Michael Gentry
ECGR 2254
Project 2 Report

Project 2 Report

Problem 1:

a)

a. Based on the graph of $n(t)$, the period was $1 * 10^{-4}s$.

b.

n	α_n
-3	$0.045e^{-j1.67e-14°}$
-2	$4.56e - 017 * e^{j132.015°}$
-1	$0.405e^{-j2.57e-015}$
1	$0.405e^{j2.57e-015}$
2	$4.56e - 017 * e^{-j132.015°}$
3	$0.045e^{-j1.67e-14°}$

To get α_n is the form of $c_n cos(n\omega_0 t + \phi_n)$, I had to double the magnitude of each α_n, and find the ϕ_n and $n\omega_0$ for each instance of n.

n	$c_n cos(n\omega_0 t + \phi_n)$
1	$0.81cos(2\pi 10000t + 2.57e - 15°)$
2	$9.12 * 10^{-17} cos(2\pi 20000t - 132.015°)$
3	$0.0901cos(2\pi 30000t + 1.67e - 14°)$

b)

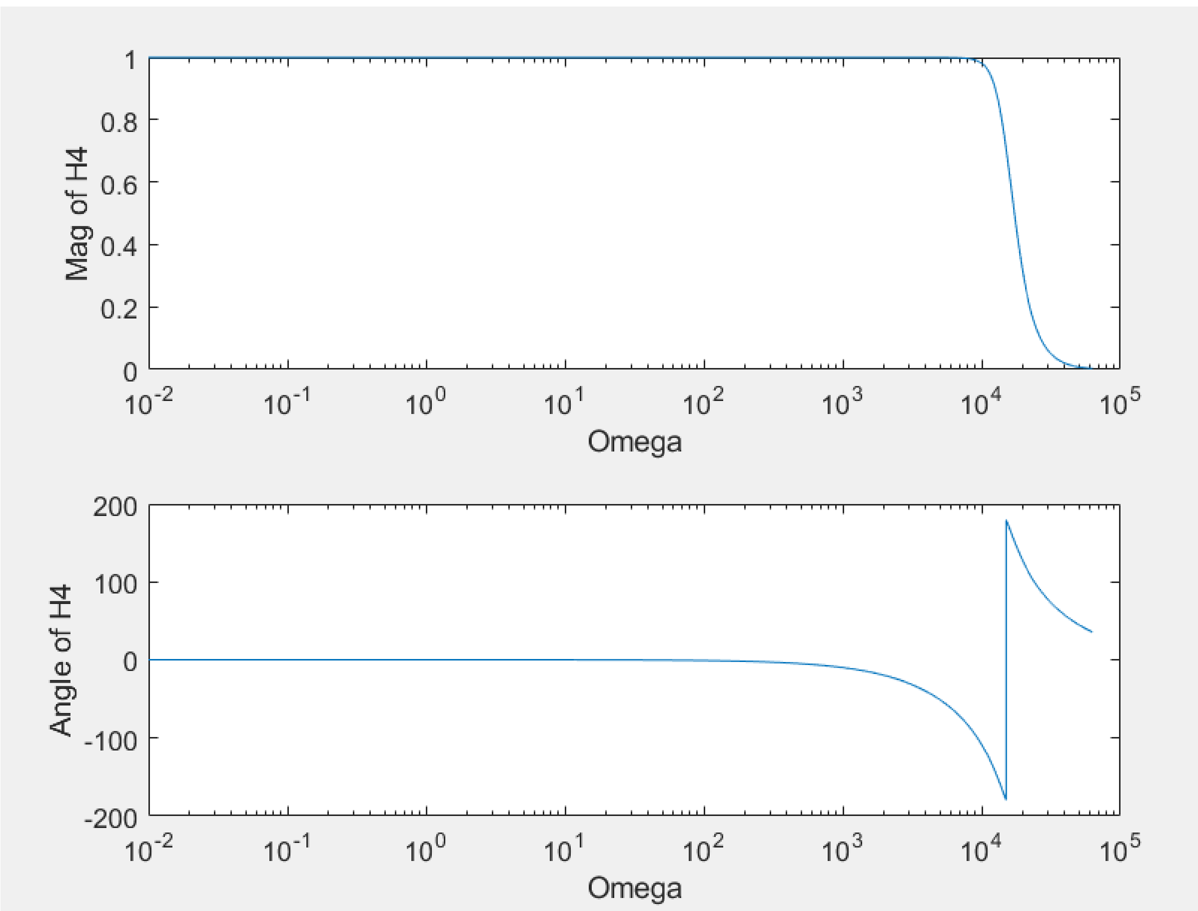
To find the appropriate filter for this problem, the filter had to meet 2 conditions. The goal of the filter was to reduce the the fundamental component of $n(t)$ to less than 1% of its original magnitude, while not significantly impacting the 60 Hz component of the input. This means that when $\omega = 2\pi 10000$, the magnitude of $H(\omega)$ should be no more than 0.01, and the phase of

$H(\omega)$ at $\omega = 2\pi 60$ should be no more than 4° and no less than -4°. To check this, I would check the magnitude and phase of filters 2 and 4 at different values of ω_c to see if they met the requirements.

ω_c	$\lvert H_2(2\pi 1000$	$\lvert H_4(2\pi 1000$	$\angle H_2(2\pi 60)$	$\angle H_4(2\pi 60)$	$\lvert H_2(2\pi 60)\rvert$	$\lvert H_4(2\pi 60)\rvert$
50	$6.3e-07$	$4.3e-13$	$-170°$	20°	0.0176	3.0942e-04
100	2.53e-06	6e-12	$-160°$	40°	0.0702	0.005
1000	2.5330e-06	6.4162e-12	-158.0247	40.1198	0.9901	0.9998
10000	0.0253	6.4162e-04	-3.05	-5.65	1	1
15000	0.057	0.0032	-2.04	-3.76	1	1

After testing multiple values for ω_c, I found that only H_4 was able to meet both conditions, and that it would meet the conditions at $\omega_c = 15000$

Plot of $\lvert H_4(2\pi 10000)\rvert$ and $\angle H_4(2\pi 60)$ at $\omega_c = 15000$:



After finding a filter that fit the specifications required, I made a vector containing the α_n values for $n = 1 - 3$ and then did a fourier transform for the $2\boldsymbol{\pi}60$ term as follows:

$2cos(2\pi 60) = \boldsymbol{\delta}(\omega - 2\pi 60) + \boldsymbol{\delta}(\omega + 2\pi 60)$

This mean that the fourier transform for the positive $2\boldsymbol{\pi}60$ has a magnitude of 1 and phase of 0. I then created a vector containing the transfer function values for those same ω values. I then multiplied the corresponding values in those 2 vector like so:

$V_{out}(\omega) = V_{in}(\omega)H(\omega)$

I then used the phase and magnitude of $V_{out}(\omega)$ to get the values in the form of $c_n cos(n\omega_0 t + \phi_n)$.

Harmonic	Output
$2\pi 60$	$2cos(2\pi 60t - 3.76°)$
$2\pi 10000(n = 1)$	$0.0026cos(2\pi 10000t + 36.04°)$
$2\pi 20000(n = 2)$	$1.8516e - 20cos(2\pi 20000t - 114.1074°)$
$2\pi 30000(n = 3)$	$3.6117e - 06cos(2\pi 30000t + 11.92°)$

Code from Problem 1:

```
clear all
%freq, period, time vector
n = 2;%which harmonic
omega_0 = 2*pi*10000;%fund harmonic
T = 10^(-4);%period
t = [-T/2:T/10000:T/2];%period interval


%model of n(t)
x = sawtooth(omega_0*t+pi,1/2);%creates n(t)
%plot(t,x)
%grid on

%fourier transform of n(t)
a = -j * n * omega_0;%change of variable
X = x.*exp(a.*t);%function to be integrated
alpha_n = (1/T) * trapz(t,X);%complex coeff of fourier transform
abs(alpha_n)


%find vals for c_n, phi_n
c_mag = 2 * abs(alpha_n);%magnitude
c_phase = angle(alpha_n)*180/pi;%phase shift
c_freq = n * omega_0/(2*pi);


%transfer functions
omega_plot = [0:0.01:omega_0];
omega_c = 15000;
s = j*omega_plot;
H2 = (omega_c^2)./(s.^2 + (2/sqrt(2))*s*omega_c + omega_c^2);
H4 = (omega_c^4)./((s.^2 + s.*0.7654*omega_c + omega_c^2).*(s.^2 + s.*1.8478*omega_c +
omega_c^2));

%test mag and phase of filters w/ different omega_c
figure(2);
subplot(2,1,1);
semilogx(omega_plot,abs(H4))%check mag at 2*pi*10000
xlabel('Omega');
```

```
ylabel('Mag of H4');
subplot(2,1,2)
semilogx(omega_plot,angle(H4)*180/pi)%check phase at 2*pi*60
xlabel('Omega');
ylabel('Angle of H4');



%H4 vector for different omega values
omega = [2*pi*60 omega_0 2*omega_0 3*omega_0];
H4_vec = [];
V_in = [];
for k = 1:1:length(omega)
   a2 = -j * omega(k);
   s2 = j * omega(k);
   if omega(k) == 2*pi*60
      V_in(k) = exp(-j*0);
   else
      T = 10^(-4);
      t = [-T/2:T/10000:T/2];
      X2 = x.*exp(a2.*t);
      alpha_n2 = (1/T) * trapz(t,X2);%complex coeff of fourier transform
      V_in(k) = alpha_n2;
   end
   H4_vec = [H4_vec (omega_c^4)/((s2^2 + s2*0.7654*omega_c + omega_c^2)*(s2^2 +
s2*1.8478*omega_c + omega_c^2))];
end
check = abs(V_in(1))


%V_out Vector
V_out = [];
for q = 1:1:length(omega)
   V_out = [V_out H4_vec(q)*V_in(q)];
end

V_out_60Hz = 2 * abs(V_out(1))*cos(omega(1)*t + angle(V_out(1)));
V_outHarm1 = 2 * abs(V_out(2))*cos(omega(2)*t + angle(V_out(2)));
V_outHarm2 = 2 * abs(V_out(3))*cos(omega(3)*t + angle(V_out(3)));
V_outHarm3 = 2 * abs(V_out(4))*cos(omega(4)*t + angle(V_out(4)));
```

Problem 2:

a.

$s = j\omega$

Apply KCL on circuit:

$$-v_a(t) + Ri_a(t) + L\frac{di_a}{dt} + K\Omega(t) = 0$$

Isolate terms with $i_a(t)$:

$$L\frac{di_a}{dt} + Ri_a(t) = v_a(t) - K\Omega(t)$$

$$\frac{di_a}{dt} = s * i_a(t)$$

$$i_a(sL + R) = v_a - K\Omega$$

$$i_a = \frac{v_a - K\Omega}{(sL+R)}$$

b.

Given: $J\frac{d\Omega}{dt} = Ki_a - \beta\Omega$

Plug in equation for i_a *from part a:*

$$J\frac{d\Omega}{dt} = K\frac{v_a - K\Omega}{(sL+R)} - \beta\Omega$$

$$\frac{d\Omega}{dt} = s * \Omega$$

$$J * s * \Omega = K\frac{v_a - K\Omega}{(sL+R)} - \beta\Omega$$

move $\beta\Omega$ *term to other side of equation, factor out* $-\frac{K^2\Omega}{(sL+R)}$ *from rhs and move to lhs*

$$Js\Omega + \beta\Omega + \frac{K^2\Omega}{(sL+R)} = \frac{Kv_a}{(sL+R)}$$

Factor Ω *out from Lhs and divide by* $Js + \beta + \frac{K^2}{(sL+R)}$

$$\Omega = \frac{\frac{Kv_a}{(sL+R)}}{Js + \beta + \frac{K^2}{(sL+R)}}$$

Multiply LHS by $\frac{(sL+R)}{(sL+R)}$

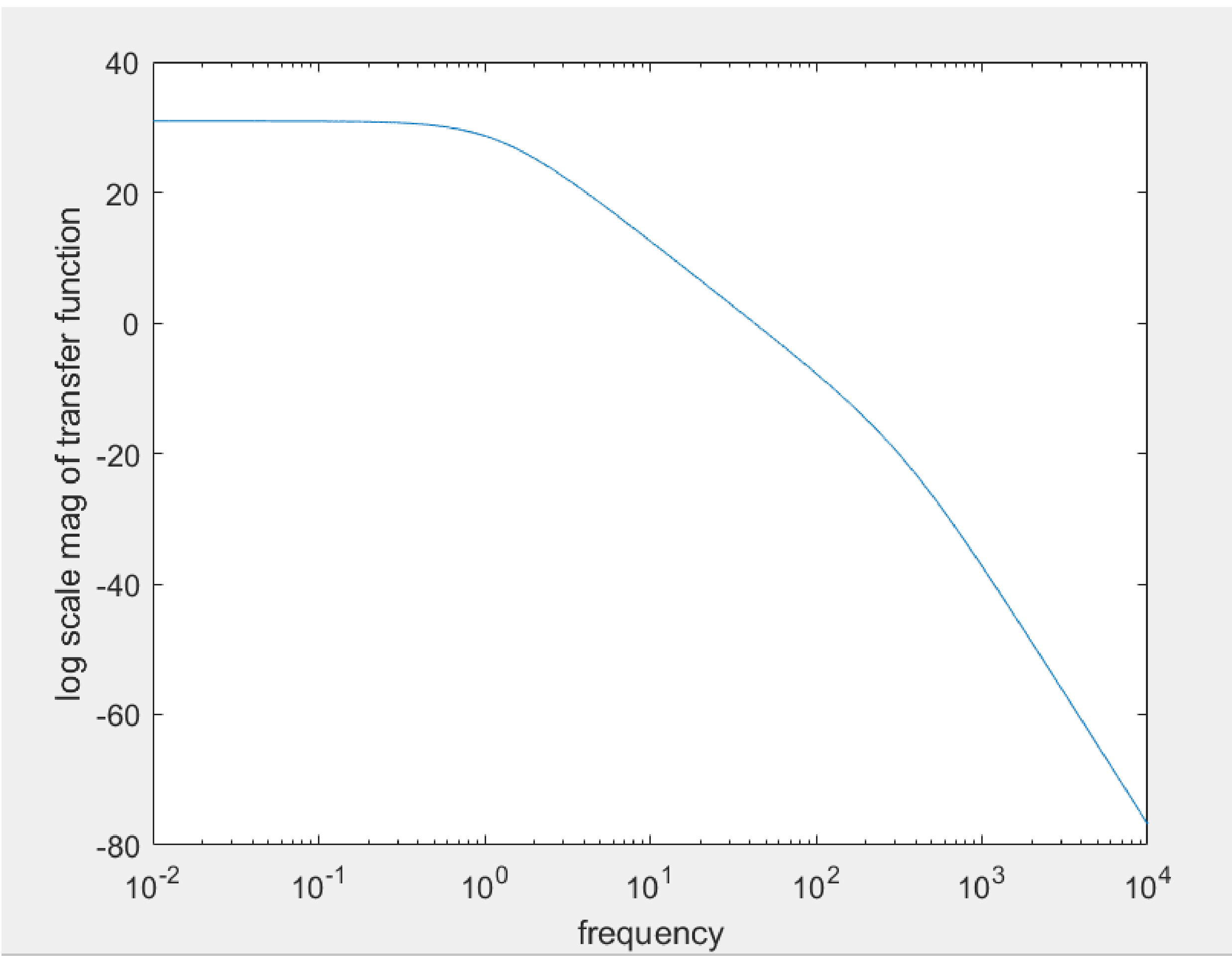
$$\Omega = \frac{Kv_a}{Js(sL+R) + \beta(sL+R) + K^2}$$

$$H(s) = \frac{K}{Js(sL+R) + \beta(sL+R) + K^2}$$

c. Code from c:

```
beta = 0.5 * 10^(-5);%damping ratio
J = 2 * 10^(-4);%moment of inertia
K = 0.029;%motor constant
L = 0.01;%inductor
R = 3.38;%ohms
y = logspace(-2,4,10000);
s = j * y;
H = K ./ (K^2 - beta .*(R + s.*L)- J.*s.*(R+s.*L));
Y = 20 * log10(abs(H));
semilogx(y,Y);
```

Graph:



d.

To find a switching frequency that would result in 1% of the Dc value, I first had to find the Dc value of the transfer function using the following code:

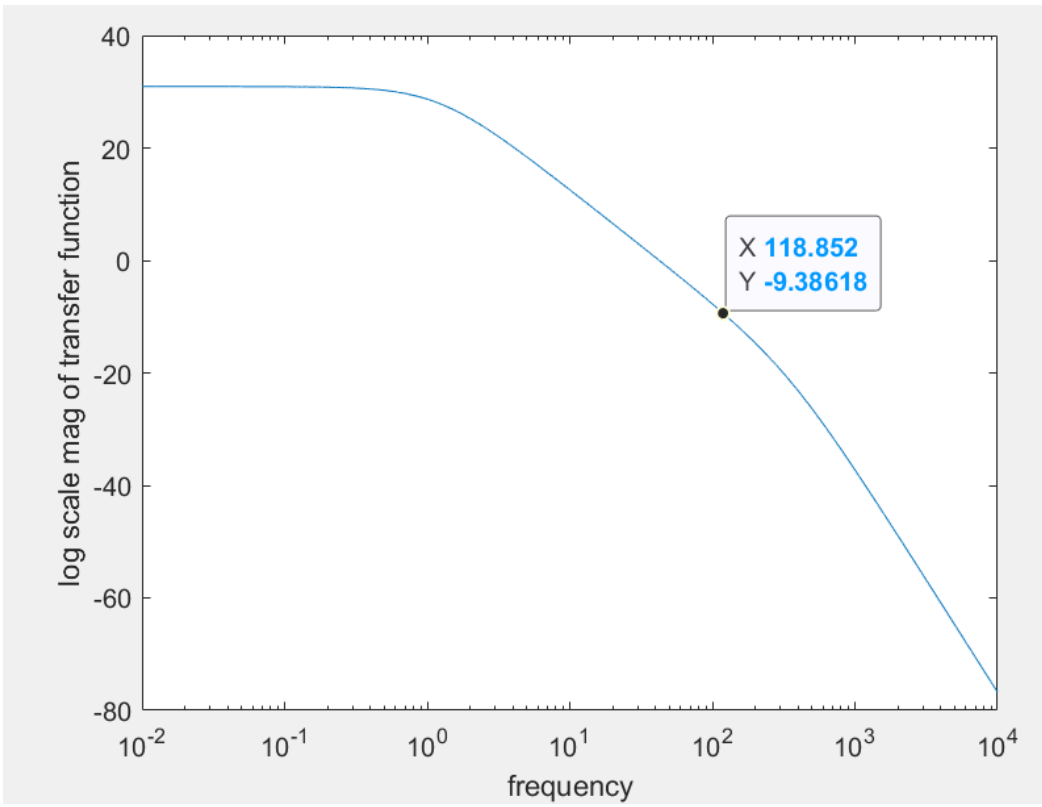
s_dc = 0;

*H_dc = K / (K^2 + beta *(R + s_dc*L)+ J*s_dc*(R+s_dc*L));%33.8035*

This resulted in a Dc value of approximately 33.8035. This means that the goal was for the transfer function to have 1% the magnitude of that, but since the bode plot was on a logarithmic scale, I had to use the following code to find my target magnitude:

*q = 20 * log10(H_dc/100);%target for switching freq = -9.42*

I then had to search the plot from part c to find the frequency where the output of the plot was close to -9.42



To find the DC voltage required for the motor to turn 324 rad/s, I had to use the DC value of the transfer function from earlier.

$V_{out}(\omega) = 324\, rad/s$

$V_{in}(\omega) * H(\omega) = 324$

$V_{in}(\omega) = \frac{324}{H(\omega)}$

I then used the following code to find that the required DC input was approximately 9.583 volts.

V_in_324 = 324 / H_dc;% = 9.5828

e.

$DC = \frac{1}{T}\int x(t)dt\ over\ one\ period$

$x(t) = 12,\ 0 < t < DT,\ x(t) = 0,\ DT < t < T$

$\rightarrow DC = \frac{1}{T}\int 12\ from\ 0\ to\ DT$

$\rightarrow DC = \frac{1}{T}(12DT - 0)$

$\rightarrow DC = 12D$

$\rightarrow 12D = 9.9.583$

$\rightarrow D = 9.21/12 = 0.7987$

f.

$\alpha_n = \frac{1}{T}\int x(t)e^{-jn\omega_0 t}dt,\ let\ a =- jn\omega_0$

$\rightarrow \alpha_n = \frac{1}{T}\int 12e^{at}dt,\ evaluated\ from\ 0\ to\ DT$

$\rightarrow \alpha_n = \frac{12}{aT}(e^{at}),\ 0\ to\ DT$

$\alpha_n = \frac{12}{aT}(e^{aDT} - 1)$

$Using\ this\ formula\ for\ \alpha_n,\ I\ used\ the\ following\ code\ to\ predict\ the\ speed\ variation\ of\ \Omega(t)$:

```
D_f = 0.5;
a_f = -j * omega_switch;
a_f_pos = j*omega_switch;
alpha_f = 12/(a_f*T) * (exp(a_f*D_f*T)-1);
H_switch =  K / (K^2 - beta *(R + a_f_pos*L)- J*a_f_pos*(R+a_f_pos*L));
f_out = alpha_f * H_switch;
c1_mag = 2 * abs(f_out);%2.5652, speed variation
```

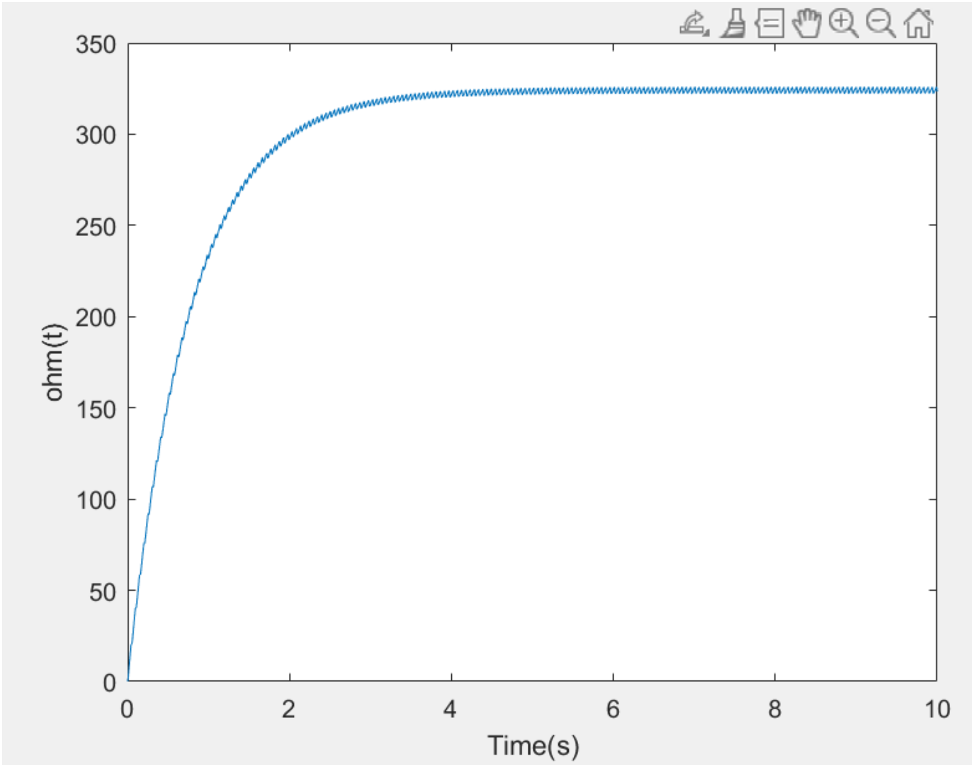
This code resulted in a c1 value of approximately 2.5652, which led me to believe the speed variation would be double that, at a value of 5.13.

g.

Code for square command, t vector, and simulation:

```
        %part g
delta_t = T/10000;
t = [0:T/10000:10];
Va = 6*square(omega_switch*t,D*100) + 6;%Va(t) for 10 seconds
%plot(t,Va)
i_a = zeros(size(t));
ohm = zeros(size(t));
alpha_g = 12/(T*a_f) * (exp(a_f*D*T)-1);
g_out = alpha_g * H_switch;



for i = 1:1:length(t)-1
  if i == 1
    i_a(1) = 0;
    ohm(1) = 0;
    i_a(i+1) = delta_t * (Va(i)/L - K*ohm(i)/L - R/L * i_a(i)) + i_a(i);
    ohm(i+1) = delta_t * (K/J * i_a(i) - beta/J * ohm(i)) + ohm(i);
  else
    i_a(i+1) = delta_t * (Va(i)/L - K*ohm(i)/L - R/L * i_a(i)) + i_a(i);
    ohm(i+1) = delta_t * (K/J * i_a(i) - beta/J * ohm(i)) + ohm(i);
  end
end
plot(t,ohm)
xlabel('Time(s)')
ylabel('ohm(t)')
```

Graph of ohm(t):



In steady state, the graph of ohm(t) leveled off with an average value around 324.108, which was almost exactly the same as the expected value of 324, along with a peak to peak variance of approximately 3.6, which was a decent bit off of my expected value of approximately 5.1. The first reason for this is that the duty ratio was different between the approximation and the simulation. There was also a smaller delta_t value for the simulation, which would also throw the result off a little.

Code from g:

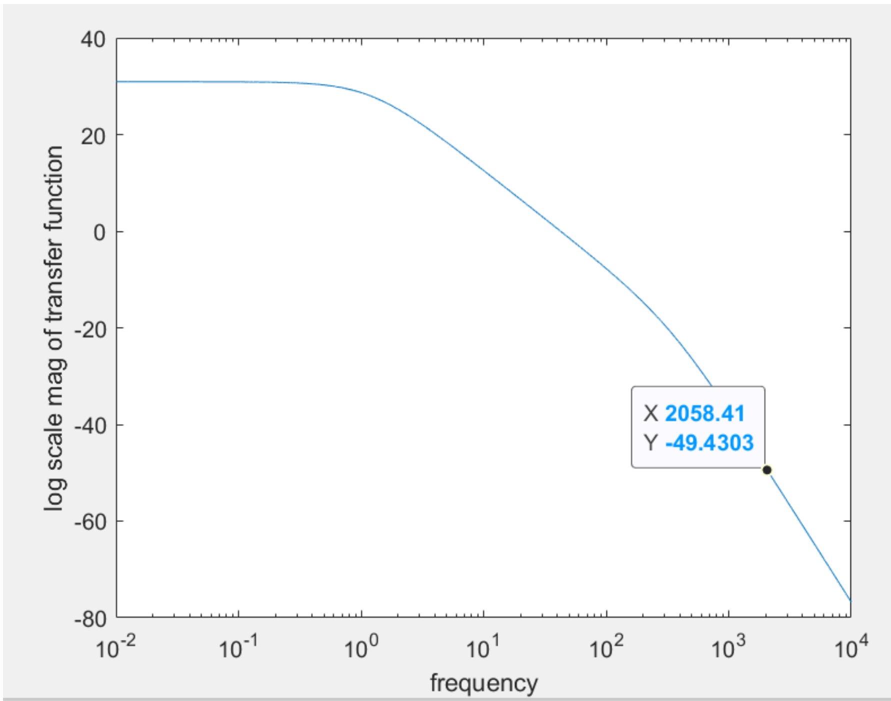
```
%part g
delta_t = T/10000;
t = [0:T/10000:10];
Va = 6*square(omega_switch*t,D*100) + 6;%Va(t) for 10 seconds
%plot(t,Va)
i_a = zeros(size(t));
ohm = zeros(size(t));
alpha_g = 12/(T*a_f) * (exp(a_f*D*T)-1);
g_out = alpha_g * H_switch;

for i = 1:1:length(t)-1
```

```
    if i == 1
        i_a(1) = 0;
        ohm(1) = 0;
        %numeric sols to ia and ohm diff eqs from earlier
        i_a(i+1) = delta_t * (Va(i)/L - K*ohm(i)/L - R/L * i_a(i)) + i_a(i);
        ohm(i+1) = delta_t * (K/J * i_a(i) - beta/J * ohm(i)) + ohm(i);
    else
        i_a(i+1) = delta_t * (Va(i)/L - K*ohm(i)/L - R/L * i_a(i)) + i_a(i);
        ohm(i+1) = delta_t * (K/J * i_a(i) - beta/J * ohm(i)) + ohm(i);
    end
end
plot(t,ohm)
xlabel('Time(s)')
ylabel('ohm(t)')
```

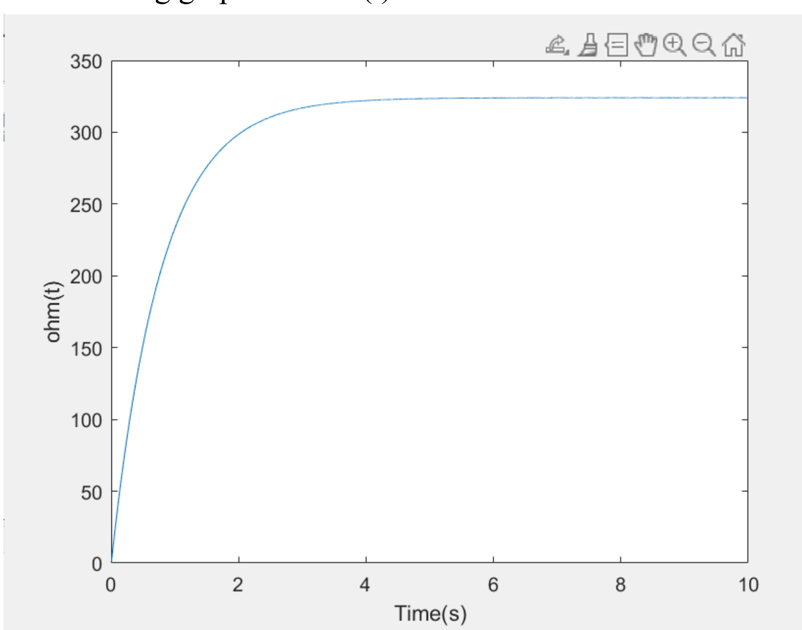
h.

To find the frequency at which the value would be 0.01% of the dc term, I first had to use the following code to find the y-value that would correspond to such a frequency:
*goal = 20*log10(H_dc/10000);%-49.42, find freq at this val on bode plot*

I then traced this y-value on the plot from part c to fund the right switching frequency:



After this, using the same process as from part h with my new switching frequency, I got the following graph for ohm(t):



Full code from problem 2:

```
clear all
beta = 0.5 * 10^(-5);%damping ratio
J = 2 * 10^(-4);%moment on inertia
K = 0.029;%motor constant
L = 0.01;%inductor
R = 3.38;%ohms
y = logspace(-2,4,10000);
s = j * y;
H = K ./ (K^2 + beta .*(R + s.*L)+ J.*s.*(R+s.*L));
Y = 20 * log10(abs(H));
semilogx(y,Y);
xlabel('frequency')
ylabel('log scale mag of transfer function')



%dc term = 35.2
%analyze bode plot to find freq where output is 1% of dc term
% find freq where output <= .35
omega_switch = 120;
f_switch = omega_switch / (2*pi);


%dc freq = 0
s_dc = 0;
H_dc = K / (K^2 + beta *(R + s_dc*L)+ J*s_dc*(R+s_dc*L));%33.8035
q = 20 * log10(H_dc/100);%target for switching freq = -9.42
%dc output = v_in * H_dc, v_in = dc_out/H_dc
%target output = 324
T = 2*pi/omega_switch;%0.0546
V_in_324 = 324 / H_dc% = 9.2072
D = V_in_324 / 12 %0.7987

%part f
D_f = 0.5;
a_f = -j * omega_switch;
a_f_pos = j*omega_switch;
alpha_f = 12/(a_f*T) * (exp(a_f*D_f*T)-1);
H_switch =  K / (K^2 + beta *(R + a_f_pos*L)+ J*a_f_pos*(R+a_f_pos*L));
```

```
f_out = alpha_f * H_switch;
c1_mag = 4 * abs(f_out)%2.5652, speed variation
c1_phase = angle(f_out);%left in rad form, = -8.562 deg
%f_signal = c1_mag * cos(omega_switch)

%part g
delta_t = T/10000;
t = [0:T/10000:10];
Va = 6*square(omega_switch*t,D*100) + 6;%Va(t) for 10 seconds
%plot(t,Va)
i_a = zeros(size(t));
ohm = zeros(size(t));
alpha_g = 12/(T*a_f) * (exp(a_f*D*T)-1);
g_out = alpha_g * H_switch;

for i = 1:1:length(t)-1
    if i == 1
        i_a(1) = 0;
        ohm(1) = 0;
        %numeric sols to ia and ohm diff eqs from earlier
        i_a(i+1) = delta_t * (Va(i)/L - K*ohm(i)/L - R/L * i_a(i)) + i_a(i);
        ohm(i+1) = delta_t * (K/J * i_a(i) - beta/J * ohm(i)) + ohm(i);
    else
        i_a(i+1) = delta_t * (Va(i)/L - K*ohm(i)/L - R/L * i_a(i)) + i_a(i);
        ohm(i+1) = delta_t * (K/J * i_a(i) - beta/J * ohm(i)) + ohm(i);
    end
end
plot(t,ohm)
xlabel('Time(s)')
ylabel('ohm(t)')

%part h
goal = 20*log10(H_dc/10000);%-49.42, find freq at this val on bode plot
f_switch2 = 2060;
a_h = -j*f_switch2;
T2 = 2*pi/f_switch2;
delta_t2 = T2/10000;
t2 = [0:T2/10000:10];
Va2 = 6*square(f_switch2*t2,D*100) + 6;%Va(t) for 10 seconds
%plot(t,Va)
```

```
i_a2 = zeros(size(t));
ohm2 = zeros(size(t));

for k = 1:1:length(t2)-1
    if k == 1
        i_a2(1) = 0;
        ohm2(1) = 0;
        i_a2(k+1) = delta_t2 * (Va2(k)/L - K*ohm2(k)/L - R/L * i_a2(k)) + i_a2(k);
        ohm2(k+1) = delta_t2 * (K/J * i_a2(k) - beta/J * ohm2(k)) + ohm2(k);
    else
        i_a2(k+1) = delta_t2 * (Va2(k)/L - K*ohm2(k)/L - R/L * i_a2(k)) + i_a2(k);
        ohm2(k+1) = delta_t2 * (K/J * i_a2(k) - beta/J * ohm2(k)) + ohm2(k);
    end
end
plot(t2,ohm2)
xlabel('Time(s)')
ylabel('ohm(t)')
```