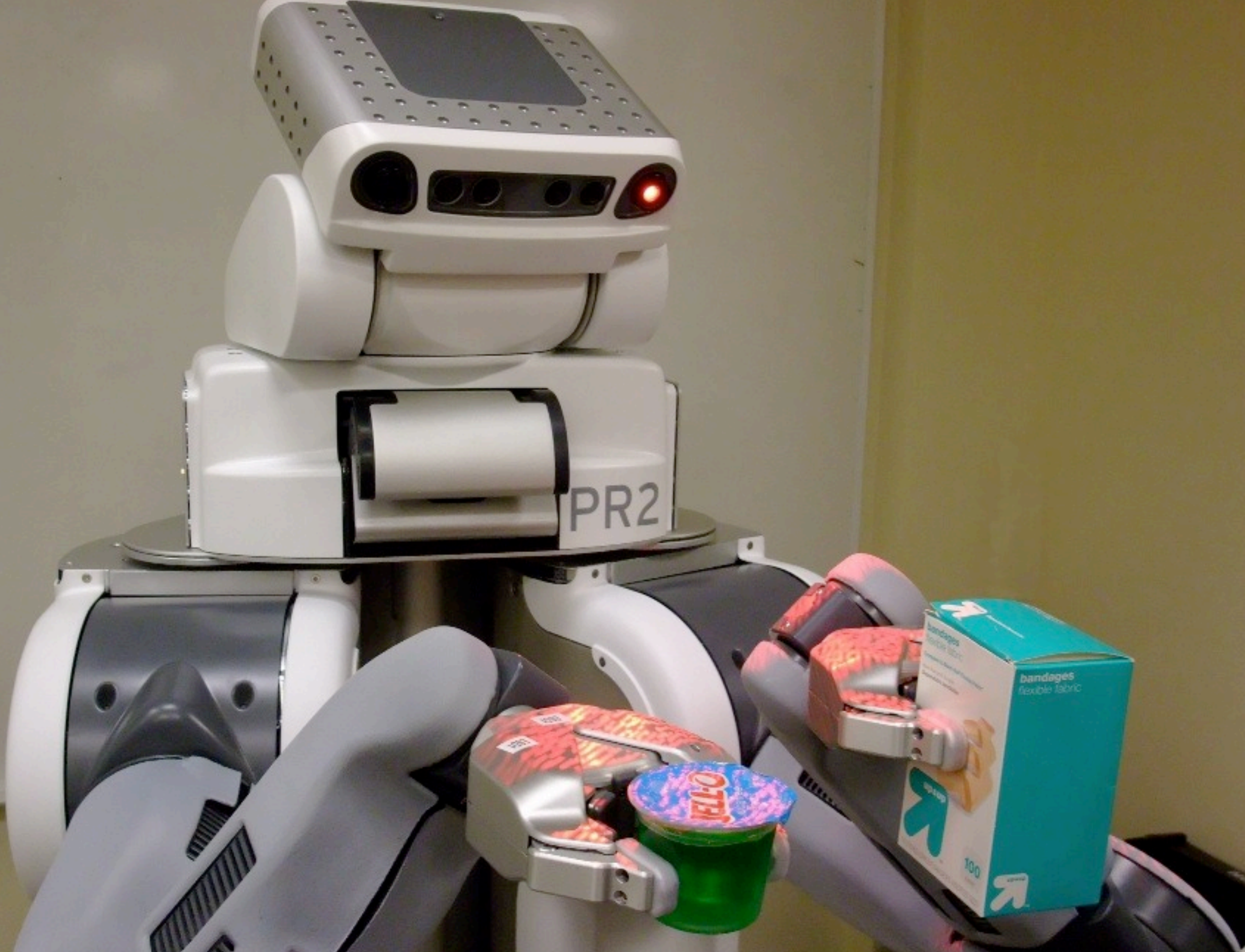
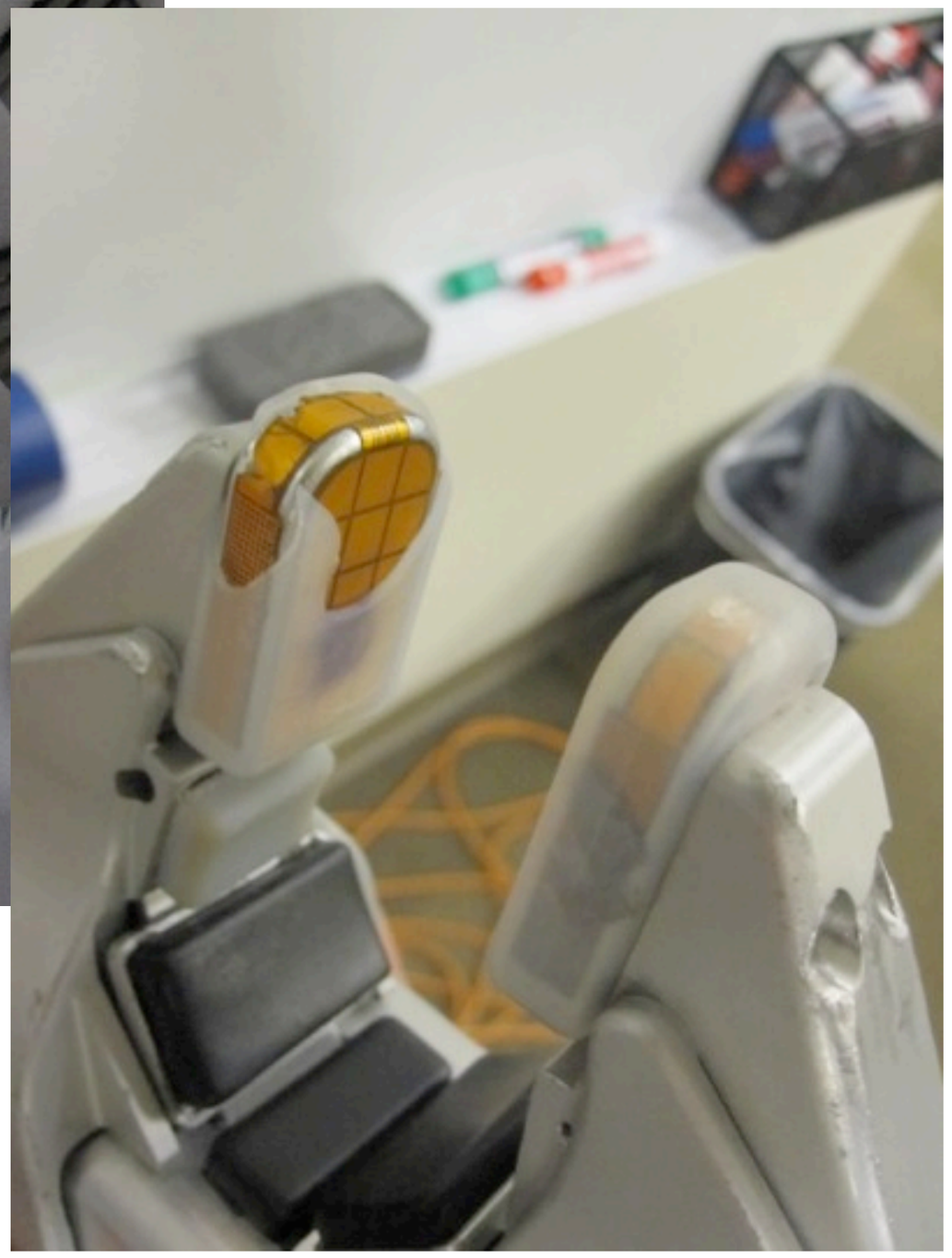
MEAM 520
Robot Dynamics

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

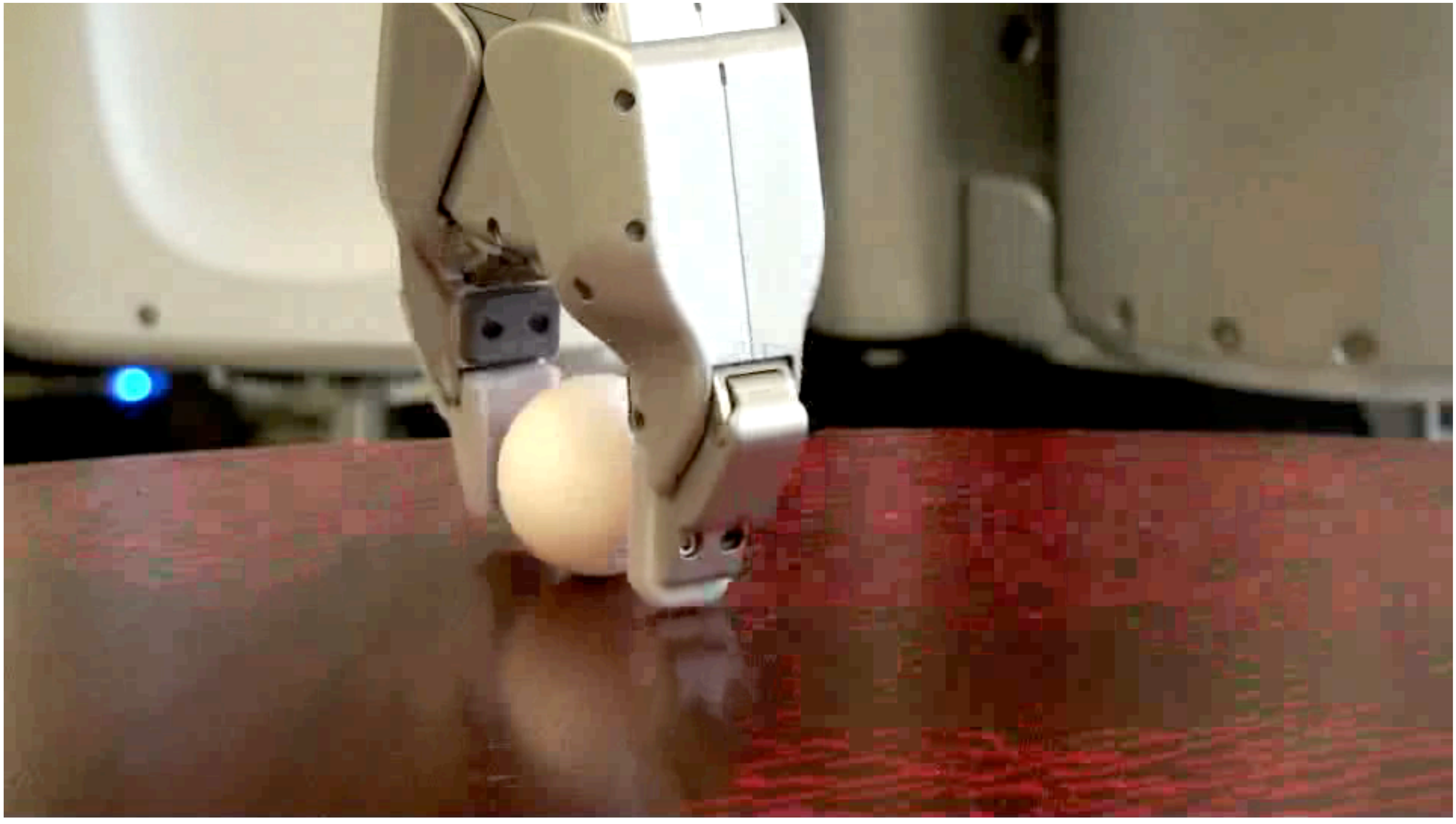
Lecture 20: November 29, 2012


PR2
JELL-O
bandages
flexible fabric
100


Peppermint
SARAN
PREMIUM
WRAP
HEAVY DUTY · DURABLE
MICROWAVE SAFE
100 SQ FT
Minute Maid
100% JUICE
Silk
ORIGINAL
CRYSTAL GEYSER
365 ORGANIC
Organic
Honey & Nut
Morning O's
Scotch
JELL-O




Peppermint
SARAN
PREMIUM
WRAP
HEAVY DUTY · DURABLE
MICROWAVE SAFE
100 SQ FT
Minute Maid
100% JUICE
Silk
ORIGINAL
CRYSTAL GEYSER
365 ORGANIC
Organic
Honey & Nut
Morning O's
Scotch
JELL-O

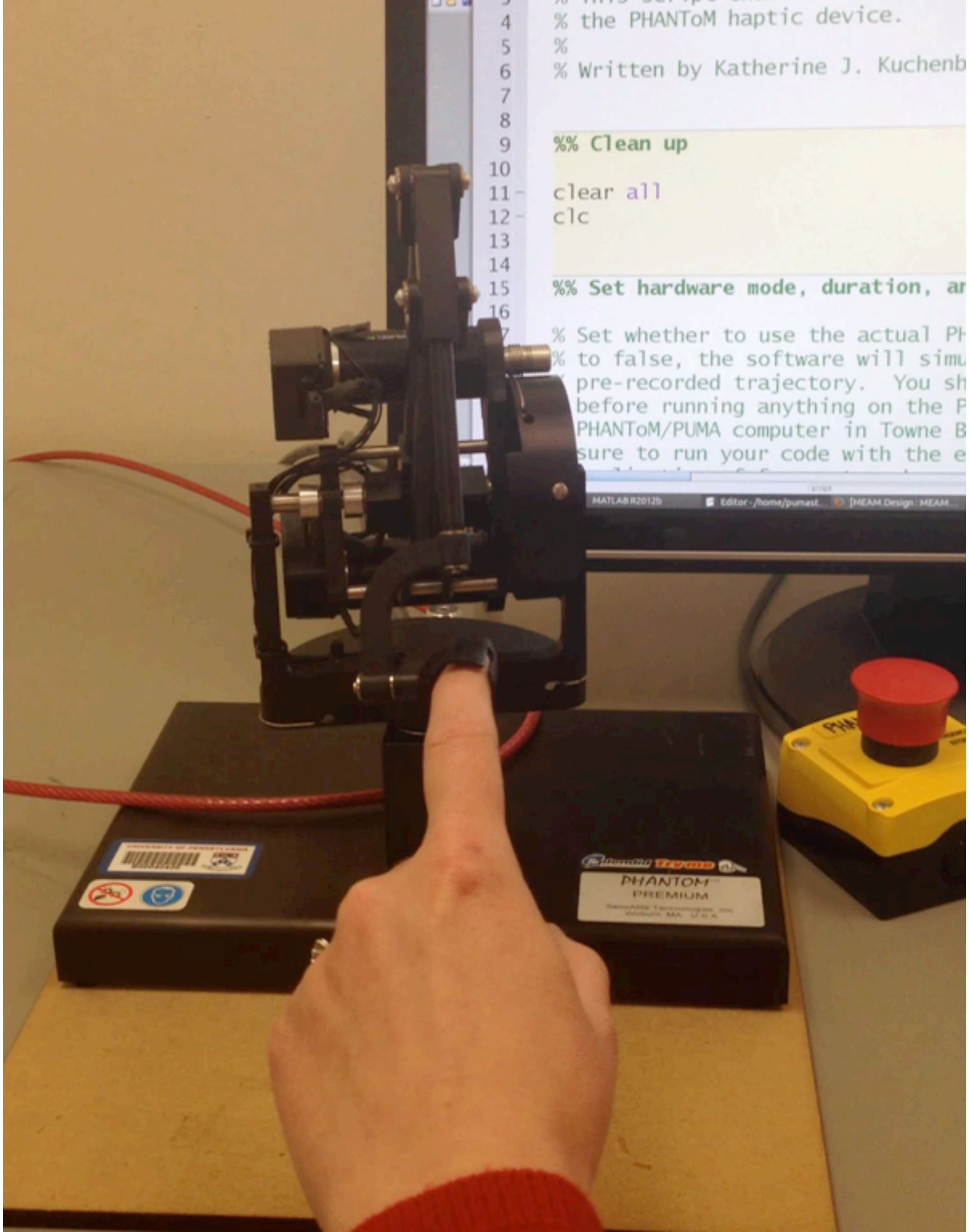



PR-M
PR2
rosrun pr2_props high_five

Last time...

(SHV 6.3)

record loops



Desired Position

$$\vec{x}_{h,\mathrm{des}} = \Lambda(\theta_{1,\mathrm{des}}, \theta_{2,\mathrm{des}}, \theta_{3,\mathrm{des}})$$

$$x_{h,\mathrm{des}}, y_{h,\mathrm{des}}, z_{h,\mathrm{des}}$$

Actual Position

$$\vec{x}_h = \Lambda(\theta_1, \theta_2, \theta_3)$$

$$x_h, y_h, z_h$$

Proportional Feedback Controller

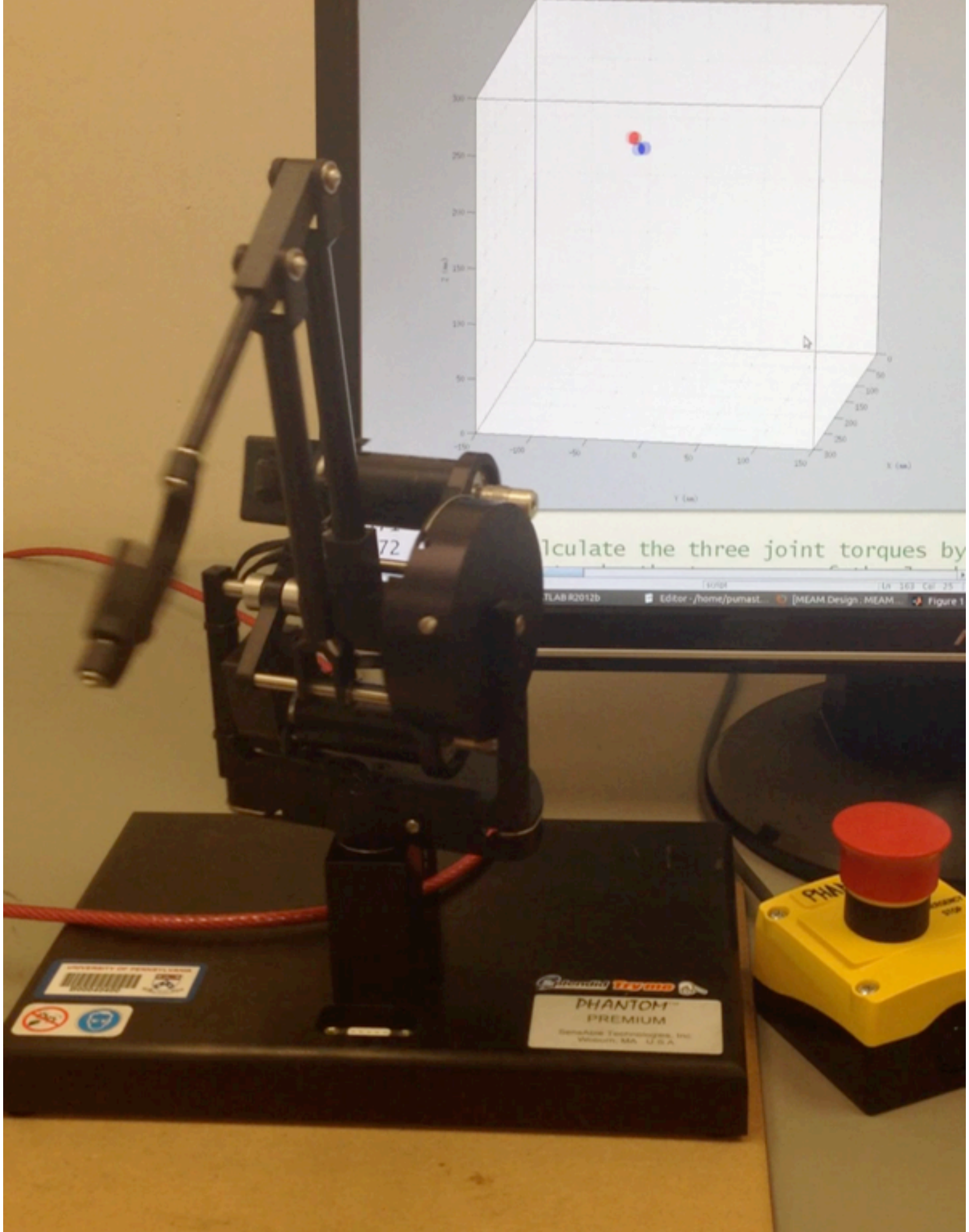
$$\vec{F} = k(\vec{x}_{h,\mathrm{des}} - \vec{x}_h)$$

$$F_x = k(x_{h,\mathrm{des}} - x_h)$$

$$F_y = k(y_{h,\mathrm{des}} - y_h)$$

$$F_z = k(z_{h,\mathrm{des}} - z_h)$$

replay loops:
spring force,
fixed kinematics



Mass on a spring: simple harmonic oscillator

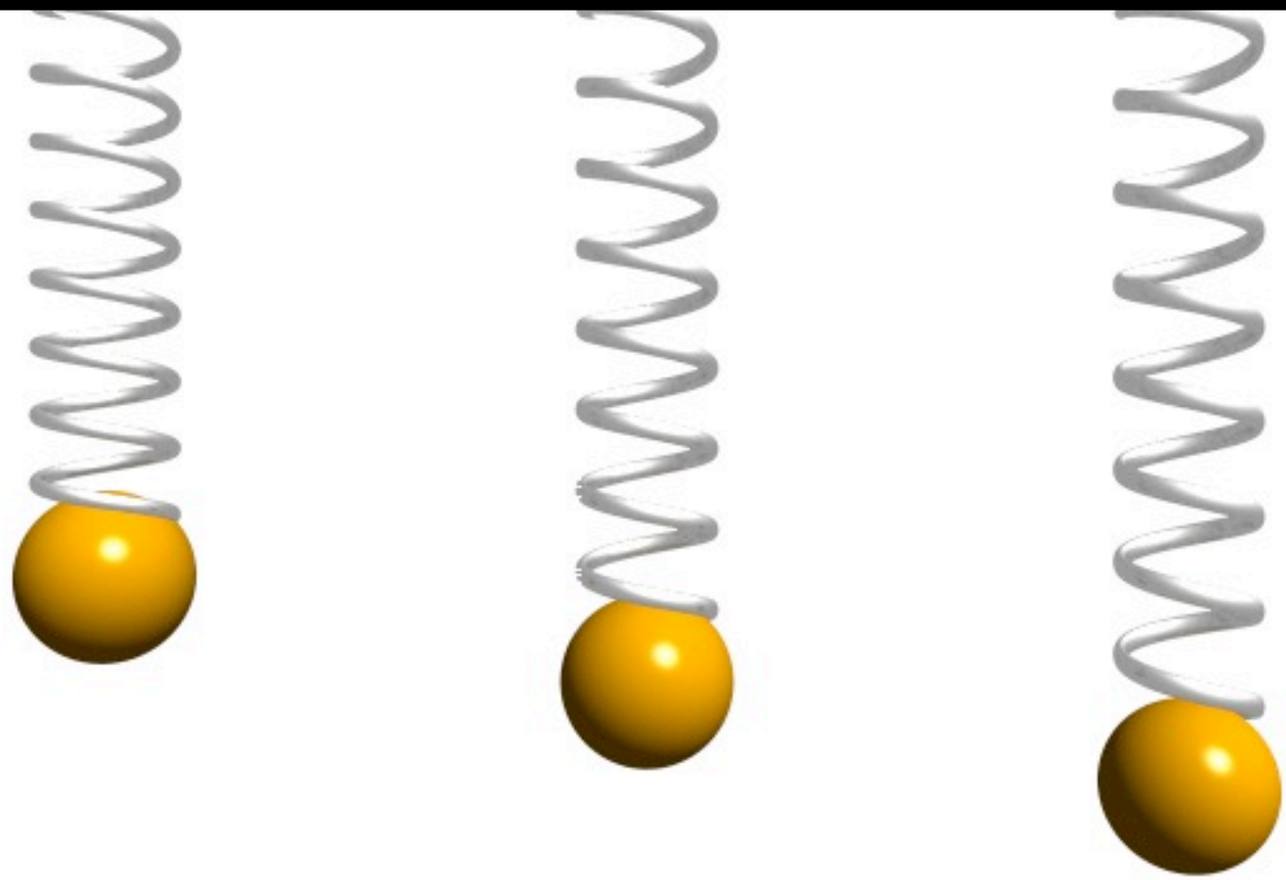


Image from http://lifshitz.ucdavis.edu/~dmartin/phy7/7C.html

Editor - /Users/kuchenbe/Documents/teaching/meam 520/lectures/20 gravity/simulate_mass_on_a_spring_ode45.m

File Edit Text Go Cell Tools Debug Desktop Window Help

```
%% Simulate a mass bouncing on a spring using ode45
% Class example for MEAM 520 on November 29, 2012 by KJK.

clear;

%% Parameters
% Set parameters of the system we want to simulate, noting units.
% Make them global so that the compute derivatives function can see t
global m k b g
m = 3.5; % kg
k = 400; % N/m
b = 10;    % Ns/m
g = 9.81; % m/s^2

%% Time Vector
% Create a time vector.  The ' makes it a column vector.
tstart = 0;
tfinal = 3;
tstepmax = 0.01; % Maximum time step, in seconds.

graphical_tstep = 0.01; % s

%% Initial Conditions
% Define the initial conditions for the mass.
y0 = .2; % m
v0 = 0; % m/s

% Put initial conditions into vector.
X0 = [y0; v0];
```

simulate_mass_on_a_spring_ode45.m | compute_mass_derivatives.m

script | Ln 17 Col 12

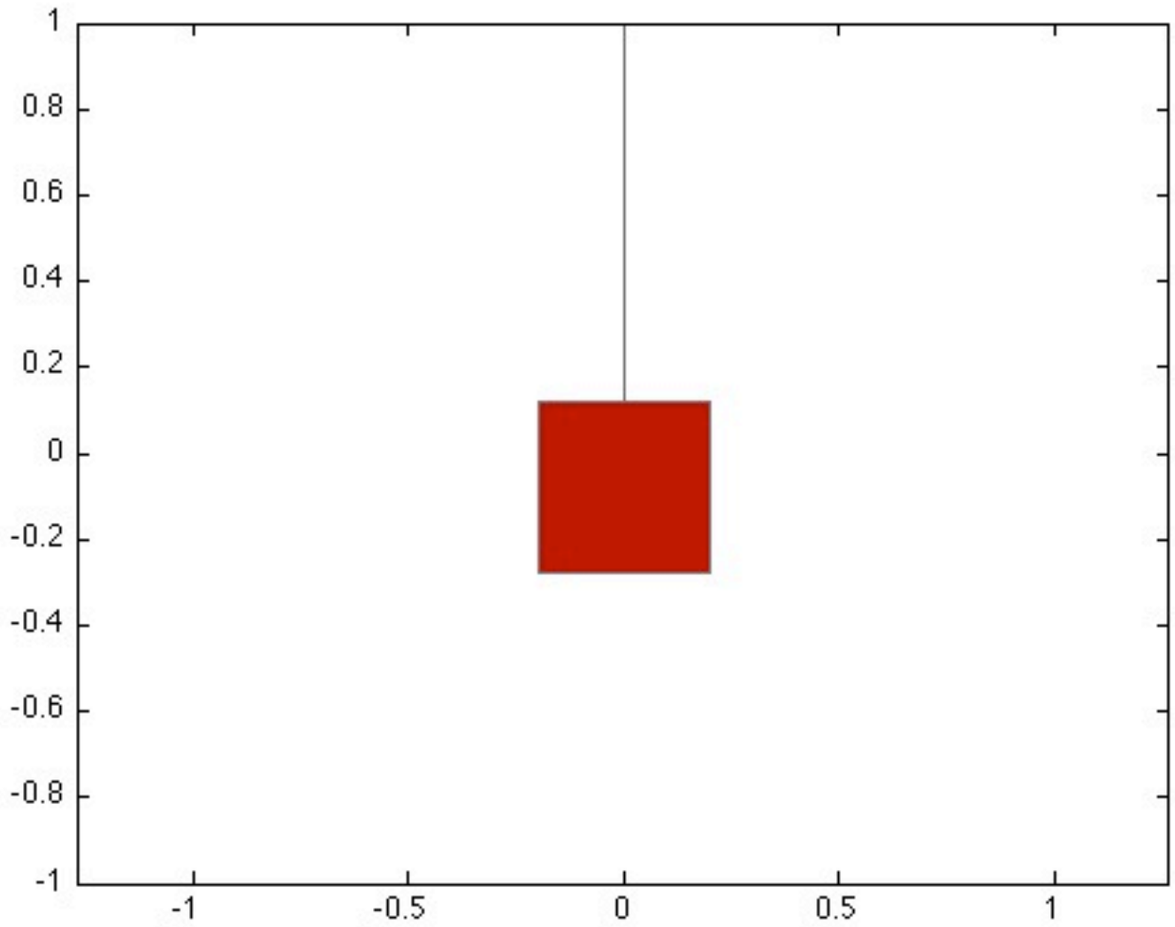
Editor - /Users/kuchenbe/Documents/teaching/meam 520/lectures/20 gravity/simulate_mass_on_a_spring_ode45.m

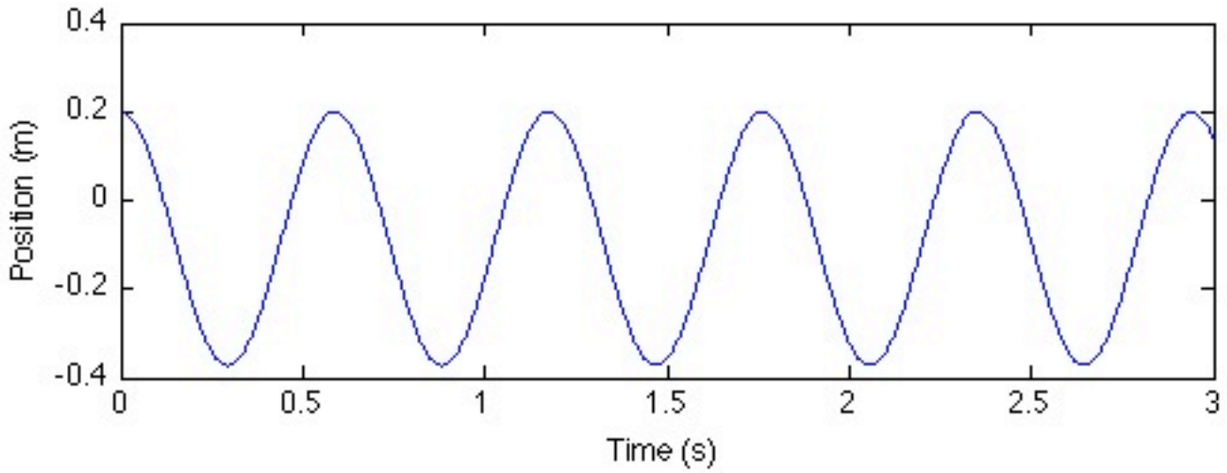
File Edit Text Go Cell Tools Debug Desktop Window Help

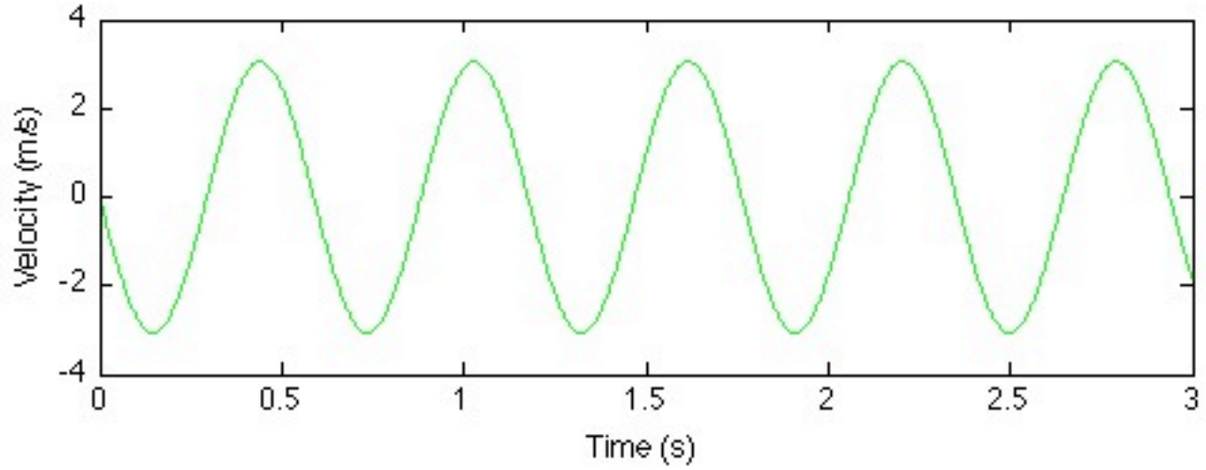
```
22
23      %% Initial Conditions
24      % Define the initial conditions for the mass.
25 -    y0 = .2; % m
26 -    v0 = 0; % m/s
27
28      % Put initial conditions into vector.
29 -    X0 = [y0; v0];
30
31      %% Simulation
32      % Show a message to explain how to cancel the graph.
33 -    disp('Click in this window and press control-c to cancel simulation c
34
35      % Run the simulation using ode45.
36      % The state equation function must be in the same directory as this s
37      % for Matlab to find it.  The @ makes the name a function handle, so
38      % can call it over and over.  The other two inputs are the time span
39      % the initial conditions.  The outputs are the resulting time vector
40      % and the resulting state vector (nx4).
41      % Here, we set the maximum time step to be tstep, to lower the likeli
42      % that the solver will accidentally miss interactions with the intere
43      % zones in the world.
44 -    options = odeset('MaxStep',tstepmax);
45 -    [t, Xhistory] = ode45(@compute_mass_derivatives, [tstart tfinal], X0,
46
47      %% Plot set up
48
49      % The simulation results are not evenly spaced in time, so graphing t
50      % directly does not let you see the speed of the puck.  Thus, we
51      % re-interpolate the data to see where the puck is at evenly spaced n
```

simulate_mass_on_a_spring_ode45.m | compute_mass_derivatives.m

script | Ln 34 Col 1


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1


0.4
0.2
0
-0.2
-0.4
Position (m)
0
0.5
1
1.5
2
2.5
3
Time (s)


4
2
0
-2
-4
Velocity (m/s)
0
0.5
1
1.5
2
2.5
3
Time (s)

$$\Sigma F_y = m\ddot{y}$$

$$-mg - ky - b\dot{y} = m\ddot{y}$$

$$-mg = m\ddot{y} + b\dot{y} + ky$$

$$-g = \ddot{y} + \frac{b}{m}\dot{y} + \frac{k}{m}y$$

Second-order system

$$\frac{k}{m} = \omega_n^2$$

$$-g = \ddot{y} + 2\zeta\omega_n\dot{y} + \omega_n^2 y$$

$$\frac{b}{m} = 2\zeta\omega_n$$

$$k_{\text{controller}} = m\omega_{n,\text{desired}}^2$$

$$b_{\text{controller}} = 2m\zeta_{\text{desired}}\omega_n - b_{\text{robot}}$$

$$\zeta_{\text{desired}} = 1$$

$$\vec{F} = k(\vec{x}_{h,\mathrm{des}} - \vec{x}_h) + b(\vec{v}_{h,\mathrm{des}} - \vec{v}_h)$$

$$F_x = k(x_{h,\mathrm{des}} - x_h) + b(\dot{x}_{h,\mathrm{des}} - \dot{x}_h)$$

$$F_y = k(y_{h,\mathrm{des}} - y_h) + b(\dot{y}_{h,\mathrm{des}} - \dot{y}_h)$$

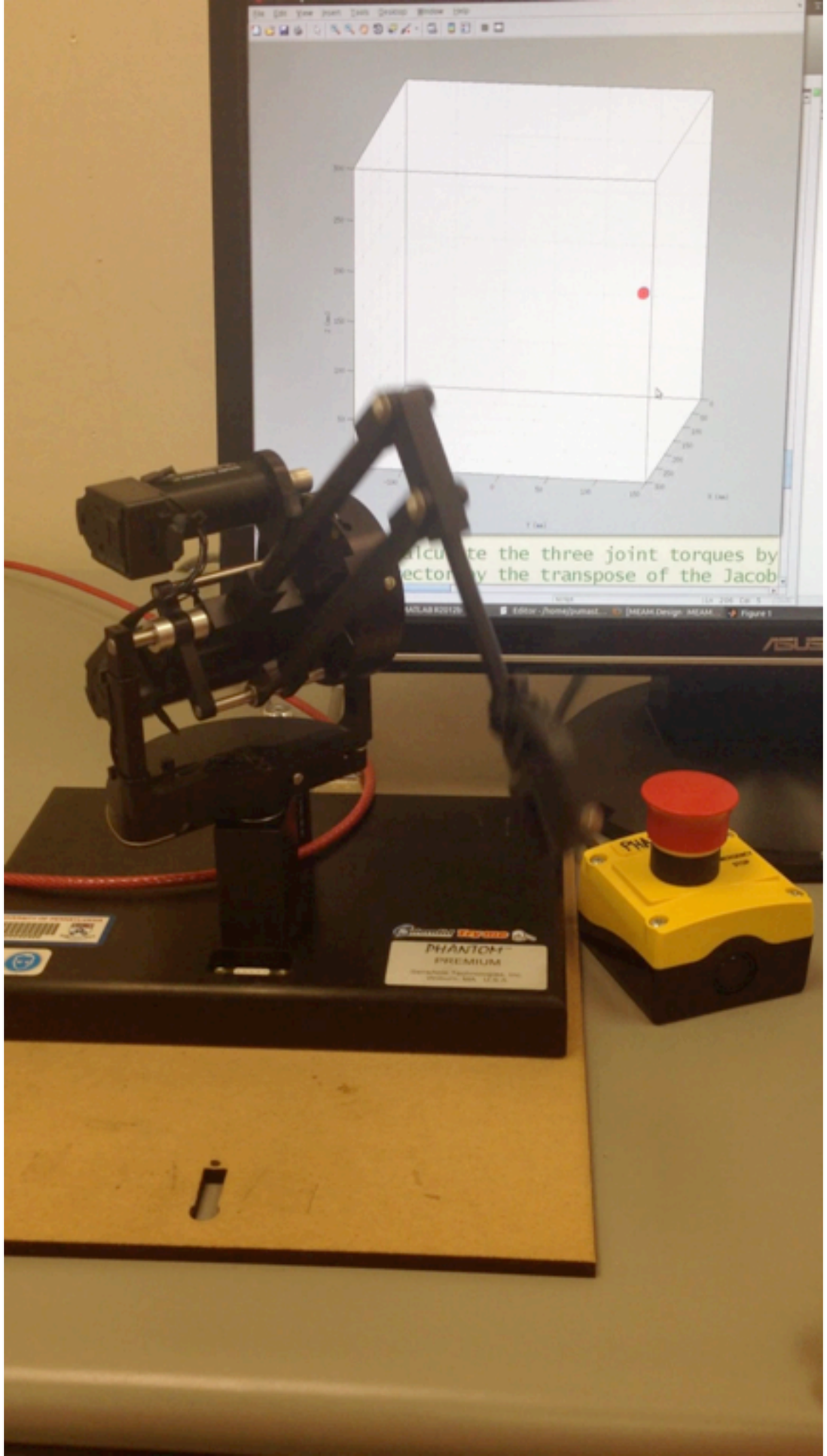
$$F_z = k(z_{h,\mathrm{des}} - z_h) + b(\dot{z}_{h,\mathrm{des}} - \dot{z}_h)$$

$$\vec{e}_h = \vec{x}_{h,\mathrm{des}} - \vec{x}_h$$

$$\dot{\vec{e}}_h = \vec{v}_{h,\mathrm{des}} - \vec{v}_h$$

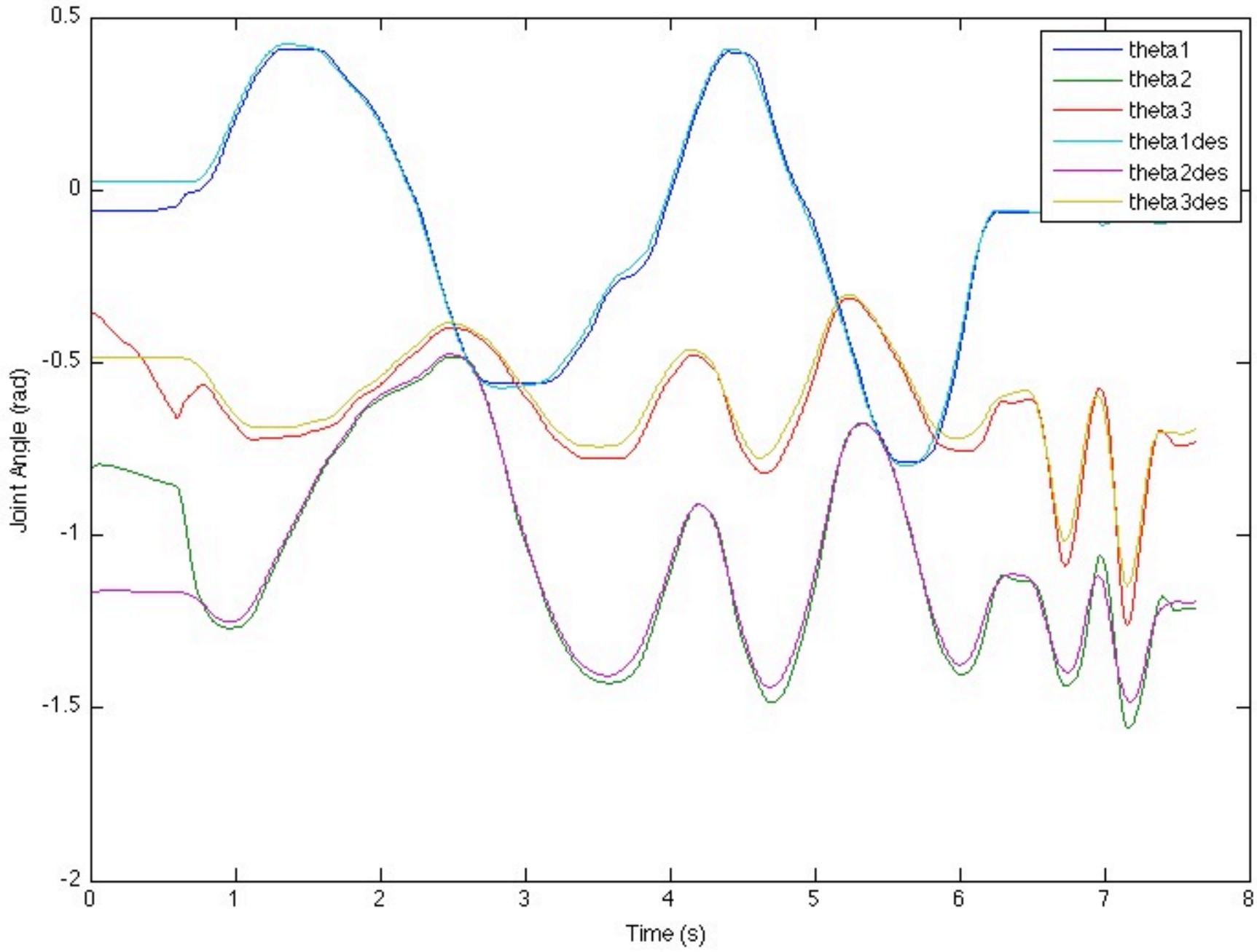
$$\vec{F} = k\,\vec{e}_h + b\,\dot{\vec{e}}_h$$

replay loops final



Questions ?

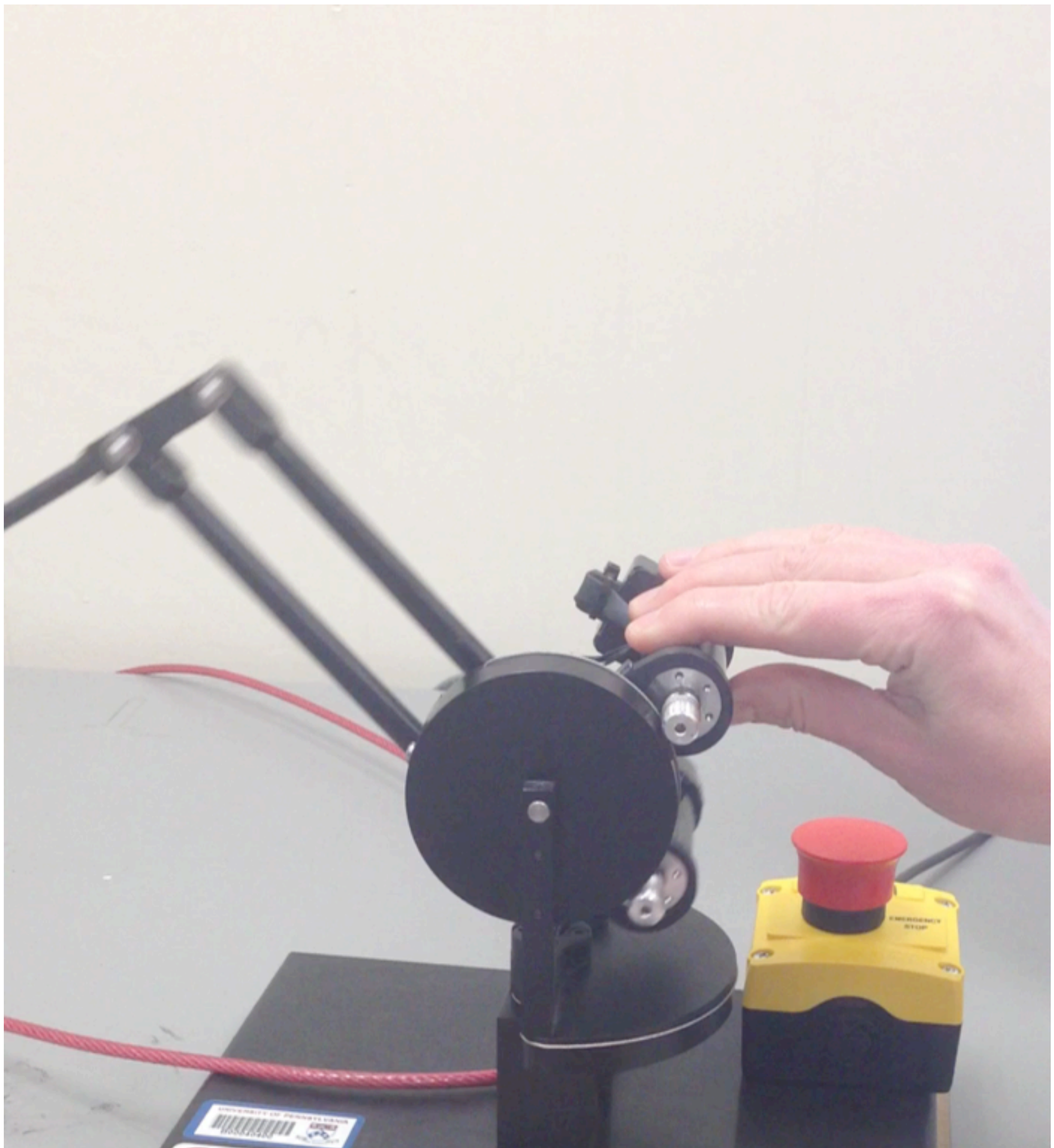
Very nice!



Is it perfect?

How can we improve the controller's tracking?

Add gravity compensation!


EMERGENCY
STOP

How should I compensate for gravity?

mechanically (add weight near the tip)
mechanically (springs)
in software (use the motors)

Which option is better?

Both approaches to gravity compensation are useful.

For haptics, a small amount of software gravity compensation is useful to avoid having to increase the inertia of the system.

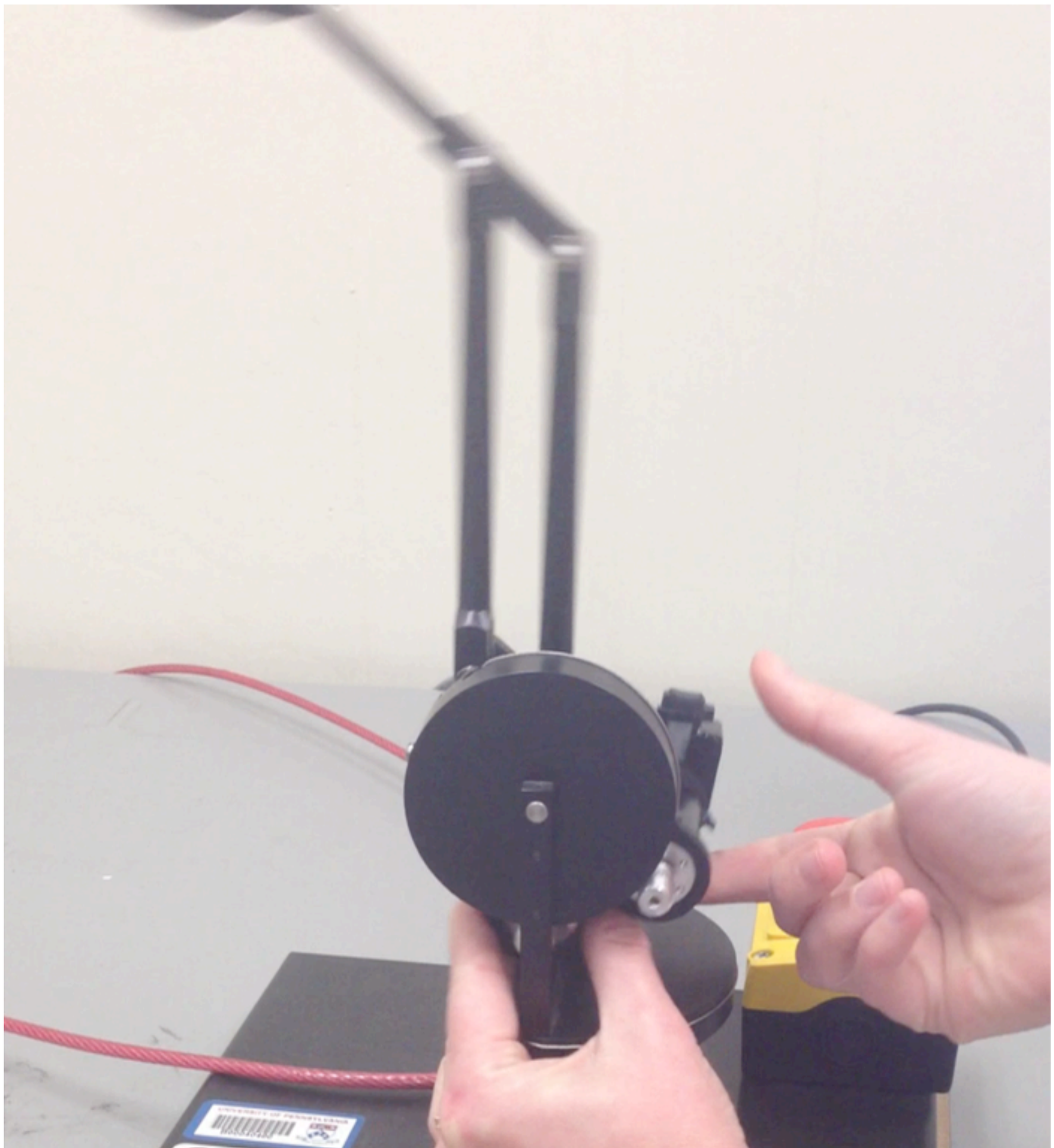
For Phantom, first focus on joint 3, because its inherent gravity balance is worse than joint 2.

Gravity compensation is a form of feedforward control
(SHV 6.4)

How do we calibrate gravity compensation?

Move the robot slowly through a trajectory and record the torque needed to hold up the weight of the robot.

What motion should we use?



$\theta_3 = 0$

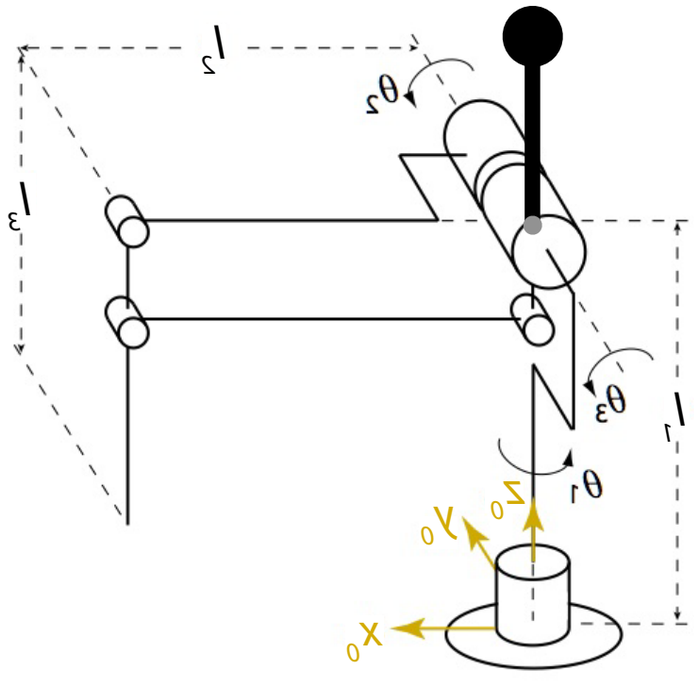
m

g

$\theta_3 = -1.3$

$\theta_3 = -\pi/2 = -1.57$

$\theta_3 = -2.6$



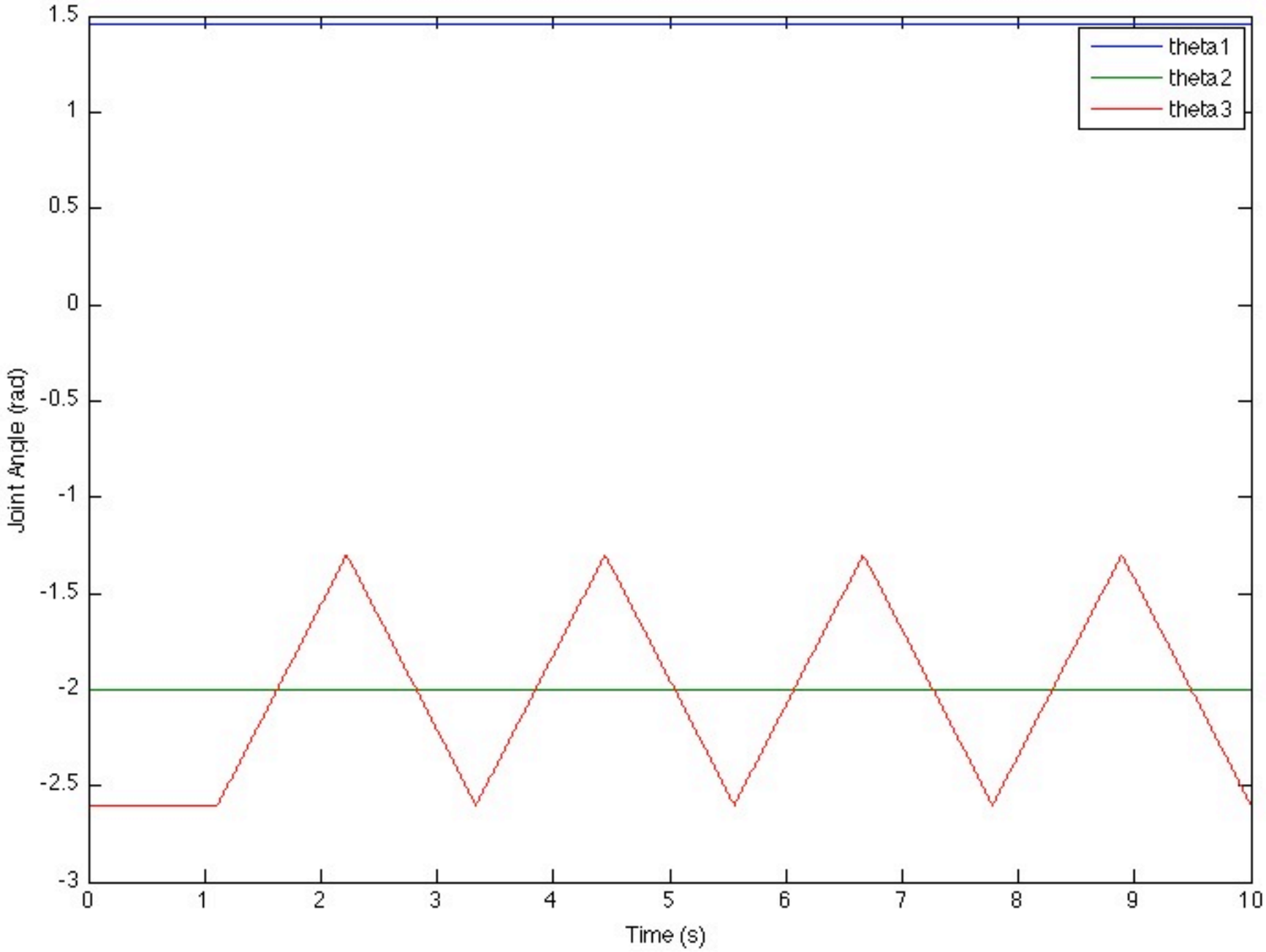
Editor - /Users/kuchenbe/Desktop/kjk/make_theta3_v1.m

File Edit Text Go Cell Tools Debug Desktop Window Help

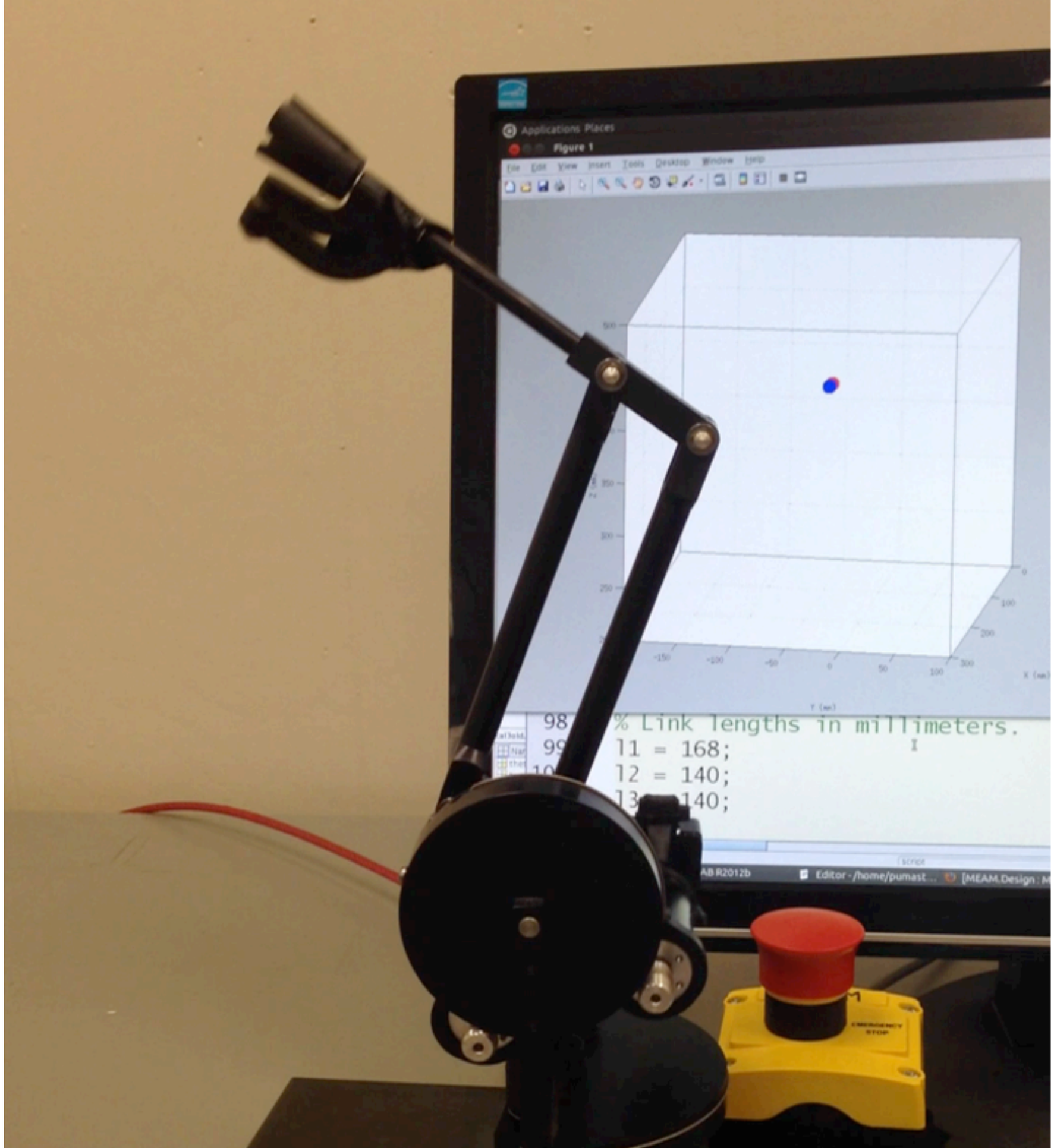
```
% Make t.
t = linspace(0,10,9000)';

% Make all thetas.
thetas = zeros(9000,3);
thetas(:,1) = 1.46;
thetas(:,2) = -2;
thetas(:,3) = -2.6;
thetas(1001:2000,3) = linspace(-2.6,-1.3,1000)';
thetas(2001:3000,3) = linspace(-1.3,-2.6,1000)';
thetas(3001:4000,3) = linspace(-2.6,-1.3,1000)';
thetas(4001:5000,3) = linspace(-1.3,-2.6,1000)';
thetas(5001:6000,3) = linspace(-2.6,-1.3,1000)';
thetas(6001:7000,3) = linspace(-1.3,-2.6,1000)';
thetas(7001:8000,3) = linspace(-2.6,-1.3,1000)';
thetas(8001:9000,3) = linspace(-1.3,-2.6,1000)';

% Plot thetas.
figure(6);
plot(t,thetas)
```

ox_demo.m | haptic_ball_team50.m | haptic_damping_team50.m | cmdhist.m | make_theta3_v1.m

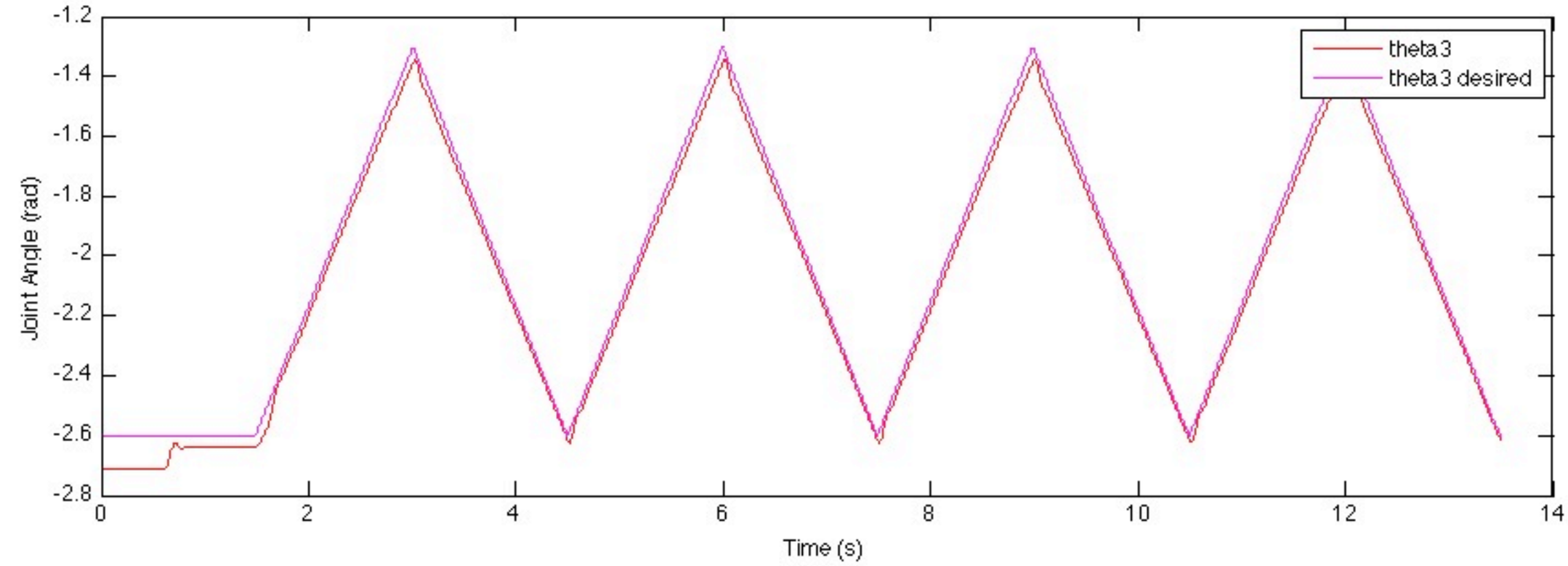
script Ln 17 Col 1

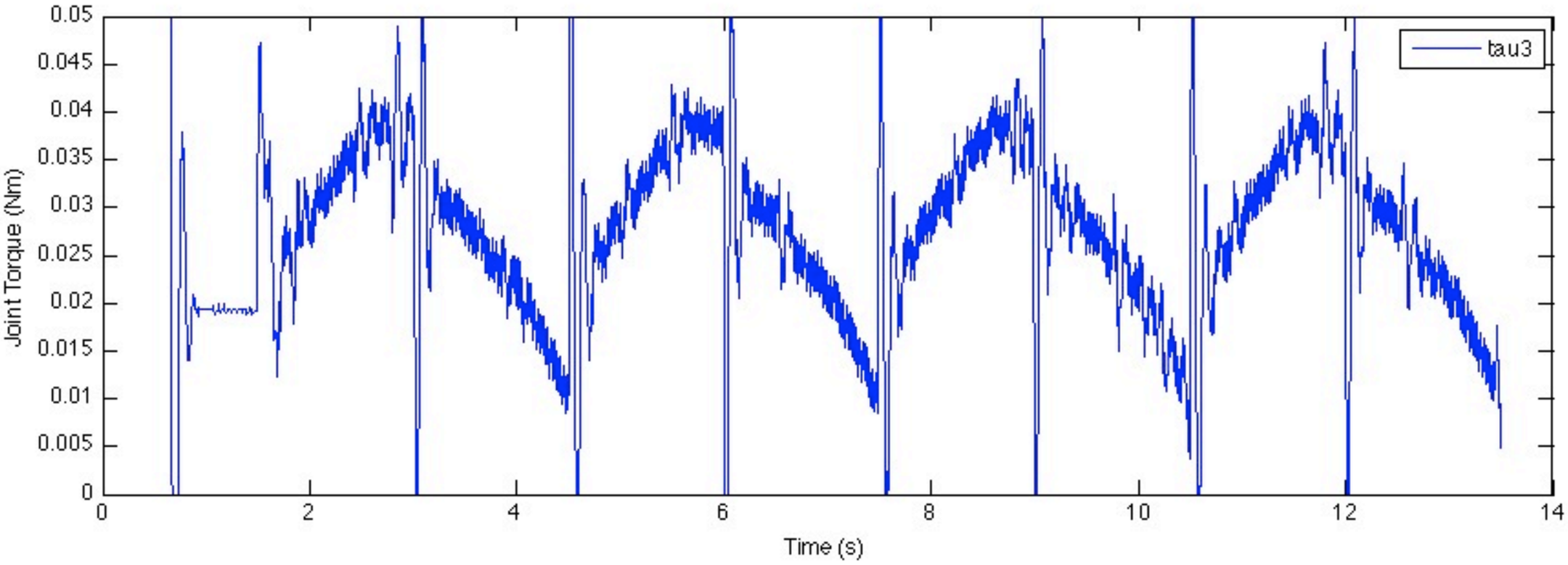

theta1
theta2
theta3
Joint Angle (rad)
Time (s)
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
0
1
2
3
4
5
6
7
8
9
10

cal3

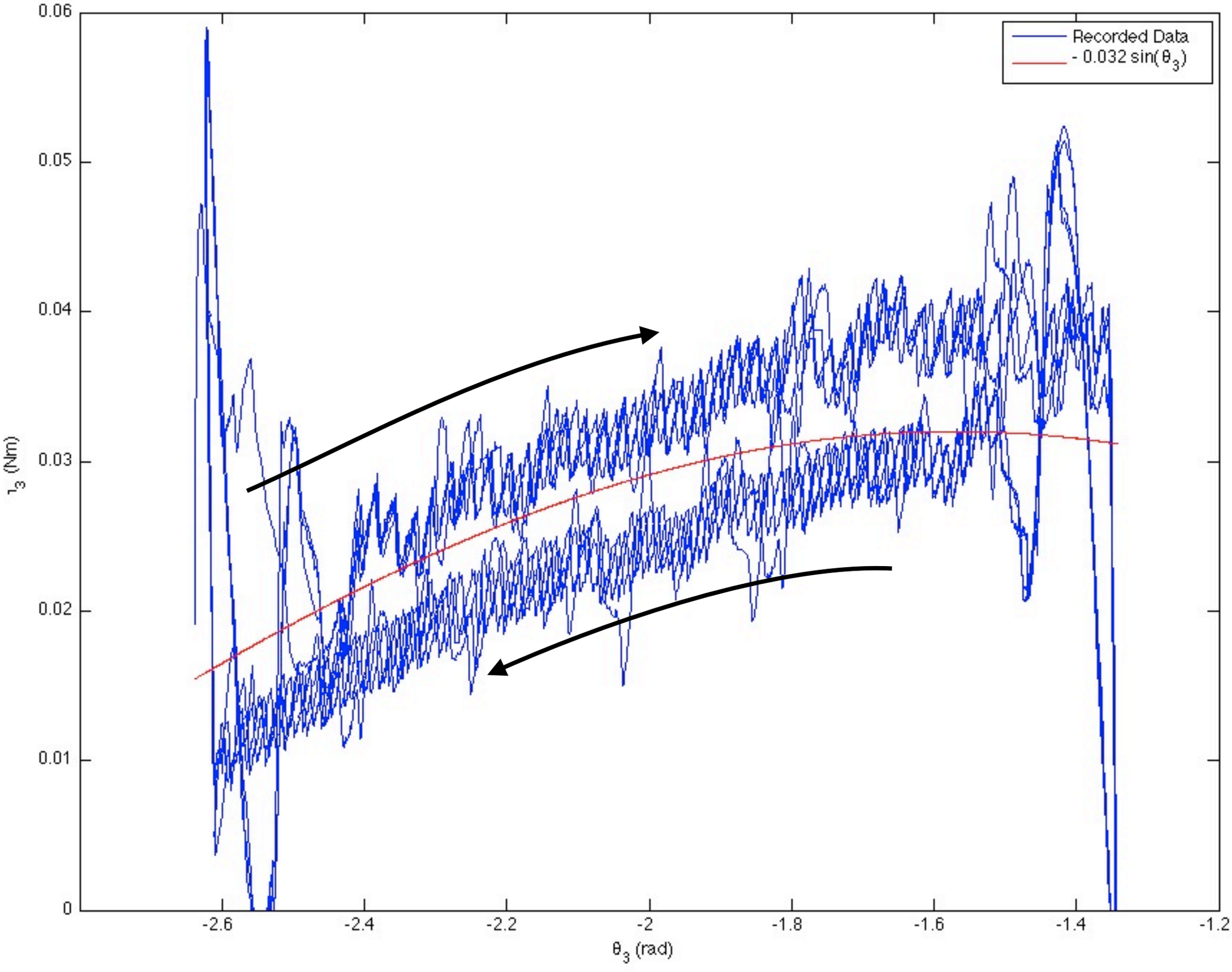

Applications Places
Figure 1
File Edit View Insert Tools Desktop Window Help
% Link lengths in millimeters.
l1 = 168;
l2 = 140;
l3 = 140;
AB R2012b
Editor - /home/pumast...
[MEAM.Design :

cal3


theta3
theta3 desired
Joint Angle (rad)
-1.2
-1.4
-1.6
-1.8
-2
-2.2
-2.4
-2.6
-2.8
0
2
4
6
8
10
12
14
Time (s)


tau3
Joint Torque (Nm)
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
2
4
6
8
10
12
14
Time (s)

cal3


Recorded Data
- 0.032 sin(θ3)
0.06
0.05
0.04
0.03
0.02
0.01
0
τ3 (Nm)
-2.6
-2.4
-2.2
-2
-1.8
-1.6
-1.4
-1.2
θ3 (rad)

Can we make this better?

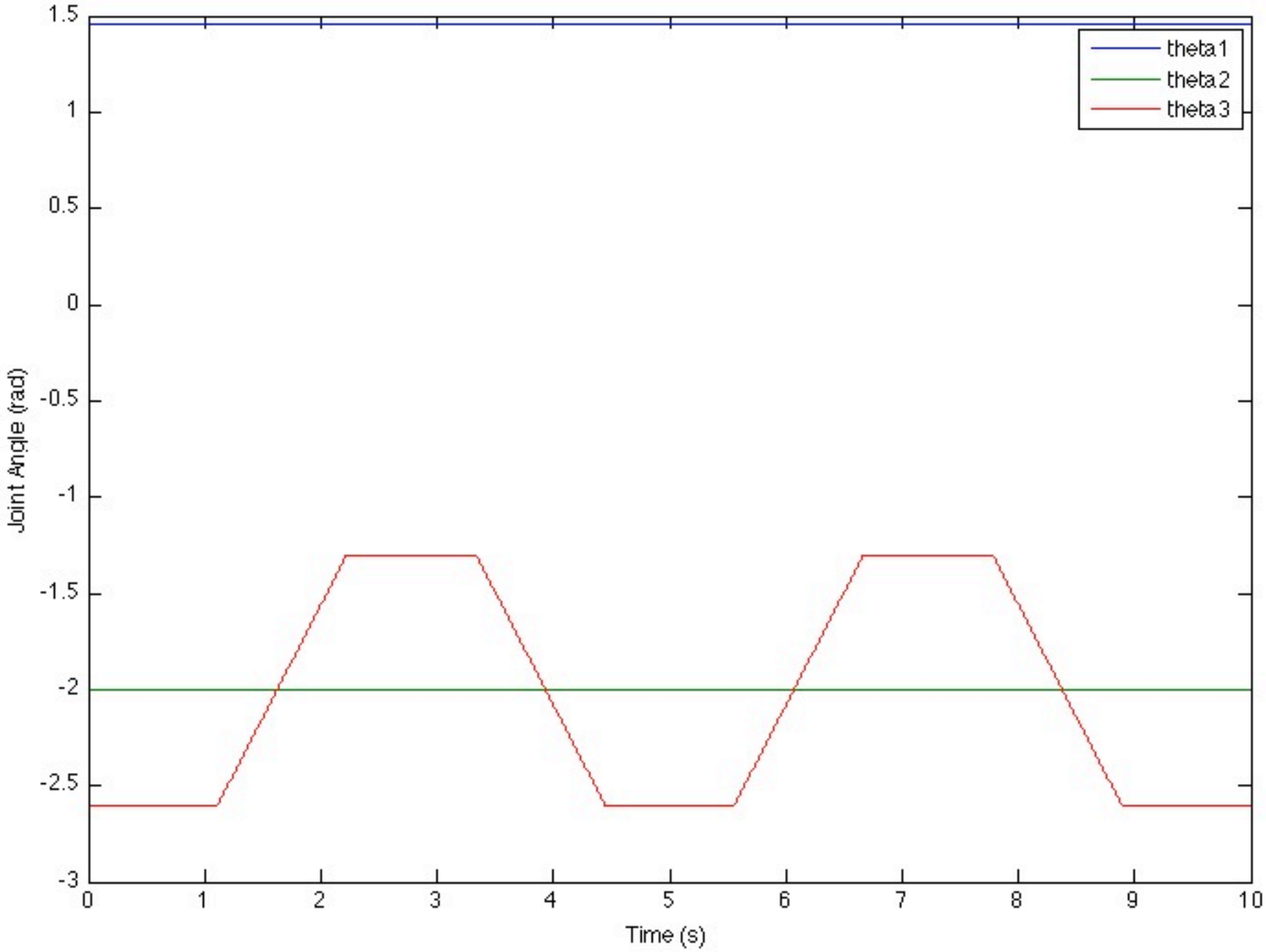
Editor - /Users/kuchenbe/Desktop/kjk/make_theta3_v2.m

File Edit Text Go Cell Tools Debug Desktop Window Help

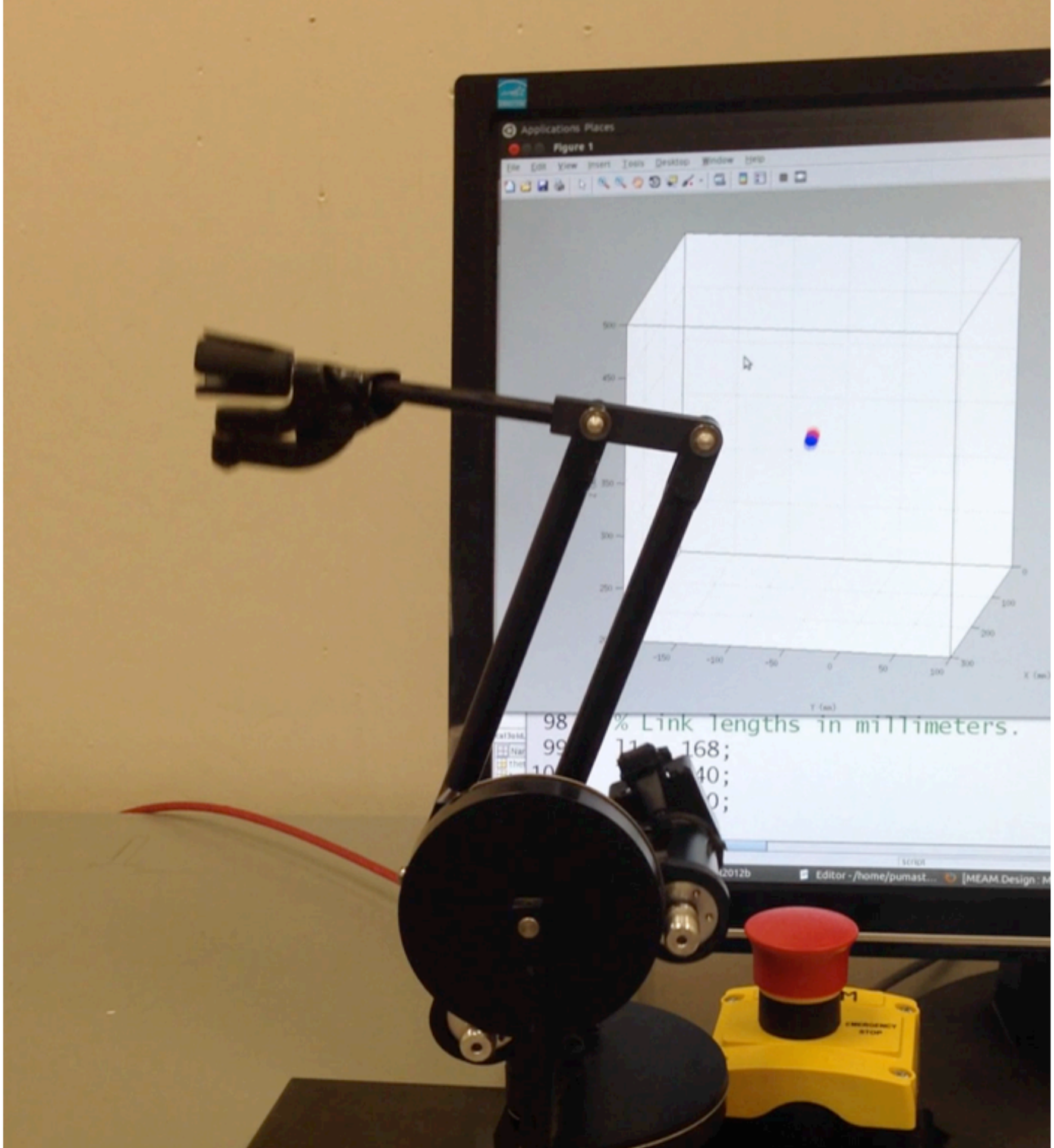
```
 1      % Make t.
 2 -    t = linspace(0,10,9000)';
 3
 4      % Make all thetas.
 5 -    thetas = zeros(9000,3);
 6 -    thetas(:,1) = 1.46;
 7 -    thetas(:,2) = -2;
 8 -    thetas(:,3) = -2.6;
 9 -    thetas(1001:2000,3) = linspace(-2.6,-1.3,1000)';
10 -    thetas(2001:3000,3) = -1.3;
11 -    thetas(3001:4000,3) = linspace(-1.3,-2.6,1000)';
12 -    thetas(4001:5000,3) = -2.6;
13 -    thetas(5001:6000,3) = linspace(-2.6,-1.3,1000)';
14 -    thetas(6001:7000,3) = -1.3;
15 -    thetas(7001:8000,3) = linspace(-1.3,-2.6,1000)';
16 -    thetas(8001:9000,3) = -2.6;
17
18     % Plot thetas.
19 -    figure(6);
20 -    plot(t,thetas)
21
```

ıox_demo.m | haptic_ball_team50.m | haptic_damping_team50.m | cmdhist.m | make_theta3_v2.m

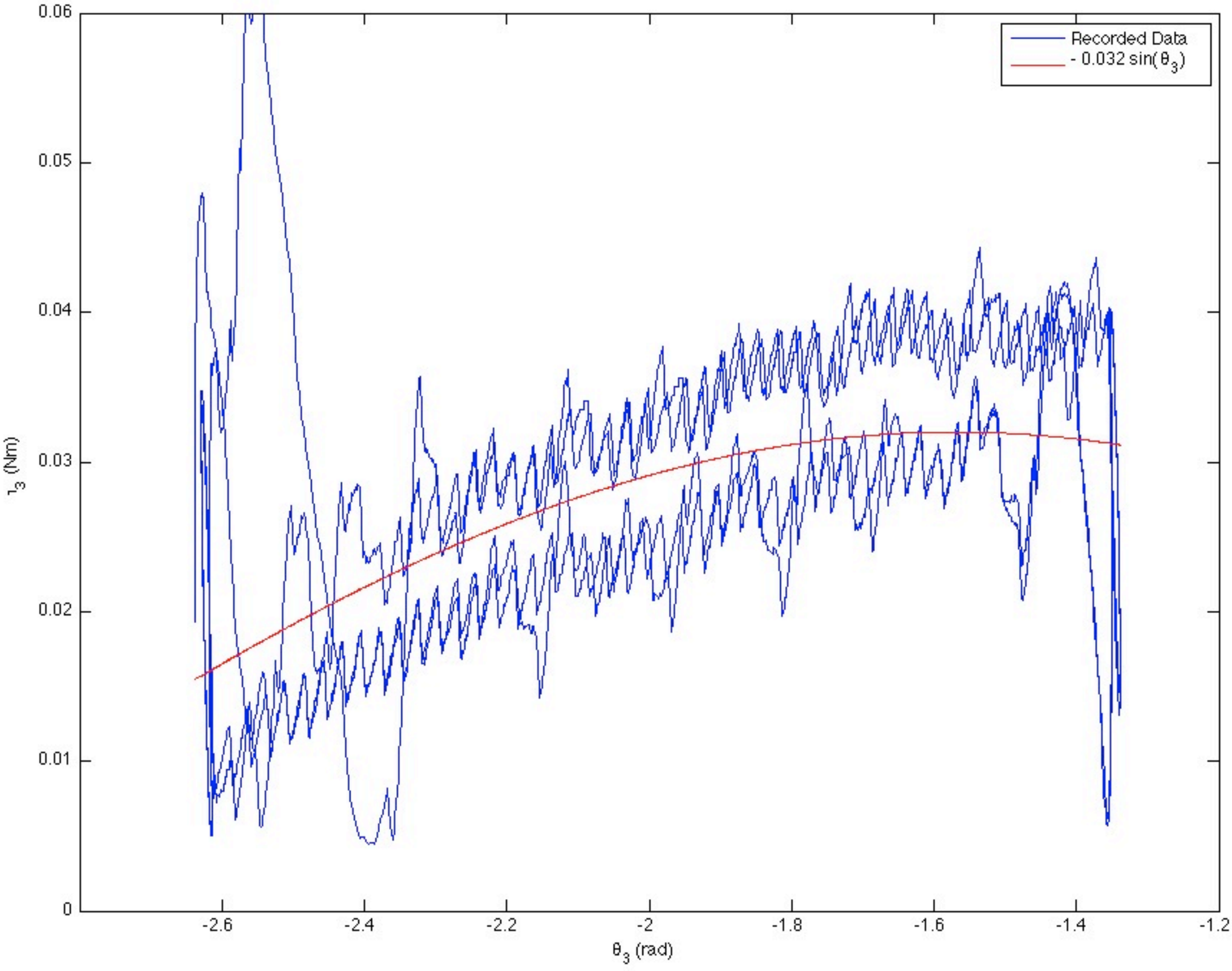
script Ln 15 Col 20


theta1
theta2
theta3
Joint Angle (rad)
Time (s)
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
0
1
2
3
4
5
6
7
8
9
10

cal3new


Applications Places
Figure 1
% Link lengths in millimeters.
Editor - /home/pumast...

cal3new


Recorded Data
- 0.032 sin(θ3)
τ3 (Nm)
θ3 (rad)
0.06
0.05
0.04
0.03
0.02
0.01
0
-2.6
-2.4
-2.2
-2
-1.8
-1.6
-1.4
-1.2

Can we make this better?

Trajectory Smoothing

start → end

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

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

cubic polynomial

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

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

Editor - /Users/kuchenbe/Desktop/kjk/make_theta3_v3.m

File Edit Text Go Cell Tools Debug Desktop Window Help

```
% Make an abbreviated time vector with 1000 elements.
t = 1:1000;

% Define initial conditions.
q0 = -2.6;
v0 = 0;

% Define final conditions.
qf = -1.3;
vf = 0;

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

% Put time elements into matrix.
mat = [1    t(1)     t(1)^2       t(1)^3;
       0       1     2*t(1)     3*t(1)^2;
       1  t(end)   t(end)^2     t(end)^3;
       0       1   2*t(end)   3*t(end)^2];

% Solve for coefficients.
as = mat \ conditions;

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

ox_demo.m | haptic_ball_team50.m | haptic_damping_team50.m | cmdhist.m | make_theta3_v3.m

script Ln 23 Col 1

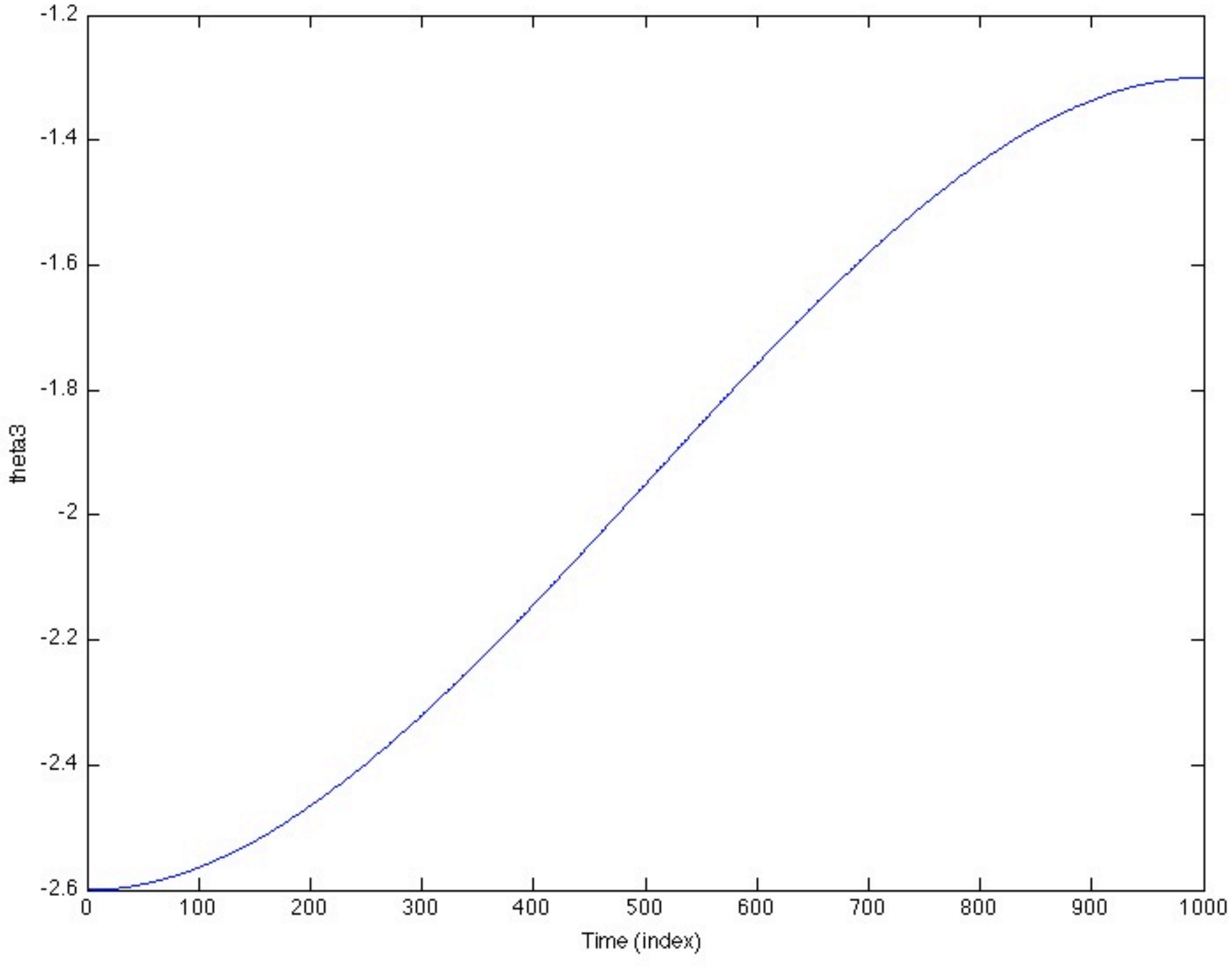
Editor - /Users/kuchenbe/Desktop/kjk/make_theta3_v3.m

File Edit Text Go Cell Tools Debug Desktop Window Help

```
10 -    vf = 0;
11
12      % Put initial and final conditions into a vector.
13 -    conditions = [q0; v0; qf; vf];
14
15      % Put time elements into matrix.
16 -    mat = [1    t(1)     t(1)^2       t(1)^3;
17             0       1     2*t(1)     3*t(1)^2;
18             1  t(end)   t(end)^2     t(end)^3;
19             0       1   2*t(end)   3*t(end)^2];
20
21      % Solve for coefficients.
22 -    as = mat \ conditions;
23
24      % Pull individual coefficients out.
25 -    a0 = as(1);
26 -    a1 = as(2);
27 -    a2 = as(3);
28 -    a3 = as(4);
29
30      % Calculate cubic trajectory with coefficients.
31 -    q = a0 + a1*t + a2*t.^2 + a3 * t.^3;
32
33      % Plot cubic trajectory.
34 -    figure(2)
35 -    plot(t,q)
36 -    xlabel('Time (index)')
37 -    ylabel('theta3')
38
```

ıox_demo.m | haptic_ball_team50.m | haptic_damping_team50.m | cmdhist.m | make_theta3_v3.m

script Ln 21 Col 13


-1.2
-1.4
-1.6
-1.8
-2
-2.2
-2.4
-2.6
theta3
0
100
200
300
400
500
600
700
800
900
1000
Time (index)

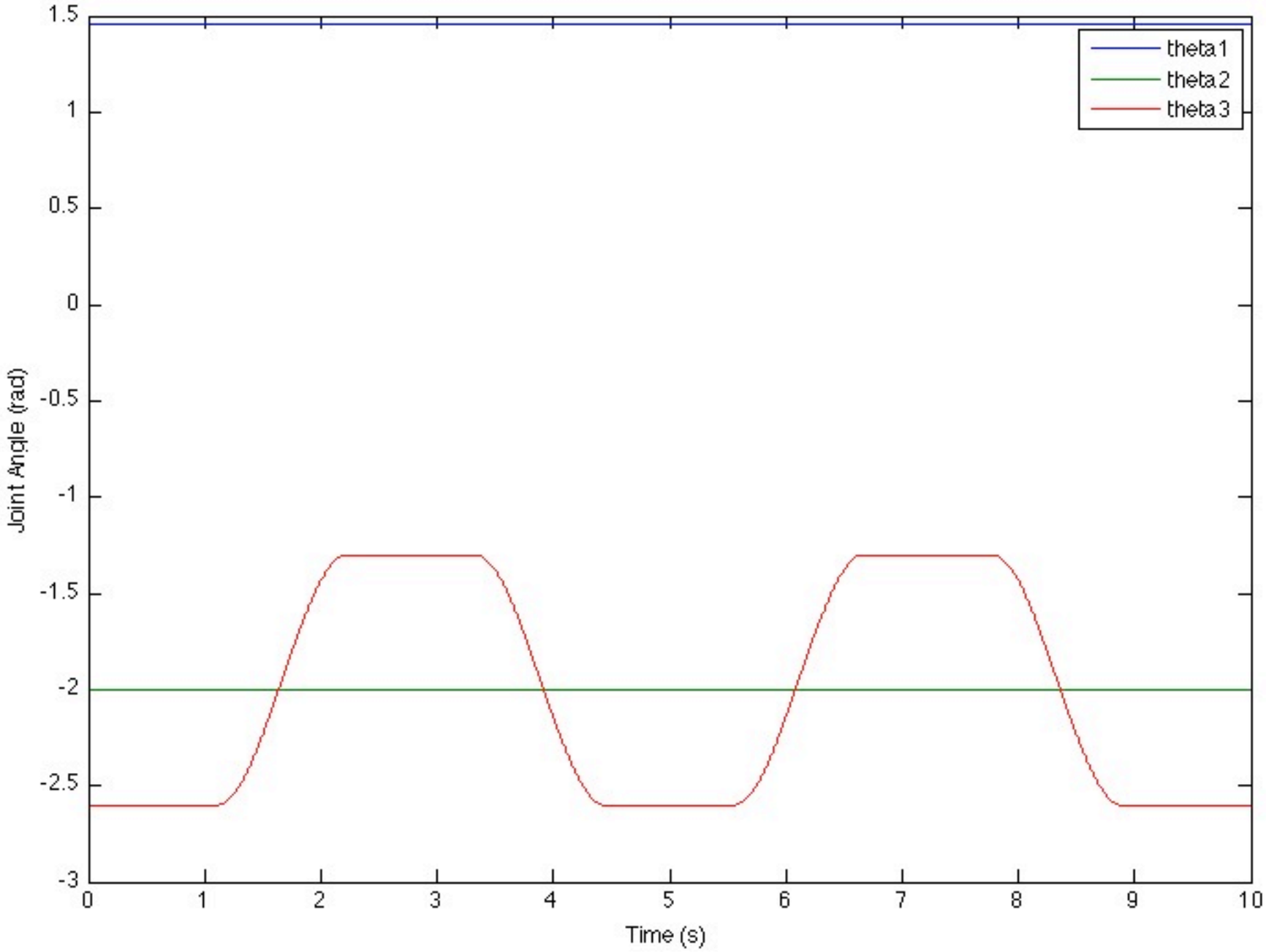
Editor - /Users/kuchenbe/Desktop/kjk/make_theta3_v3.m

File Edit Text Go Cell Tools Debug Desktop Window Help

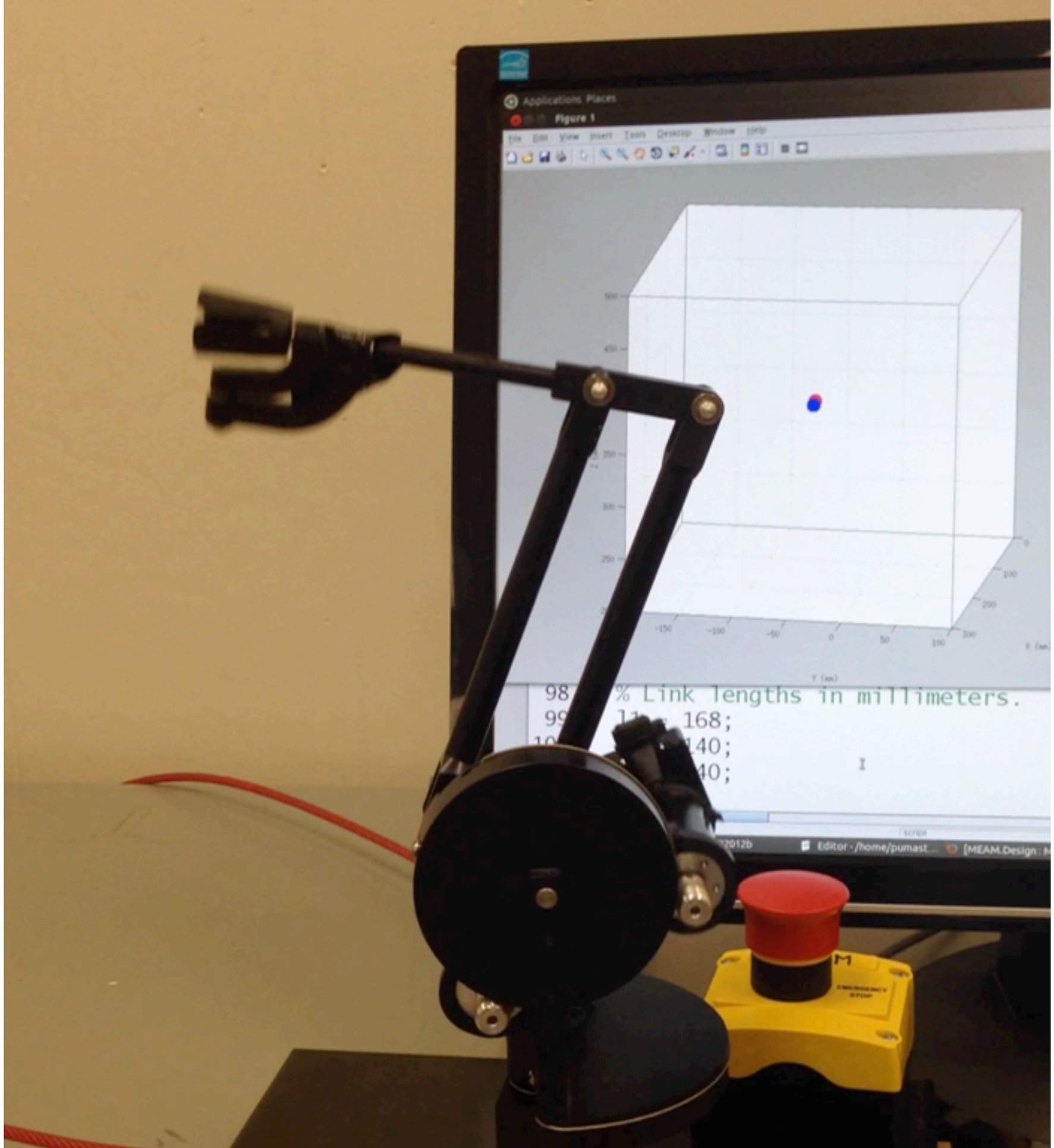
```
34 -    figure(2)
35 -    plot(t,q)
36 -    xlabel('Time (index)')
37 -    ylabel('theta3')
38
39
40      % Make t.
41 -    t = linspace(0,10,9000)';
42
43      % Make all thetas.
44 -    thetas = zeros(9000,3);
45 -    thetas(:,1) = 1.46;
46 -    thetas(:,2) = -2;
47 -    thetas(:,3) = -2.6;
48 -    thetas(1001:2000,3) = q';
49 -    thetas(2001:3000,3) = -1.3;
50 -    thetas(3001:4000,3) = flipud(q');
51 -    thetas(4001:5000,3) = -2.6;
52 -    thetas(5001:6000,3) = q';
53 -    thetas(6001:7000,3) = -1.3;
54 -    thetas(7001:8000,3) = flipud(q');
55 -    thetas(8001:9000,3) = -2.6;
56
57      % Plot thetas.
58 -    figure(6);
59 -    plot(t,thetas)
60 -    xlabel('Time (s)')
61 -    ylabel('Joint Angle (rad)')
62 -    legend('theta1', 'theta2', 'theta3')
```

ic_box_demo.m | haptic_ball_team50.m | haptic_damping_team50.m | cmdhist.m | make_theta3_v3.m

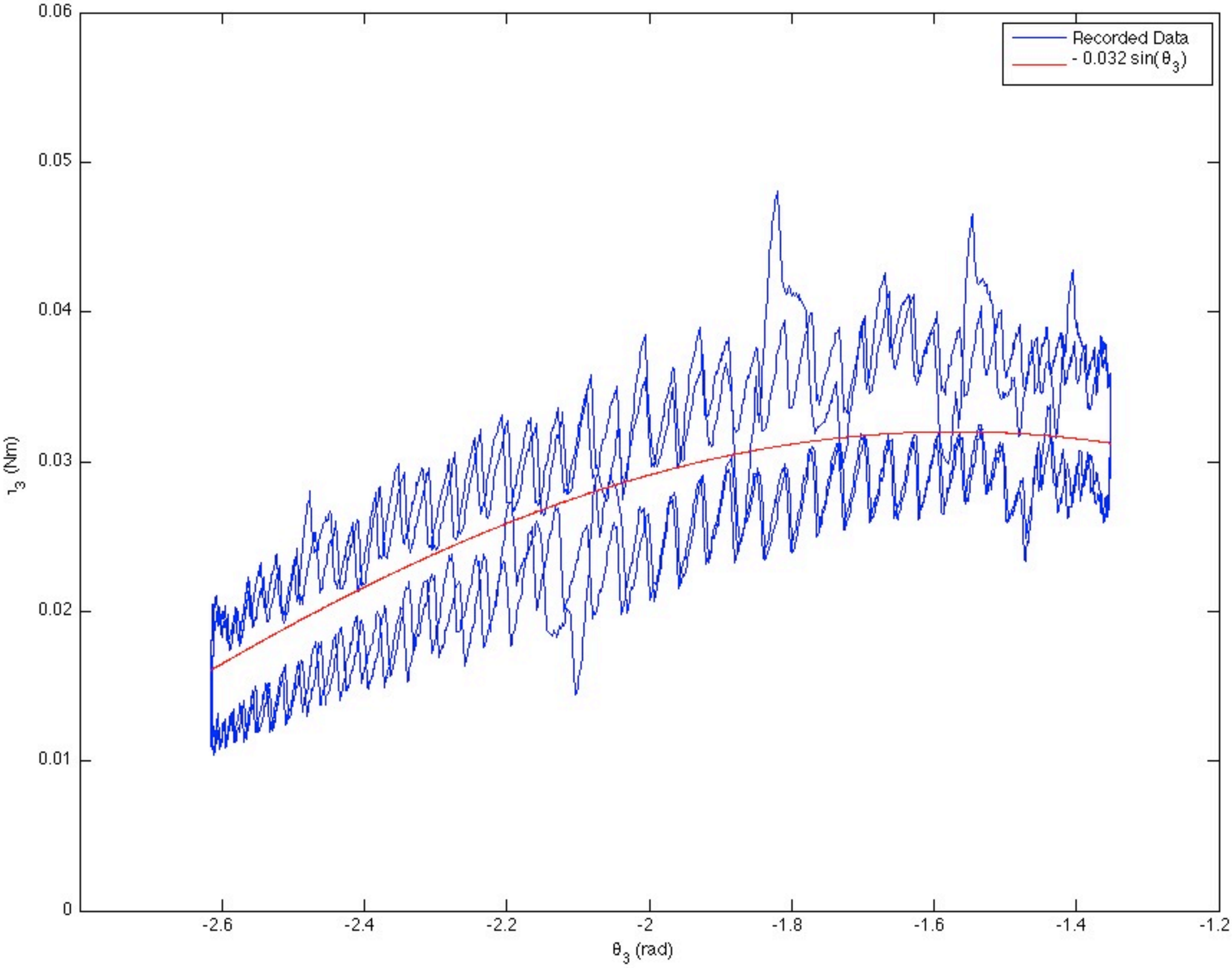
script Ln 46 Col 18


theta1
theta2
theta3
Joint Angle (rad)
Time (s)
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
0
1
2
3
4
5
6
7
8
9
10

cal3newnew


Applications Places
Figure 1
File Edit View Insert Tools Desktop Window Help
% Link lengths in millimeters.
168;
140;
40;
Editor - /home/pumast...
[MEAM.Design :

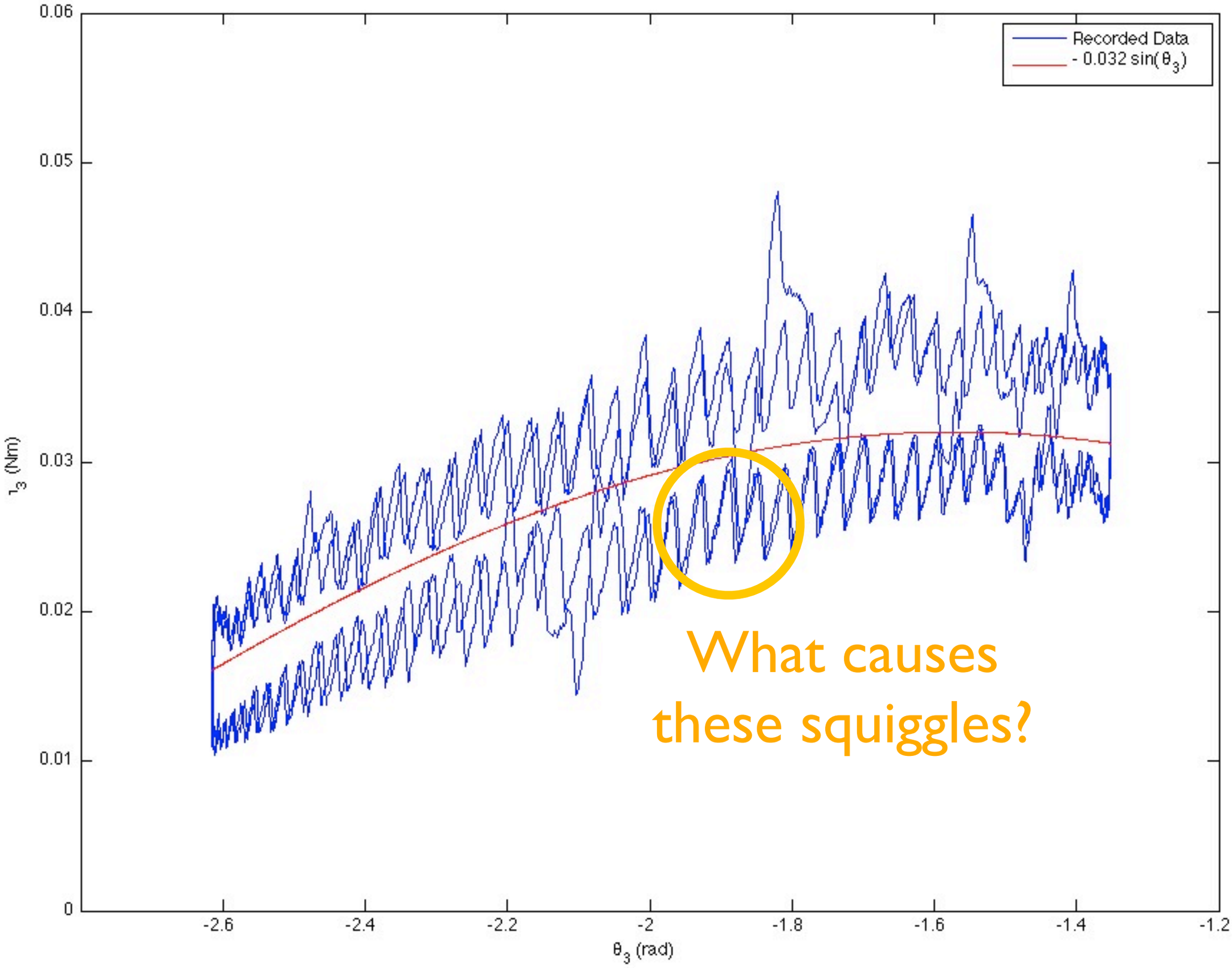
cal3newnew


Recorded Data
- 0.032 sin(θ3)
0.06
0.05
0.04
0.03
0.02
0.01
0
τ3 (Nm)
-2.6
-2.4
-2.2
-2
-1.8
-1.6
-1.4
-1.2
θ3 (rad)

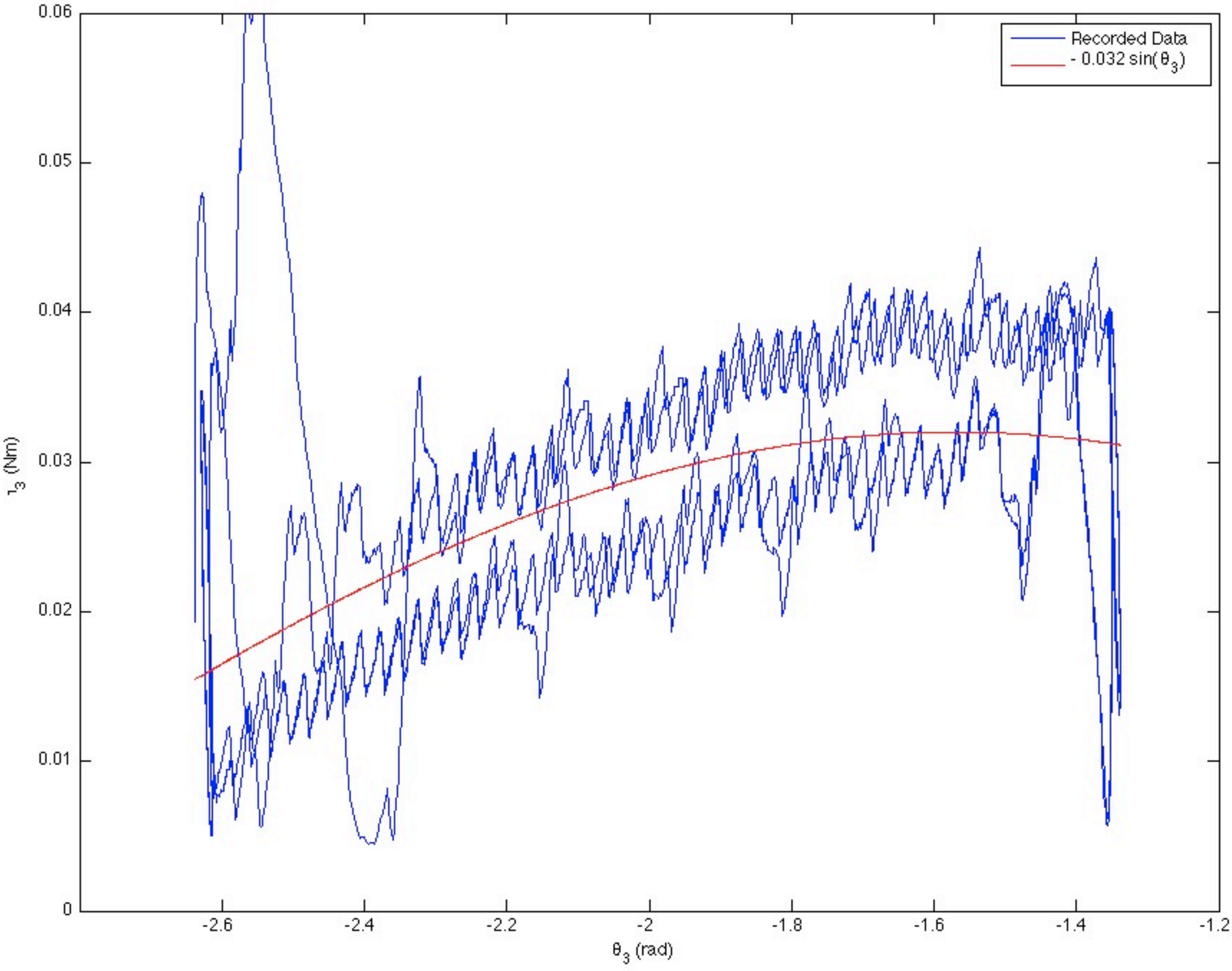
Questions ?

Can we make this better?

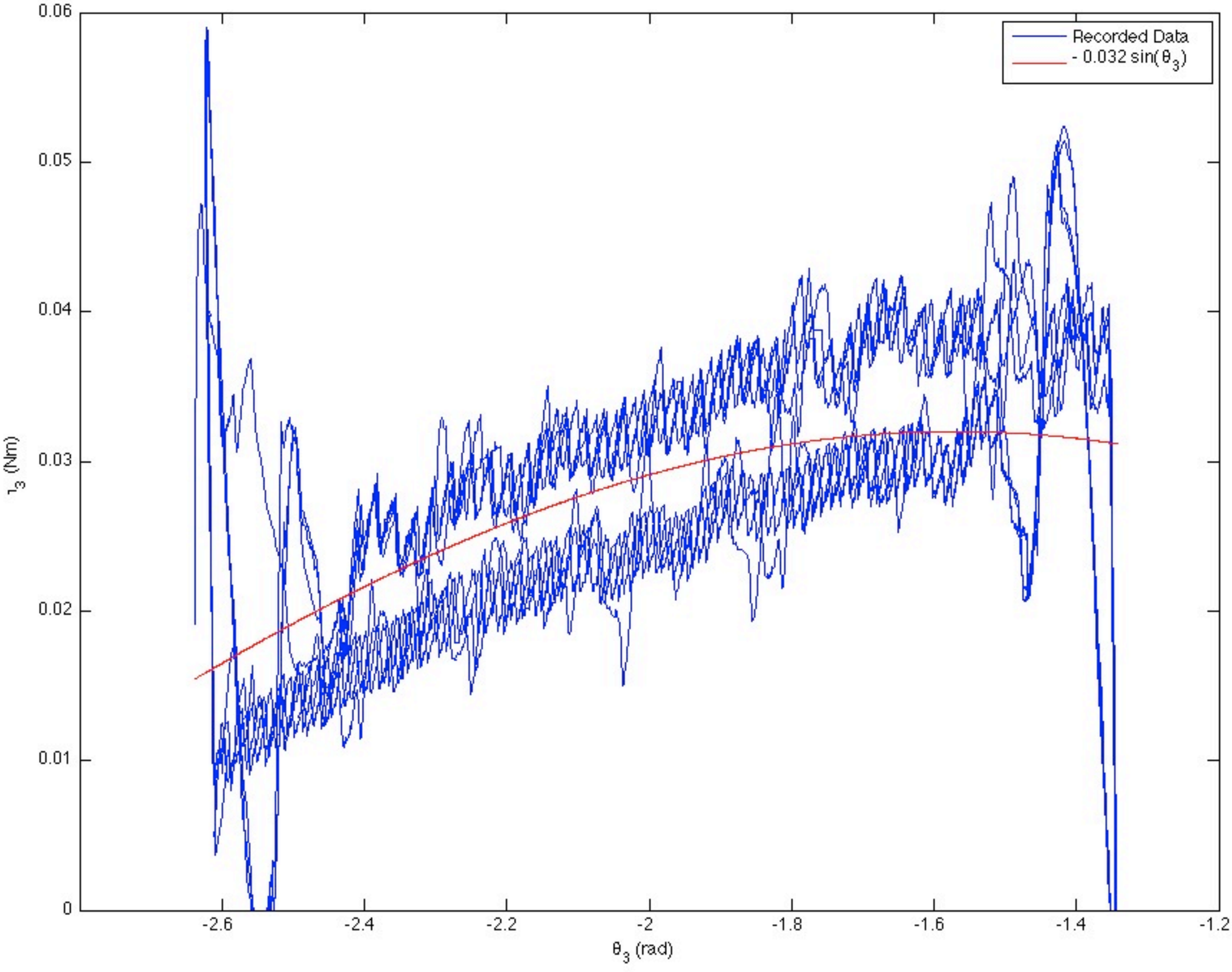
cal3newnew

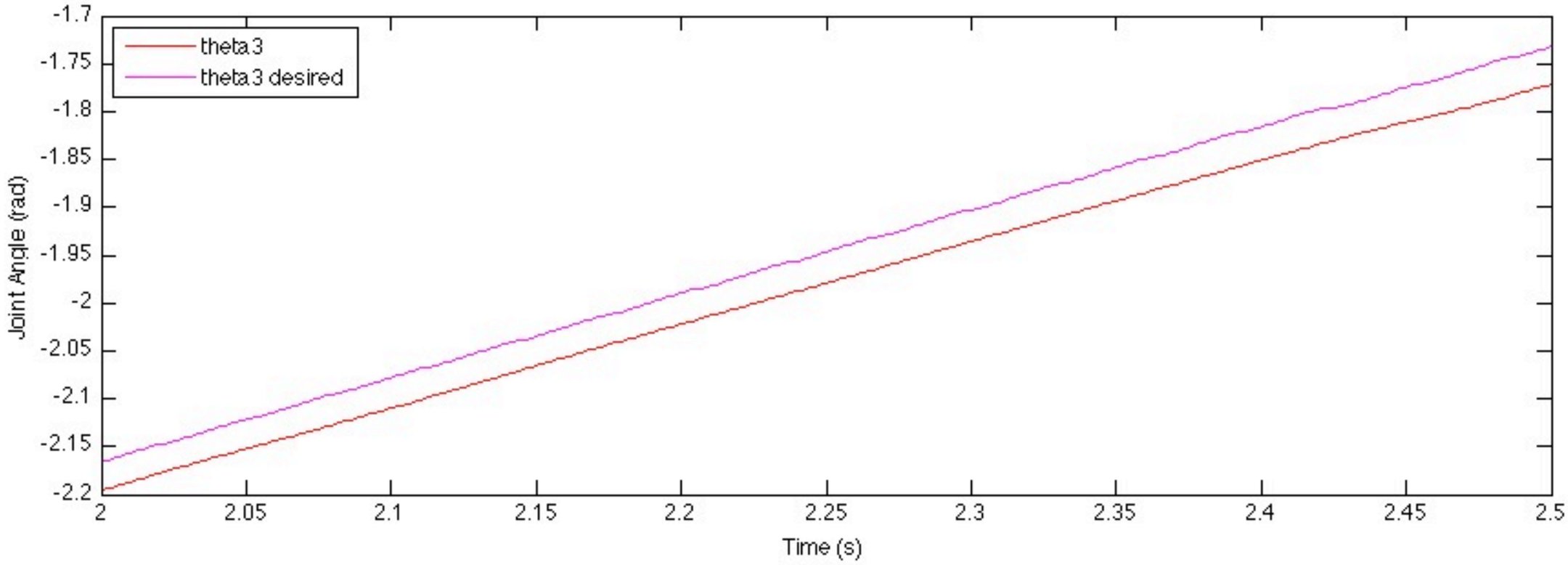

Recorded Data
- 0.032 sin(θ3)
0.06
0.05
0.04
0.03
0.02
0.01
0
τ3 (Nm)
-2.6
-2.4
-2.2
-2
-1.8
-1.6
-1.4
-1.2
θ3 (rad)
What causes
these squiggles?

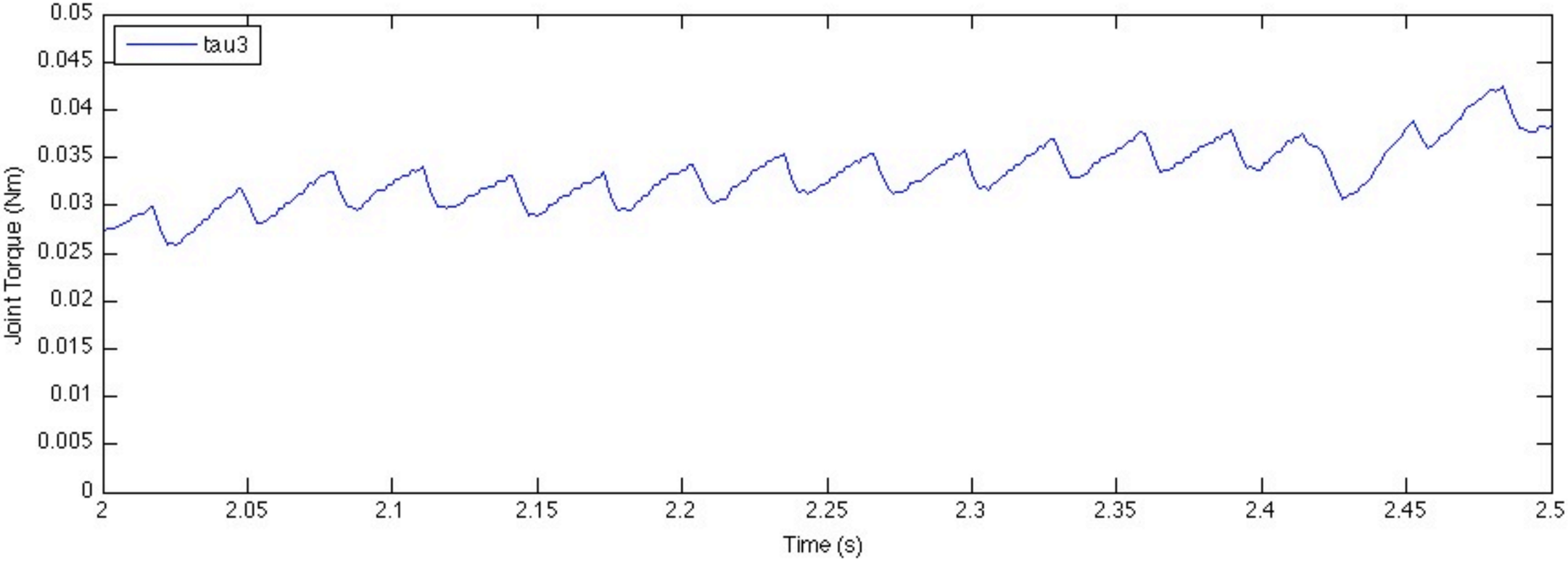
cal3new

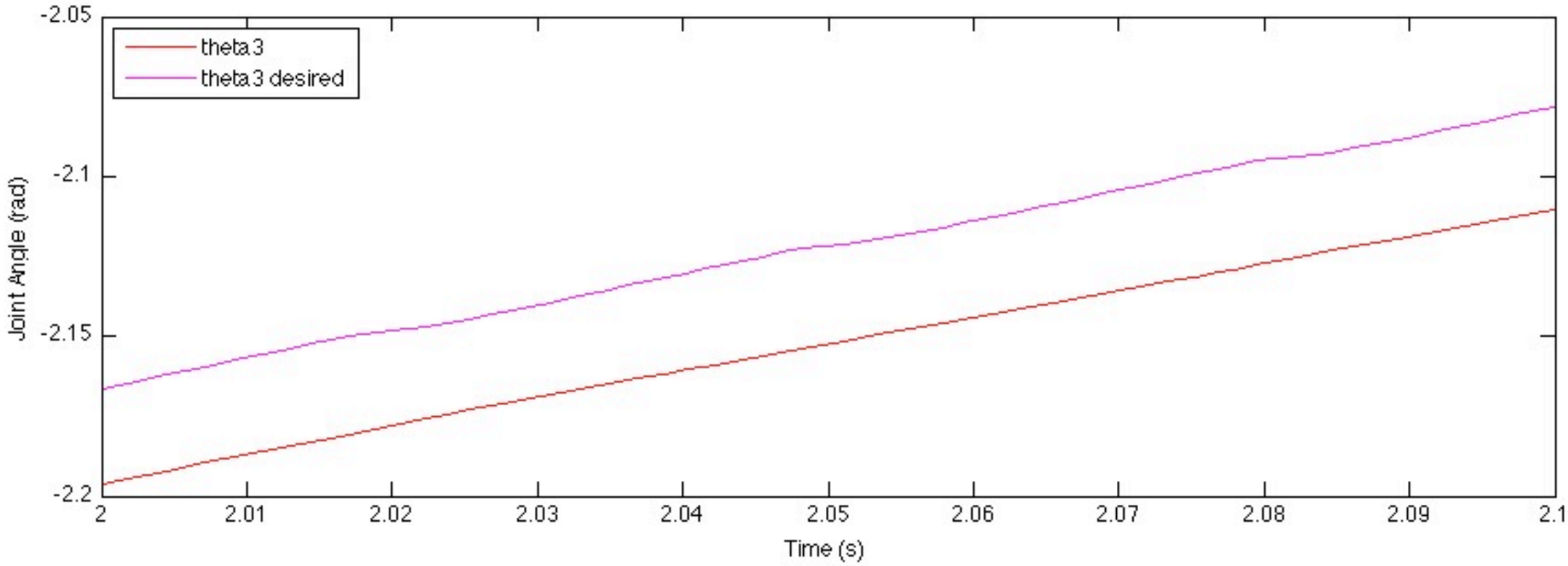

Recorded Data
- 0.032 sin(θ3)
τ3 (Nm)
θ3 (rad)
0.06
0.05
0.04
0.03
0.02
0.01
0
-2.6
-2.4
-2.2
-2
-1.8
-1.6
-1.4
-1.2

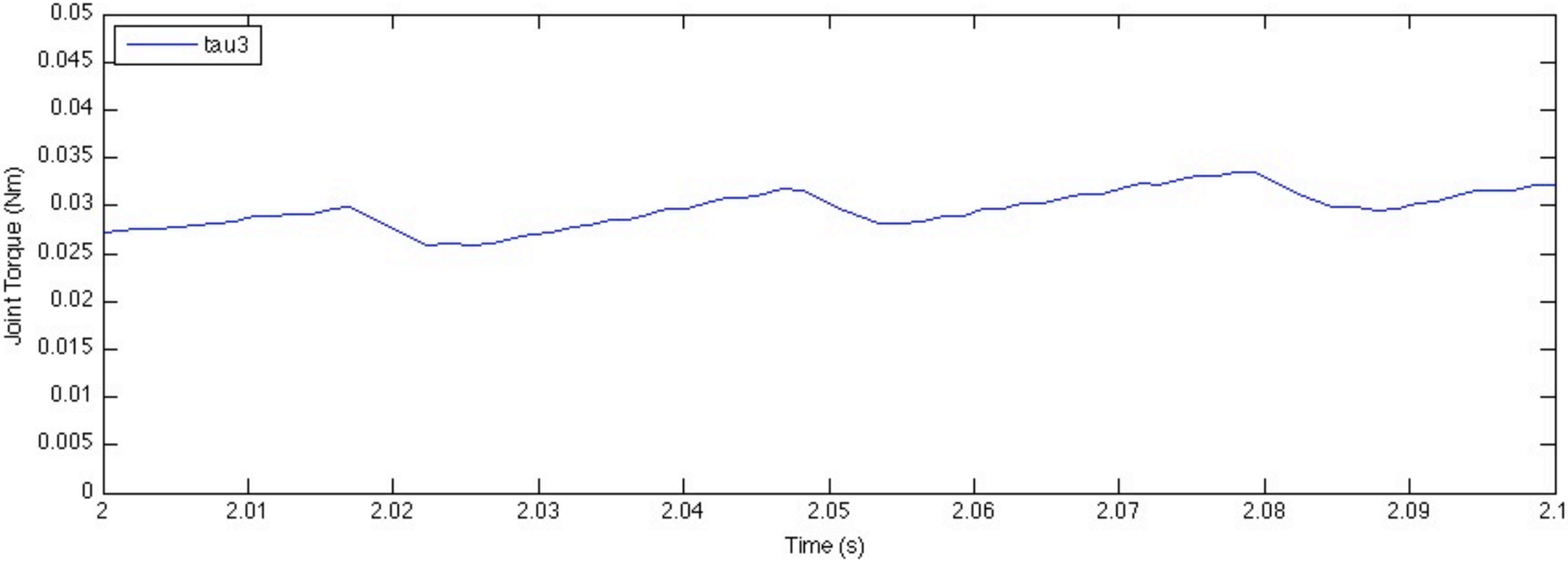
cal3


Recorded Data
- 0.032 sin(θ3)
τ3 (Nm)
θ3 (rad)
0.06
0.05
0.04
0.03
0.02
0.01
0
-2.6
-2.4
-2.2
-2
-1.8
-1.6
-1.4
-1.2


theta3
theta3 desired
Joint Angle (rad)
-1.7
-1.75
-1.8
-1.85
-1.9
-1.95
-2
-2.05
-2.1
-2.15
-2.2
2
2.05
2.1
2.15
2.2
2.25
2.3
2.35
2.4
2.45
2.5
Time (s)


tau3
Joint Torque (Nm)
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
2
2.05
2.1
2.15
2.2
2.25
2.3
2.35
2.4
2.45
2.5
Time (s)


theta3
theta3 desired
Joint Angle (rad)
-2.05
-2.1
-2.15
-2.2
2
2.01
2.02
2.03
2.04
2.05
2.06
2.07
2.08
2.09
2.1
Time (s)


tau3
Joint Torque (Nm)
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
2
2.01
2.02
2.03
2.04
2.05
2.06
2.07
2.08
2.09
2.1
Time (s)

What is going on?