MEAM 520
Lecture 10: Trajectory Planning in Joint Space

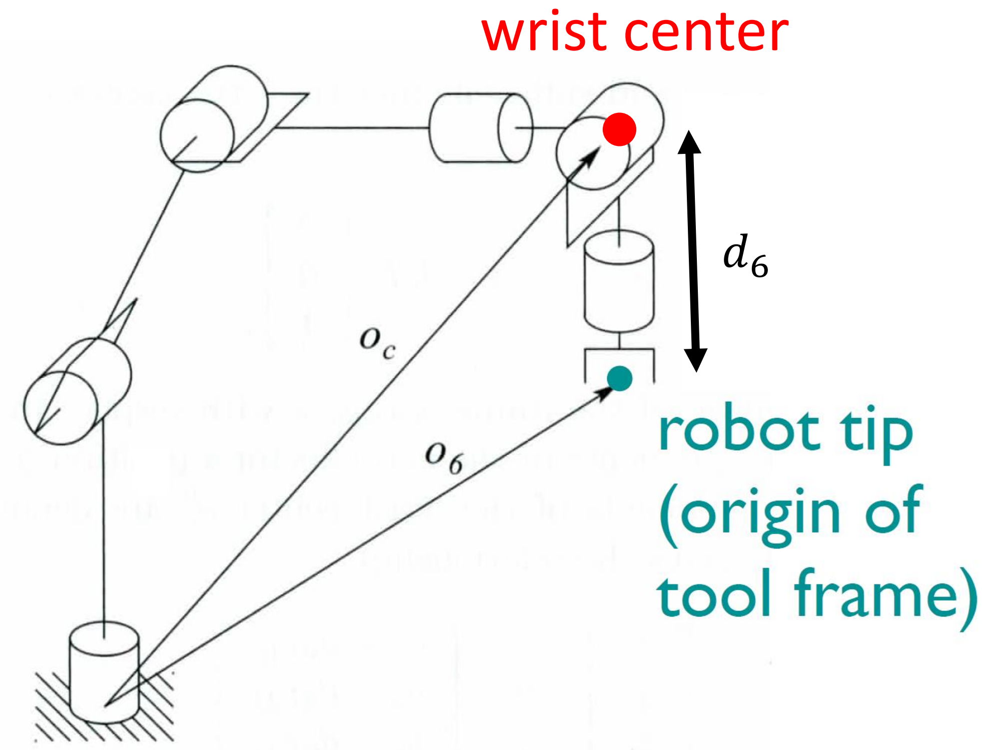
Cynthia Sung, Ph.D.
Mechanical Engineering & Applied Mechanics
University of Pennsylvania

Given $\mathbf{H} = \begin{bmatrix} \mathbf{R} & o \\ 0 & 1 \end{bmatrix}$ and a certain manipulator with *n* joints,

find $q_1,\ldots,q_n$ such that $\mathbf{T}_n^0(q_1, \ldots, q_n) = \mathbf{H}$

UGLY!

Last Time: Kinematic Decoupling



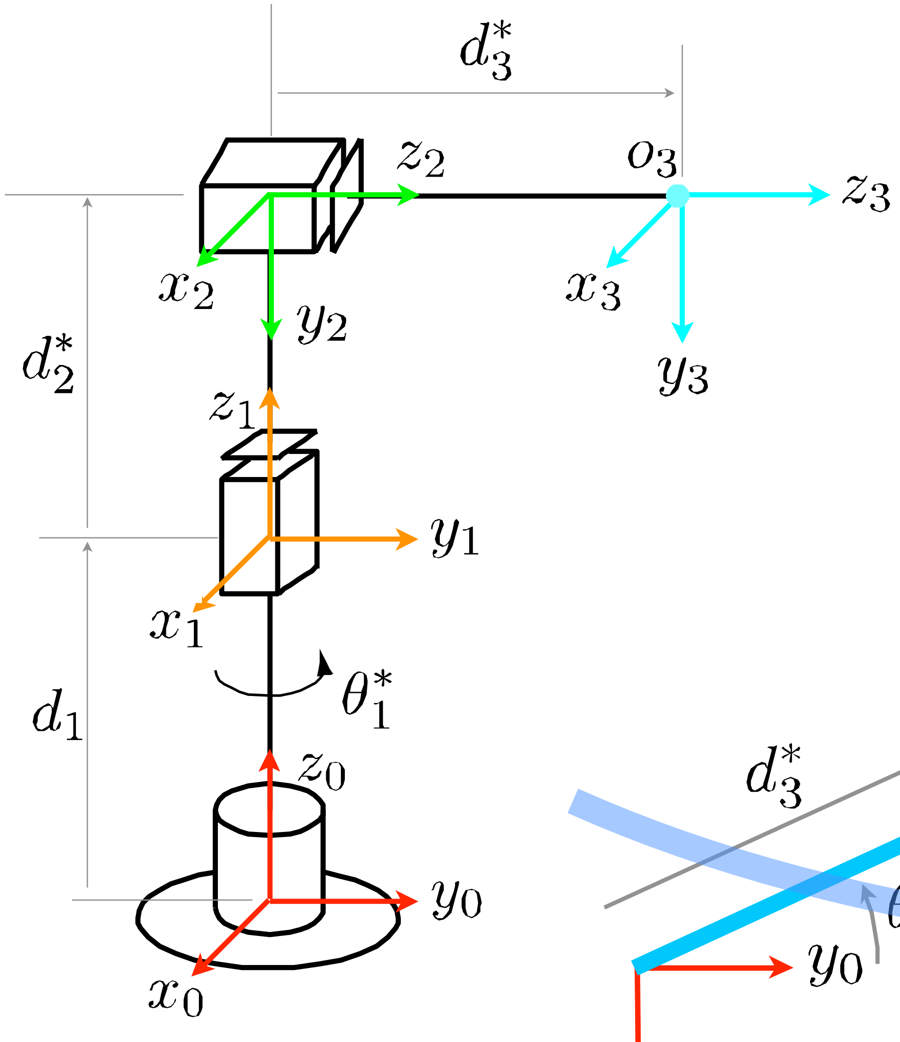
$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} o_x - d_6 r_{13} \\ o_y - d_6 r_{23} \\ o_z - d_6 r_{33} \end{bmatrix}$$

position

$$\mathbf{R}_6^3 = (\mathbf{R}_3^0)^{-1}\mathbf{R} = (\mathbf{R}_3^0)^{\mathrm{T}}\mathbf{R}$$

orientation

Two Inverse Position Kinematics Approaches



$$o_3^0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$x = -d_3^* \sin(\theta_1^*)$$

$$y = d_3^* \cos(\theta_1^*)$$

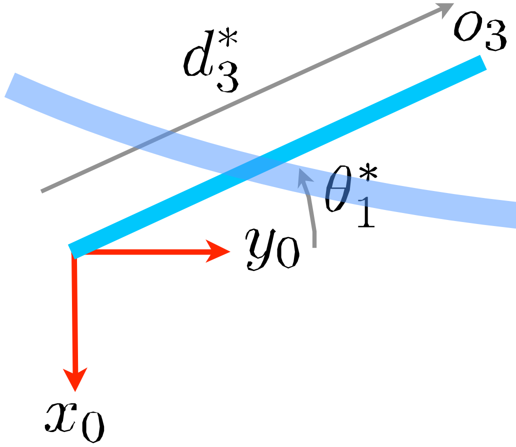
$$z = d_1 + d_2^*$$

$$\theta_1^* = ?$$

$$d_2^* = ?$$

$$d_3^* = ?$$

Algebra



Geometry

$$\boxed{\theta_1^* = \operatorname{atan2}\left(\frac{-x/d_3^*}{y/d_3^*}\right)}$$

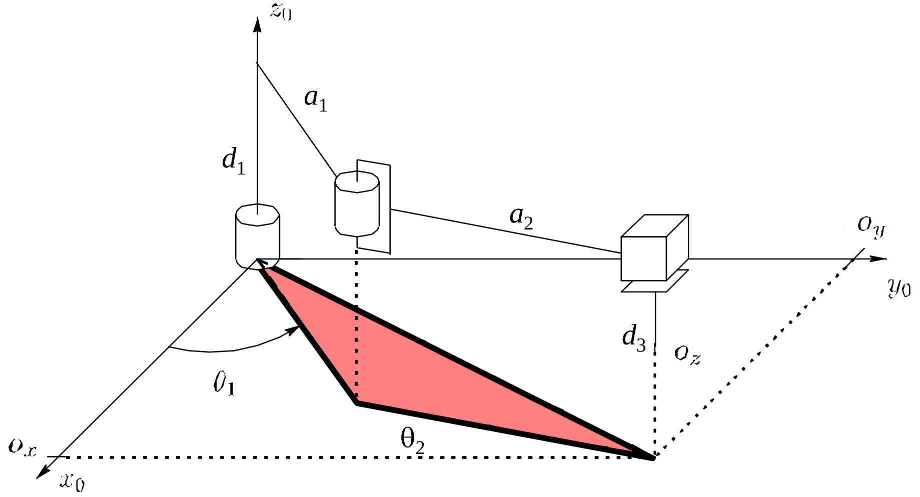
$$\boxed{d_2^* = z - d_1}$$

$$\boxed{d_3^* = \pm\sqrt{x^2 + y^2}}$$

Complete SCARA IK Example

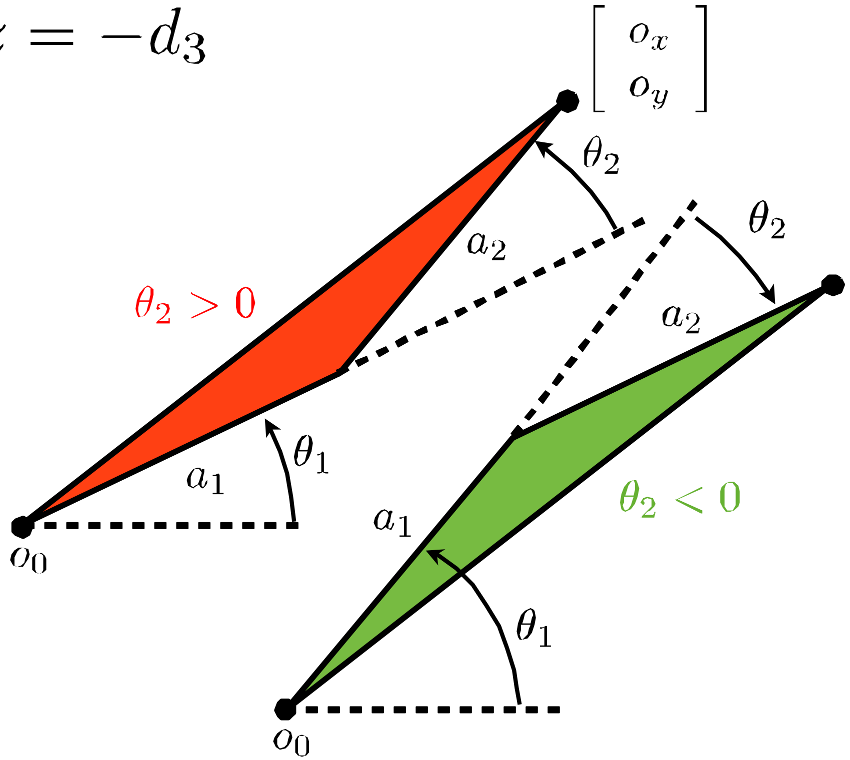
$$o_4^0 = \begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix}$$

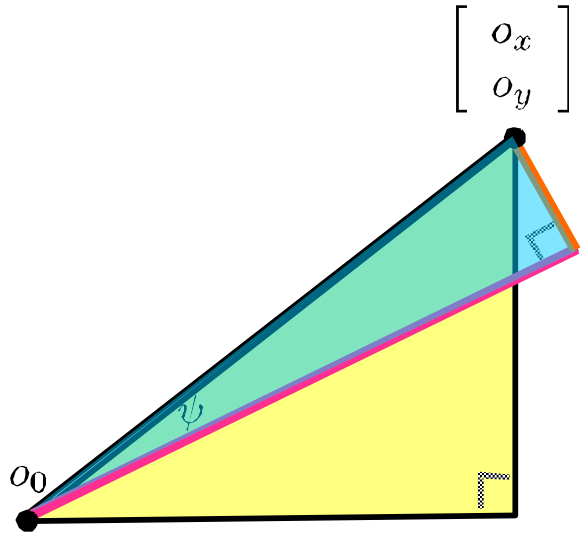
$$\theta_1 = ? \quad \theta_2 = ? \quad d_3 = ?$$



Start with Forward Position Kinematics

$$x = a_1 c_1 + a_2 c_{12}$$
$$y = a_1 s_1 + a_2 s_{12}$$
$$z = -d_3$$





$$\boxed{d_3 = -o_z}$$

$$\boxed{\cos\theta_2 = \frac{o_x^2 + o_y^2 - a_1^2 - a_2^2}{2a_1 a_2}}$$

$$\boxed{\theta_2 = \operatorname{atan2}\left(\frac{\pm\sqrt{1-\cos^2\theta_2}}{\cos\theta_2}\right)}$$

$$\boxed{\theta_1 = \operatorname{atan2}\left(\frac{o_y}{o_x}\right) - \operatorname{atan2}\left(\frac{a_2 \sin\theta_2}{a_1 + a_2\cos\theta_2}\right)}$$

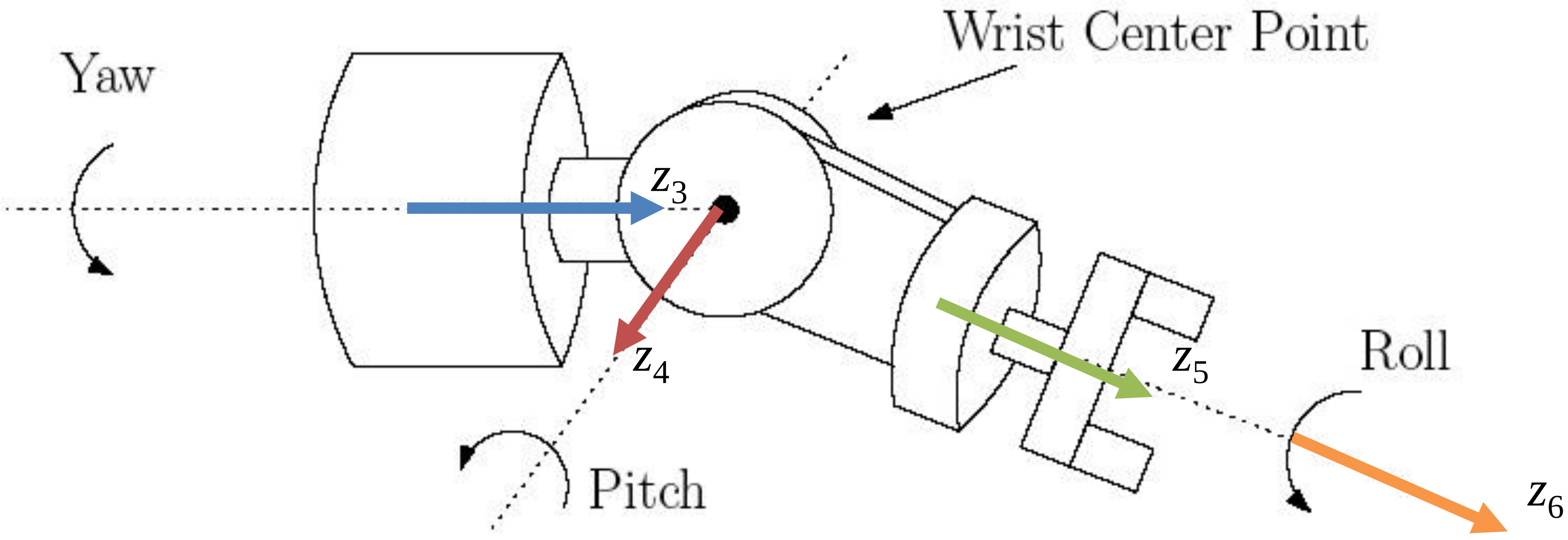
Geometric Interpretation of Solution Method

$$\begin{array}{cccc} & \hat{x}_6^3 & \hat{y}_6^3 & \hat{z}_6^3 \\ \mathbf{R}_6^3 = & \left[\begin{array}{c} c_\phi c_\theta c_\psi - s_\phi s_\psi \\ s_\phi c_\theta c_\psi + c_\phi s_\psi \\ -s_\theta c_\psi \end{array}\right. & \begin{array}{c} -c_\phi c_\theta s_\psi - s_\phi c_\psi \\ -s_\phi c_\theta s_\psi + c_\phi c_\psi \\ s_\theta s_\psi \end{array} & \left.\begin{array}{c} c_\phi s_\theta \\ s_\phi s_\theta \\ c_\theta \end{array}\right] \end{array}$$

Plug in to solve for ϕ

Plug in to solve for ψ

Solve for θ



Given a desired pose for our end-effector, $\boldsymbol{H}_e^0 = \begin{bmatrix} R & o \\ 0 & 1 \end{bmatrix}$, we use **inverse kinematics** to find the necessary joint values, $q_1 \ldots q_n$

What do we use to control ***how*** the robot gets to this pose?

Two arm poses... How do I move between them?

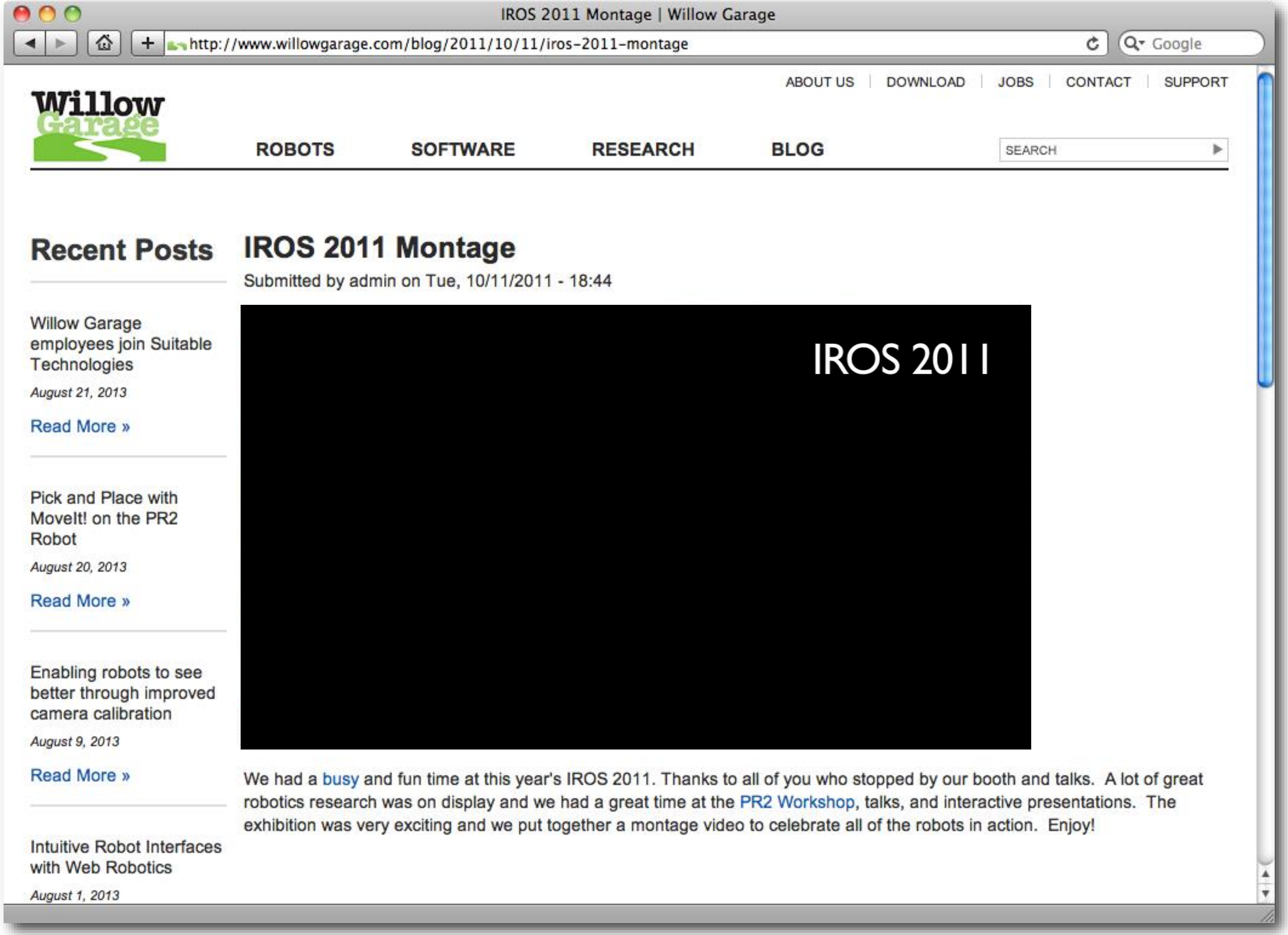
A trajectory is a function of time $\vec{q}(t)$

Such that $\vec{q}(t_0) = \vec{q}_s$

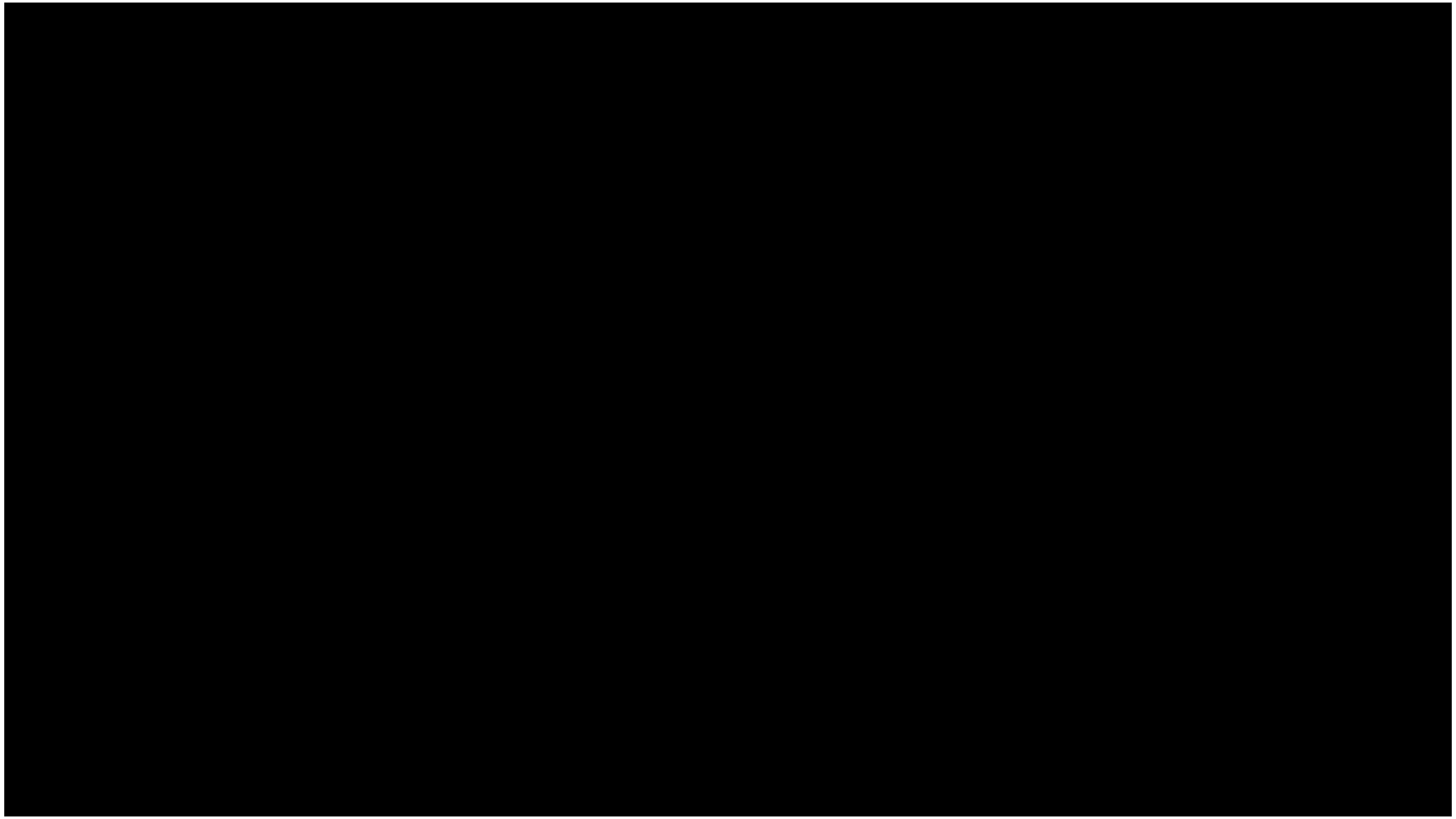
and $\vec{q}(t_f) = \vec{q}_f$

Parameterized by time, so we can compute velocities and accelerations along the trajectory by differentiation.

https://youtu.be/l9UFsRAM_XM



http://www.willowgarage.com/blog/2011/10/11/iros-2011-montage



https://youtu.be/tIIJME8-au8

Outline of the planning strategy

- Today: Trajectory planning between two points in the absence of obstacles
- 10/6: Configuration space planning using grid-based methods (1960s)
- 10/8: Configuration space planning using sampling (1980s)
- 10/21: Lab 3 due (implement a planner)

Later this semester:

- 10/27: Workspace planning with artificial potential fields (1980s)
- 10/29: Real-time planning paper reading (2000s)

A trajectory is a function of time $\vec{q}(t)$

Such that $\vec{q}(t_0) = \vec{q}_s$

and $\vec{q}(t_f) = \vec{q}_f$

How many trajectories exist that satisfy these constraints?

Infinitely many.

What if I also specify starting and final velocities?

There are still infinitely many trajectories.

Roboticists typically choose trajectories from a finitely parameterizable family, such as polynomials of degree n.

How many constraints may we impose when calculating an nth-order polynomial?

cubic polynomial

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

We may impose n+1 constraints because there are n+1 coefficients in an nth-order polynomial.

For point-to-point motion, each joint's motion is typically planned independently, so we'll consider just a single joint angle.

$$\text{instead of } \vec{q}(t)$$

$$q(t) = \theta_i(t) \quad \text{or} \quad q(t) = d_i(t)$$

Simplest Situation: Specifying Joint Value Only

Initial Condition

$$q(t_0) = q_0$$

Final Condition

$$q(t_f) = q_f$$

The equation $q(t) = a_0 + a_1 t$ defines a line. Solve for the coefficients a_0 and a_1 that satisfy the initial and final position constraints of $q(t_0) = q_0$ and $q(t_f) = q_f$.

$$q(t) = a_0 + a_1 t$$

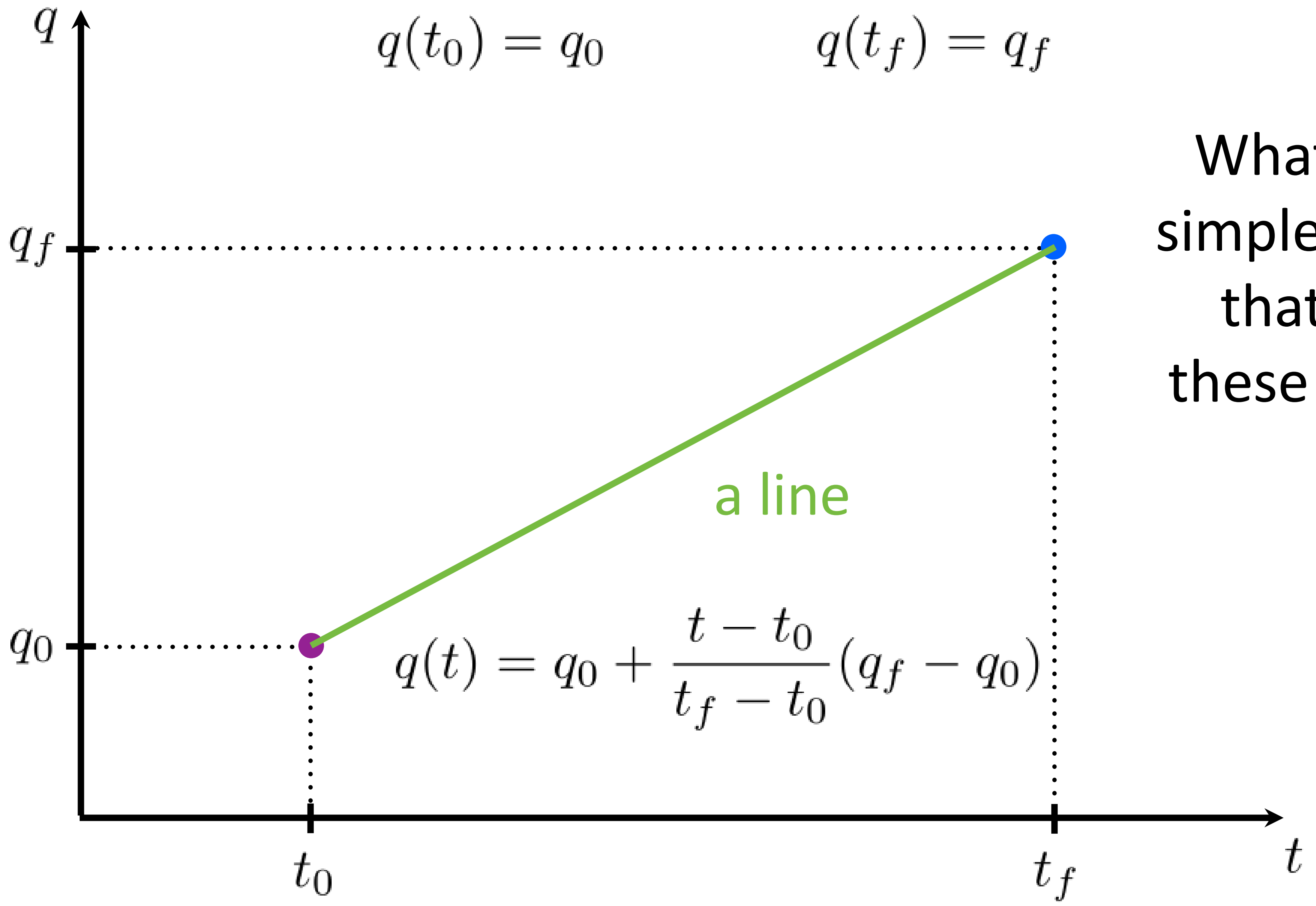
$$q_0 = a_0 + a_1 t_0$$

$$q_f = a_0 + a_1 t_f$$

$$\boxed{a_0 = q_0 - \frac{q_f - q_0}{t_f - t_0} \cdot t_0}$$

$$\boxed{a_1 = \frac{q_f - q_0}{t_f - t_0}}$$

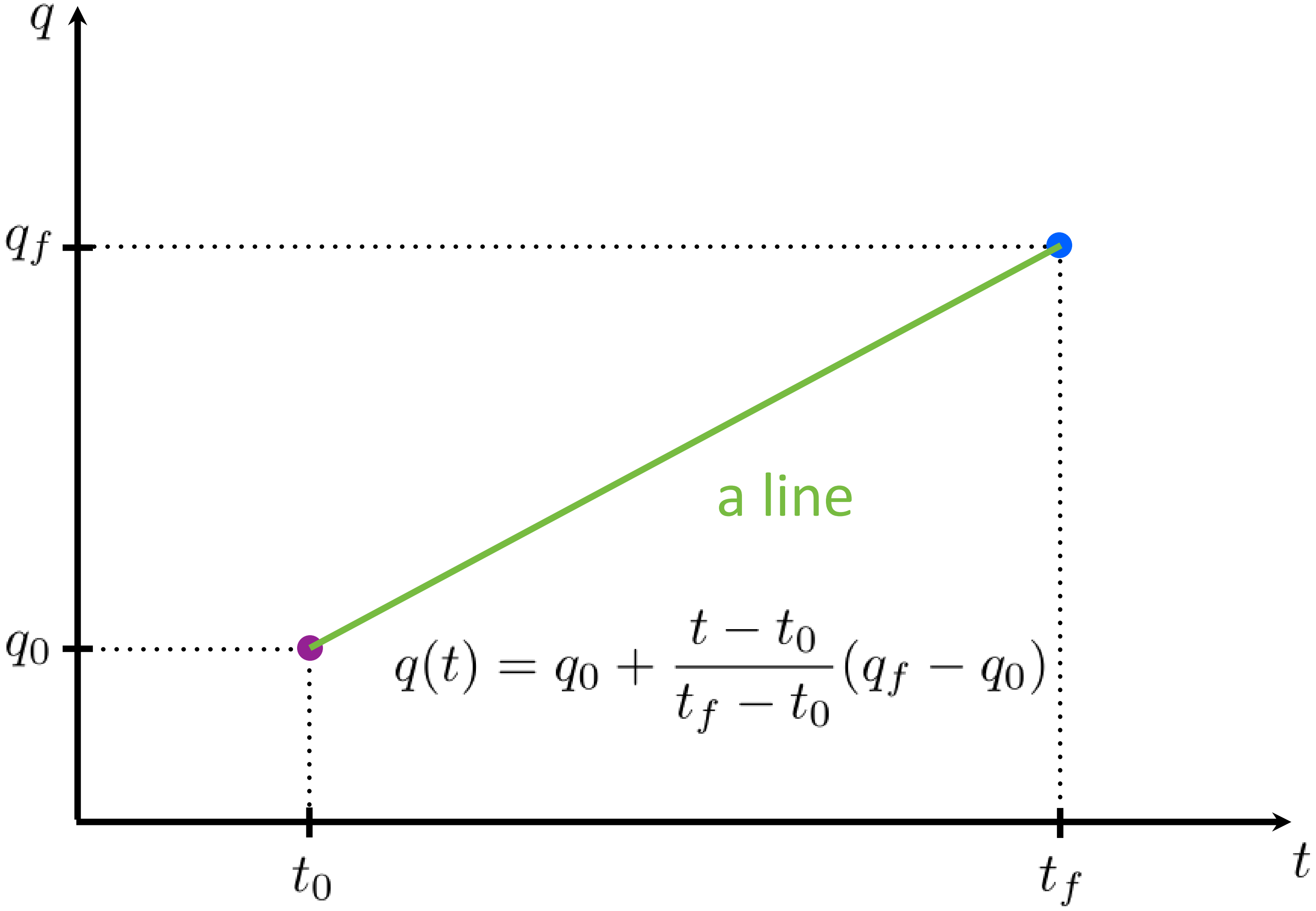
Simplest Situation: Specifying Joint Value Only

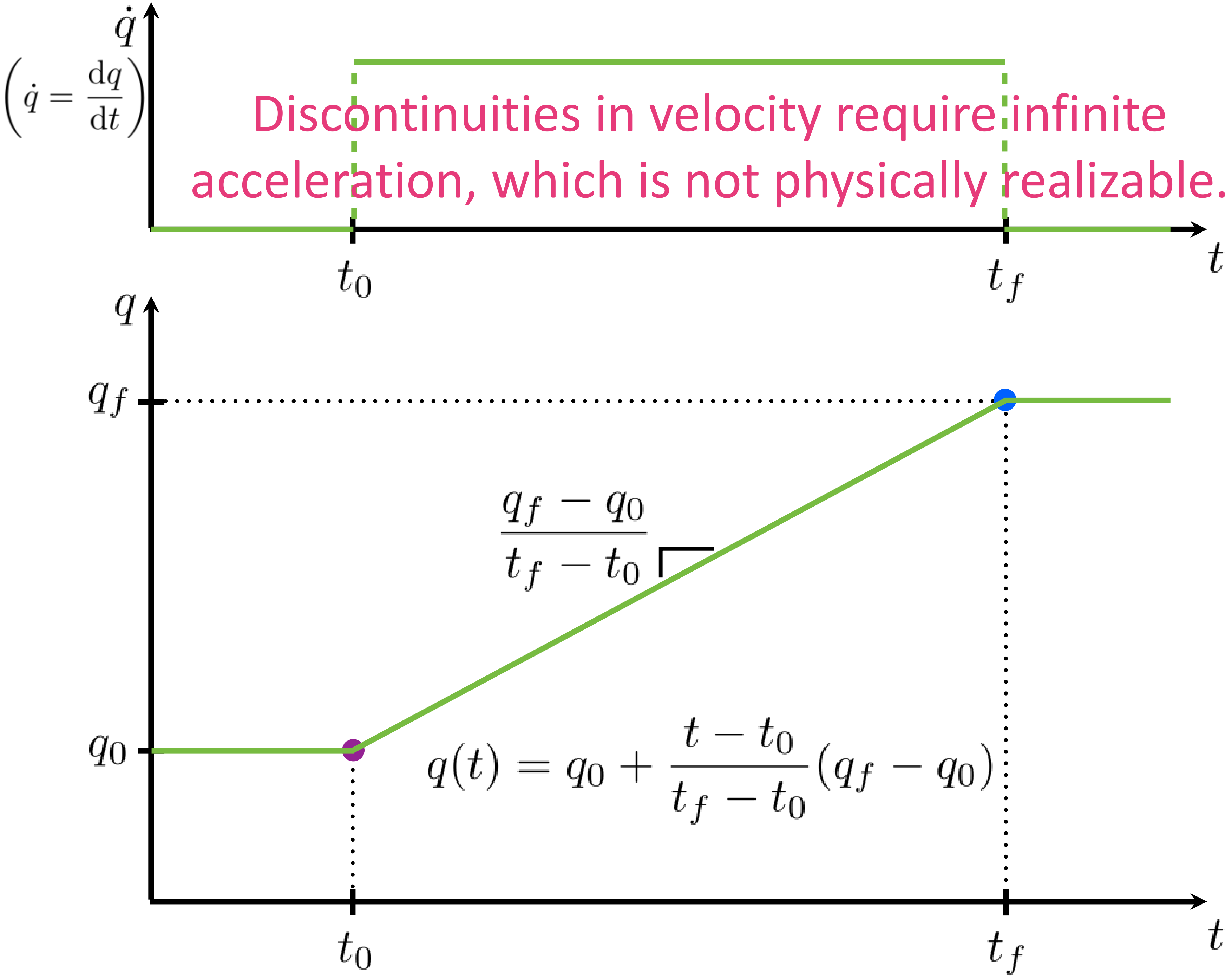


What is the simplest curve that joins these points?

Linear interpolation is really useful!

Why do you think SHV doesn't present lines?





$\dot{q}$
$\left(\dot{q} = \frac{\mathrm{d}q}{\mathrm{d}t}\right)$
Discontinuities in velocity require infinite acceleration, which is not physically realizable.
t_0
t_f
t
q
q_f
q_0
$\frac{q_f - q_0}{t_f - t_0}$
$q(t) = q_0 + \frac{t - t_0}{t_f - t_0}(q_f - q_0)$
t_0
t_f
t

Robots are actually flexible!

Command smooth trajectories to avoid exciting flexibilities.

Robot manipulators are composed of:

- Rigid links
- Connected by joints
- To form a kinematic chain

Specifying Joint Values and First Time Derivatives

Initial Conditions

$$q(t_0) = q_0$$

Units angle or distance, e.g., rad or m

Units?

$$\dot{q}(t_0) = v_0$$

Units angle per time or distance per time, e.g., rad/s or m/s

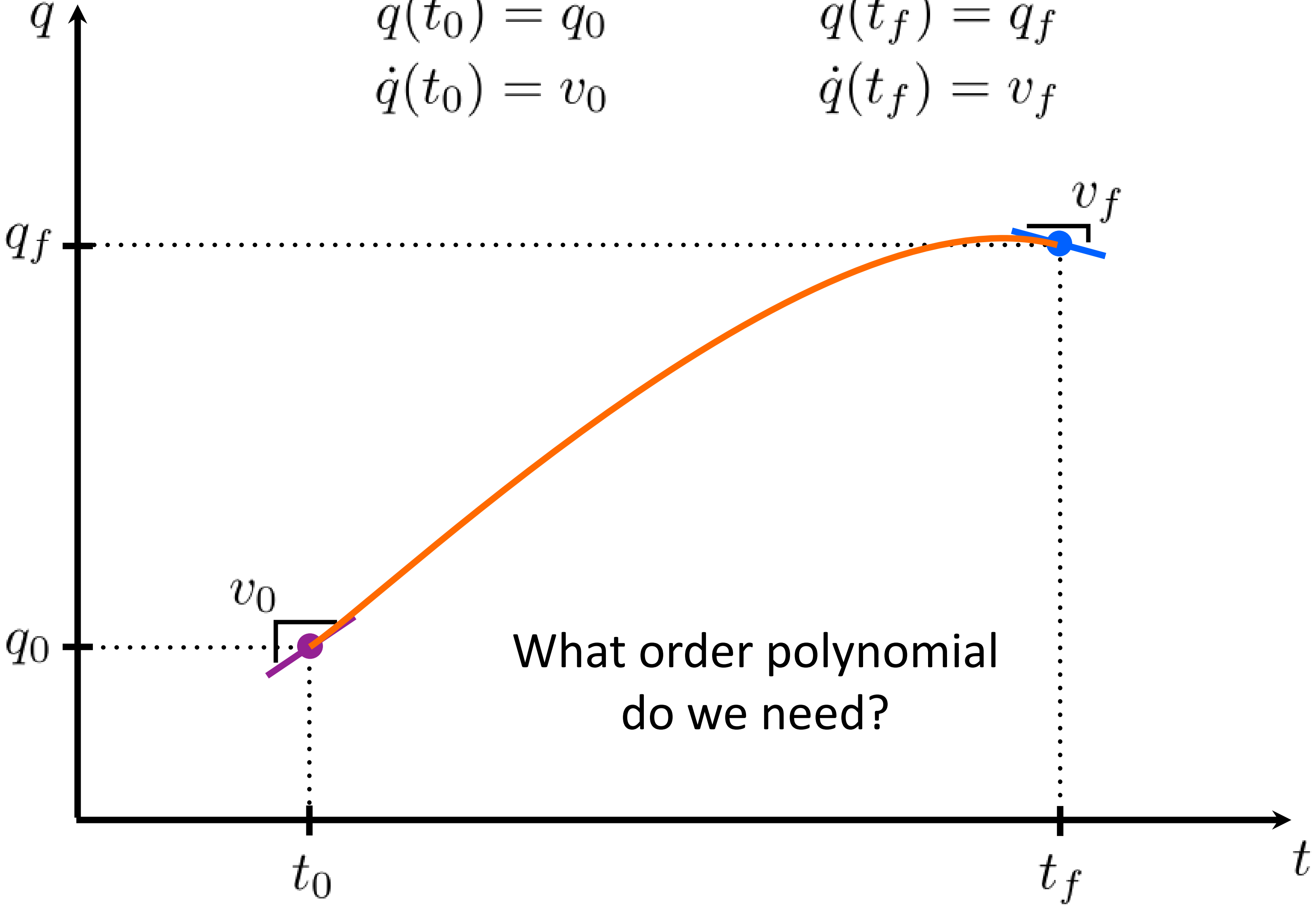
Final Conditions

$$q(t_f) = q_f$$

$$\dot{q}(t_f) = v_f$$

Specifying Joint Values and First Time Derivatives

$$q(t_0) = q_0 \qquad q(t_f) = q_f$$
$$\dot{q}(t_0) = v_0 \qquad \dot{q}(t_f) = v_f$$



What order polynomial do we need?

Specifying Joint Values and First Time Derivatives

Cubic Polynomial Trajectories

start		end
$q(t_0) = q_0$	$\longrightarrow$	$q(t_f) = q_f$
$\dot{q}(t_0) = v_0$	$\longrightarrow$	$\dot{q}(t_f) = v_f$

cubic polynomial

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

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

$$q_0 = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3$$

$$v_0 = a_1 + 2a_2 t_0 + 3a_3 t_0^2$$

$$q_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3$$

$$v_f = a_1 + 2a_2 t_f + 3a_3 t_f^2$$

Specifying Joint Values and First Time Derivatives

Cubic Polynomial Trajectories

System of Four Equations

$$q_0 = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3$$

$$v_0 = a_1 + 2a_2 t_0 + 3a_3 t_0^2$$

$$q_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3$$

$$v_f = a_1 + 2a_2 t_f + 3a_3 t_f^2$$

time matrix

conditions

unknown coefficients

Reformulate in $\vec{b} = A\vec{x}$ form

$$\vec{x} = A^{-1}\vec{b}$$

$$\vec{x} = A\backslash\vec{b}$$

$$\begin{bmatrix} q_0 \\ v_0 \\ q_f \\ v_f \end{bmatrix} = \begin{bmatrix} 1 & t_0 & {t_0}^2 & {t_0}^3 \\ 0 & 1 & 2t_0 & 3{t_0}^2 \\ 1 & t_f & {t_f}^2 & {t_f}^3 \\ 0 & 1 & 2t_f & 3{t_f}^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

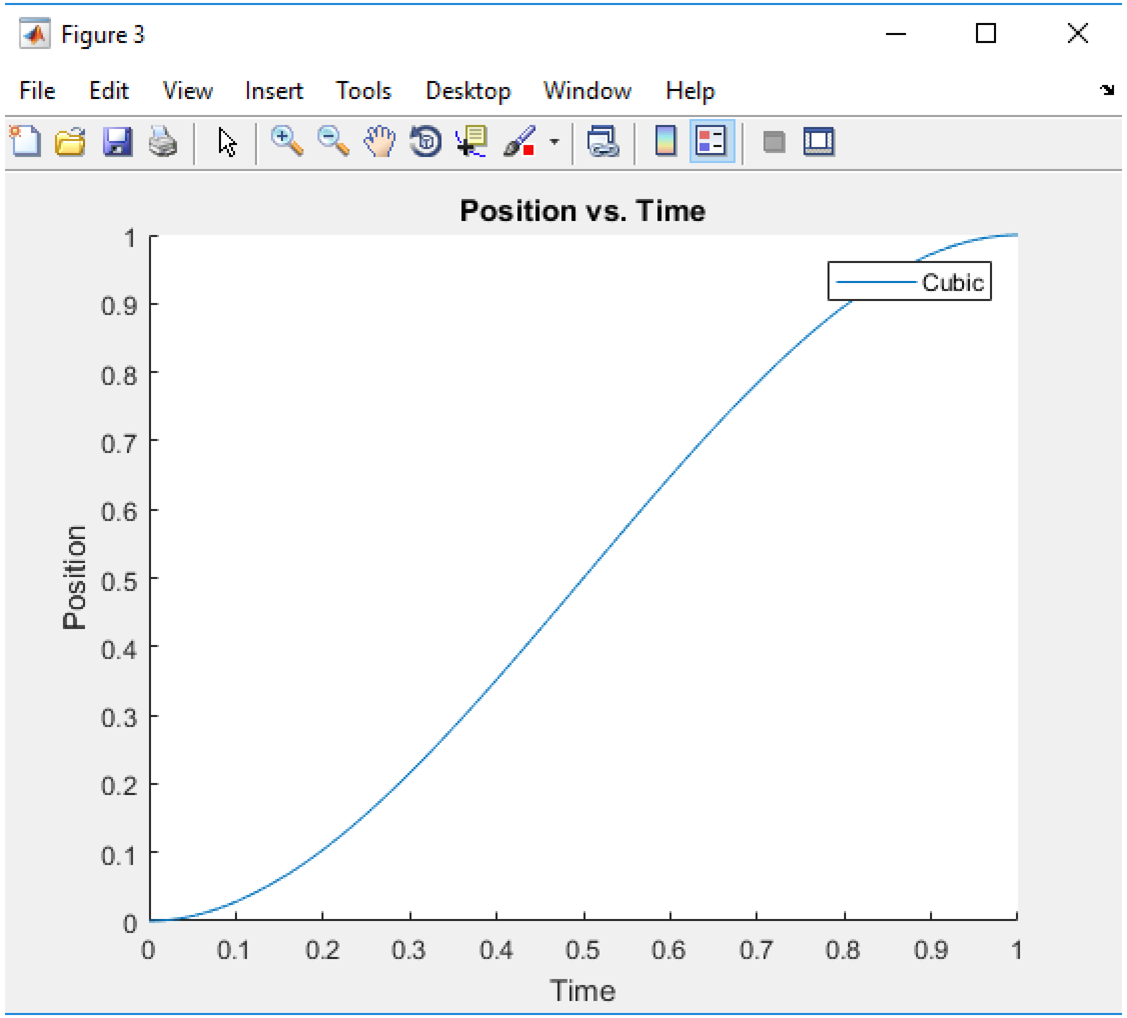


Figure 3
File
Edit
View
Insert
Tools
Desktop
Window
Help
Position vs. Time
Cubic
Position
Time
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

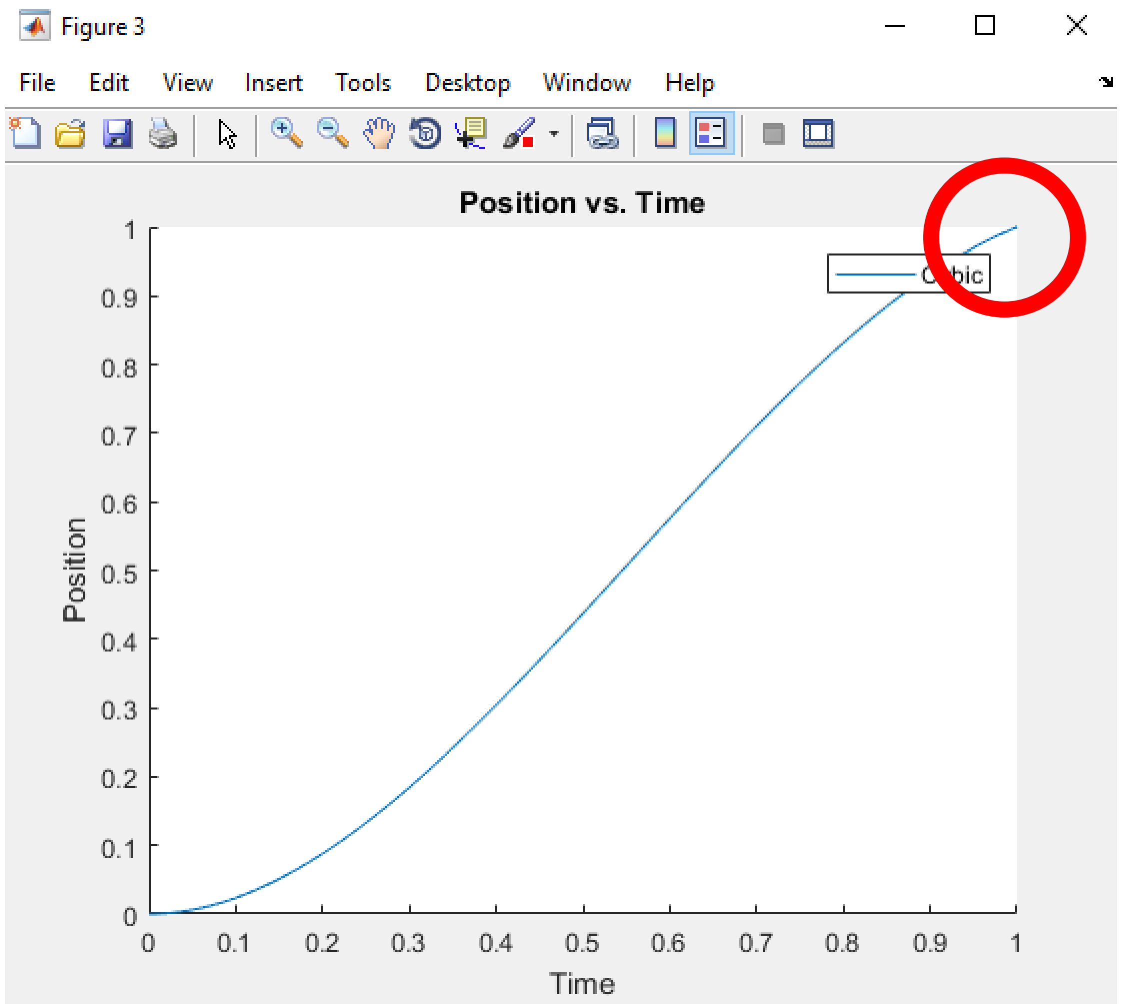
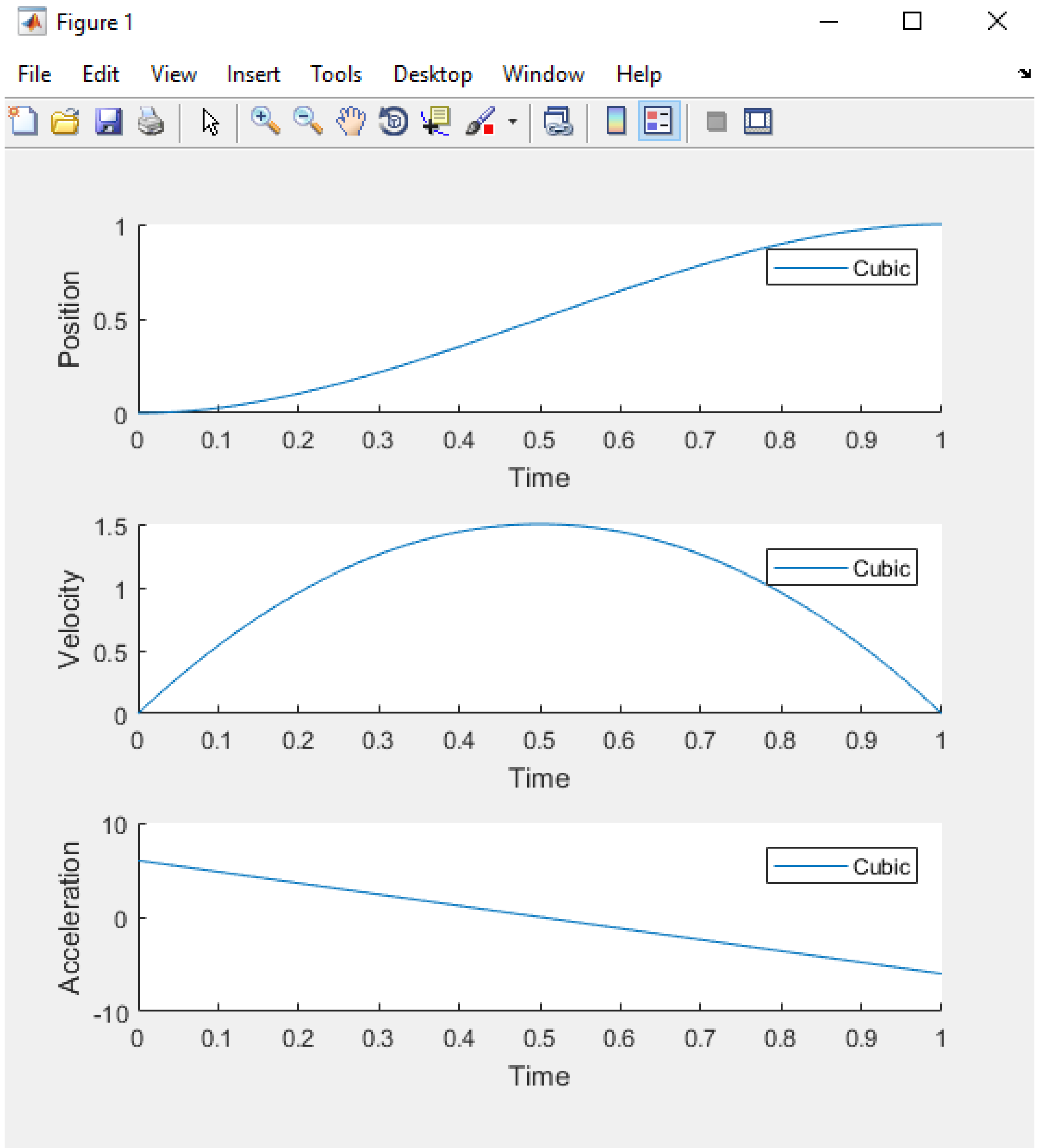


Figure 3
File
Edit
View
Insert
Tools
Desktop
Window
Help
Position vs. Time
Position
Time
Cubic
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

$v_f = 0.5$



start

end

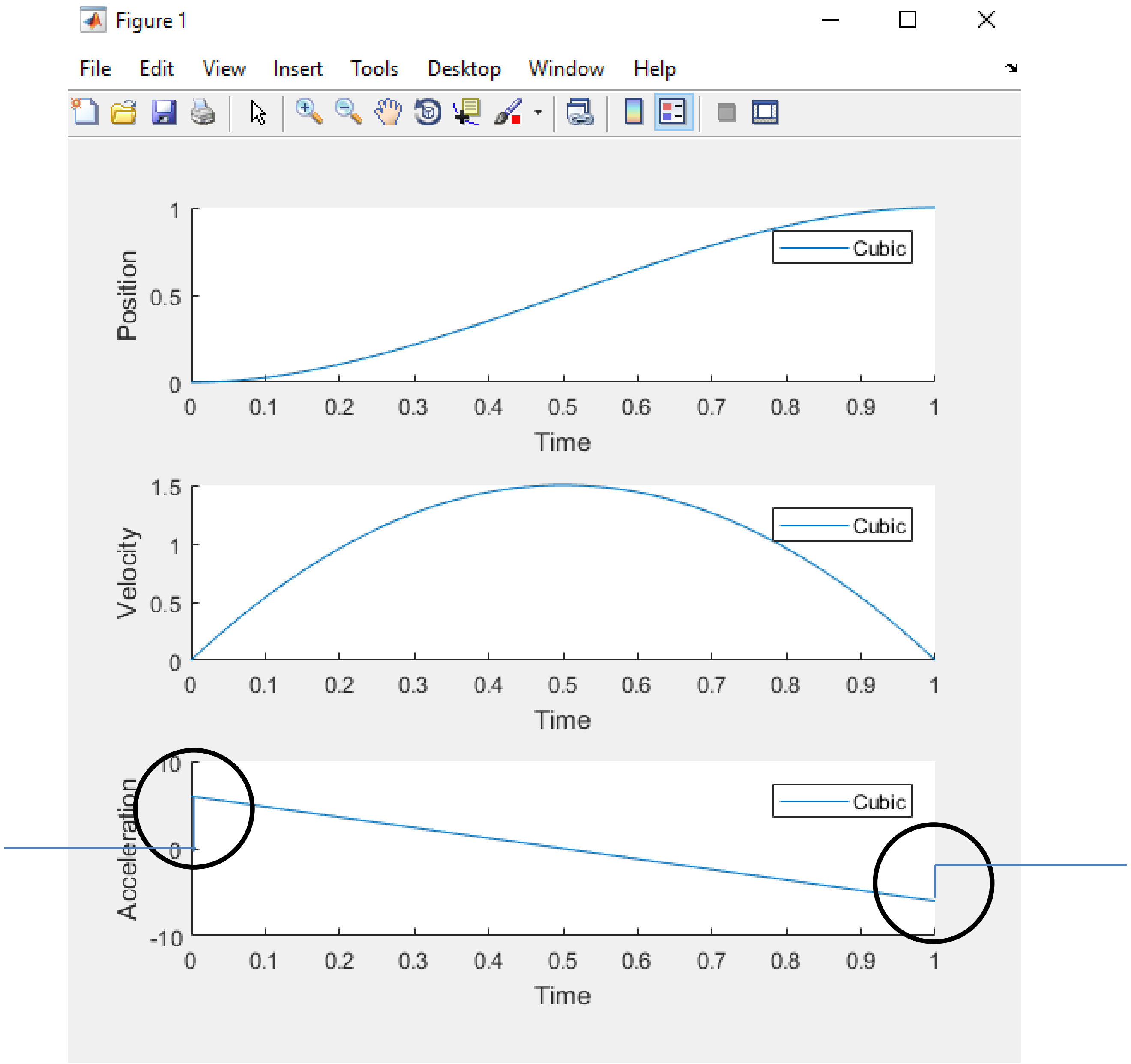
$$q(t_0) = q_0 \longrightarrow q(t_f) = q_f$$

$$\dot{q}(t_0) = v_0 \longrightarrow \dot{q}(t_f) = v_f$$

cubic polynomial

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

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



Discontinuities in acceleration require **step changes in force/torque**, which excites vibrational modes in the robot.

Time derivative of acceleration is **jerk**.

We don't want infinite **jerk**.

Specifying Joint Values Plus First and Second Time Derivatives

Initial Conditions

$$q(t_0) = q_0$$

Units angle or distance, e.g., rad or m

Units?

$$\dot{q}(t_0) = v_0$$

Units angle per time or distance per time, e.g., rad/s or m/s

$$\ddot{q}(t_0) = \alpha_0$$

Units angle per time per time or distance per time per time, e.g., rad/s^2 or m/s^2

Not the same α as in DH!

Final Conditions

$$q(t_f) = q_f$$

$$\dot{q}(t_f) = v_f$$

$$\ddot{q}(t_f) = \alpha_f$$

What kind of curve to use?

Specifying Joint Values Plus First and Second Time Derivatives

Quintic Polynomial Trajectories

start		end
$q(t_0) = q_0$	$\longrightarrow$	$q(t_f) = q_f$
$\dot{q}(t_0) = v_0$	$\longrightarrow$	$\dot{q}(t_f) = v_f$
$\ddot{q}(t_0) = \alpha_0$	$\longrightarrow$	$\ddot{q}(t_f) = \alpha_f$

quintic polynomial

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

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

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

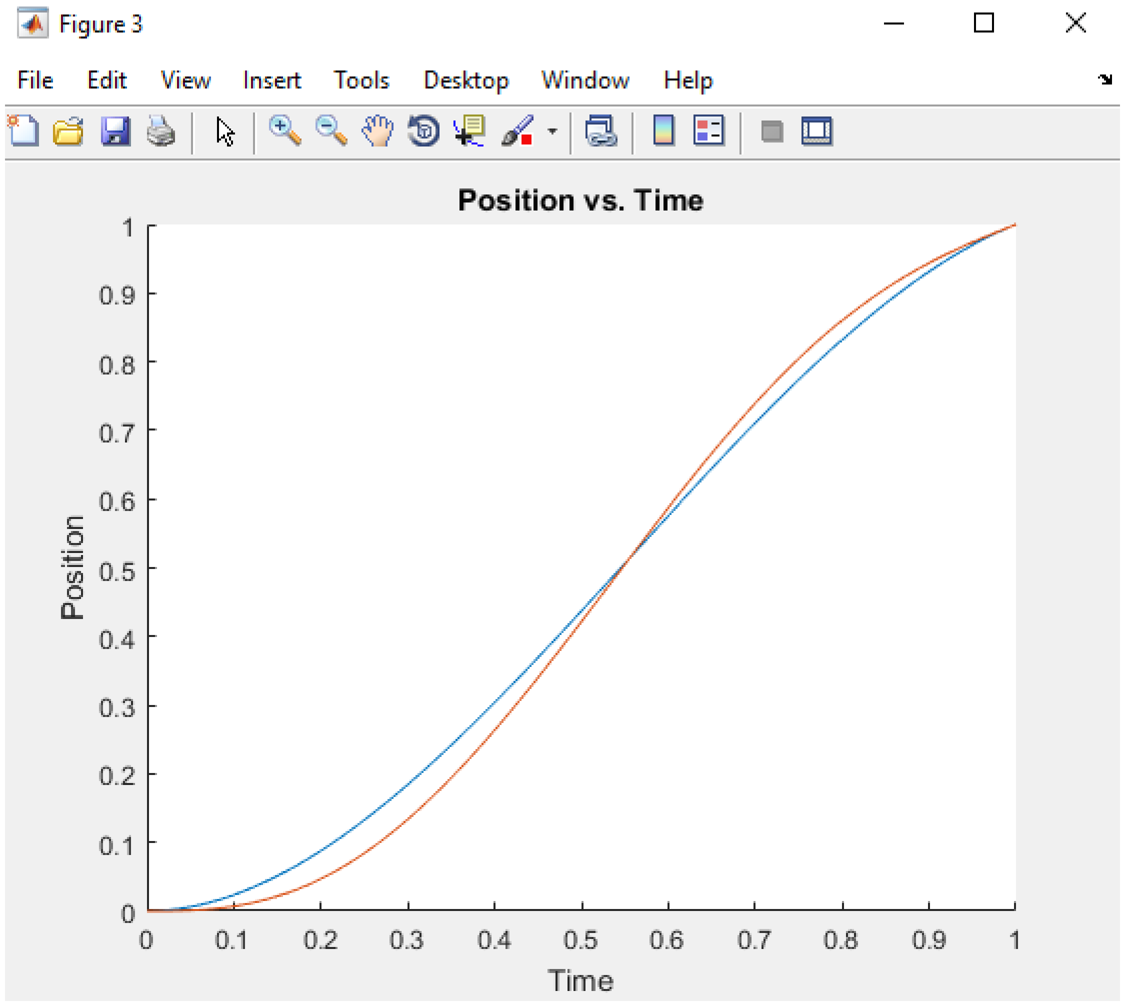


Figure 3
File Edit View Insert Tools Desktop Window Help
Position vs. Time
Position
Time
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

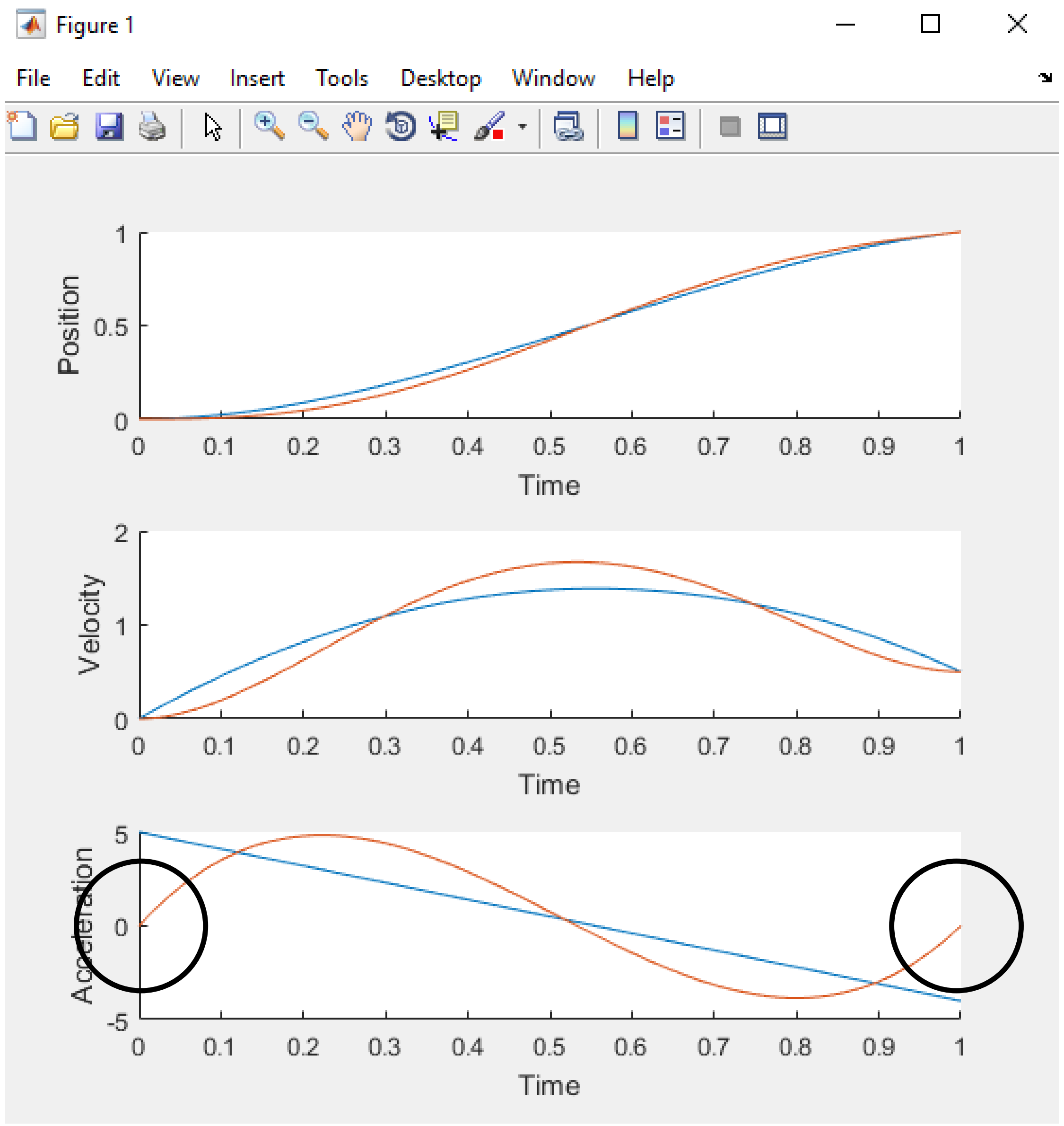


Figure 1
File Edit View Insert Tools Desktop Window Help
Position
Velocity
Acceleration
Time
Time
Time

start		end
$q(t_0) = q_0$	$\longrightarrow$	$q(t_f) = q_f$
$\dot{q}(t_0) = v_0$	$\longrightarrow$	$\dot{q}(t_f) = v_f$
$\ddot{q}(t_0) = \alpha_0$	$\longrightarrow$	$\ddot{q}(t_f) = \alpha_f$

quintic polynomial

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

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

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

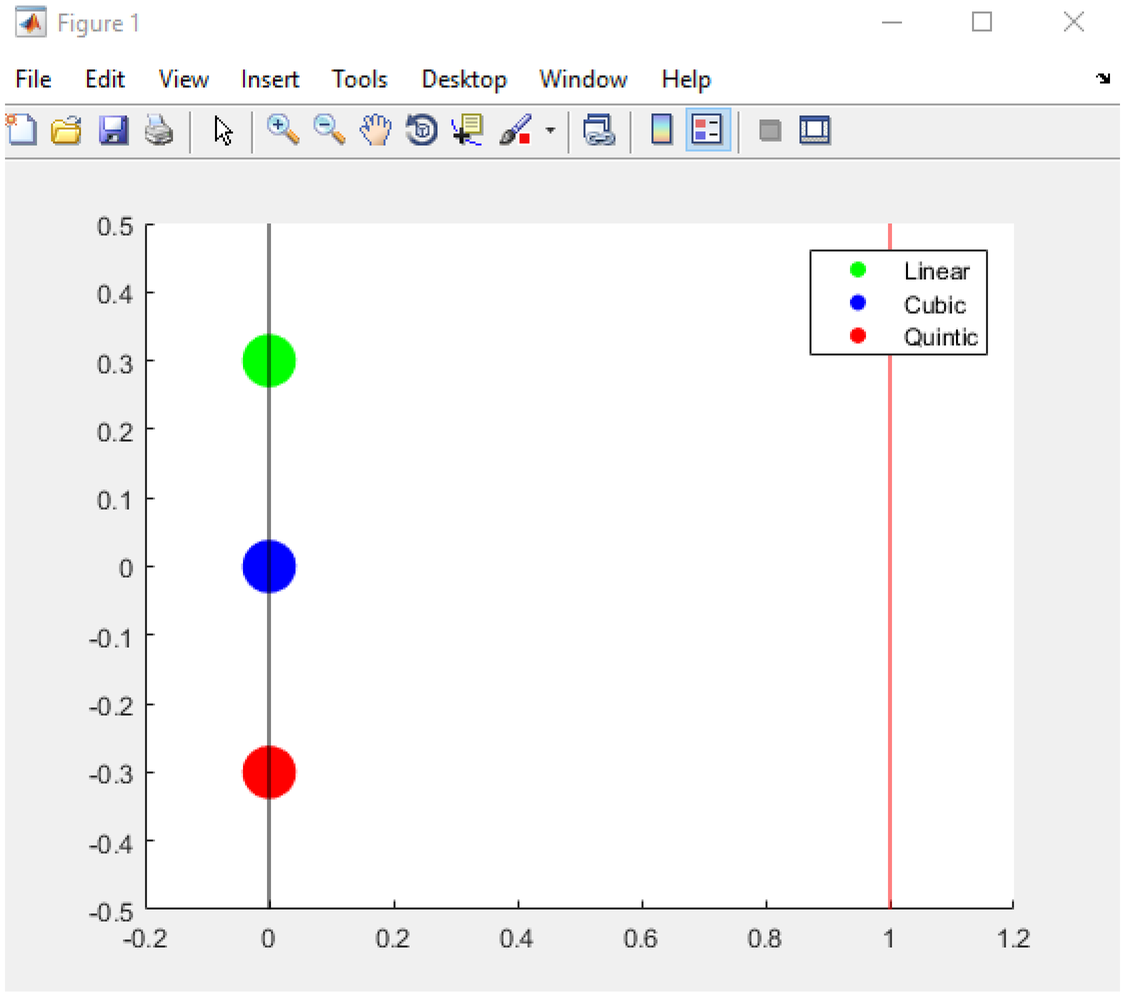


Figure 1
File Edit View Insert Tools Desktop Window Help
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
Linear
Cubic
Quintic

0270-6474/85/0509-2318$02.00/0

Printed in U.S.A.

The Journal of Neuroscience
Vol. 5, No. 9, pp. 2318–2330
September 1985

Kinematic Features of Unrestrained Vertical Arm Movements[1]

CHRISTOPHER G. ATKESON AND JOHN M. HOLLERBACH[2]

Artificial Intelligence Laboratory and Department of Psychology, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Abstract

Unrestrained human arm trajectories between point targets have been investigated using a three-dimensional tracking apparatus, the Selspot system. Movements were executed between different points in a vertical plane under varying conditions of speed and hand-held load. In contrast to past results which emphasized the straightness of hand paths, movement regions were discovered in which the hand paths were curved. All movements, whether curved or straight, showed an invariant tangential velocity profile when normalized for speed and distance. The velocity profile invariance with speed and load is interpreted in terms of simplification of the underlying arm dynamics, extending the results of Hollerbach and Flash (Hollerbach, J. M., and T. Flash (1982) Biol. Cybern. *44:* 67–77).

We have investigated unrestrained human arm trajectories between point targets using a three-dimensional tracking apparatus, the Selspot system. Our studies indicate the importance of examining natural unrestricted movements, as our results agree only in part with previous studies of arm movement. Past observations on multijoint human arm trajectories obtained from restricted horizontal planar movements measured with a gripped pantograph have shown in both humans and monkeys that point-to-point trajectories are essentially straight with bell-shaped velocity profiles (Morasso, 1981; Abend et al., 1982). Moreover, they satisfy a time scaling property that may be related to the underlying dynamics (Hollerbach and Flash, 1982). We sought to corroborate these observations for more natural unrestricted arm movements and also to examine the effects of different loads and of gravity on the arm trajectories. Our research on load effects has also led to the discovery of scaling laws for arm loads.

Path shape. A strategy for gaining insight into planning and control processes of the motor system is to look for kinematic invariances in trajectories of movement. The significance of straight-line movements of the hand during arm trajectories is that they imply movement planning at the hand or object level (Morasso, 1981; Hollerbach, 1982), that is to say, in terms of coordinates or variables that are external to the biological system and that could be matched to tasks or outside constraints.

If movements were planned in terms of joint variables, one would expect curved hand paths. The observed straight-line hand paths would seem to preclude this possibility (Morasso, 1981), yet in a series of papers examining unrestrained vertical arm movement (Soechting and Lacquaniti, 1981; Lacquaniti and Soechting, 1982; Lacquaniti et al., 1982), the hand trajectories were evidently straight at the same time that the joint rate ratio of shoulder and elbow tended toward a constant. This apparently contradictory situation of straight lines in both hand space and joint space has nevertheless been resolved recently in favor of hand space straight lines due to an artifact of two-joint kinematics near the workspace boundary (Hollerbach and Atkeson, 1984).

When hand movements are curved in response to task requirements or to internal control, it is not as clear what the planning variables are. For handwriting movements, Hollerbach (1981) proposed orthogonal task coordinates in the writing plane that yielded cursive script through coupled oscillation and modulation. Viviani and Terzuolo (1982) proposed hand variable planning for drawing as well as writing through proportional control of tangential velocity and radius of curvature. Morasso (1983) examined three-dimensional curved motion and proposed independent control of the curvature and torsion of the hand cartesian coordinates. Again arguing for joint-level planning but also for actuator-level planning, Soechting and Lacquaniti (1983) investigated curved movements resulting from change of target location during two-joint arm movement, and inferred both a linear relation between elbow and shoulder accelerations and stereotypical muscle electromyogram activity.

Time profile. In addition to the path of the arm, the other aspect of a trajectory is the time sequence along the path. This tangential velocity profile may through its shape also give insight into movement planning strategies. For motions under low spatiotemporal accuracy constraints, a common observation is a symmetrical and unimodal velocity profile. Crossman and Goodeve (1983) characterized these profiles as Gaussian for two different single degree of freedom movements: a pen-tapping movement constrained by a measurement wire and wrist rotation about the forearm axis. More recently, Hogan (1984) modeled the velocity profiles for single-joint elbow movement as fourth-order polynomials derived from a minimum-jerk cost function. In examining optimization criteria for single-joint movement, Nelson (1983) deduced that a minimum-jerk velocity profile is almost indistinguishable from simple harmonic motion for repetitive movement. Stein et al. (1985) modeled muscle activation and energetics for a single degree of freedom point-to-point movement, and showed that muscle force rise time or minimum energy yields a velocity profile very similar to minimum jerk.

The previous experiments involved single degree of freedom movement, either with one joint or an apparatus with one degree of freedom, for which the only independent parameter is the time dependence. Nevertheless, similar results have been found for multi-

Received June 26, 1984; Revised February 4, 1985; Accepted March 13, 1985

[1] This paper describes research done at the Department of Psychology and at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for this research was provided by National Institutes of Health Research Grant AM 26710, awarded by the National Institute of Arthritis, Metabolism, and Digestive Diseases, and by a National Science Foundation graduate fellowship (C. G. A.). We also acknowledge early contributions of Michael Propp and Jonathan Delatizky toward development of our Selspot system, and of Eric Saund for display software development.

[2] To whom correspondence should be addressed.

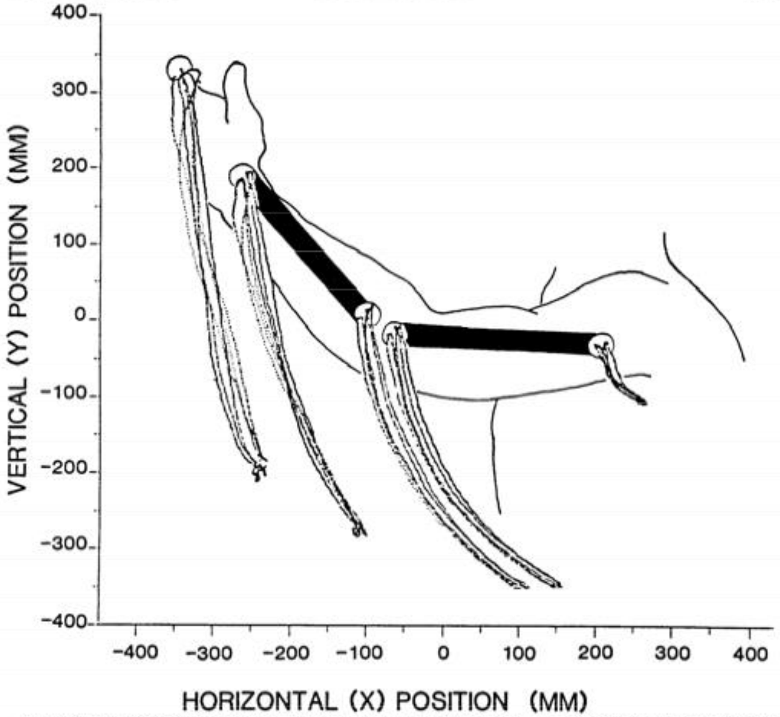


Figure 2. Attachment of Selspot markers and data presentation. Locations of the Selspot infrared LED markers and the typical format of the data presentation are shown. Note that the wrist and one of the elbow LEDs are connected by a rigid bar aligned with the forearm, and the shoulder and other elbow LEDs are similarly connected on a rigid bar aligned with the upper arm. The three-dimensional Selspot data are projected onto the *XY* plane. This projection shows most of the features of the path because these movements were almost planar (for the finger, wrist, and shoulder) and oriented parallel to the *XY* plane. In each data plot several movements are presented. Three upward movements (*dotted lines*) are indicated here by a *dot* at the location of the infrared LED for each sample (sampling frequency 315 Hz). Three downward movements are indicated by *solid lines* marking the path of each infrared LED.

camera and recording the average measured positions. Deviations between expected and actual measurements were calculated and mapped into a 25 × 25 correction table. Standard interpolation techniques were used to calculate the table originally and to read corrections from the table.

Three-dimensional positions of the LEDs were calculated from the corrected data using the known positions and orientations of the cameras and geometry. Points were marked as bad for which the vectors to the reconstructed LED position from each camera origin missed by greater than a certain threshold (3 cm), since with four parameters from the two cameras there is one redundant measurement. The Selspot system in our configuration can detect movements of the markers as small as 1 mm. Currently, the absolute accuracy of the system is within ±1 cm.

Normalization of tangential velocity profiles. To check invariance of tangential velocity profile shape, movements must be normalized for time and distance. Define $v(t)$ as the experimental tangential velocity profile as a function of time t, v_{max} as the maximum tangential velocity in $v(t)$, d as the experimental movement distance, v_{ref} as the reference velocity, and d_{ref} as the reference distance. The reference velocity and distance are chosen arbitrarily, and all data records are scaled to them. Since the tangential velocity profiles $v(t)$ are almost always unimodal, the maximum tangential velocity v_{max} is well defined. We use v_{max} rather than movement duration because of imprecision in determining movement start and stop points.

Now define time and distance scaling factors c and a as

$$c = \frac{v_{ref}}{v_{max}}, \qquad a = \frac{d_{ref}}{d}$$

The velocity profile $v'(t)$ normalized first for distance is $v'(t) = av(t)$. The maximum velocity for the new velocity profile is then $v'_{max} = av_{max}$. Define a new time scaling factor $c' = v_{ref}/v'_{max} = c/a$. Then the time-normalized velocity profile $v''(t)$ is

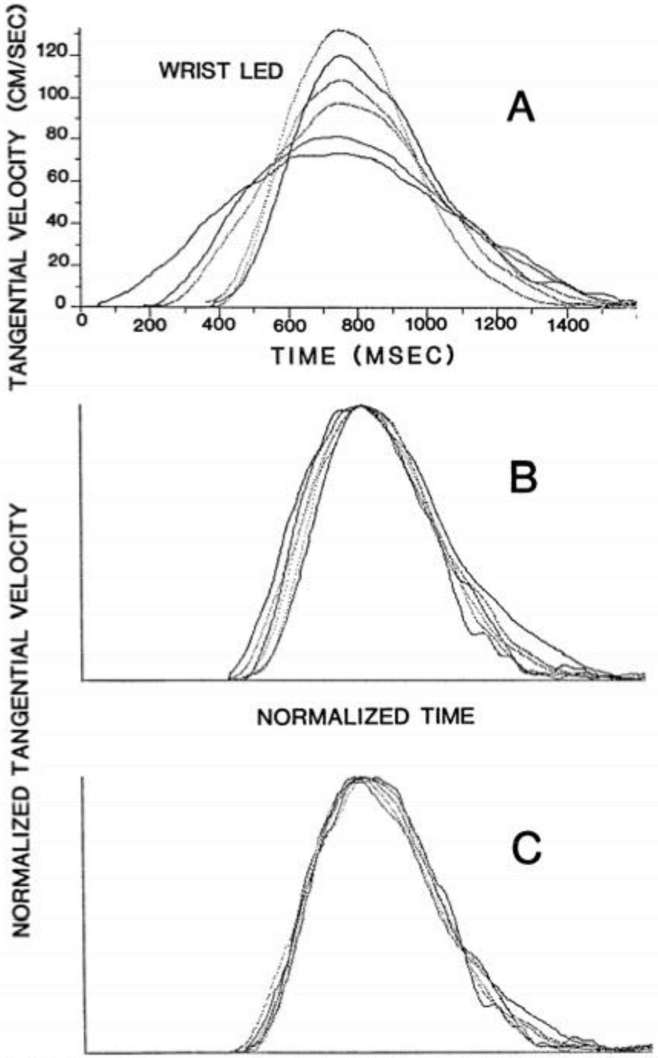


Figure 3. Normalization of tangential velocity profiles. *A*, A set of typical tangential velocity profiles of the wrist LED, derived from six unloaded movements at slow speeds between targets 3 and 7. *B*, Normalized tangential velocity profiles indicating substantial overlap. The units for the normalized tangential velocity profile have no physical meaning and are therefore not indicated. *C*, Realigned profiles through minimization of the similarity measure *w* with respect to the averaged velocity profile.

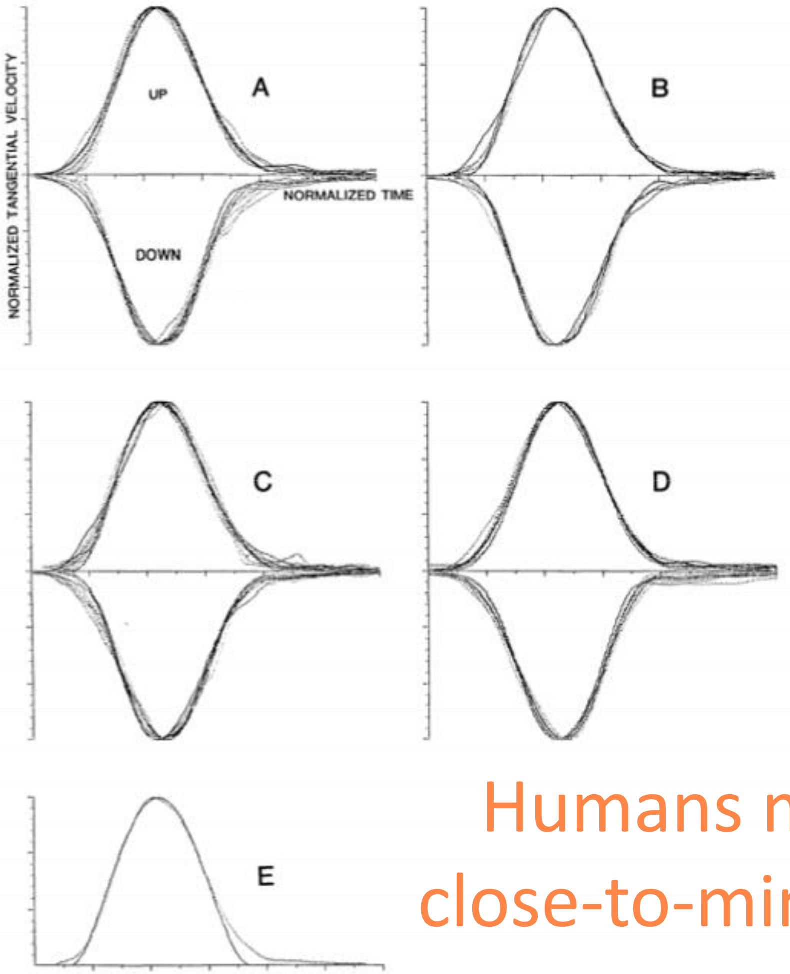


Figure 7. Tangential velocity profile shape invariance across different conditions. *A*, Different speeds. One subject's unloaded movements between targets 3 and 7 at slow, medium, and fast speeds are shown. Slow and fast movements are *dotted lines* here, whereas medium-speed movements are presented as *solid lines*. *B*, Different loads. One subject's medium-speed movements between targets 3 and 7 were executed unloaded (*solid lines*) and loaded by a 4-lb. hand-held load (*dotted lines*). *C*, Different movements. One subject's medium-speed unloaded movements between all target pairs are shown. Movements between targets 3 and 7 are shown as *solid lines*, and those between targets 1-5, 2-6, and 4-8 are shown as *dotted lines*. *D*, Different subjects. Fast speed unloaded movements between targets 3 and 7 from three subjects. Movements of the same subject as presented in *A-C* are shown as *solid lines*; movements of the other two subjects are shown as *dotted lines*. *E*, Comparison of reference profile (*dotted lines*) to minimum-jerk profile (*solid line*).

Humans move with close-to-minimum jerk.

What other kinds of trajectories can you think of?

Specifying Constant Velocity for Central Portion

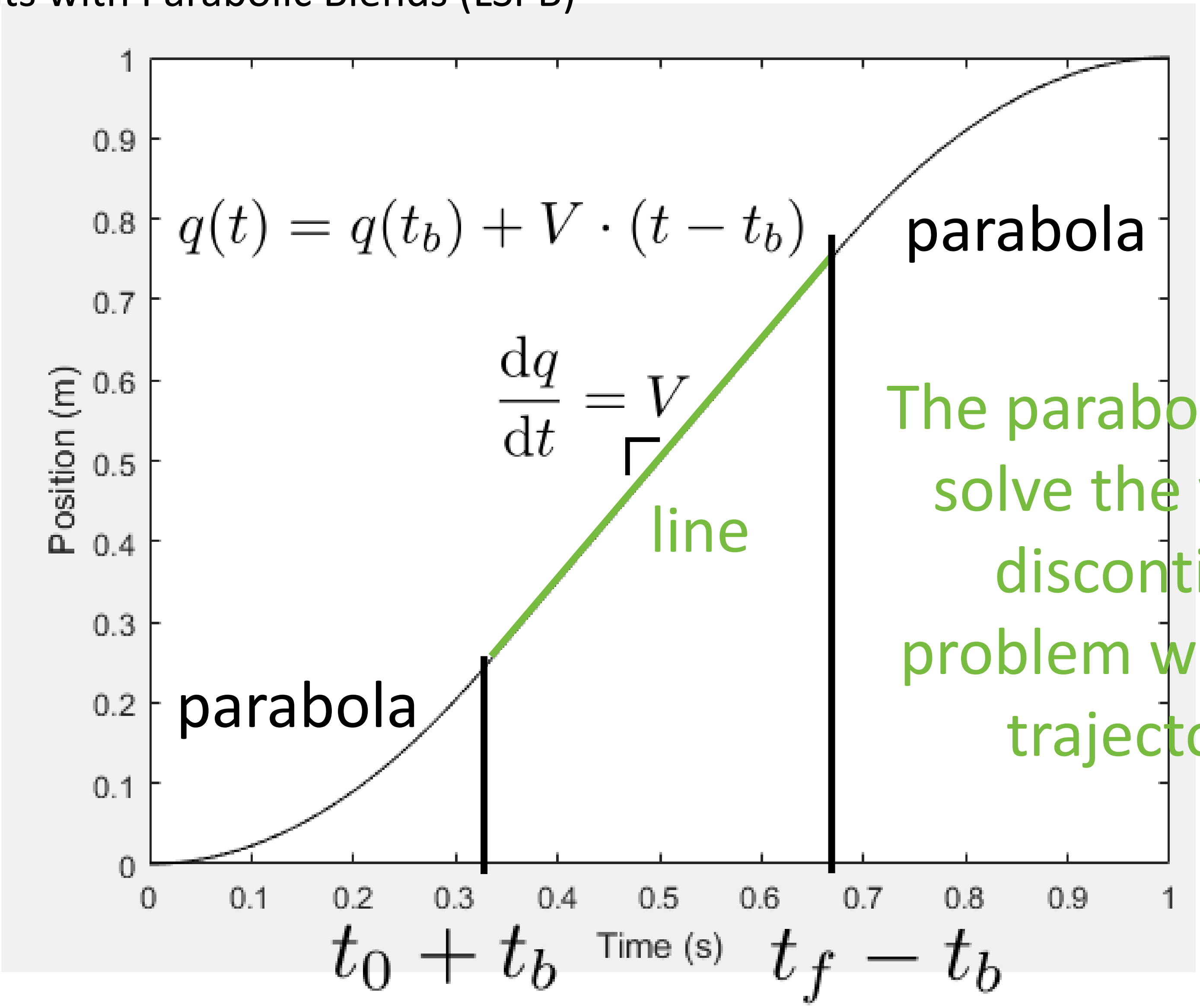
Linear Segments with Parabolic Blends (LSPB)

Ramp up velocity to desired value for a short time at start.
Move at constant velocity for a while.
Ramp down velocity to final value for a short time at end.

Start and end **blend times** are usually equal.

Specifying Constant Velocity for Central Portion

Linear Segments with Parabolic Blends (LSPB)

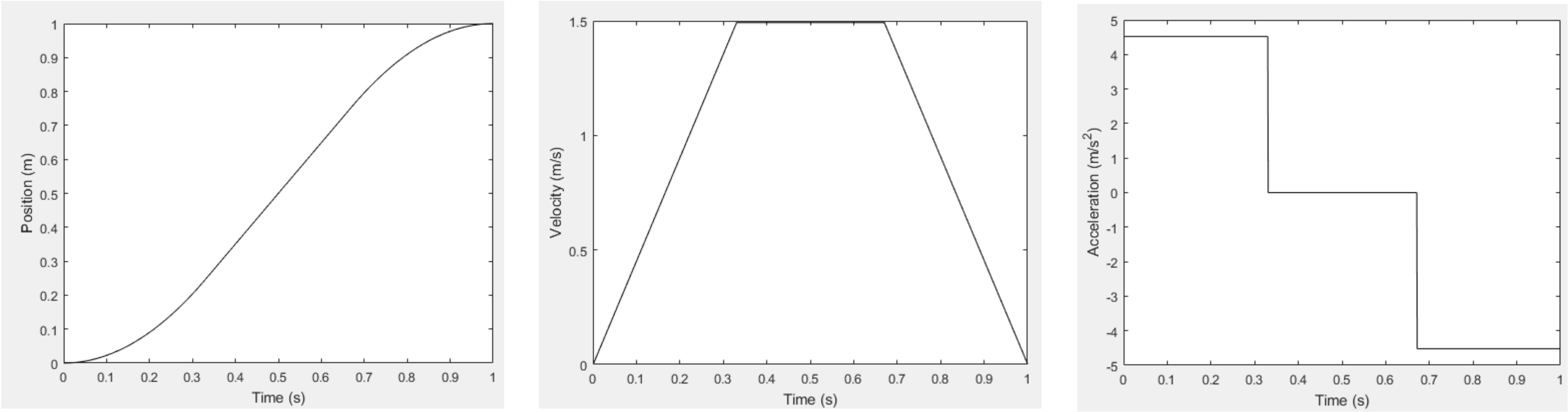


The parabolic blends solve the velocity discontinuity problem with linear trajectories.

Specifying Constant Velocity for Central Portion

Linear Segments with Parabolic Blends (LSPB)

Piecewise constant accelerations



Trapezoidal velocity profile

Limits

$$0 < t_b \leq \frac{t_f}{2}$$

$$\frac{q_f - q_0}{t_f} < V \leq \frac{2(q_f - q_0)}{t_f}$$

Not minimum jerk...

Getting There As Fast As Possible

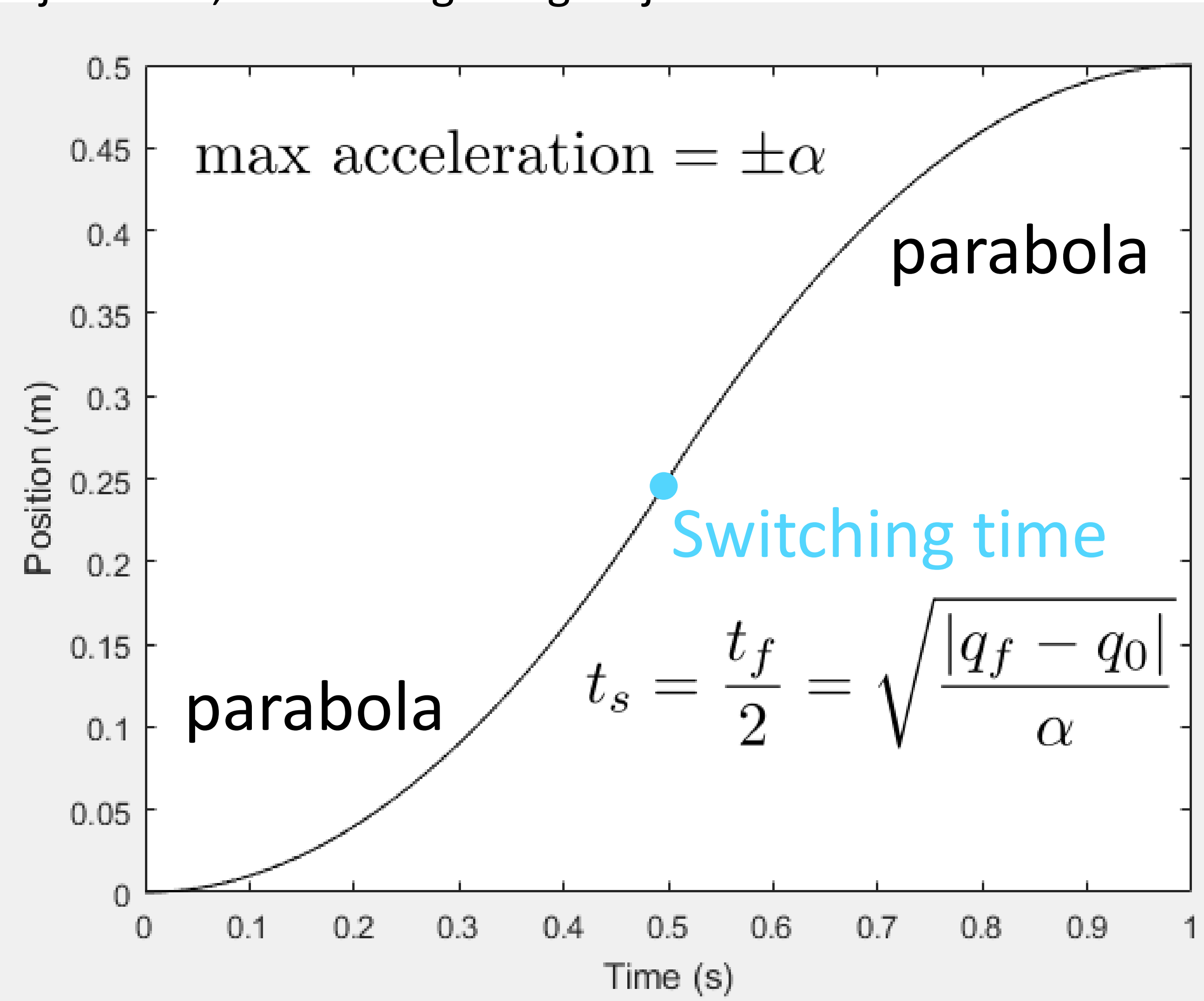
Minimum Time Trajectories, a.k.a. Bang-Bang Trajectories

Leave final time unspecified.
Specify the **maximum acceleration** possible, typically set by actuator limit.
Apply maximum acceleration in one direction, then abruptly **switch** to negative maximum acceleration.

Typically starting and ending at rest.
Switching time is halfway through the trajectory.

Getting There As Fast As Possible

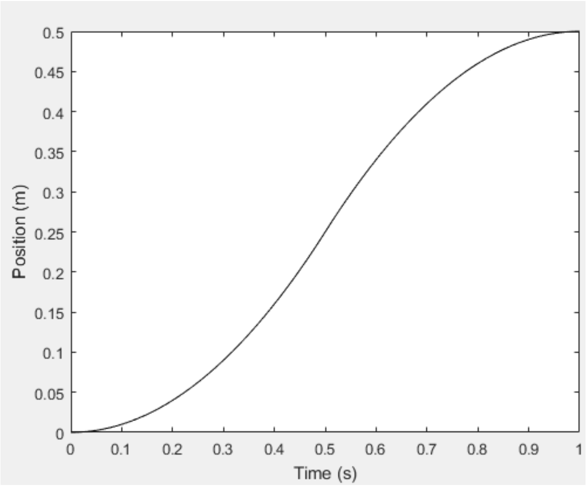
Minimum Time Trajectories, a.k.a. Bang-Bang Trajectories

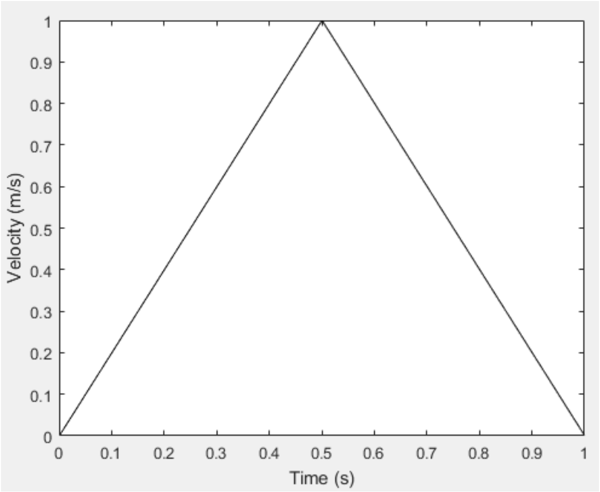


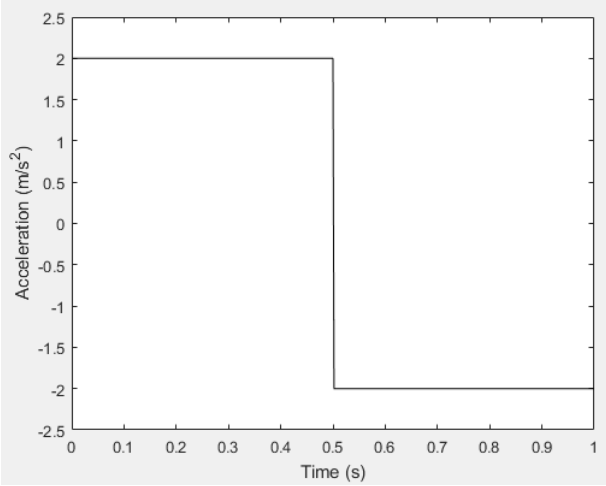
Getting There As Fast As Possible

Minimum Time Trajectories, a.k.a. Bang-Bang Trajectories

Piecewise constant max accelerations

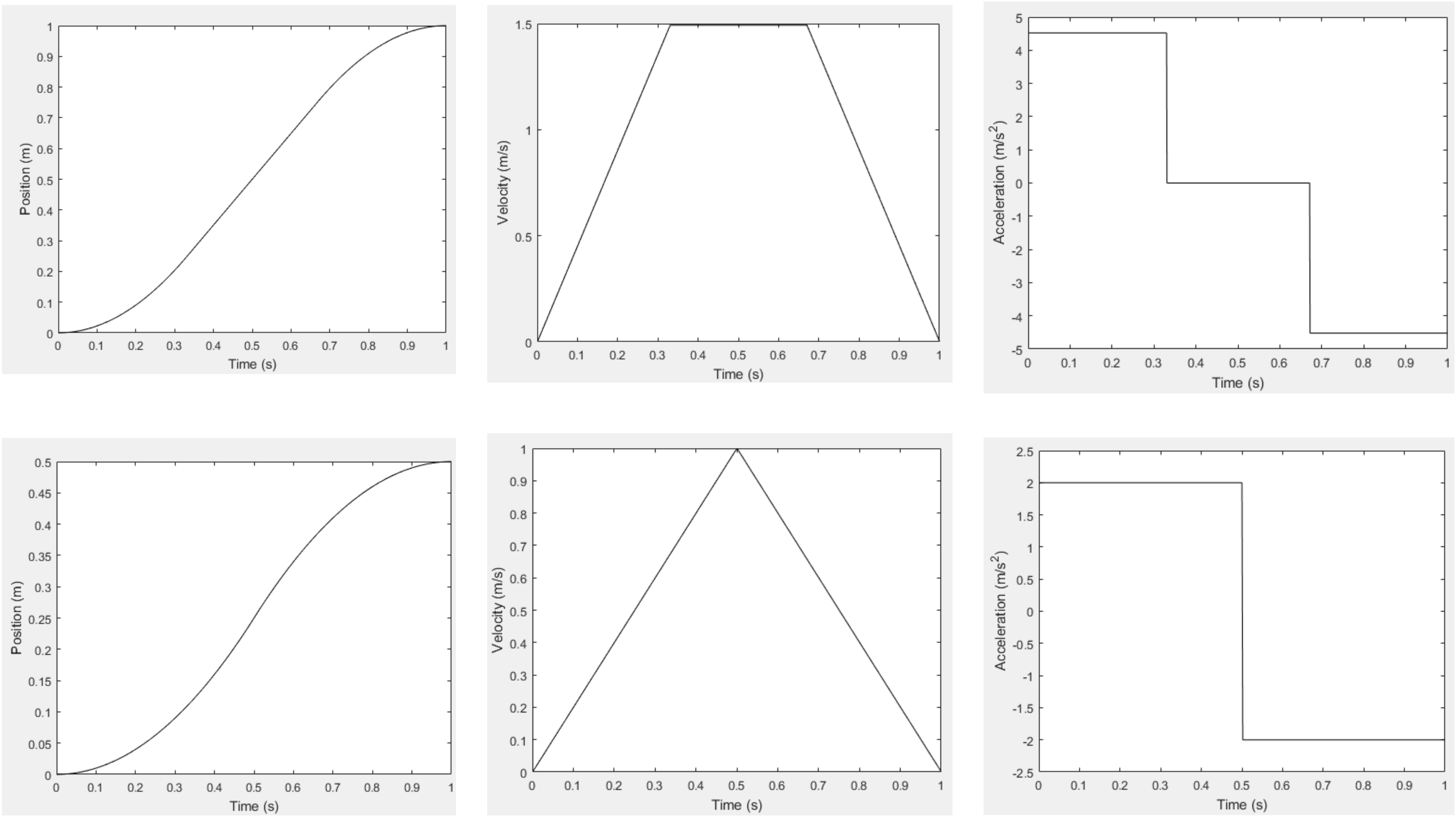






Triangular velocity profile

Not minimum jerk...
...but fast!



Position (m)
Time (s)
Velocity (m/s)
Time (s)
Acceleration (m/s^2)
Time (s)
Position (m)
Time (s)
Velocity (m/s)
Time (s)
Acceleration (m/s^2)
Time (s)

Moving through Waypoints

You could solve for a **single** high-order polynomial that hits all your waypoints.

This approach yields a nice **continuously differentiable** curve.

However, it is intractable when many waypoints are used because the linear system's dimension become very large.

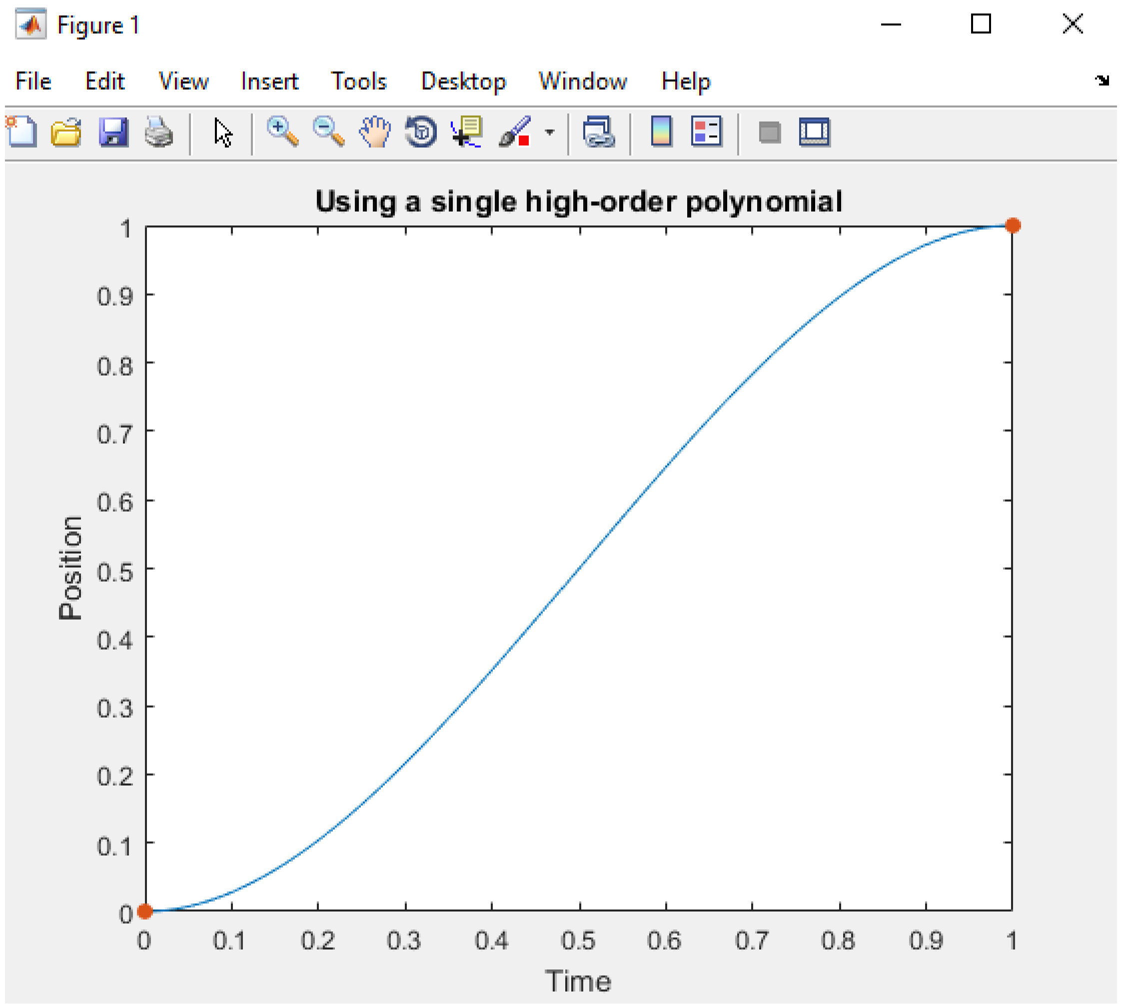


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
Using a single high-order polynomial
Position
Time
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

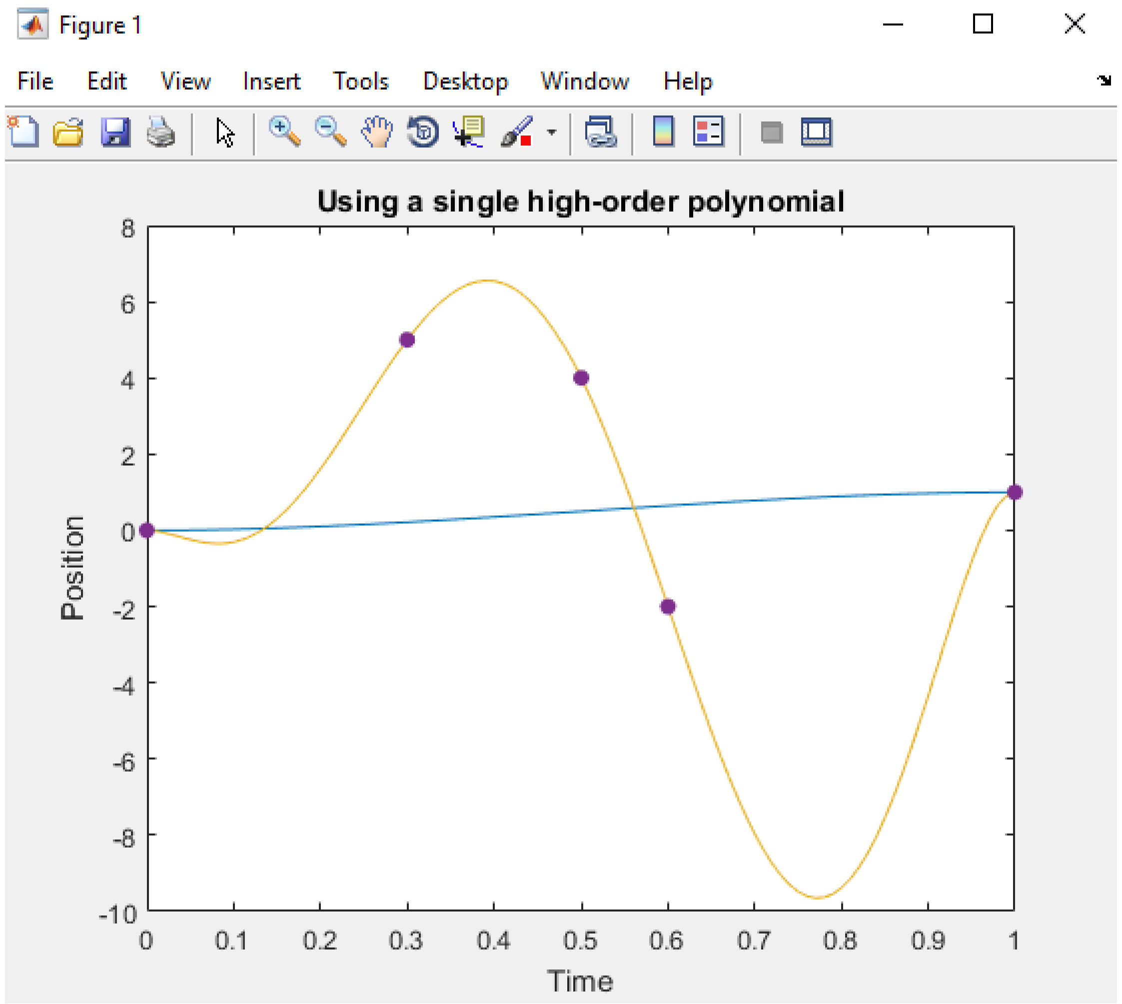


Figure 1
File Edit View Insert Tools Desktop Window Help
Using a single high-order polynomial
Position
Time
8
6
4
2
0
-2
-4
-6
-8
-10
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

Moving through Waypoints

Instead, use **low-order polynomials** for trajectory segments between adjacent via points.

Ensure that position, velocity, and acceleration constraints are satisfied at the via points, where we switch from one polynomial to the next.

Final conditions for one polynomial become the initial conditions for the next!

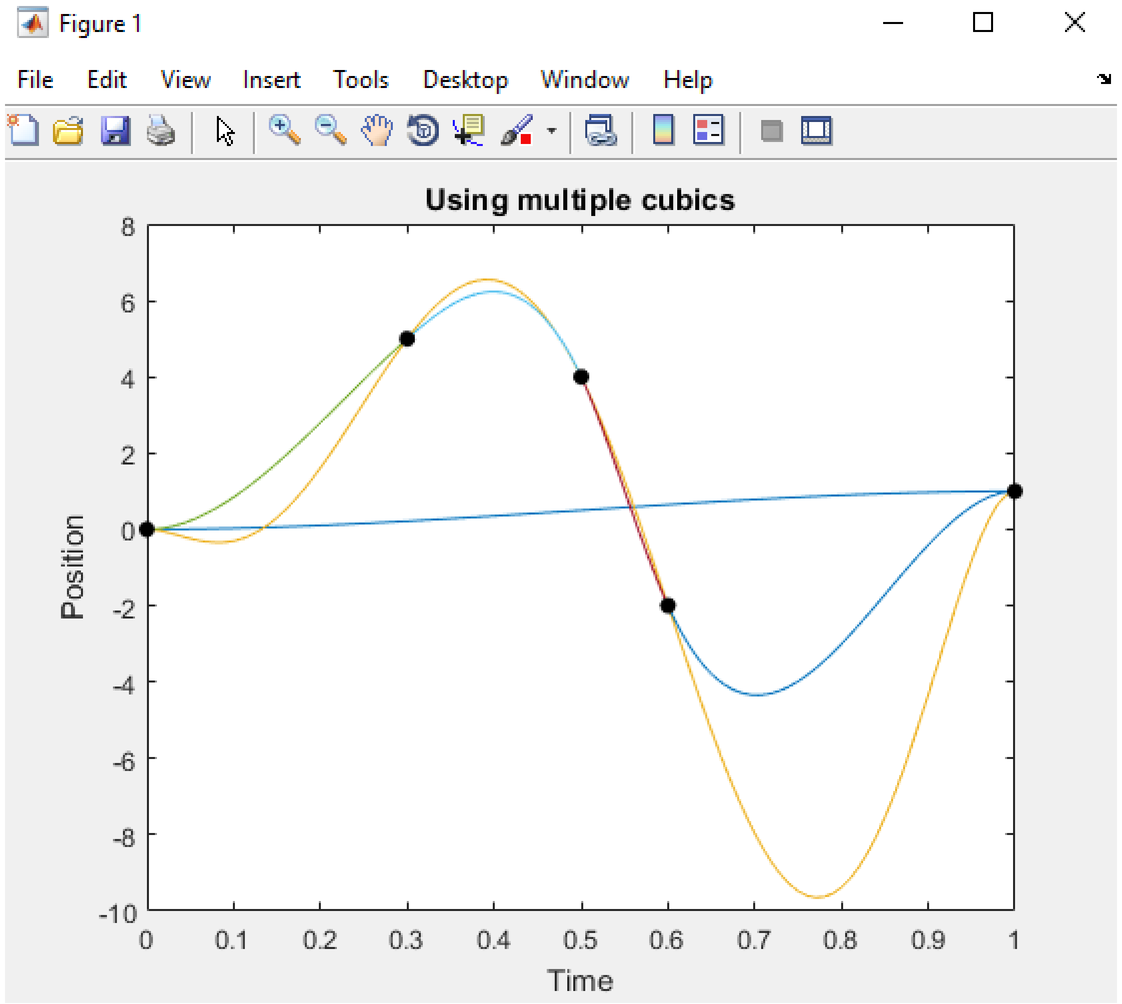


Figure 1
File Edit View Insert Tools Desktop Window Help
Using multiple cubics
Position
Time
8
6
4
2
0
-2
-4
-6
-8
-10
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

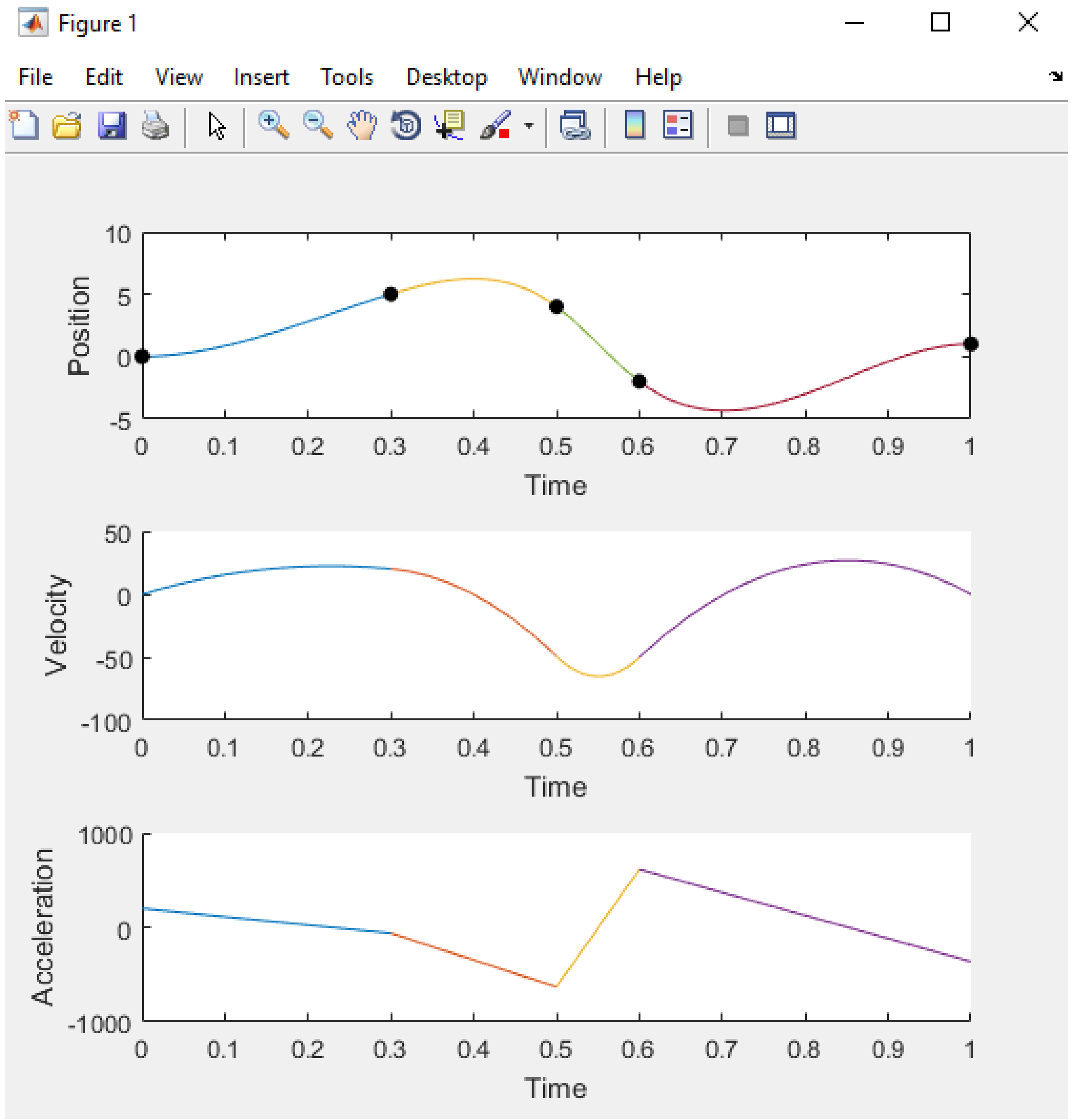


Figure 1
File Edit View Insert Tools Desktop Window Help
Position
10
5
0
-5
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Time
Velocity
50
0
-50
-100
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Time
Acceleration
1000
0
-1000
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Time

For which of the following five trajectory types can q leave the interval between q_0 and q_f for the time span $t_0 \leq t \leq t_f$?

First-Order Polynomial (Line) Does not leave interval.

$$q(t) = a_0 + a_1 t$$

Third-Order Polynomial (Cubic) Could leave interval*.

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

*Depends on initial and final velocities. When both are zero, does not leave interval.

Fifth-Order Polynomial (Quintic) Could leave interval**.

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

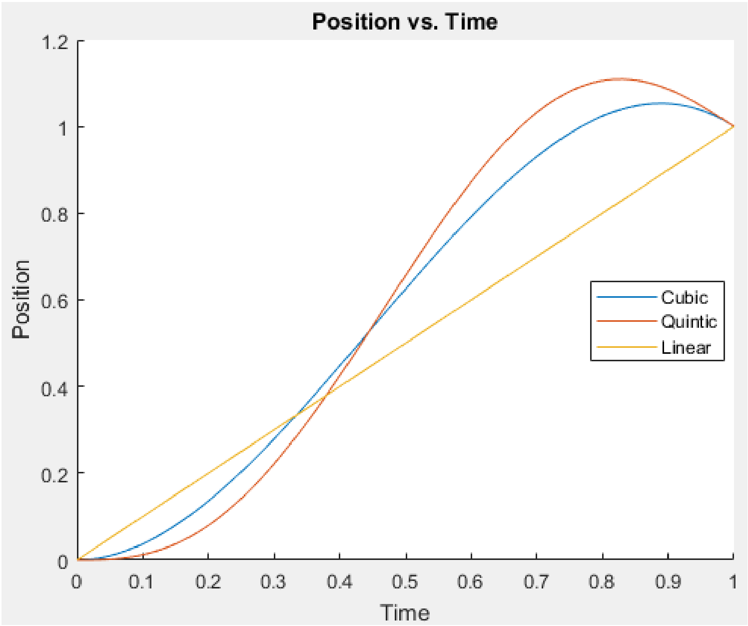
**Depends on initial and final velocities and accelerations. When all are zero, does not leave interval.

Linear Segment with Parabolic Blends (LSPB, 1 Line + 2 Quadratics) Could leave interval*.

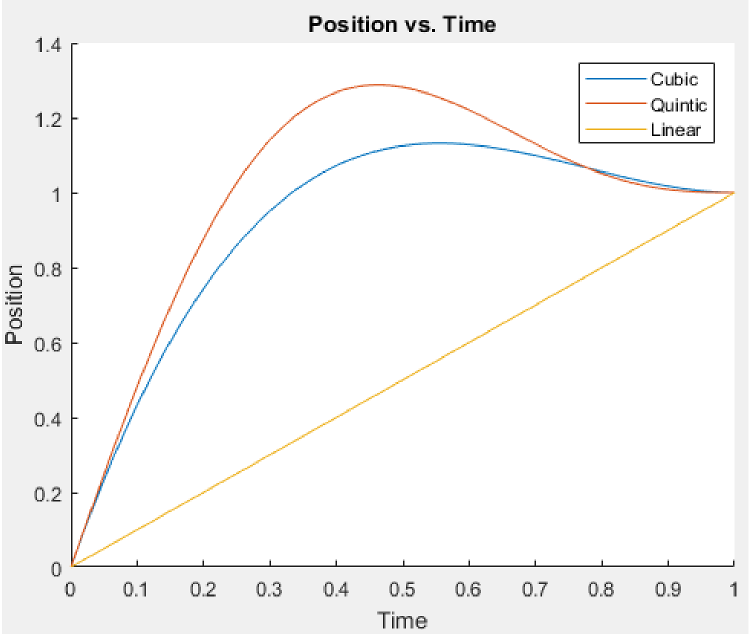
$$q(t) = b_0 + b_1 t + b_2 t^2 \quad q(t) = a_0 + a_1 t \quad q(t) = c_0 + c_1 t + c_2 t^2$$

Minimum Time Trajectory (Bang-Bang, 2 Quadratics) Could leave interval*.

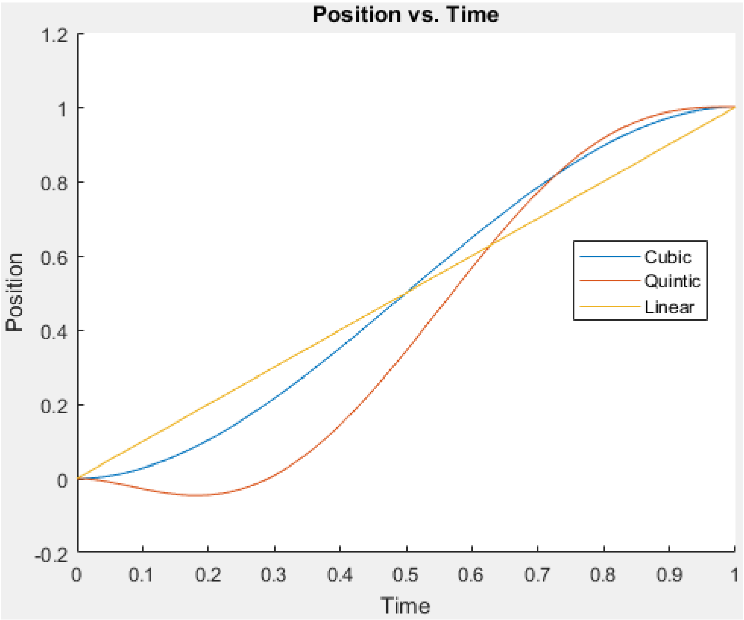
$$q(t) = b_0 + b_1 t + b_2 t^2 \quad q(t) = c_0 + c_1 t + c_2 t^2$$



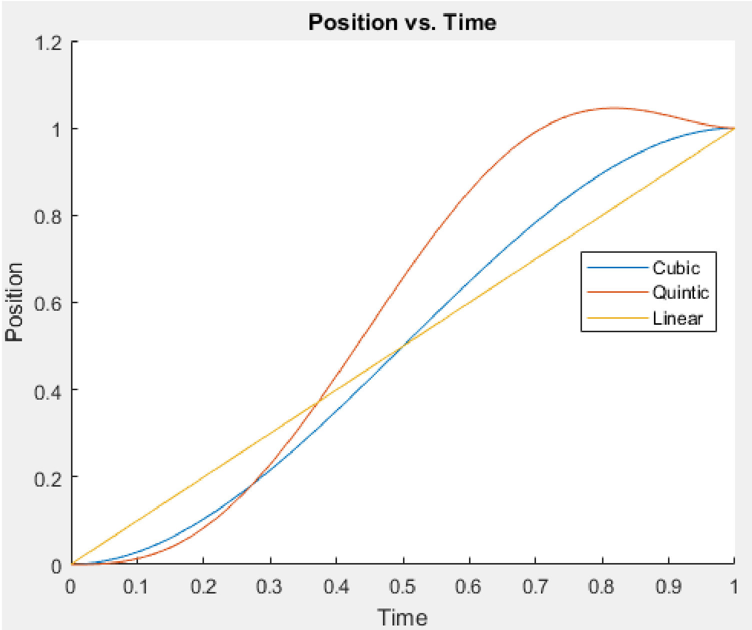
Final velocity less than zero



Initial velocity greater than zero and large



Initial acceleration less than zero



Final acceleration greater than zero

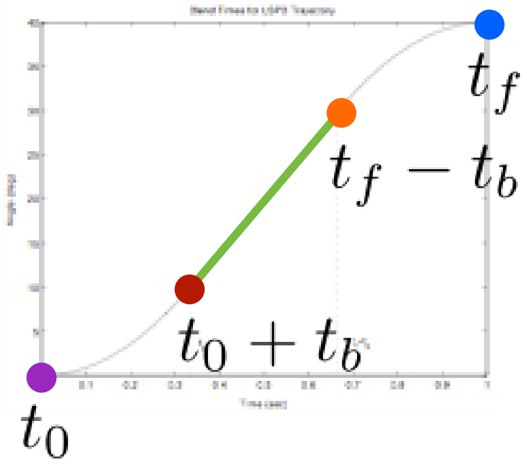
Why would one ever use a line or a cubic polynomial instead of a quintic polynomial?

- Want constant velocity (line).
- Your robot is sufficiently rigid, so you don’t care about minimal jerk.
- Need lower computational complexity, e.g., real-time calculations on a microcontroller.
- Need lower memory usage, e.g., implementation on a microcontroller.
- Want to limit maximum speed.
- More ideas from class?

Set up the equations to solve for all the coefficients of a general LSPB given initial time t_0, final time t_f, initial position q_0, final position q_f, initial velocity v_0, final velocity v_f, and blend duration t_b.

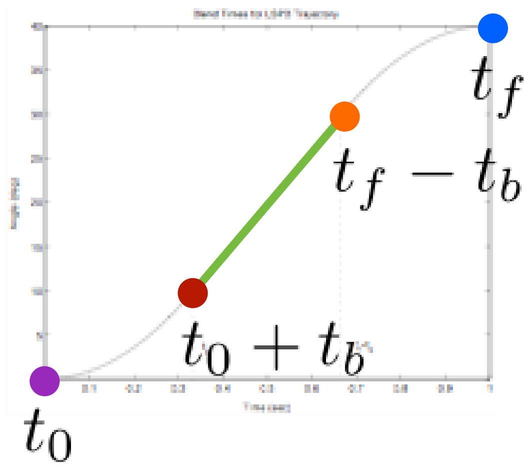
$$q(t) = b_0 + b_1 t + b_2 t^2 \qquad q(t) = a_0 + a_1 t \qquad q(t) = c_0 + c_1 t + c_2 t^2$$
$$\dot{q}(t) = b_1 + 2b_2 t \qquad \dot{q}(t) = a_1 \qquad \dot{q}(t) = c_1 + 2c_2 t$$

8 parameters – need 8 equations



Position and velocity at four points in time

$$q_0 = b_0 + b_1 t_0 + b_2 t_0^2$$
$$v_0 = b_1 + 2b_2 t_0$$
$$b_0 + b_1(t_0 + t_b) + b_2(t_0 + t_b)^2 = a_0 + a_1(t_0 + t_b)$$
$$b_1 + 2b_2(t_0 + t_b) = a_1$$
$$\cdots$$



$$q_0 = b_0 + b_1 t_0 + b_2 t_0^2$$

$$v_0 = b_1 + 2b_2 t_0$$

$$b_0 + b_1(t_0 + t_b) + b_2(t_0 + t_b)^2 = a_0 + a_1(t_0 + t_b)$$

$$b_1 + 2b_2(t_0 + t_b) = a_1$$

$$a_0 + a_1(t_f - t_b) = c_0 + c_1(t_f - t_b) + c_2(t_f - t_b)^2$$

$$a_1 = c_1 + 2c_2(t_f - t_b)$$

$$q_f = c_0 + c_1 t_f + c_2 t_f^2$$

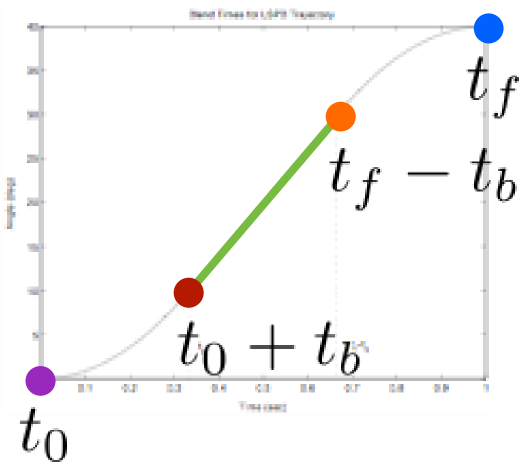
$$v_f = c_1 + 2c_2 t_f$$

time matrix

conditions

unknown coefficients

$$b = Ax$$



$$q_0 = b_0 + b_1 t_0 + b_2 t_0^2$$

$$v_0 = b_1 + 2 b_2 t_0$$

$$b_0 + b_1(t_0 + t_b) + b_2(t_0 + t_b)^2 = a_0 + a_1(t_0 + t_b)$$

$$b_1 + 2b_2(t_0 + t_b) = a_1$$

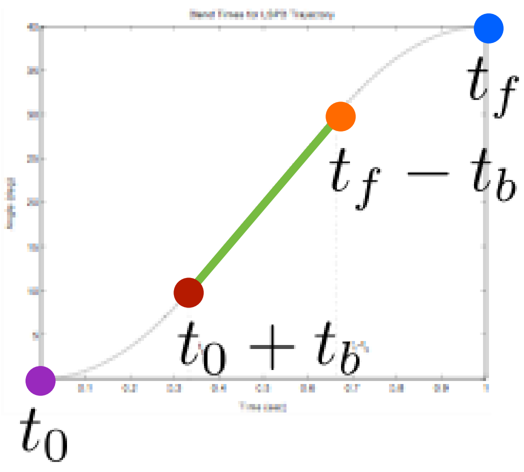
$$a_0 + a_1(t_f - t_b) = c_0 + c_1(t_f - t_b) + c_2(t_f - t_b)^2$$

$$a_1 = c_1 + 2c_2(t_f - t_b)$$

$$q_f = c_0 + c_1 t_f + c_2 t_f^2$$

$$v_f = c_1 + 2c_2 t_f$$

$$\begin{bmatrix} ? \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? & ? & ? & ? & ? \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ a_0 \\ a_1 \\ c_0 \\ c_1 \\ c_2 \end{bmatrix}$$



$$q_0 = b_0 + b_1 t_0 + b_2 t_0^2$$

$$v_0 = b_1 + 2b_2 t_0$$

$$b_0 + b_1(t_0 + t_b) + b_2(t_0 + t_b)^2 = a_0 + a_1(t_0 + t_b)$$

$$b_1 + 2b_2(t_0 + t_b) = a_1$$

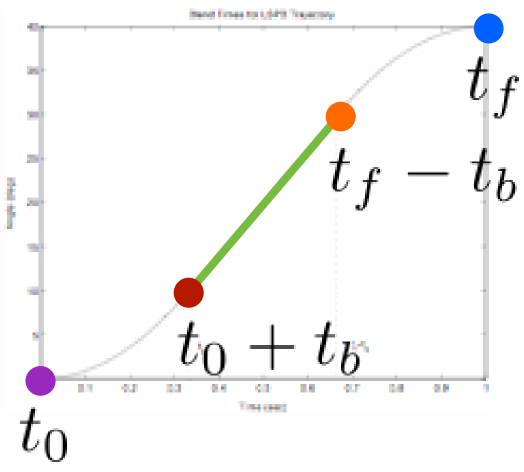
$$a_0 + a_1(t_f - t_b) = c_0 + c_1(t_f - t_b) + c_2(t_f - t_b)^2$$

$$a_1 = c_1 + 2c_2(t_f - t_b)$$

$$q_f = c_0 + c_1 t_f + c_2 t_f^2$$

$$v_f = c_1 + 2c_2 t_f$$

$$\begin{bmatrix} q_0 \\ ? \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & 0 & 0 & 0 & 0 & 0 \\ ? & ? & ? & ? & ? & ? & ? & ? \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ a_0 \\ a_1 \\ c_0 \\ c_1 \\ c_2 \end{bmatrix}$$



$$q_0 = b_0 + b_1 t_0 + b_2 t_0^2$$

$$v_0 = b_1 + 2b_2 t_0$$

$$b_0 + b_1(t_0 + t_b) + b_2(t_0 + t_b)^2 = a_0 + a_1(t_0 + t_b)$$

$$b_1 + 2b_2(t_0 + t_b) = a_1$$

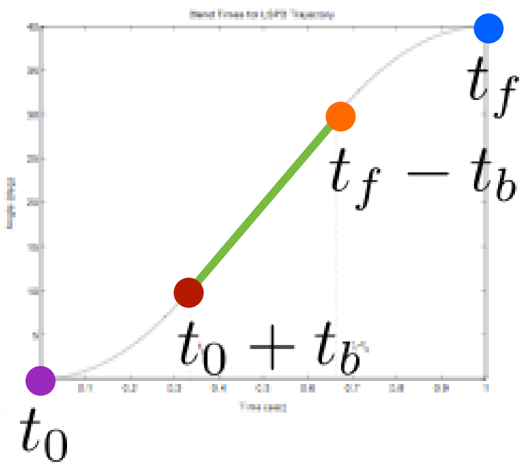
$$a_0 + a_1\left(t_f - t_b\right) = c_0 + c_1\left(t_f - t_b\right) + c_2\left(t_f - t_b\right)^2$$

$$a_1 = c_1 + 2c_2\left(t_f - t_b\right)$$

$$q_f = c_0 + c_1 t_f + c_2 t_f^2$$

$$v_f = c_1 + 2c_2 t_f$$

$$\begin{bmatrix} q_0 \\ v_0 \\ ? \\ \\ \\ \\ \\ \\ \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2t_0 & 0 & 0 & 0 & 0 & 0 \\ ? & ? & ? & ? & ? & ? & ? & ? \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ a_0 \\ a_1 \\ c_0 \\ c_1 \\ c_2 \end{bmatrix}$$



$$q_0 = b_0 + b_1 t_0 + b_2 t_0^2$$

$$v_0 = b_1 + 2b_2 t_0$$

$$b_0 + b_1(t_0 + t_b) + b_2(t_0 + t_b)^2 = a_0 + a_1(t_0 + t_b)$$

$$b_1 + 2b_2(t_0 + t_b) = a_1$$

$$a_0 + a_1(t_f - t_b) = c_0 + c_1(t_f - t_b) + c_2(t_f - t_b)^2$$

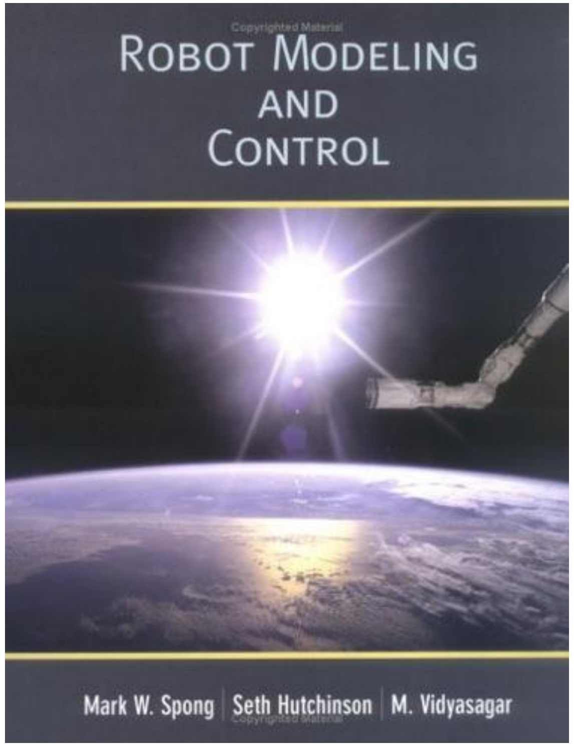
$$a_1 = c_1 + 2c_2(t_f - t_b)$$

$$q_f = c_0 + c_1 t_f + c_2 t_f^2$$

$$v_f = c_1 + 2c_2 t_f$$

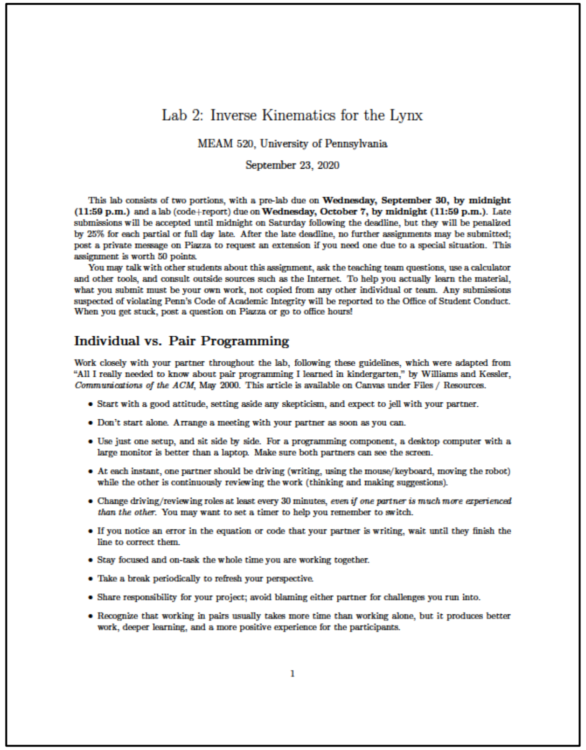
$$\begin{bmatrix} q_0 \\ v_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ q_f \\ v_f \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2t_0 & 0 & 0 & 0 & 0 & 0 \\ 1 & t_0 + t_b & (t_0 + t_b)^2 & -1 & -(t_0 + t_b) & 0 & 0 & 0 \\ 0 & 1 & 2(t_0 + t_b) & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & (t_f - t_b) & -1 & -(t_f - t_b) & -(t_f - t_b)^2 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & -2(t_f - t_b) \\ 0 & 0 & 0 & 0 & 0 & 1 & t_f & t_f^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2t_f \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ a_0 \\ a_1 \\ c_0 \\ c_1 \\ c_2 \end{bmatrix}$$

Next time: Trajectory Planning in Configuration Space!



Chapter 5: Path and Trajectory Planning

- Read 5.1, 5.5



Lab 2: Inverse Kinematics for the Lynx

MEAM 520, University of Pennsylvania

September 23, 2020

This lab consists of two portions, with a pre-lab due on **Wednesday, September 30, by midnight (11:59 p.m.)** and a lab (code+report) due on **Wednesday, October 7, by midnight (11:59 p.m.)**. Late submissions will be accepted until midnight on Saturday following the deadline, but they will be penalized by 25% for each partial or full day late. After the late deadline, no further assignments may be submitted; post a private message on Piazza to request an extension if you need one due to a special situation. This assignment is worth 50 points.

You may talk with other students about this assignment, ask the teaching team questions, use a calculator and other tools, and consult outside sources such as the Internet. To help you actually learn the material, what you submit must be your own work, not copied from any other individual or team. Any submissions suspected of violating Penn's Code of Academic Integrity will be reported to the Office of Student Conduct. When you get stuck, post a question on Piazza or go to office hours!

Individual vs. Pair Programming

Work closely with your partner throughout the lab, following these guidelines, which were adapted from "All I really needed to know about pair programming I learned in kindergarten," by Williams and Kessler, *Communications of the ACM*, May 2000. This article is available on Canvas under Files / Resources.

- Start with a good attitude, setting aside any skepticism, and expect to jell with your partner.
- Don't start alone. Arrange a meeting with your partner as soon as you can.
- Use just one setup, and sit side by side. For a programming component, a desktop computer with a large monitor is better than a laptop. Make sure both partners can see the screen.
- At each instant, one partner should be driving (writing, using the mouse/keyboard, moving the robot) while the other is continuously reviewing the work (thinking and making suggestions).
- Change driving/reviewing roles at least every 30 minutes, *even if one partner is much more experienced than the other*. You may want to set a timer to help you remember to switch.
- If you notice an error in the equation or code that your partner is writing, wait until they finish the line to correct them.
- Stay focused and on-task the whole time you are working together.
- Take a break periodically to refresh your perspective.
- Share responsibility for your project; avoid blaming either partner for challenges you run into.
- Recognize that working in pairs usually takes more time than working alone, but it produces better work, deeper learning, and a more positive experience for the participants.

1

Lab 2: Inverse Kinematics due 10/7