Table of Contents

MAE 283A: Homework 2, Question 3

```
% By David Lim
```

3.1: Parameter estimation

```
% Load data
clear
load('mass_spring_damper.mat')

% Define indices
t_i = 3;
N = length(t);
idx = (t_i:N);

% Define data matrices
PHI = [ u(idx-1) u(idx-2) -y(idx-1) -y(idx-2) ];
Y = y(idx);

% Compute LS estimate
theta_LS = PHI\Y
```

```
theta_LS =
    0.0327
    0.0344
   -1.7538
    0.9011
```

3.2: Simulation

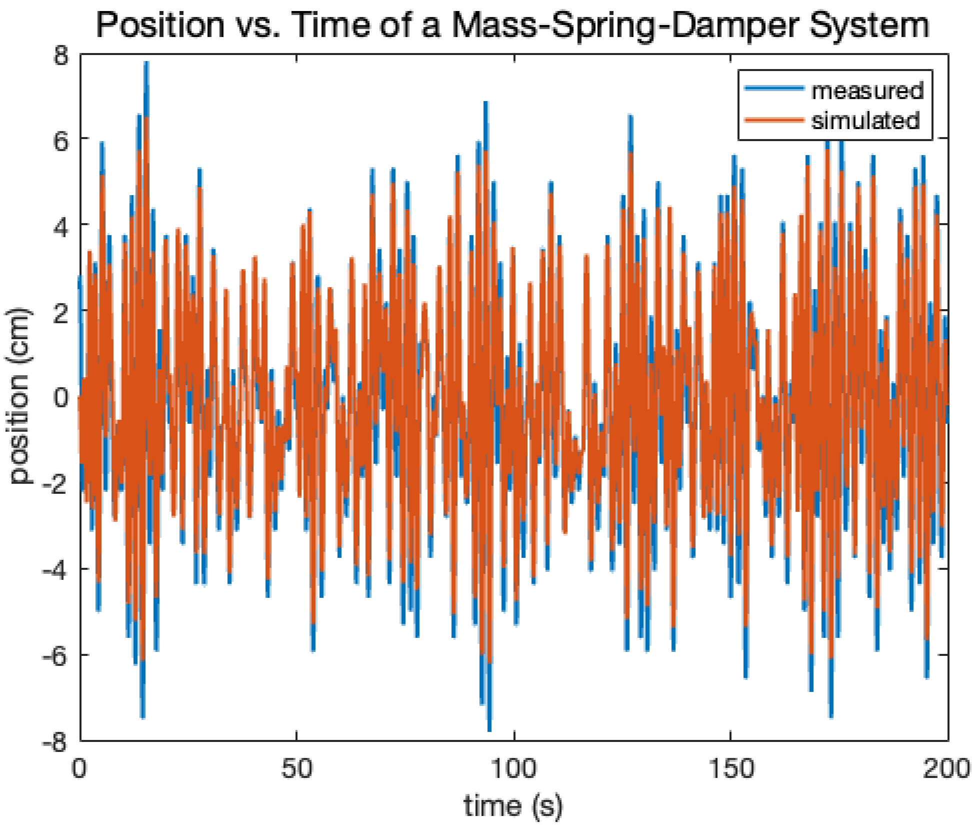
```
% Sampling time:
Delta_T = mean(diff(t));

% Define discrete-time transfer function model
num = theta_LS(1:2)';
den = [1 theta_LS(3:4)'];
G = tf(num,den,Delta_T);

% Simulate model with input data
y_sim = lsim(G,u,t);

% Plot results
```

```
figure(1)
h = plot(t,[y y_sim],'LineWidth',2);
xlabel('time (s)')
ylabel('position (cm)')
legend('measured','simulated')
set(gca,'FontSize',14)
title('Position vs. Time of a Mass-Spring-Damper
System','FontWeight','Normal','FontSize',18)
```



3.3: Physical parameter estimation

```
% Obtain DC gain
gain_dc = evalfr(G,1);

% Obtain approximate damped resonant frequency and peak gain
[gain_max,w_d] = getPeakGain(G);

% Estimate damping ratio based on normalized resonant peak gain
syms zeta; zeta = min( double( solve( (2*zeta*sqrt(1-zeta^2))^-1 == gain_max/
gain_dc, zeta ) ) );

% Estimate natural frequency
w_n = w_d/sqrt(1-zeta^2);

% Estimate spring constant (N/m) (made sure to convert cm to m)
k_hat = 1/(0.01*gain_dc)
```

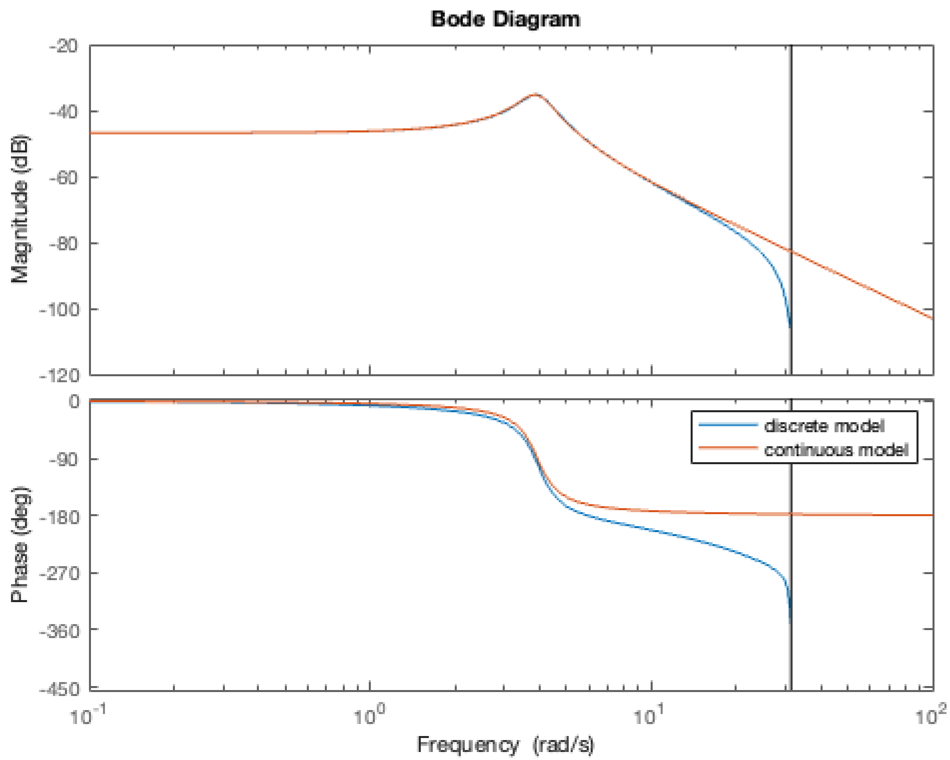
```
% Estimate mass (kg)
m_hat = k_hat/w_n^2

% Estimate damping coefficient (Ns/m)
b_hat = zeta*2*m_hat*w_n

% I would have preferred using this:
% Gc = d2c(G,'zoh')

% Compare frequency responses of each model
Gc = tf(1,[m_hat b_hat k_hat]);
bode(0.01*G)
hold on
bode(Gc)
hold off
shg
legend({'discrete model','continuous model'})
```

```
k_hat =
  219.2911
m_hat =
   14.2115
b_hat =
   14.7625
```



Published with MATLAB® R2024a