SEA Project Report 1

Yi Yang

09/19/2016

This week, I re-plot driving point impedance theoretical results(virtual stiffness), codes and plots shown below:

```
% This program is used for obtaining bode plots of linear model
% Admittance Control Devices
s = tf('s');
dataID = sysID();
Z_h = dataID.M * s + dataID.B; % the inherent impedance of haptic devices
Kve = 400; % [N/m], virtual environment spring constant
Z_ve = Kve/s; % impedance of virtual environment
C = 2; % portional control law
Z_padm = (dataID.n * C * Z_ve + Z_h)/(dataID.n * C + 1);

% Series Elastic Actuator Device
k = 904; % [N/m], Physical spring constant
kk = k/s + dataID.b;
Z_e = dataID.m * s;
Z_psea = (Z_e * (Z_h + k * dataID.n * C/s + kk) + kk * (Z_h + dataID.n * C * Z_ve))/(Z_h + ...
    k * dataID.n * C/s + kk);

close all; figure(1);
opts = bodeoptions('cstprefs');
opts.PhaseVisible = 'off';
opts.FreqUnits = 'Hz';
opts.Grid = 'on';
bodemag(Z_padm, opts);
hold on;
bodemag(Z_psea, opts);
bodemag(Z_ve, '--', opts);
legend('Admittance', 'SEA(k= 904)', 'ideal line');
title('Virtual Stiffness');
hold off;
```

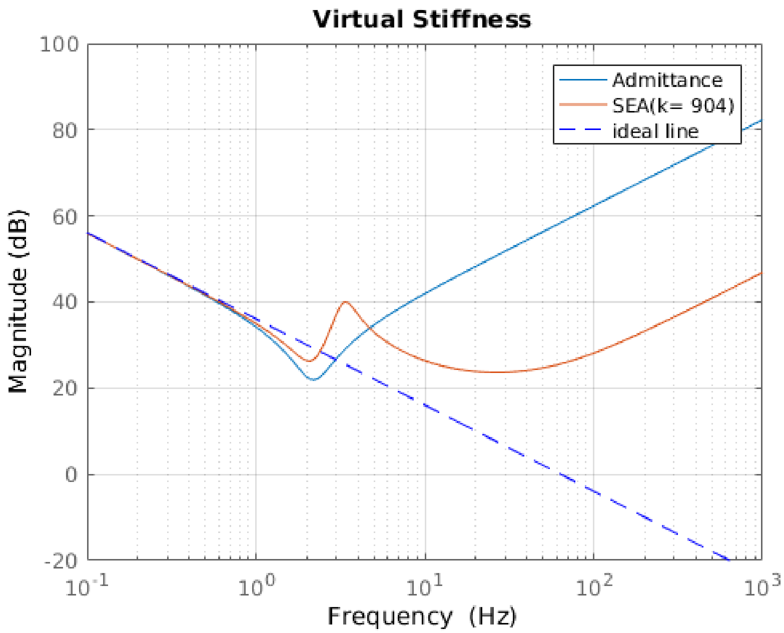
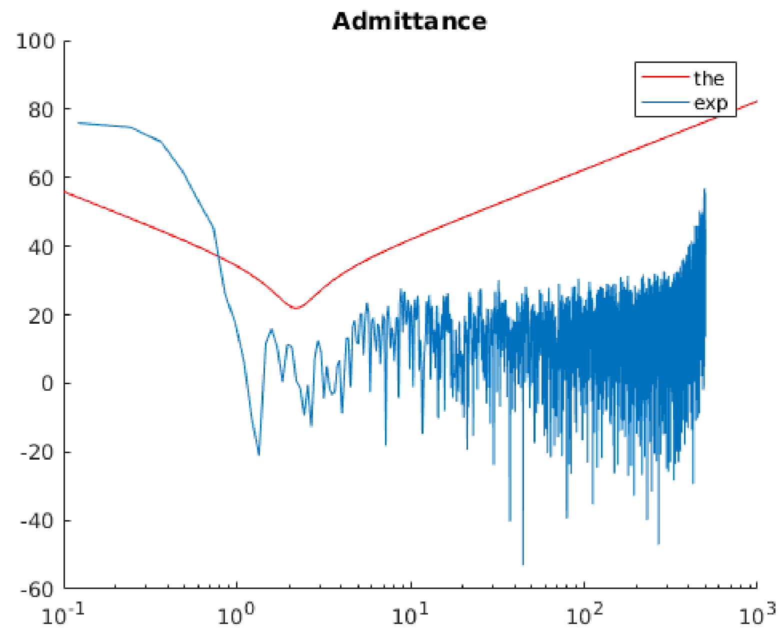


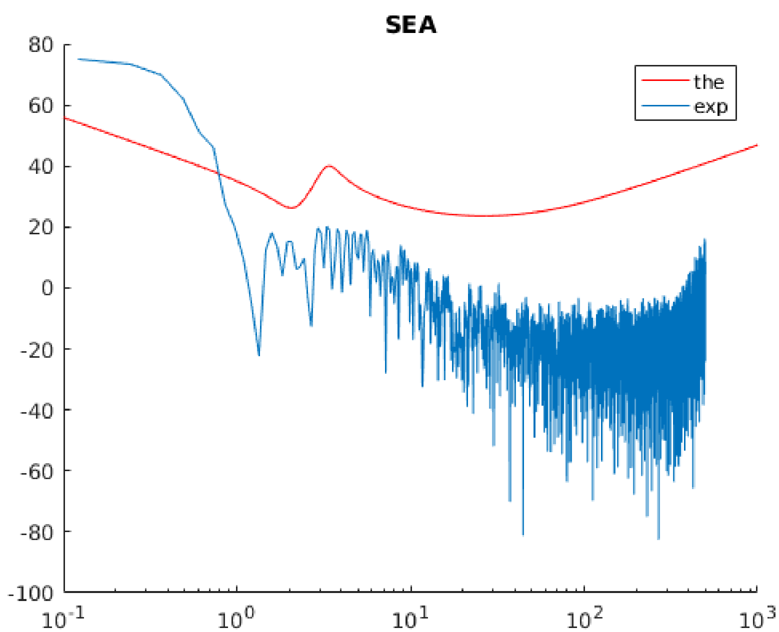
Figure 1: Driving point impedance theoretical results(virtual stiffness)

After plotting the theoretical transfer function, I want to use some real data to estimate the transfer functions and compare them with the theoretical ones. First, I input a step signal with time step size $0.001s$, I add a white noise ($SNR = 10$) to the input signal and then apply it to the linear matlab model. By using *tfestimate* function in Matlab, I can get the transfer function for admittance and SEA schemes respectively. The codes and plots are attached below:

```
% This program is to generate Bode plots using the real recorded data, but
% the model we use is still linear Matlab model
% by Yi Yang
% Date: 9/23/2016
%--------------------------------------------------------------------------
% Admittance Control Devices
clear all; close all;
s = tf('s');
dataID = sysID();
Z_h = dataID.M * s + dataID.B; % the inherent impedance of haptic devices
Kve = 400; % [N/m], virtual environment spring constant
Z_ve = Kve/s; % impedance of virtual environment
C = 2; % portional control law
Z_padm = (dataID.n * C * Z_ve + Z_h)/(dataID.n * C + 1);

% Series Elastic Actuator Device
k = 904; % [N/m], Physical spring constant
kk = k/s + dataID.b;
Z_e = dataID.m * s;
Z_psea = (Z_e * (Z_h + k * dataID.n * C/s + kk) + kk * (Z_h + dataID.n * C * Z_ve))/(Z_h + ...
    k * dataID.n * C/s + kk);

% Add sinusoidal signal with white noise of snr = 10 to the linear model
t = linspace(0, 10, 10001); % time step chosen to be ¬ 0.001
f0 = ones(length(t), 1);
fi = awgn(f0, 10, 0); % the measured loadCellForceA
```

```matlab
uo_adm = lsim(1/Z_padm, fi, t);
uo_sea = lsim(1/Z_psea, fi, t);
%figure(1);
%hold on;
%plot(t, uo_adm);
%plot(t, uo_sea);
%hold off;

% Now use output signal to estimate the transfer function
fs = 1/(t(2) - t(1));
NFFT = 2^(nextpow2(length(t)) - 1);
[TF_adm, freq1] = tfestimate(uo_adm, f0, [], [], NFFT, fs);
[TF_sea, freq2] = tfestimate(uo_sea, f0, [], [], NFFT, fs);
Z_adm_e = 1./TF_adm;
Z_sea_e = 1./TF_sea;

% In order to compare theoretical and test data generated curve
[MagTh_adm, PhaseTh_adm, FreqTh_adm] = bode(Z_padm, {2*pi*1e-1, 2*pi*1e3});
[MagTh_sea, PhaseTh_sea, FreqTh_sea] = bode(Z_psea, {2*pi*1e-1, 2*pi*1e3});
Mag_adm = zeros(length(MagTh_adm), 1);
Freq_adm = FreqTh_adm./(2*pi);
Mag_sea = zeros(length(MagTh_sea), 1);
Freq_sea = FreqTh_sea./(2*pi);
for i = 1:length(MagTh_adm)
    Mag_adm(i, 1) = MagTh_adm(1, 1, i);
    Mag_adm(i, 1) = 20 * log10(Mag_adm(i, 1));
end
for i = 1:length(MagTh_sea)
    Mag_sea(i, 1) = MagTh_sea(1, 1, i);
    Mag_sea(i, 1) = 20 * log10(Mag_sea(i, 1));
end
% opts = bodeoptions('cstprefs');
% opts.PhaseVisible = 'off';
% opts.FreqUnits = 'Hz';
% opts.Grid = 'on';
figure(2);
% bodemag(Z_padm, opts);
hold on;
semilogx(Freq_adm, Mag_adm, 'r');
semilogx(freq1, mag2db(abs(TF_adm)));
legend('the','exp');
set(gca,'xscale','log');
title('Admittance')
hold off;
figure(3);
% bodemag(Z_ve, '--', opts);
hold on;
semilogx(Freq_sea, Mag_sea, 'r');
semilogx(freq2, mag2db(abs(TF_sea)));
set(gca,'xscale','log');
legend('the', 'exp');
title('SEA');
hold off;
```





We can find that there is large discrepancy between theoretical curve and the curve generated by adding noise to the input signal. And I am not sure whether the method is correct to generate these results.

Appendix

System ID is shown below:

```
% system Identification
function dataID = sysID()
    dataID.M = 176; % [kg], Stage and motor inertia
    dataID.B = 1051; % [kg/s], Stage damping
    dataID.m = 0.035; % [kg], End-effector mass
    dataID.b = 15.05; % [kg/s], End-effector damping
    dataID.k = 1300; % [N/m], Physical spring stiffness
    dataID.n = 41.8; % [-], Gear ratio
end
```