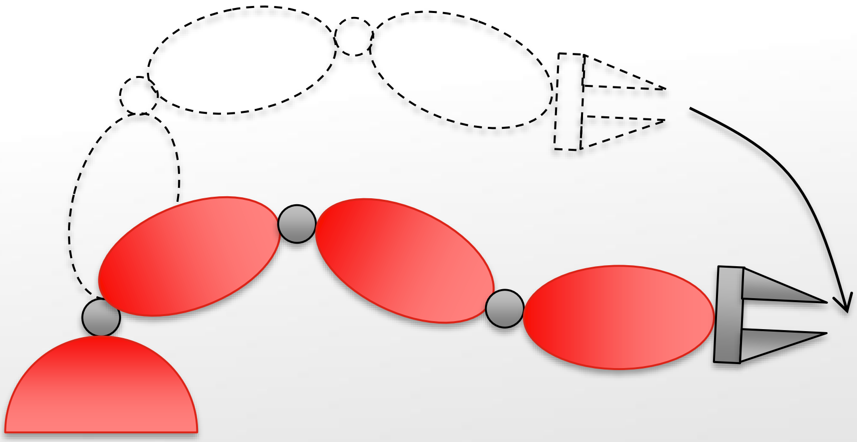
Start and end points reachable but in different configurations



High joint rates near Singularities

Joint Space Schemes

- Joint Space Schemes means specifying trajectory directly through the joint angles.
  - Advantages:
    - Easy to compute.
    - No issue with singularities.
  - Disadvantage:
    - Path will not be linear.
    - This may be a problem if there are possible collisions.

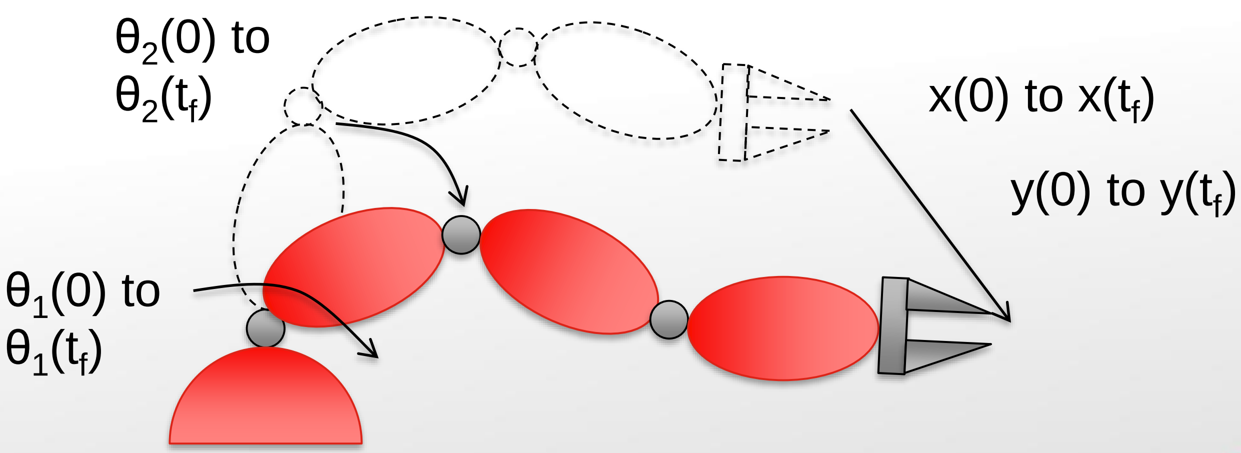


Content

- Introduction
- Cartesian Space Schemes vs. Joint Space Schemes
- Cubic Polynomial
- Quintic Polynomial
- Linear Function with Parabolic Blends
- Matlab / Simulink Simulation

General Solution

- In this section, we will use "u(t)" to represent any of the variables, be it the Cartesian terms (x, y, z, angles) or the joint variables (θ).
- A reminder of our question:
  - Given the start, end, and possibly some via points;
  - And given the time to reach goal position;
  - Generate a trajectory u(t) for the robot to follow.



Use "u(t) to represent any of the variables. Therefore: $u(0)$ to $u(t_f)$

Straight Line

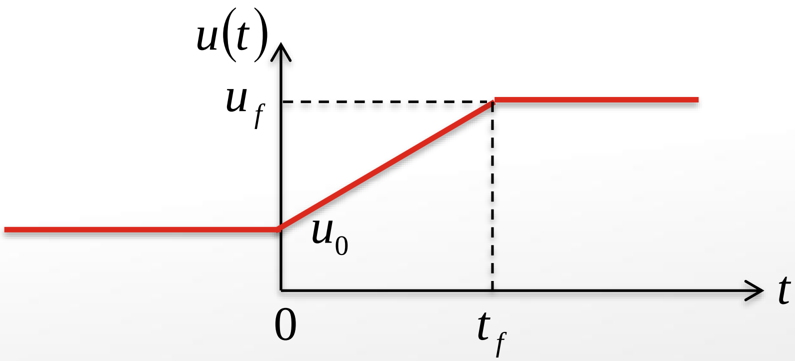
- We want to move from initial position to target position within time t_f.
- To do this, the general variable "u(t)" will change from its initial value to the target value within time t_f.

Straight Line

- We want to move from initial position to target position within time t_f.
- To do this, the general variable "u(t)" will change from its initial value to the target value within time t_f.

$$u(0) = u_0$$
$$u(t_f) = u_f$$

- The simplest path would be a straight line between the two values.



- Disadvantage: Discontinuous velocities at start and end points.

Cubic Polynomial

- To ensure that the velocities at the start and end points are zero, we can use a cubic polynomial:

- There are four parameters which can satisfy four constraints:

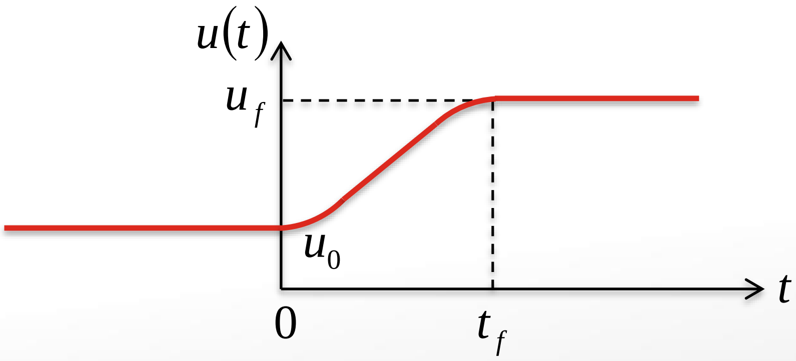
Cubic Polynomial

- To ensure that the velocities at the start and end points are zero, we can use a cubic polynomial:

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

- There are four parameters which can satisfy four constraints:

$$\begin{aligned} u(0) &= u_0 \\ u(t_f) &= u_f \\ \dot{u}(0) &= 0 \\ \dot{u}(t_f) &= 0 \end{aligned}$$



- The task is then to calculate the parameters a_0, a_1, a_2 and a_3.

Cubic Polynomial

- This can be done by solving the following simultaneous equations:

$$u(0) = a_0 + a_1 0 + a_2 0^2 + a_3 0^3 = u_0$$

$$\dot{u}(0) = a_1 + 2a_2 0 + 3a_3 0^2 = 0$$

$$u(t_f) = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 = u_f$$

$$\dot{u}(t_f) = a_1 + 2a_2 t_f + 3a_3 t_f^2 = 0$$

- And the solution is:

$$a_0 = u_0 \qquad a_1 = 0 \qquad a_2 = \frac{3}{t_f^2}\left(u_f - u_0\right) \qquad a_3 = -\frac{2}{t_f^3}\left(u_f - u_0\right)$$

Cubic Polynomial

- Example:

$$t_f = 3\sec$$
$$u(0) = u_0 = 15\deg$$
$$u(t_f) = u_f = 75\deg$$
$$\dot{u}(0) = 0$$
$$\dot{u}(t_f) = 0$$

- The solution is:

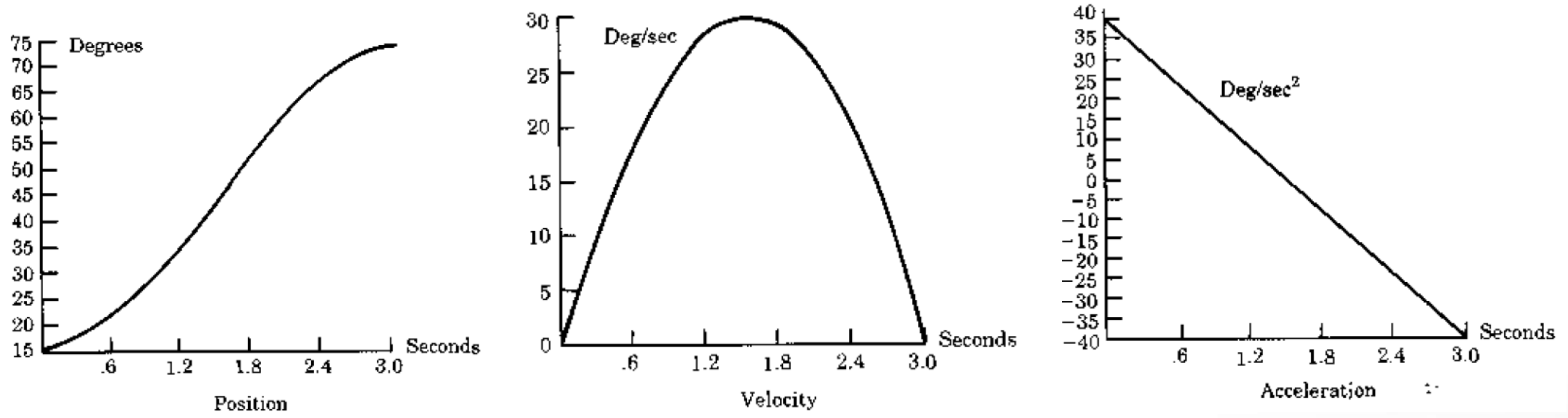
$$a_0 = u_0 = 15 \qquad a_1 = 0 \qquad a_2 = \frac{3}{t_f^2}(u_f - u_0) = 20 \qquad a_3 = -\frac{2}{t_f^3}(u_f - u_0) = -4.44$$

- Thus the trajectory is:

$$u(t) = 15 + 20t^2 - 4.44t^3$$

Cubic Polynomial

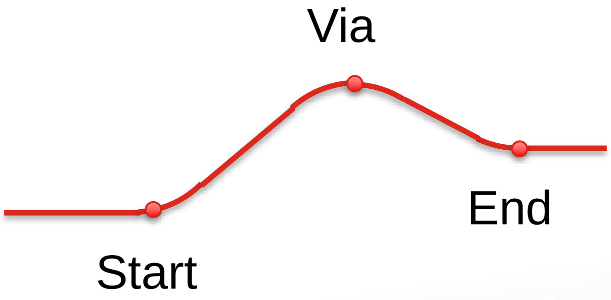
- The plots for position, velocity and acceleration are:

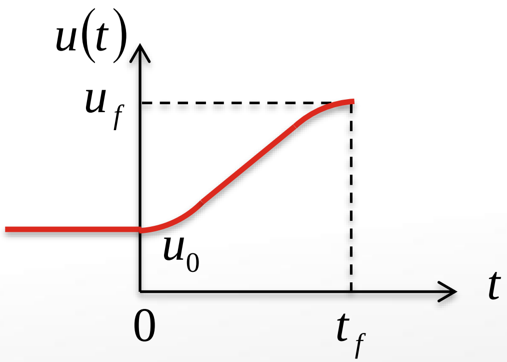


- Note that the accelerations at start and end positions are not zero.
  - This might create jerky motions.

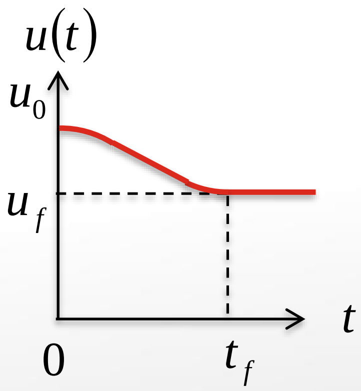
Cubic Polynomial – Via Points

- We have looked at cubic polynomial connecting start and end point.
- What if we need the path to pass through a via point?
- Simple! Just split the path into segments (start to via point, via point to end) and derive cubic polynomial for each of them.





Segment 1



Segment 2

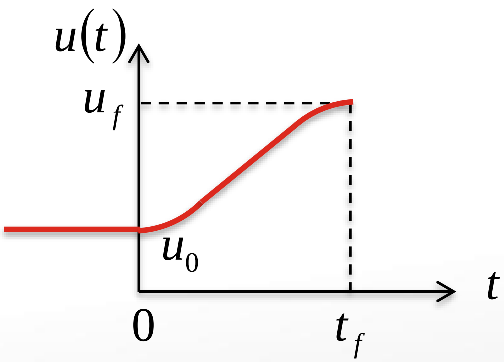
- However, the velocity at via point need not be zero:

Cubic Polynomial – Via Points

- For each segment, we will solve for a_0, a_1, a_2 and a_3 for the cubic polynomial

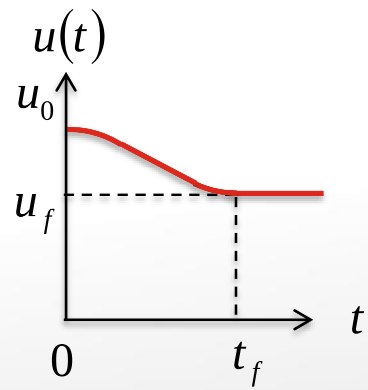
$$u(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

- Of course, the constraints will be different:



$$\begin{aligned} u(0) &= u_0 \\ u(t_f) &= u_{\text{via}} \\ \dot{u}(0) &= 0 \\ \boxed{\dot{u}(t_f) = \dot{u}_{\text{via}}} \end{aligned}$$

Segment 1

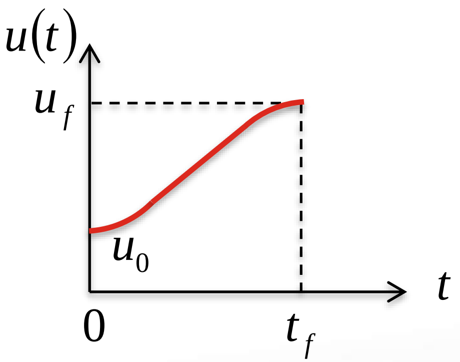


$$\begin{aligned} u(0) &= u_{\text{via}} \\ u(t_f) &= u_f \\ \boxed{\dot{u}(0) = \dot{u}_{\text{via}}} \\ \dot{u}(t_f) &= 0 \end{aligned}$$

Segment 2

Cubic Polynomial – Via Points

- For an even more general case where we have more than 1 via point, we will do the same: Just split the path into many segments and solve the simultaneous solutions for each of the segments.
- The start and end velocities of the i^{th} segment need not be zero. Therefore:



Segment i

$$
\begin{aligned}
u(0) &= u_{\text{via}(i-1)} \\
u(t_f) &= u_{\text{via}(i)} \\
\dot{u}(0) &= \dot{u}_{\text{via}(i-1)} \\
\dot{u}(t_f) &= \dot{u}_{\text{via}(i)}
\end{aligned}
$$

Cubic Polynomial

- The simultaneous equations to be solved are thus:

$$u(0) = a_0 + a_1 0 + a_2 0^2 + a_3 0^3 = u_{\text{via}(i-1)}$$

$$u(t_f) = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 = u_{\text{via}(i)}$$

$$\left.\begin{aligned} \dot{u}(0) &= a_1 + 2a_2 0 + 3a_3 0^2 = \dot{u}_{\text{via}(i-1)} \\ \dot{u}(t_f) &= a_1 + 2a_2 t_f + 3a_3 t_f^2 = \dot{u}_{\text{via}(i)} \end{aligned}\right\}$$

By differentiation of

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

- And the solution is:

$$a_0 = u_{\text{via}(i-1)} \qquad a_1 = \dot{u}_{\text{via}(i-1)} \qquad a_2 = \frac{3}{t_f^2}\left(u_{\text{via}(i)} - u_{\text{via}(i-1)}\right) - \frac{2}{t_f}\dot{u}_{\text{via}(i-1)} - \frac{1}{t_f}\dot{u}_{\text{via}(i)}$$

$$a_3 = -\frac{2}{t_f^3}\left(u_{\text{via}(i)} - u_{\text{via}(i-1)}\right) + \frac{1}{t_f^2}\left(\dot{u}_{\text{via}(i)} - \dot{u}_{\text{via}(i-1)}\right)$$

Content

- Introduction
- Cartesian Space Schemes vs. Joint Space Schemes
- Cubic Polynomial
- Quintic Polynomial
- Linear Function with Parabolic Blends
- Matlab / Simulink Simulation

Quintic Polynomial

- Using the cubic polynomial, we can only specify 4 constraints, because the cubic polynomial only has 4 parameters.
  - Start and end positions
  - Start and end velocities
- We had no control over the accelerations.
  - From the numerical example, we saw that the acceleration started and ended at 40 and -40 respectively.
- If we want to be able to specify the start and end accelerations as well (i.e. now altogether 6 constraints), then we will need to use a polynomial with 6 parameters – The Quintic Polynomial.

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

Quintic Polynomial

- The constraints are:

$$u(0) = a_0 + a_1 0 + a_2 0^2 + a_3 0^3 + a_4 0^4 + a_5 0^5 = a_0$$

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

$$\dot{u}(0) = a_1 + 2a_2 0 + 3a_3 0^2 + 4a_4 0^3 + 5a_5 0^4 = a_1$$

$$\dot{u}(t_f) = a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4$$

By differentiation of

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

$$\ddot{u}(0) = 2a_2 + 6a_3 0 + 12a_4 0^2 + 20a_5 0^3 = 2a_2$$

$$\ddot{u}(t_f) = 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3$$

By differentiation of

$$\dot{u}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4$$

Quintic Polynomial

- Solving the simultaneous equations, the parameters are:

$$a_0 = u_0 \qquad a_1 = \dot{u}_0 \qquad a_2 = \frac{\ddot{u}_0}{2}$$

$$a_3 = \frac{20u_f - 20u_0 - (8\dot{u}_f + 12\dot{u}_0)t_f - (3\ddot{u}_0 - \ddot{u}_f)t_f^2}{2t_f^3}$$

$$a_4 = \frac{30u_0 - 30u_f + (14\dot{u}_f + 16\dot{u}_0)t_f + (3\ddot{u}_0 - 2\ddot{u}_f)t_f^2}{2t_f^4}$$

$$a_5 = \frac{12u_f - 12u_0 - (6\dot{u}_f + 6\dot{u}_0)t_f - (\ddot{u}_0 - \ddot{u}_f)t_f^2}{2t_f^5}$$

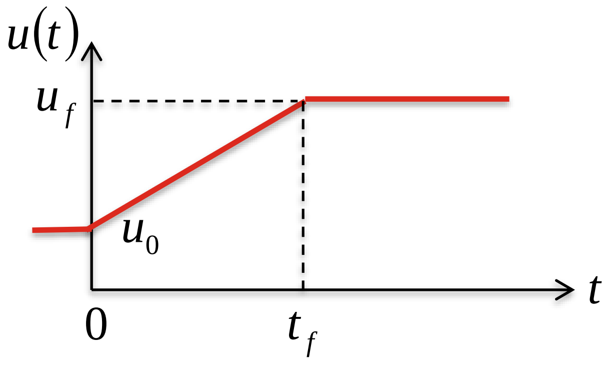
Note!

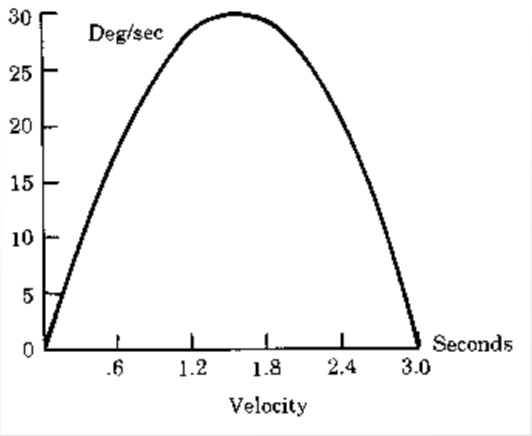
Content

- Introduction
- Cartesian Space Schemes vs. Joint Space Schemes
- Cubic Polynomial
- Quintic Polynomial
- Linear Function with Parabolic Blends
- Matlab / Simulink Simulation

Linear Function w. Parabolic Blends

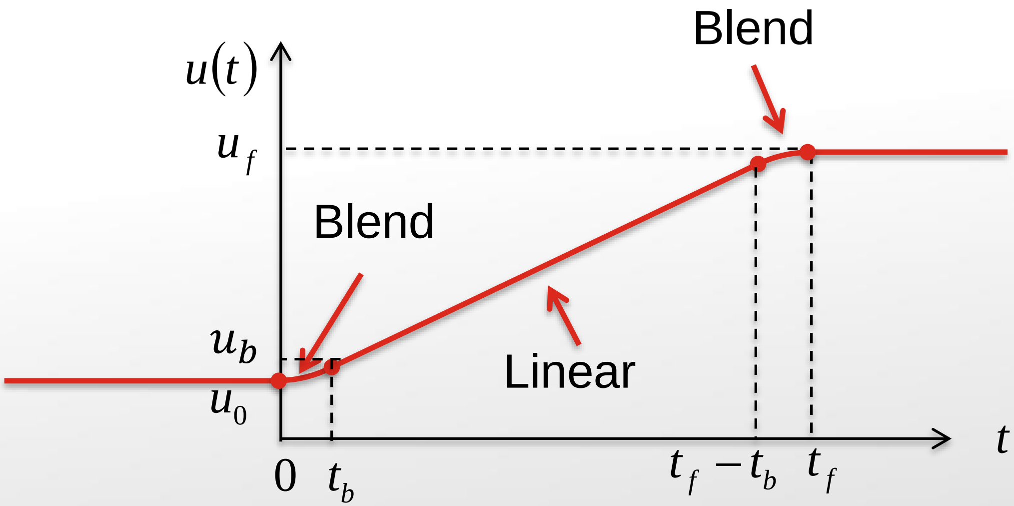
- The following summarizes the trajectories we have learnt so far:
  - Straight line:
    - Advantage: Constant velocity during motion.
    - Disadvantage: Discontinuous velocity at start and end points.
  - Polynomials:
    - Advantage: Smooth motion
    - Disadvantage: Velocity is not constant during motion.





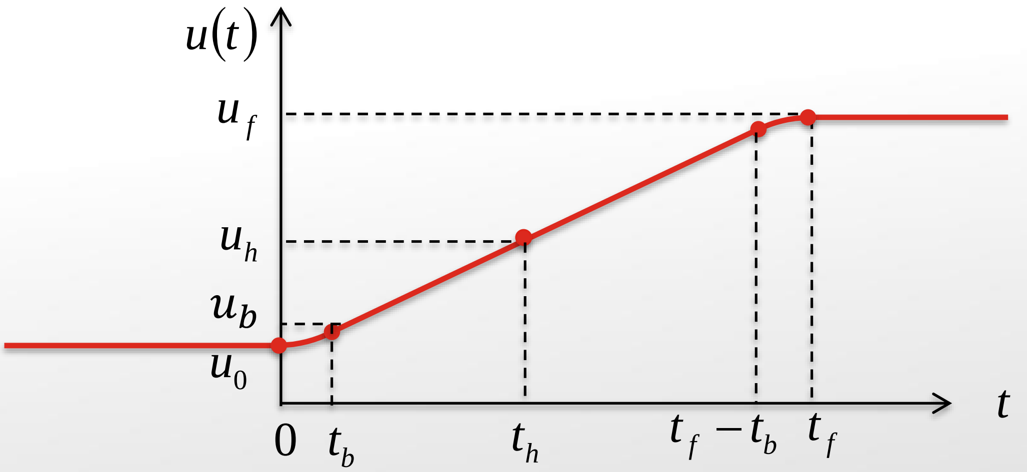
Linear Function w. Parabolic Blends

- Can we achieve:
  - Constant velocity during motion, AND
  - Smooth and continuous motion at the start and end points?
- Yes! We combine the ideas from the straight line and from the polynomial curves:
  - Linear Function with Parabolic Blends



Linear Function w. Parabolic Blends

- Assumptions / requirements:
  - Both the parabolic blends have the same time duration.
    - Therefore the same acceleration (apart from the sign) is used for both blends.
  - The solution is symmetric about the halfway point in time (t_h) and position (u_h).
  - The velocity at the end of blend region same as that of linear region.



- Question: how to get u_b and t_b?

Linear Function w. Parabolic Blends

- The last requirement translates to the following equations:
  - $\ddot{u} \cdot t_b = \frac{u_h - u_b}{t_h - t_b}$ where $\ddot{u}$ is the constant acceleration during blend region.
- Next, u_b is given by: $u_b = u_0 + \frac{1}{2}\ddot{u} \cdot t_b^2$
- At the desired end point, the position is u_f and the time is t_f.
- Note that: $u_h = \frac{1}{2}(u_0 + u_f)$ and $t_h = \frac{1}{2}t_f$
- Combining all above equations and eliminating u_b, we have:

$$\Rightarrow \quad \ddot{u} \cdot t_b^2 - \ddot{u} \cdot t_f \cdot t_b + (u_f - u_0) = 0$$

- Thus we can solve the above quadratic equation to get t_b.
- And then calculate u_b using $u_b = u_0 + \frac{1}{2}\ddot{u} \cdot t_b^2$

Linear Function w. Parabolic Blends

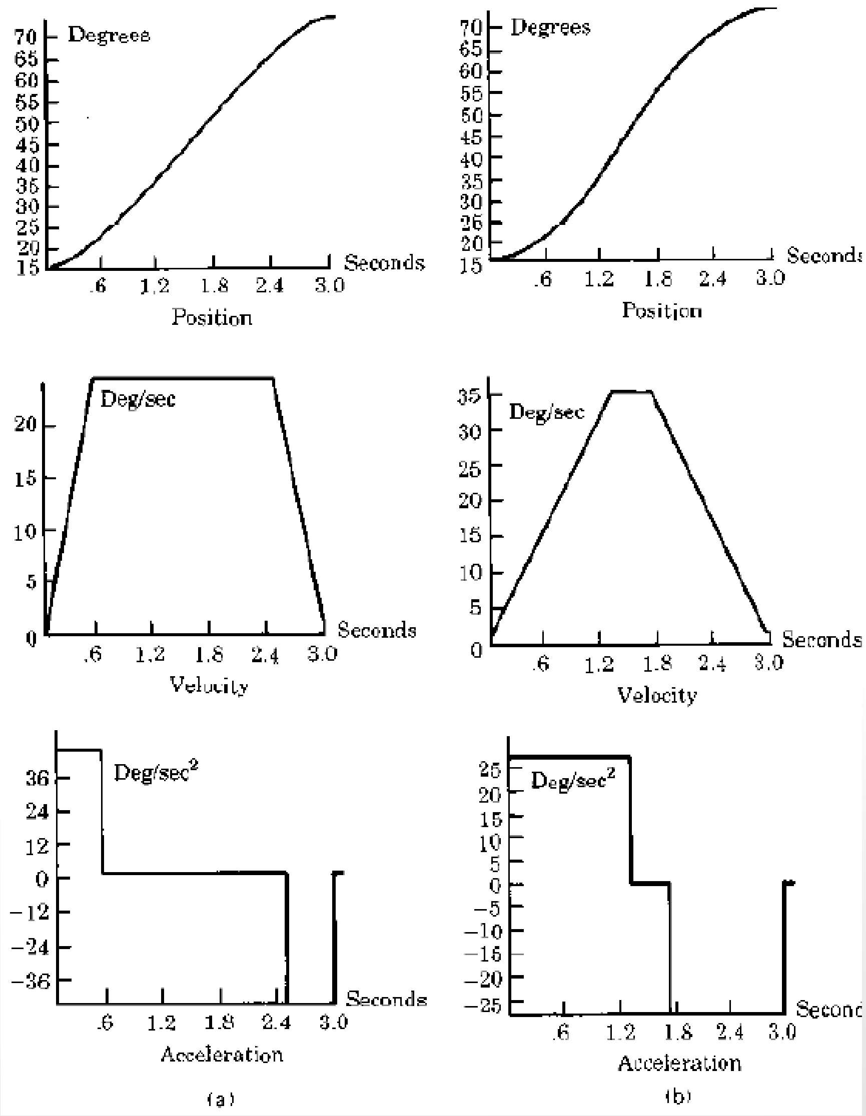
- Summary:
- The steps in obtaining the linear function with parabolic blends are:
  - Given u_0, u_f and t_f.
  - Choose desired acceleration $\ddot{u}$.
  - Calculate t_b based on: $\ddot{u}\cdot t_b^2 - \ddot{u}\cdot t_f \cdot t_b + \left(u_f - u_0\right) = 0$
    - i.e. $$t_b = \frac{\ddot{u}t_f \pm \sqrt{\ddot{u}^2 t_f^2 - 4\ddot{u}\left(u_f - u_0\right)}}{2\ddot{u}}$$

Choose "minus" only since t_b should be less than $\frac{1}{2}t_f$

  - Finally, calculate u_b based on:

$$u_b = u_0 + \frac{1}{2}\ddot{u}\cdot t_b^2$$

Linear Function w. Parabolic Blends



- Notes: Acceleration must be chosen to be high enough. Otherwise solution to t_b will not exist.
- E.g. if acceleration is small, the linear region shrinks.
- If acceleration is too small, there may be no more linear region.

Content

- Introduction
- Cartesian Space Schemes vs. Joint Space Schemes
- Cubic Polynomial
- Quintic Polynomial
- Linear Function with Parabolic Blends
- Matlab / Simulink Simulation

Quintic Polynomial

- We will use Quintic Polynomial for the simulation:

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

- The parameters are:

$$a_0 = u_0 \qquad a_1 = \dot{u}_0 \qquad a_2 = \frac{\ddot{u}_0}{2}$$

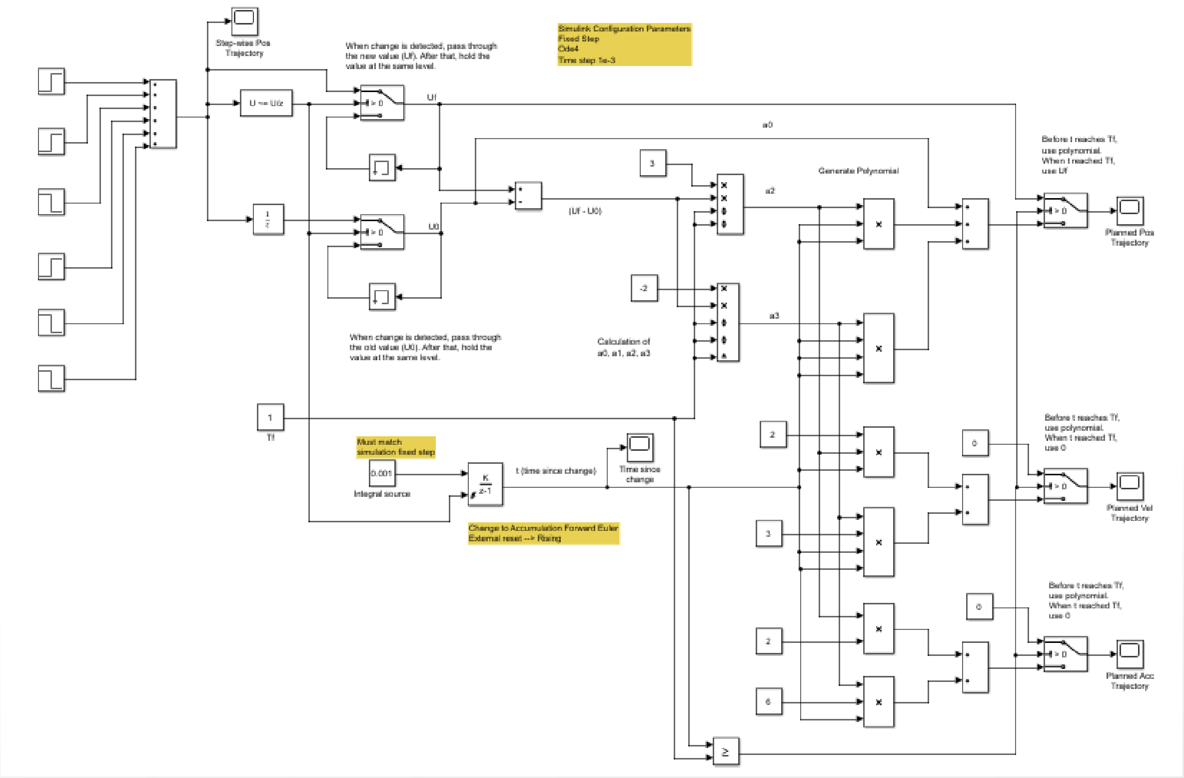
$$a_3 = \frac{20u_f - 20u_0 - (8\dot{u}_f + 12\dot{u}_0)t_f - (3\ddot{u}_0 - \ddot{u}_f)t_f^2}{2t_f^3}$$

$$a_4 = \frac{30u_0 - 30u_f + (14\dot{u}_f + 16\dot{u}_0)t_f + (3\ddot{u}_0 - 2\ddot{u}_f)t_f^2}{2t_f^4}$$

$$a_5 = \frac{12u_f - 12u_0 - (6\dot{u}_f + 6\dot{u}_0)t_f - (\ddot{u}_0 - \ddot{u}_f)t_f^2}{2t_f^5}$$

Quintic Polynomial

- This can be done in the following way:



- Please see attached Matlab Simulink file in Canvas.

Tutorial Assignments

- **Question 1:**
  - A single-link robot with a rotary joint is motionless at $\theta = -5^o$. It is desired to move the joint in a smooth manner to $\theta = 80^o$ in 4 seconds.
  - Find the coefficients of a cubic which accomplishes this motion and brings the arm to rest at the goal.
  - Sketch the position, velocity and acceleration of the joint as a function of time.

Tutorial Assignments

- **Question 2:**
  - A single-link robot with a rotary joint is motionless at $\theta = -5^o$. It is desired to move the joint in a smooth manner to $\theta = 80^o$ in 4 seconds and also stop smoothly.
  - Find the corresponding parameters of a linear trajectory with parabolic blends.
  - Sketch the position, velocity and acceleration of the joint as a function of time.

Tutorial Assignments

- **Question 3:**
  - We wish to move a single joint from θ_0 to θ_f starting from rest, ending at rest, in time t_f.
  - The values of θ_0 to θ_f are given, but we wish to compute t_f so that the velocity never exceeds a maximum value (θ-dot max), and the acceleration never exceeds a maximum value (θ-double-dot max).
  - Use a single cubic segment, and give an expression for t_f and for the cubic's coefficients.

Thank you!

Have a good evening.