MEAM 520

Project 1: PUMA Dance

Katherine J. Kuchenbecker, Ph.D.

General Robotics, Automation, Sensing, and Perception Lab (GRASP)

MEAM Department, SEAS, University of Pennsylvania

GRASP LABORATORY

Lecture 12: October 8, 2013

Homework 4:
Forward Kinematics and DH Parameters

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

September 19, 2013

This paper-based assignment is due on **Thursday, September 26, 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 bin or under the door. Late submissions will be accepted until Sunday, September 29, 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.

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 should be your own work, not copied from any other individual or a solution manual. 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 adapted from the printed version of the textbook, *Robot Modeling and Control* by Spong, Hutchinson, and Vidyasagar (SHV). Please follow the extra clarifications and instructions when provided. Write in pencil, show your work clearly, box your answers, and staple together all pages of your assignment. This assignment is worth a total of 20 points.

1. Custom problem – Kinematics of Baxter *(2 points)*
   Rethink Robotics sells a two-armed manufacturing robot named Baxter. Watch YouTube videos of Baxter (e.g., http://www.youtube.com/watch?v=rjPFqkFyrOY) to learn about its kinematics. Draw a schematic of the serial kinematic chain of Baxter's left arm (the one the woman is touching in the picture below.) Use the book's conventions for how to draw revolute and prismatic joints in 3D.

1

Homework 4
has been graded.

2. Custom Problem – 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.

3. Custom Problem – Interpreting a Transformation Matrix *(2 points)*
Equation (3.24) on page 93 of SHV gives the SCARA manipulator's T_4^0 transformation matrix. What is the practical (geometric) meaning of each of the four columns of this matrix? 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. Keep the intuitive meaning of the elements of these matrices in your mind as you solve the remaining problems in this assignment.

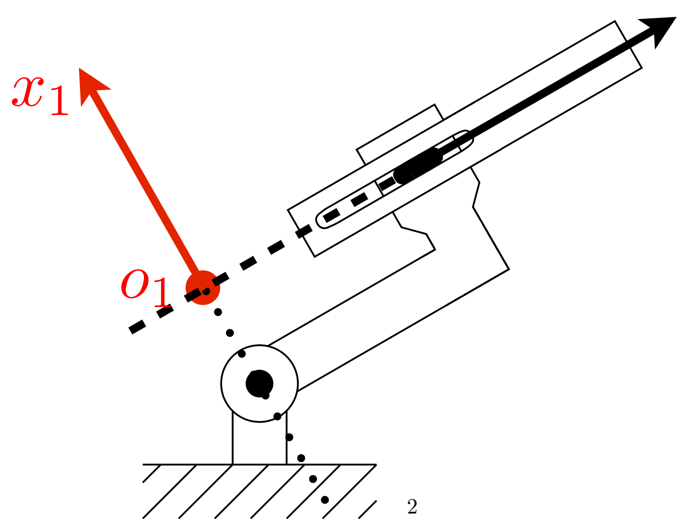
Do the following steps for each of the next three problems:

- Draw a schematic of the robot in its zero configuration.
- Draw your frames on the diagram, following the DH convention.
- Use a superscript star, e.g., θ_1^*, to denote all joint variables.
- Use an arrow labeled with the joint variable name to mark the positive direction for all joint variables on the diagram.
- Use your diagram to create a table of DH parameters for the manipulator.
- Label all DH parameters that you introduce on the diagram.
- Calculate the final transformation matrix.
- Check your work by examining the final transformation matrix to determine whether it gives the answers you expect for simple situations, such as the zero configuration. Fix any problems you uncover.

4. SHV 3-4, page 112 – Forward Kinematics of a Two-Link Planar RP Arm With Offset *(4 points)*
You may choose the zero configuration.

5. SHV 3-7, page 113 – Forward Kinematics of the Three-Link Cartesian Robot *(4 points)*
Use the depicted pose as the zero configuration.

6. SHV 3-6, page 113 – Forward Kinematics of the Three-Link Articulated Robot *(6 points)*
Use the depicted pose as the zero configuration.



2

Common mistakes:

- Drawing Baxter wrong.
- Answering Q3 generically instead of for SCARA.
- Wrong sign on the DH parameters **alpha** and **a**.
- Not marking full extent of dimensions on diagrams.
- Forgetting 90 degree rotation between non-aligned successive x axes.
- Misplacing frame 1 for the planar RP robot.

2. Custom Problem – 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.

3. Custom Problem – Interpreting a Transformation Matrix *(2 points)*
Equation (3.24) on page 93 of SHV gives the SCARA manipulator's T_4^0 transformation matrix. What is the practical (geometric) meaning of each of the four columns of this matrix? 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. Keep the intuitive meaning of the elements of these matrices in your mind as you solve the remaining problems in this assignment.

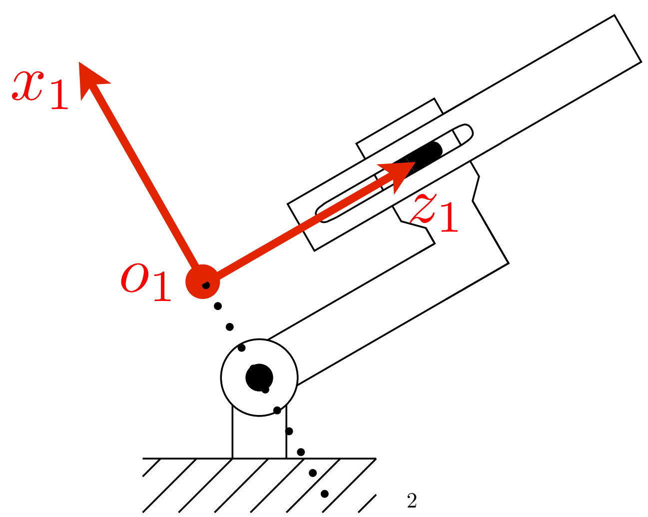
Do the following steps for each of the next three problems:

- Draw a schematic of the robot in its zero configuration.
- Draw your frames on the diagram, following the DH convention.
- Use a superscript star, e.g., θ_1^*, to denote all joint variables.
- Use an arrow labeled with the joint variable name to mark the positive direction for all joint variables on the diagram.
- Use your diagram to create a table of DH parameters for the manipulator.
- Label all DH parameters that you introduce on the diagram.
- Calculate the final transformation matrix.
- Check your work by examining the final transformation matrix to determine whether it gives the answers you expect for simple situations, such as the zero configuration. Fix any problems you uncover.

4. SHV 3-4, page 112 – Forward Kinematics of a Two-Link Planar RP Arm With Offset *(4 points)*
You may choose the zero configuration.

5. SHV 3-7, page 113 – Forward Kinematics of the Three-Link Cartesian Robot *(4 points)*
Use the depicted pose as the zero configuration.

6. SHV 3-6, page 113 – Forward Kinematics of the Three-Link Articulated Robot *(6 points)*
Use the depicted pose as the zero configuration.



2

Common mistakes:

- Drawing Baxter wrong.
- Answering Q3 generically instead of for SCARA.
- Wrong sign on the DH parameters **alpha** and **a**.
- Not marking full extent of dimensions on diagrams.
- Forgetting 90 degree rotation between non-aligned successive x axes.
- Misplacing frame 1 for the planar RP robot.

Homework 5:
PUMA 260

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

September 26, 2013

This assignment is due on **Tuesday, October 1, by midnight (11:59:59 p.m.)** Your code should be submitted via email according to the instructions at the end of this document. Late submissions will be accepted until Thursday, October 3, by midnight (11:59:59 p.m.), but they will be penalized by 10% for each partial or full day late, up to 20%. After the late deadline, no further assignments may be submitted.

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 should 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

We encourage you to do this assignment with a partner. If you prefer, you may also do this assignment individually. If you do this homework with a partner, you may work with anyone you choose, even someone with substantial MATLAB experience. If you are looking for a partner, 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.

My solution for Homework 5

Editor – /Users/kuchenbe/Documents/teaching/meam 520/assignments/05 puma/matlab/puma_robot_kuchenbe.m

EDITOR PUBLISH VIEW

puma_robot_kuchenbe.m

```matlab
%% puma_robot_kuchenbe.m
%
% This Matlab script animates the PUMA 260 robot for Homework 5 in MEAM 520
% at the University of Pennsylvania.


%% SETUP

% Clear all variables from the workspace.
clear all

% Home the console, so we can more easily find any errors that may occur.
home

% My name.
my_name = 'Katherine J. Kuchenbecker (Solution)';

% Set whether to animate the robot's movement. Set this to off if you don't
% want to watch the animation.
pause on;

% Set the length of time that Matlab should pause between positions when
% graphing, if at all, in seconds.
GraphingTimeDelay = 0.001;

% Length of coordinate frame axes, in inches.
alen = 6;


%% ROBOT PARAMETERS AND MOTION

% This problem is about the PUMA 260 robot, a 6-DOF manipulator.

% Define the robot's measurements.  These correspond to the diagram in the
% homework and are constant.
a = 13.0; % inches
b =  2.5; % inches
c =  8.0; % inches
d =  2.5; % inches
e =  8.0; % inches
f =  2.5; % inches

% Load the desired motion.
```

script Ln 5 Col 1

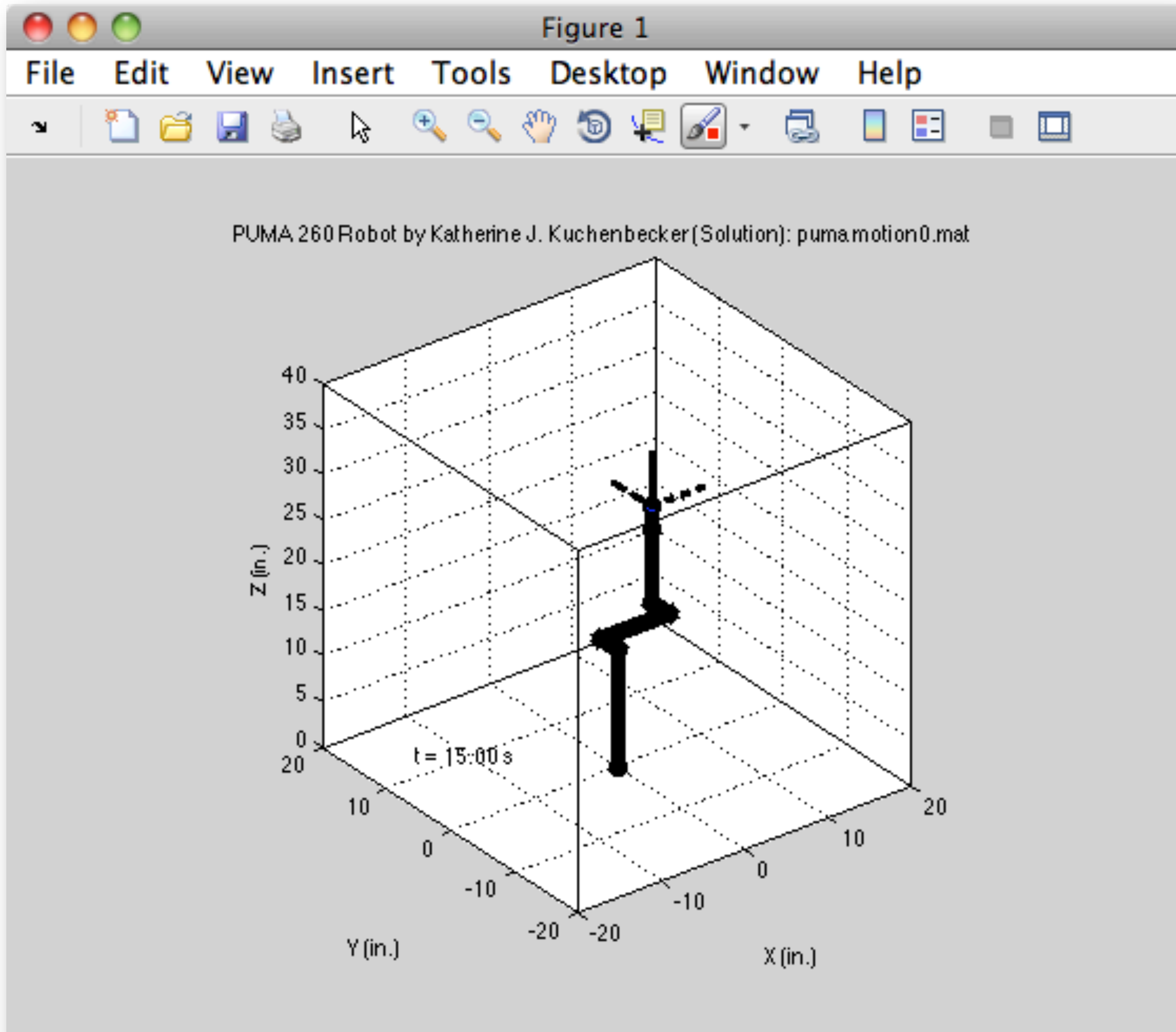


Figure 1
File Edit View Insert Tools Desktop Window Help
PUMA 260 Robot by Katherine J. Kuchenbecker (Solution): puma motion0.mat
Z (in.)
t = 15.00 s
Y (in.)
X (in.)

Questions?

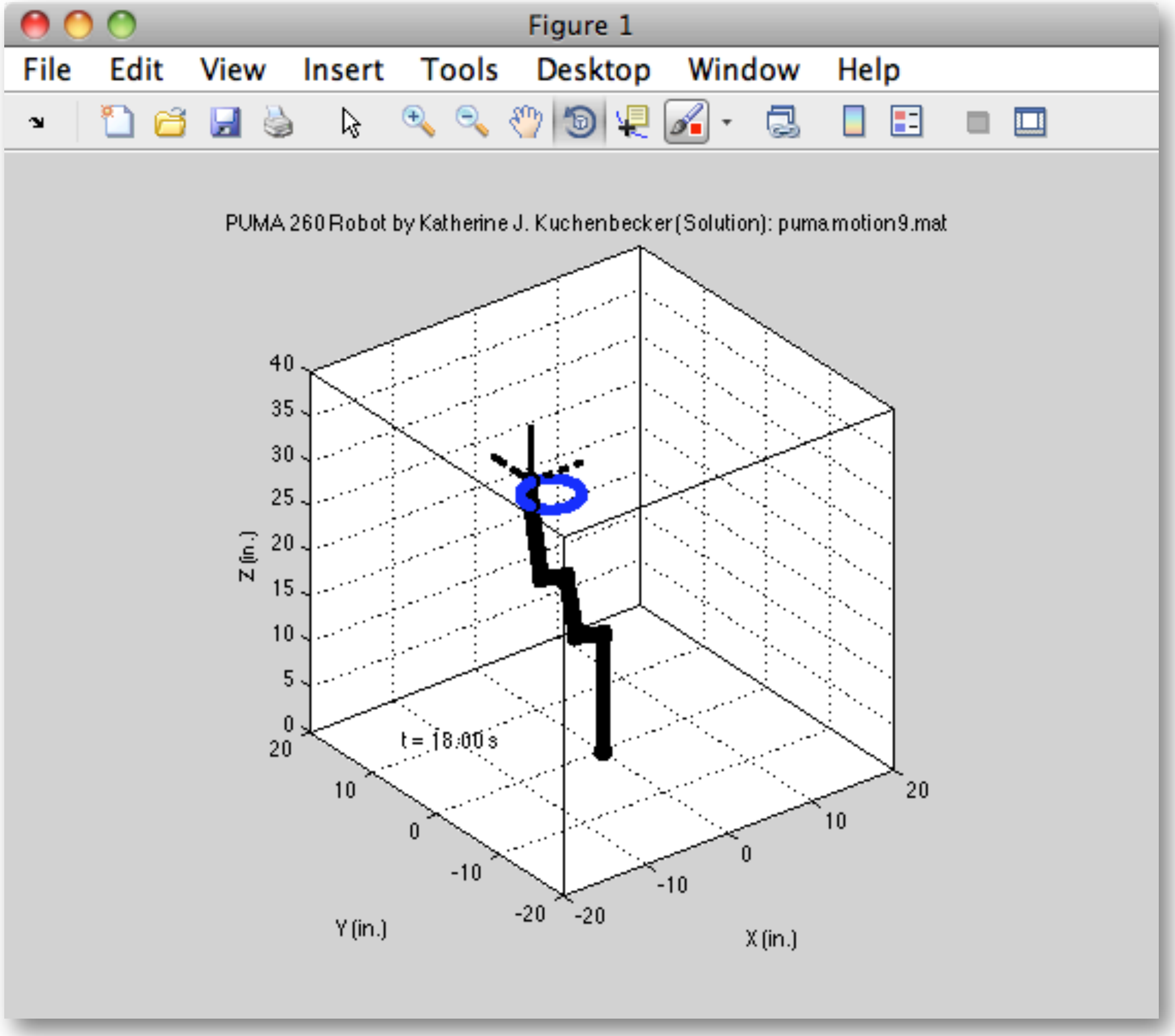


Figure 1
File Edit View Insert Tools Desktop Window Help
PUMA 260 Robot by Katherine J. Kuchenbecker (Solution): pumamotion9.mat
40
35
30
25
20
15
10
5
0
Z (in.)
t = 18.00 s
20
10
0
-10
-20
Y (in.)
-20
-10
0
10
20
X (in.)

MEAM 520 – October 3, 2013 – 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. Will there always be a solution? If no, when does it fail?

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

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.

MEAM 520 – October 3, 2013 – 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.

These would have been good homework questions, but instead I want you to use these concepts in a project....

Project 1

PUMA Dance

Please ask questions!

- You will work in pairs (2 students).
- Because we have 94 students, there will be 47 teams of two.

note stop following 74 views
Actions

Teams for Project 1

Here is a list of team numbers and team members for Project 1. I will keep updating this as more teams are formed, following the guidelines explained in @98.

Team # - Members
101 - John Nappo and Wyatt Shapiro
102 - Jay Davey and Alexander McCraw
103 - Rafael Pelles and Jordan Landis
104 - Vivienne Clayton and Nick Parrotta
105 - Sangyi Cheng and Yike Chen
106 - Lili Gu and Siyao Hu
107 - Gaylord Swaby and Eduardo Garcia
108 - Sanket Chitagopkar and Honnesh Ramachandra
109 - Michael Latimer and Joseph Hill
110 - Jianqiao Li and Jing Tang
111 - Mitch Graves and Antonino Mazzurco
112 - Kristopher Li and Mark Gallagher
113 - Deeksha Yogish and Rahul Ajay Nafde
114 - Charlotte Dean and Solomon Gardner
115 - Michael Rosenman and Prathik Prakash
116 - Matt Novick and Yik Lung Chan
117 - Mabel Zhang and Marcus Goudie
118 - Weiyiping Huang and Daniel Chabolla
119 - Sam Ettinger and Tyler Caron
120 - Sarah Leung and Vishnu Purushothaman Sreenivasan
121 - Nick Pesta and Iason Stamatiadis
122 - Valerie Cohen and Adam Baitch
123 - Zimeng Yang and Bokang Wang

Copyright © 2013 Piazza Technologies, Inc. All Rights Reserved. Privacy Policy Copyright Policy Terms of Use Blog Report Bug!

120 - Sarah Leung and Vishnu Purushothaman Sreenivasan
121 - Nick Pesta and Iason Stamatiadis
122 - Valerie Cohen and Adam Baitch
123 - Zimeng Yang and Bokang Wang
124 - Emily Plumb and Matthew Lisle
125 - Bahram Banisadr and Kate Wessels
126 - Priyanka Shreeniwas Shirsat and Shweta Krishnan
127 - Yixuan Wu and Yu Fu
128 - Dave Kotfis and Sichao Wang
129 - Spyridon Karachalios and Clara Midgley
130 - Krithika Baskaran and Akshitha Sriraman
131 - Addwiteey Chrungoo and Anushree Singh
132 - James Sui and Pete Furlong
133 - Dingwei Zhang and Zhao Liang
134 - John Whitman and Noah Frick
135 - Ya-Yun Lee and Te-Chuan Chou
136 - Jonathan Cousins and Brandon Jennings
137 - Shyamsundar Ramanathan and Vamsikrishna Sonti
138 - Srinivasan Ekambaram and Mariah Clark
139 - Carlie Badder and Kevin Burke
140 - Isabel Siheng Ji and Kent Fremon
141 - Markus Beissinger and Mike Lautman
142 - Sarah Martezian and Kristin Marra
143 - Conor O'Brien and Gareth Cross
144 -
145 -
146 -
147 -

project1

Copyright © 2013 Piazza Technologies, Inc. All Rights Reserved. Privacy Policy Copyright Policy Terms of Use Blog Report Bug!

- People still looking for a partner: consider meeting in the lobby after class.

- Please choose your partner as soon as possible or request a random partner.

- Send your team choice to meam520@seas.upenn.edu with a subject of Project 1 Team: Full Name 1 and Full Name 2

- Each team needs to choose a music clip from the provided library.
- There are 113 options in the library, spanning a range of genres.
- The music came from GarageBand, Naomi, and Evan. We appreciate their music!
- You must choreograph at least of 30 seconds of robot dancing.
- Slower songs are probably easier.
- Follow instructions on Piazza for accessing songs and claiming one for your team.

Music for PUMA Dance

Each project 1 team needs to pick a song for their PUMA Dance. We are providing 113 options, so there should be plenty to choose from.

I posted all of the files in wav format in this Penn Box folder:
https://upenn.box.com/s/674tmis8v5o6lq05qdff

You should be able to log in with your PennKey after you make a free Box account. Then you can browse the files and play them in your browser to hear them.

As explained in the followup discussions below, a student reported they couldn't edit the filenames with just this level of access, so I added everyone in the class as a collaborator on the folder. Box incorrectly assumes your email address is yourpennkey@upenn.edu, so you have to add your actual email address in your account settings and verify it before you'll be able to change the filenames. I'm sorry for this extra step.

Post a follow-up if you have any trouble accessing the music. I can send a collaboration invitation to a different email address if you prefer.

When you find the song you want, claim it by changing its filename to start with your three-digit team number. For example, if I was on team 100 and wanted to claim the song "Buddy G.wav", I would simply rename it "100 Buddy G.wav".

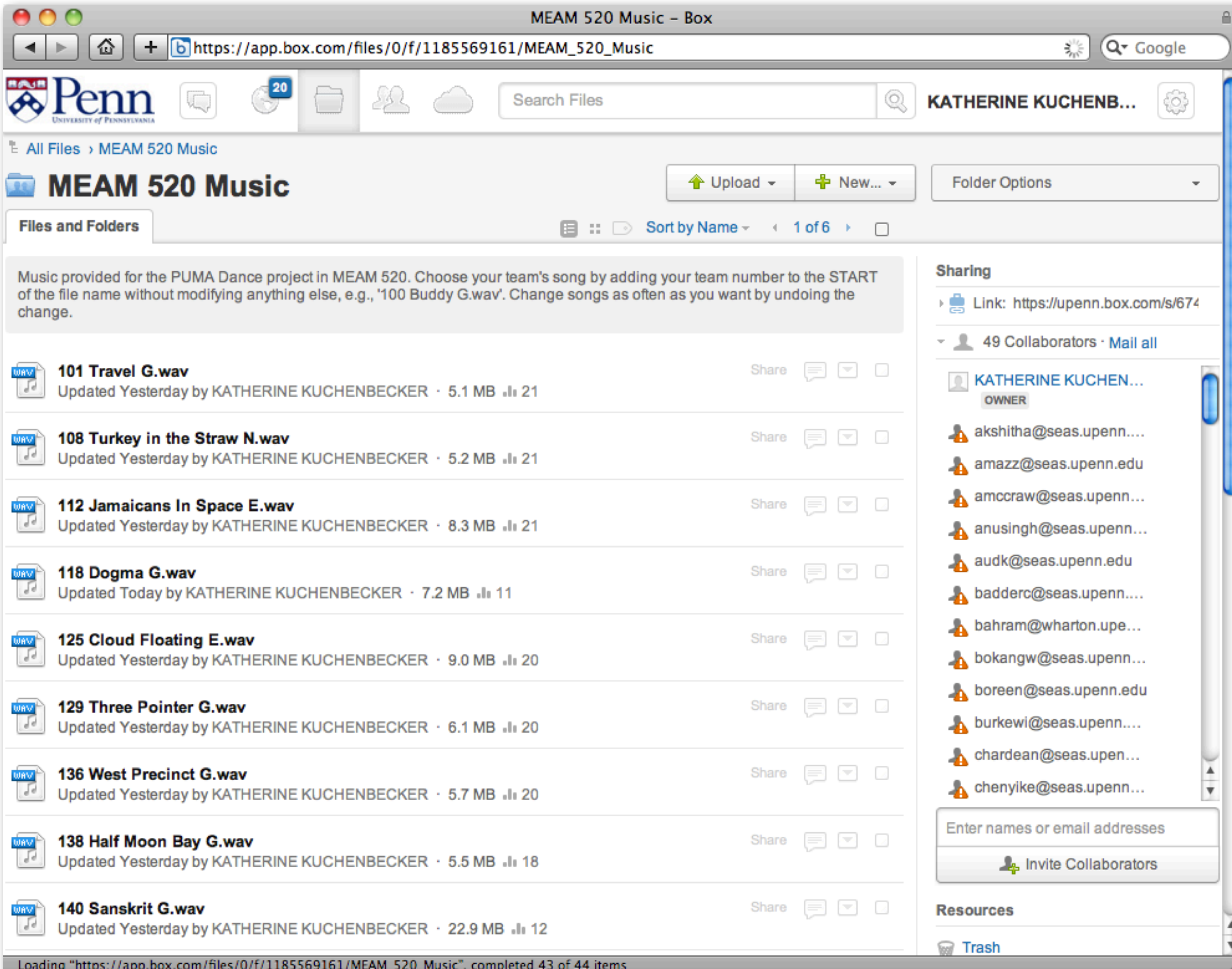
Your team may claim only one song at a time. You may change your choice as often as you like. To release a song, change its title back to the original string.

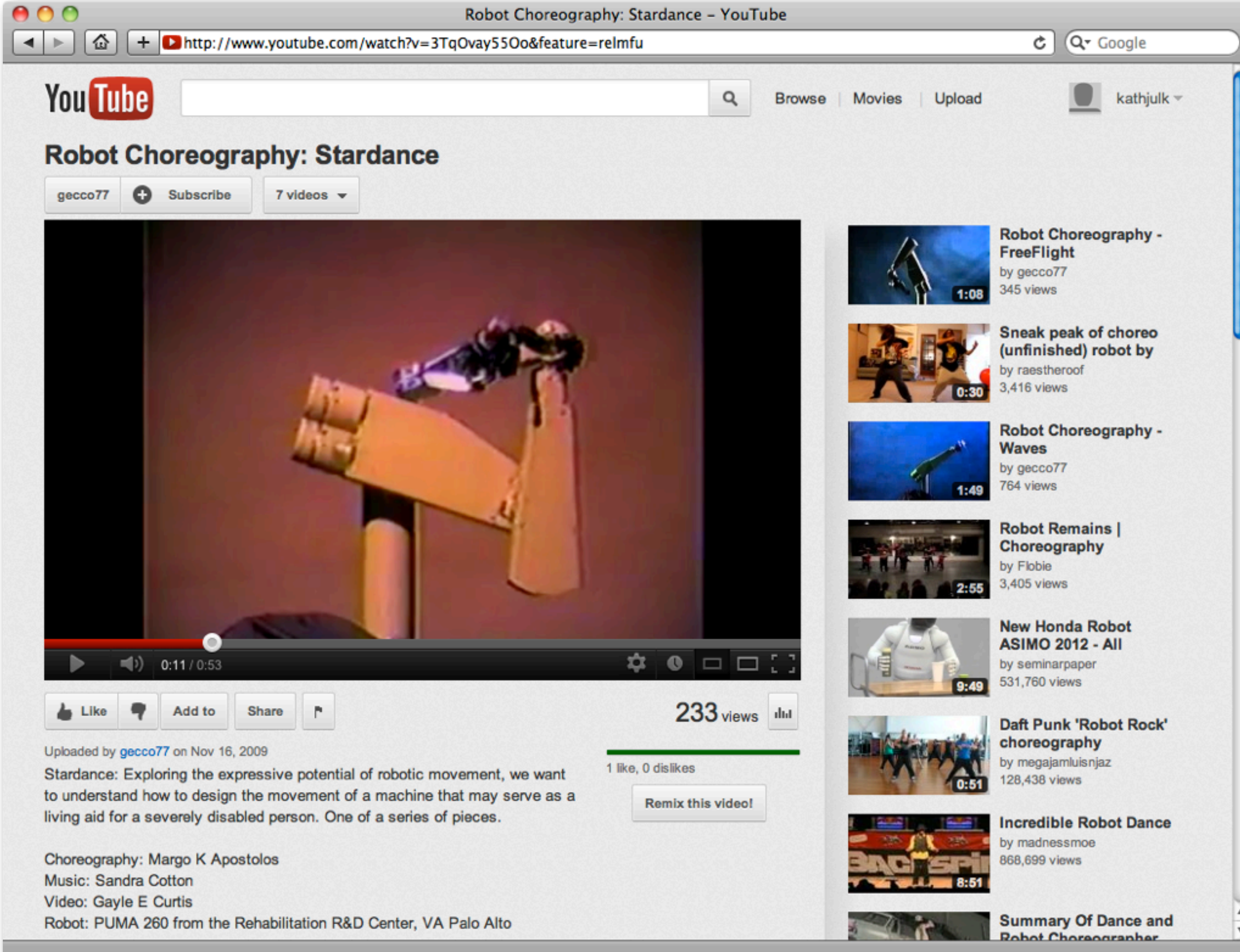
I encourage you to find a song you like, since you will probably listen to it many, many times. The easiest songs to use in this project will be those that are slower paced, since you'll generally need to create fewer via points. The songs do also vary in length. If you find a longer song that you like, you're welcome to choreograph only part of it; the minimum is 30 seconds.

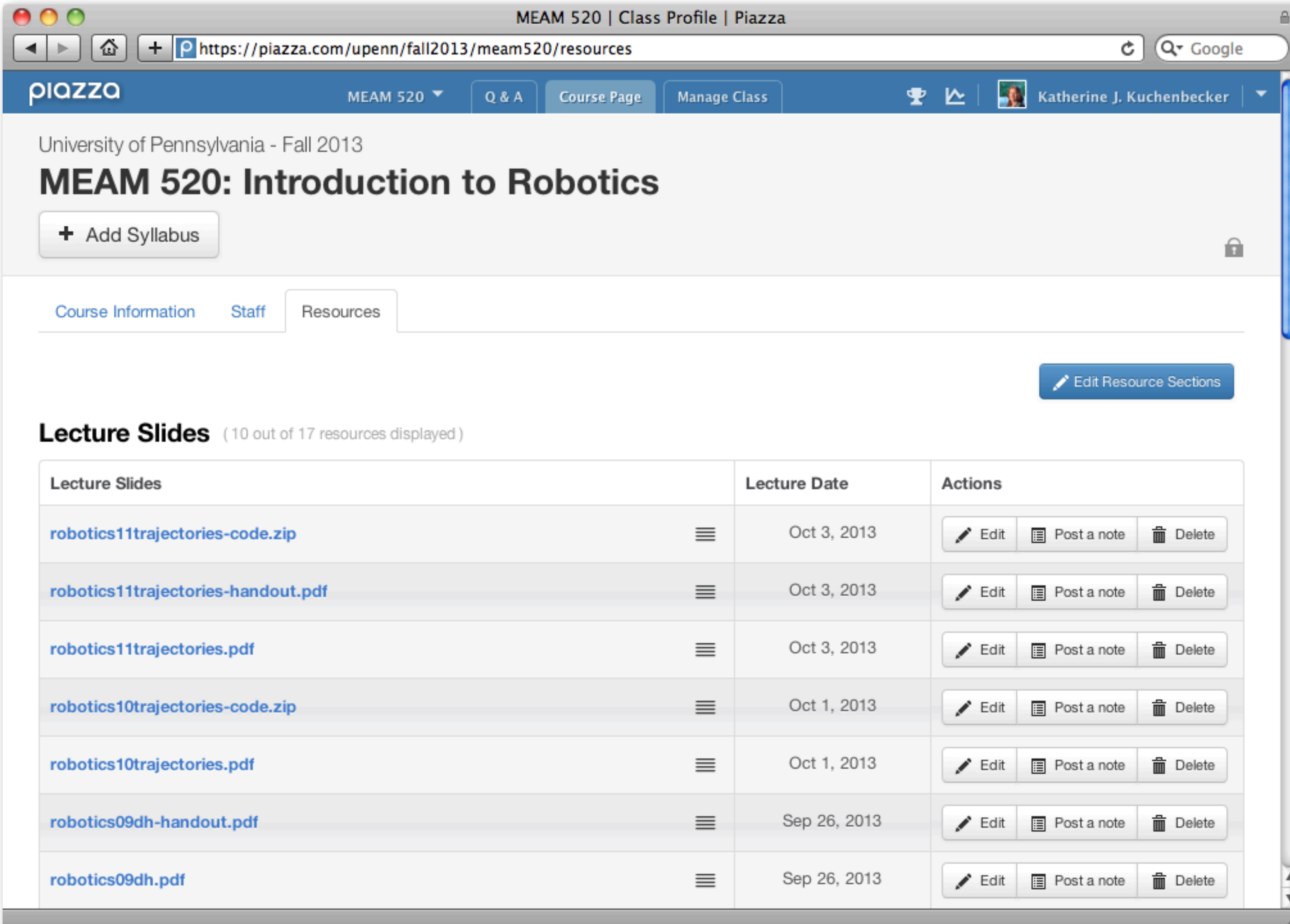
Once you and your partner agree on a song, download it to your computer and start thinking about how the PUMA should move while dancing to your song.

Where did this music come from? Songs whose titles end with a G are from the Apple music-authoring program GarageBand. Songs that end in E are from Evan, and songs that end in N are from our very own TA, Naomi Fitter. When we publish the music videos we need to acknowledge Evan and Naomi for their music; please don't use it for any purpose aside from this class. Many thanks to Naomi for putting together this library.

Copyright © 2013 Piazza Technologies, Inc. All Rights Reserved. Privacy Policy Copyright Policy Terms of Use Blog Report Bug!


MEAM 520 Music – Box
https://app.box.com/files/0/f/1185569161/MEAM_520_Music
Google
Penn
20
Search Files
KATHERINE KUCHENB...
All Files › MEAM 520 Music
MEAM 520 Music
Upload
New...
Folder Options
Files and Folders
Sort by Name
1 of 6
Music provided for the PUMA Dance project in MEAM 520. Choose your team's song by adding your team number to the START of the file name without modifying anything else, e.g., '100 Buddy G.wav'. Change songs as often as you want by undoing the change.
101 Travel G.wav
Updated Yesterday by KATHERINE KUCHENBECKER · 5.1 MB 21
108 Turkey in the Straw N.wav
Updated Yesterday by KATHERINE KUCHENBECKER · 5.2 MB 21
112 Jamaicans In Space E.wav
Updated Yesterday by KATHERINE KUCHENBECKER · 8.3 MB 21
118 Dogma G.wav
Updated Today by KATHERINE KUCHENBECKER · 7.2 MB 11
125 Cloud Floating E.wav
Updated Yesterday by KATHERINE KUCHENBECKER · 9.0 MB 20
129 Three Pointer G.wav
Updated Yesterday by KATHERINE KUCHENBECKER · 6.1 MB 20
136 West Precinct G.wav
Updated Yesterday by KATHERINE KUCHENBECKER · 5.7 MB 20
138 Half Moon Bay G.wav
Updated Yesterday by KATHERINE KUCHENBECKER · 5.5 MB 18
140 Sanskrit G.wav
Updated Yesterday by KATHERINE KUCHENBECKER · 22.9 MB 12
Share
Sharing
Link: https://upenn.box.com/s/674
49 Collaborators · Mail all
KATHERINE KUCHEN...
OWNER
akshitha@seas.upenn....
amazz@seas.upenn.edu
amccraw@seas.upenn...
anusingh@seas.upenn...
audk@seas.upenn.edu
badderc@seas.upenn....
bahram@wharton.upe...
bokangw@seas.upenn...
boreen@seas.upenn.edu
burkewi@seas.upenn....
chardean@seas.upen...
chenyike@seas.upenn...
Enter names or email addresses
Invite Collaborators
Resources
Trash
Loading "https://app.box.com/files/0/f/1185569161/MEAM_520_Music", completed 43 of 44 items


Robot Choreography: Stardance – YouTube
http://www.youtube.com/watch?v=3TqOvay55Oo&feature=relmfu
Google
YouTube
Browse
Movies
Upload
kathjulk
Robot Choreography: Stardance
gecco77
Subscribe
7 videos
0:11 / 0:53
Like
Add to
Share
233 views
Uploaded by gecco77 on Nov 16, 2009
Stardance: Exploring the expressive potential of robotic movement, we want to understand how to design the movement of a machine that may serve as a living aid for a severely disabled person. One of a series of pieces.
Choreography: Margo K Apostolos
Music: Sandra Cotton
Video: Gayle E Curtis
Robot: PUMA 260 from the Rehabilitation R&D Center, VA Palo Alto
1 like, 0 dislikes
Remix this video!
Robot Choreography - FreeFlight
by gecco77
345 views
1:08
Sneak peak of choreo (unfinished) robot by
by raestheroof
3,416 views
0:30
Robot Choreography - Waves
by gecco77
764 views
1:49
Robot Remains | Choreography
by Flobie
3,405 views
2:55
New Honda Robot ASIMO 2012 - All
by seminarpaper
531,760 views
9:49
Daft Punk 'Robot Rock' choreography
by megajamluisnjaz
128,438 views
0:51
Incredible Robot Dance
by madnessmoe
868,699 views
8:51
Summary Of Dance and


MEAM 520 | Class Profile | Piazza
https://piazza.com/upenn/fall2013/meam520/resources
Google
piazza
MEAM 520
Q & A
Course Page
Manage Class
Katherine J. Kuchenbecker
University of Pennsylvania - Fall 2013
MEAM 520: Introduction to Robotics
Add Syllabus
Course Information
Staff
Resources
Edit Resource Sections
Lecture Slides
(10 out of 17 resources displayed)
Lecture Slides
Lecture Date
Actions
robotics11trajectories-code.zip
Oct 3, 2013
robotics11trajectories-handout.pdf
Oct 3, 2013
robotics11trajectories.pdf
Oct 3, 2013
robotics10trajectories-code.zip
Oct 1, 2013
robotics10trajectories.pdf
Oct 1, 2013
robotics09dh-handout.pdf
Sep 26, 2013
robotics09dh.pdf
Sep 26, 2013
Edit
Post a note
Delete

MEAM 520 | Class Profile | Piazza

https://piazza.com/upenn/fall2013/meam520/resources

robotics08dh-handout.pdf	Sep 24, 2013	Edit Post a note Delete
robotics08dh.pdf	Sep 24, 2013	Edit Post a note Delete
robotics07dh-handout.pdf	Sep 19, 2013	Edit Post a note Delete

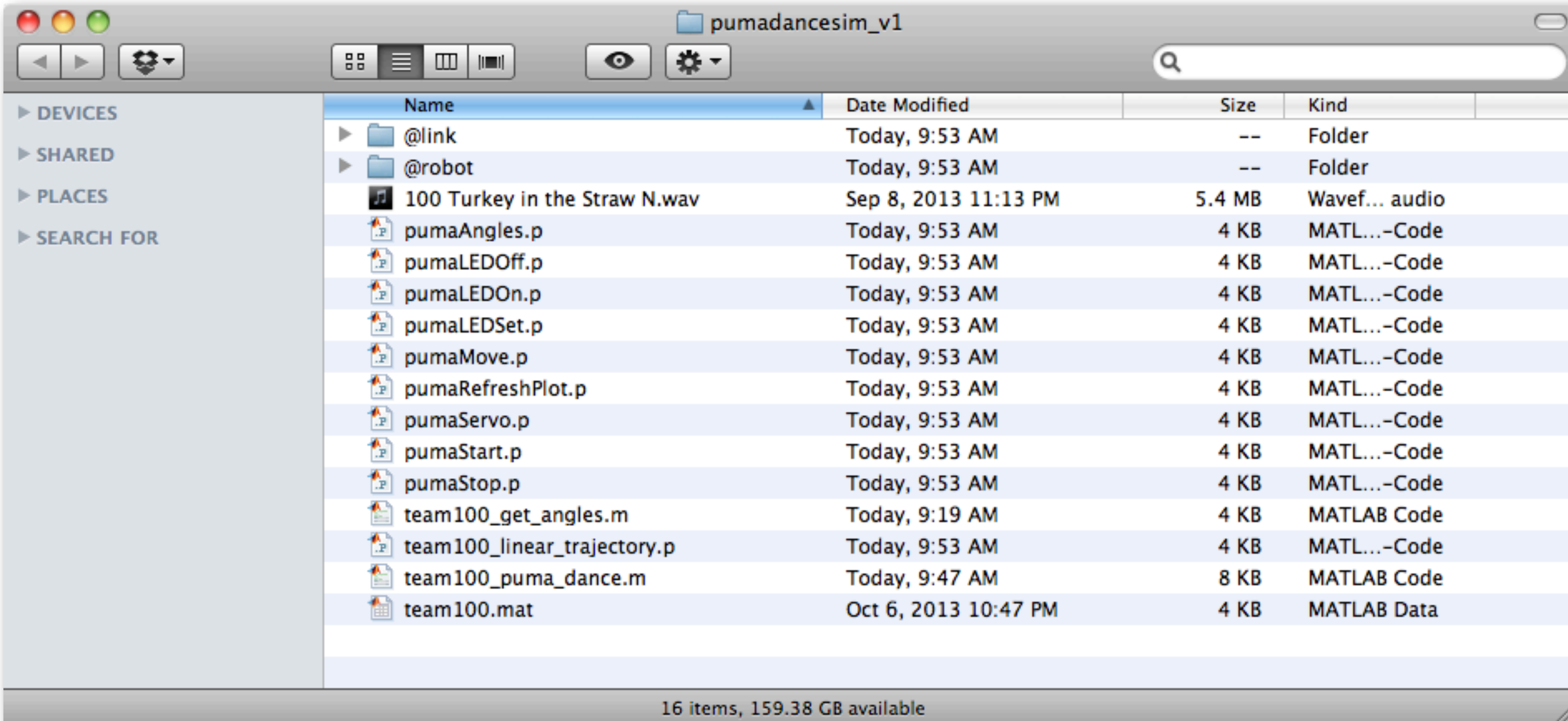
Show all resources

Add Links Add Files

Due by 11:59 p.m. on Sunday, October 20

Homework (10 out of 12 resources displayed)

Homework	Due Date	Actions
pumadancesim_v1.zip	Oct 20, 2013	Edit Post a note Delete
puma_motions.zip	Oct 1, 2013	Edit Post a note Delete
hw05.pdf	Oct 1, 2013	Edit Post a note Delete
hw04.pdf	Sep 26, 2013	Edit Post a note Delete
hw03-v2.pdf	Sep 20, 2013	Edit Post a note Delete
flying_box.mov	Sep 20, 2013	Edit Post a note Delete
flying_box.mat	Sep 20, 2013	Edit Post a note Delete


pumadancesim_v1
DEVICES
SHARED
PLACES
SEARCH FOR
Name
Date Modified
Size
Kind
@link Today, 9:53 AM -- Folder
@robot Today, 9:53 AM -- Folder
100 Turkey in the Straw N.wav Sep 8, 2013 11:13 PM 5.4 MB Wavef... audio
pumaAngles.p Today, 9:53 AM 4 KB MATL...-Code
pumaLEDOff.p Today, 9:53 AM 4 KB MATL...-Code
pumaLEDOn.p Today, 9:53 AM 4 KB MATL...-Code
pumaLEDSet.p Today, 9:53 AM 4 KB MATL...-Code
pumaMove.p Today, 9:53 AM 4 KB MATL...-Code
pumaRefreshPlot.p Today, 9:53 AM 4 KB MATL...-Code
pumaServo.p Today, 9:53 AM 4 KB MATL...-Code
pumaStart.p Today, 9:53 AM 4 KB MATL...-Code
pumaStop.p Today, 9:53 AM 4 KB MATL...-Code
team100_get_angles.m Today, 9:19 AM 4 KB MATLAB Code
team100_linear_trajectory.p Today, 9:53 AM 4 KB MATL...-Code
team100_puma_dance.m Today, 9:47 AM 8 KB MATLAB Code
team100.mat Oct 6, 2013 10:47 PM 4 KB MATLAB Data
16 items, 159.38 GB available

- Starter code is labeled team100.
- You will need to duplicate these files and rename them for your team number.
- Run and modify the team100 starter code to see how everything works.
- **team100_puma_dance.m** is the main script. It initializes the robot simulator, plots the music, plays the music, moves the robot, and plots the results.

/Users/kuchenbe/Desktop/pumadancesim_v1/team100_puma_dance.m

EDITOR PUBLISH VIEW

```matlab
%% PUMA Dance
%
% Starter code by Katherine J. Kuchenbecker
% MEAM 520 at the University of Pennsylvania


%% Clean up

% Clear all variables and functions.  You should do this before calling any PUMA functions.
clear all

% Move the cursor to the top of the command window so new text is easily seen.
home


%% Definitions

% Define team number.
teamnumber = 100;

% Define student names.
studentnames = 'First Name and Second Name';

% Load the dance file from disk.
load team100

% Pull the list of via point times out of the dance matrix for use below.
tvia = dance(:,1);

% Initialize the function that calculates angles.
team100_get_angles

% Define music filename (without team number).
musicfilename = 'Turkey in the Straw N';


%% Music

% Load the piece of music for the robot to dance to.
[y,Fs] = audioread([num2str(teamnumber) ' ' musicfilename '.wav']);

% Calculate the duration of the music.
musicduration = (length(y)-1)/Fs;
```

script Ln 22 Col 16

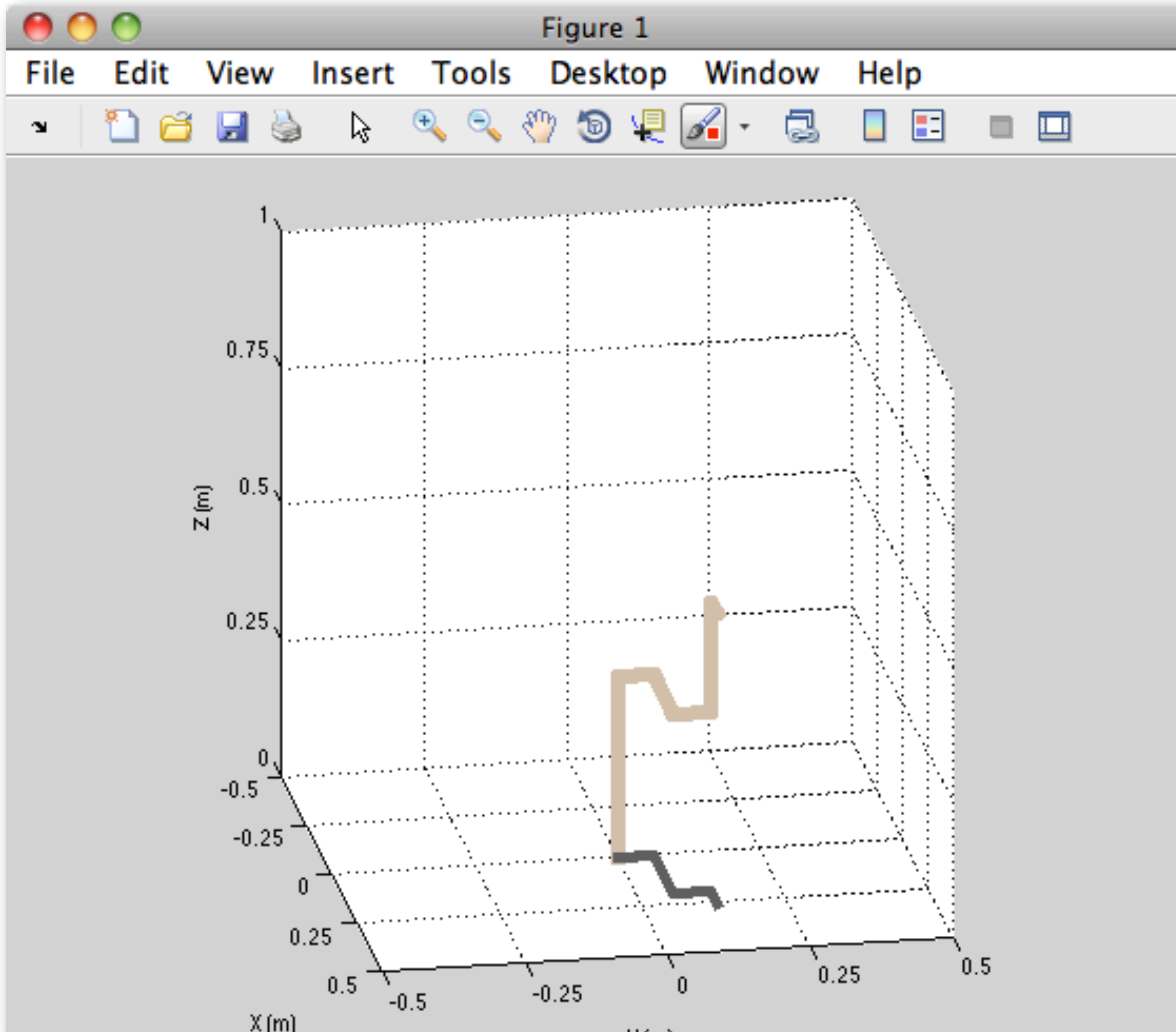


Figure 1
File Edit View Insert Tools Desktop Window Help
1
0.75
0.5
0.25
0
Z (m)
-0.5
-0.25
0
0.25
0.5
X (m)
-0.5
-0.25
0
0.25
0.5
Y (m)

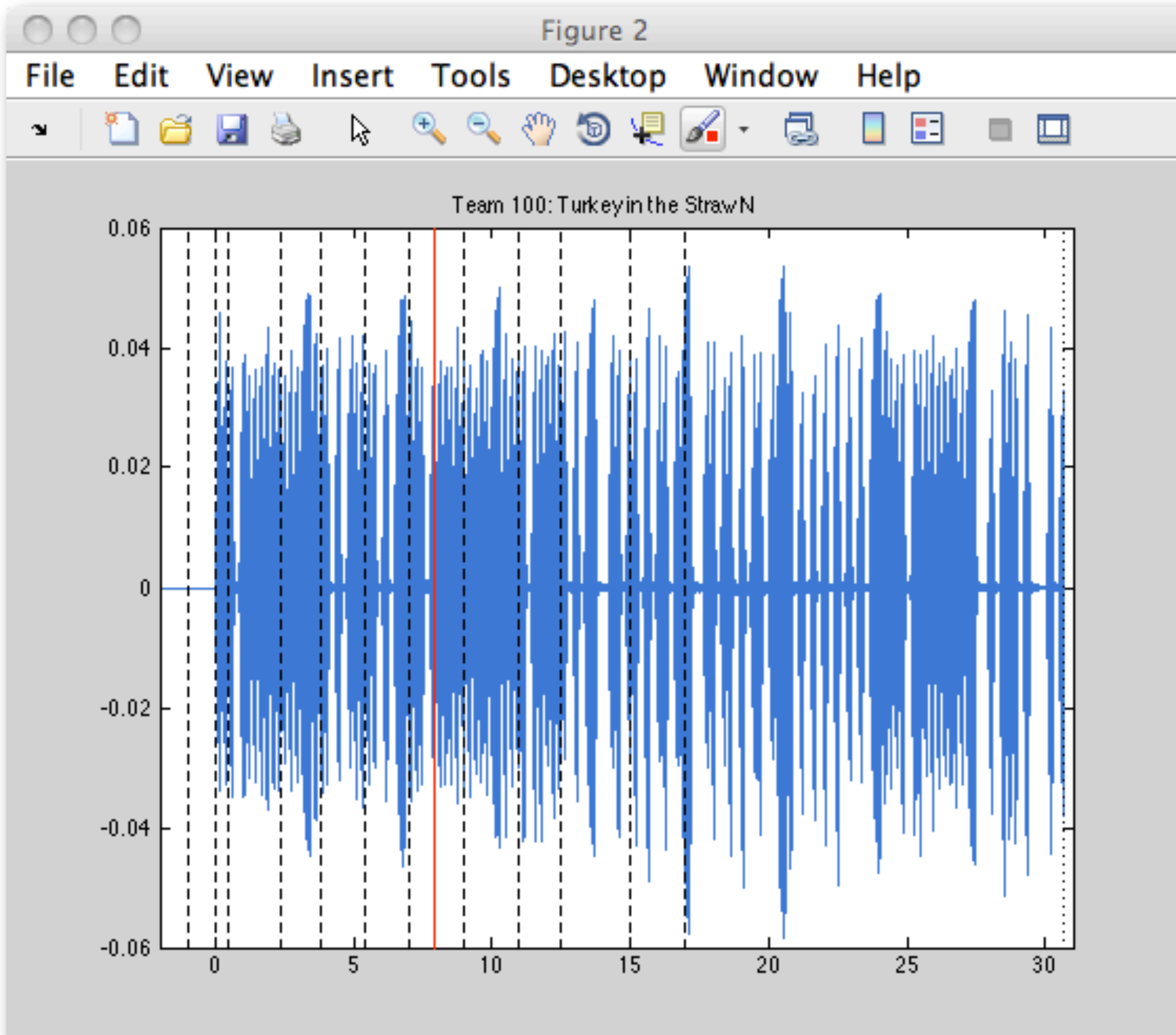


Figure 2
File Edit View Insert Tools Desktop Window Help
Team 100: Turkey in the Straw N
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
0 5 10 15 20 25 30

- The music always starts at t = 0 seconds.
- Negative values of time are for the robot to get into its starting position.
- Change the segment of the dance to be performed by modifying **tstart** and **tstop** on lines 59 and 60 of starter code.
- Starting and stopping times are marked on music plot by black dotted lines.

/Users/kuchenbe/Documents/teaching/meam 520/projects/1 puma dance/pumadancesim_v1/team100_puma_dance.m

EDITOR PUBLISH VIEW

```
% Define music filename (without team number).
musicfilename = 'Turkey in the Straw N';


%% Music

% Load the piece of music for the robot to dance to.
[y,Fs] = audioread([num2str(teamnumber) ' ' musicfilename '.wav']);

% Calculate the duration of the music.
musicduration = (length(y)-1)/Fs;

% Calculate the duration of silence at the start of the dance.
silenceduration = abs(min(tvia));

% Create a time vector for the entire piece of music, for use in plotting.
t = ((min(tvia):(1/Fs):musicduration))';

% Pad the start of the music file with zeros for the silence.
y = [zeros(length(t)-length(y),2); y];


%% Choose duration

% Set the start and stop times of the segment we want to test.
% To play the entire dance, set tstart = t(1) and tstop = t(end).
tstart = t(1);
tstop = t(end);

% Select only the part of the music that we want to play right now, from
% tstart to tstop.
yplay = y(1+round(Fs*(tstart - t(1))):round(Fs*(tstop-t(1))),:);

% Put this snippet into an audio player so we can listen to it.
music = audioplayer(yplay,Fs);


%% Plot music

% Pull first audio channel and downsample it for easier display.
factordown = 30;
ydown = downsample(y(:,1),factordown);
```

script | Ln 59 Col 14

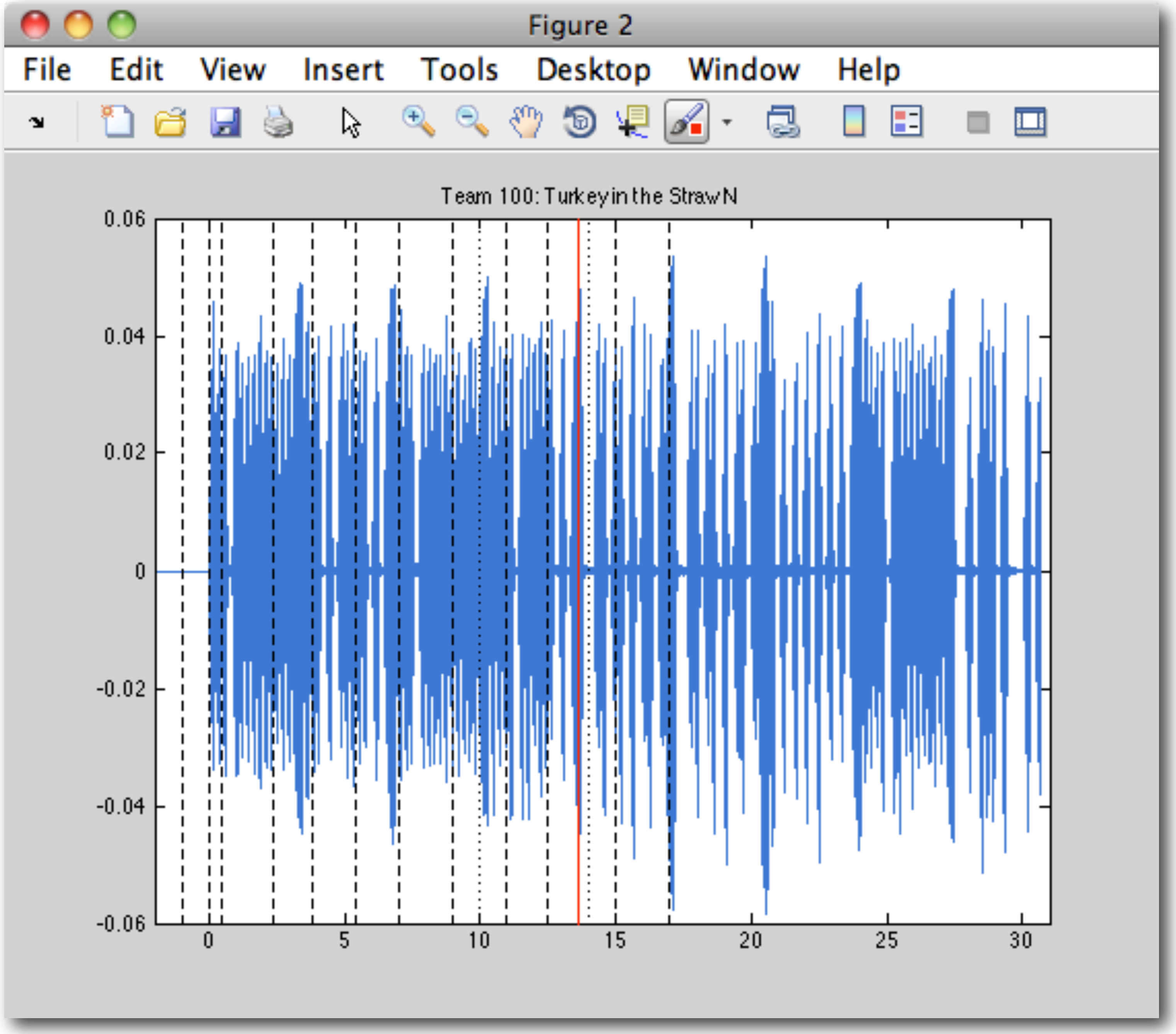
/Users/kuchenbe/Documents/teaching/meam 520/projects/1 puma dance/pumadancesim_v1/team100_puma_dance.m

EDITOR PUBLISH VIEW

```
% Define music filename (without team number).
musicfilename = 'Turkey in the Straw N';


%% Music

% Load the piece of music for the robot to dance to.
[y,Fs] = audioread([num2str(teamnumber) ' ' musicfilename '.wav']);

% Calculate the duration of the music.
musicduration = (length(y)-1)/Fs;

% Calculate the duration of silence at the start of the dance.
silenceduration = abs(min(tvia));

% Create a time vector for the entire piece of music, for use in plotting.
t = ((min(tvia):(1/Fs):musicduration))';

% Pad the start of the music file with zeros for the silence.
y = [zeros(length(t)-length(y),2); y];


%% Choose duration

% Set the start and stop times of the segment we want to test.
% To play the entire dance, set tstart = t(1) and tstop = t(end).
tstart = 10;
tstop = 14;

% Select only the part of the music that we want to play right now, from
% tstart to tstop.
yplay = y(1+round(Fs*(tstart - t(1))):round(Fs*(tstop-t(1))),:);

% Put this snippet into an audio player so we can listen to it.
music = audioplayer(yplay,Fs);


%% Plot music

% Pull first audio channel and downsample it for easier display.
factordown = 30;
ydown = downsample(y(:,1),factordown);
```

script Ln 59 Col 12

What are the black dashed lines?
timed via points



- Your team will design timed via points for the robot move through.
- Each via point should be the full state of the robot (six joint angles and six joint velocities) for a specific time in the song.
- You will write code to calculate trajectories between successive timed via points.
- **team100_get_angles.m** is the function that takes the current time and returns the joint angles that the robot should have.

/Users/kuchenbe/Documents/teaching/meam 520/projects/1 puma dance/pumadancesim_v1/team100_get_angles.m

EDITOR PUBLISH VIEW

```matlab
function thetas = team100_get_angles(t)

% Define dance and all other variables needed multiple times to be persistent.
persistent tvia thetavia thetadotvia trajectorytypevia

% This function is initialized by calling it with no argument.
if (nargin == 0)

    % Load the team's dance file from disk.  It contains the variable dance.
    load team100

    % Pull the list of via point times out of the dance matrix.
    tvia = dance(:,1);

    % Pull the list of joint angles out of the dance matrix.
    thetavia = dance(:,2:7);

    % Pull the list of joint angle velocities out of the dance matrix.
    thetadotvia = dance(:,8:13);

    % Pull the list of trajectory types out of the dance matrix.
    trajectorytypevia = dance(:,14);

    % Return from initialization.
    return

end

% Determine which trajectory we should be executing.  Assuming the via
% point times are monotonically increasing, we look for the first via point
% time that is greater than the current time.  Subtract 1 to get to the
% index of the via point that starts this trajectory.
traj = find(t < tvia,1) - 1;

% Select the correct trajectory types.
switch (trajectorytypevia(traj))
    case 0
        % Linearly interpolate between the via points using a pcoded file.
        % You may not use this pcoded file in your solution.
        thetas = team100_linear_trajectory(t,tvia(traj),tvia(traj+1),thetavia(traj,:)',thetavia(tra
    case 1
        % Just return the joint angles of the via point that starts this
```

team100_get_angles | Ln 1 Col 40

/Users/kuchenbe/Documents/teaching/meam 520/projects/1 puma dance/pumadancesim_v1/team100_get_angles.m

EDITOR PUBLISH VIEW

```
        load team100

        % Pull the list of via point times out of the dance matrix.
        tvia = dance(:,1);

        % Pull the list of joint angles out of the dance matrix.
        thetavia = dance(:,2:7);

        % Pull the list of joint angle velocities out of the dance matrix.
        thetadotvia = dance(:,8:13);

        % Pull the list of trajectory types out of the dance matrix.
        trajectorytypevia = dance(:,14);

        % Return from initialization.
        return

    end

    % Determine which trajectory we should be executing.  Assuming the via
    % point times are monotonically increasing, we look for the first via point
    % time that is greater than the current time.  Subtract 1 to get to the
    % index of the via point that starts this trajectory.
    traj = find(t < tvia,1) - 1;

    % Select the correct trajectory types.
    switch (trajectorytypevia(traj))
        case 0
            % Linearly interpolate between the via points using a pcoded file.
            % You may not use this pcoded file in your solution.
            thetas = team100_linear_trajectory(t,tvia(traj),tvia(traj+1),thetavia(traj,:)',thetavia(traj
        case 1
            % Just return the joint angles of the via point that starts this
            % trajectory.  This is not a good interpolation method.
            thetas = thetavia(traj,:)';
        case 2
            % You should define and write other interpolation methods.
        otherwise
            error(['Unknown trajectory type: ' num2str(trajectorytypevia(traj))])
    end
```

team100_get_angles   Ln 36   Col 33

- **team100.mat** contains the variable **dance**, which defines the robot's dance.
- **dance** has fourteen columns and as many rows as there are via points in the dance.
  - Column 1 is the time in seconds.
  - Columns 2 through 7 are PUMA joint angles in radians.
  - Columns 8 through 13 are PUMA joint velocities in radians per second.
  - Column 14 is the trajectory type (integer).
- You can augment this structure if you want.

time (s)

theta1 to theta6 (rad)

thetadot1 to thetadot6 (rad/s)

trajectory type

Variables – dance

PLOTS | VARIABLE | VIEW

dance

dance <14x14 double>

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	-2	0	0	0	0	-1.5708	0	0	0	0	0	0	0	0
2	-1	0	0.2000	0	0	-1.5708	0	0	0	0	0	0	0	0
3	0	0	0.2000	0	0	-1.5708	0	0	0	0	0	0	0	0
4	0.5000	0.2000	0.2000	0	0	-1.5708	0	0	0	0	0	0	0	0
5	2.3500	-0.5000	0.2000	0	0	-1.5708	0	0	0	0	0	0	0	0
6	3.8000	0.8000	0.2000	0	0	-1.5708	0	0	0	0	0	0	0	0
7	5.4000	-0.5000	0.2000	0	0	-1.5708	0	0	0	0	0	0	0	0
8	7	0.8000	0.2000	0	0	-1.5708	0	0	0	0	0	0	0	0
9	9	0	0.2000	0	0	-1.5708	0	0	0	0	0	0	0	0
10	11	0	0.6000	0	0	-1.5708	0	0	0	0	0	0	0	0
11	12.5000	0	0	0	0	-1.5708	0	0	0	0	0	0	0	0
12	15	0	-1	0	0	0	0	0	0	0	0	0	0	0
13	17	0	0	0	0	-1.5708	0	0	0	0	0	0	0	0
14	31	0	0	0	0	-1.5708	0	0	0	0	0	0	0	0

First via point MUST be at a negative or zero time.

First via point angles MUST be the home position.

Robot should probably be stationary before music starts.

- The starter code includes **team100_linear_trajectory.p**, a p-coded function that does linear interpolation to get the robot into its starting pose and demonstrate a simple dance.
- You may not use this function in your PUMA dance (except for starting pose.)
- Your dance should include 3 or more types of trajectories.
    - At least one cubic polynomial.
    - At least one LSPB.

- You should move all of the robot's joints at some point in the dance.
- All joint angle limits must be obeyed.
- The robot must not collide with the table or itself. The simulator does not check for these collisions, so you need to watch.
- The maximum joint velocity is 1 rad/s, and the maximum joint angle change is 0.5 radians. These might be increased.
- If running into joint velocity and angle change limits, use **pumaMove** instead of **pumaServo**. (Won't work on hardware.)

- The dance routine should relate to your chosen music clip. It can be beautiful, funny, sad, interesting, bold, etc.

- The dance should be repeatable (deterministic not random).

- When your robot finishes dancing, your script must plot the joint angle history and joint velocity history versus time for visual examination.

- Time and joint angles are already being stored in **thistory** and **thetahistory**.

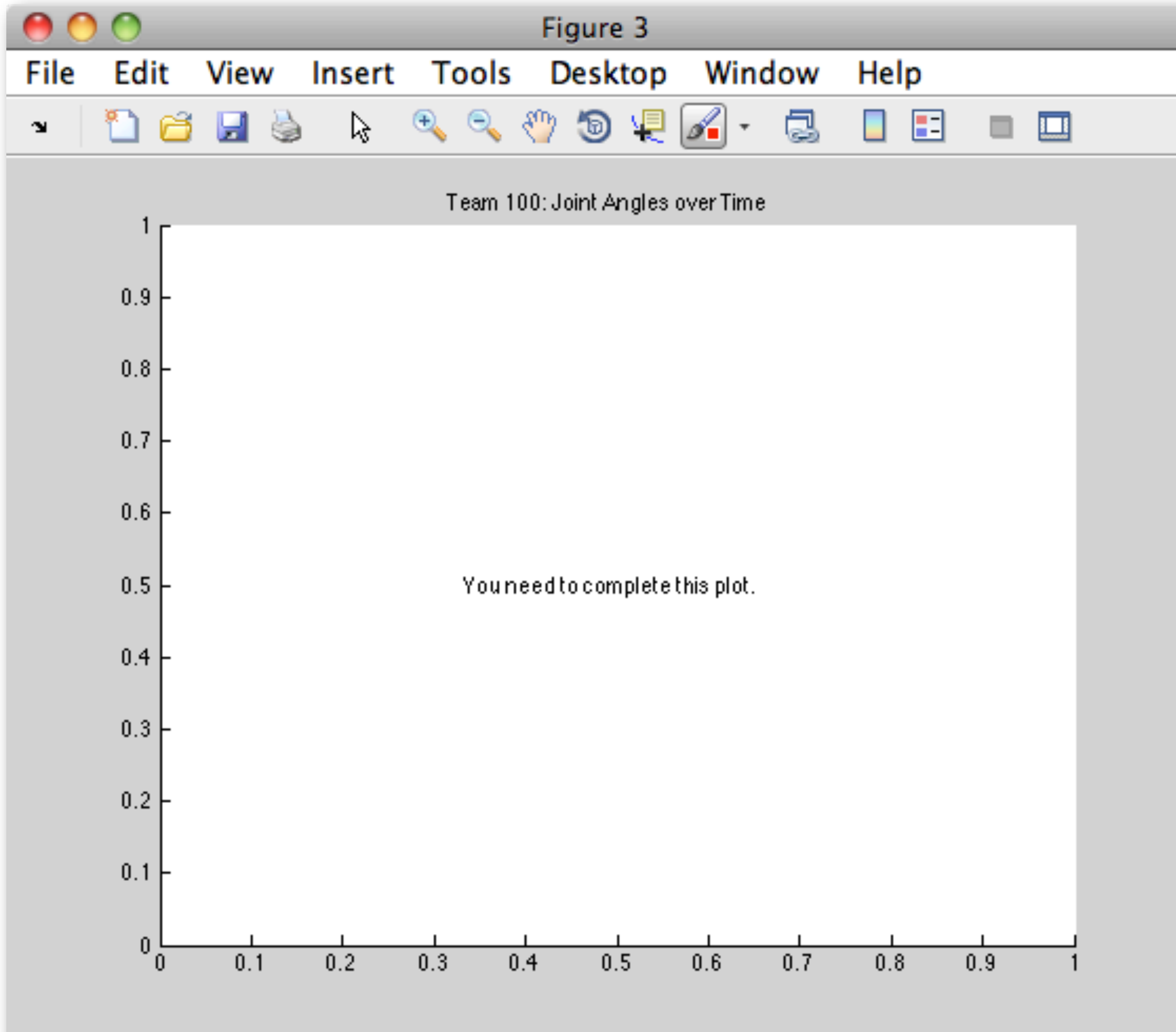


Figure 3
File Edit View Insert Tools Desktop Window Help
Team 100: Joint Angles over Time
You need to complete this plot.
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

- Project 1 is due by 11:59 p.m. on Sunday, October 20 (the Sunday after fall break).
- Late by 11:59 p.m. on Monday, Oct. 21, with a 10% penalty per day. But don't submit late.
- Submit via email to meam520@seas.upenn.edu
- Make the subject Project 1: Team 1XX
- Attach your correctly named MATLAB files (team1XX_puma_dance.m, team1XX.mat, etc.) to the email. Do not put them in a zip file or include any other attachments. All files should start with team1XX_....
- Optionally include any comments you have about the project, and send the email.

- I expect all teams to do very well on this project.
- Getting stuck? Post on Piazza or come to OH.
- Find a bug in simulator? Post on Piazza. I will release new versions as needed.
- We will check submissions as they come in.
- Once your dance meets the requirements, your team will get trained to run the robot and film a music video.
- We'll post all the PUMA dances online.

Recommended Strategy

- FIRST program joint angle and joint velocity plotting.
  - Can put all angles and angular velocities on the same plot or on different plots.
  - Try putting all on different axes using **subplot**.
  - Try the command **linkaxes**.
  - Maybe also plot joint angle limits.
  - Maybe also plot via point times and angles.
- Use this plot for debugging.

Recommended Strategy

- SECOND pick your via point times.
- Ideas on how to do this?
  - Trial and error (very frustrating).
  - Look at the music in another program.
  - Analyze the music in MATLAB.
  - Run an experiment where you hit a key or click the mouse while listening to the music to naturally capture good via point times.
  - Other ideas?
- Use **pumaMove** for prototyping.

Recommended Strategy

- THIRD pick interesting poses for the robot.
- Use your HW5 code or our simulator to test out various poses and find ones you like before viewing them in the full dance routine.
- Ideas on how to pick poses?
  - Manual selection (not too frustrating).
  - Try to mimic human arm or body motions.
  - Random sampling in joint space.
  - Capture from real robot? (KJK looking into this.)
  - Make small model and move it around.

Recommended Strategy

- FOURTH program at least three trajectory interpolation schemes.
  - Must include cubic polynomial.
  - Must include linear segment with polynomial blends.
  - At least one other type.
- Consider pre-calculating the curve parameters so you don't have to solve for them every time you call **team1xx_get_joint_angles(tnow)**.
- FIFTH finetune everything to make a great dance.

Important Tips

- Resist the urge to switch to another song.
  - The grass is always greener on the other side.
- You need a recent version of MATLAB to use the **audioread()** and **audioplayer()** functions we're using for handling and playing the music.
- Press **control-c** to stop the robot's dance.
- Press **control-c** again or run **stop(music)** to stop the song from playing.

What questions do you have ?