```
%--------------------
% Author: Akira Nagamori
% Last update: 6:8/17
% This is the code used to generate all the data for the common drive
paper
%--------------------

clear all
close all
clc

%--------------------
% User-defined parameters
%--------------------
% Architectural parameters of muscle of interest (TA in this study)
(Elias et al. 2014;Arnold et al. 2010)
muscle_parameter.pennationAngle = 9.6*pi/180; % pennation angle
(radians)
muscle_parameter.mass = 0.15; % muscle mass (grams)
muscle_parameter.optimalLength = 6.8; % muscle opimtal length (cm)
muscle_parameter.tendonLength = 24.1; % tendon length (cm)
muscle_parameter.muscleInitialLength = 6.8; % initial muscle length
(cm)
muscle_parameter.tendonInitialLength = 24.1; % initial tendon length
(cm)
muscle_parameter.MVIC = 363.4008-2.043; % maximal voluntary isometric
force
muscle_parameter.baseline = 2.043; % passive force at zero activaiton

%--------------------
% Delays based on limb length and conduction velocity of each pathway
(Elias et al. 2014; Kandel et al., 2000)
distanceMuscleSpinalCord = 0.8; % distance between muscle and spinal
cord (cm)
conductionVelocity_efferent = 48.5; % conduction velocity of alpha-
motoneurone (m/s)
conductionVelocity_Ia = 64.5; % conduction velocity of Ia afferent
(m/s)
conductionVelocity_II = 32.5; % conIduction velocity of I afferent
(m/s)
conductionVelocity_Ib = 59; % conduction velocity of II afferent (m/s)
synaptic_delay = 2; %ms

% Calculate delays based on the conduction velocity and limb length
delay_parameter.efferent =
round(distanceMuscleSpinalCord/conductionVelocity_efferent*1000); %
conduction delays along alpha-motoneuron (ms)
delay_parameter.Ia =
round(distanceMuscleSpinalCord/conductionVelocity_Ia*1000) +
synaptic_delay; % conduction delays along Ia afferent (ms)
delay_parameter.II =
round(distanceMuscleSpinalCord/conductionVelocity_II*1000) +
2*synaptic_delay; % conduction delays along II afferent (ms)
delay_parameter.Ib =
round(distanceMuscleSpinalCord/conductionVelocity_Ib*1000) +
2*synaptic_delay; % conduction delays along Ib afferent (ms)
delay_parameter.tracking = 50;
```

```
%--------------------
% Gain parameters for each neural pathway
gain_parameter.gammaDynamic = 70; % gamma dynamic
gain_parameter.gammaStatic = 70; % gamma static
gain_parameter.Ia = 300; % constant factor to normalize Ia firing rate
into a value between 0 and 1
gain_parameter.Ia_PC = -0.1; % presynaptic control of Ia afferent
gain_parameter.II = 3000; % constant factor to normalize II firing rate
into a value between 0 and 1
gain_parameter.Ib = 400; % constant factor to normalize Ib firing rate
into a value between 0 and 1
gain_parameter.Ib_PC = -.3; % presynaptic control of Ib afferent
%--------------------
% Define force trajectory that model needs to track
Fs = 10000;
t = 0:1/Fs:35; % time vector for simulation
amp = 0.2; % target %MVC
targetTrajectory = [zeros(1,1*Fs) (amp)*1/2*t(1:2*Fs)
amp*ones(1,length(t)-3*Fs)]; % target force trajectory

% Define type of control
% 0 = no feedback = only feedforward input
% 1 = tracking controller + proprioceptive feedback
feedbackOption = 1;

%% Run afferented muscle model
% output (see l.334 of AffremetedMuscleModel.m for the details)
output =
AfferentedMuscleModel(muscle_parameter,delay_parameter,gain_parameter,t
argetTrajectory,feedbackOption);

%%
% calculate power spectrum and plot it
[pxx,f] = pwelch(output.ForceTendon(10*Fs:end)-
mean(output.ForceTendon(10*Fs:end)),gausswin(5*Fs),2.5*Fs,[],Fs);
figure()
plot(f(1:50),pxx(1:50))
```