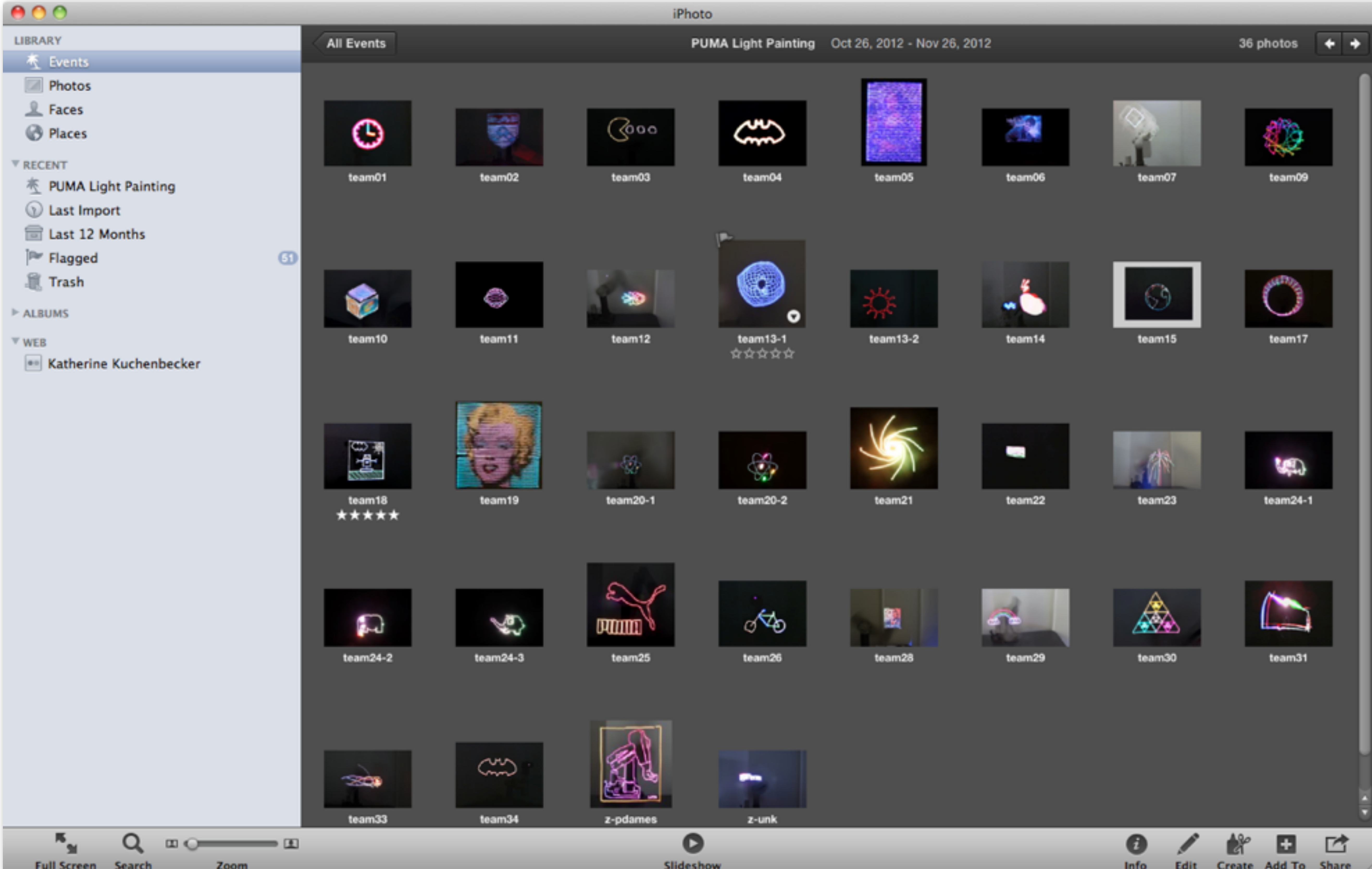
MEAM 520

PHANToM

Katherine J. Kuchenbecker, Ph.D.
General Robotics, Automation, Sensing, and Perception Lab (GRASP)
MEAM Department, SEAS, University of Pennsylvania

Lecture 19: November 27, 2012

Project 1 : PUMA Light Painting


iPhoto
LIBRARY
Events
Photos
Faces
Places
RECENT
PUMA Light Painting
Last Import
Last 12 Months
Flagged
51
Trash
ALBUMS
WEB
Katherine Kuchenbecker
All Events
PUMA Light Painting
Oct 26, 2012 - Nov 26, 2012
36 photos
team01
team02
team03
team04
team05
team06
team07
team09
team10
team11
team12
team13-1
team13-2
team14
team15
team17
team18
team19
team20-1
team20-2
team21
team22
team23
team24-1
team24-2
team24-3
team25
team26
team28
team29
team30
team31
team33
team34
z-pdames
z-unk
Full Screen
Search
Zoom
Slideshow
Info
Edit
Create
Add To
Share

Homework 5: Input/Output Calculations for a Real Robot

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

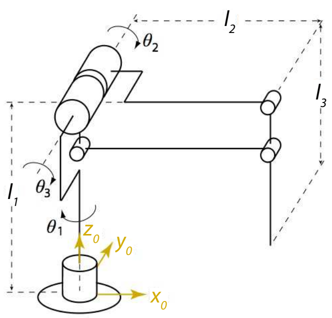
November 13, 2012

This assignment is due on **Tuesday, November 20**, by 5:00 p.m. sharp. You should aim to turn the paper part 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. Late submissions will be accepted until 5:00 p.m. on Wednesday, November 21, but they will be penalized by 25%. After that 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 a peer or a solution manual.

SensAble Phantom Premium 1.0 *(60 points)*

This entire assignment is focused on a particular robot – the SensAble Phantom Premium 1.0. As shown in the photo below left, the Phantom is an impedance-type haptic interface with three actuated rotational joints. Designed to be lightweight, stiff, smooth, and easily backdrivable, this type of haptic interface enables a human user to interact with a virtual environment or control the movement of a remote robot through the movement of their fingertip while simultaneously feeling force feedback.



A thimble is attached to the tip of the robot via a passive non-encoded three-axis gimbal to allow the user to move the robot around while freely rotating their fingertip. As shown in the diagram above right, the Phantom haptic device looks similar to the standard RRR articulated manipulator base, but it uses a unique four-bar mechanism to co-locate the shoulder and elbow joints while also keeping the upper arm and forearm in the plane that intersects the axis of the waist joint.

Each of the four questions below includes both a written explanation and the programming of a specific Matlab function. For the paper parts, write in pencil, show your work clearly, $\boxed{\text{box your answers}}$, and staple your pages together. For the programming, download the starter code from this assignment's page on the class wiki, change all function and file names to include your PennKey, comment your code, and follow the instructions at the end of this document to submit all of your Matlab files.

Solutions to Homework 5
MEAM 520
Introduction to Robotics
University of Pennsylvania
Professor Kuchenbecker
Fall 2012

On reserve in engineering library.

Encoder counts to motor angles.

Encoders are HEDM 5500 B02 : 1000 lines/ ~~pulses~~ per revolution

Pulses go through a quadrature decoder chip.

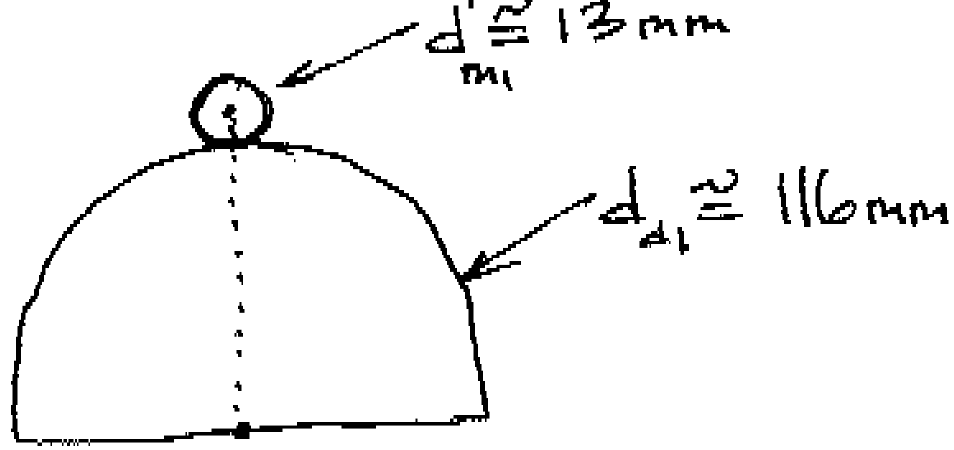
$$\underset{\text{radians}}{\Theta_{mi}} = \underset{\text{counts}}{Q_i} \cdot \frac{1\,\text{pulse}}{4\,\text{counts}} \cdot \frac{1\,\text{revolution}}{1000\,\text{pulses}} \cdot \frac{2\pi\,\text{radians}}{1\,\text{revolution}}$$

where $i = 1, 2, 3$

$$\boxed{\Theta_{mi} = Q_i \cdot \frac{2\pi\,\text{radians}}{4000\,\text{counts}}}$$

Motor 1

view from top.

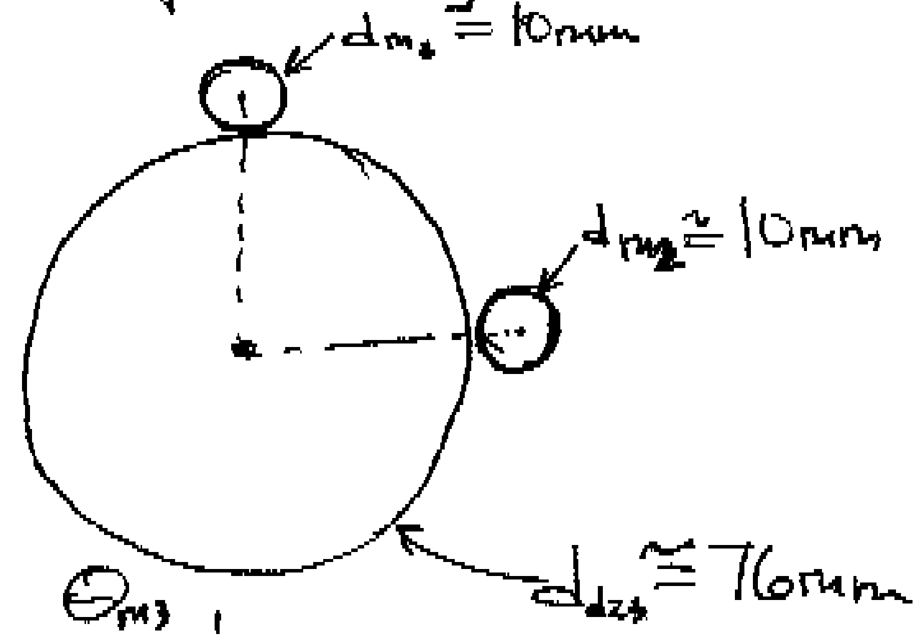


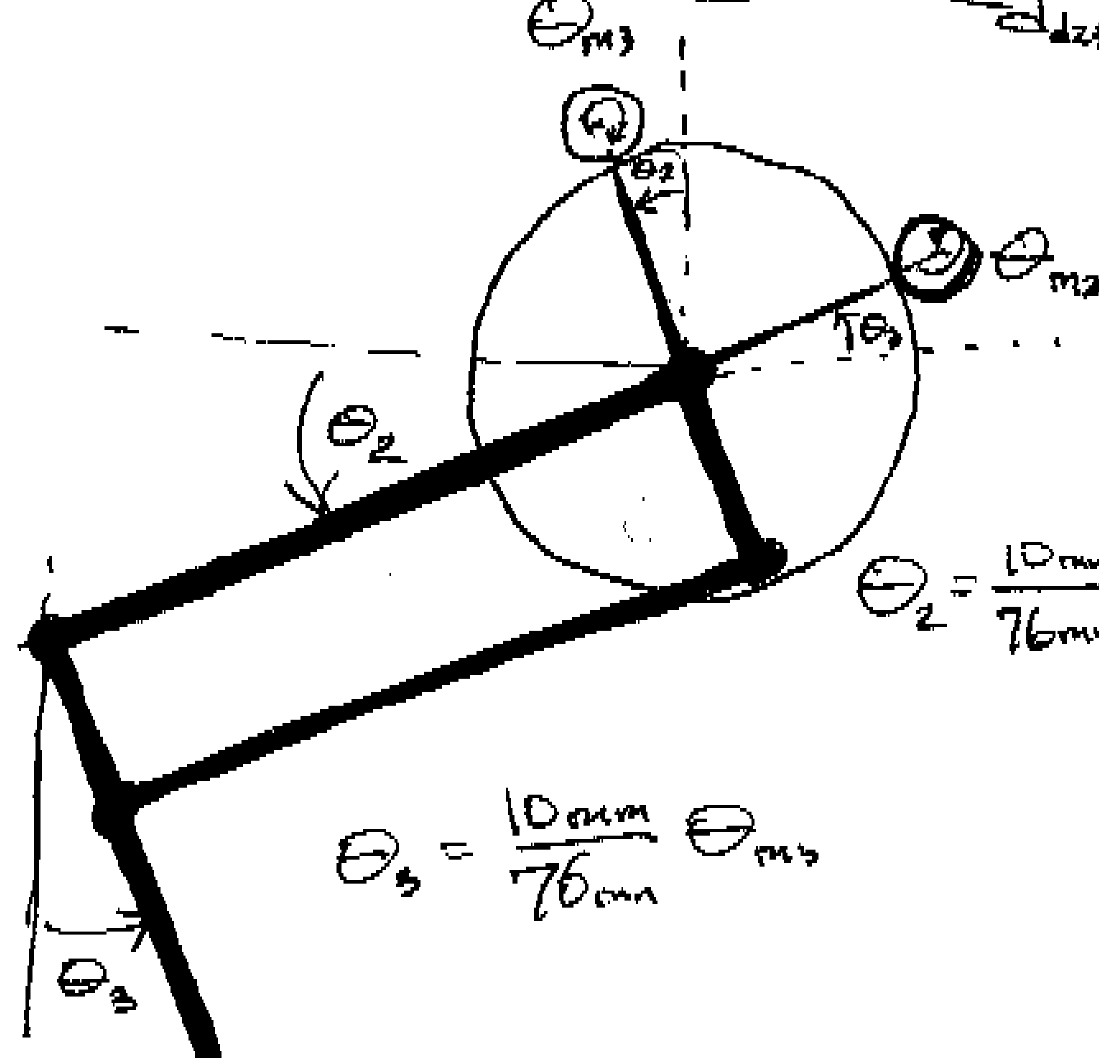
$$\theta_1 = \frac{13mm}{116mm}\,\theta_{m1}$$

Joint angle definitions

Motors 2 and 3

view from side

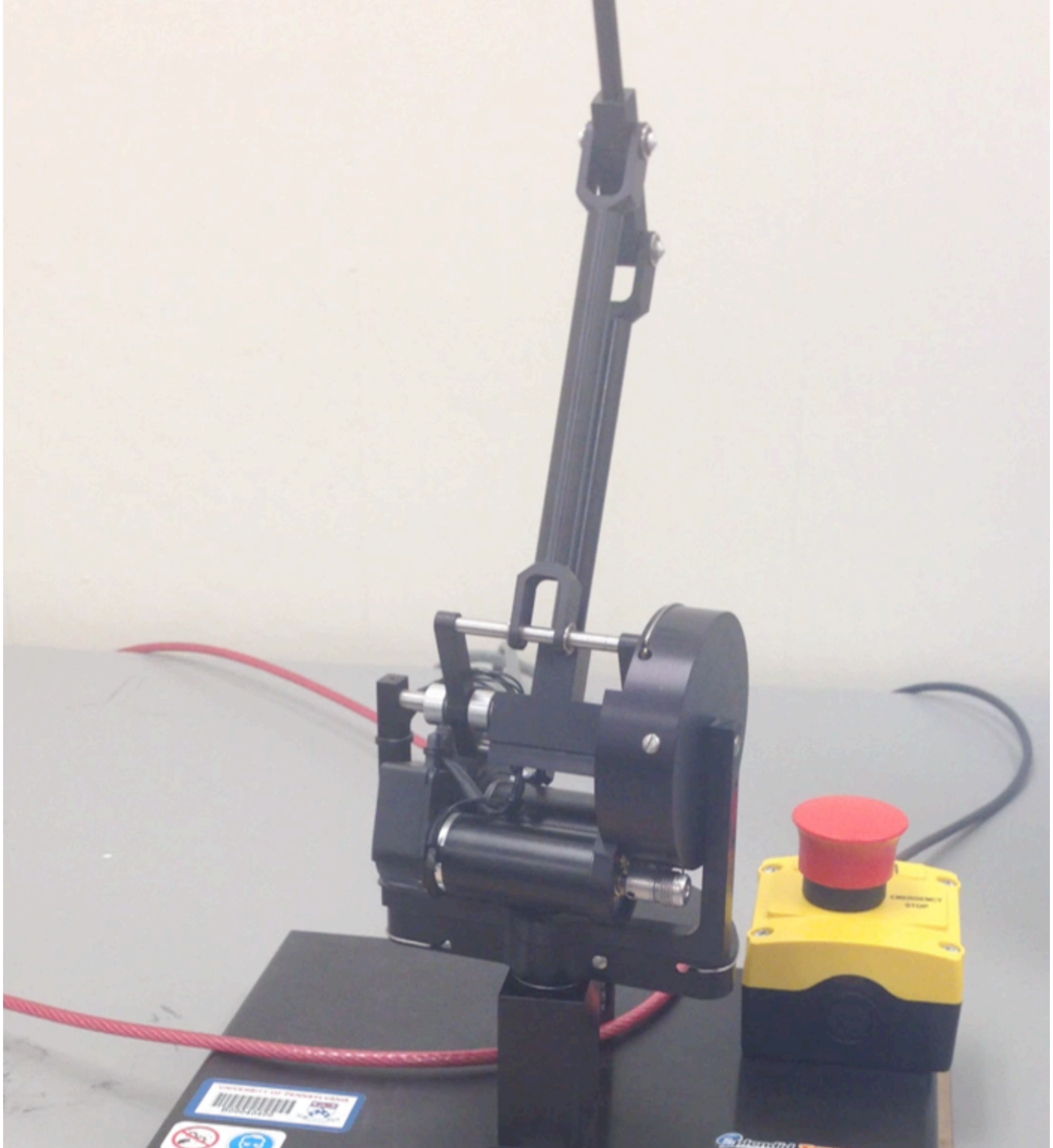




$$\theta_2 = \frac{10mm}{76mm}\,\theta_{m2}$$

$$\theta_3 = \frac{10mm}{76mm}\,\theta_{m3}$$

joints

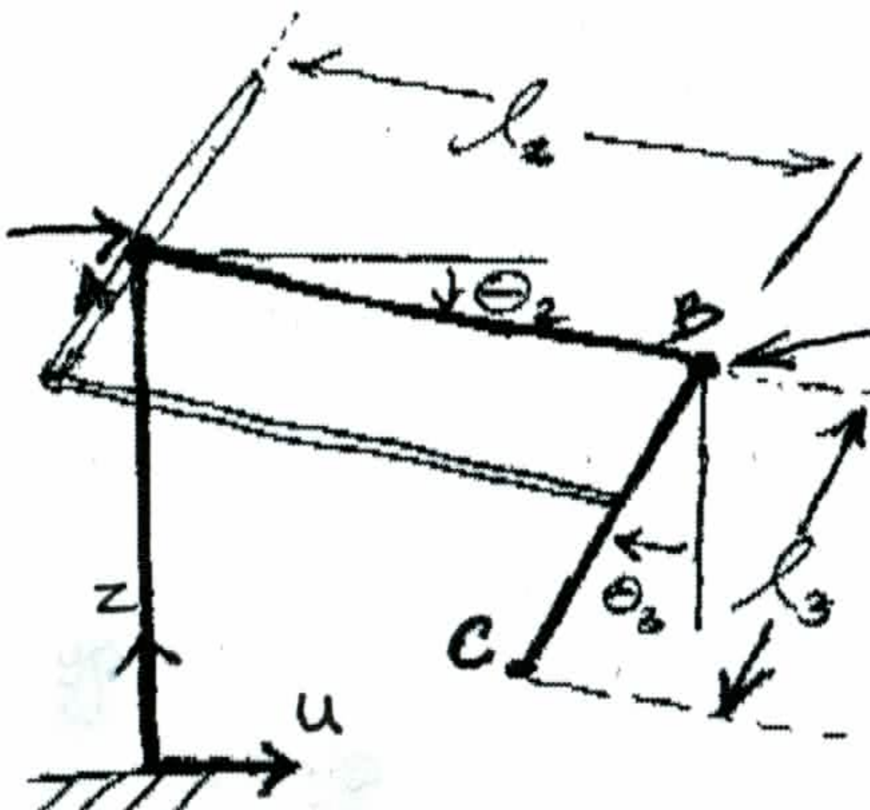


$$\theta_1 = \frac{13}{116}\,\theta_{m1} \quad \text{(base)}$$

$$\theta_2 = \frac{10}{76}\,\theta_{m2} \quad \text{(shoulder)}$$

$$\theta_3 = \frac{10}{76}\,\theta_{m3} \quad \text{(elbow)}$$

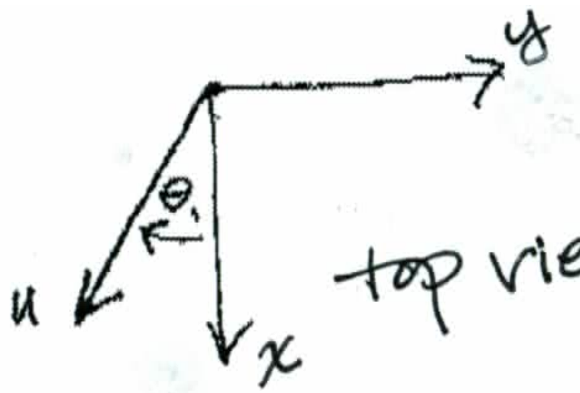
$$\begin{bmatrix} u_A \\ z_A \end{bmatrix} = \begin{bmatrix} 0 \\ l_1 \end{bmatrix}$$



$$\begin{bmatrix} u_B \\ z_B \end{bmatrix} = \begin{bmatrix} l_2 \cos\theta_2 \\ l_1 - l_2 \sin\theta_2 \end{bmatrix}$$

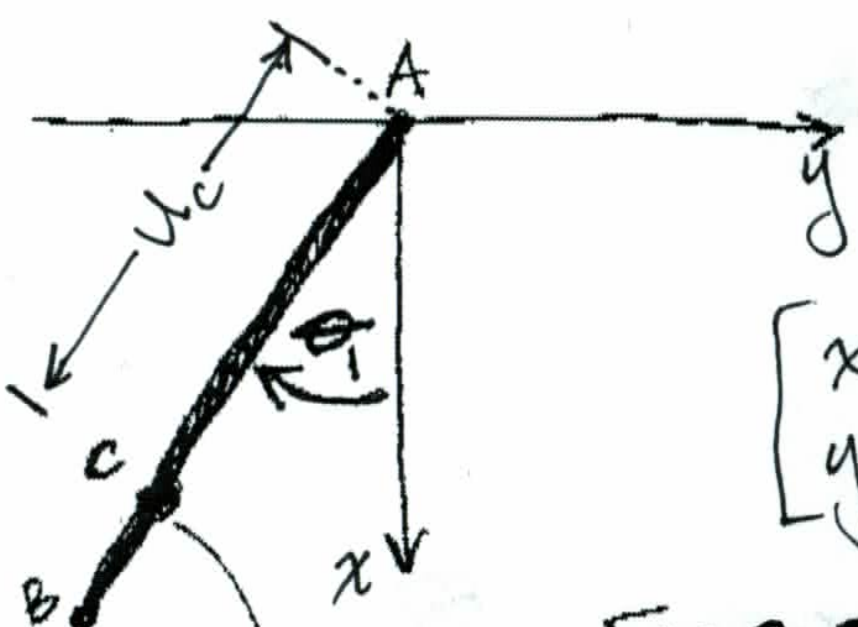
$$\begin{bmatrix} u_C \\ z_C \end{bmatrix} = \begin{bmatrix} l_2 \cos\theta_2 - l_3 \sin\theta_3 \\ l_1 - l_2 \sin\theta_2 - l_3 \cos\theta_3 \end{bmatrix}$$

side view
rotated by θ_1



top view

Now we know the coordinates of this point in the u-z plane. Viewed from above, the robot arm is in cylindrical coordinates, with θ_1 rotation from the x axis and a radius of $u_c = l_2 \cos\theta_2 - l_3 \sin\theta_3$



$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} \cos\theta_1 \cdot u_c \\ -\sin\theta_1 \cdot u_c \end{bmatrix}$$

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} \cos\theta_1 \, (l_2 \cos\theta_2 - l_3 \sin\theta_3) \\ -\sin\theta_1 \, (l_2 \cos\theta_2 - l_3 \sin\theta_3) \end{bmatrix}$$

θ_1 is base

θ_2 is shoulder

θ_3 is elbow

all relative to ground.

(b)

$$\boxed{\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\theta_1 (l_2 \cos\theta_2 - l_3 \sin\theta_3) \\ -\sin\theta_1 (l_2 \cos\theta_2 - l_3 \sin\theta_3) \\ l_1 - l_2 \sin\theta_2 - l_3 \cos\theta_3 \end{bmatrix}}$$

Note you should not be adding θ_2 and θ_3.

From measurements:

$$\boxed{l_1 \cong 168\text{mm} \qquad l_2 \cong 140\text{mm} \qquad l_3 \cong 140\text{mm}}$$

(b) shown again

(b) $$\boxed{\begin{aligned} x_{tip} &= (\cos\theta_1)(l_2\cos\theta_2 - l_3\sin\theta_3) \\ y_{tip} &= -(\sin\theta_1)(l_2\cos\theta_2 - l_3\sin\theta_3) \\ z_{tip} &= l_1 - l_2\sin\theta_2 - l_3\cos\theta_3 \end{aligned}} = \begin{bmatrix} C1\,(l_2 C2 - l_3 S \\ -S1\,(l_2 C2 - l_3 \\ l_1 - l_2 S2 - l_3 \cdot C3 \end{bmatrix}$$

) Jacobian Transpose : $\vec{\tau} = J^T \vec{F}$

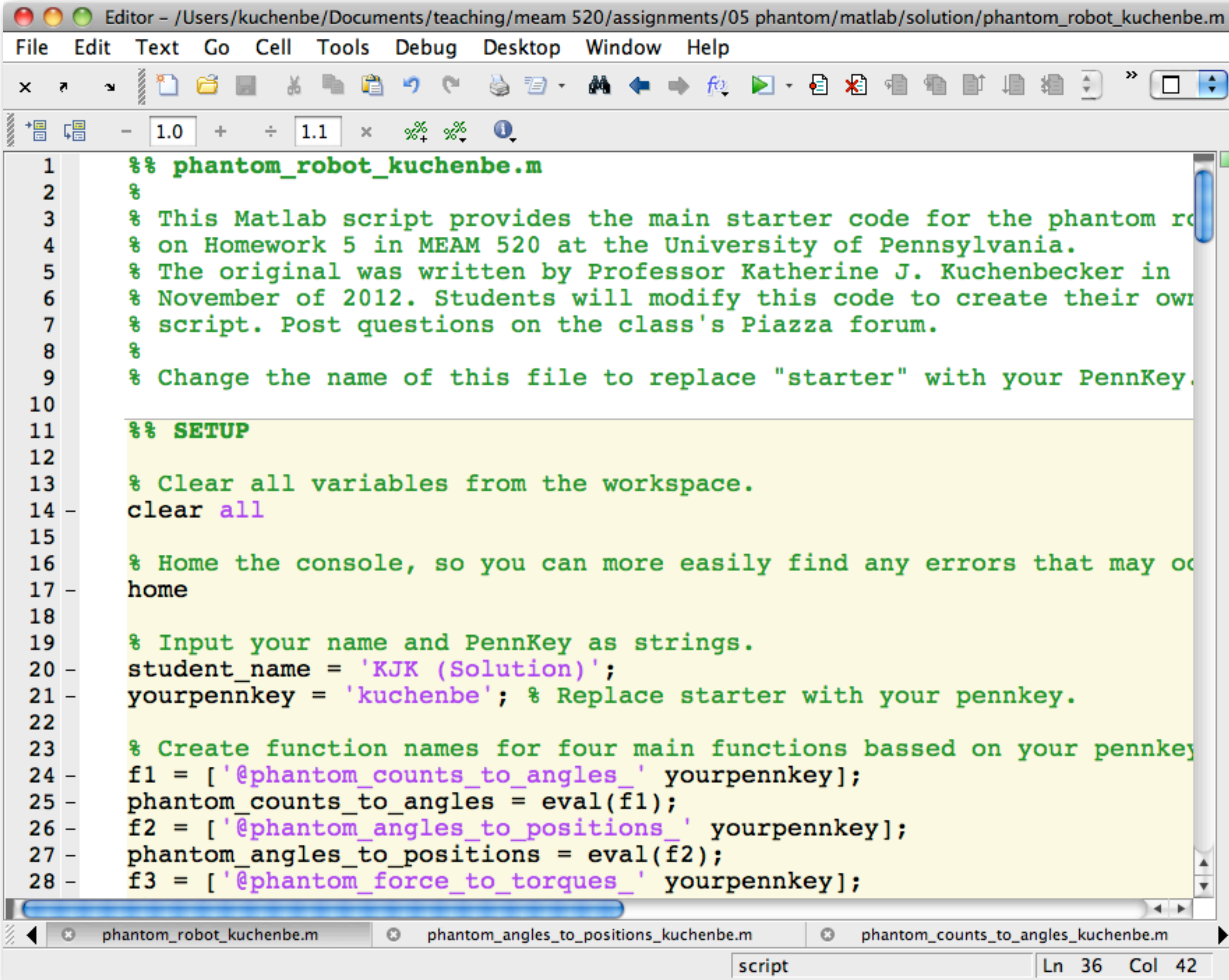
$$J_v = \begin{bmatrix} -S1\,(l_2 \cdot C2 - l_3 \cdot S3) & -l_2 \cdot C1 \cdot S2 & -l_3 \cdot C1 \cdot C3 \\ -C1\,(l_2 C2 - l_3 \cdot S3) & l_2 \cdot S1 \cdot S2 & l_3 \cdot S1 \cdot C3 \\ 0 & -l_2 \cdot C2 & l_3 \cdot S3 \end{bmatrix}$$

(c) $$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = J^T \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}$$

$$V_1 = \tau_1 \cdot \frac{13}{116} \cdot \frac{5\Omega}{0.0234\,Nm/A}$$

$$V_2 = \tau_2 \cdot \frac{10}{76} \cdot \frac{5\Omega}{0.0234\,Nm/A}$$

$$V_3 = \tau_3 \cdot \frac{10}{76} \cdot \frac{5\Omega}{0.0234\,Nm/A}$$


Editor – /Users/kuchenbe/Documents/teaching/meam 520/assignments/05 phantom/matlab/solution/phantom_robot_kuchenbe.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1.0
1.1
%% phantom_robot_kuchenbe.m
%
% This Matlab script provides the main starter code for the phantom r
% on Homework 5 in MEAM 520 at the University of Pennsylvania.
% The original was written by Professor Katherine J. Kuchenbecker in
% November of 2012. Students will modify this code to create their ow
% script. Post questions on the class's Piazza forum.
%
% Change the name of this file to replace "starter" with your PennKey
%% SETUP
% Clear all variables from the workspace.
clear all
% Home the console, so you can more easily find any errors that may o
home
% Input your name and PennKey as strings.
student_name = 'KJK (Solution)';
yourpennkey = 'kuchenbe'; % Replace starter with your pennkey.
% Create function names for four main functions bassed on your pennke
f1 = ['@phantom_counts_to_angles_' yourpennkey];
phantom_counts_to_angles = eval(f1);
f2 = ['@phantom_angles_to_positions_' yourpennkey];
phantom_angles_to_positions = eval(f2);
f3 = ['@phantom_force_to_torques_' yourpennkey];
phantom_robot_kuchenbe.m
phantom_angles_to_positions_kuchenbe.m
phantom_counts_to_angles_kuchenbe.m
script
Ln 36 Col 42

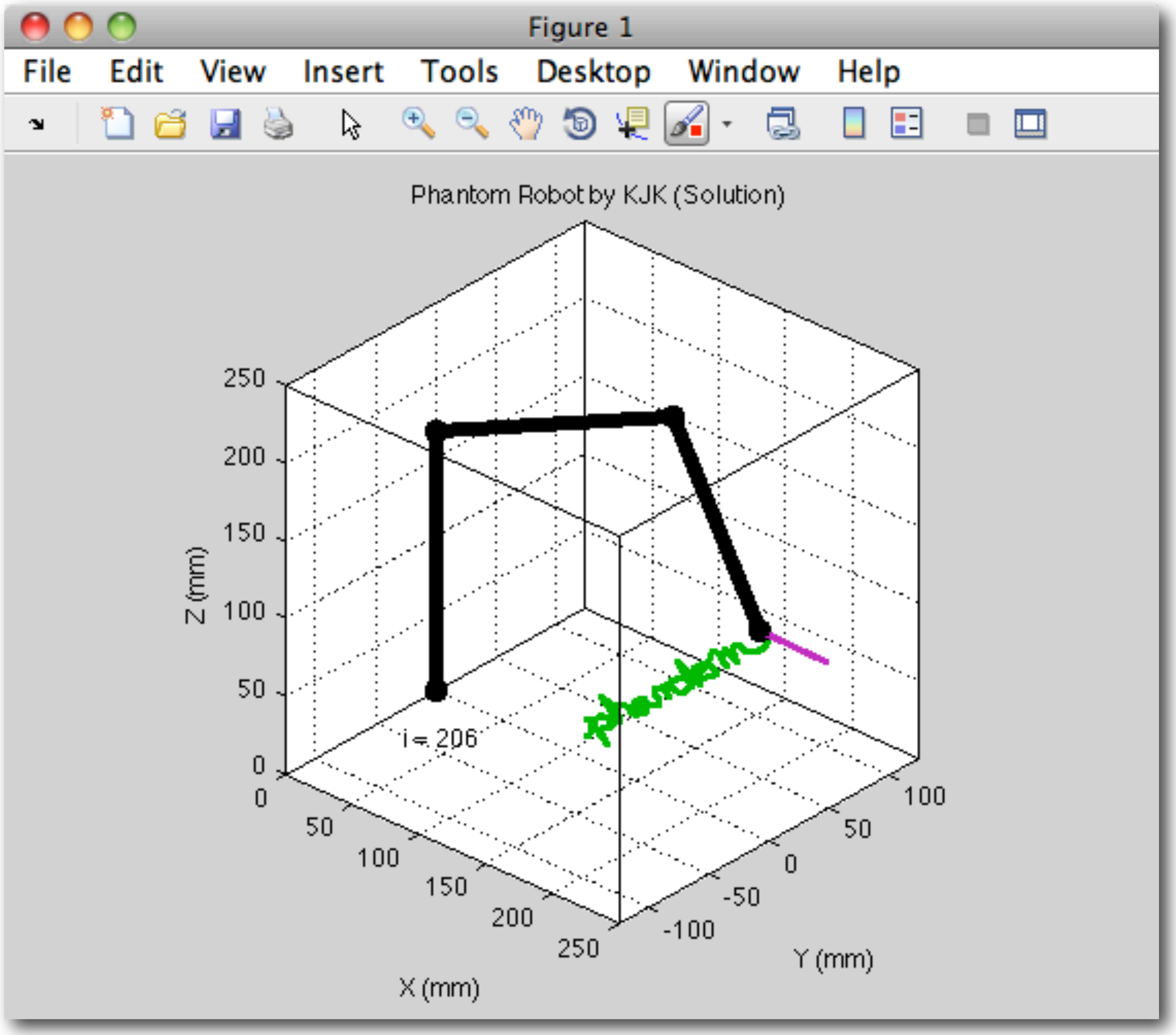


Figure 1
File Edit View Insert Tools Desktop Window Help
Phantom Robot by KJK (Solution)
Z (mm)
X (mm)
Y (mm)
i = 206
250
200
150
100
50
0
0
50
100
150
200
250
100
50
0
-50
-100

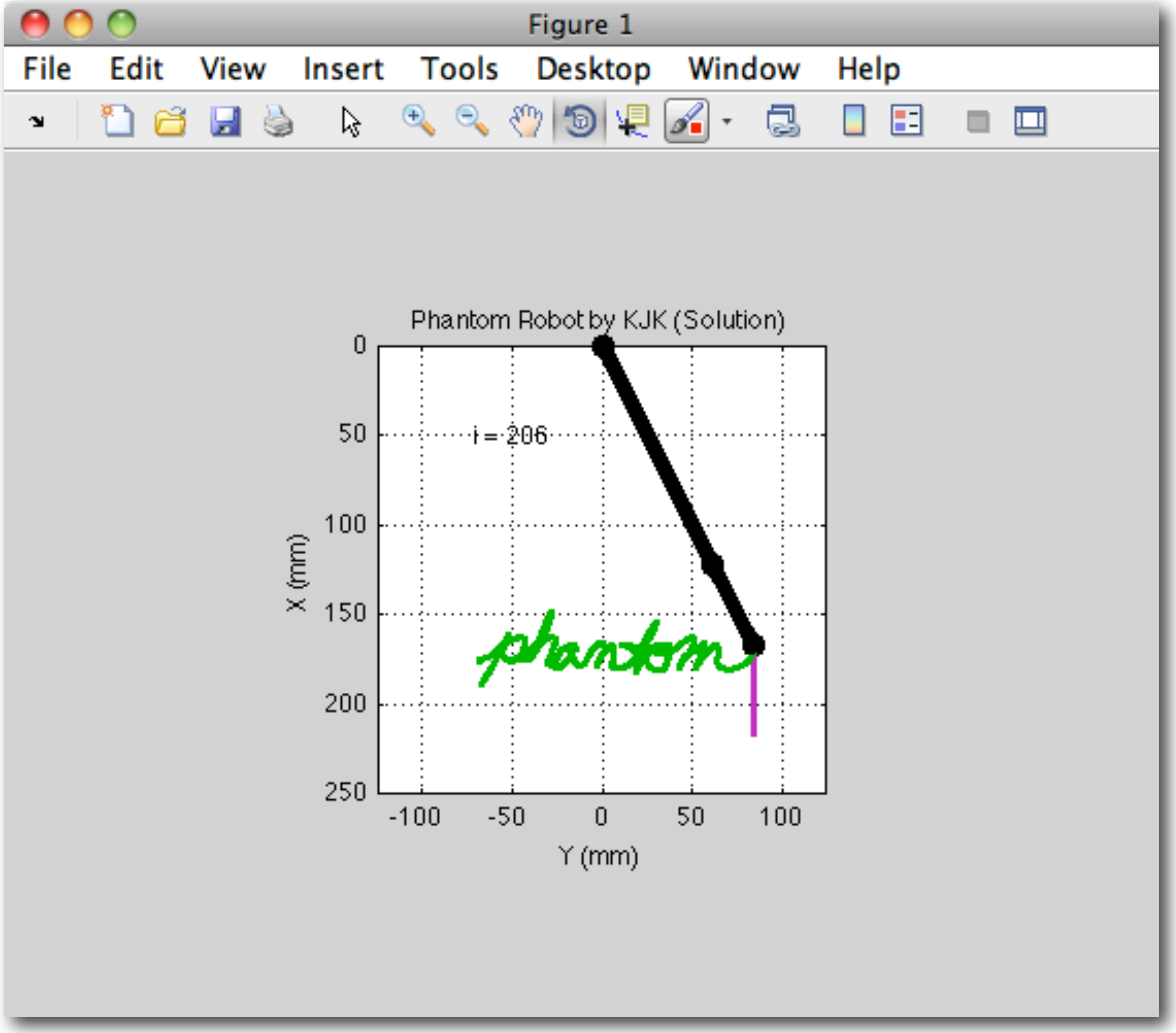


Figure 1
File Edit View Insert Tools Desktop Window Help
Phantom Robot by KJK (Solution)
i = 206
X (mm)
Y (mm)
0
50
100
150
200
250
-100
-50
0
50
100
phantom

Questions ?

Project 2

Team Formation

You will work in a team of 3
(33 teams of 3, only one team of 4)

You may not keep the same team as for project 1.
I strongly encourage you to work with all new people.

If you insist, you can stay with one other person, but beware the negative effects this may have on the dynamics of your subsequent team.

I encourage you to mix undergrad and grad, MEAM and non-MEAM, but this is not required.

Please pick your team ASAP.
We will randomly assign people who do not select a team.

MEAM.Design : MEAM520-12C-P02-Teams

http://medesign.seas.upenn.edu/index.php/Courses/MEAM520-12C-P02-Teams

MEAM.Design : MEAM520-12C-P02-Teams

View Logout
Edit Upload

MEAM.Design - MEAM 520 - **PHANToM Haptics: Team Selection**

Project 2 is PHANToM Haptics. The first part of this project is selecting your team. You must pick your team for PHANToM Haptics by midnight on **Tuesday, November 27**.

Below are the team selection guidelines:

- Each team will include exactly three students.
- You may not keep the same team as for project 1. We strongly encourage you to work with all new people.
- If you insist, you can stay with one other person, but beware the negative effects this may have on the dynamics of your subsequent team.
- We encourage you to mix undergrad and grad, MEAM and non-MEAM, but this is not required.

Looking for teammates? Try using the Project Teams & Study Groups tool on Piazza.

When you have chosen your team, please send a list of all three names to Denise Wong (MEAM 520 TA).

We will be happy to assign you to a team if you can't find two partners. Just send an email to Denise to explain the situation. Include your name, email address, and degree program to assist her in matching you with other students. Teams will be assigned on a rolling basis to give you as much time as possible to work together on the project.

If Denise does not hear from you by midnight on Tuesday, she will assign you to a team on Wednesday. The list below shows all confirmed teams.

Team numbers for project 2 will start at 51 to avoid confusion with team numbers from project 1.

Team - Members
51 - Michael Gosselin, Jialue Huang, and Qiong Wang
52 - Pablo Castillo, Wei-Ting Lo, and Chao Qu
53 - Sarah Badin, Dalton Banks, and Kunal Mahajan
54 - Shaojun Zhu, Hao Min, and Tianyi Zhang
55 - Noor Bosch, Nicholas Mcgill, and William Price
56 - Tianyu Dong, Yanwei Du, and Quangeng Xu
57 - Chaoyi Huang, Bo Yuan, and Rui Zhang
58 - Manisha Golas, Alex Jose, and Anand Mahusoohanan
59 - Sawyer Brooks, Dean Wilhelmi, and James Yang
60 - Bofei Wang, Di Wu, and Shiming Zhao
61 - Christine Kappeyne, Niko Vladimirov, and Brett Wittmerhaus
62 - Jon Balloch, Yunkai Cui, and Jason Gui
63 - Anne Mroz, Alex Sher, and Fiona Strain
64 - Noam Eisen, Eza Koch, and Paige Willoughby

COURSES
MEAM 101
MEAM 201
MEAM 410/510
MEAM 520
IPD 501
SAAST

GUIDES
Materials
Laser Cutting
3D Printing
Machining
ProtoTRAK
PUMA 260
PHANToM
BeagleBoard
MAEVARM
Phidget
Tap Chart

SOFTWARE
SolidWorks
Matlab
NX
Nastran
Fluent, Gambit
SolidCAM
Eagle

OTHER
Vendor List
Design Links

- You may not keep the same team as for project 1. We strongly encourage you to work with all new people.
- If you insist, you can stay with one other person, but beware the negative effects this may have on the dynamics of your subsequent team.
- We encourage you to mix undergrad and grad, MEAM and non-MEAM, but this is not required.

Looking for teammates? Try using the Project Teams & Study Groups tool on Piazza.

When you have chosen your team, please send a list of all three names to Denise Wong (MEAM 520 TA).

We will be happy to assign you to a team if you can't find two partners. Just send an email to Denise to explain the situation. Include your name, email address, and degree program to assist her in matching you with other students. Teams will be assigned on a rolling basis to give you as much time as possible to work together on the project.

If Denise does not hear from you by midnight on Tuesday, she will assign you to a team on Wednesday. The list below shows all confirmed teams.

Team numbers for project 2 will start at 51 to avoid confusion with team numbers from project 1.

Team - Members
51 - Michael Gosselin, Jialue Huang, and Qiong Wang
52 - Pablo Castillo, Wei-Ting Lo, and Chao Qu
53 - Sarah Badin, Dalton Banks, and Kunal Mahajan
54 - Shaojun Zhu, Hao Min, and Tianyi Zhang
55 - Noor Bosch, Nicholas Mcgill, and William Price
56 - Tianyu Dong, Yanwei Du, and Quangeng Xu
57 - Chaoyi Huang, Bo Yuan, and Rui Zhang
58 - Manisha Golas, Alex Jose, and Anand Mahusoohanan
59 - Sawyer Brooks, Dean Wilhelmi, and James Yang
60 - Bofei Wang, Di Wu, and Shiming Zhao
61 - Christine Kappeyne, Niko Vladimirov, and Brett Wittmerhaus
62 - Jon Balloch, Yunkai Cui, and Jason Gui
63 - Anne Mroz, Alex Sher, and Fiona Strain
64 - Noam Eisen, Eza Koch, and Paige Willoughby
65 - Annett Bordoley, Dieter Neckermann, and Rafi Holzer
66 - Jennifer Hui, Hardik Gupta, and Robert Parajon
67 - Erica Aduh, Daniel Friedman, and Cristina Sorice 68 - Leslie Callaghan, Vivian Chu, and Stella Latscha
69 - Sukreeti Sehrawat, Samarch Manoj Brahmbhatt, and Alexandre Miranda Anon
70 -

MEAM.Design : Phantom

http://medesign.seas.upenn.edu/index.php/Guides/Phantom

MEAM.Design : Phantom

View Logout
Edit Upload

GENERAL
- Hall of Fame
- Laboratories
- Contact Info

COURSES
- MEAM 101
- MEAM 201
- MEAM 410/510
- MEAM 520
- IPD 501
- SAAST

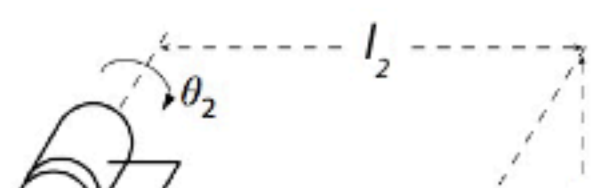
GUIDES
- Materials
- Laser Cutting
- 3D Printing
- Machining
- ProtoTRAK
- PUMA 260
- PHANToM
- BeagleBoard
- MAEVARM
- Phidget
- Tap Chart

SOFTWARE
- SolidWorks
- Matlab

SensAble PHANToM Premium 1.0

Overview

As shown in the photo below, the PHANToM is an impedance-type haptic interface with three actuated rotational joints. Designed to be lightweight, stiff, smooth, and easily backdrivable, this type of haptic interface enables a human user to interact with a virtual environment or control the movement of a remote robot through the movement of their fingertip while simultaneously feeling force feedback. A thimble is attached to the tip of the robot via a passive non-encoded three-axis gimbal to allow the user to move the robot around while freely rotating their fingertip.



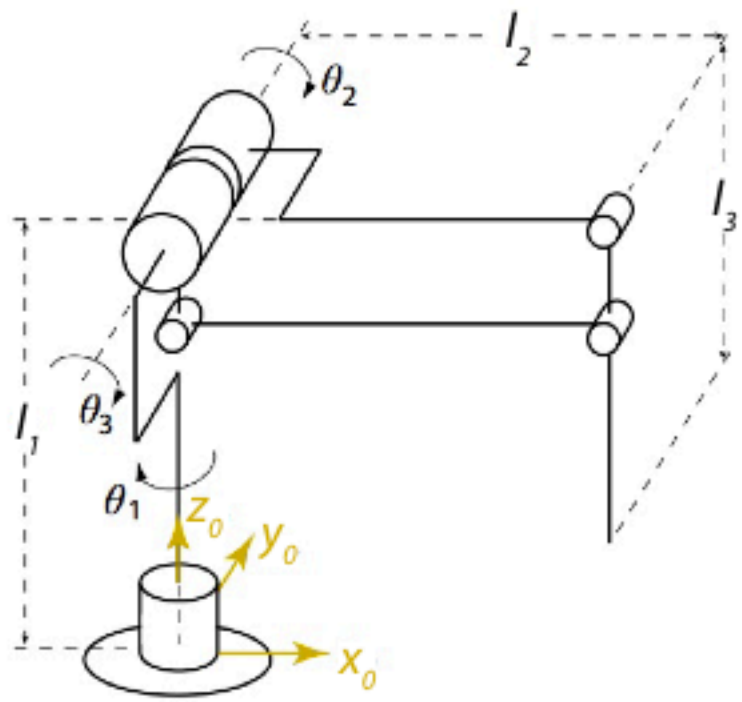
BeagleBoard
MAEVARM
Phidget
Tap Chart

SOFTWARE
SolidWorks
Matlab
NX
Nastran
Fluent, Gambit
SolidCAM
Eagle

OTHER
Vendor List
Design Links

Editing Tips
edit SideBar

SEARCH
Go



As shown in the diagram above, the Phantom haptic device looks similar to the standard RRR articulated manipulator base, but it uses a unique four-bar mechanism to co-locate the shoulder and elbow joints while also keeping the upper arm and forearm in the plane that intersects the axis of the waist joint.

Mechanical Design

Each of the three arm joints (1, 2, and 3) is driven by a Maxon 118743 DC brushed motor with a shaft-mounted HEDM-5500-B02 optical encoder.

Drum and capstan cable drives are used to connect the motors to the respective joints.

The schematic shows Phantom's zero configuration (the configuration where we zero the encoders). Here, the upper arm is horizontal, the lower arm is vertical, and the tip is located above the x-axis. The encoder counts increase when each joint is rotated in the direction indicated on the schematic; these are the positive directions for the joint angles as well.

Matlab Interface

phantomStart(*hardware*); Call this at the start of your code to initialize the PHANToM driver. Pass in ***false*** for ***hardware*** when you are developing your code - the software will simulate the presence of a user by playing a pre-recorded movement trajectory. After you are confident your code will

Go

Mechanical Design

Each of the three arm joints (1, 2, and 3) is driven by a Maxon 118743 DC brushed motor with a shaft-mounted HEDM-5500-B02 optical encoder.

Drum and capstan cable drives are used to connect the motors to the respective joints.

The schematic shows Phantom's zero configuration (the configuration where we zero the encoders). Here, the upper arm is horizontal, the lower arm is vertical, and the tip is located above the x-axis. The encoder counts increase when each joint is rotated in the direction indicated on the schematic; these are the positive directions for the joint angles as well.

Matlab Interface

phantomStart(*hardware*); Call this at the start of your code to initialize the PHANToM driver. Pass in ***false*** for ***hardware*** when you are developing your code - the software will simulate the presence of a user by playing a pre-recorded movement trajectory. After you are confident your code will have the effect you desire and you're working on the computer in B2 Towne, pass in ***true*** for ***hardware*** to run your code on the actual PHANToM. Read about the emergency stop button as well!

Encoders = phantomEncoders(); Return the instantaneous encoder values, in counts, with respect to the zero position. Encoders is a 3x1 array of the encoder values. If ***hardware*** is set to ***false***, this function will return the next set of encoder values from a pre-recorded trajectory.

Angles = phantomJointAngles(); Return the instantaneous joint angles, in radians, with respect to the zero position. Angles is a 3x1 array of the joint angles. If ***hardware*** is set to ***false***, this function will return the next set of joint angles from a pre-recorded trajectory.

phantomJointTorques(tau1, tau2, tau3); Command all three joint torques in newton-meters. A positive torque moves the corresponding joint in the positive direction. Be sure to specify **all three** values each time you call this function. A warning will be triggered if you request more torque than can be delivered via the maximum current; you can turn off this warning by calling **warning('off','PHANToM:JointTorque')** on the command line. This function has no effect if ***hardware*** is set to ***false***.

phantomCurrents(i1, i2, i3); where **(-1.0 ≤ ix ≤ 1.0)** - Command all three motor currents in amps. A positive current moves the corresponding joint in the positive direction. Be sure to specify **all three** values each time you call this function. This function has no effect if ***hardware*** is set to ***false***.

phantomStop(); End your code with this function. It closes the PHANToM driver and releases memory back to the Linux Kernel.

phantomZero(); Zero the encoders. If the PHANToM is not calibrated, run "phantomZero" in the **command window**, and follow the on-screen instructions. This command rarely needs to be run, so it should not be in your actual code. This function does not need to be run when ***hardware*** is set to ***false***.

Emergency Stop

The amplifiers in the PHANToM are governed by an emergency stop button. You must pull up the emergency stop to enable motor control, and you can see the state of the switch by looking at the center-top LED on the front of the amplifier box located under the table. You should **always** run your code with the emergency stop down (no forces) before running it with the emergency stop up (with forces).

MEAM.Design : MEAM520-12C-P02-Rendering

http://medesign.seas.upenn.edu/index.php/Courses/MEAM520-12C-P02-Rendering

MEAM.Design : MEAM520-12C-P02-Rendering

View Logout
Edit Upload

GENERAL
Hall of Fame
Laboratories
Contact Info

COURSES
MEAM 101
MEAM 201
MEAM 410/510
MEAM 520
IPD 501
SAAST

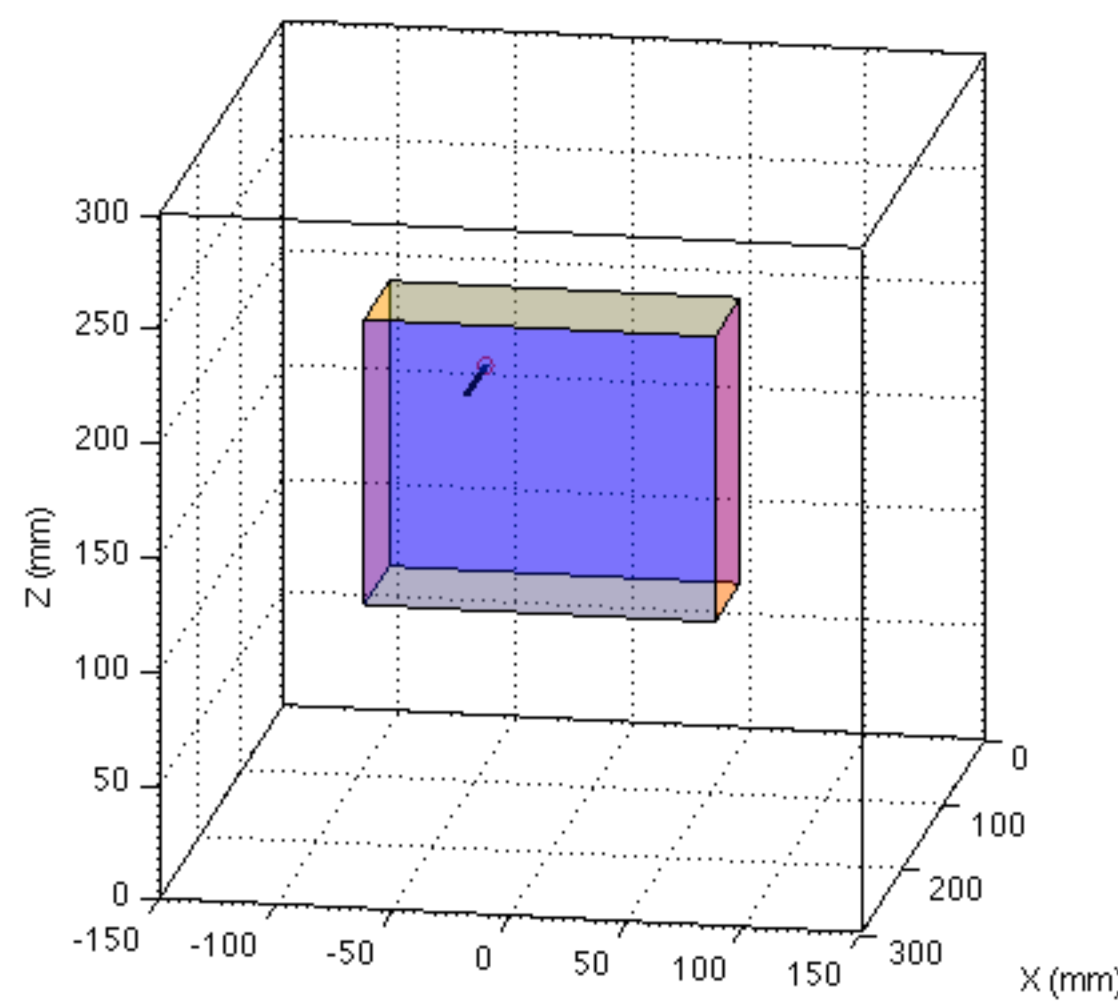
GUIDES
Materials
Laser Cutting
3D Printing
Machining
ProtoTRAK
PUMA 260
PHANToM
BeagleBoard
MAEVARM
Phidget
Tap Chart

SOFTWARE
SolidWorks
Matlab

MEAM.Design - MEAM 520 - **PHANToM Haptics: Rendering**

Now that you have your team, it's time to get to work on project 2. This assignment is due by **5:00 p.m. on Tuesday, December 4**.

Start by downloading the starter code (v1). We are providing you with p-coded versions of all the functions described in the Phantom Guide. For example, calling `phantomStart(false);` starts the simulated phantom so you can work on your code on any computer. Instead of getting encoder readings from the real PHANToM, the system simulates the presence of a human user by reading a pre-recorded trajectory from the included `encsHistory.mat` file.

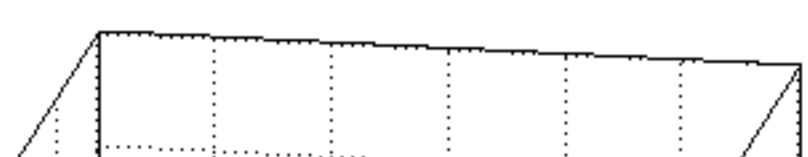


Demo: Haptic Box

Run `haptic_box_demo.m` and look at how it is written. This demonstration creates a virtual haptic box for the user to feel, as seen in the top illustration at right. The user is trapped inside the virtual box and feels a virtual spring force each time they contact a wall. The position of the PHANToM tip is shown as a red circle, the box is shown in transparent colors, and a scaled version of the force vector is shown as a thick black line.

The system simulates the presence of a human user by default because you probably don't have a PHANToM connected to your computer. Look at how the forces Fx, Fy, and Fz are calculated from the positions hx, hy, and hz. This is the type of mapping you will need to create in this assignment.

Once you understand how the haptic box demo works, your team's task on this project is to complete the following two haptic rendering scenes created for the PHANToM.



Task 1: Haptic Ball

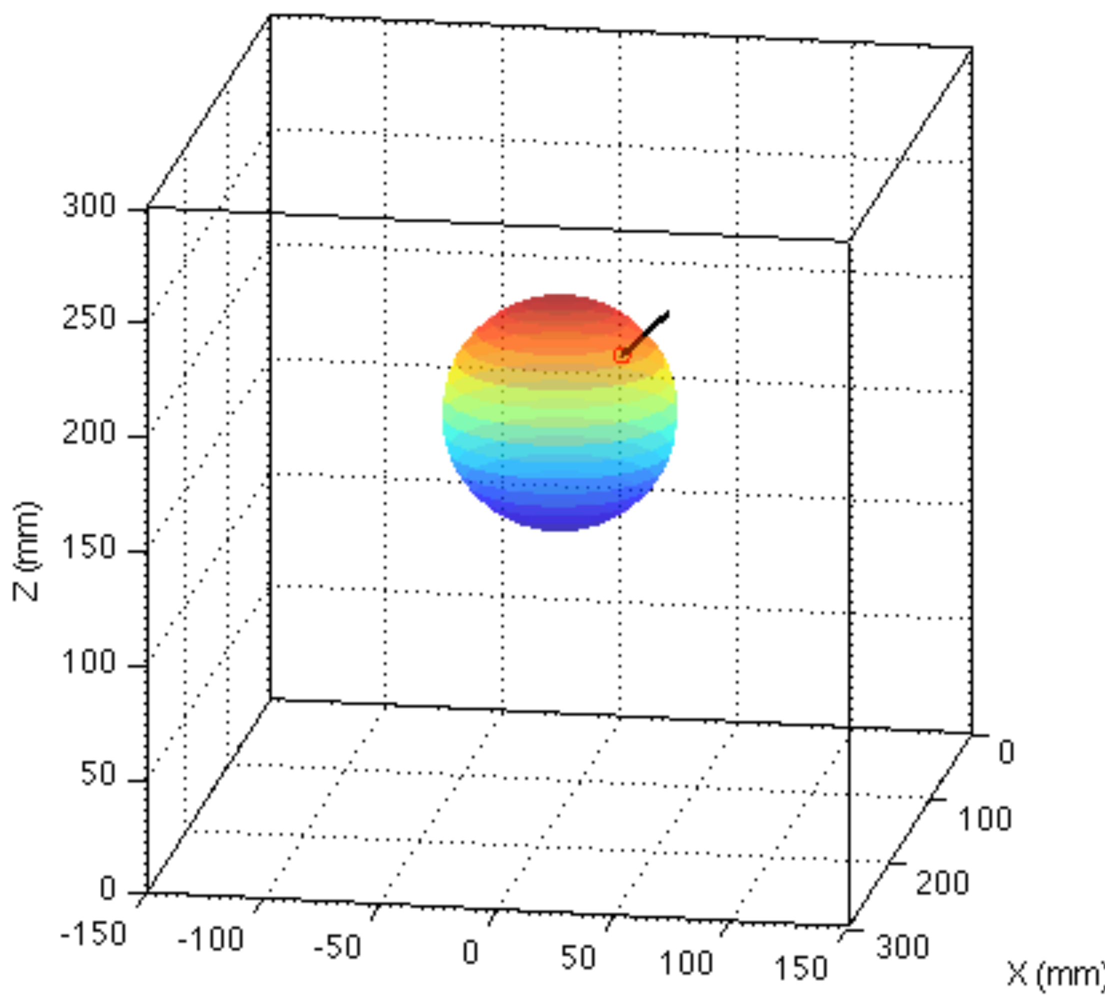
Complete the **haptic ball** scene that has been started for you in `haptic_ball_team50.m`. Change the filename to match

SOFTWARE
SolidWorks
Matlab
NX
Nastran
Fluent, Gambit
SolidCAM
Eagle

OTHER
Vendor List
Design Links

Editing Tips
edit SideBar

SEARCH
Go



Task 1: Haptic Ball

Complete the **haptic ball** scene that has been started for you in `haptic_ball_team50.m`. Change the filename to match your team number, and list your team number and the names of your team members at the top of the file.

The graphics for this scene are shown in the middle image at left. Again, the position of the PHANToM tip is shown as a red circle, the ball is shown as a transparent rainbow sphere, and a scaled version of the force vector is shown as a thick black line.

Modify the code between the two lines of stars to push the user out of the ball whenever they come inside, using a virtual spring mapping. The force should push them straight out of the ball, and the magnitude of the force should be proportional to their penetration depth. The stiffness of your spring should be k (already defined for you in the code). Please comment your code.

Use the graphics to debug your calculations in simulation (with hardware set to false) until all of your team members are convinced the calculations are correct. Only then are you allowed to try your code on the real PHANToM. You are welcome to show your code's output to a member of the teaching team to get their opinion of its correctness if you are not sure.

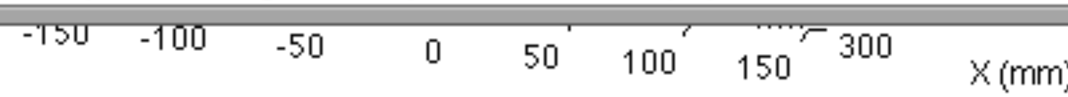
Task 2: Haptic Damping

Complete the **haptic damping** scene that has been started for you in `haptic_damping_team50.m`. Change the filename to match your team number, and note your team number and the names of your team members at the top of the file.

The graphics for this scene are shown in the bottom image at right. Again, the position of the PHANToM tip is shown as a red circle, and a scaled version of the force vector is shown as a thick black line.

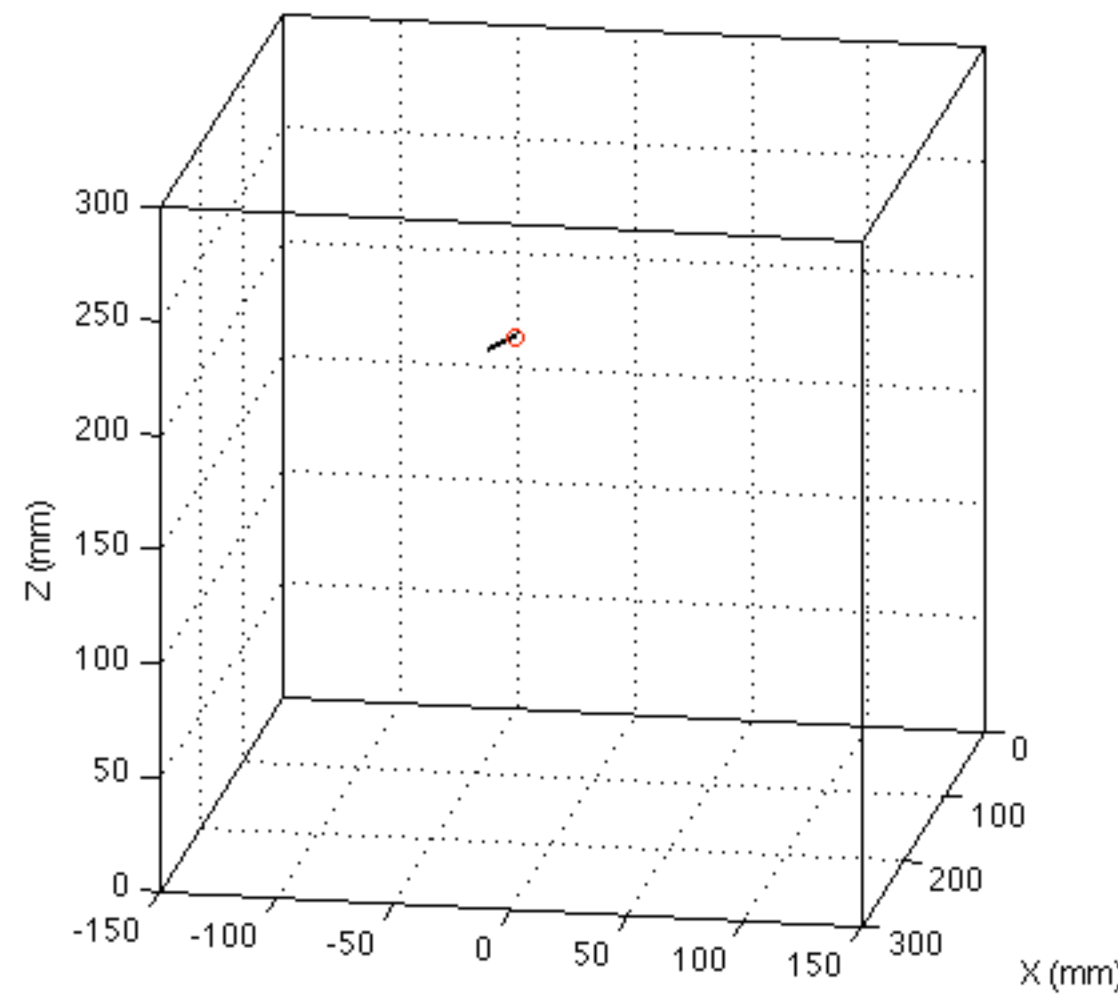
Modify the code between the two lines of stars to output a viscous

Go



code on the real PHANToM. You are welcome to show your code's output to a member of the teaching team to get their opinion of its correctness if you are not sure.

Task 2: Haptic Damping



Complete the **haptic damping** scene that has been started for you in `haptic_damping_team50.m`. Change the filename to match your team number, and note your team number and the names of your team members at the top of the file.

The graphics for this scene are shown in the bottom image at right. Again, the position of the PHANToM tip is shown as a red circle, and a scaled version of the force vector is shown as a thick black line.

Modify the code between the two lines of stars to output a viscous damping force that always acts to slow the user down. This force should act in the opposite direction to the user's velocity vector, and its magnitude should scale with the magnitude of their velocity vector. The viscosity of your damper should be b (already defined for you in the code). Please comment your code.

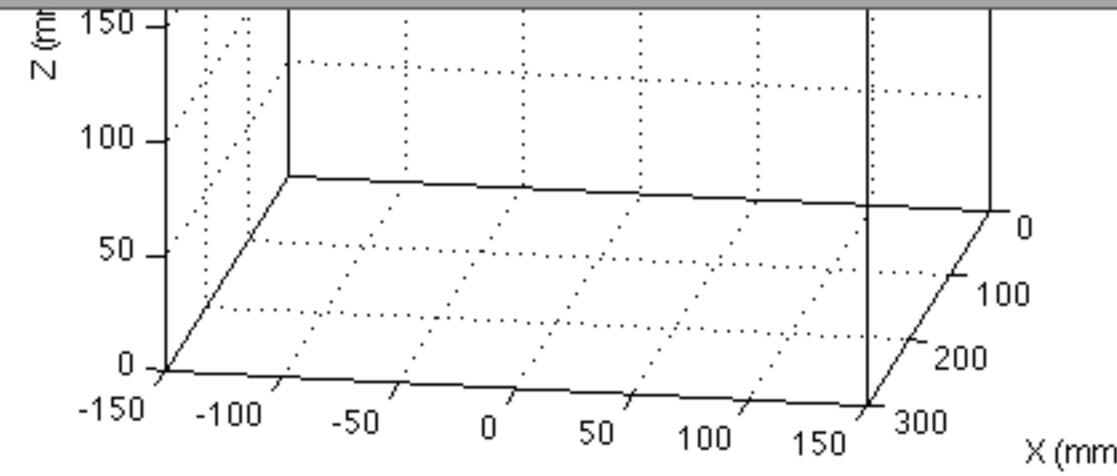
When correctly implemented, viscous damping should feel like you are moving your hand through molasses or honey. There should not be any jittering in the force. This means you will need to low-pass filter your velocity vector to reduce quantization noise.

Use the graphics to debug your calculations in simulation (with hardware set to false) until all of your team members are convinced the calculations are correct. Only then are you allowed to try your code on the real PHANToM. You are welcome to show your code's output to a member of the teaching team to get their opinion of its correctness if you are not sure.

PHANToM Rules

- You will need to reserve the PHANToM computer in order to test your code. Here is the link to the reservation system.
- Copy only your `haptic_box_demo.m`, `haptic_ball_teamXX.m`, and `haptic_damping_teamXX.m` scripts, not the rest of the phantom simulator files. Use your team's directory in the working directory on the computer's desktop.
- Check the calibration of the PHANToM by calling phantomJointAngles from the command line. If the encoders are not correctly zeroed, run phantomZero from the command line.

damping force that always acts to slow the user down. This force should act in the opposite direction to the user's velocity vector, and its magnitude should scale with the magnitude of their velocity vector. The viscosity of your damper should be b (already defined for you in the code). Please comment your code.



When correctly implemented, viscous damping should feel like you are moving your hand through molasses or honey. There should not be any jittering in the force. This means you will need to low-pass filter your velocity vector to reduce quantization noise.

Use the graphics to debug your calculations in simulation (with hardware set to false) until all of your team members are convinced the calculations are correct. Only then are you allowed to try your code on the real PHANToM. You are welcome to show your code's output to a member of the teaching team to get their opinion of its correctness if you are not sure.

PHANToM Rules

- You will need to reserve the PHANToM computer in order to test your code. Here is the link to the reservation system.
- Copy only your `haptic_box_demo.m`, `haptic_ball_teamXX.m`, and `haptic_damping_teamXX.m` scripts, not the rest of the phantom simulator files. Use your team's directory in the working directory on the computer's desktop.
- Check the calibration of the PHANToM by calling phantomJointAngles from the command line. If the encoders are not correctly zeroed, run phantomZero from the command line.
- Start by running `haptic_box_demo.m` to get a feel for how it works.
- **You must have the PHANToM emergency stop down (no forces) the first time you run any code, even the demo code.**
- **You must firmly grip the PHANToM gimbal every time you run any code.**
- **Someone must have their hand on the emergency stop whenever the PHANToM is running.**

Submission

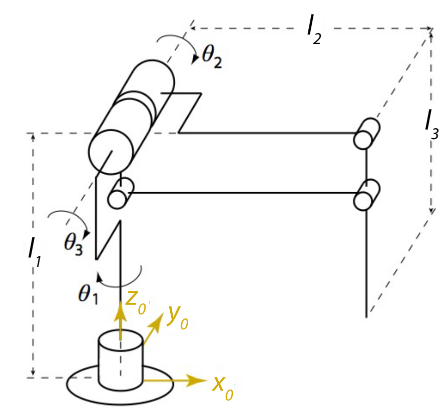
1. Start an email to meam520@seas.upenn.edu
2. Make the subject *PHANToM Haptics: Team XX*, replacing XX with your team number.
3. Attach both of your correctly named MATLAB files to the email. It should be `haptic_ball_teamXX.m` and `haptic_damping_teamXX.m`, where XX is your team number, plus any additional files you may have created, also named according to this convention.
4. Send the email.

Please come talk to the teaching team or post questions on Piazza if you get stuck on any part of this assignment.

PHANToM Haptics Starter Code (v1)

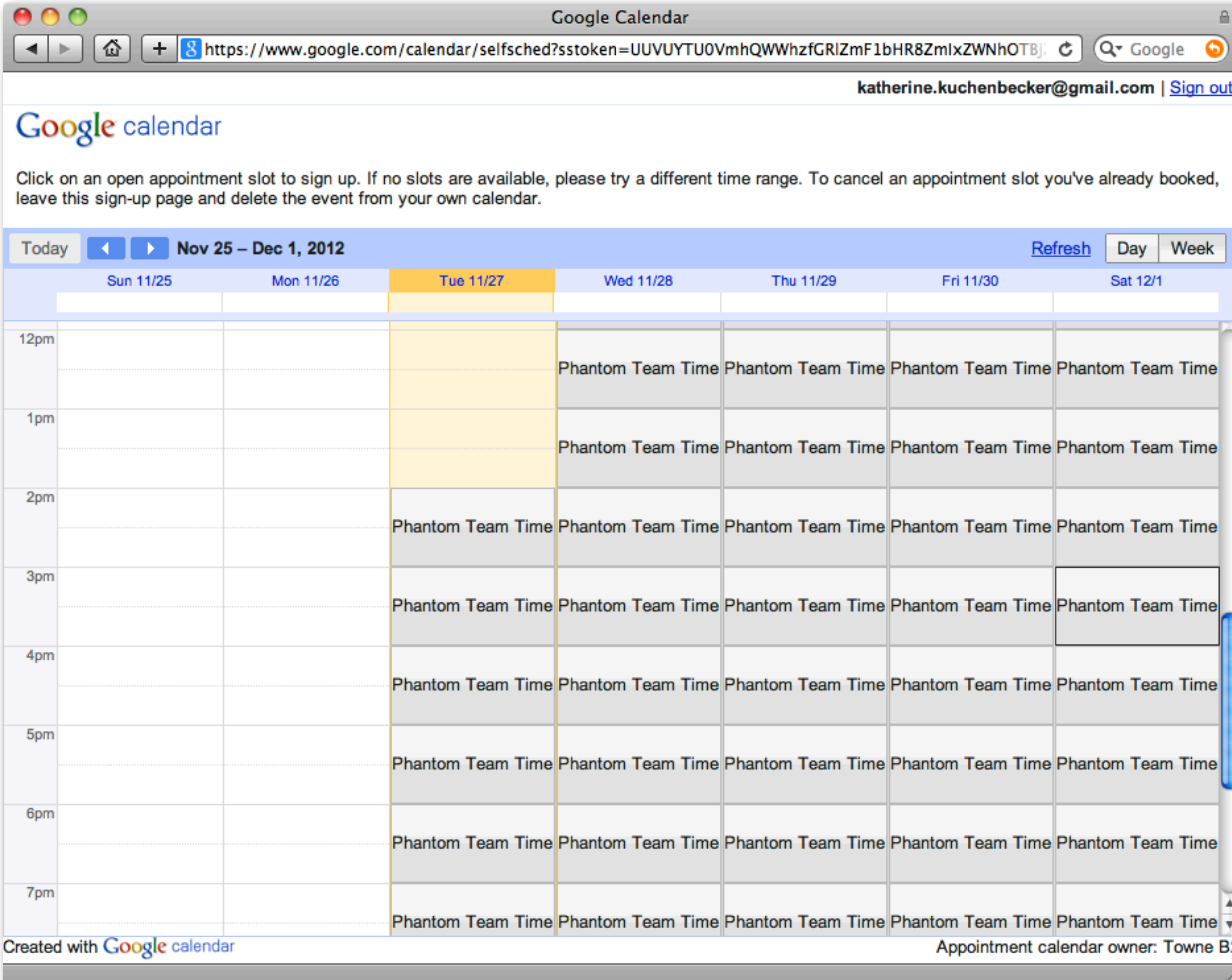
PHANToM Rules

1. You must reserve the PHANToM computer in order to test your code. You may reserve only one one-hour slot at a time. Respect the next team by finishing on time.
2. Copy only your haptic_box_demo.m, haptic_ball_teamXX.m, and haptic_damping_teamXX.m scripts, not the rest of the phantom simulator files. Use your team's folder in the working directory on the computer's desktop.
3. Check the calibration of the PHANToM by calling phantomJointAngles from the MATLAB command line. If the joint angles are not very close to zero when the PHANToM is in the zero pose (right), run phantomZero from the command line and follow the prompts.



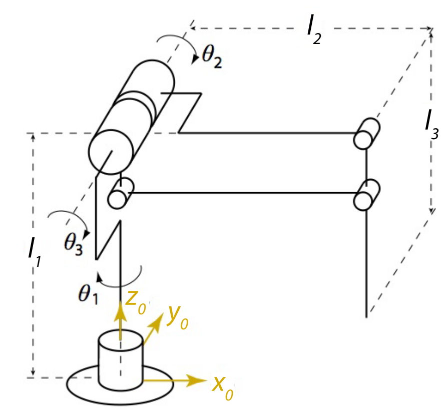
4. Start by running haptic_box_demo.m to get a feel for how it works.
5. You must have the PHANToM emergency stop down (no forces) the first time you run any code, even the demo code. Watch the graphical output to see if your code is doing what it should. The black line shows a scaled force vector.
6. You must firmly grip the PHANToM gimbal every time you run any code.
7. Someone must have their hand on the emergency stop whenever the PHANToM is running. Depress it right after your code finishes.

Please immediately post on Piazza if you notice any problems with the PHANToM!


Google Calendar
https://www.google.com/calendar/selfsched?sstoken=UUVUYTU0VmhQWWhzfGRlZmF1bHR8ZmlxZWNhOTBj
Google
katherine.kuchenbecker@gmail.com | Sign out
Google calendar
Click on an open appointment slot to sign up. If no slots are available, please try a different time range. To cancel an appointment slot you've already booked, leave this sign-up page and delete the event from your own calendar.
Today
Nov 25 – Dec 1, 2012
Refresh
Day
Week
Sun 11/25
Mon 11/26
Tue 11/27
Wed 11/28
Thu 11/29
Fri 11/30
Sat 12/1
12pm
1pm
2pm
3pm
4pm
5pm
6pm
7pm
Phantom Team Time
Created with Google calendar
Appointment calendar owner: Towne B2

PHANToM Rules

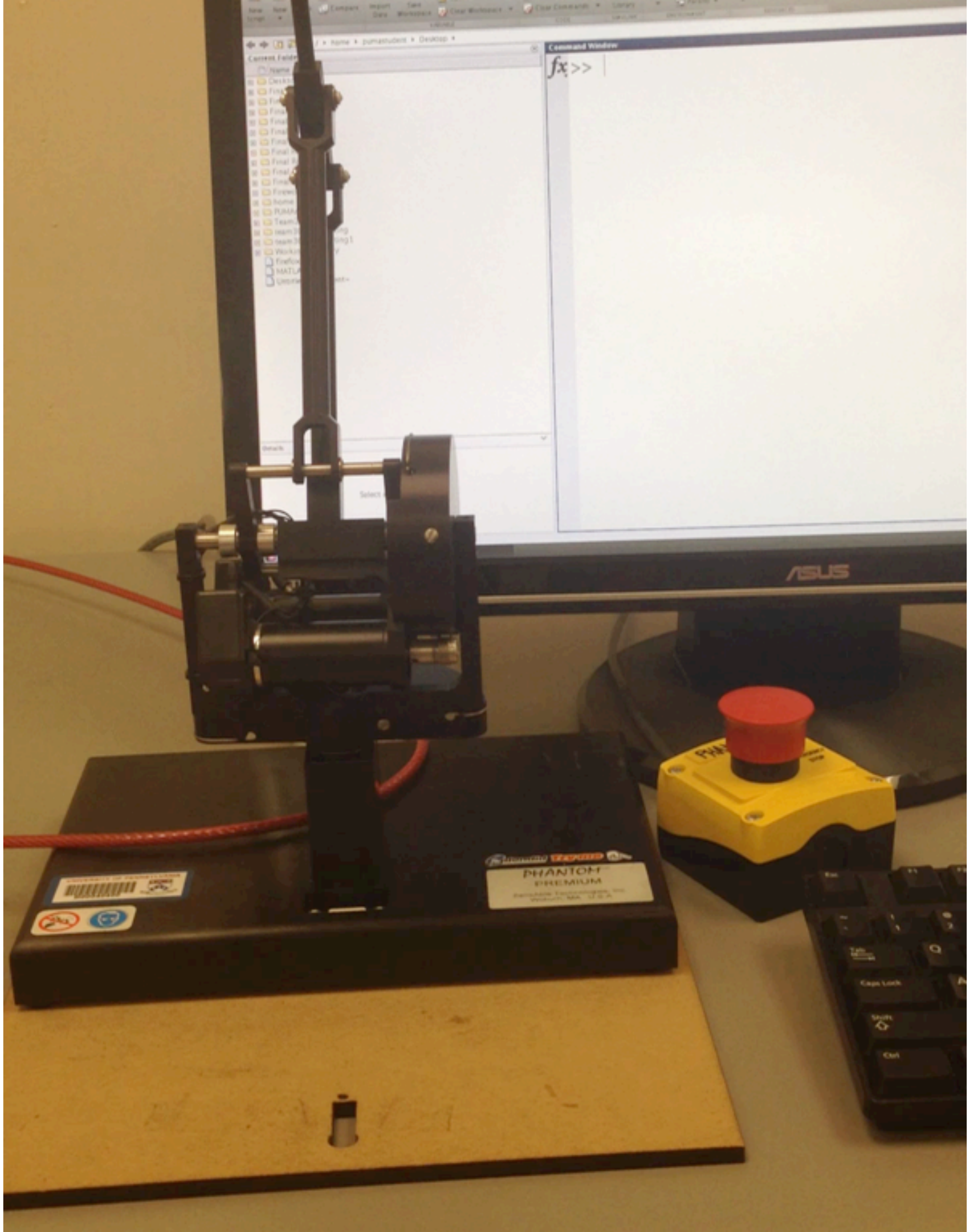
1. You must reserve the PHANToM computer in order to test your code. You may reserve only one one-hour slot at a time. Respect the next team by finishing on time.
2. Copy only your haptic_box_demo.m, haptic_ball_teamXX.m, and haptic_damping_teamXX.m scripts, not the rest of the phantom simulator files. Use your team's folder in the working directory on the computer's desktop.
3. Check the calibration of the PHANToM by calling phantomJointAngles from the MATLAB command line. If the joint angles are not very close to zero when the PHANToM is in the zero pose (right), run phantomZero from the command line and follow the prompts.



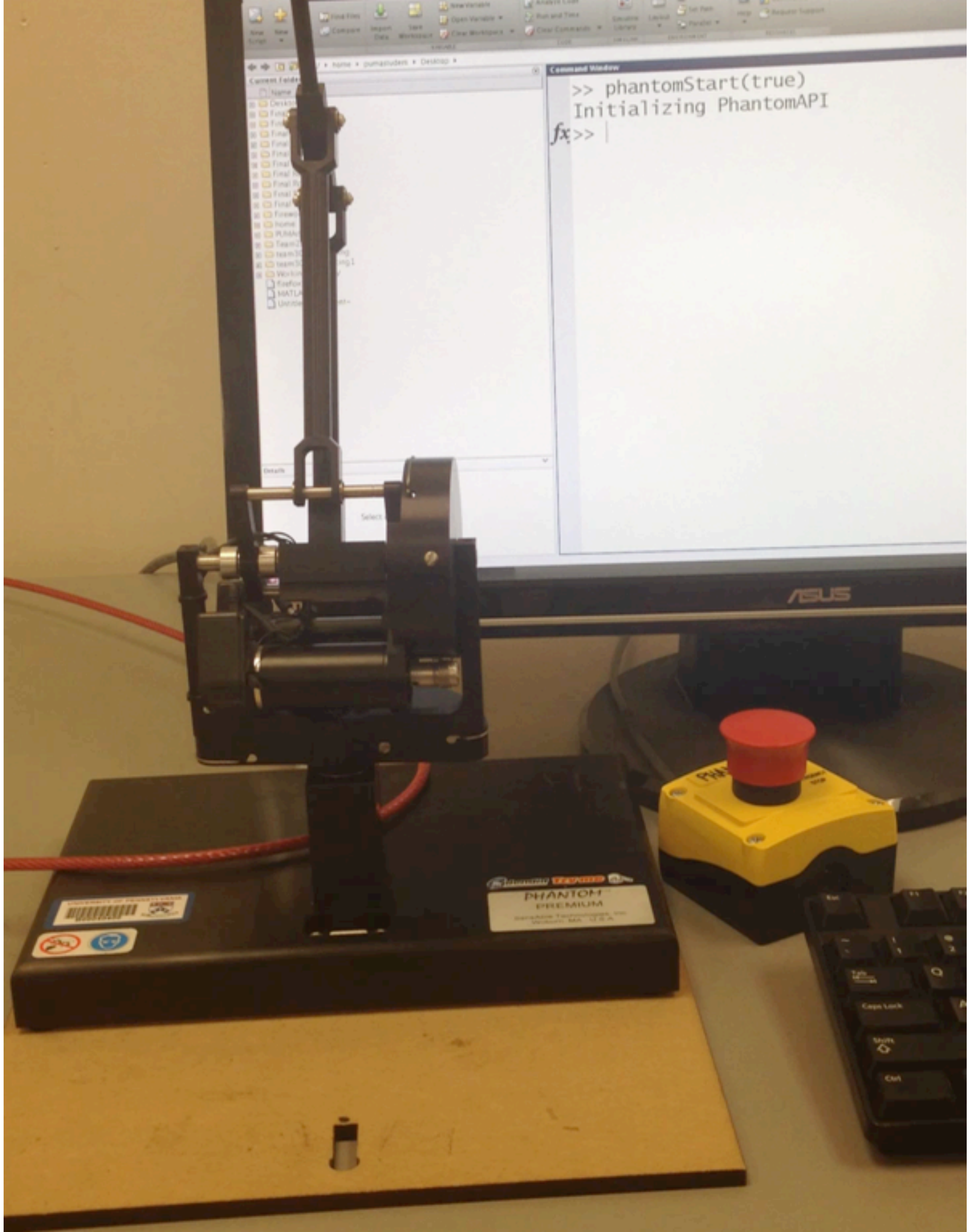
4. Start by running haptic_box_demo.m to get a feel for how it works.
5. You must have the PHANToM emergency stop down (no forces) the first time you run any code, even the demo code. Watch the graphical output to see if your code is doing what it should. The black line shows a scaled force vector.
6. You must firmly grip the PHANToM gimbal every time you run any code.
7. Someone must have their hand on the emergency stop whenever the PHANToM is running. Depress it right after your code finishes.

Please immediately post on Piazza if you notice any problems with the PHANToM!

check zero

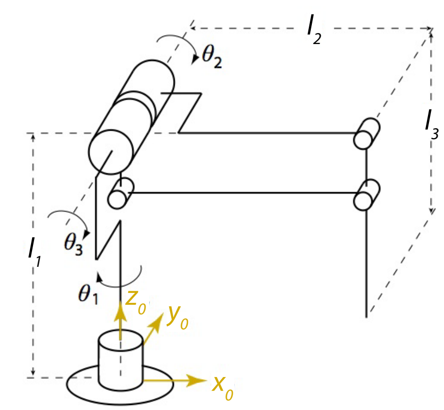


phantomZero



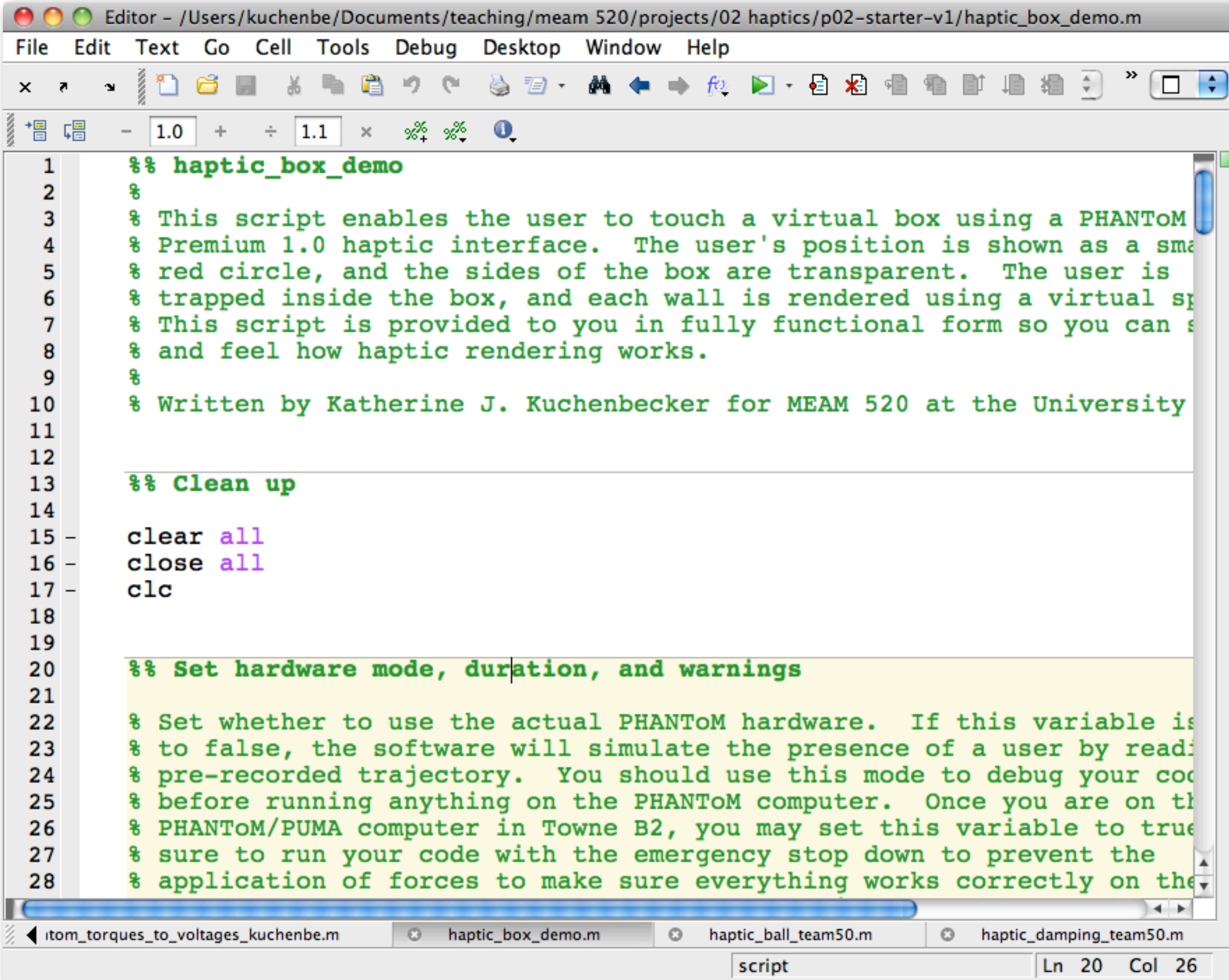
PHANToM Rules

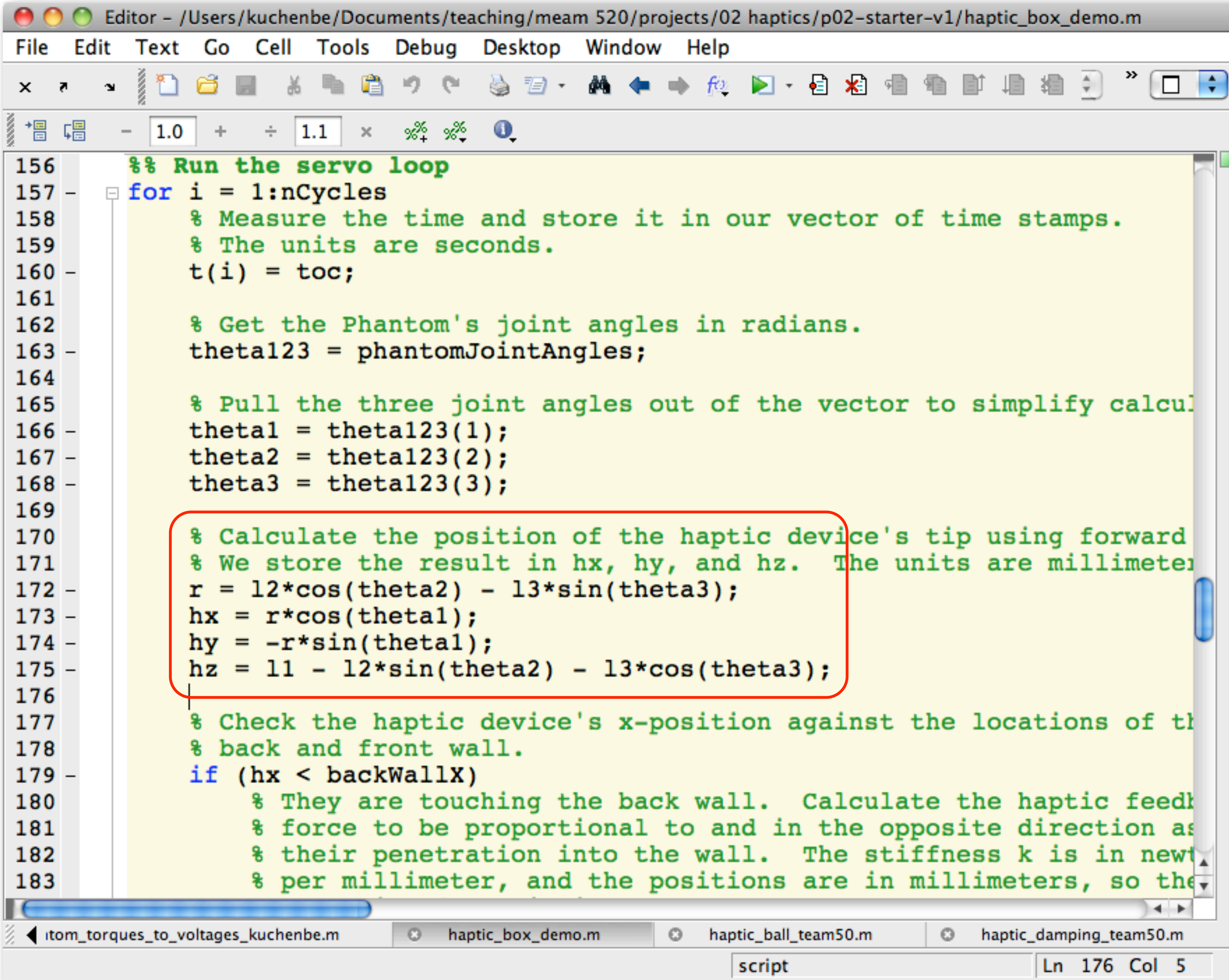
1. You must reserve the PHANToM computer in order to test your code. You may reserve only one one-hour slot at a time. Respect the next team by finishing on time.
2. Copy only your haptic_box_demo.m, haptic_ball_teamXX.m, and haptic_damping_teamXX.m scripts, not the rest of the phantom simulator files. Use your team's folder in the working directory on the computer's desktop.
3. Check the calibration of the PHANToM by calling phantomJointAngles from the MATLAB command line. If the joint angles are not very close to zero when the PHANToM is in the zero pose (right), run phantomZero from the command line and follow the prompts.

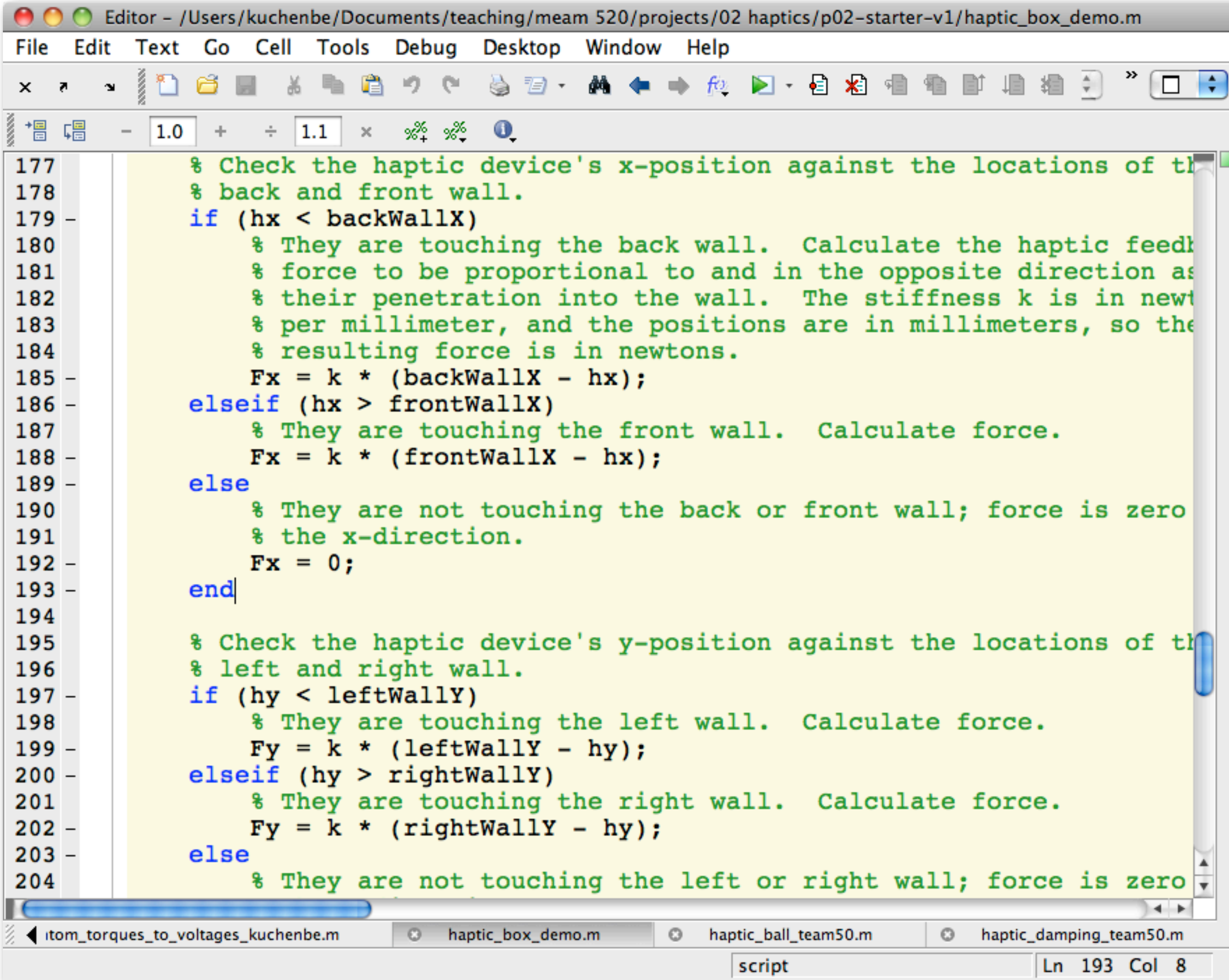


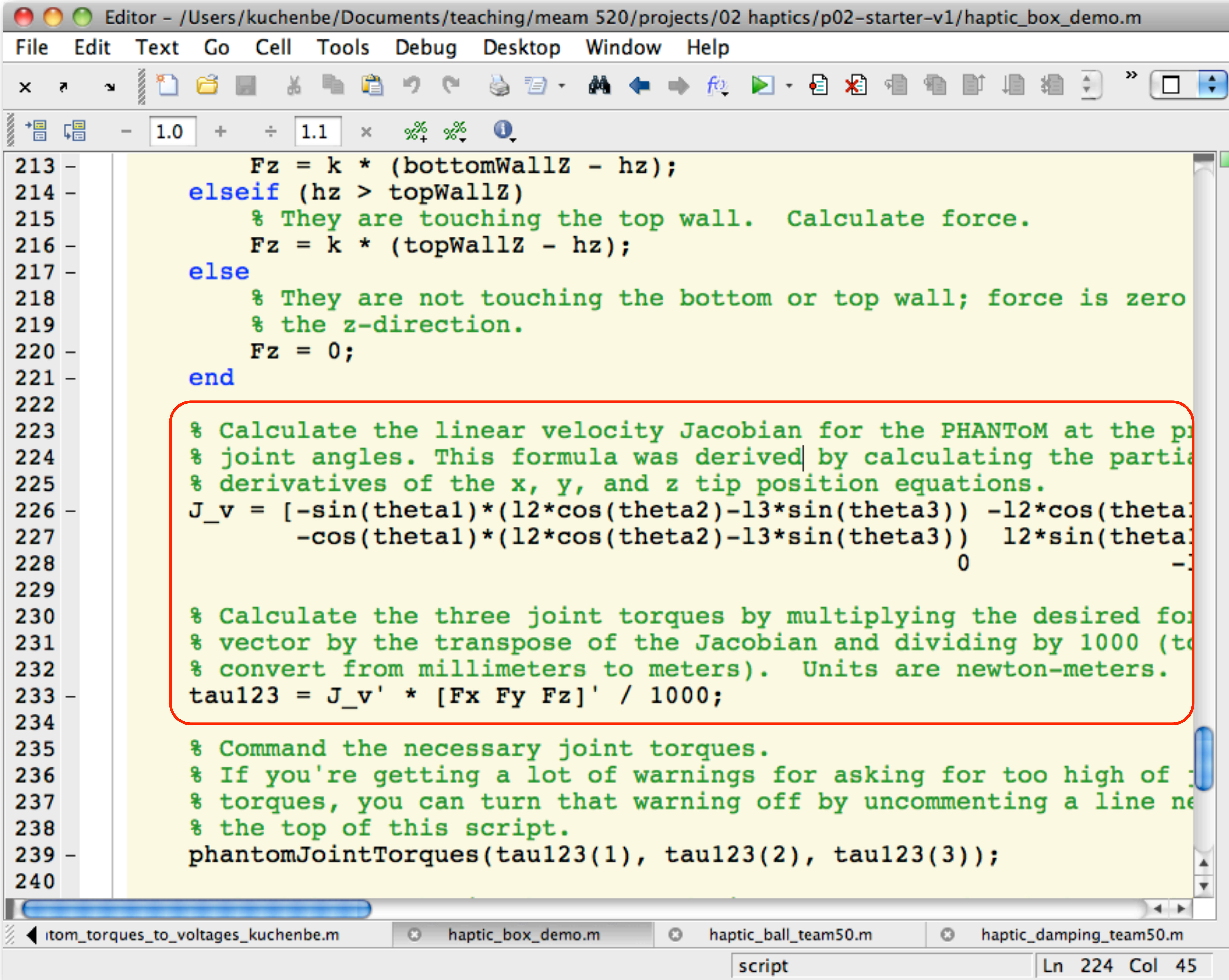
4. Start by running haptic_box_demo.m to get a feel for how it works.
5. You must have the PHANToM emergency stop down (no forces) the first time you run any code, even the demo code. Watch the graphical output to see if your code is doing what it should. The black line shows a scaled force vector.
6. You must firmly grip the PHANToM gimbal every time you run any code.
7. Someone must have their hand on the emergency stop whenever the PHANToM is running. Depress it right after your code finishes.

Please immediately post on Piazza if you notice any problems with the PHANToM!


Editor – /Users/kuchenbe/Documents/teaching/meam 520/projects/02 haptics/p02-starter-v1/haptic_box_demo.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1.0
1.1
%% haptic_box_demo
%
% This script enables the user to touch a virtual box using a PHANToM
% Premium 1.0 haptic interface. The user's position is shown as a sma
% red circle, and the sides of the box are transparent. The user is
% trapped inside the box, and each wall is rendered using a virtual sp
% This script is provided to you in fully functional form so you can s
% and feel how haptic rendering works.
%
% Written by Katherine J. Kuchenbecker for MEAM 520 at the University
%% Clean up
clear all
close all
clc
%% Set hardware mode, duration, and warnings
% Set whether to use the actual PHANToM hardware. If this variable is
% to false, the software will simulate the presence of a user by readi
% pre-recorded trajectory. You should use this mode to debug your coc
% before running anything on the PHANToM computer. Once you are on th
% PHANToM/PUMA computer in Towne B2, you may set this variable to true
% sure to run your code with the emergency stop down to prevent the
% application of forces to make sure everything works correctly on the
ıtom_torques_to_voltages_kuchenbe.m
haptic_box_demo.m
haptic_ball_team50.m
haptic_damping_team50.m
script
Ln 20 Col 26


Editor – /Users/kuchenbe/Documents/teaching/meam 520/projects/02 haptics/p02-starter-v1/haptic_box_demo.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1.0
1.1
%% Run the servo loop
for i = 1:nCycles
% Measure the time and store it in our vector of time stamps.
% The units are seconds.
t(i) = toc;
% Get the Phantom's joint angles in radians.
theta123 = phantomJointAngles;
% Pull the three joint angles out of the vector to simplify calcul
theta1 = theta123(1);
theta2 = theta123(2);
theta3 = theta123(3);
% Calculate the position of the haptic device's tip using forward
% We store the result in hx, hy, and hz. The units are millimete
r = l2*cos(theta2) - l3*sin(theta3);
hx = r*cos(theta1);
hy = -r*sin(theta1);
hz = l1 - l2*sin(theta2) - l3*cos(theta3);
% Check the haptic device's x-position against the locations of th
% back and front wall.
if (hx < backWallX)
% They are touching the back wall. Calculate the haptic feedb
% force to be proportional to and in the opposite direction as
% their penetration into the wall. The stiffness k is in newt
% per millimeter, and the positions are in millimeters, so the
itom_torques_to_voltages_kuchenbe.m
haptic_box_demo.m
haptic_ball_team50.m
haptic_damping_team50.m
script
Ln 176 Col 5

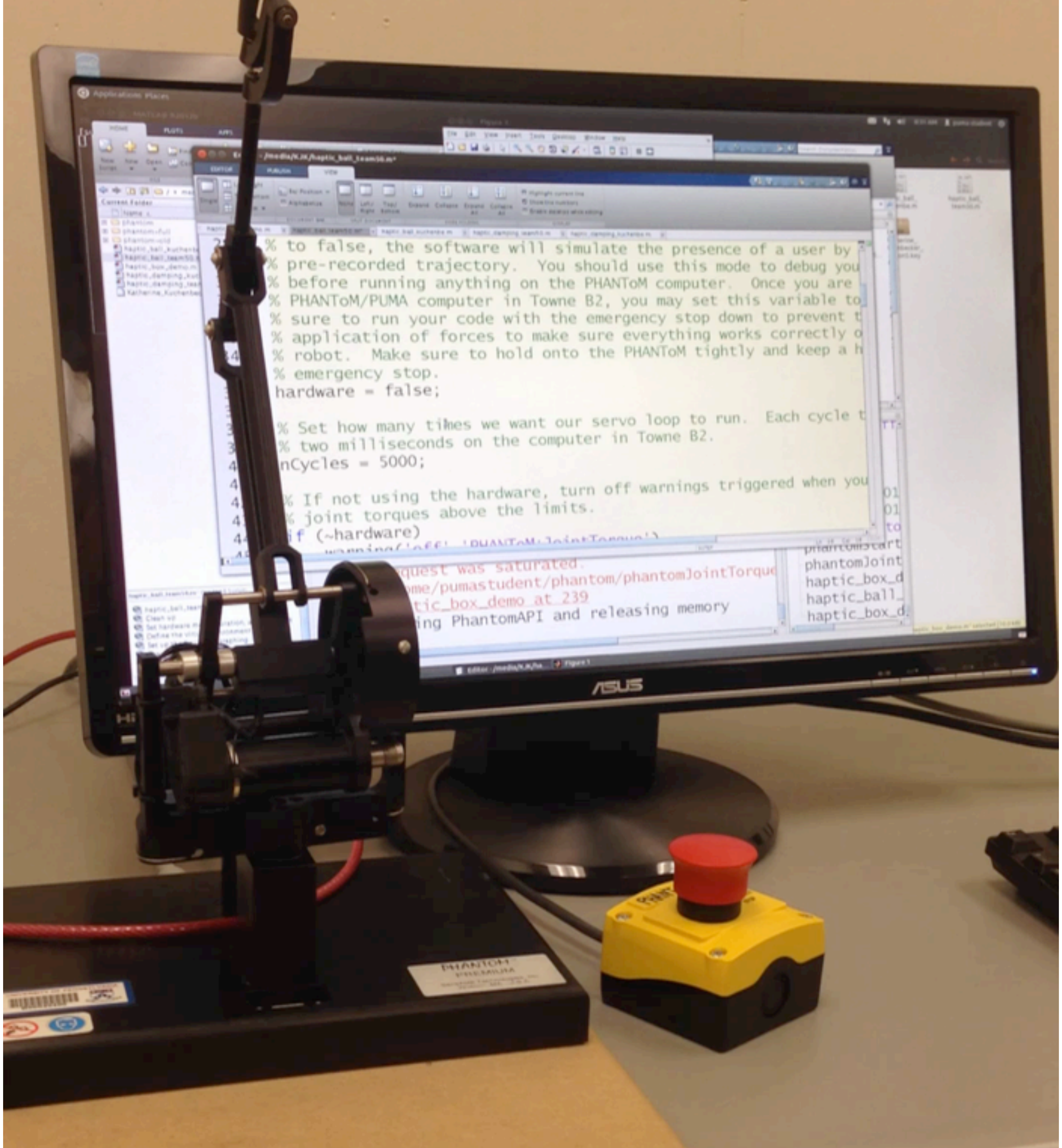

Editor – /Users/kuchenbe/Documents/teaching/meam 520/projects/02 haptics/p02-starter-v1/haptic_box_demo.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1.0
1.1
% Check the haptic device's x-position against the locations of th
% back and front wall.
if (hx < backWallX)
% They are touching the back wall. Calculate the haptic feedb
% force to be proportional to and in the opposite direction as
% their penetration into the wall. The stiffness k is in newt
% per millimeter, and the positions are in millimeters, so the
% resulting force is in newtons.
Fx = k * (backWallX - hx);
elseif (hx > frontWallX)
% They are touching the front wall. Calculate force.
Fx = k * (frontWallX - hx);
else
% They are not touching the back or front wall; force is zero
% the x-direction.
Fx = 0;
end
% Check the haptic device's y-position against the locations of th
% left and right wall.
if (hy < leftWallY)
% They are touching the left wall. Calculate force.
Fy = k * (leftWallY - hy);
elseif (hy > rightWallY)
% They are touching the right wall. Calculate force.
Fy = k * (rightWallY - hy);
else
% They are not touching the left or right wall; force is zero
itom_torques_to_voltages_kuchenbe.m
haptic_box_demo.m
haptic_ball_team50.m
haptic_damping_team50.m
script
Ln 193 Col 8


Editor – /Users/kuchenbe/Documents/teaching/meam 520/projects/02 haptics/p02-starter-v1/haptic_box_demo.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1.0
1.1
Fz = k * (bottomWallZ - hz);
elseif (hz > topWallZ)
% They are touching the top wall. Calculate force.
Fz = k * (topWallZ - hz);
else
% They are not touching the bottom or top wall; force is zero
% the z-direction.
Fz = 0;
end
% Calculate the linear velocity Jacobian for the PHANToM at the pr
% joint angles. This formula was derived by calculating the partia
% derivatives of the x, y, and z tip position equations.
J_v = [-sin(theta1)*(l2*cos(theta2)-l3*sin(theta3)) -l2*cos(theta
-cos(theta1)*(l2*cos(theta2)-l3*sin(theta3)) l2*sin(theta
0
% Calculate the three joint torques by multiplying the desired for
% vector by the transpose of the Jacobian and dividing by 1000 (to
% convert from millimeters to meters). Units are newton-meters.
tau123 = J_v' * [Fx Fy Fz]' / 1000;
% Command the necessary joint torques.
% If you're getting a lot of warnings for asking for too high of
% torques, you can turn that warning off by uncommenting a line ne
% the top of this script.
phantomJointTorques(tau123(1), tau123(2), tau123(3));
itom_torques_to_voltages_kuchenbe.m
haptic_box_demo.m
haptic_ball_team50.m
haptic_damping_team50.m
script
Ln 224 Col 45

box_demo

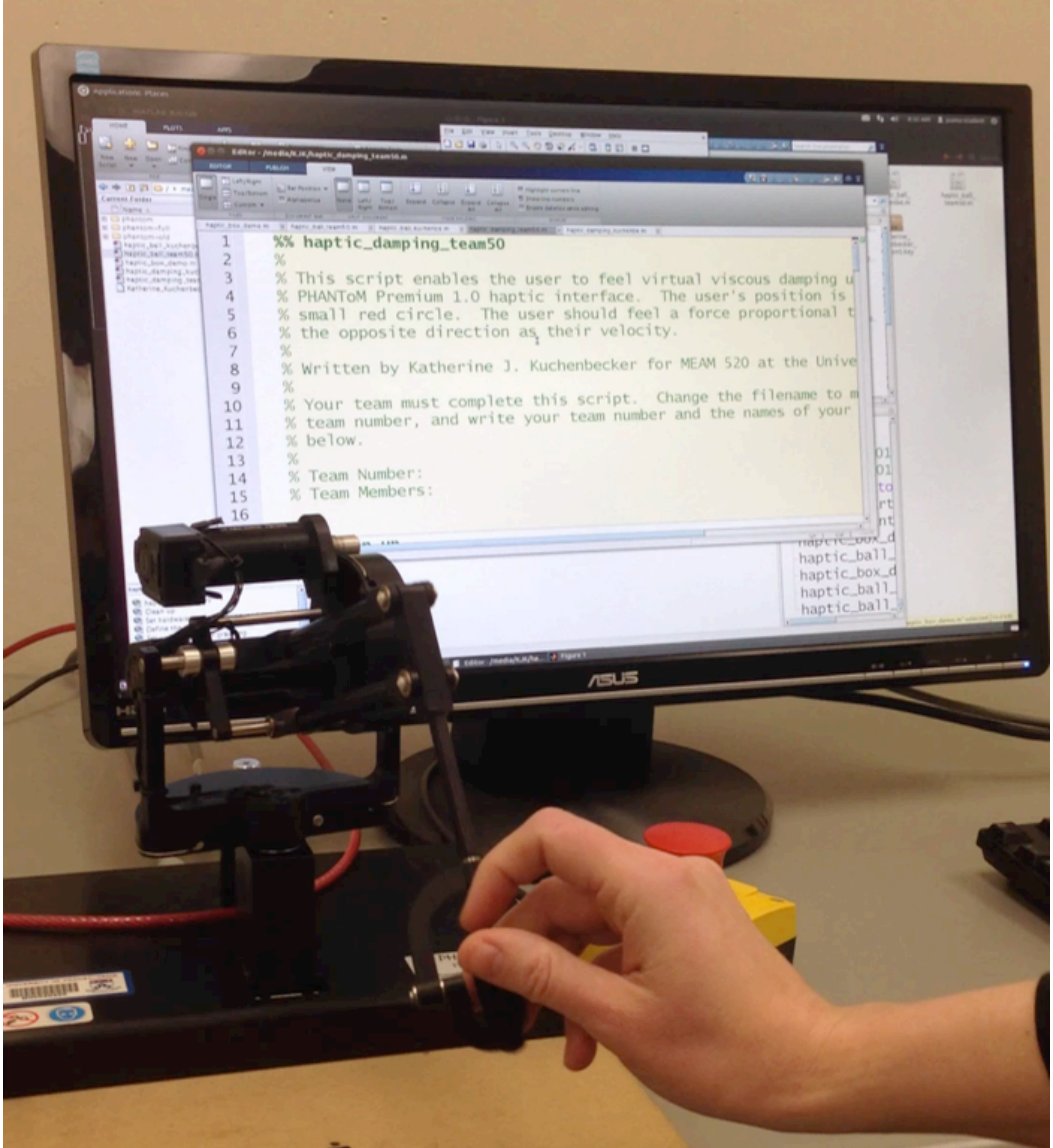
%% haptic_box_demo
%
% This script enables the user to touch a virtual box using a PHA
% Premium 1.0 haptic interface. The user's position is shown as
% red circle, and the sides of the box are transparent. The user
% trapped inside the box, and each wall is rendered using a virtu
% This script is provided to you in fully functional form so you
% and feel how haptic rendering works.
%
% Written by Katherine J. Kuchenbecker for MEAM 520 at the Univer
%% Clean up
clear all
close all
phantomJoint
haptic_box_d
haptic_ball_
haptic_box_d
ASUS

ball



% to false, the software will simulate the presence of a user by
% pre-recorded trajectory. You should use this mode to debug you
% before running anything on the PHANToM computer. Once you are
% PHANToM/PUMA computer in Towne B2, you may set this variable to
% sure to run your code with the emergency stop down to prevent t
% application of forces to make sure everything works correctly o
% robot. Make sure to hold onto the PHANToM tightly and keep a h
% emergency stop.
hardware = false;
% Set how many times we want our servo loop to run. Each cycle t
% two milliseconds on the computer in Towne B2.
nCycles = 5000;
% If not using the hardware, turn off warnings triggered when you
% joint torques above the limits.
if (~hardware)
quest was saturated.
ome/pumastudent/phantom/phantomJointTorque
tic_box_demo at 239
ing PhantomAPI and releasing memory
phantomJoint
haptic_box_d
haptic_ball_
haptic_box_d
ASUS
PHANToM
PREMIUM

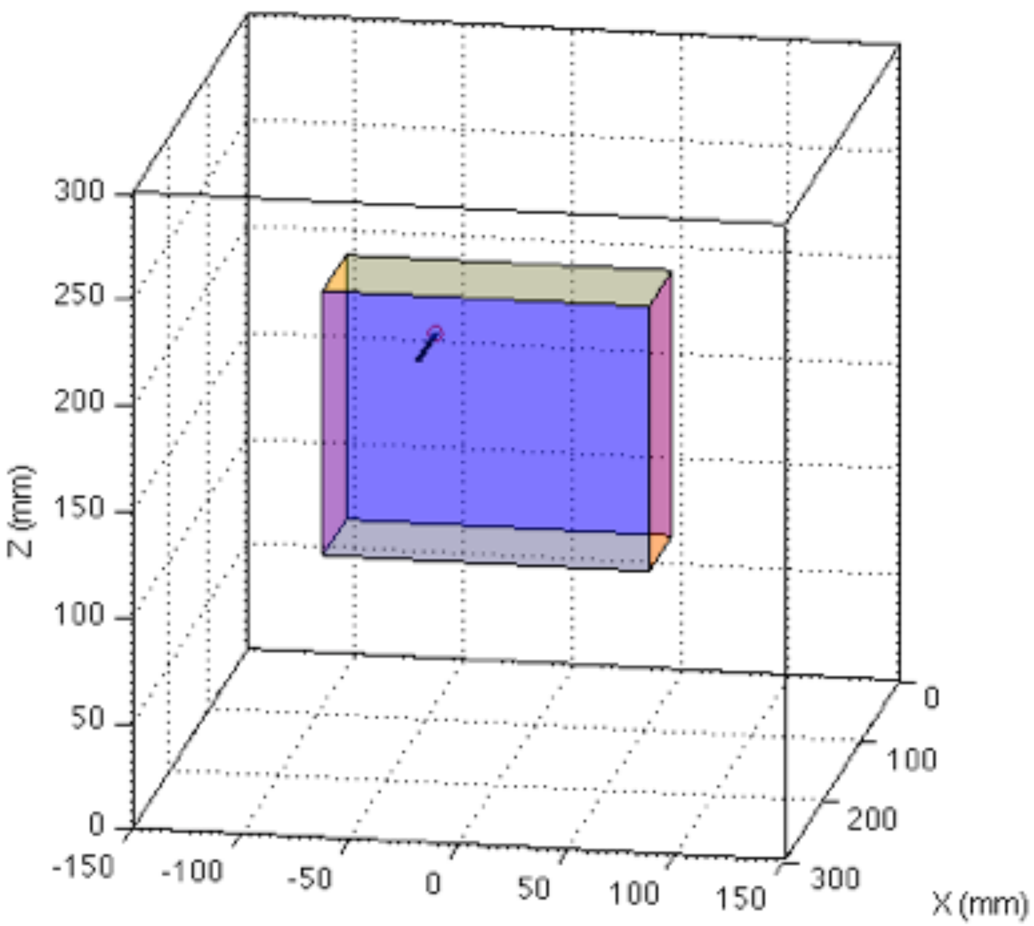
damping


%% haptic_damping_team50
%
% This script enables the user to feel virtual viscous damping u
% PHANToM Premium 1.0 haptic interface. The user's position is
% small red circle. The user should feel a force proportional t
% the opposite direction as their velocity.
%
% Written by Katherine J. Kuchenbecker for MEAM 520 at the Unive
%
% Your team must complete this script. Change the filename to m
% team number, and write your team number and the names of your
% below.
%
% Team Number:
% Team Members:
haptic_ball_
haptic_box_d
haptic_ball_
haptic_ball
ASUS

Questions ?

Project 2 Extra Credit

Update the haptic box demo so the user is *outside* the box and can touch all surfaces without popping through, especially at the corners



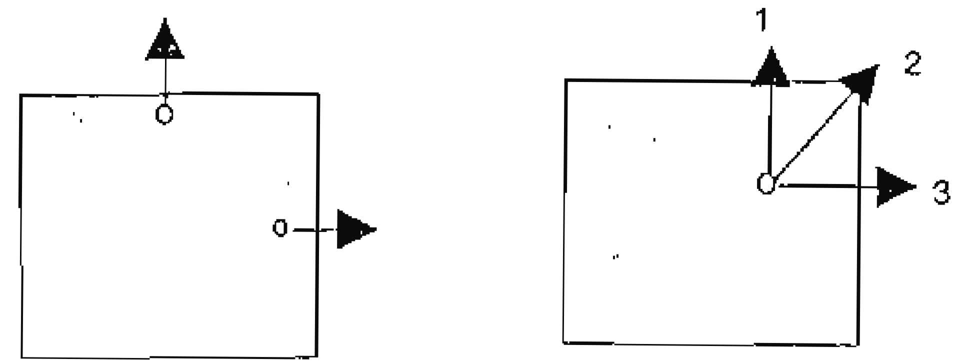


Figure 6: Virtual Cubes. Figure at left illustrates reasonable force vectors for cube with slightly compliant surface. For the point shown in figure at right, it is not clear which of forces 1, 2 and 3 should be exerted. This is path dependent.

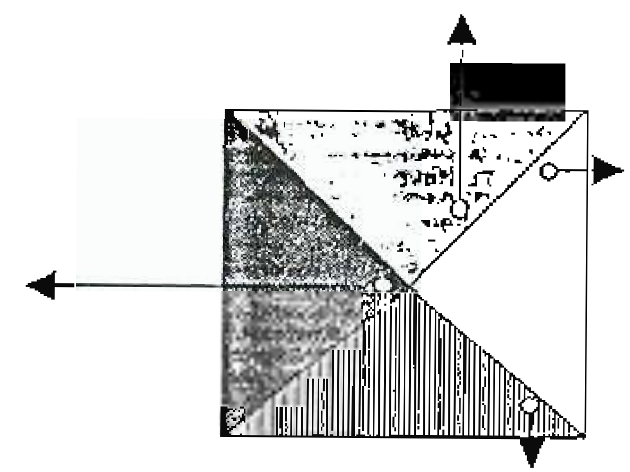


Figure 7: Solution to corner problem with cubes. Dividing cube into regions shown provides simple solution to path ambiguity problem. In 3-D, regions are pyramid in shape and permit stable behavior at edges and corners. If large forces are exerted at corners, probe point may move into the adjacent region and be pushed off object giving the sensation of "plucking" the corner.

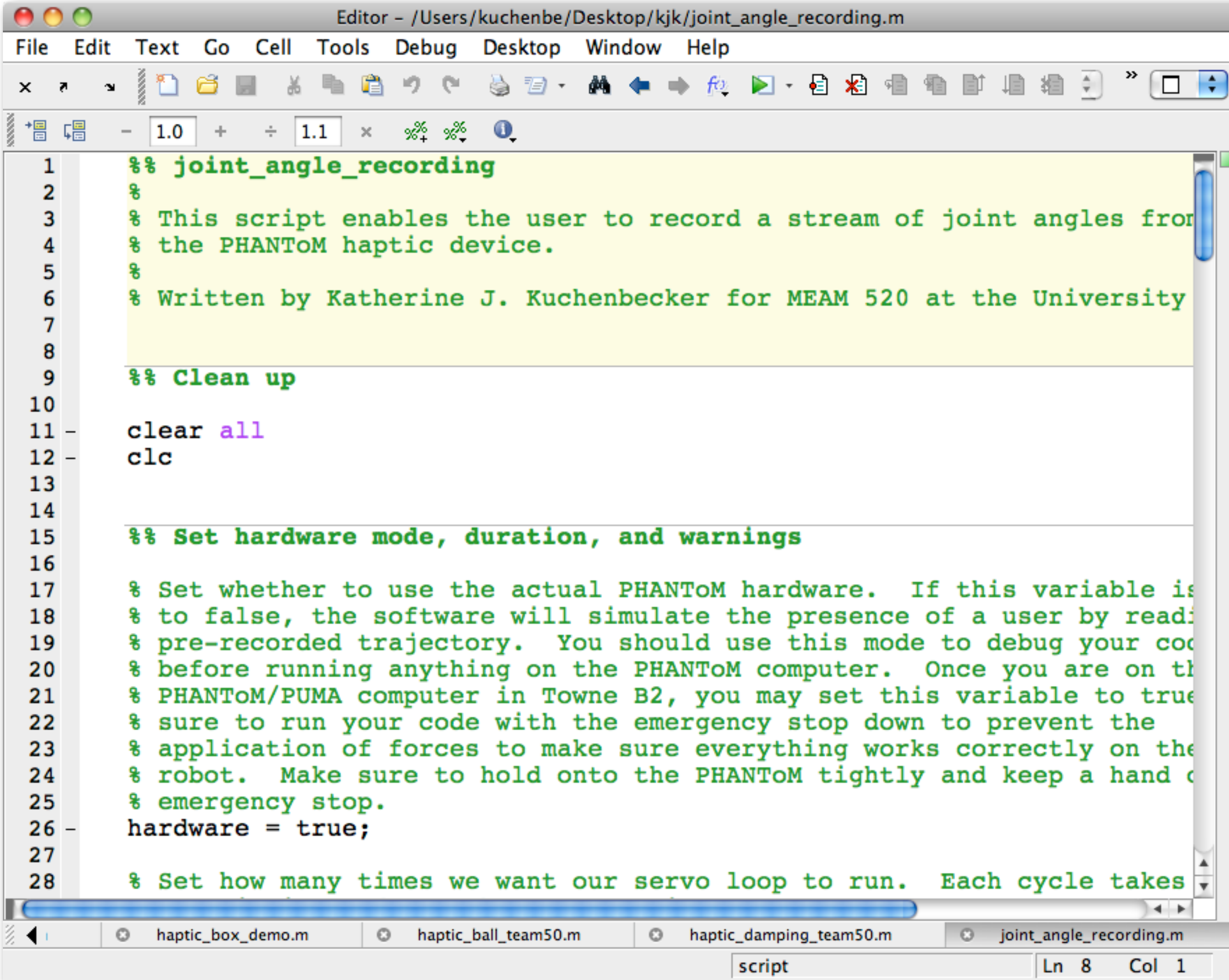
Solve it better than Massie and Salisbury....

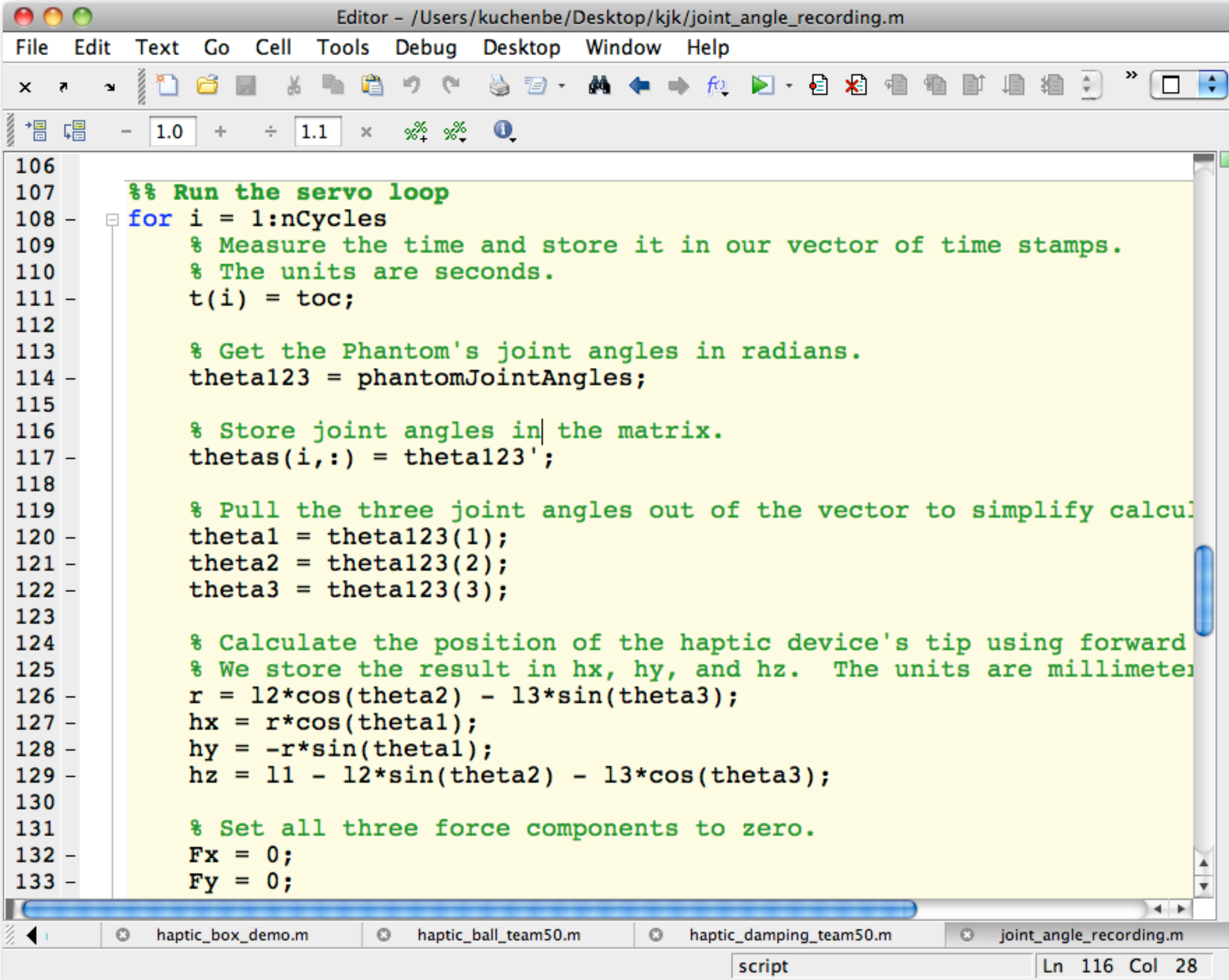
Questions ?

What else can we do with a PHANToM?

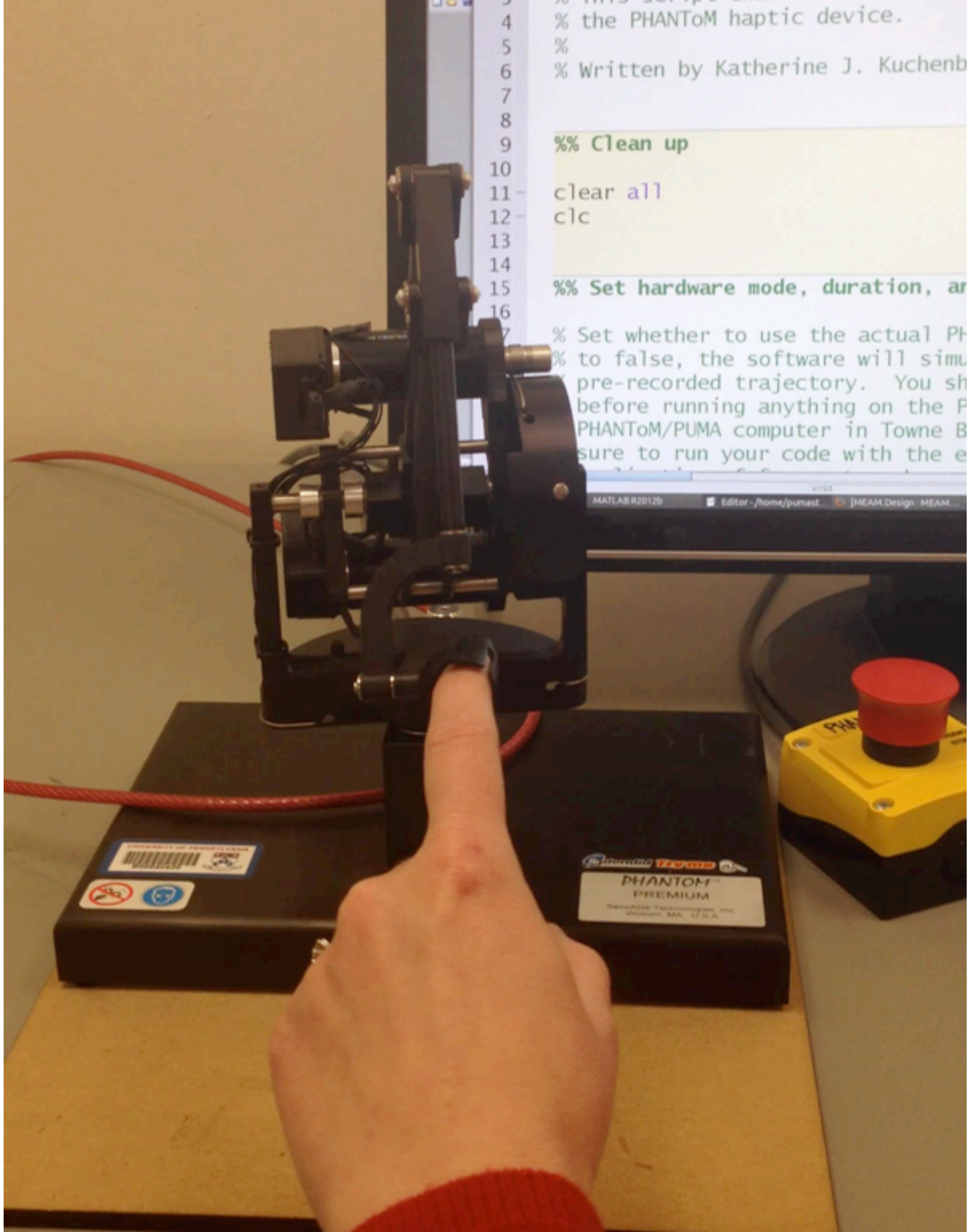
We can use it as an autonomous robot.

What movement should we have it make?

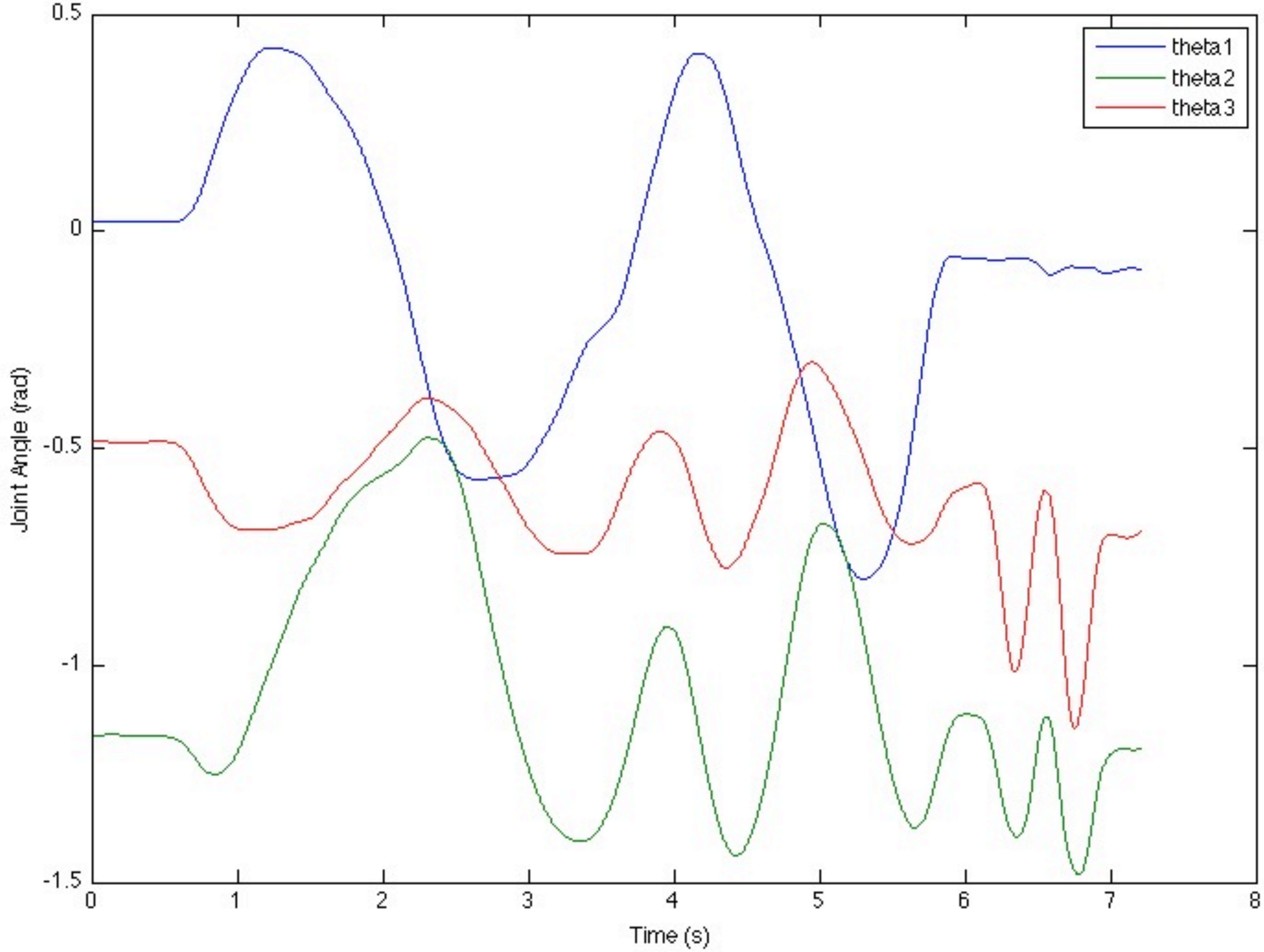

Editor – /Users/kuchenbe/Desktop/kjk/joint_angle_recording.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1.0
1.1
%% joint_angle_recording
%
% This script enables the user to record a stream of joint angles fror
% the PHANToM haptic device.
%
% Written by Katherine J. Kuchenbecker for MEAM 520 at the University
%% Clean up
clear all
clc
%% Set hardware mode, duration, and warnings
% Set whether to use the actual PHANToM hardware. If this variable is
% to false, the software will simulate the presence of a user by readi
% pre-recorded trajectory. You should use this mode to debug your coc
% before running anything on the PHANToM computer. Once you are on th
% PHANToM/PUMA computer in Towne B2, you may set this variable to true
% sure to run your code with the emergency stop down to prevent the
% application of forces to make sure everything works correctly on the
% robot. Make sure to hold onto the PHANToM tightly and keep a hand c
% emergency stop.
hardware = true;
% Set how many times we want our servo loop to run. Each cycle takes
haptic_box_demo.m
haptic_ball_team50.m
haptic_damping_team50.m
joint_angle_recording.m
script
Ln 8 Col 1


Editor – /Users/kuchenbe/Desktop/kjk/joint_angle_recording.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1.0
1.1
%% Run the servo loop
for i = 1:nCycles
% Measure the time and store it in our vector of time stamps.
% The units are seconds.
t(i) = toc;
% Get the Phantom's joint angles in radians.
theta123 = phantomJointAngles;
% Store joint angles in the matrix.
thetas(i,:) = theta123';
% Pull the three joint angles out of the vector to simplify calcul
theta1 = theta123(1);
theta2 = theta123(2);
theta3 = theta123(3);
% Calculate the position of the haptic device's tip using forward
% We store the result in hx, hy, and hz. The units are millimete
r = l2*cos(theta2) - l3*sin(theta3);
hx = r*cos(theta1);
hy = -r*sin(theta1);
hz = l1 - l2*sin(theta2) - l3*cos(theta3);
% Set all three force components to zero.
Fx = 0;
Fy = 0;
haptic_box_demo.m
haptic_ball_team50.m
haptic_damping_team50.m
joint_angle_recording.m
script
Ln 116 Col 28

record loops

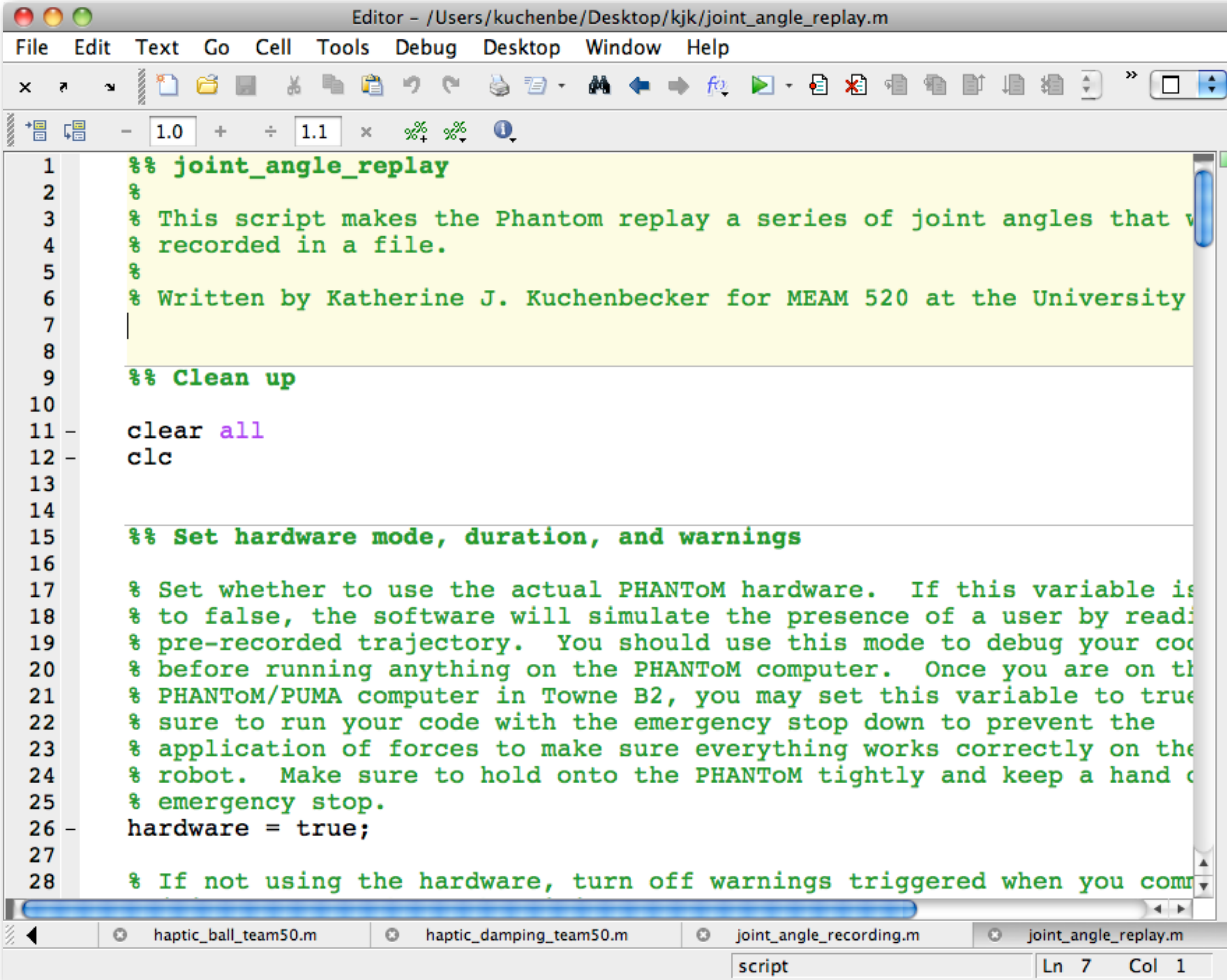


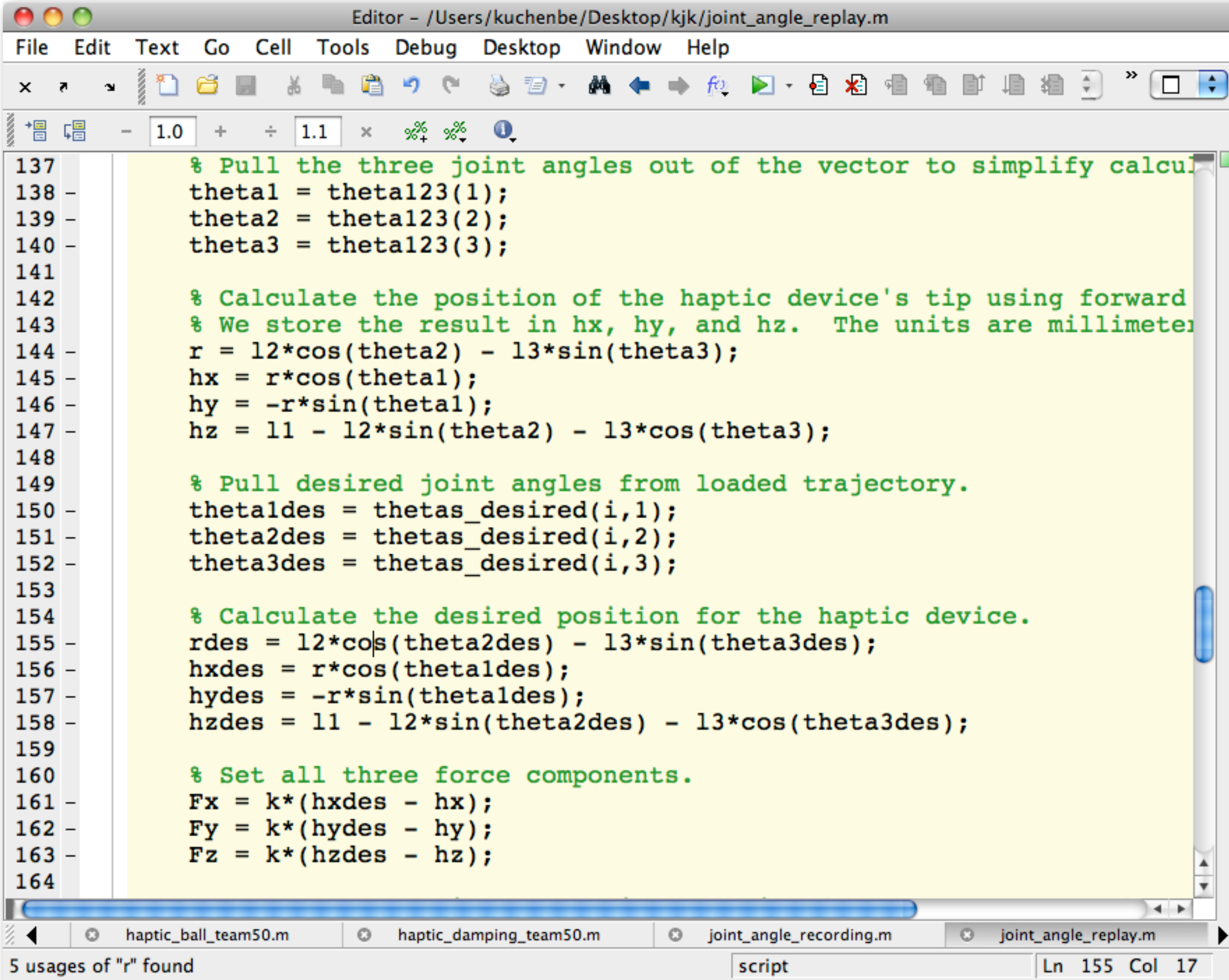
loops data



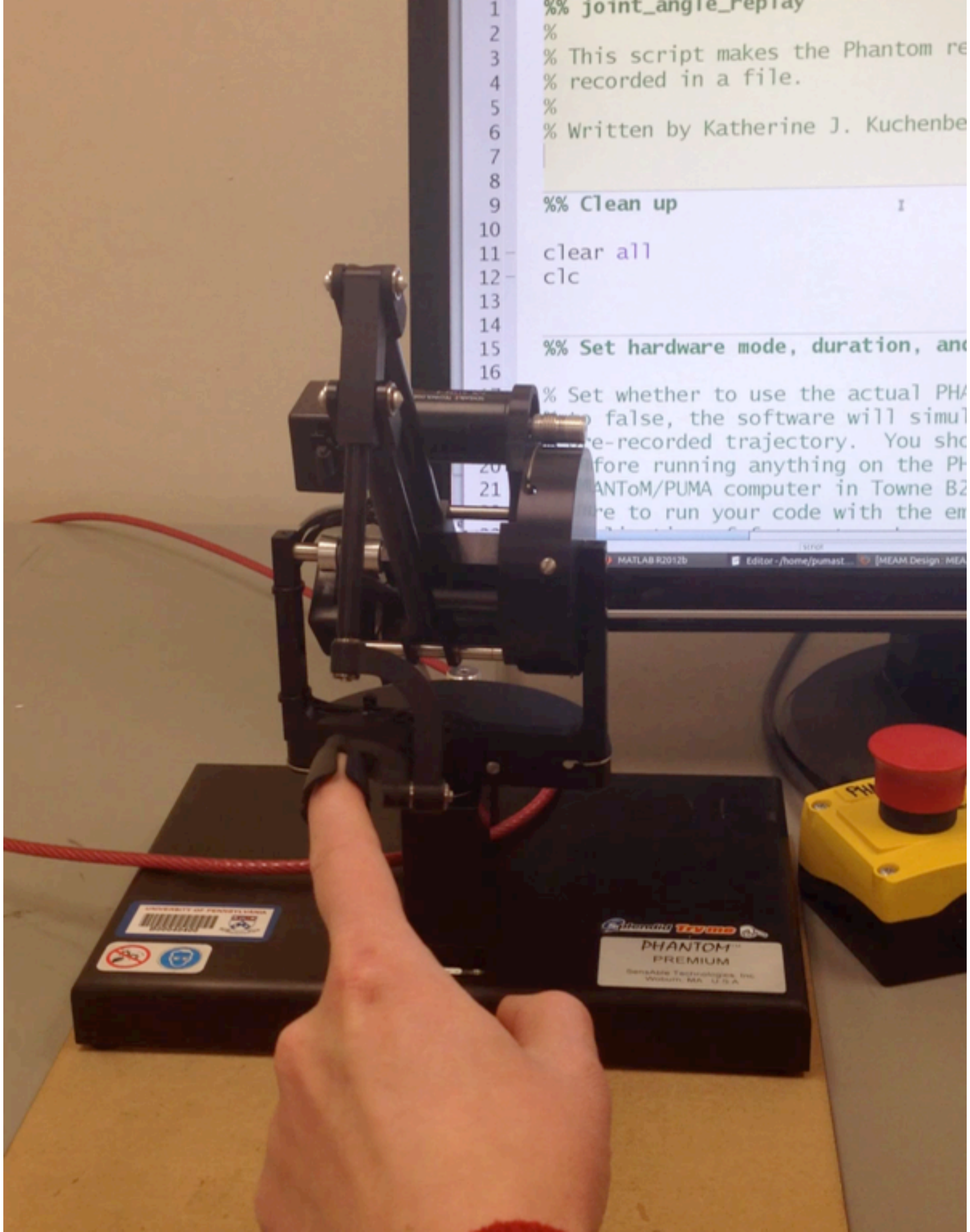
0.5
0
-0.5
-1
-1.5
0
1
2
3
4
5
6
7
8
Time (s)
Joint Angle (rad)
theta1
theta2
theta3

How can we make the PHANToM replicate this recorded movement?

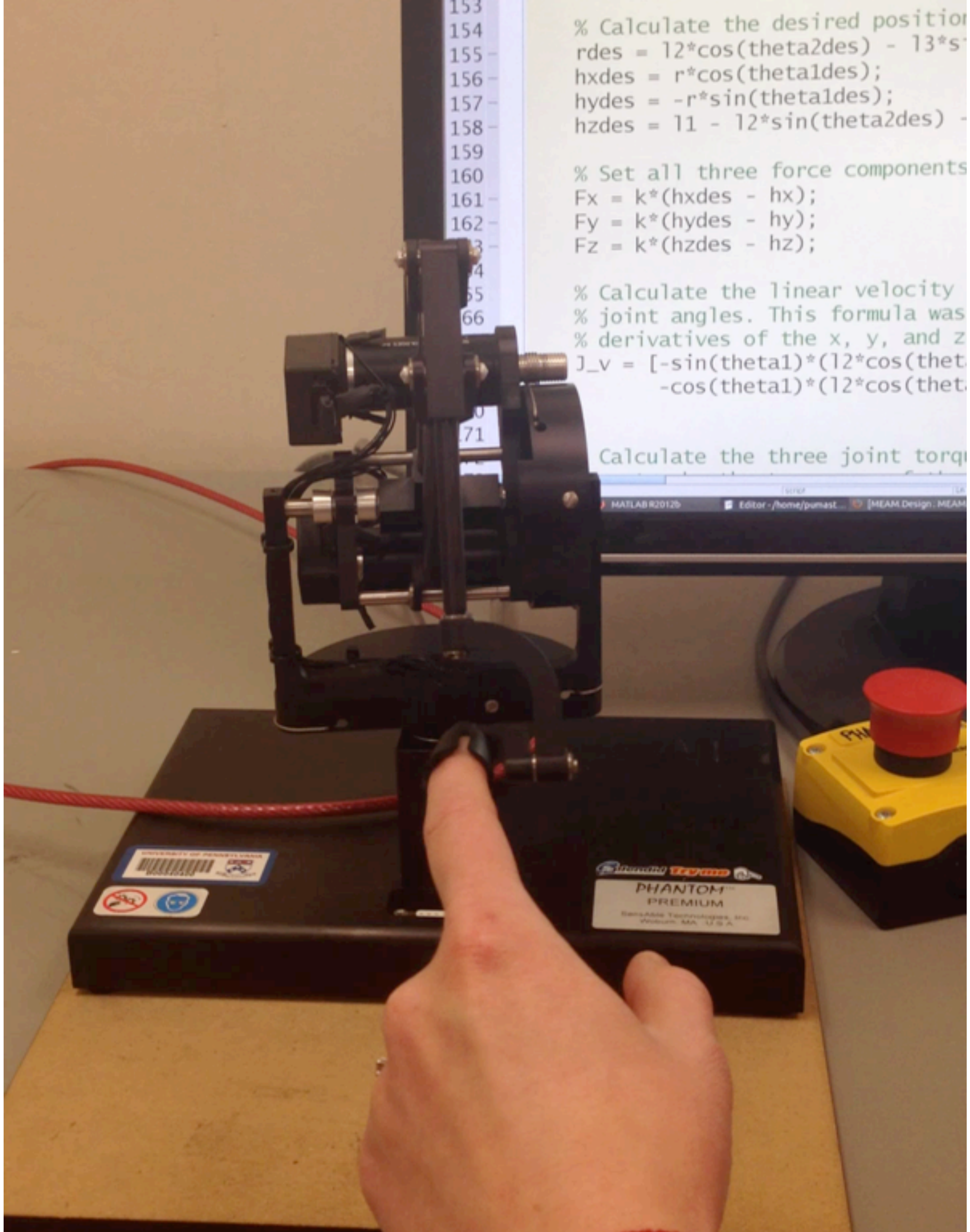

Editor – /Users/kuchenbe/Desktop/kjk/joint_angle_replay.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1.0
1.1
%% joint_angle_replay
%
% This script makes the Phantom replay a series of joint angles that w
% recorded in a file.
%
% Written by Katherine J. Kuchenbecker for MEAM 520 at the University
%% Clean up
clear all
clc
%% Set hardware mode, duration, and warnings
% Set whether to use the actual PHANToM hardware. If this variable is
% to false, the software will simulate the presence of a user by readi
% pre-recorded trajectory. You should use this mode to debug your coc
% before running anything on the PHANToM computer. Once you are on th
% PHANToM/PUMA computer in Towne B2, you may set this variable to true
% sure to run your code with the emergency stop down to prevent the
% application of forces to make sure everything works correctly on the
% robot. Make sure to hold onto the PHANToM tightly and keep a hand c
% emergency stop.
hardware = true;
% If not using the hardware, turn off warnings triggered when you comm
haptic_ball_team50.m
haptic_damping_team50.m
joint_angle_recording.m
joint_angle_replay.m
script
Ln 7 Col 1


Editor – /Users/kuchenbe/Desktop/kjk/joint_angle_replay.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1.0
1.1
% Pull the three joint angles out of the vector to simplify calcu
theta1 = theta123(1);
theta2 = theta123(2);
theta3 = theta123(3);
% Calculate the position of the haptic device's tip using forward
% We store the result in hx, hy, and hz. The units are millimete
r = l2*cos(theta2) - l3*sin(theta3);
hx = r*cos(theta1);
hy = -r*sin(theta1);
hz = l1 - l2*sin(theta2) - l3*cos(theta3);
% Pull desired joint angles from loaded trajectory.
theta1des = thetas_desired(i,1);
theta2des = thetas_desired(i,2);
theta3des = thetas_desired(i,3);
% Calculate the desired position for the haptic device.
rdes = l2*cos(theta2des) - l3*sin(theta3des);
hxdes = r*cos(theta1des);
hydes = -r*sin(theta1des);
hzdes = l1 - l2*sin(theta2des) - l3*cos(theta3des);
% Set all three force components.
Fx = k*(hxdes - hx);
Fy = k*(hydes - hy);
Fz = k*(hzdes - hz);
haptic_ball_team50.m
haptic_damping_team50.m
joint_angle_recording.m
joint_angle_replay.m
5 usages of "r" found
script
Ln 155 Col 17

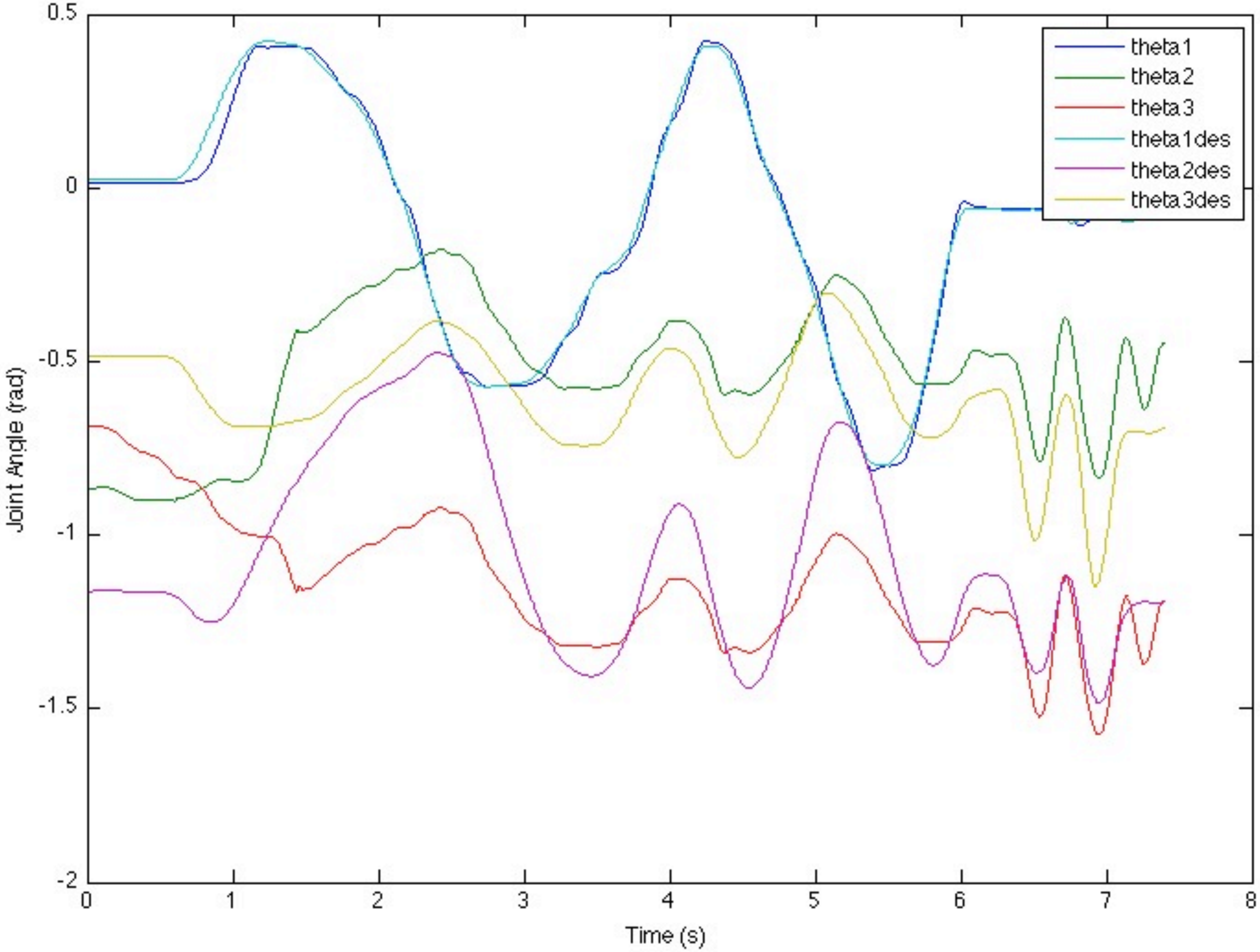
replay loops:
no forces



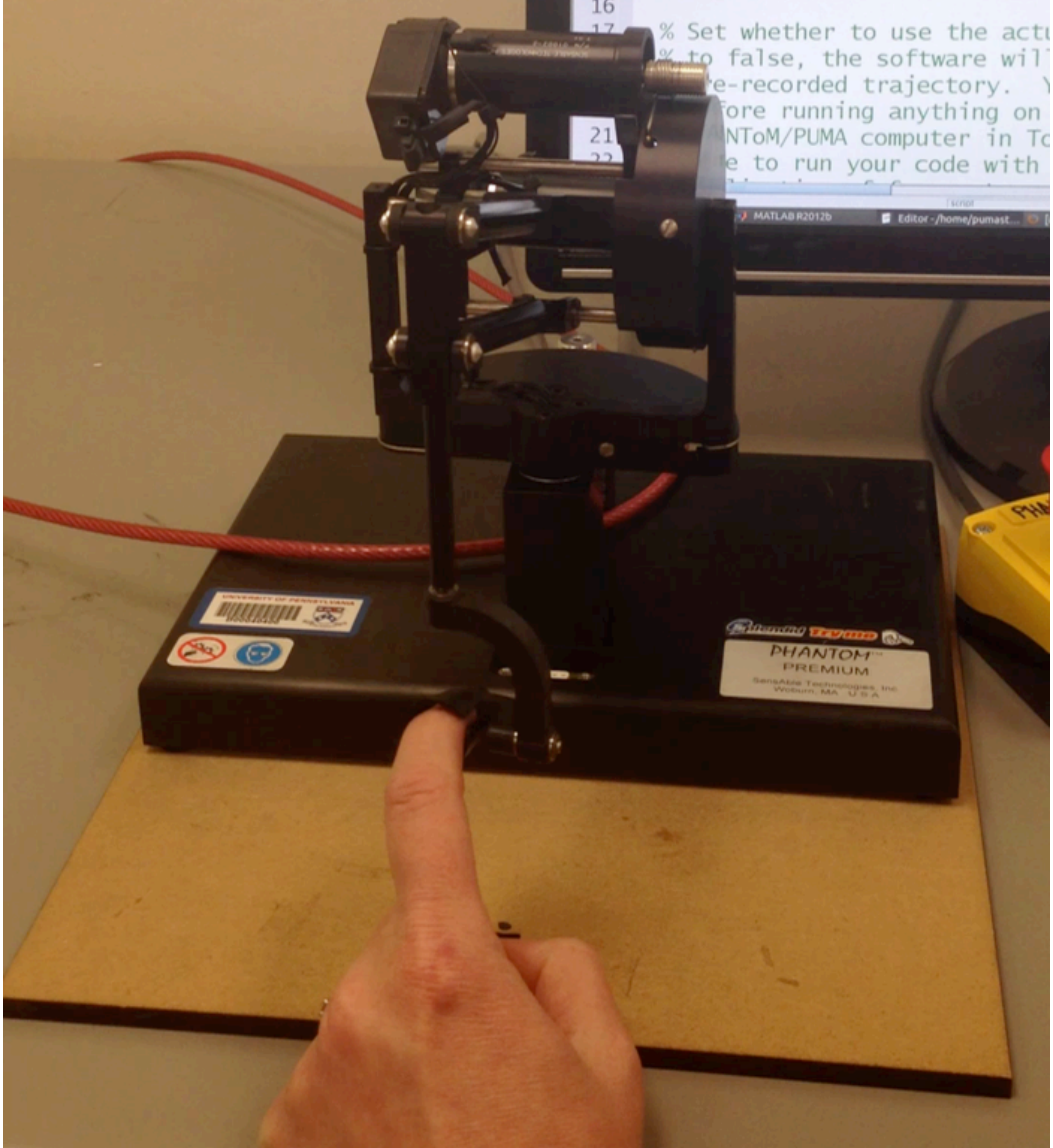
replay loops:
spring force

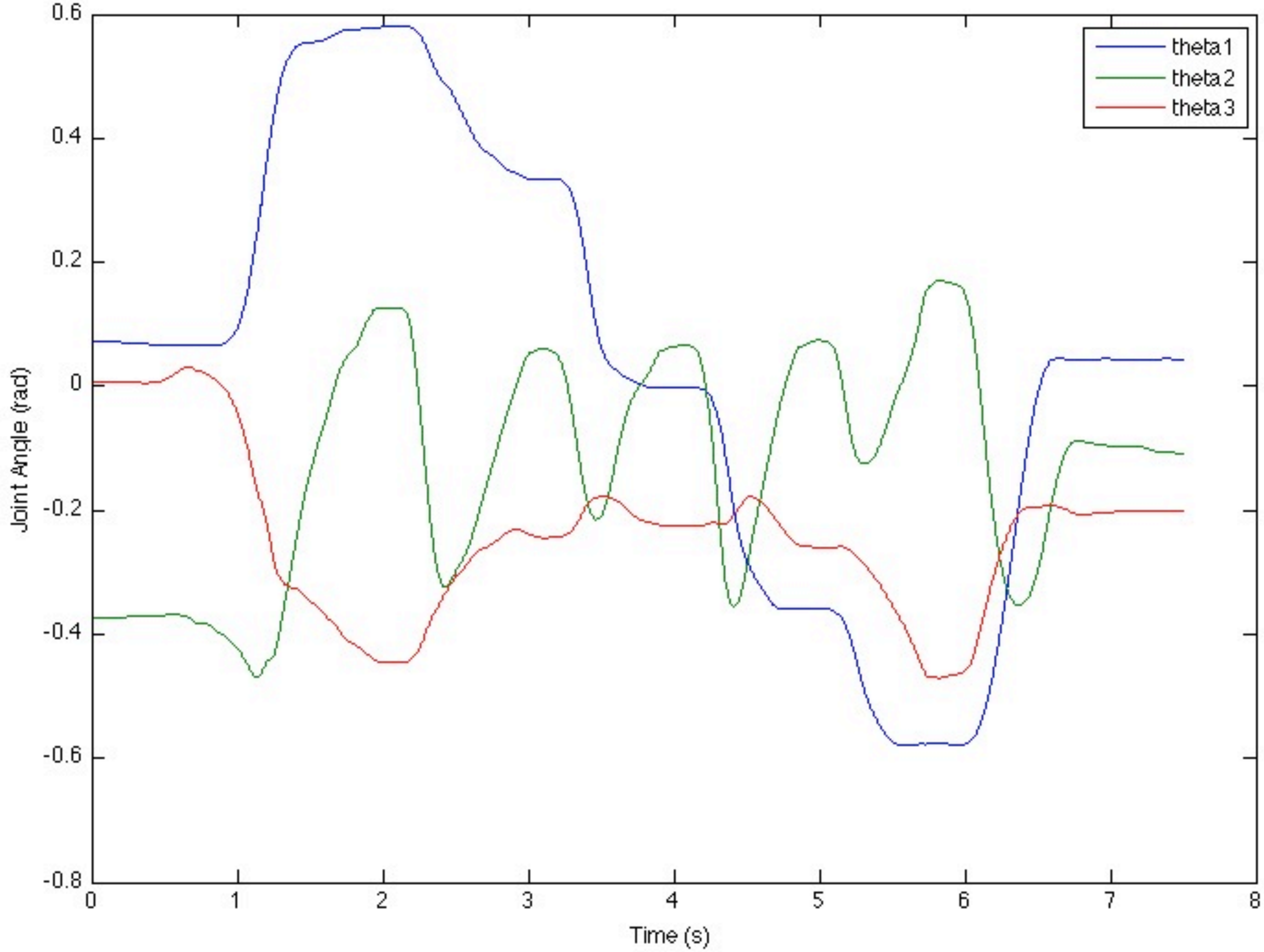


Does this look good?

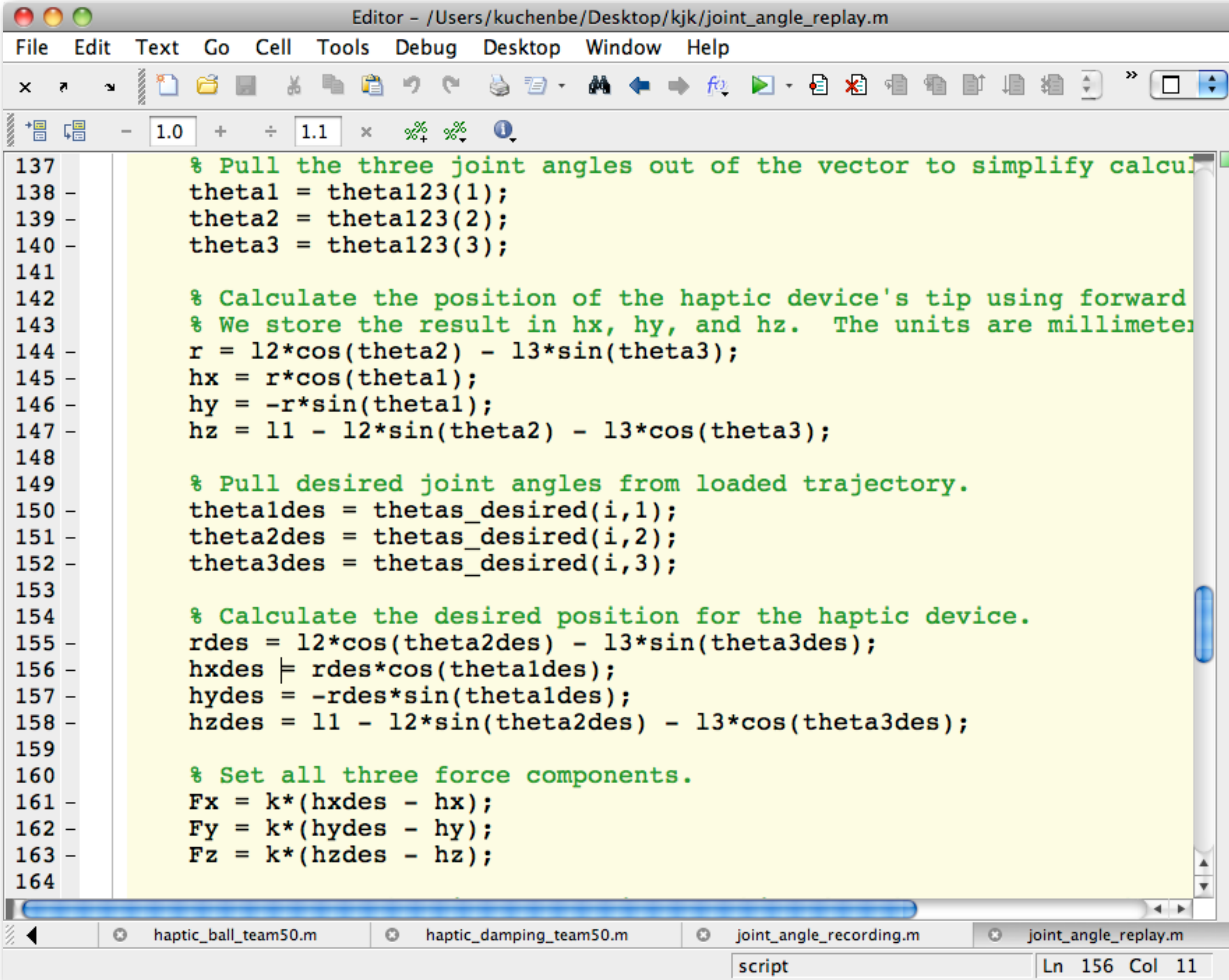


record taps



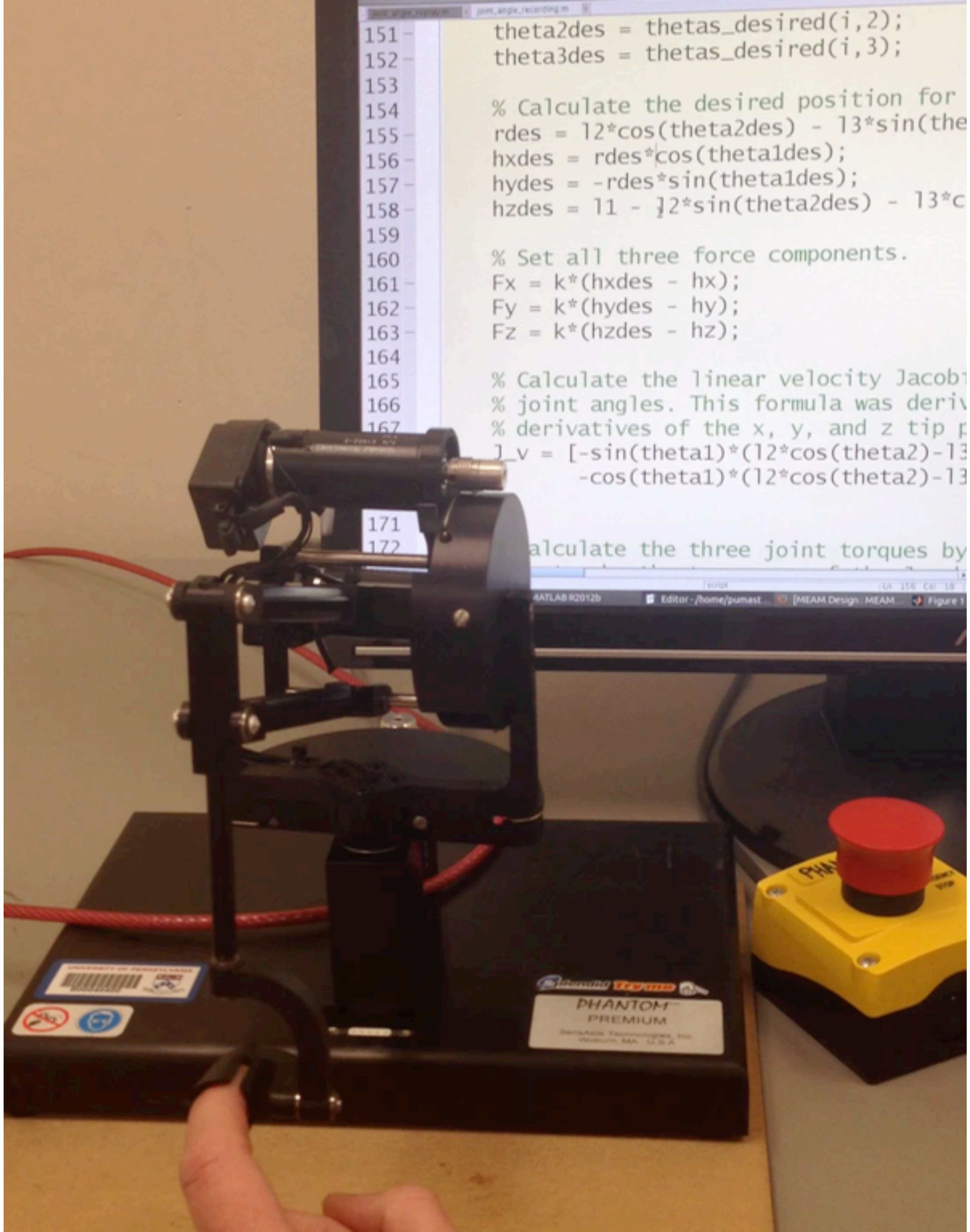


theta1
theta2
theta3
Joint Angle (rad)
Time (s)
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
1
2
3
4
5
6
7
8

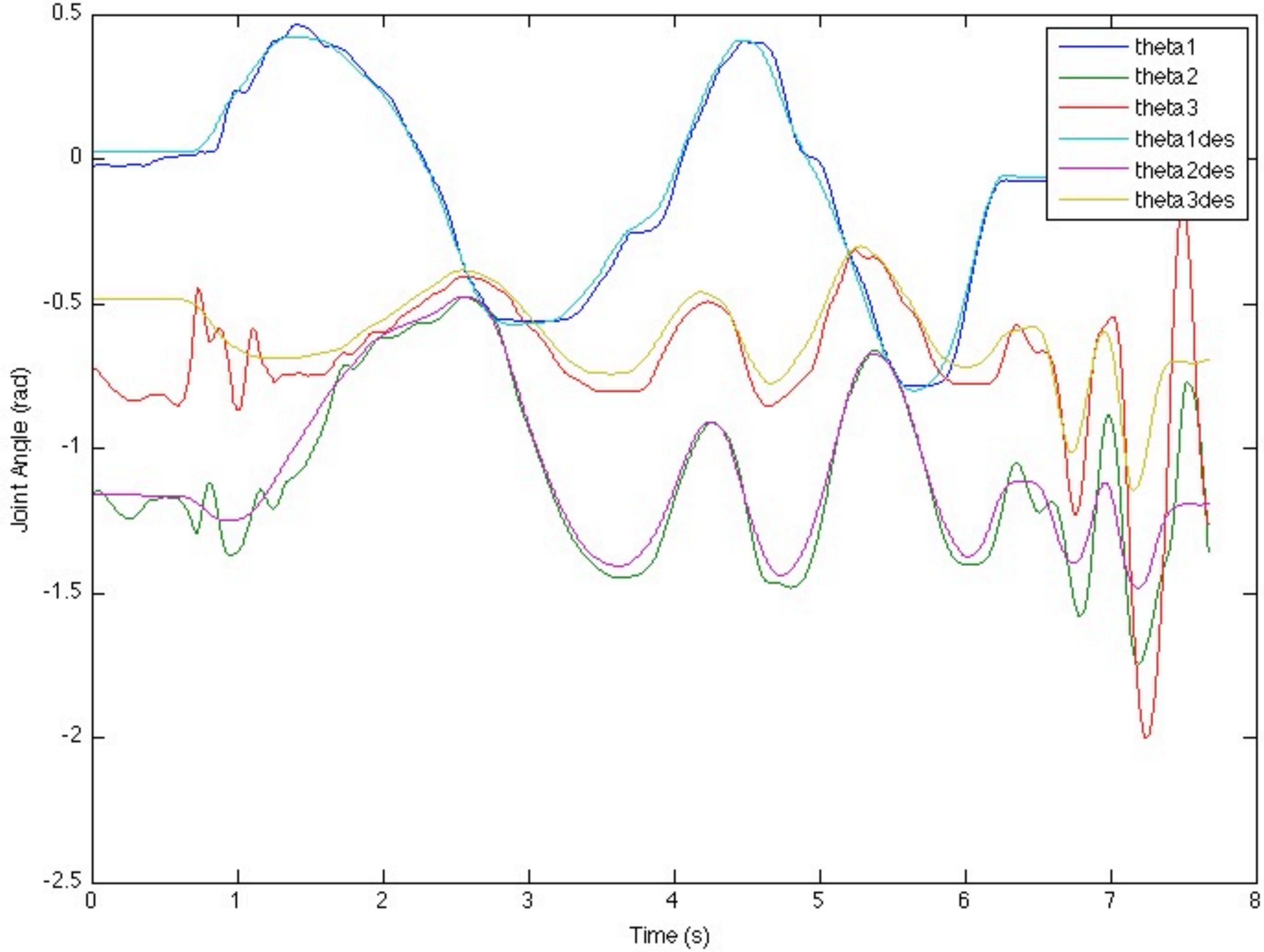


Editor - /Users/kuchenbe/Desktop/kjk/joint_angle_replay.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1.0
1.1
% Pull the three joint angles out of the vector to simplify calcul
theta1 = theta123(1);
theta2 = theta123(2);
theta3 = theta123(3);
% Calculate the position of the haptic device's tip using forward
% We store the result in hx, hy, and hz. The units are millimeter
r = l2*cos(theta2) - l3*sin(theta3);
hx = r*cos(theta1);
hy = -r*sin(theta1);
hz = l1 - l2*sin(theta2) - l3*cos(theta3);
% Pull desired joint angles from loaded trajectory.
theta1des = thetas_desired(i,1);
theta2des = thetas_desired(i,2);
theta3des = thetas_desired(i,3);
% Calculate the desired position for the haptic device.
rdes = l2*cos(theta2des) - l3*sin(theta3des);
hxdes = rdes*cos(theta1des);
hydes = -rdes*sin(theta1des);
hzdes = l1 - l2*sin(theta2des) - l3*cos(theta3des);
% Set all three force components.
Fx = k*(hxdes - hx);
Fy = k*(hydes - hy);
Fz = k*(hzdes - hz);
haptic_ball_team50.m
haptic_damping_team50.m
joint_angle_recording.m
joint_angle_replay.m
script
Ln 156 Col 11

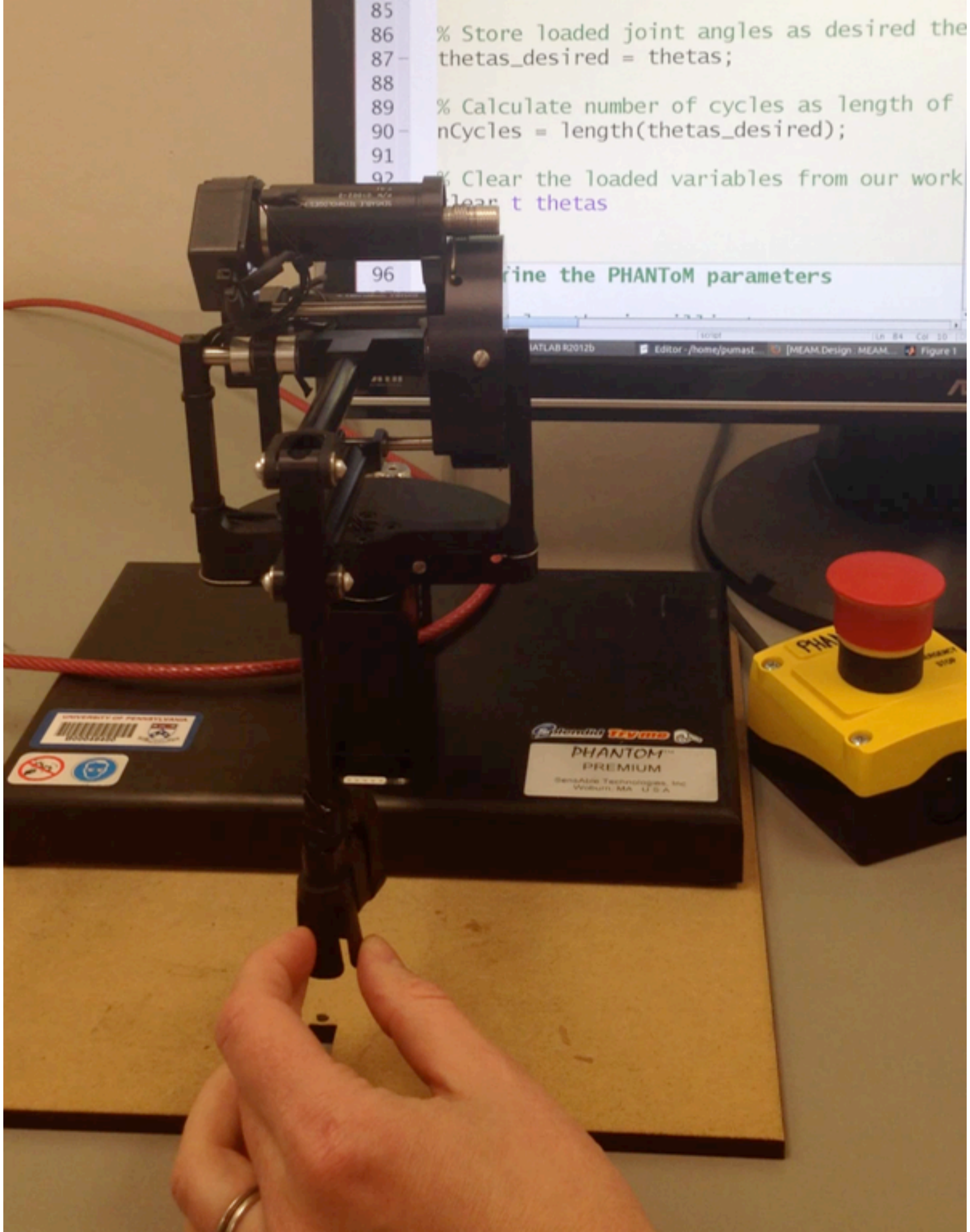
replay loops:
spring force,
fixed kinematics



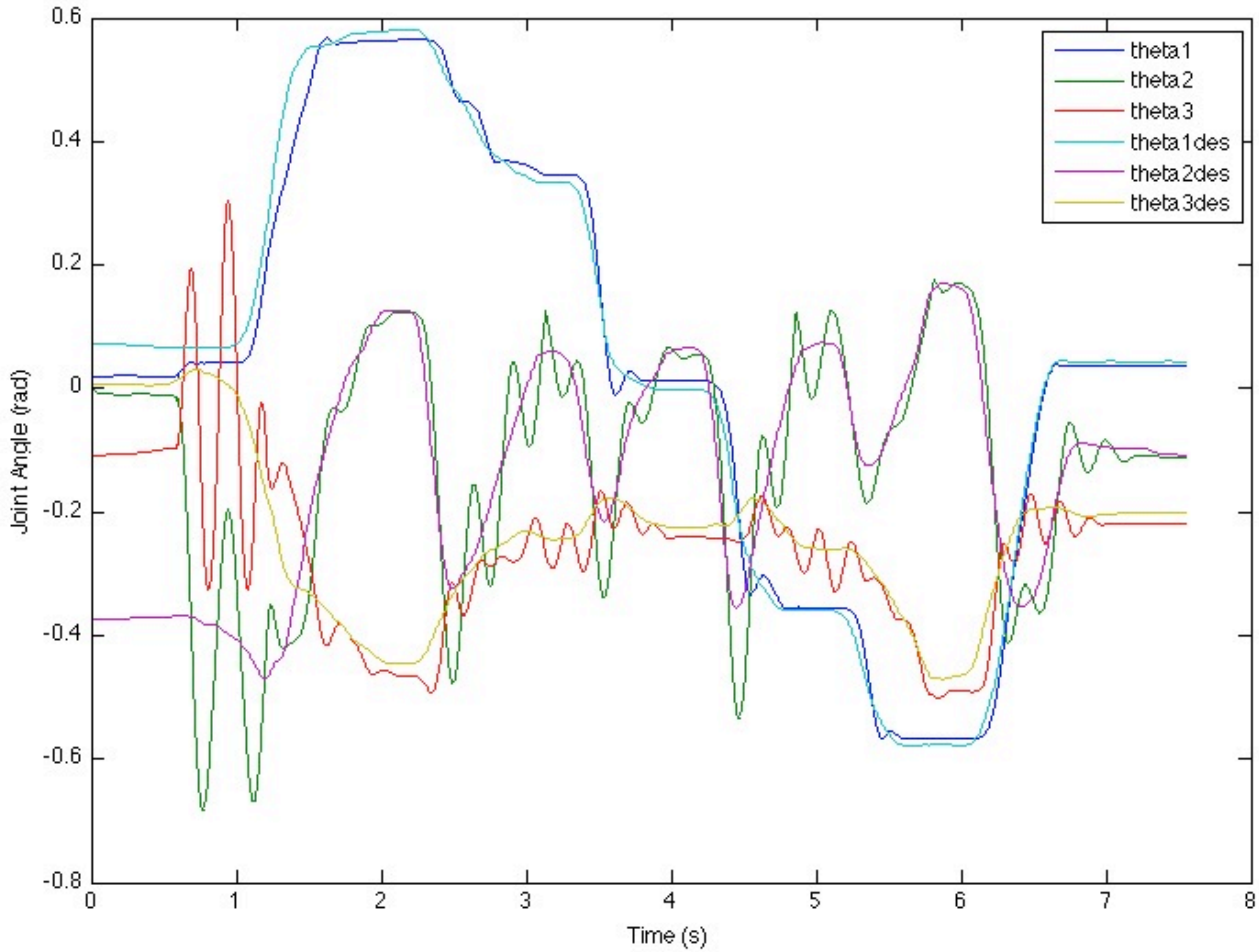
Much better!


theta1
theta2
theta3
theta1des
theta2des
theta3des
Joint Angle (rad)
Time (s)
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
1
2
3
4
5
6
7
8

replay tap:
spring force

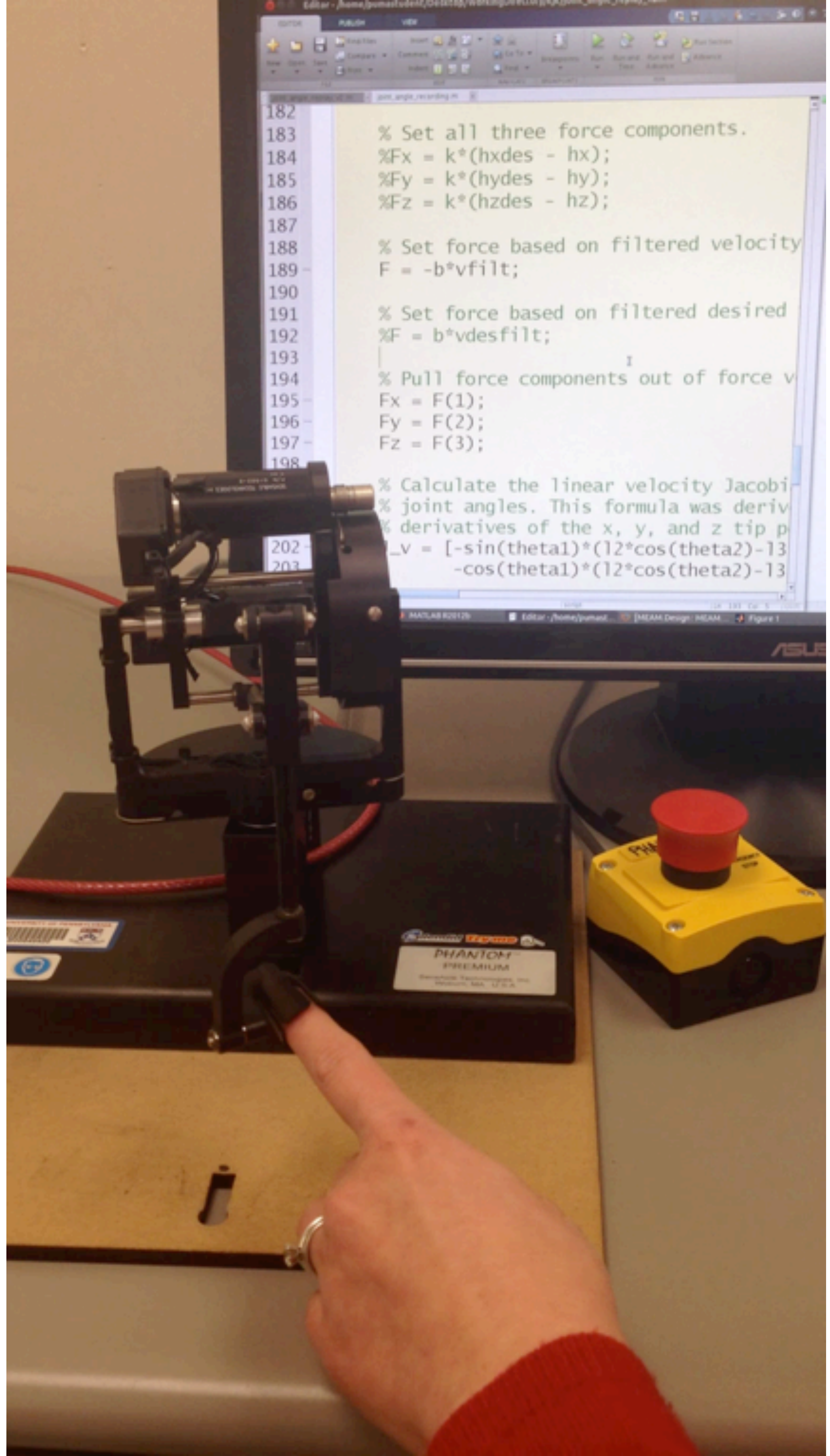


Does this look good?

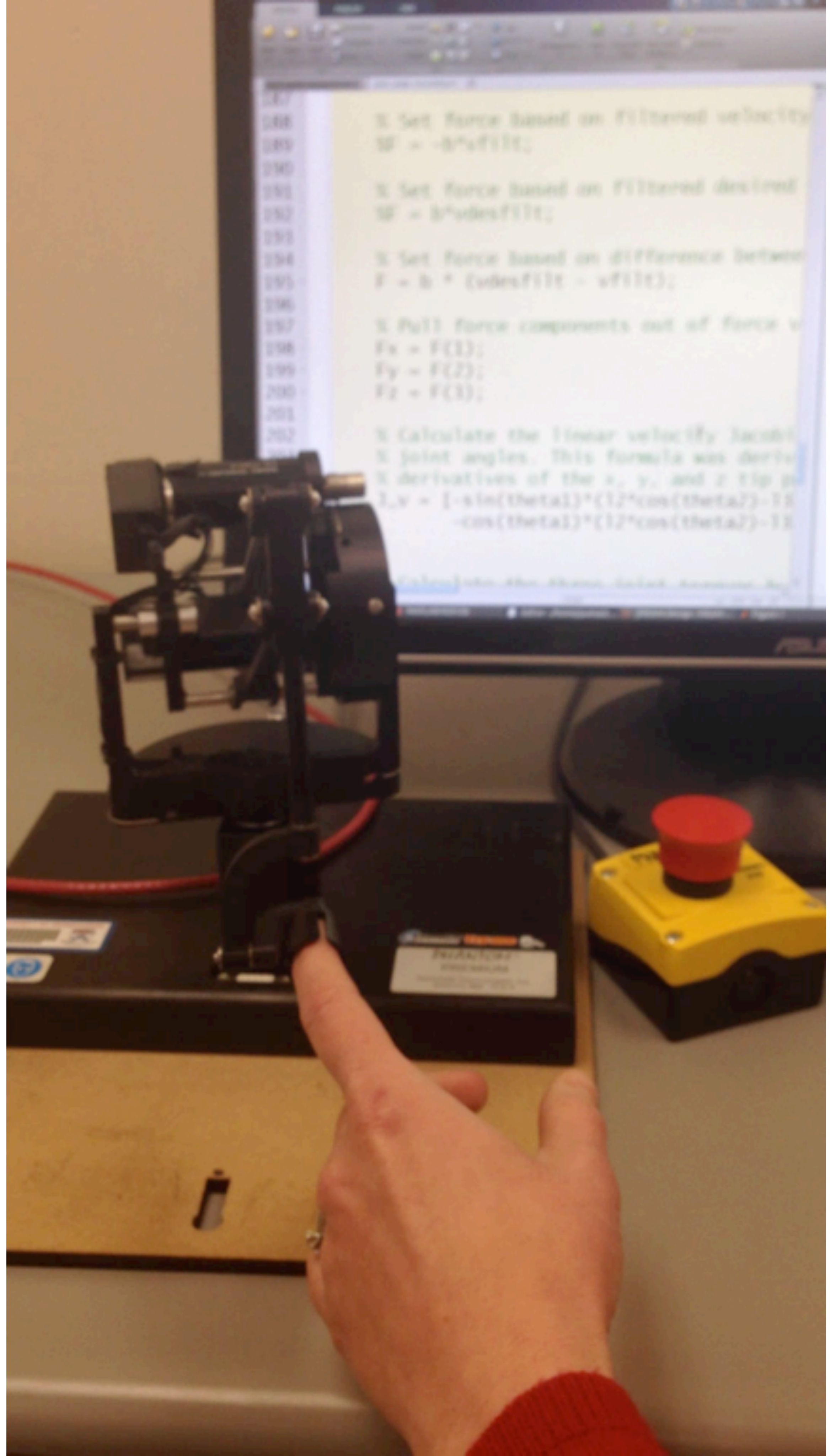


How can we improve the controller’s tracking?

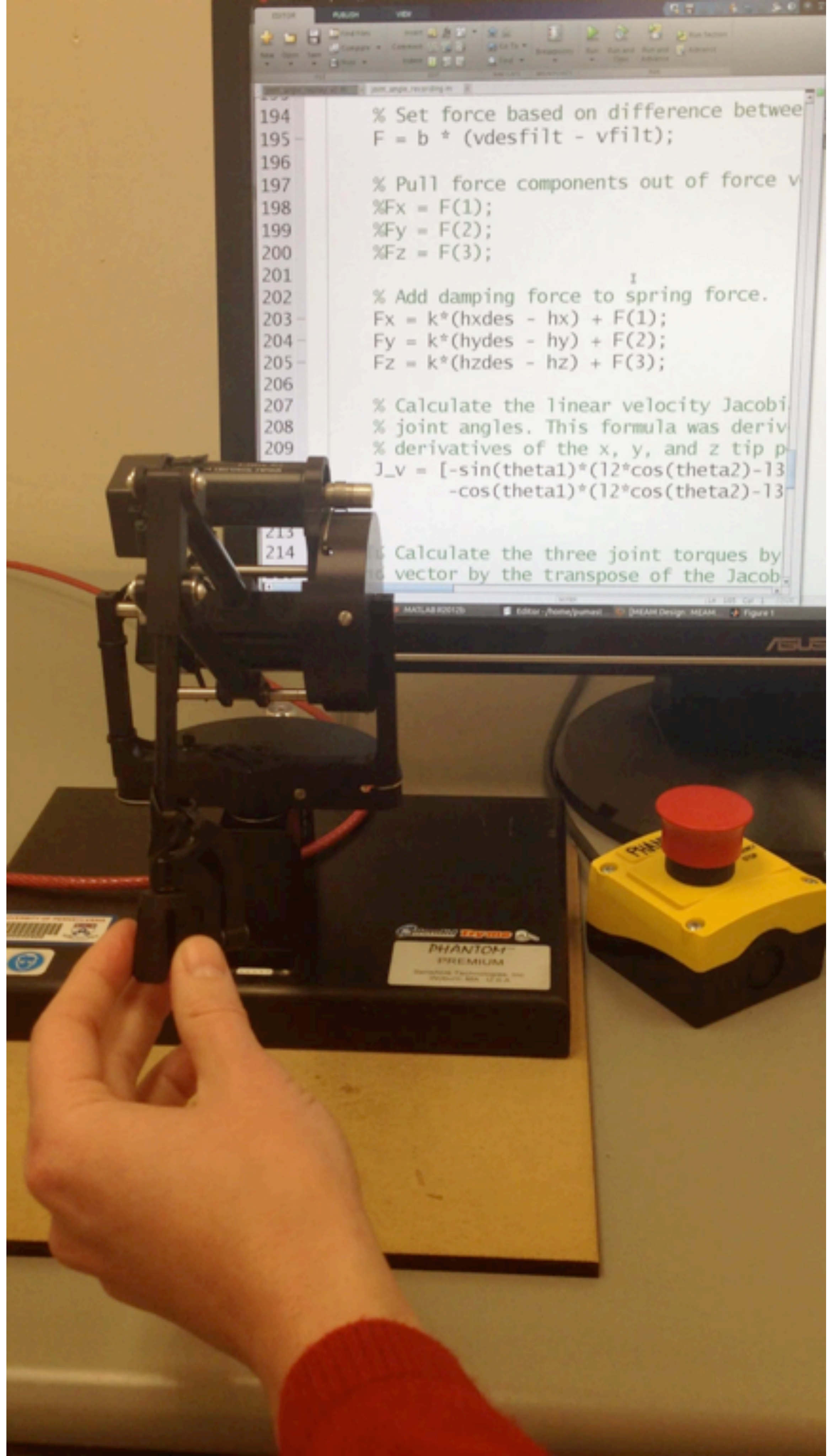
damping individual



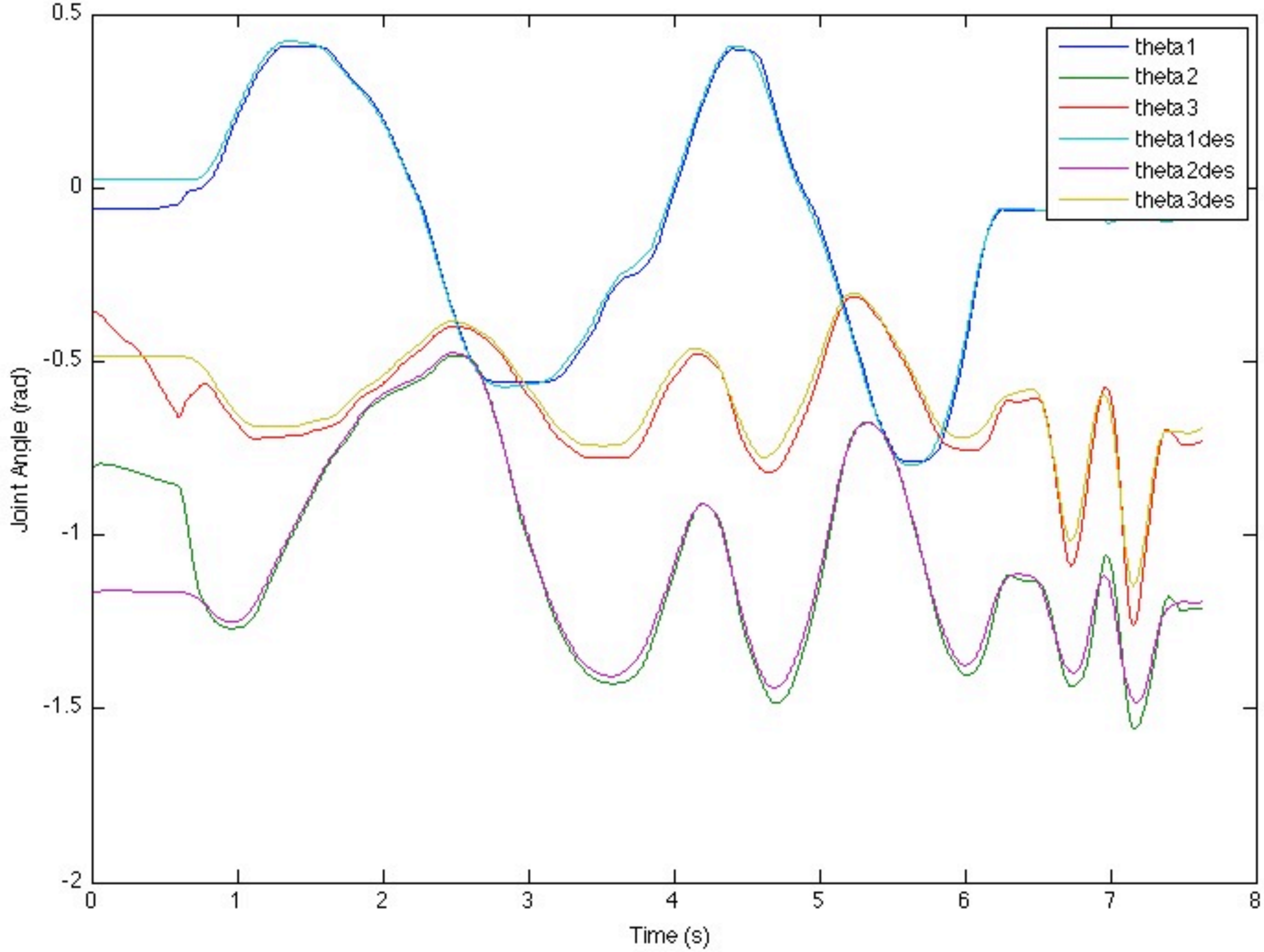
damping both (blurry)



replay loops final

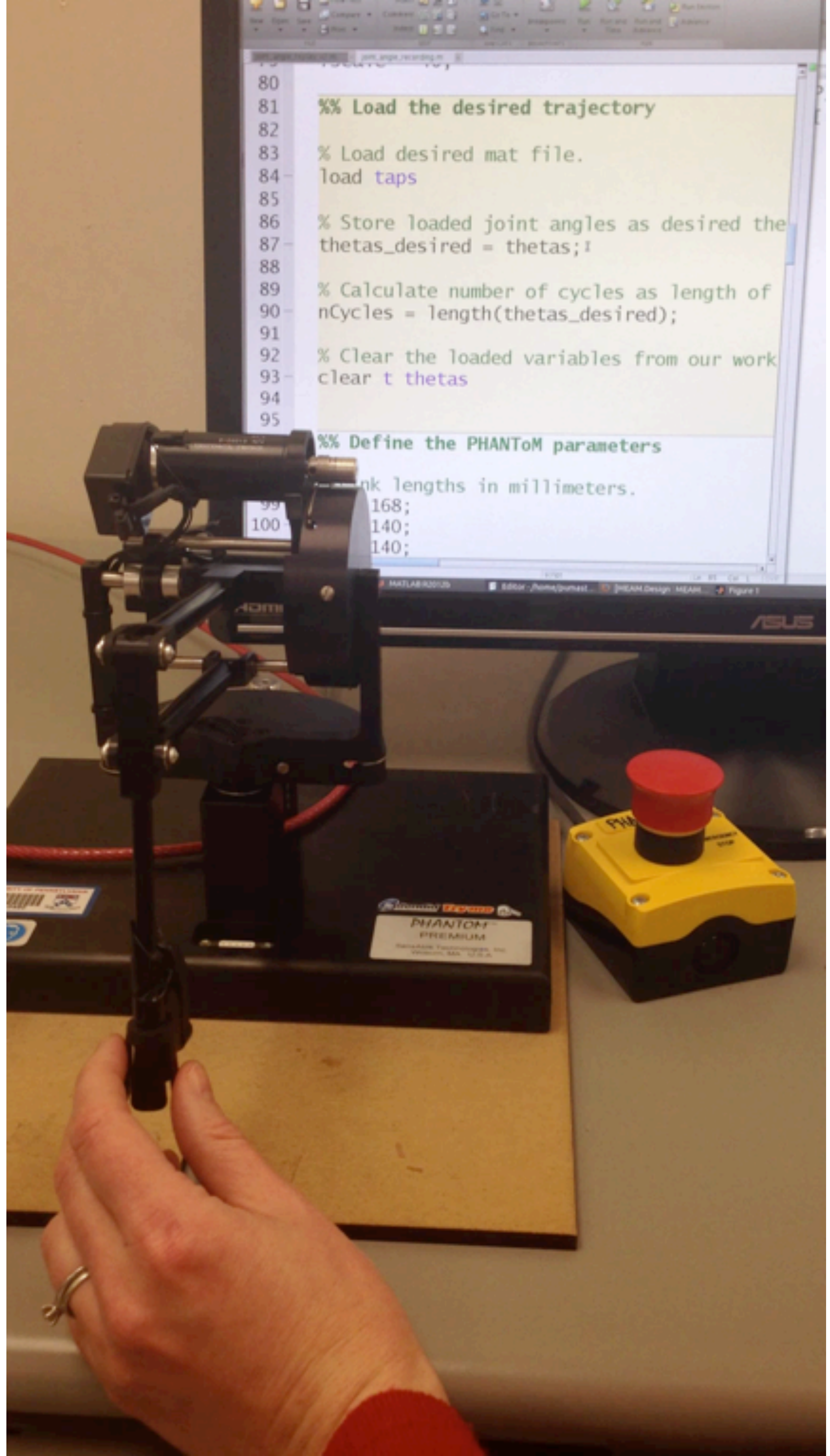


Very nice!



theta1
theta2
theta3
theta1des
theta2des
theta3des
Joint Angle (rad)
Time (s)
0.5
0
-0.5
-1
-1.5
-2
0
1
2
3
4
5
6
7
8

replay taps final



Is it perfect?

Questions ❓

Homework 4:
Velocity Kinematics and Jacobians

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

October 23, 2012

This assignment is due on **Friday, November 2 (updated)**, by 5:00 p.m. sharp. You should aim to turn the paper part in during class the day before. If you don't finish until later in the day, you can turn it in to Professor Kuchenbecker's office, Towne 224. Late submissions will be accepted until 5:00 p.m. on Monday, November 5, but they will be penalized by 25%. After that 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 a peer or a solution manual.

Written Problems *(60 points)*

This entire assignment is written and consists of two significantly adapted problems from the textbook, *Robot Modeling and Control* by Spong, Hutchinson, and Vidyasagar (SHV). Please follow the extra clarifications and instructions on both questions. Write in pencil, show your work clearly, $\boxed{\text{box your answers}}$, and staple your pages together.

1. **Adapted SHV 4-20, page 160 – Three-link Cylindrical Manipulator** ***(30 points)***
   The book works out the DH parameters and the transformation matrix T_3^0 for this robot on pages 85 and 86; you are welcome to use these results directly without rederiving them.

   (a) Use the position of the end-effector in the base frame to calculate the 3×3 linear velocity Jacobian J_v for the three-link cylindrical manipulator of Figure 3.7 on page 85.

   (b) Use the positions of the origins o_i and the orientations of the z-axes z_i to calculate the 3×3 linear velocity Jacobian J_v for the same robot. You should get the same answer as before.

   (c) Find the 3×3 angular velocity Jacobian J_ω for the same robot.

   (d) Find this robot's 6×3 Jacobian J.

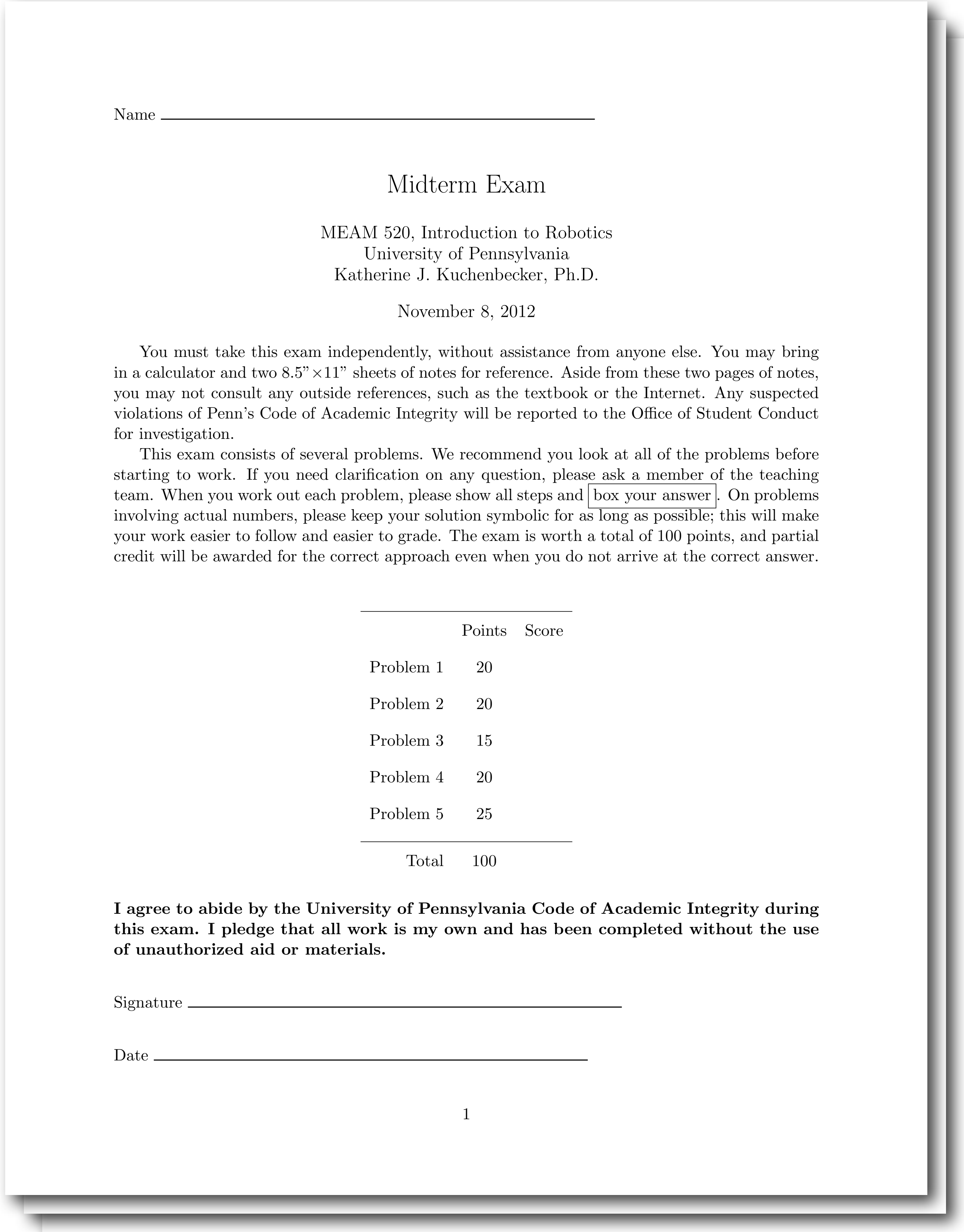
   (e) Imagine this robot is at $\theta_1 = \pi/2$ rad, $d_2 = 0.2$ m, and $d_3 = 0.3$ m, and its joint velocities are $\dot{\theta}_1 = 0.1$ rad/s, $\dot{d}_2 = 0.25$ m/s, and $d_3 = -0.05$ m/s. What is v_3^0, the linear velocity vector of the end-effector with respect to the base frame, expressed in the base frame? Make sure to provide units with your answer.

   (f) For the same situation, what is ω_3^0, the angular velocity vector of the end-effector with respect to the base frame, expressed in the base frame? Make sure to provide units with your answer.

   (g) Use your answers from above to derive the singular configurations of the arm, if any. Here we are concerned with the linear velocity of the end-effector, not its angular velocity. Be persistent with the calculations; they should reduce to something nice.

   (h) Sketch the cylindrical manipulator in each singular configuration that you found, and explain what effect the singularity has on the robot's motion in that configuration.

Graded Homework 4 available.



Name ________________________________

Midterm Exam

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

November 8, 2012

You must take this exam independently, without assistance from anyone else. You may bring in a calculator and two 8.5"×11" sheets of notes for reference. Aside from these two pages of notes, you may not consult any outside references, such as the textbook or the Internet. Any suspected violations of Penn's Code of Academic Integrity will be reported to the Office of Student Conduct for investigation.

This exam consists of several problems. We recommend you look at all of the problems before starting to work. If you need clarification on any question, please ask a member of the teaching team. When you work out each problem, please show all steps and box your answer. On problems involving actual numbers, please keep your solution symbolic for as long as possible; this will make your work easier to follow and easier to grade. The exam is worth a total of 100 points, and partial credit will be awarded for the correct approach even when you do not arrive at the correct answer.

	Points	Score
Problem 1	20	
Problem 2	20	
Problem 3	15	
Problem 4	20	
Problem 5	25	
Total	100	

I agree to abide by the University of Pennsylvania Code of Academic Integrity during this exam. I pledge that all work is my own and has been completed without the use of unauthorized aid or materials.

Signature ________________________________

Date ________________________________

Regrades available.