Table of Contents


```
% SD_Hw4_prob1 % reference: transfer_func_example
close all;clear;clc;
opengl hardware;
% parameters
m = 100;                % kg
wn = 2*pi*0.5;          % 2pi*fn
damping = 0.03;
k = wn^2*m;
c = 2*damping*wn*m;
```

(C) transfer function

```
%--------------------------------------------------------------------------
% Form transfer function
% Hup = 1/(ms^2+cs+k)
%--------------------------------------------------------------------------
num = 1;
param = [m c k];
Hup = tf(num,param);
display(Hup);
Hvp = tf([1 0],param);
display(Hvp);
```

(D) plot Impulse response function

```
%--------------------------------------------------------------------------
% Impulse response function
%--------------------------------------------------------------------------
t = (0:0.001:40)';          % better to use a column vector % 40sec
wd = wn*sqrt(1-damping^2);
hd = 1/m/wd*exp(-damping*wn*t).*sin(wd*t);
figure
plot(t,hd)
xlabel('time (sec)')
ylabel('displacement (m)')
title('(d) impulse response function')
```

(E) calculate impulse response function

MATLAB Function#(d)##

```
he = impulse(Hup,t);
figure
plot(t,hd,'r','linewidth', 2)
hold on
plot(t,he,'b--')
hold off
grid on
legend('(d) equation','(e) matlab')
xlabel('time (sec)')
ylabel('displacement (m)')
title('(e)')
```

(F) Step response function

```
%-------------------------------------------------------------------------
% Step response function
%-------------------------------------------------------------------------
p0 = 10;
u = p0/k*(1-exp(-damping*wn*t).*(cos(wd*t)+damping*wn/wd*sin(wd*t)));
v = p0/k*(damping*wn*exp(-damping*wn*t).*(cos(wd*t)+damping*wn/
wd*sin(wd*t))-...
 exp(-damping*wn*t).*(-wd*sin(wd*t)+damping*wn*cos(wd*t)));

figure
subplot(211);
plot(t,u)
xlabel('time (sec)')
ylabel('displacement (m)')
title('(f) step response function')
subplot(212);
plot(t,v)
xlabel('time (sec)')
ylabel('velocity (m/sec)')
```

(G) redo F using MATLAB

```
ug = p0*step(Hup,t);
vg = p0*step(Hvp,t);
figure()
subplot(211);
plot(t,u,'r-','linewidth',2)
hold on
plot(t,ug,'b--')
hold off
grid on
legend('(f) equation','(g) matlab')
xlabel('time (sec)')
ylabel('displacement (m)')
```

```
title('(g)');
subplot(212);
plot(t,v,'r-','linewidth',2)
hold on
plot(t,vg,'b--')
hold off
grid on
legend('(f) equation','(g) matlab')
xlabel('time (sec)');
ylabel('velocity (m/sec)');
```

(H) divide velocity response in (G) by 10 and compare with it

```
figure
plot(t,he,'r-','linewidth',2)
hold on
plot(t,vg/10,'b--')
hold off
grid on
title('(h)');
legend('impulse response in (e)','velocity response in (g)/10')
xlabel('time (sec)')
ylabel('response')
% The derivative of the step response function is identical to the impulse
 response function.
```

(I) Complex frequency response functions

or transfer functions in the complex domain

```
%--------------------------------------------------------------------------
% Transfer function
%--------------------------------------------------------------------------
ff = (0:0.01:10)'; % better to use a column vector

Hup_c = (1/m)./(wn^2-(2*pi*ff).^2+sqrt(-1)*2*damping*wn*(2*pi*ff));
Hvp_c = (sqrt(-1)*2*pi*ff/m)./(wn^2-
(2*pi*ff).^2+sqrt(-1)*2*damping*wn*(2*pi*ff));
figure
subplot(221);
plot(ff,db(abs(Hup_c)))
xlabel('frequency (Hz)')
ylabel('magnitude of displacement (dB)')

subplot(222);
plot(ff,db(abs(Hvp_c)))
xlabel('frequency (Hz)')
ylabel('magnitude of velocity (dB)')

subplot(223);
plot(ff,mod(angle(Hup_c)*180/pi+180,360)-180)
```

```
xlabel('frequency (Hz)')
ylabel('phase of displacement (deg)')

subplot(224);
plot(ff,mod(angle(Hvp_c)*180/pi+180,360)-180)
xlabel('frequency (Hz)')
ylabel('phase of velocity (deg)')

sgtitle('(i)');
% dB is one kind of log scale --> google it or type "help db" in command
 window
```

(J) Plot the magnitude and phase

```
Hup_m = squeeze(freqresp(Hup,2*pi*ff));
Hvp_m = squeeze(freqresp(Hvp,2*pi*ff));

figure
subplot(221);
plot(ff,db(abs(Hup_c)),'r-','linewidth',2)
hold on
plot(ff,db(abs(Hup_m)),'b--')
hold off
legend('equation','matlab')
xlabel('frequency (Hz)')
ylabel('magnitude of displacement (dB)')

subplot(222);
plot(ff,db(abs(Hvp_c)),'r-','linewidth',2)
hold on
plot(ff,db(abs(Hvp_m)),'b--')
hold off
legend('equation','matlab')
xlabel('frequency (Hz)')
ylabel('magnitude of velocity (dB)')

subplot(223);
plot(ff,mod(angle(Hup_c)*180/pi+180,360)-180,'r-','linewidth',2)
hold on
plot(ff,mod(angle(Hup_m)*180/pi+180,360)-180,'b--')
hold off
legend('equation','matlab')
xlabel('frequency (Hz)')
ylabel('phase of displacement (deg)')

subplot(224);
plot(ff,mod(angle(Hvp_c)*180/pi+180,360)-180,'r-','linewidth',2)
hold on
plot(ff,mod(angle(Hvp_m)*180/pi+180,360)-180,'b--')
hold off
legend('equation','matlab')
xlabel('frequency (Hz)')
ylabel('phase of velocity (deg)')
```

```
sgtitle('(j)');
```

(K)(L) state-space model

reference: state_space_example

```
%-----------------------------------------------------------------------------
% sate-space model
%-----------------------------------------------------------------------------
Ac = [0 1;-k/m -c/m];
Bc = [0;1/m];
Cc = [eye(2);[-k/m -c/m]];
Dc = [0;0;1/m];
%==
sys = ss(Ac,Bc,Cc,Dc)
Hup_ss = squeeze(freqresp(tf(sys(1,:)),2*pi*ff));
Hvp_ss = squeeze(freqresp(tf(sys(2,:)),2*pi*ff));

figure()
subplot(221);
plot(ff,db(abs(Hup_ss)),'r-','linewidth',2)
hold on
plot(ff,db(abs(Hup_m)),'b--')
hold off
legend('(l)','(j)')
xlabel('frequency (Hz)')
ylabel('magnitude of displacement (dB)')

subplot(222);
plot(ff,db(abs(Hvp_ss)),'r-','linewidth',2)
hold on
plot(ff,db(abs(Hvp_m)),'b--')
hold off
legend('(l)','(j)')
xlabel('frequency (Hz)')
ylabel('magnitude of velocity (dB)')

subplot(223);
plot(ff,mod(angle(Hup_ss)*180/pi+180,360)-180,'r-','linewidth',2)
hold on
plot(ff,mod(angle(Hup_m)*180/pi+180,360)-180,'b--')
hold off
legend('(l)','(j)')
xlabel('frequency (Hz)')
ylabel('phase of displacement (deg)')

subplot(224);
plot(ff,mod(angle(Hvp_ss)*180/pi+180,360)-180,'r-','linewidth',2)
hold on
plot(ff,mod(angle(Hvp_m)*180/pi+180,360)-180,'b--')
hold off
legend('(l)','(j)')
```

```
xlabel('frequency (Hz)')
ylabel('phase of velocity (deg)')

sgtitle('(l)');
```

Published with MATLAB® R2022a