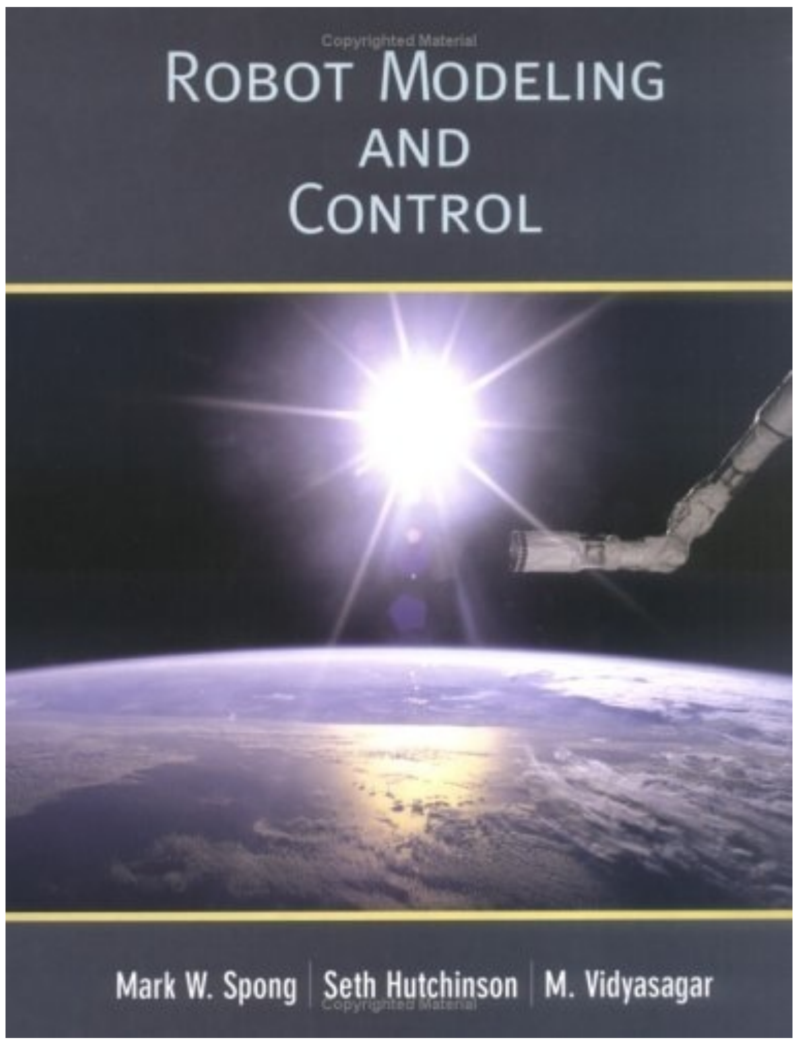
MEAM 520
Trajectory Planning

Katherine J. Kuchenbecker, Ph.D.
Mechanical Engineering and Applied Mechanics Department
SEAS, University of Pennsylvania

GRASP
LABORATORY

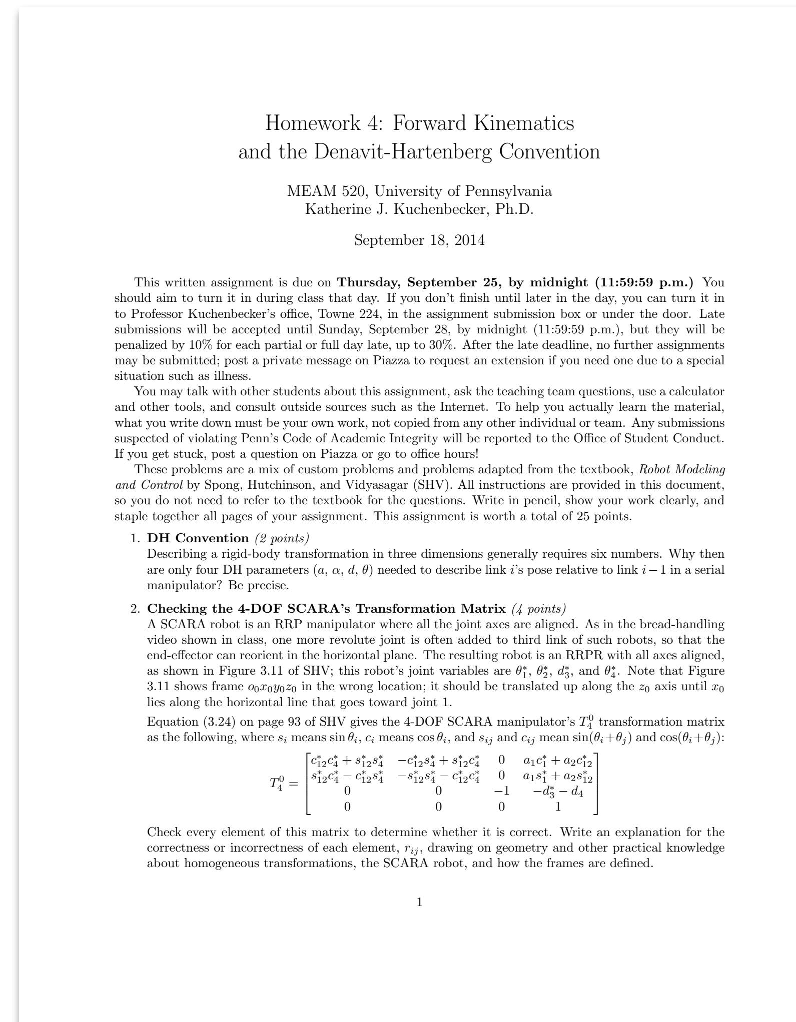
Lecture 9: September 25, 2014

Reading



This week, read Section 5.5 in SHV, which covers Trajectory Planning.

Homework



Homework 4: Forward Kinematics
and the Denavit-Hartenberg Convention

MEAM 520, University of Pennsylvania
Katherine J. Kuchenbecker, Ph.D.

September 18, 2014

This written assignment is due on **Thursday, September 25, by midnight (11:59:59 p.m.)** You should aim to turn it in during class that day. If you don't finish until later in the day, you can turn it in to Professor Kuchenbecker's office, Towne 224, in the assignment submission box or under the door. Late submissions will be accepted until Sunday, September 28, by midnight (11:59:59 p.m.), but they will be penalized by 10% for each partial or full day late, up to 30%. 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 such as illness.

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 write down 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. If you get stuck, post a question on Piazza or go to office hours!

These problems are a mix of custom problems and problems adapted from the textbook, *Robot Modeling and Control* by Spong, Hutchinson, and Vidyasagar (SHV). All instructions are provided in this document, so you do not need to refer to the textbook for the questions. Write in pencil, show your work clearly, and staple together all pages of your assignment. This assignment is worth a total of 25 points.

1. **DH Convention** *(2 points)*
   Describing a rigid-body transformation in three dimensions generally requires six numbers. Why then are only four DH parameters (a, α, d, θ) needed to describe link i's pose relative to link $i-1$ in a serial manipulator? Be precise.

2. **Checking the 4-DOF SCARA's Transformation Matrix** *(4 points)*
   A SCARA robot is an RRP manipulator where all the joint axes are aligned. As in the bread-handling video shown in class, one more revolute joint is often added to third link of such robots, so that the end-effector can reorient in the horizontal plane. The resulting robot is an RRPR with all axes aligned, as shown in Figure 3.11 of SHV; this robot's joint variables are θ_1^*, θ_2^*, d_3^*, and θ_4^*. Note that Figure 3.11 shows frame $o_0x_0y_0z_0$ in the wrong location; it should be translated up along the z_0 axis until x_0 lies along the horizontal line that goes toward joint 1.

   Equation (3.24) on page 93 of SHV gives the 4-DOF SCARA manipulator's T_4^0 transformation matrix as the following, where s_i means $\sin\theta_i$, c_i means $\cos\theta_i$, and s_{ij} and c_{ij} mean $\sin(\theta_i+\theta_j)$ and $\cos(\theta_i+\theta_j)$:

$$T_4^0 = \begin{bmatrix} c_{12}^*c_4^* + s_{12}^*s_4^* & -c_{12}^*s_4^* + s_{12}^*c_4^* & 0 & a_1c_1^* + a_2c_{12}^* \\ s_{12}^*c_4^* - c_{12}^*s_4^* & -s_{12}^*s_4^* - c_{12}^*c_4^* & 0 & a_1s_1^* + a_2s_{12}^* \\ 0 & 0 & -1 & -d_3^* - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

   Check every element of this matrix to determine whether it is correct. Write an explanation for the correctness or incorrectness of each element, r_{ij}, drawing on geometry and other practical knowledge about homogeneous transformations, the SCARA robot, and how the frames are defined.

1

Homework 4 is due by 11:59 p.m. today.

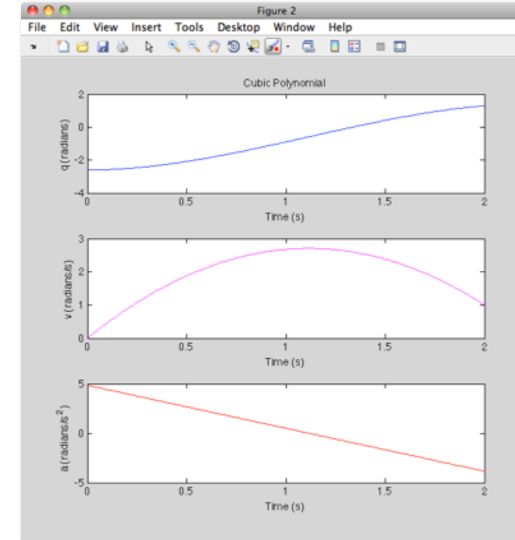
A **trajectory** is a function of time $\vec{q}(t)$

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

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

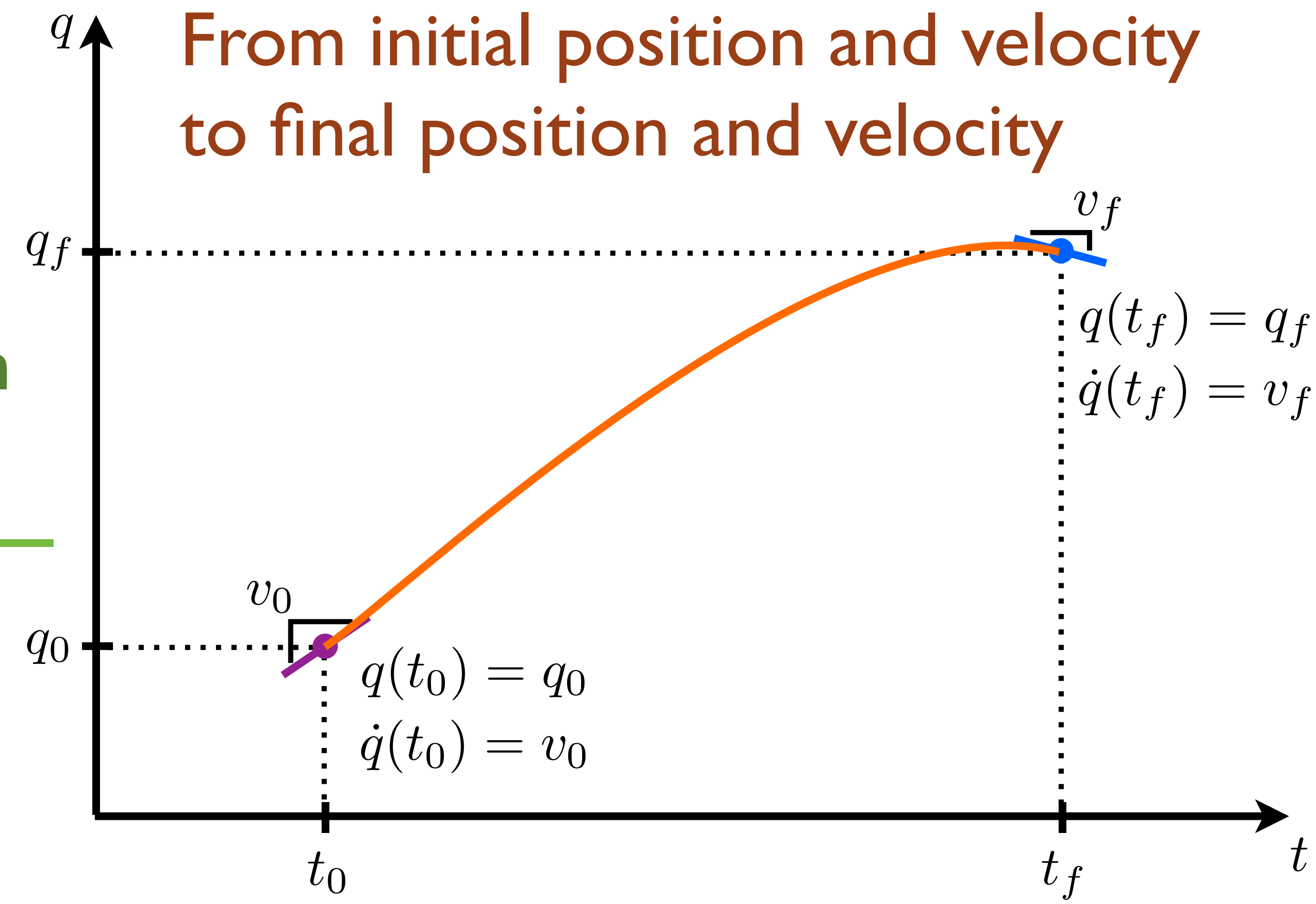
instead of $\vec{q}(t)$

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



Cubic Polynomial

From initial position and velocity to final position and velocity



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

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

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

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

time matrix

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

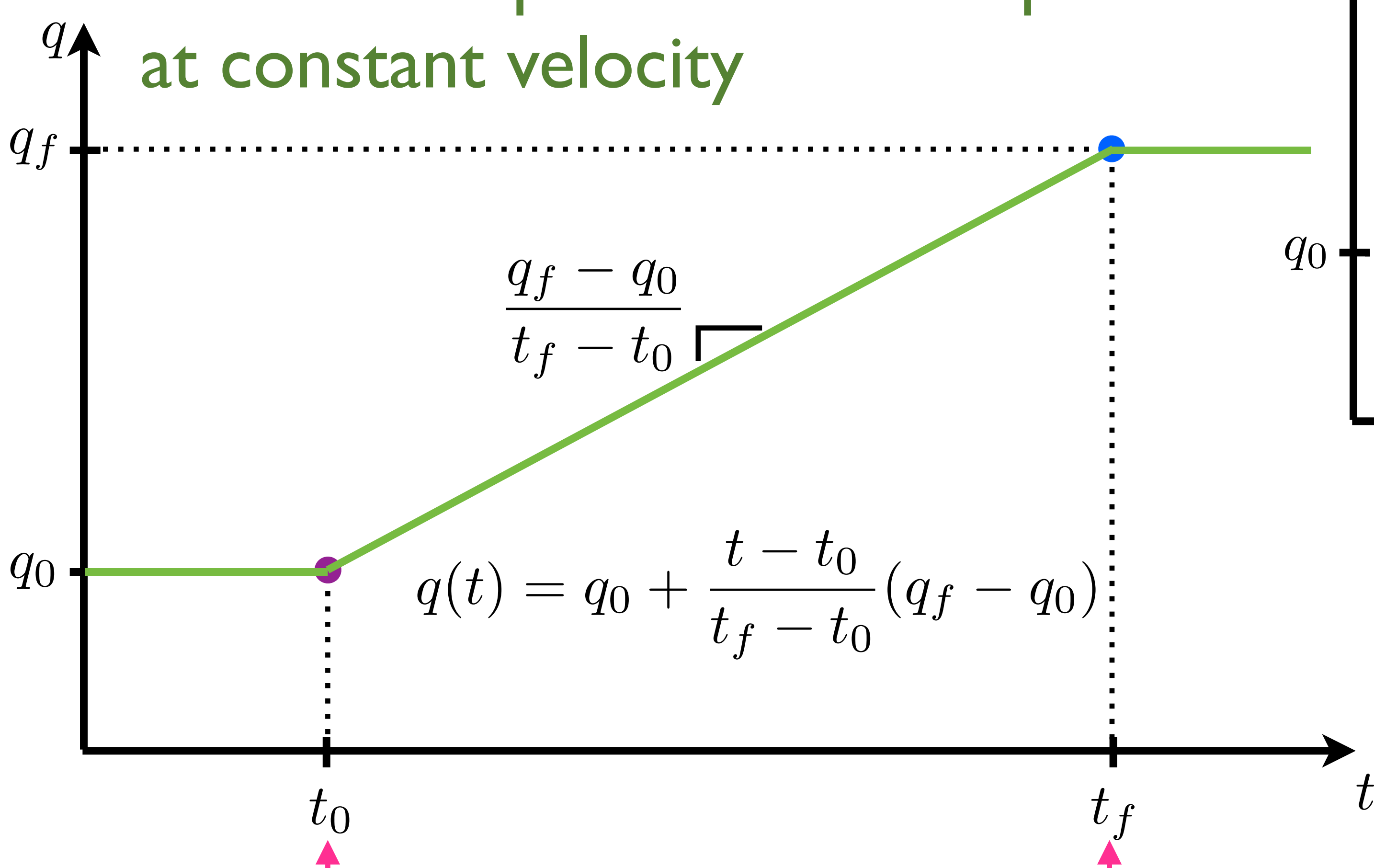
conditions

unknown coefficients

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

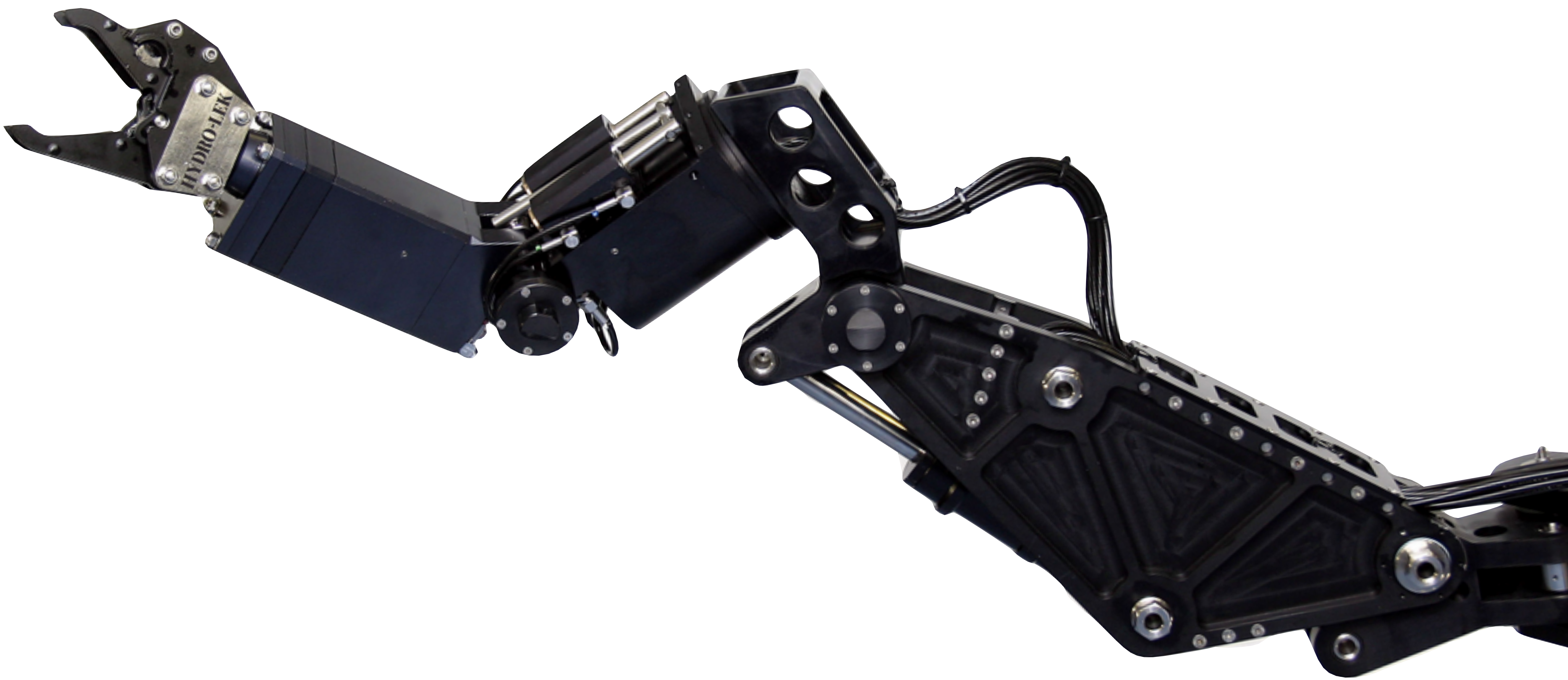
Linear Interpolation

From initial position to final position at constant velocity

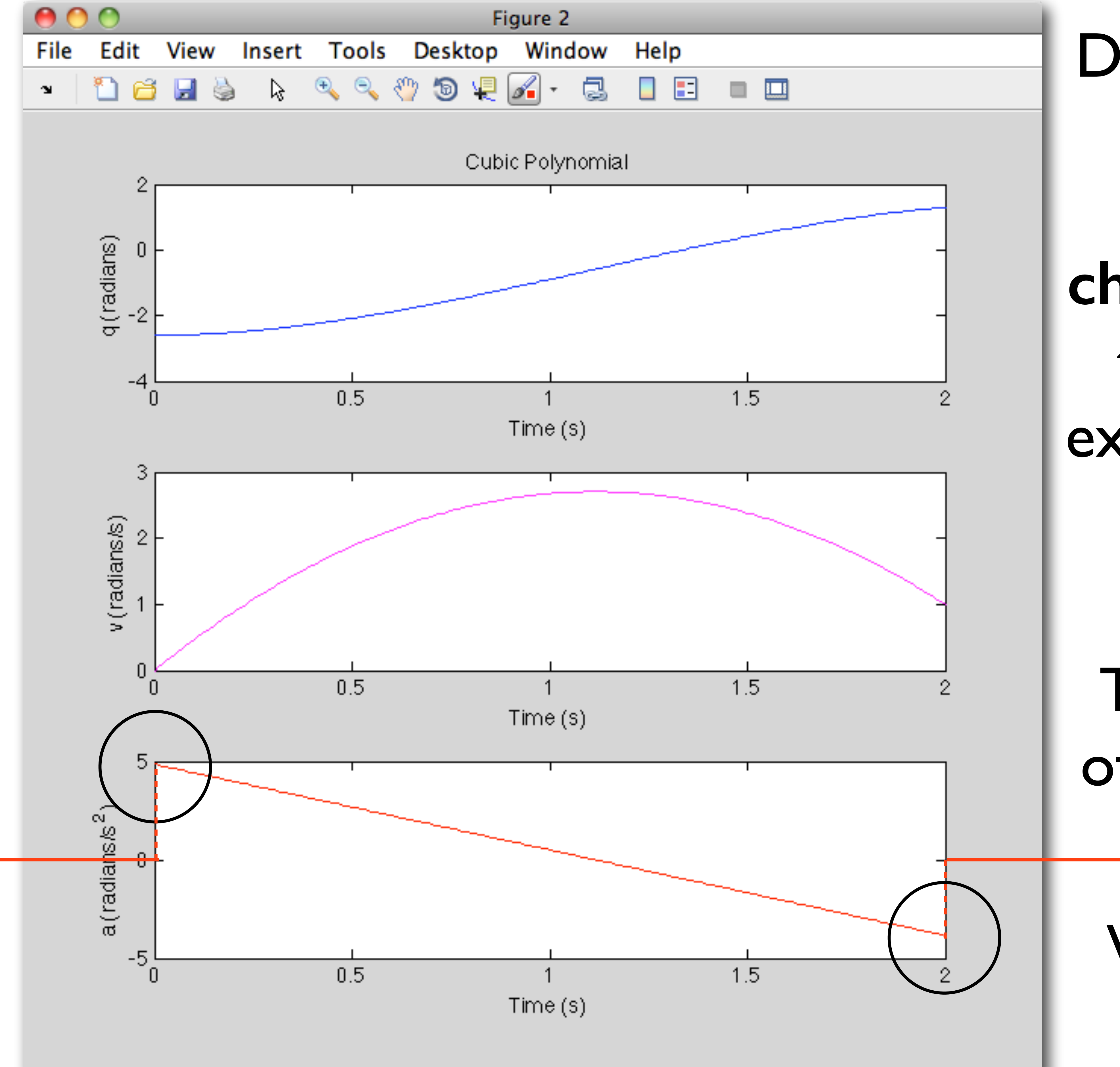


Discontinuities in velocity!

More Trajectory Planning



SHV 5.5



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

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

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

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

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

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

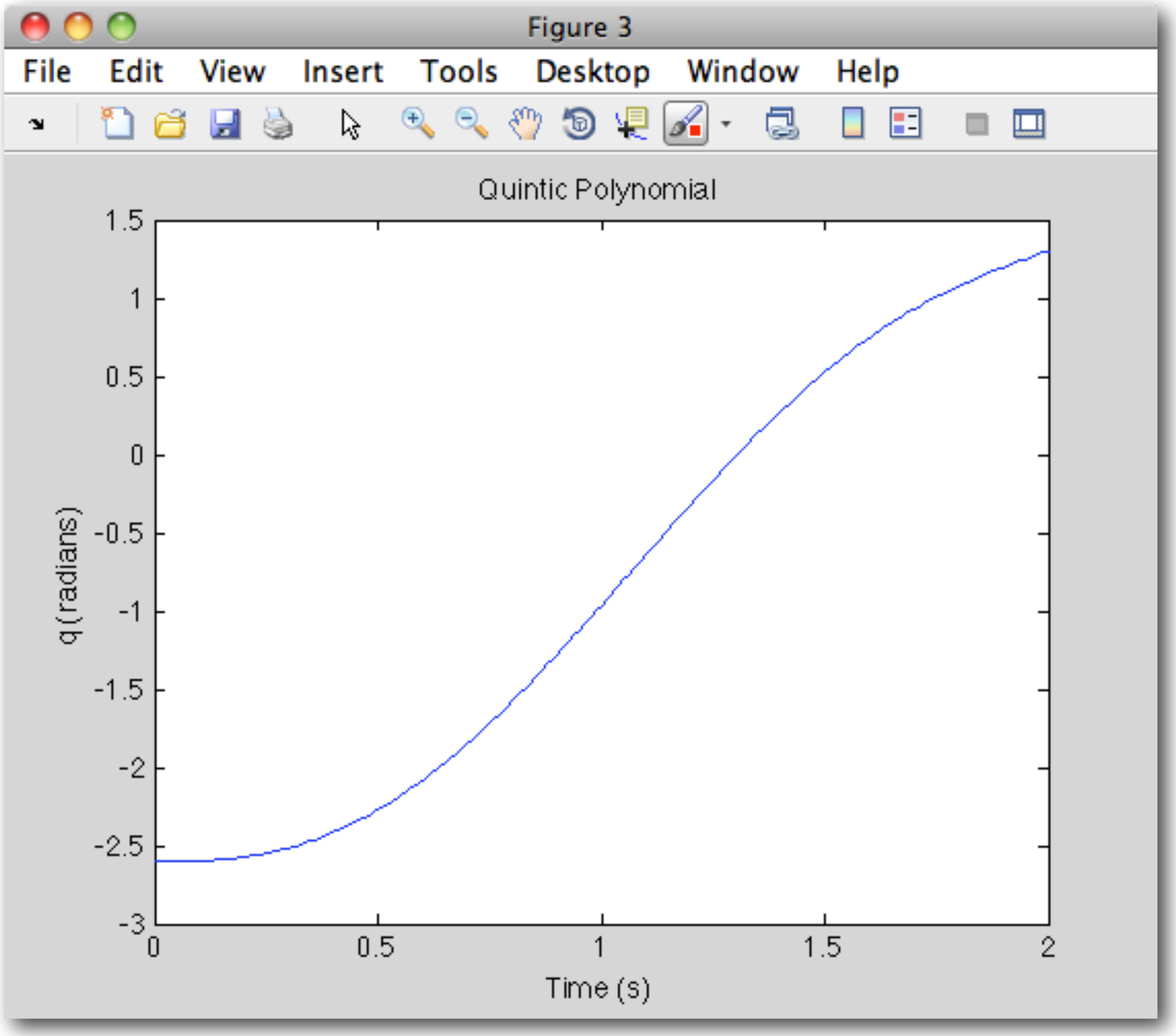


Figure 3
File Edit View Insert Tools Desktop Window Help
Quintic Polynomial
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
q (radians)
0
0.5
1
1.5
2
Time (s)

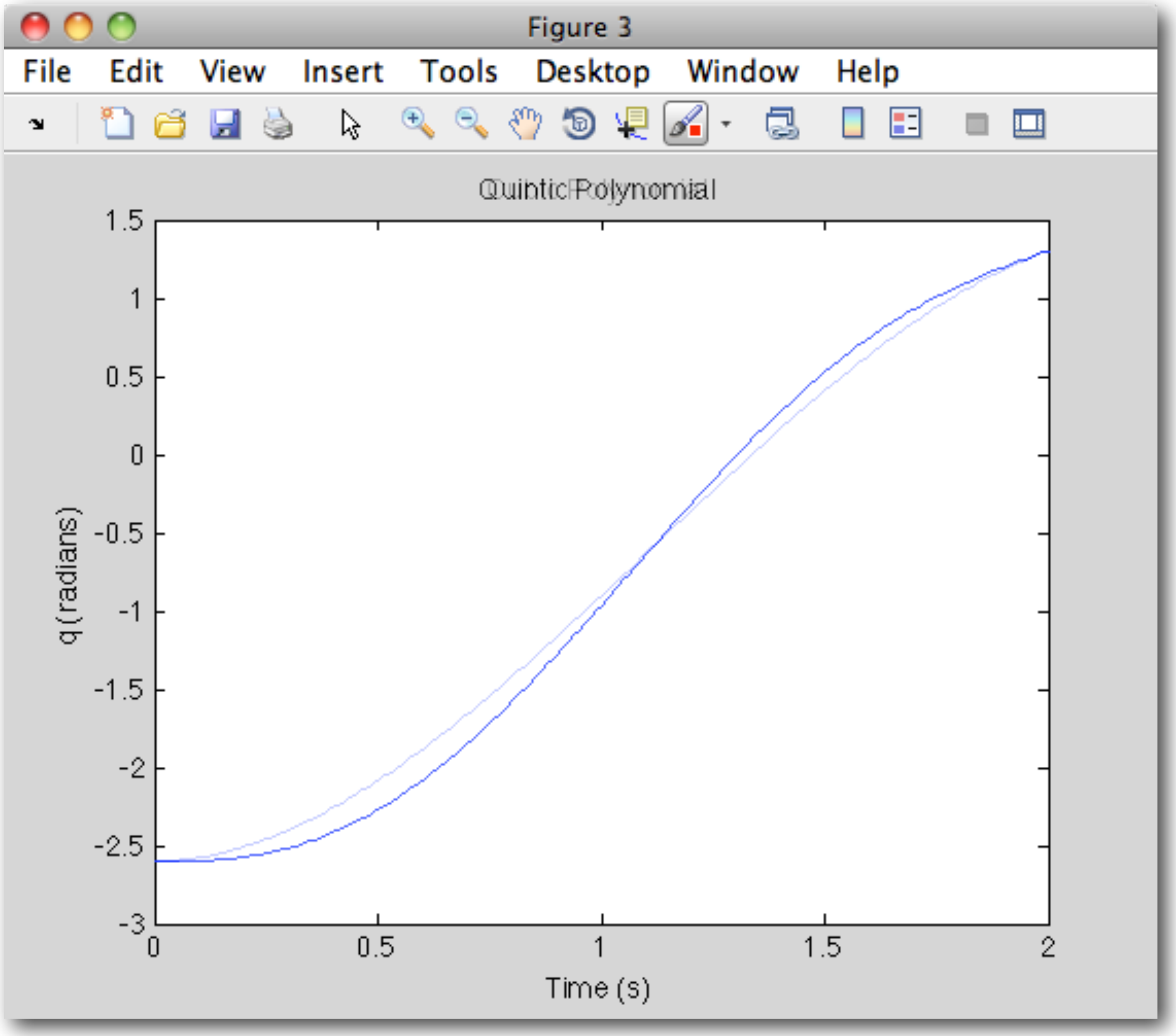


Figure 3
File Edit View Insert Tools Desktop Window Help
Quintic Polynomial
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
q (radians)
0
0.5
1
1.5
2
Time (s)

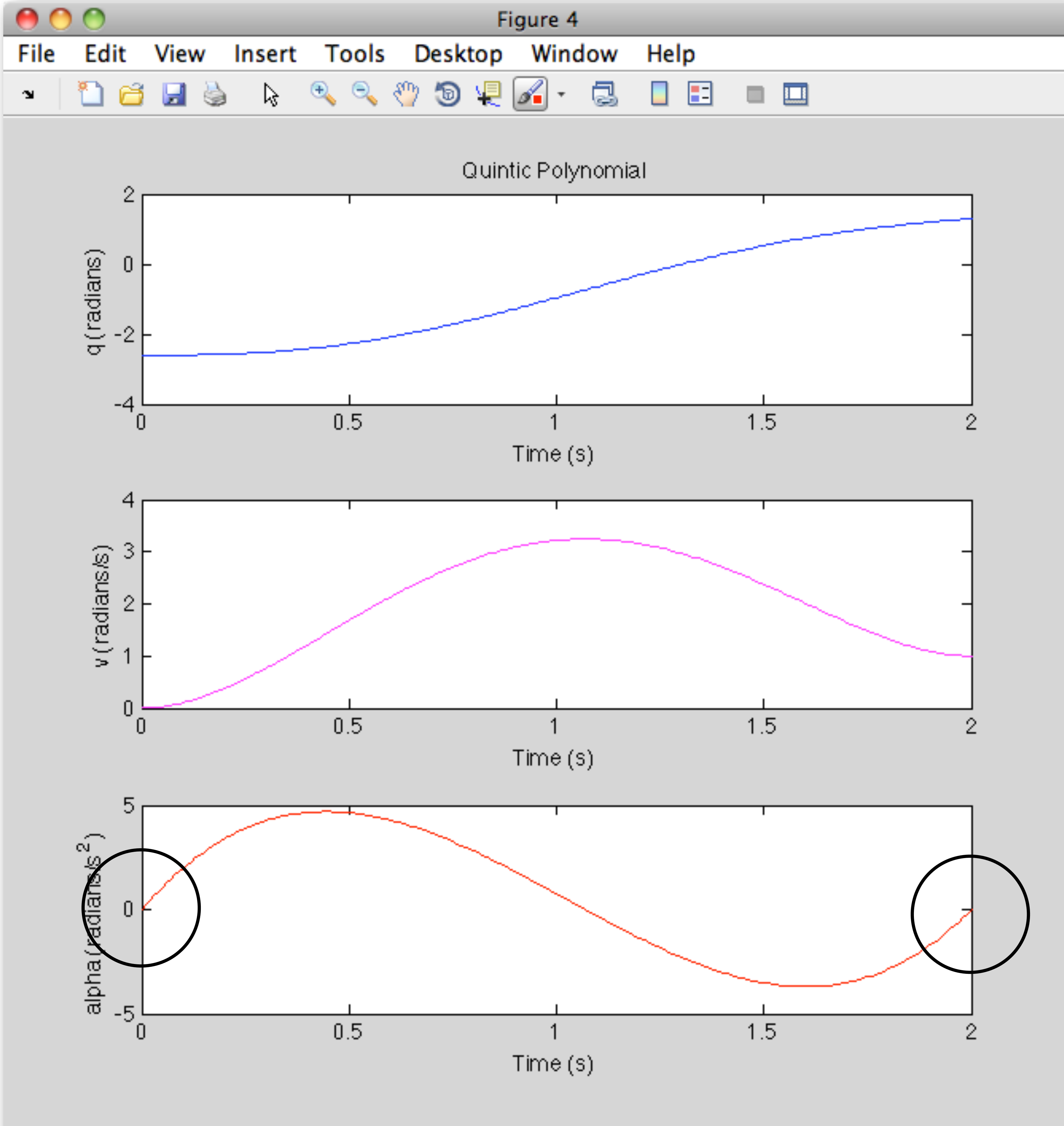


Figure 4
File Edit View Insert Tools Desktop Window Help
Quintic Polynomial
q (radians)
Time (s)
v (radians/s)
Time (s)
alpha (radians/s^2)
Time (s)

What questions do you have ?

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

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

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

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

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.

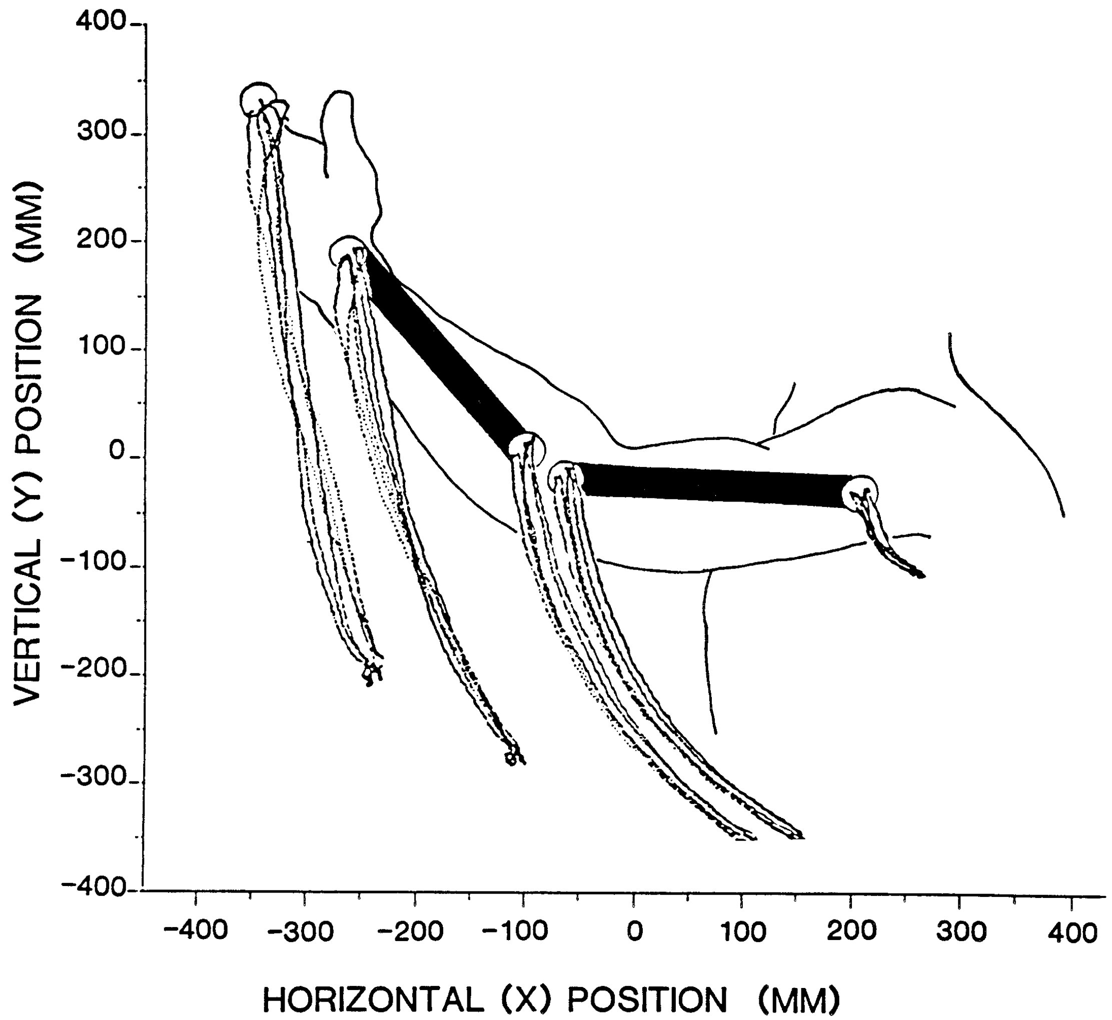


Figure 2. Attachment of Selspot markers and data presentation. Locations of the Selspot infrared LED markers and the typical format of the data presentation are shown. Note that the wrist and one of the elbow LEDs are connected by a rigid bar aligned with the forearm, and the shoulder and other elbow 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 reconstructured 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

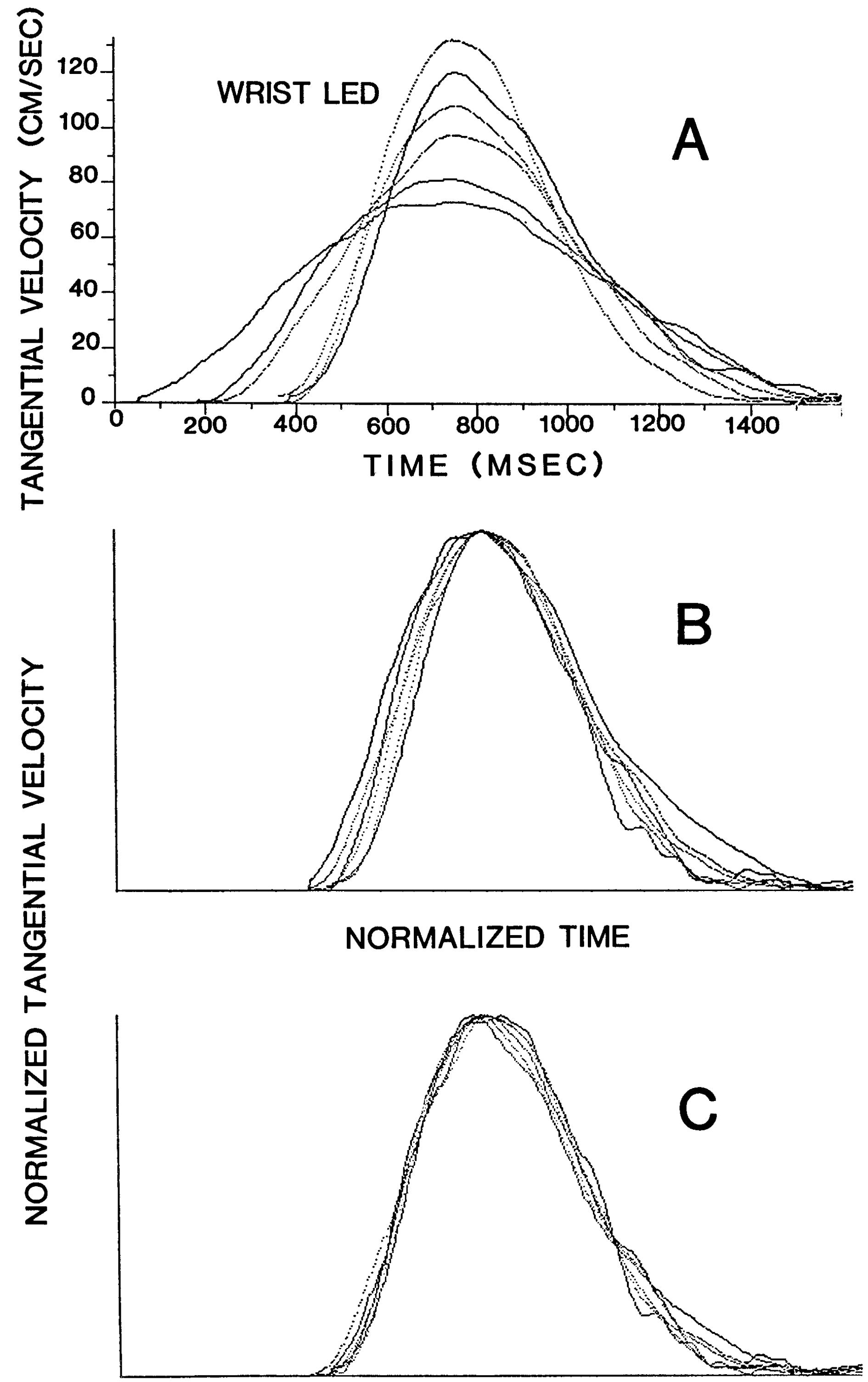


Figure 3. Normalization of tangential velocity profiles. *A*, A set of typical tangential velocity profiles of the wrist LED, derived from six unloaded movements at slow speeds between targets 3 and 7. *B*, Normalized tangential velocity profiles indicating substantial overlap. The units for the normalized 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.

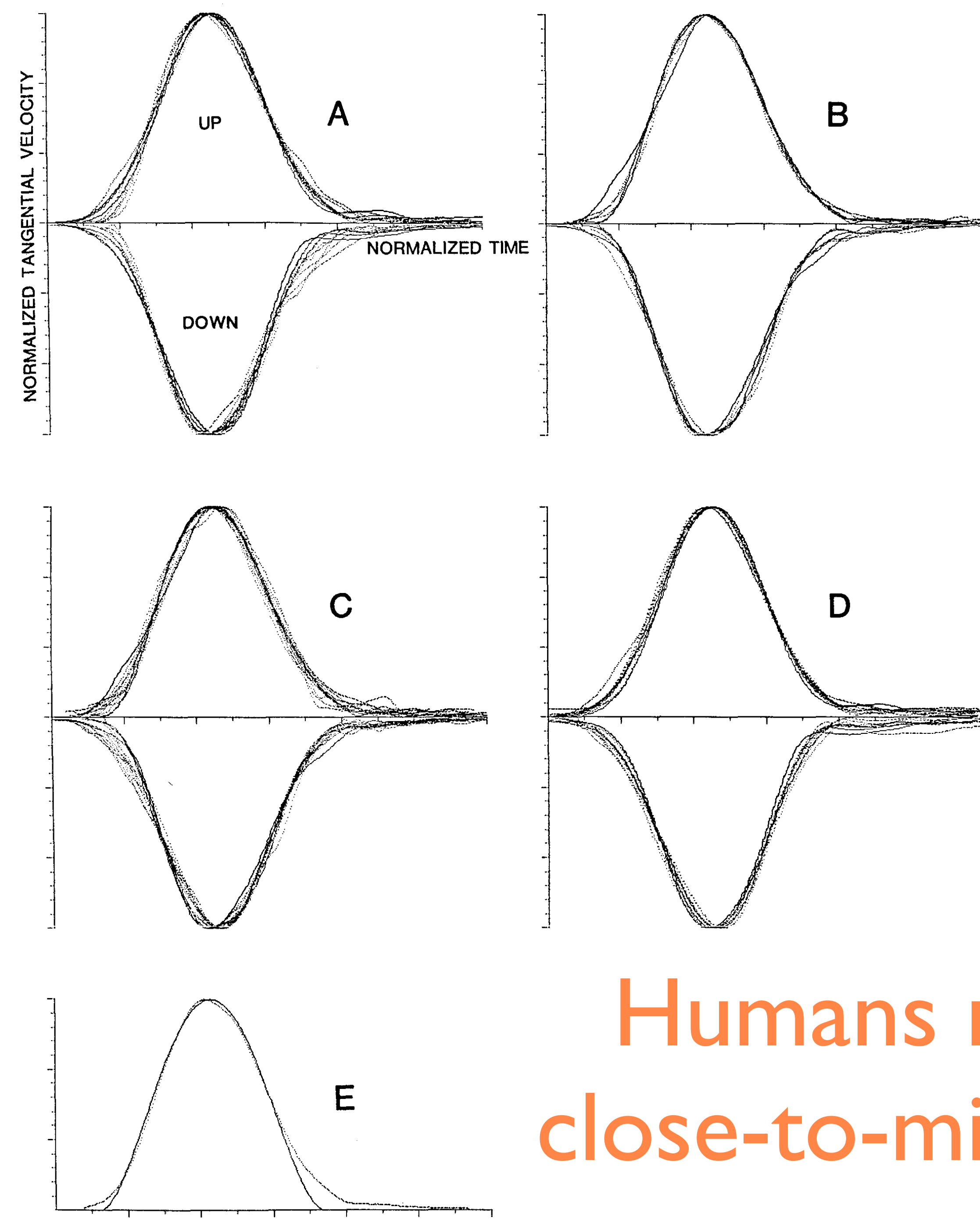


Figure 7. Tangential velocity profile shape invariance across different conditions. *A*, Different speeds. One subject's unloaded movements between targets 3 and 7 at slow, medium, and fast speeds are shown. Slow and fast movements are *dotted lines* here, whereas medium-speed movements are presented 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.

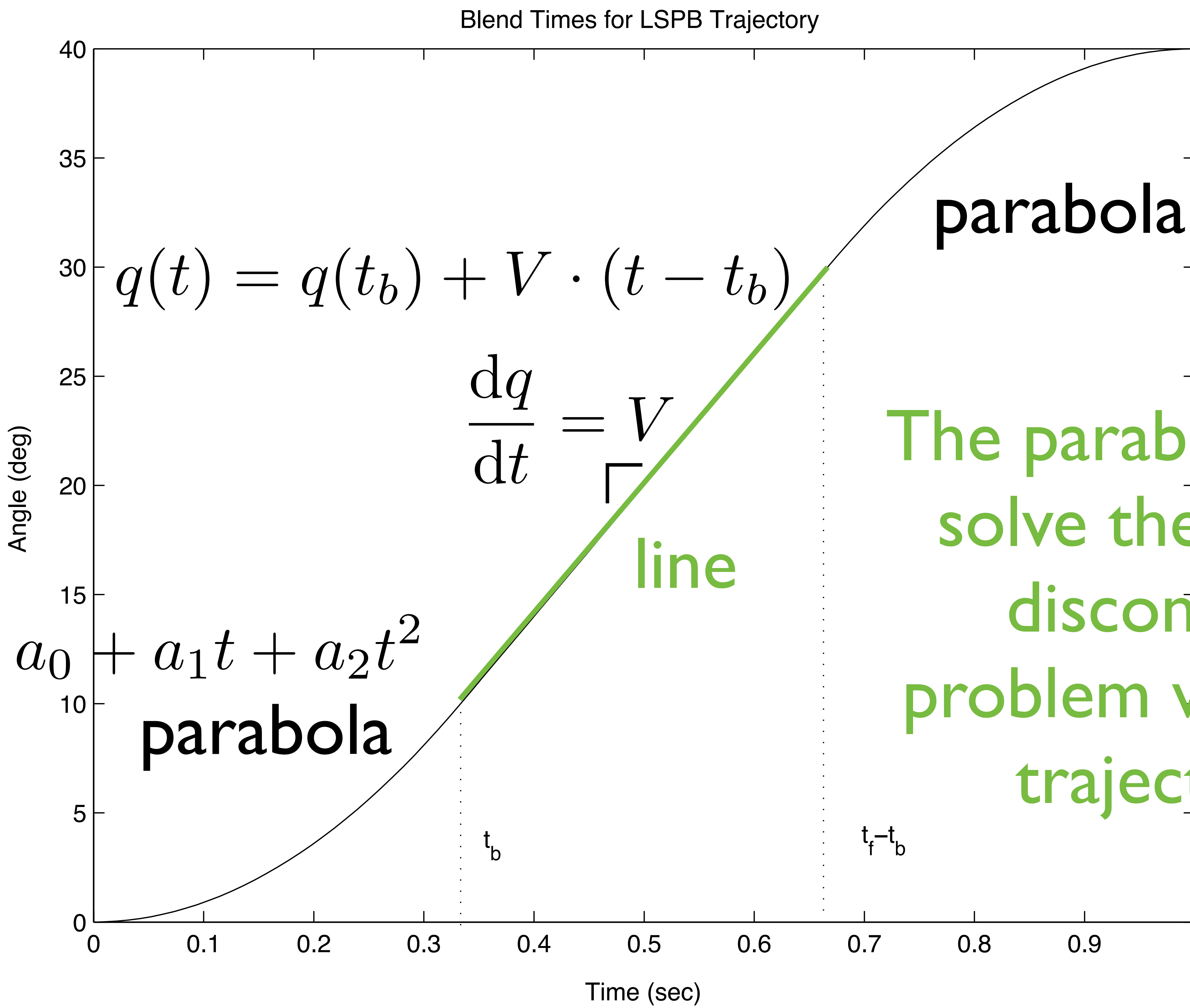
Move at constant velocity for a while.

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)



$$q(t) = q(t_b) + V \cdot (t - t_b)$$

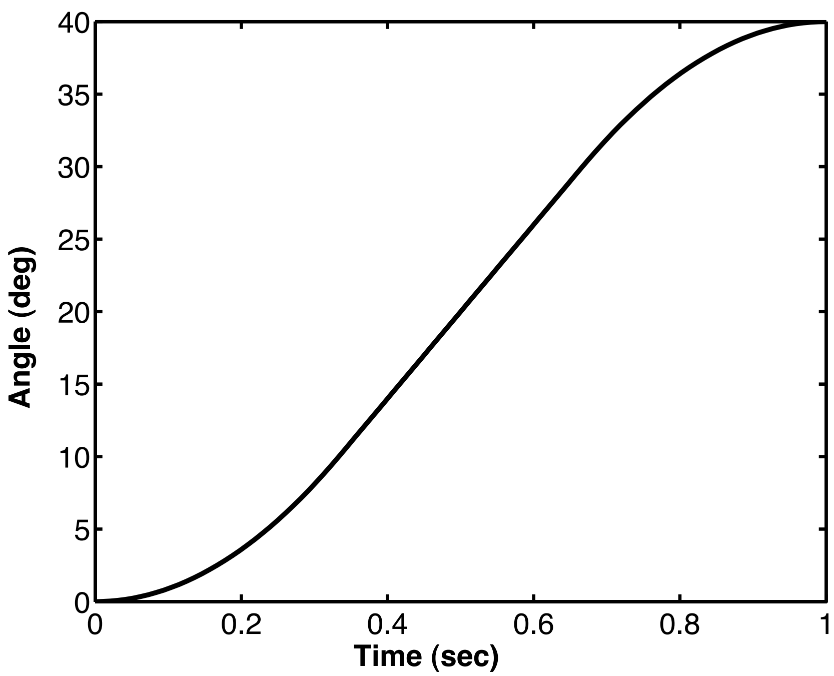
$$\frac{\mathrm{d}q}{\mathrm{d}t} = V$$

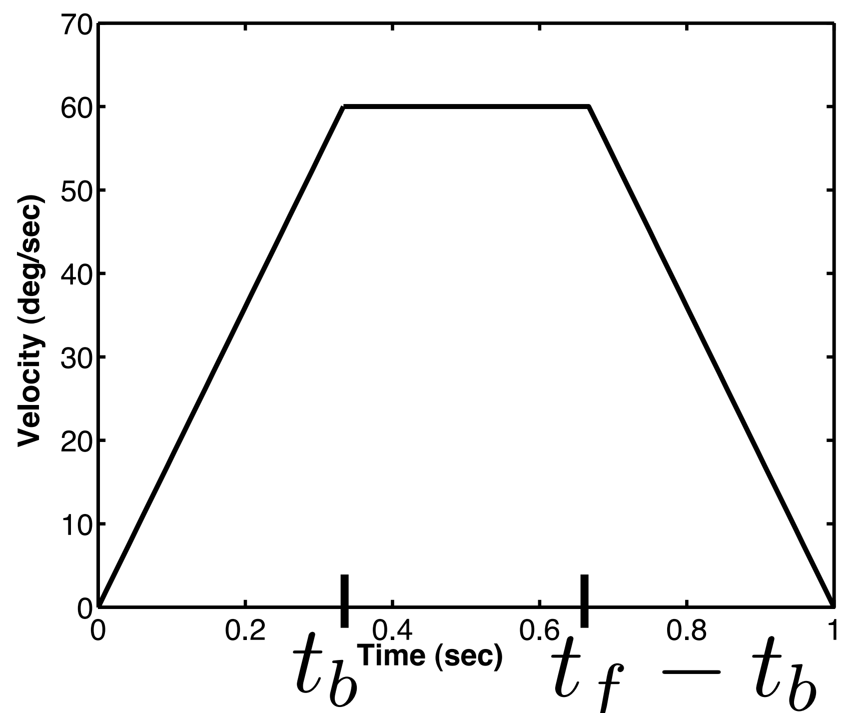
$$q(t) = a_0 + a_1 t + a_2 t^2$$

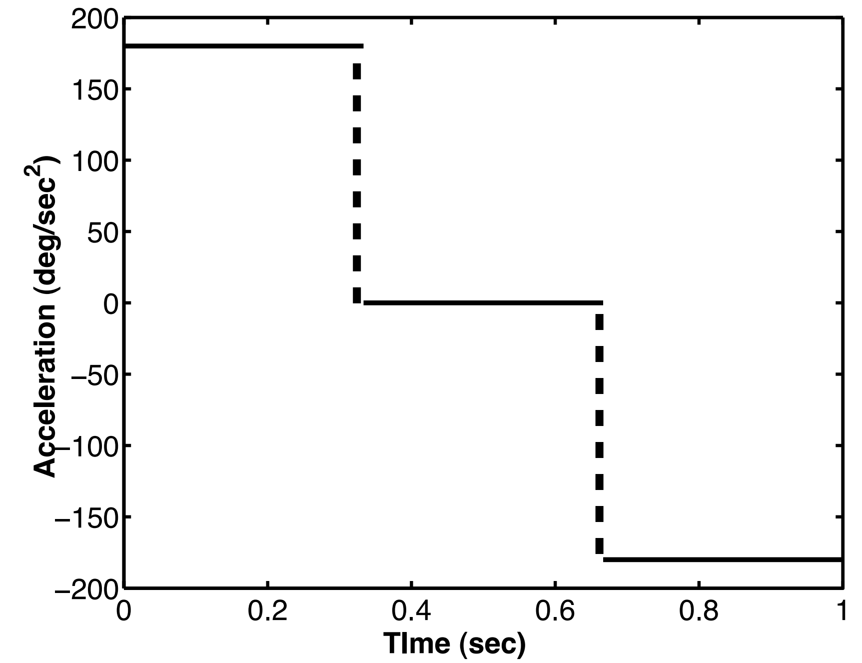
The parabolic blends solve the velocity discontinuity problem with linear trajectories.

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.

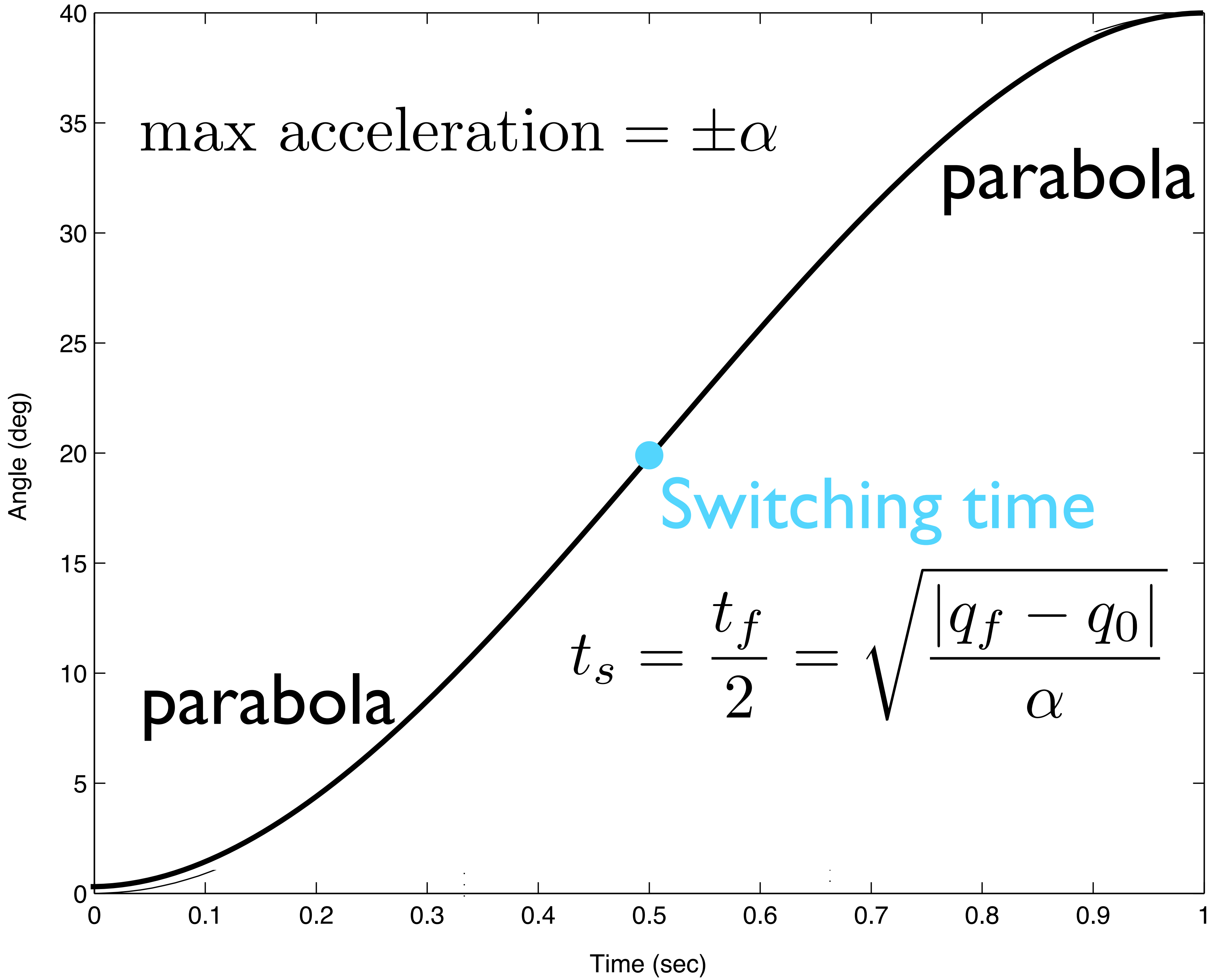
Apply maximum acceleration in one direction, then abruptly switch to negative maximum acceleration.

Typically starting and ending at rest.

Switching time is halfway through the trajectory.

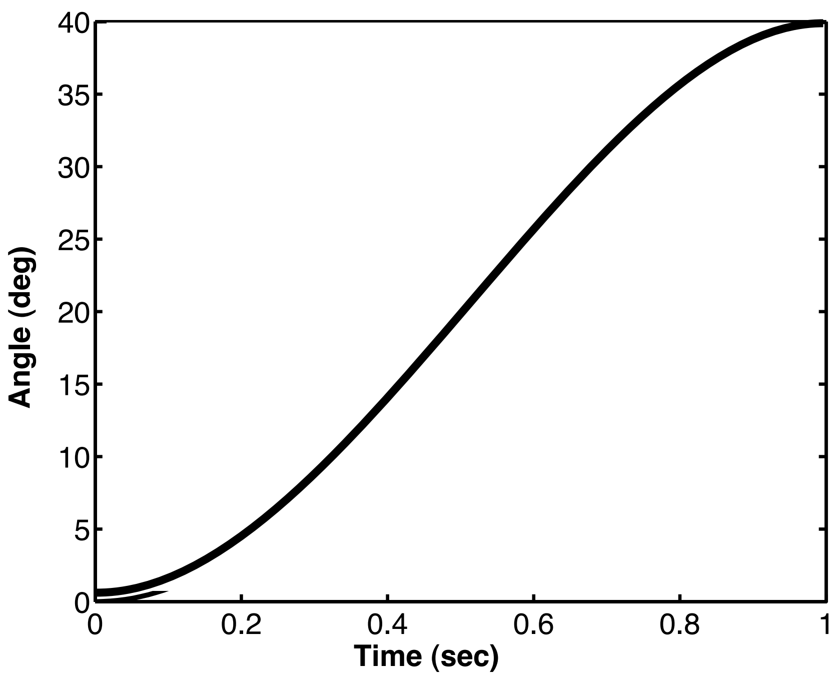
Getting There As Fast As Possible

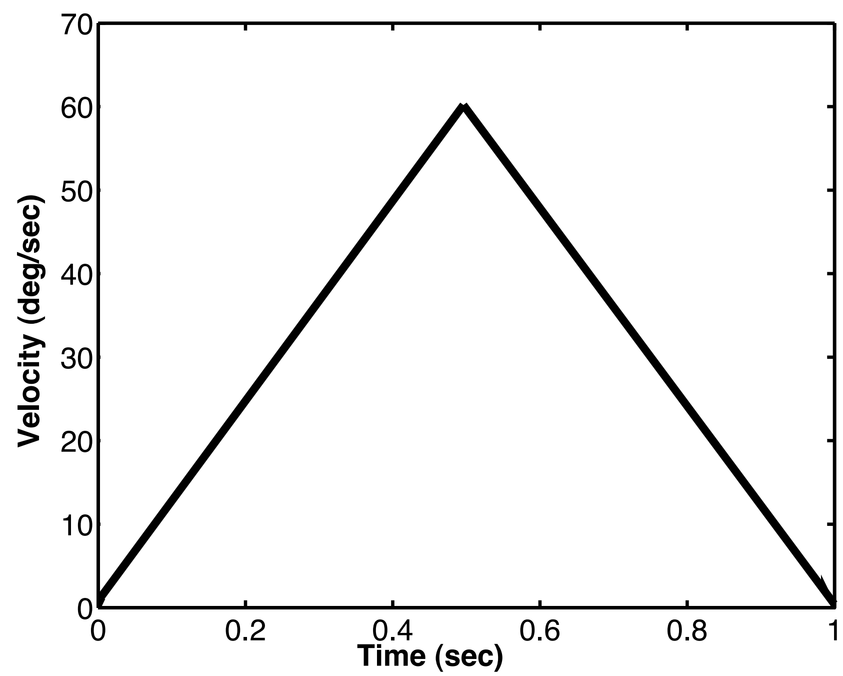
Minimum Time Trajectories, a.k.a. Bang-Bang Trajectories

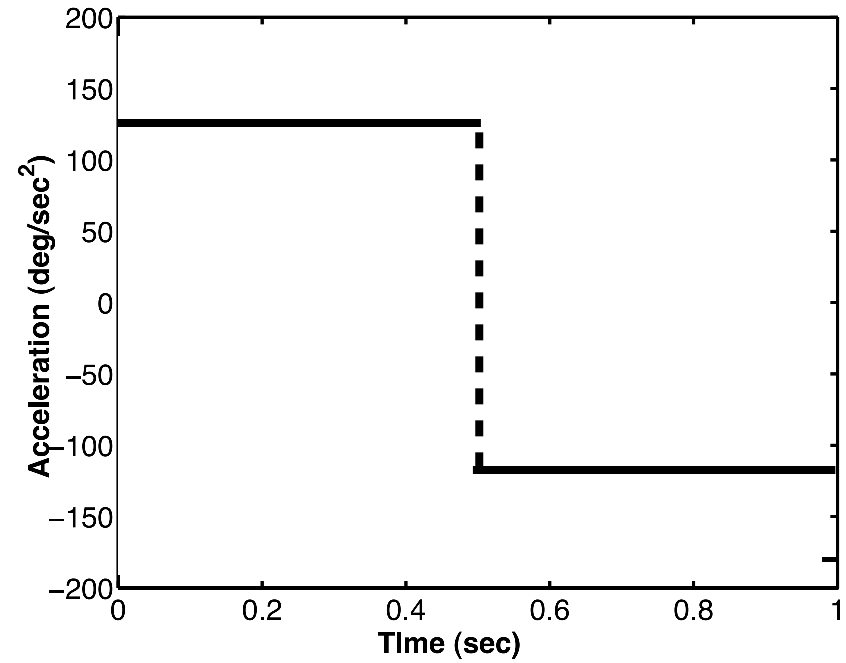


Getting There As Fast As Possible
Minimum Time Trajectories, a.k.a. Bang-Bang Trajectories

Piecewise constant max accelerations



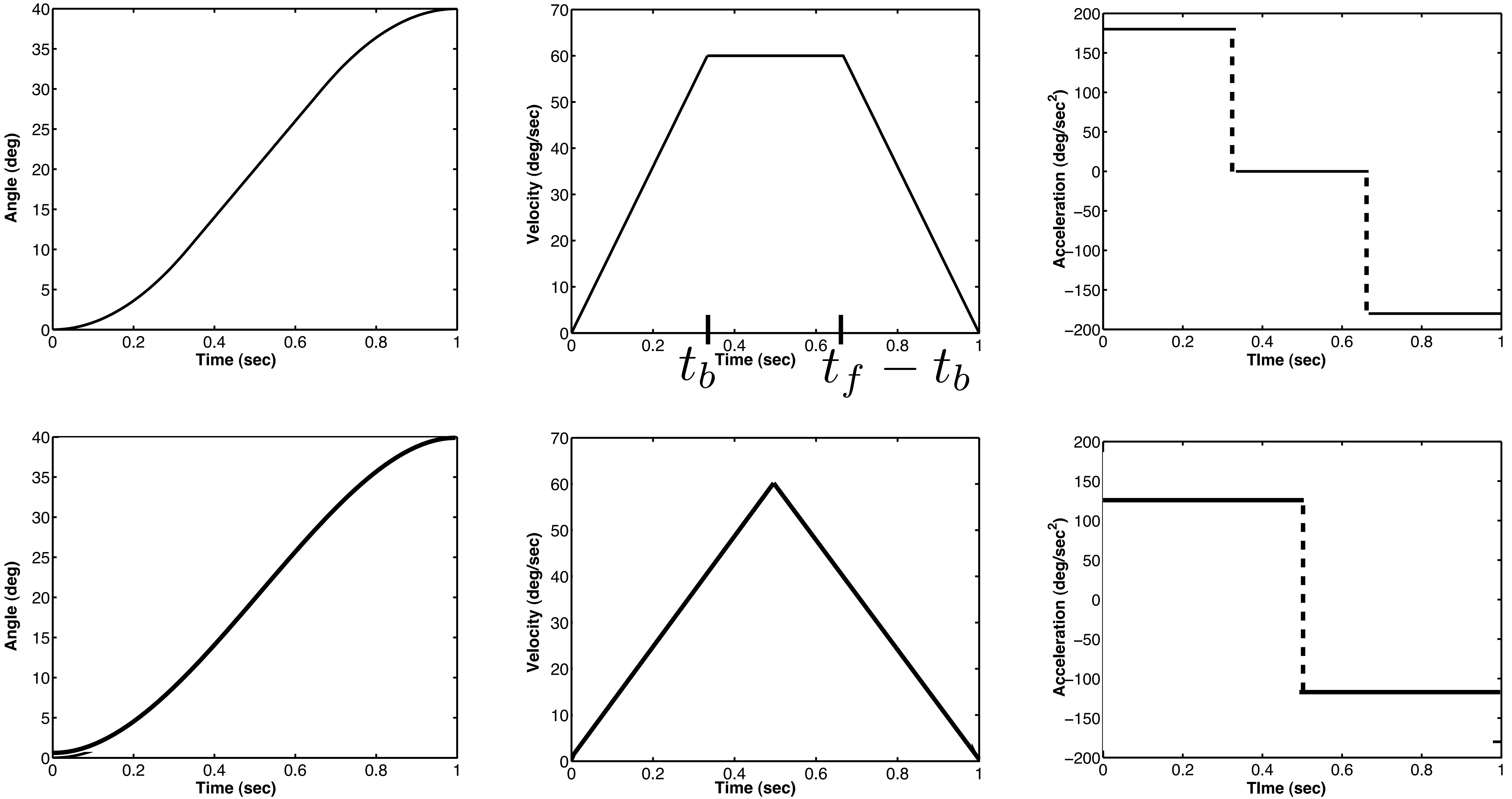




Triangular velocity profile

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

What questions do you have ?



Moving Through Via Points

You could solve for a single high-order polynomial that hits all your via points.

This approach yields a nice continuously differentiable curve.

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

Moving Through Via Points

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.

The final conditions for one polynomial become the initial conditions for the next!

Editor – /Users/kuchenbe/Documents/teaching/meam 520/lectures/10 trajectories/make_two_cubic_polynomials.m

EDITOR | PUBLISH | VIEW

make_cubic_polynomial.m | make_quintic_polynomial.m | make_two_cubic_polynomials.m

```matlab

%% Define the problem.

% Define initial and final times.
t0 = 0; % s
tf = 2; % s

% Define initial conditions.
q0 = -2.6; % radians
v0 = 0;    % rad/s

% Define final conditions.
qf = 1.3;  % radians
vf = 1;    % rad/s


%% Solve for the cubic polynomial coefficients that meet these conditions.

% Put initial and final conditions into a column vector.
conditions = [q0 v0 qf vf]';

% Put time elements into matrix.
mat = [1     t0    t0^2      t0^3;
       0      1    2*t0    3*t0^2;
       1     tf    tf^2      tf^3;
       0      1    2*tf    3*tf^2];

% Solve for coefficients.
coeffs = mat \ conditions;

% Pull individual coefficients out.
a0 = coeffs(1);
a1 = coeffs(2);
a2 = coeffs(3);
a3 = coeffs(4);


%% Plot the cubic polynomial we calculated.
```

script | Ln 14 Col 1

Editor – /Users/kuchenbe/Documents/teaching/meam 520/lectures/10 trajectories/make_two_cubic_polynomials.m

EDITOR | PUBLISH | VIEW

make_cubic_polynomial.m | make_quintic_polynomial.m | make_two_cubic_polynomials.m

```matlab
    set(gca,'fontsize',14)
    plot(t,qdoubledot,'r')
    xlabel('Time (s)')
    ylabel('a (radians/s^2)')

end

%% Define the second problem.

% Define initial and final times.
t0 = tf; % s
tf = 5; % s

% Define initial conditions.
q0 = qf; % radians
v0 = vf;    % rad/s

% Define final conditions.
qf = -2.3;  % radians
vf = -.5;     % rad/s


%% Solve for the cubic polynomial coefficients that meet these conditions.

% Put initial and final conditions into a column vector.
conditions = [q0 v0 qf vf]';

% Put time elements into matrix.
mat = [1    t0    t0^2     t0^3;
       0     1    2*t0   3*t0^2;
       1    tf    tf^2     tf^3;
       0     1    2*tf   3*tf^2];

% Solve for coefficients.
coeffs = mat \ conditions;

% Pull individual coefficients out.
a0 = coeffs(1);
```

script | Ln 113 Col 21

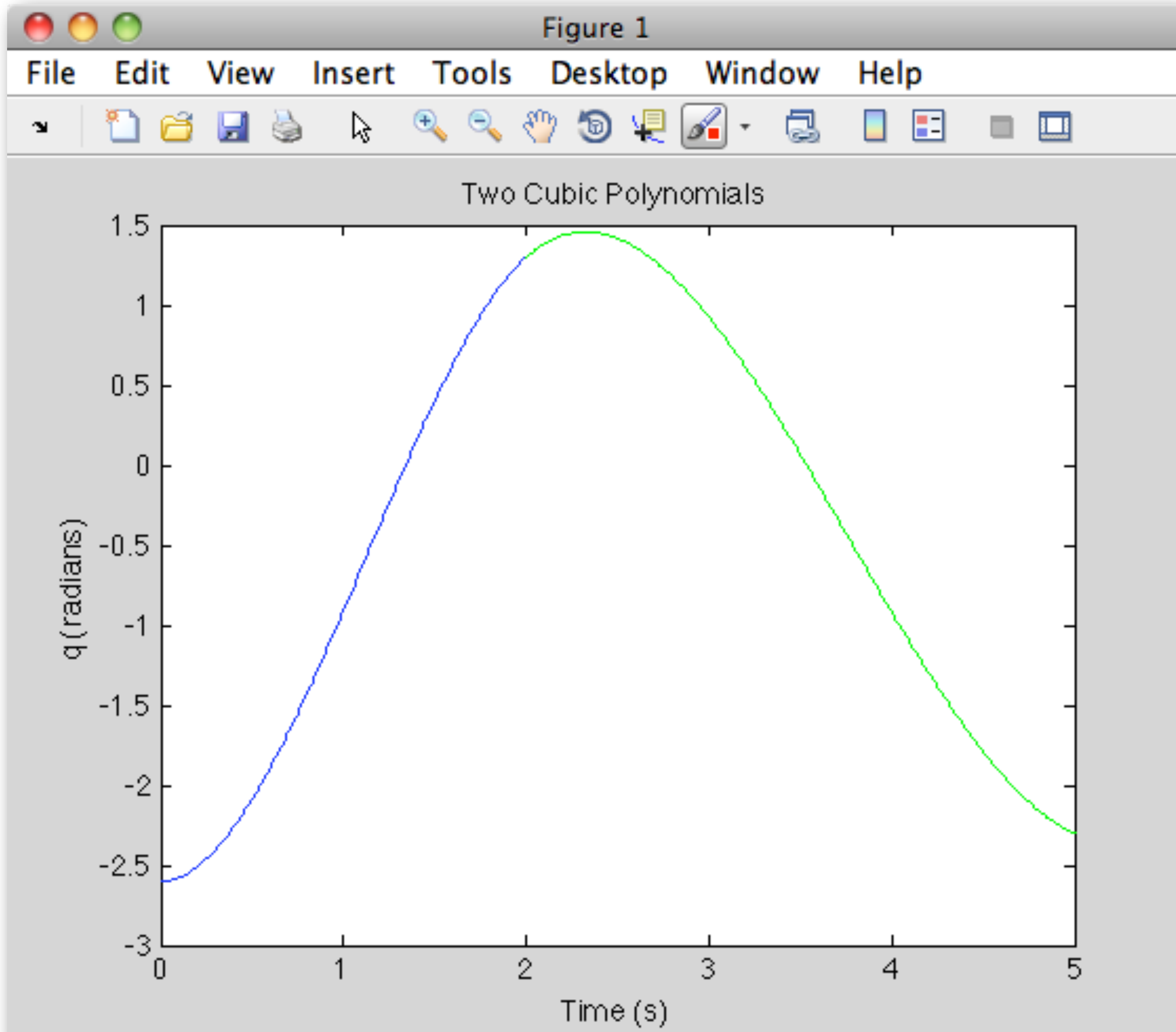
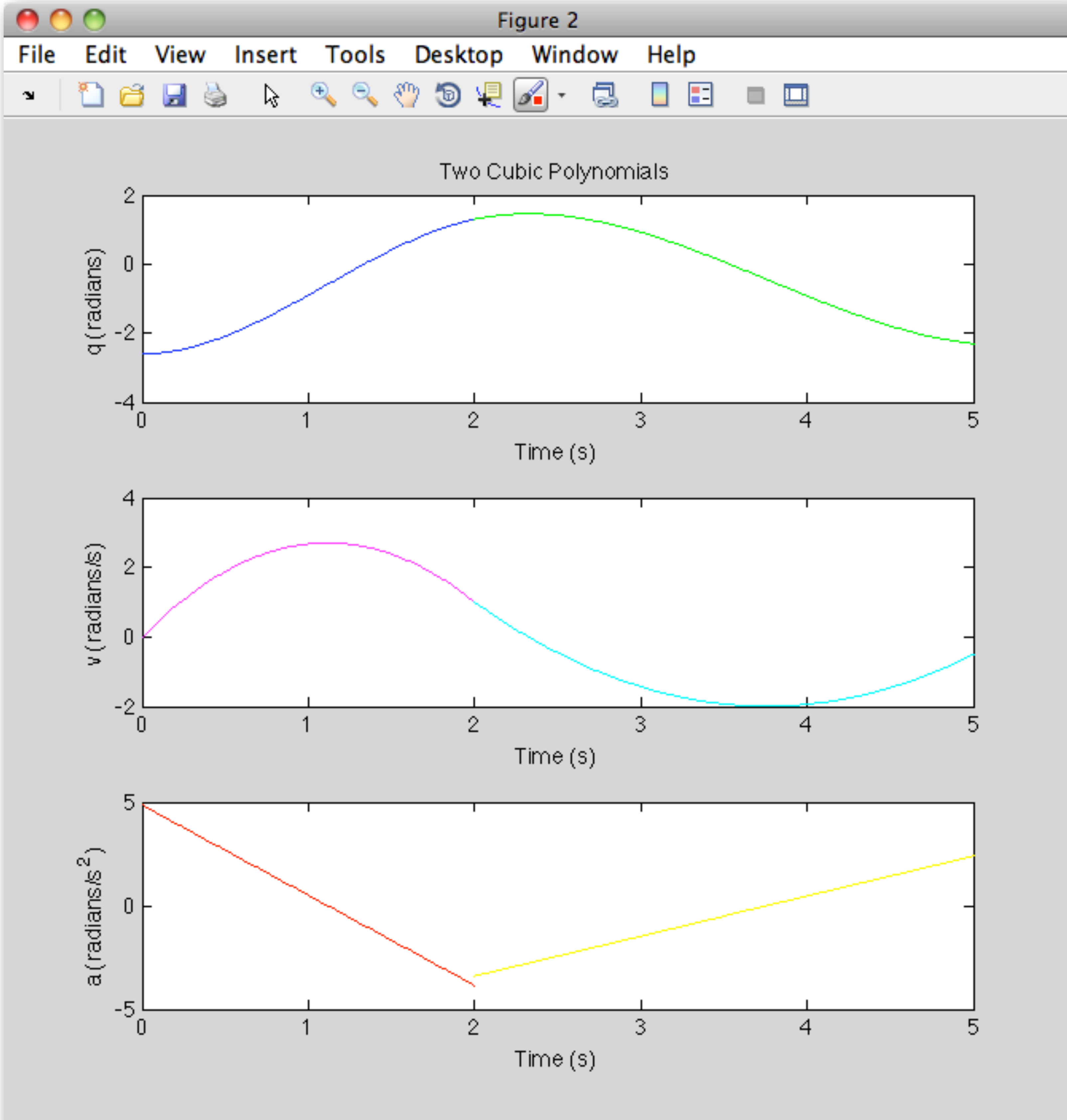


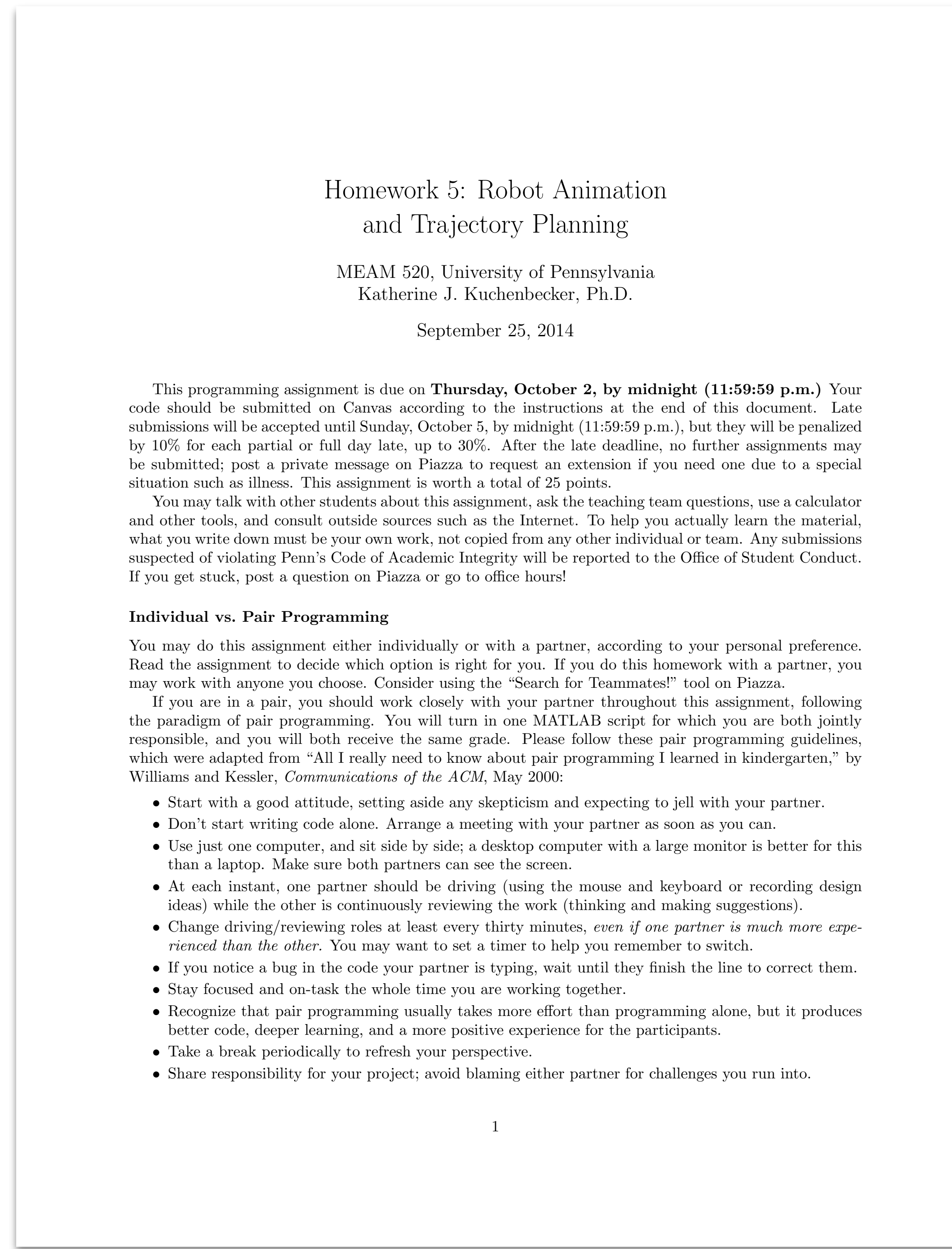
Figure 1
File Edit View Insert Tools Desktop Window Help
Two Cubic Polynomials
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
q (radians)
0 1 2 3 4 5
Time (s)

What questions do you have ?



Homework 5

- Due by midnight (11:59 p.m.) on Thursday, October 2.
- Includes only programming, which can be done individually or in pairs, following the paradigm of pair programming.
- The first part is on trajectory planning, worth 10 points.
- The second part is on robot animation, worth 15 points.



Homework 5: Robot Animation
and Trajectory Planning

MEAM 520, University of Pennsylvania
Katherine J. Kuchenbecker, Ph.D.

September 25, 2014

This programming assignment is due on **Thursday, October 2, by midnight (11:59:59 p.m.)** Your code should be submitted on Canvas according to the instructions at the end of this document. Late submissions will be accepted until Sunday, October 5, by midnight (11:59:59 p.m.), but they will be penalized by 10% for each partial or full day late, up to 30%. 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 such as illness. This assignment is worth a total of 25 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 write down 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. If you get stuck, post a question on Piazza or go to office hours!

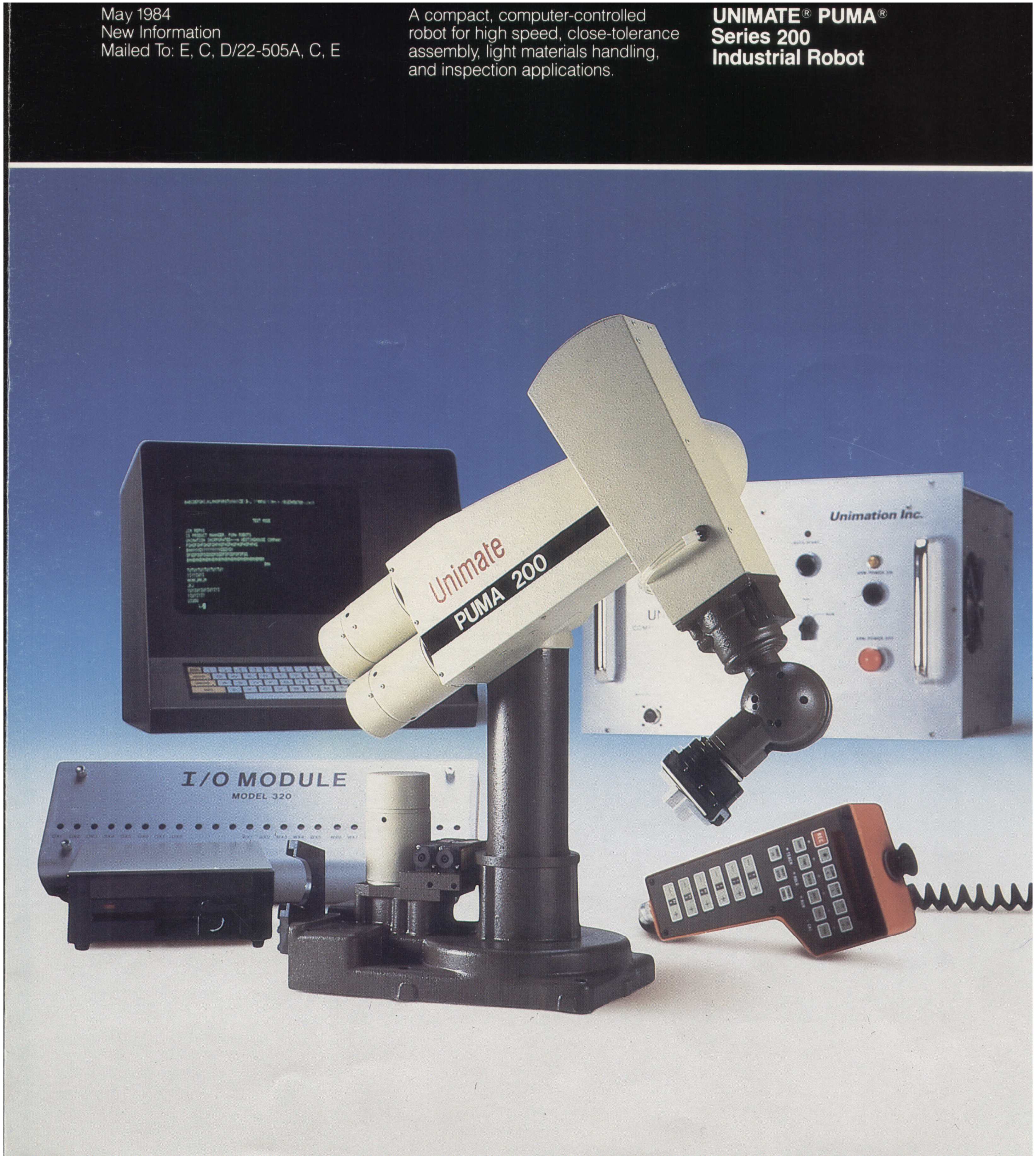
Individual vs. Pair Programming

You may do this assignment either individually or with a partner, according to your personal preference. Read the assignment to decide which option is right for you. If you do this homework with a partner, you may work with anyone you choose. Consider using the "Search for Teammates!" tool on Piazza.

If you are in a pair, you should work closely with your partner throughout this assignment, following the paradigm of pair programming. You will turn in one MATLAB script for which you are both jointly responsible, and you will both receive the same grade. Please follow these pair programming guidelines, which were adapted from "All I really need to know about pair programming I learned in kindergarten," by Williams and Kessler, *Communications of the ACM*, May 2000:

- Start with a good attitude, setting aside any skepticism and expecting to jell with your partner.
- Don't start writing code alone. Arrange a meeting with your partner as soon as you can.
- Use just one computer, and sit side by side; a desktop computer with a large monitor is better for this than a laptop. Make sure both partners can see the screen.
- At each instant, one partner should be driving (using the mouse and keyboard or recording design ideas) while the other is continuously reviewing the work (thinking and making suggestions).
- Change driving/reviewing roles at least every thirty 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 a bug in the code your partner is typing, wait until they finish the line to correct them.
- Stay focused and on-task the whole time you are working together.
- Recognize that pair programming usually takes more effort than programming alone, but it produces better code, deeper learning, and a more positive experience for the participants.
- Take a break periodically to refresh your perspective.
- Share responsibility for your project; avoid blaming either partner for challenges you run into.

1



- Starter code chooses random joint angles for PUMA 260 to start at and end at.
- User selects between five different trajectory types.
- You edit code to calculate all of the trajectories.
- Typo at top of p. 4: "cubic" should be "quintic"
- You edit code to animate the PUMA as it follows the trajectories.
- Includes extra credit opportunities.

MATLAB Tips

MATLAB does exactly what you tell it to do and no more.

Turn on MATLAB's beep (beep on) so you can hear when there is an error in your code.

Read the first error message carefully.
(MATLAB has bad social skills and speaks cryptically, but these messages give you clues!)

Click the link to where the error is in your code.

Examine your variables in the workspace.

Test commands at the command line.

MEAM 520 – September 25, 2014 – Prof. K. J. Kuchenbecker – University of Pennsylvania

Trajectory Planning Questions

1. 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 $q(t_f) = q_f$.

2. We discussed using linear algebra to solve for the coefficients of the cubic polynomial that satisfies the specified conditions. Dr. Kuchenbecker said that the only time one cannot find a solution is when the initial time equals the final time. How could you prove her statement?

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

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

Work with one or two partners to answer the first three questions.

1. 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 $q(t_f) = q_f$.

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

$$q_0 = a_0 + a_1 t_0$$

$$q_f = a_0 + a_1 t_f$$

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

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

2. We discussed using linear algebra to solve for the coefficients of the cubic polynomial that satisfies the specified conditions. Dr. Kuchenbecker said that the only time one cannot find a solution is when the initial time equals the final time. How could you prove her statement?

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

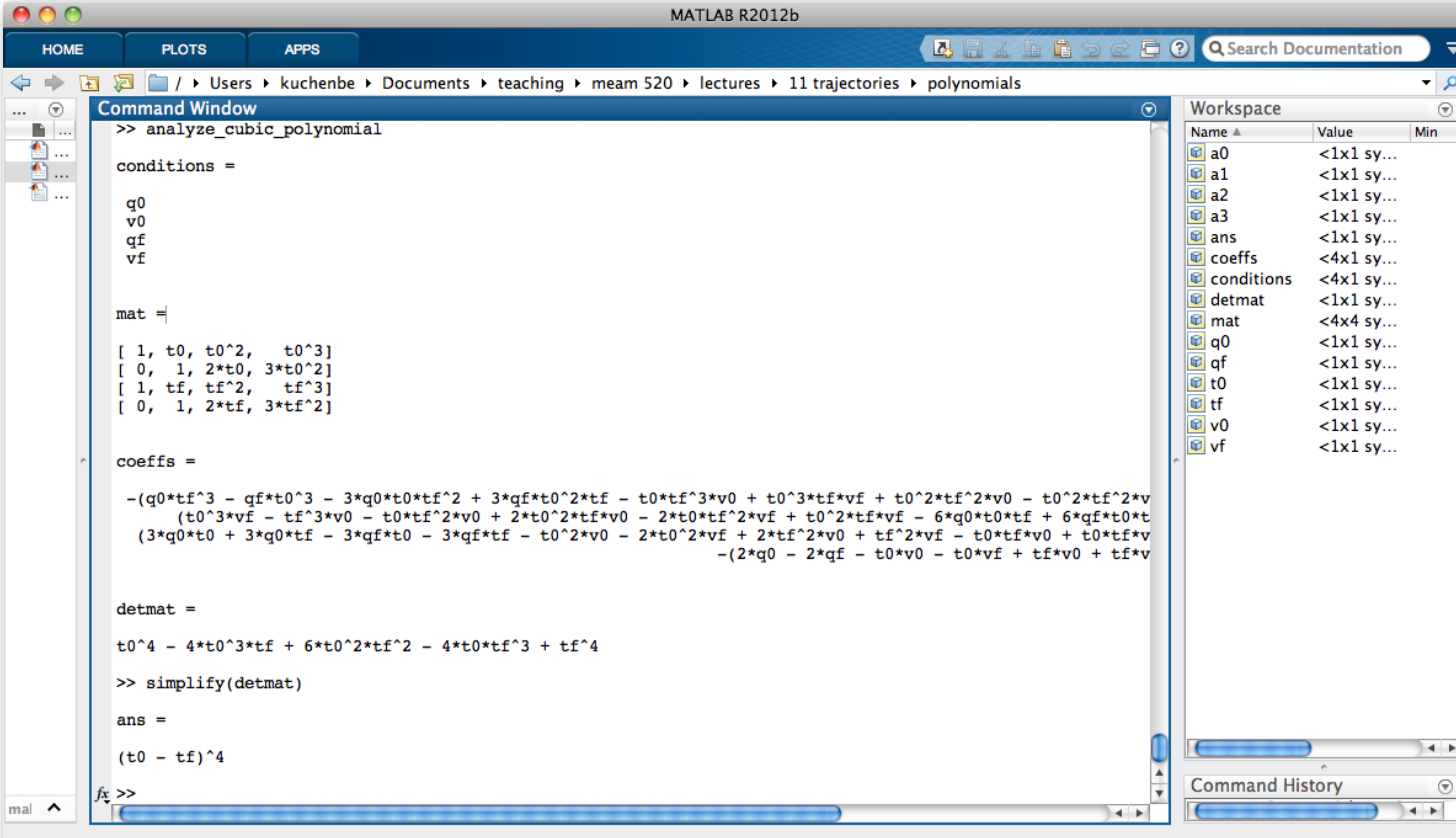
fails when $t_f = t_0$

/Users/kuchenbe/Documents/teaching/meam 520/lectures/11 trajectories/polynomials/analyze_cubic_polynomial.m

EDITOR PUBLISH VIEW

```
%% Clear the workspace.
clear


%% Define the variables as symbols that represent real numbers.
syms t0 tf q0 qf v0 vf real


%% Solve for the cubic polynomial coefficients that meet these conditions.

% Put initial and final conditions into a column vector.
conditions = [q0 v0 qf vf]'

% Put time elements into matrix.
mat = [1     t0    t0^2      t0^3;
       0      1    2*t0    3*t0^2;
       1     tf    tf^2      tf^3;
       0      1    2*tf    3*tf^2]

% Solve for coefficients.
coeffs = mat \ conditions

% Pull individual coefficients out.
a0 = coeffs(1);
a1 = coeffs(2);
a2 = coeffs(3);
a3 = coeffs(4);


%% Figure out when there is a solution.

% There is no solution when the determinant of the matrix equals zero.

% Calculate determinant of matrix.
detmat = det(mat)

%detmat = simplify(detmat)

```

script Ln 12 Col 28


MATLAB R2012b
HOME
PLOTS
APPS
Search Documentation
/ Users kuchenbe Documents teaching meam 520 lectures 11 trajectories polynomials
Command Window
>> analyze_cubic_polynomial
conditions =
q0
v0
qf
vf
mat =
[1, t0, t0^2, t0^3]
[0, 1, 2*t0, 3*t0^2]
[1, tf, tf^2, tf^3]
[0, 1, 2*tf, 3*tf^2]
coeffs =
-(q0*tf^3 - qf*t0^3 - 3*q0*t0*tf^2 + 3*qf*t0^2*tf - t0*tf^3*v0 + t0^3*tf*vf + t0^2*tf^2*v0 - t0^2*tf^2*v
(t0^3*vf - tf^3*v0 - t0*tf^2*v0 + 2*t0^2*tf*v0 - 2*t0*tf^2*vf + t0^2*tf*vf - 6*q0*t0*tf + 6*qf*t0*t
(3*q0*t0 + 3*q0*tf - 3*qf*t0 - 3*qf*tf - t0^2*v0 - 2*t0^2*vf + 2*tf^2*v0 + tf^2*vf - t0*tf*v0 + t0*tf*v
-(2*q0 - 2*qf - t0*v0 - t0*vf + tf*v0 + tf*v
detmat =
t0^4 - 4*t0^3*tf + 6*t0^2*tf^2 - 4*t0*tf^3 + tf^4
>> simplify(detmat)
ans =
(t0 - tf)^4
>>
Workspace
Name
Value
Min
a0 <1x1 sy...
a1 <1x1 sy...
a2 <1x1 sy...
a3 <1x1 sy...
ans <1x1 sy...
coeffs <4x1 sy...
conditions <4x1 sy...
detmat <1x1 sy...
mat <4x4 sy...
q0 <1x1 sy...
qf <1x1 sy...
t0 <1x1 sy...
tf <1x1 sy...
v0 <1x1 sy...
vf <1x1 sy...
Command History

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

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

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

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

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

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

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

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

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

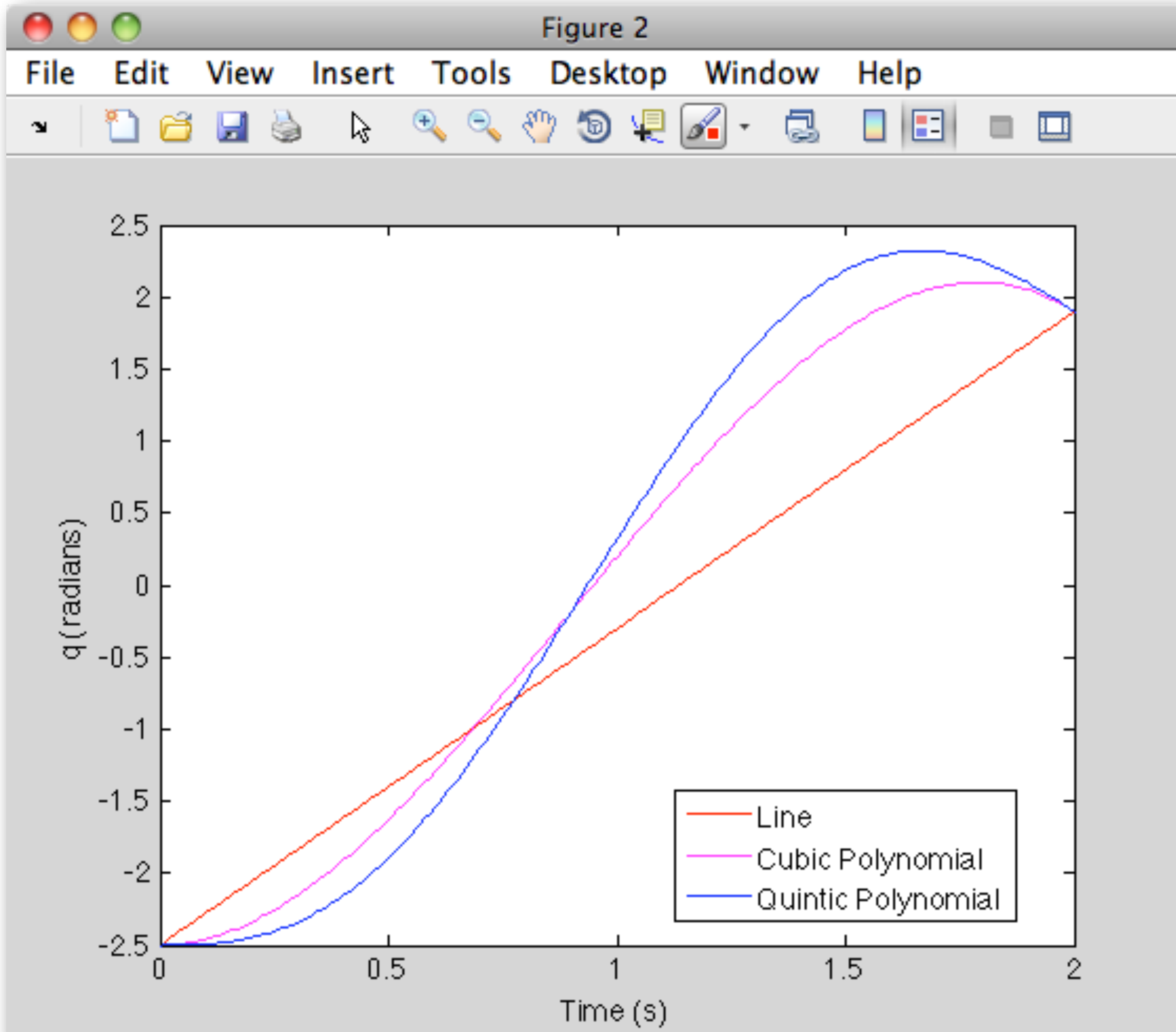
Linear Segment with Parabolic Blends (LSPB, 1 Line + 2 Quadratics) Could leave interval*.

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

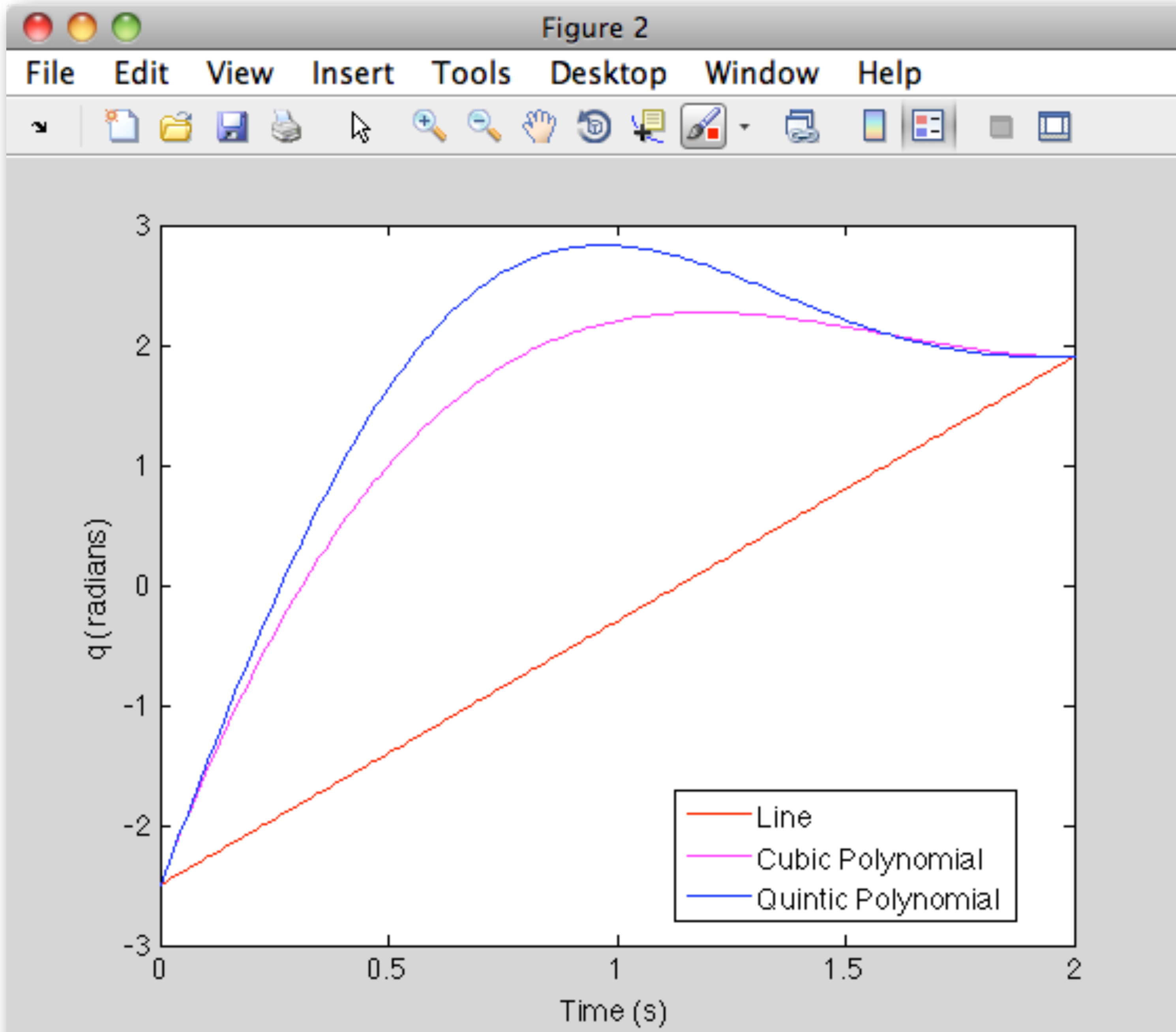
Minimum Time Trajectory (Bang-Bang, 2 Quadratics) Could leave interval*.

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

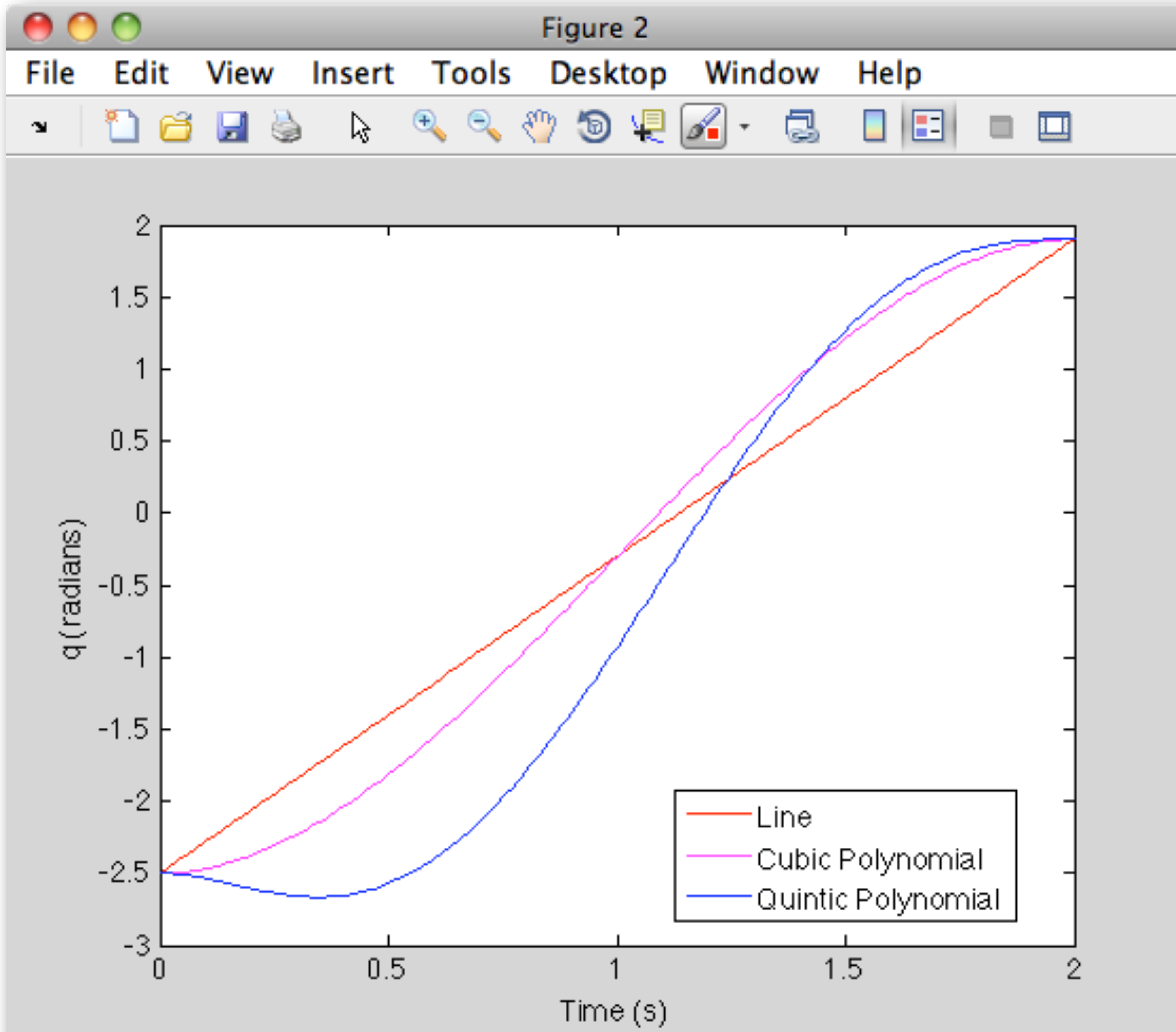
Final velocity less than zero



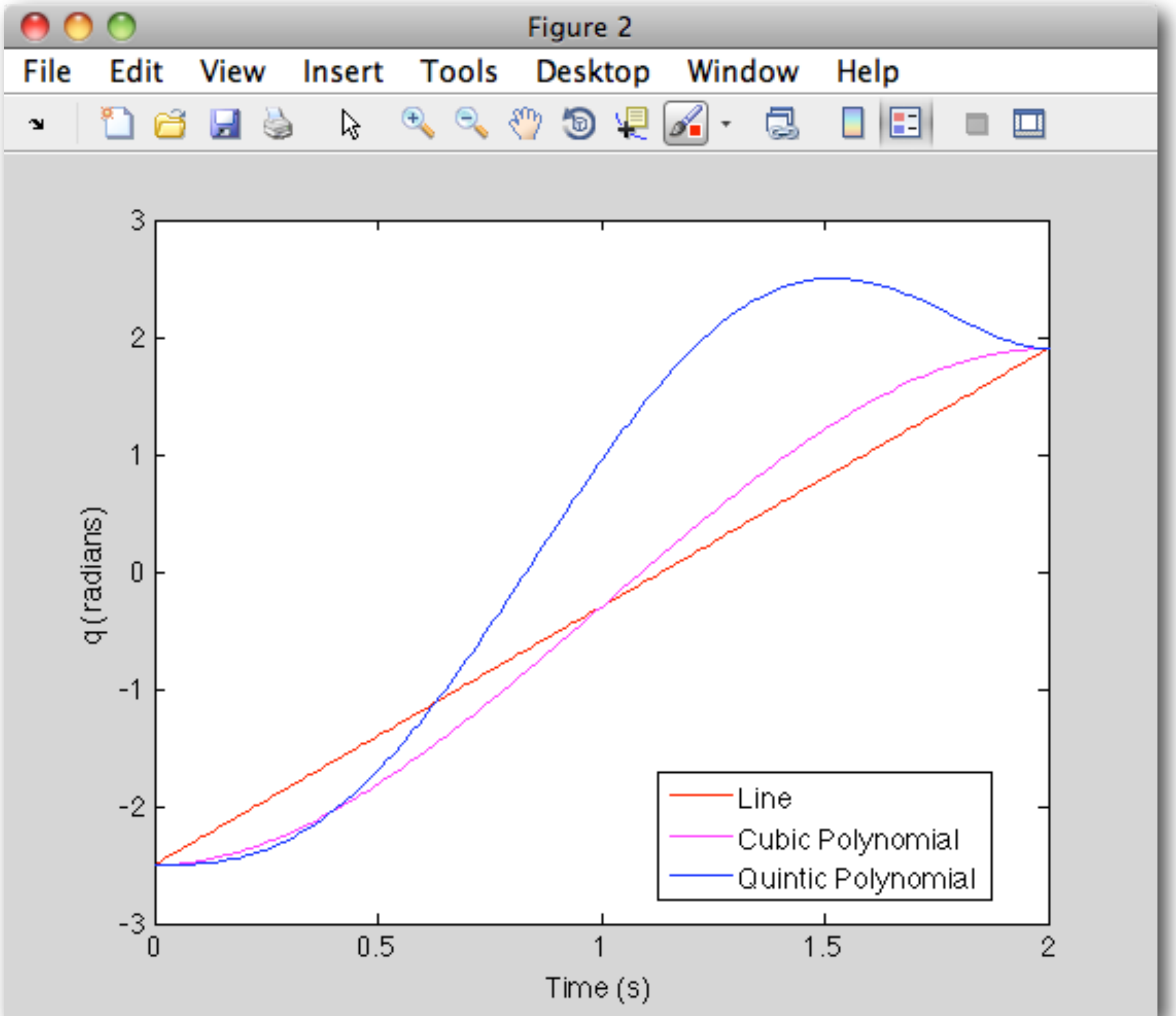
Initial velocity greater than zero and large



Initial acceleration less than zero



Final acceleration greater than zero



MEAM 520 – September 25, 2014 – Prof. K. J. Kuchenbecker – University of Pennsylvania

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

5. How does the idea of sequencing low-order polynomials such as cubics through multiple via points relate to LSPB and Bang-Bang trajectories?

6. 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 (duration of starting parabola and ending parabola). Here are the equations for the three curves: $q(t) = b_0 + b_1 t + b_2 t^2$ $\quad q(t) = a_0 + a_1 t$ $\quad q(t) = c_0 + c_1 t + c_2 t^2$

Work with one or two partners to answer the last three questions.

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

4. 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?

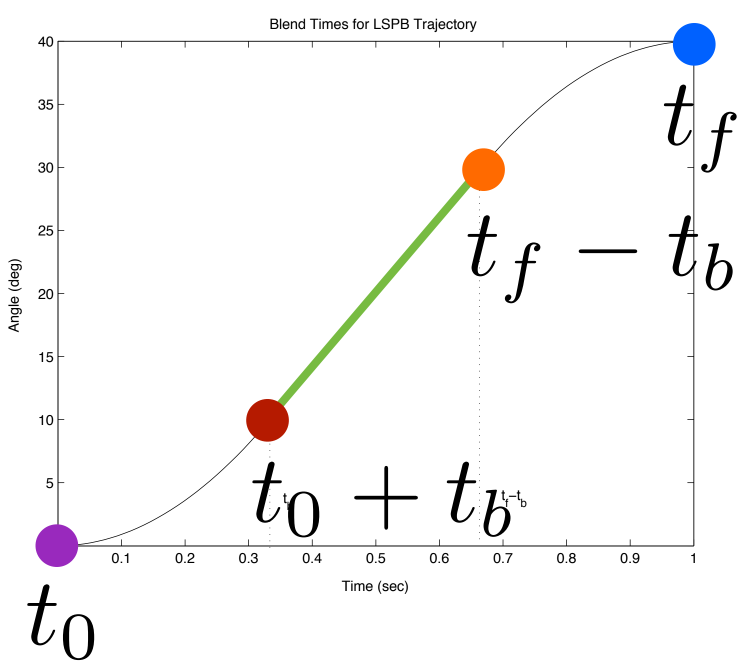
5. How does the idea of sequencing low-order polynomials such as cubics through multiple via points relate to LSPB and Bang-Bang trajectories?

- A linear segment with parabolic blends *is* a sequence of low-order polynomials: quadratic + line + quadratic.
- A bang-bang trajectory *is* a sequence of low-order polynomials: quadratic + quadratic.
- But, for LSPB and BB we don't care about the particular position or velocity of the robot at the switching times. We just require position and velocity to be continuous at these points.

6. 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.

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

8 parameters – need 8 equations



Position and velocity at four points in time

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

What questions do you have ?

MEAM 520 – September 25, 2014 – Prof. K. J. Kuchenbecker – University of Pennsylvania

Trajectory Planning Questions

1. 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 $q(t_f) = q_f$.

2. We discussed using linear algebra to solve for the coefficients of the cubic polynomial that satisfies the specified conditions. Dr. Kuchenbecker said that the only time one cannot find a solution is when the initial time equals the final time. How could you prove her statement?

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

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

MEAM 520 – September 25, 2014 – Prof. K. J. Kuchenbecker – University of Pennsylvania

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

5. How does the idea of sequencing low-order polynomials such as cubics through multiple via points relate to LSPB and Bang-Bang trajectories?

6. 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 (duration of starting parabola and ending parabola). Here are the equations for the three curves: $q(t) = b_0 + b_1 t + b_2 t^2$ $\quad q(t) = a_0 + a_1 t$ $\quad q(t) = c_0 + c_1 t + c_2 t^2$

Reading Assignment

Within Chapter 5: Path and Trajectory Planning

- Finish reading Section 5.5, which covers Trajectory Planning (pages 186–198)

Deadline: Thursday lecture