Master of Science in

Engineering

Power Electronic Systems

Attestation

Author
Ervin Mazlagić

Professor
Prof. Dr. Kurt Schenk

Place and date of admission
Altishofen, 2017-11-17

Abstract

This report shows the individual solution of exercise 5.5 from the module Power Electronic Systems (TSM_PowElSys) which is neccessary for admission to the final exam.

The report shows how a controller can be designed for a simple boost converter with ideal components. The power stage transfer function is developed with the state space averaging method.

Contents

1 Exercise A – Steady State Duty Ratio **2**
- 1.1 First Interval . . . 2
- 1.2 Second Interval . . . 2
- 1.3 Volt-Second Balance . . . 3
- 1.4 Numerical Results . . . 3
  - 1.4.1 By Hand . . . 3
  - 1.4.2 With GNU Octave . . . 4

2 Exercise B – Power Stage Transfer Function **5**
- 2.1 Differential Equations . . . 5
  - 2.1.1 First Interval . . . 5
  - 2.1.2 Second Interval . . . 5
- 2.2 Complete System Equation . . . 5
- 2.3 Steady State Equation . . . 6
- 2.4 Small Signal Pertubation . . . 6
- 2.5 Bode-Plots . . . 7

3 Exercise C – Controller Transfer Function **8**
- 3.1 Available Controller . . . 8
  - 3.1.1 PI . . . 8
  - 3.1.2 PIT1 . . . 8
  - 3.1.3 PIDT2 . . . 8

4 Exercise D – Open Loop Transfer Function **9**
- 4.1 Bode-Plots . . . 9

5 GNU Octave Code **10**

1 Exercise A – Steady State Duty Ratio

To calculate the duty cycle we can use the volt-second balance. To do so, we have to evaluate the proper mathematical formulation for the active (first) and passive (second) interval.

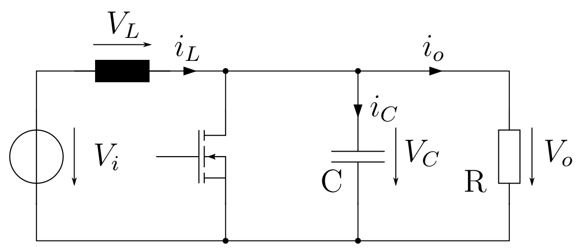


Figure 1: *Complete booster circuit.*

1.1 First Interval

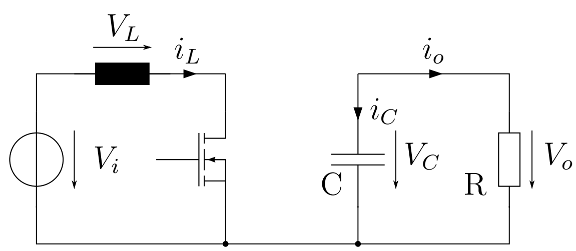


Figure 2: *Equivalent booster circuit for the first interval.*

In the active interval the MOSFET is on. Since the inductance is connected to the negative terminal of the source the voltage across the inductance L_1 is the input voltage V_i.

$$V_{L_1} = V_i$$

1.2 Second Interval

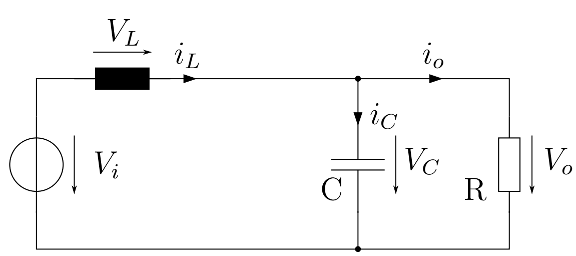


Figure 3: *Equivalent booster circuit for the second interval.*

In the passive interval the diode D is conducting the inductor current to the load and the output capacitor. The capacitor is assumed to be ideal, hence no output voltage ripple is assumed.

Therefore, the voltage loop is defined as

$$V_i = V_{L_2} + V_o$$

Using this loop we can obtain the inductors voltage for the passive interval as

$$V_{L_2} = V_i - V_o$$

1.3 Volt-Second Balance

To calculate the duty cycle we can use the general volt-second balance equation

$$V_{L_1} D + V_{L_2}(1 - D) = 0$$

and rearrange the equation so isolate the duty cycle D. Herefore we can expand the factors to

$$V_{L_1} D + V_{L_2} - V_{L_2} D = 0$$

and separate the sums

$$V_{L_1} D - V_{L_2} D = -V_{L_2}$$

and factorize for

$$D(V_{L_1} - V_{L_2}) = V_{L_2}$$

and finally isolate

$$D = \frac{-V_{L_2}}{V_{L_1} - V_{L_2}}$$

Using the previously obtained expressions for V_{L_1} and V_{L_2} we get

$$D = \frac{-(V_i - V_o)}{V_i - (V_i - V_o)}$$

Expanding this expression leads to

$$D = \frac{-V_i + V_o}{V_i - V_i + V_o} = \frac{-V_i + V_o}{V_o}$$

1.4 Numerical Results

1.4.1 By Hand

$$V_{i_1} = 50\,\mathrm{V} \Rightarrow D = \frac{-50\,\mathrm{V} + 200\,\mathrm{V}}{200\,\mathrm{V}} = 0.75$$
$$V_{i_1} = 100\,\mathrm{V} \Rightarrow D = \frac{-100\,\mathrm{V} + 200\,\mathrm{V}}{200\,\mathrm{V}} = 0.50$$
$$V_{i_1} = 150\,\mathrm{V} \Rightarrow D = \frac{-150\,\mathrm{V} + 200\,\mathrm{V}}{200\,\mathrm{V}} = 0.25$$

1.4.2 With GNU Octave

../../data/duty.csv

```
Duty Cycle Calculation Results:
        D(Vi = 50V)     = 0.75
        D(Vi = 100V)    = 0.50
        D(Vi = 100V)    = 0.25
---
```

2 Exercise B – Power Stage Transfer Function

2.1 Differential Equations

2.1.1 First Interval

$$V_L = L\frac{d\,i_L(t)}{d\,t}$$

$$\frac{d\,i_L(t)}{d\,t} = \frac{v_L(t)}{L} = \frac{v_i(t)}{L} \quad ,v_L(t) = v_i(t)$$

$$i_C(t) = C\frac{d\,v_C(t)}{d\,t}$$

$$\frac{d\,v_C(t)}{d\,t} = \frac{i_C(t)}{C} = \frac{-i_O(t)}{C} = \frac{-v_O(t)}{RC} = \frac{-v_C(t)}{RC}$$

$$x' = A_1 x + B_1$$

$$\begin{bmatrix} {i_L}' \\ {v_C}' \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{RC} \end{bmatrix} \cdot \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} \frac{v_i}{L} \\ 0 \end{bmatrix}$$

2.1.2 Second Interval

$$V_L = L\frac{d\,i_L(t)}{d\,t}$$

$$\frac{d\,i_L(t)}{d\,t} = \frac{v_L(t)}{L} = \frac{v_i(t) - v_o(t)}{L} = \frac{v_i(t) - v_c(t)}{L} \quad ,v_o(t) = v_c(t)$$

$$i_C(t) = C\frac{d\,v_C(t)}{d\,t}$$

$$\frac{d\,v_C(t)}{d\,t} = \frac{i_C(t)}{C} = \frac{i_L(t) - \frac{v_C(t)}{R}}{C} = \frac{i_L(t)}{C} - \frac{v_C(t)}{RC}$$

$$x' = A_2 x + B_2$$

$$\begin{bmatrix} {i_L}' \\ {v_C}' \end{bmatrix} = \begin{bmatrix} 0 & \frac{-1}{L} \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix} \cdot \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} \frac{v_i}{L} \\ 0 \end{bmatrix}$$

2.2 Complete System Equation

$$x' = \underbrace{(A_1 D + A_2 D')}_{A}\, x + \underbrace{B_1 D + B_2 D'}_{B} \quad ,D + D' = 1$$

$$A = \begin{bmatrix} 0 & 0 \\ 0 & \frac{-1}{RC} \end{bmatrix} \cdot D + \begin{bmatrix} 0 & \frac{-1}{L} \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix} \cdot D' = \textbf{[TODO -]} = \begin{bmatrix} 0 & \frac{-D'}{L} \\ \frac{D'}{C} & \frac{-1}{RC} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{v_i(t)}{L} \\ 0 \end{bmatrix} \cdot D + \begin{bmatrix} \frac{v_i(t)}{L} \\ 0 \end{bmatrix} \cdot D' = \textbf{[TODO -]} = \begin{bmatrix} \frac{v_i(t)}{L} \\ 0 \end{bmatrix}$$

2.3 Steady State Equation

$$x' = 0$$
$$0 = AX + B$$
$$-AX = B$$
$$X = -\frac{1}{A}B$$
$$X = -A^{-1}B$$

$$X = \begin{bmatrix} I_L \\ V_C \end{bmatrix} = \begin{bmatrix} \frac{v_i(t)}{D'^2R} \\ \frac{v_i(t)}{D'} \end{bmatrix}$$

$$V_O = FX$$
$$V_O = \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{v_i(t)}{D'^2R} \\ \frac{v_i(t)}{D'} \end{bmatrix}$$
$$I_L = \frac{V_i}{D'^2R} = \frac{V_i}{(1-D)^2R}$$

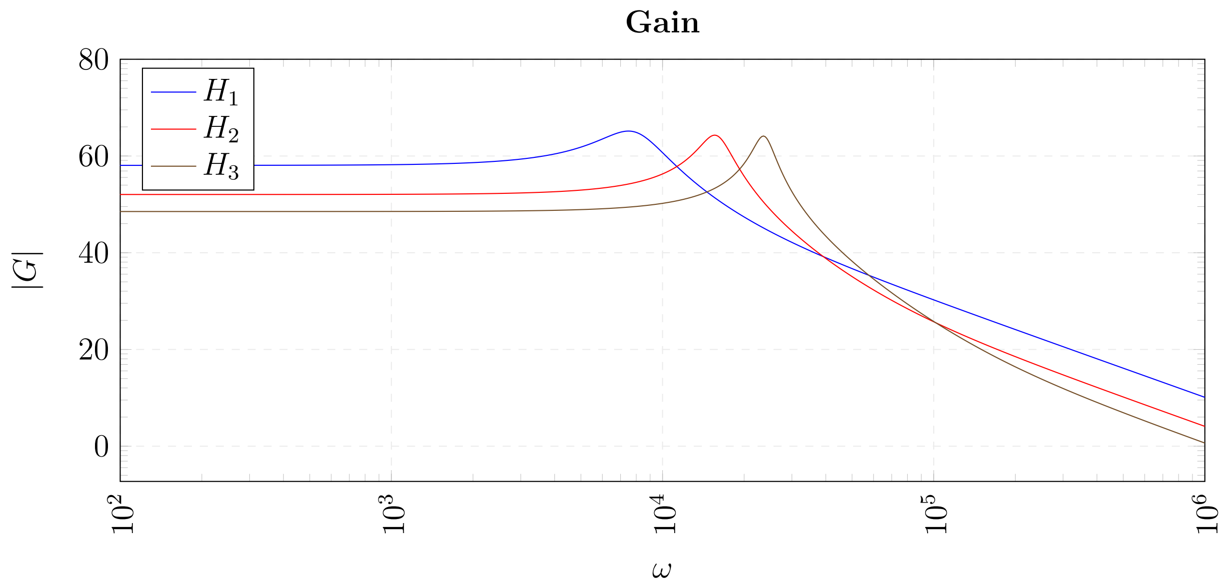
2.4 Small Signal Pertubation

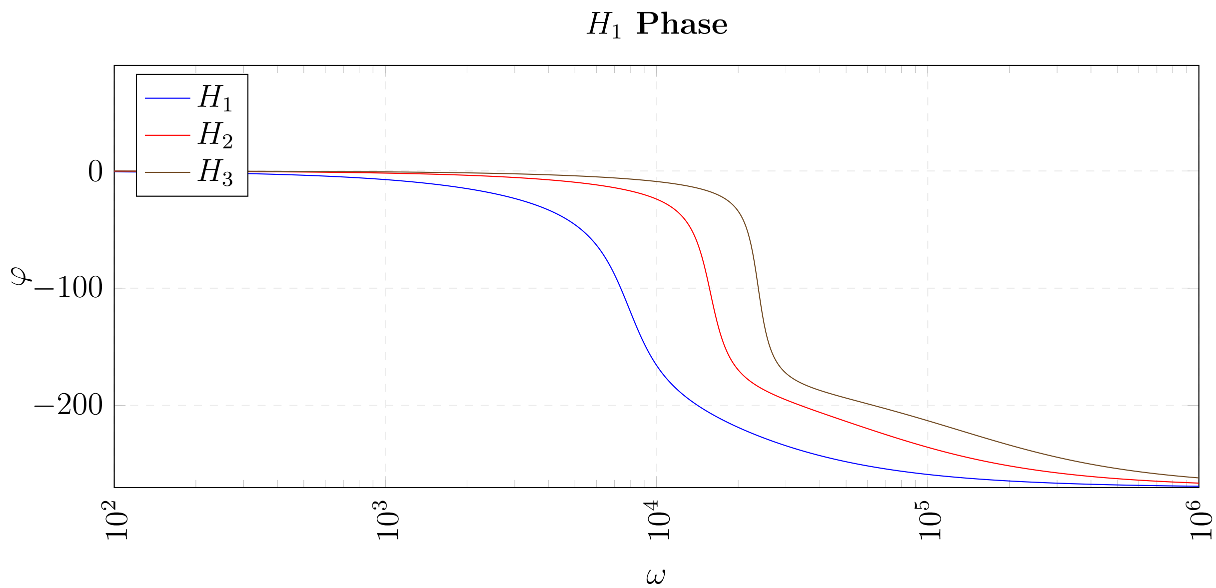
$$E = (A_1 - A_2)X + B1 - B2$$
$$E = \begin{bmatrix} 0 & \frac{1}{L} \\ \frac{-1}{C} & 0 \end{bmatrix} \cdot \begin{bmatrix} I_L \\ V_C \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{V_i}{L} \\ 0 \end{bmatrix} - \begin{bmatrix} \frac{V_i}{L} \\ 0 \end{bmatrix}}_{0} \quad , B_1 = B_2$$

$$\hat{V}_O(s) = F\hat{X}(s)$$

$$H(s) = \frac{\hat{V}_O(s)}{\hat{D}(s)} = F(sI - A)^{-1}E$$
$$\frac{\hat{V}_O(s)}{\hat{D}(s)} = \begin{bmatrix} 0 & 1 \end{bmatrix} \left(s \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & \frac{-D'}{L} \\ \frac{D'}{C} & \frac{-1}{RC} \end{bmatrix} \right)^{-1} \begin{bmatrix} \frac{V_C}{L} \\ \frac{-I_L}{C} \end{bmatrix}$$
$$H(s) = \frac{V_O}{D'} \cdot \frac{1 - s\frac{L}{RD'^2}}{1 + s\frac{L}{RD'^2} + s^2\frac{LC}{D'^2}}$$

2.5 Bode-Plots





3 Exercise C – Controller Transfer Function

3.1 Available Controller

3.1.1 PI

3.1.2 PIT1

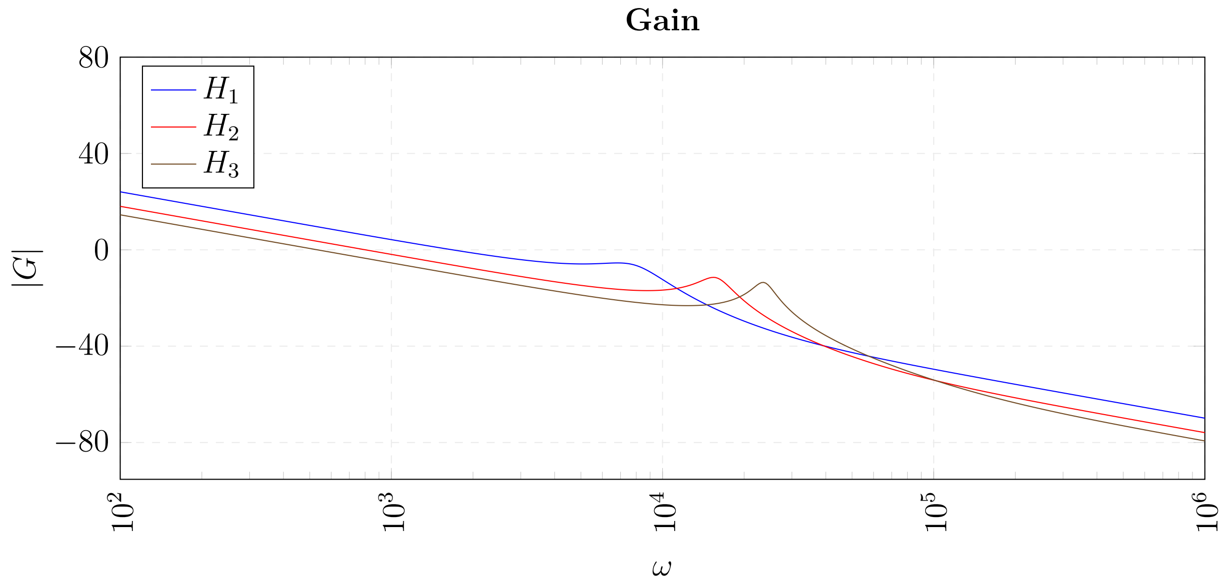
$$G = -K_P \frac{\left(1 + \frac{\omega_1}{s}\right)}{\left(1 + \frac{s}{\omega_T}\right)}$$

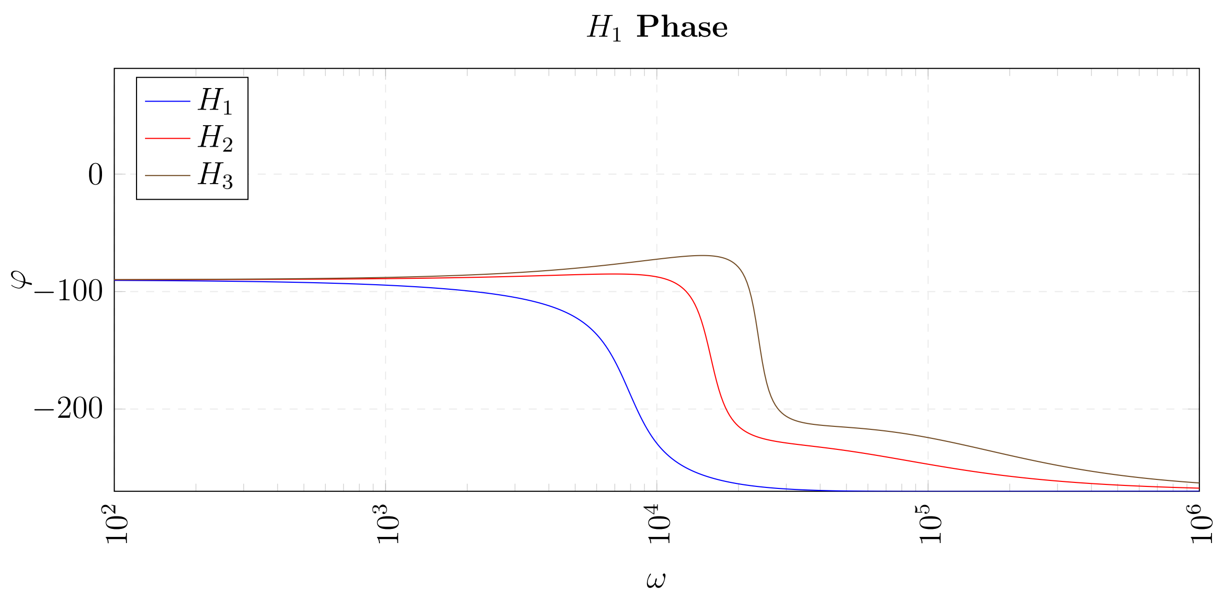
3.1.3 PIDT2

$$G = -K_P \frac{\left(1 + \frac{\omega_{i1}}{s}\right)\left(1 + \frac{\omega_s}{\omega_{i2}}\right)}{\left(1 + \frac{s}{\omega_{T2}}\right)\left(1 + \frac{s}{\omega_{T1}}\right)}$$

4 Exercise D – Open Loop Transfer Function

4.1 Bode-Plots





5 GNU Octave Code

../../sim/boostduty.m

```
function d = boostduty(vo, vi)
% BOOSTDUTY Calculate the duty cycle of a boost converter.
%
%   d = BOOSTDUTY(vo,vi)
%     Description:
%       Using absolute input and output voltages to calculate the duty cycle.
%     Output:
%       d   duty cycle as unitless factor [0..1]
%     Input:
%       vi  input voltage in [V]
%       vo  output voltage [V]
%
%   d = BOOSTDUTY(vo)     using relative output voltage (out / in)
%     Description:
%       Using relative output voltage to calculate the duty cycle.
%     Output:
%       d   duty cycle as unitless factor [0..1]
%     Input:
%       vo  relative output voltage (output / input) as unitless factor [1..n]

  switch nargin
    case 2
      d = (-vi + vo) / (vo) ;
    case 1
      d = (-1 + vo) / (vo);
    otherwise
      d = -1;
  endswitch
endfunction
```

../../sim/attestation.m

```
% cleanup ------------------------------------------------------------------
clear all;
close all;

% load packages ------------------------------------------------------------
pkg load control;
pkg load signal;

% setup graphics -----------------------------------------------------------
graphics_toolkit gnuplot
%graphics_toolkit qt

% definitions --------------------------------------------------------------
vo   = 200;     % [V]   output voltage
L    = 200E-6; % [H]   inductance
```

```
C   = 5E-6;    % [F]   output capacitance
R   = 50;      % [Ohm] load resistance
fs  = 100E3;   % [Hz]  switching frequency

% derived definitions
wlb  = 0;       % [1]  angular frequency: lower bound exponent (10^x)
wub  = 6;       % [1]  angular frequency: upper bound exponent (10^x)
wn   = 1E3;     % [1]  angular frequency: number of points
angf = logspace(wlb,wub,wn);

% controller options
ctrl_pi    = 1;   % PI controller
ctrl_pit1  = 2;   % PIT1 Controller
ctrl_pidt2 = 3;   % PIDT2 Controller

% controler selection
control = ctrl_pi;
%control = ctrl_pidt2;

% 1A ---------------------------------------------------------------------------
vi1 = 50;
vi2 = 100;
vi3 = 150;
vi  = [vi1, vi2, vi3];

d1 = boostduty(vo, vi1);
d2 = boostduty(vo, vi2);
d3 = boostduty(vo, vi3);

datapath = "./../data/";
filename = "duty.csv";
fid = fopen(strcat(datapath, filename),"wt");
  fprintf(fid, "Duty Cycle Calculation Results:");
  fprintf(fid, "\n\tD(Vi = %dV) \t= %.2f", vi1, d1);
  fprintf(fid, "\n\tD(Vi = %dV) \t= %.2f", vi2, d2);
  fprintf(fid, "\n\tD(Vi = %dV) \t= %.2f", vi2, d3);
  fprintf(fid, "\n---\n");
fclose(fid);

% 1B ---------------------------------------------------------------------------
s  = tf('s');
h1 = tf((vo/(1-d1))*(1-s*L/R/(1-d1)^2)/(1+s*L/R/(1-d1)^2 + s^2*L*C/(1-d1)^2));
h2 = tf((vo/(1-d2))*(1-s*L/R/(1-d2)^2)/(1+s*L/R/(1-d2)^2 + s^2*L*C/(1-d2)^2));
h3 = tf((vo/(1-d3))*(1-s*L/R/(1-d3)^2)/(1+s*L/R/(1-d3)^2 + s^2*L*C/(1-d3)^2));

[mag1, pha1, angf1] = bode(h1, angf);
[mag2, pha2, angf2] = bode(h2, angf1);
[mag3, pha3, angf3] = bode(h3, angf1);

data = [mag1, pha1, angf1'];
datapath = "./../data/";
```

```
filename = "h1.csv";

fid = fopen(strcat(datapath, filename), "wt");
  fprintf(fid, "%s, %s, %s\n", "magnitude", "phase", "angfreq");
fclose(fid);
dlmwrite(strcat(datapath, filename), data, ",", "-append", "precision", 6);

data = [mag2, pha2, angf2'];
datapath = "./../data/";
filename = "h2.csv";

fid = fopen(strcat(datapath, filename), "wt");
  fprintf(fid, "%s, %s, %s\n", "magnitude", "phase", "angfreq");
fclose(fid);
dlmwrite(strcat(datapath, filename), data, ",", "-append", "precision", 6);

data = [mag3, pha3, angf3'];
datapath = "./../data/";
filename = "h3.csv";

fid = fopen(strcat(datapath, filename), "wt");
  fprintf(fid, "%s, %s, %s\n", "magnitude", "phase", "angfreq");
fclose(fid);
dlmwrite(strcat(datapath, filename), data, ",", "-append", "precision", 6);

% 1C
switch control
  case ctrl_pi
    kp = 0.00010;
    ki = 20e3;
    g = kp * (1 + ki/s);
  case ctrl_pit1
    g = 1;
  case ctrl_pidt2
    kp  = 0.0002;
    wi1 = 1000;
    wi2 = 3e3;
    wt1 = 10e3;
    wt2 = 30e3;
    g   = kp * (1 + wi1/s) * (1 + s/wi2) / (1 + s/wt2) / (1 + s/wt1);
  otherwise
    g   = 1;
endswitch

% 1D

go1 = h1*g;
go2 = h2*g;
go3 = h3*g;

[mag1, pha1, angf1] = bode(go1);
```

```
[mag2, pha2, angf2] = bode(go2, angf1);
[mag3, pha3, angf3] = bode(go3, angf1);

data = [mag1, pha1, angf1'];
datapath = "./../data/";
filename = "go1.csv";

fid = fopen(strcat(datapath, filename), "wt");
  fprintf(fid, "%s, %s, %s\n", "magnitude", "phase", "angfreq");
fclose(fid);
dlmwrite(strcat(datapath, filename), data, ",", "-append", "precision", 6);

data = [mag2, pha2, angf2'];
datapath = "./../data/";
filename = "go2.csv";

fid = fopen(strcat(datapath, filename), "wt");
  fprintf(fid, "%s, %s, %s\n", "magnitude", "phase", "angfreq");
fclose(fid);
dlmwrite(strcat(datapath, filename), data, ",", "-append", "precision", 6);

data = [mag3, pha3, angf3'];
datapath = "./../data/";
filename = "go3.csv";

fid = fopen(strcat(datapath, filename), "wt");
  fprintf(fid, "%s, %s, %s\n", "magnitude", "phase", "angfreq");
fclose(fid);
dlmwrite(strcat(datapath, filename), data, ",", "-append", "precision", 6);
```