# Simulation exercise 2: Control design of buck converter

The control system is designed for buck converter by using the loop-shaping technique. The control is created by using PI-controller.  In this exercise, the guideline for designing output voltage controller of VF-VO (voltage fed and voltage output) buck converter is given. It is also shown, how to implement a closed-loop simulation model in MATLAB Simulink.

**Create the Matlab Simulink model**

1. Open the boost converter model done in the simulation exercise 1.

2. Modify the converter to be buck-converter by changing the places of inductor, MOSFET and diode.

   Use the following simulation parameters
   - Input voltage DC 100V
   - Inductance 125µH (ESR 1Ω) and capacitance 2.4µF (ESR 20mΩ)
   - Diode voltage drop 0.8V and on-time resistance 0.01Ω
   - MOSFET on-time resistance 0.01Ω
   - Load AC current source 1A
   - Switching frequency 100 kHz
   - Simulation time 30ms

3. Test that the simulation model operates correctly with constant duty cycle, i.e.

   $V_o = D*V_{in}$. For example the output voltage should be approximately 50 V if the duty cycle is 0.5.

4. The duty ratio was constant in the previous exercise. In this case, the duty cycle is changed by using the control system. The reference value for the duty ratio is created bv using PI-controller. The positive input of the controller is the output voltage reference and the negative input is the measured output voltage as shown in Fig. 1.

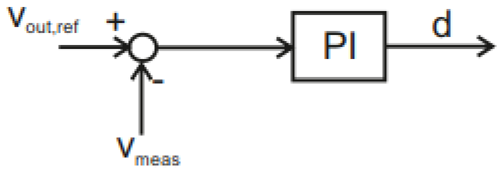

**Fig 1.** Voltage control of buck converter

5. The PI-controller is formed to Simulink in transfer function form.

$$G_{oc} = K_f \frac{(1+s/\omega_{z1})(1+s/\omega_{z2})}{s(1+s/\omega_{p1})(1+s/\omega_{p2})}$$

The controller includes gain ($K_f$), integrator and two *transfer function* blocks as shown in Figure

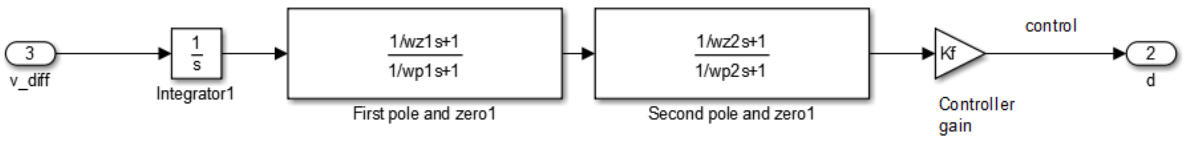

**Fig 2.** PI-controller

6. The control parameters are set by defining the required places of poles ($\omega_{p1}$, $\omega_{p2}$) and zeros ($\omega_{z1}$, $\omega_{z2}$).

$$K_f = 1$$

$$\omega_{z1} = \omega_{z2} = \frac{1}{\sqrt{LC}}$$

$$\omega_{p1} = \frac{1}{r_C C}$$

$$\omega_{p2} = \frac{\pi f_s}{2}$$

After this, the gain is set by looking the gain at 10 kHz frequency in the Bode diagram. The gain needs to be as large as the gain, $K_f$ = 10^(X dB/20).

The following commands can be used in Matlab

```
% Control of Buck converter

% Create the Laplace variable.
s = tf('s');

% Calculation of steady-state values
D = (Vo + (rL + rd)*Io + Vd) / (Vin + (rd - rds)*Io + Vd);
Dpil = 1-D;
Iin = D*Io;

% Additonal constants to simplify the state matrices.
R1 = rL + rC + D*rds + Dpil*rd;
V1 = Vin + (rd - rds)*Io + Vd;
Am = [-R1/L, -1/L
        1/C,   0];
Bm = [D/L, rC/L, V1/L
```

```
      0,    -1/C, 0];
Cm = [D,   0
      rC, 1];
Dm = [0,   0,   Io
      0, -rC, 0];

% 2x2 identify matrix
Im = eye(2);


% Solve the transfer functions
G = Cm * inv(s*Im - Am) * Bm + Dm;

% All six transfer functions are not required.
% Concentrate to transfer function from control to output
% presenting the relationship between duty ratio and output voltage
Gco_o = G(2,3);

% Setting the bode plot options, x-axis rad/s -> Hz
P = bodeoptions;
P.FreqUnits = 'Hz'; % Setting the x-axis rad/s -> Hz
P.Xlim = [1e1 1e5]; % Setting the x-axis from 10 Hz to 100 kHz

figure(1)
bode(Gco_o,'k',P)
grid on
legend('Gco @ Vin = 100V')

h = findobj(gcf,'type','line'); % bode plot line width
set(h,'linewidth',1.5);

% PI-controller, unity gain Kf=1
 wz1 = 1/sqrt(L*C)
 wz2 = 1/sqrt(L*C)
 wp1 = 1/(rC*C)
 wp2 = (pi*fs)/2
 Kf = 1;
 Gcc = Kf*(1+s/wz1)*(1+s/wz2)/(s*(1+s/wp1)*(1+s/wp2));

% Plot the bode plot of the controller
figure(2)
bode(Gcc,'k',P)
grid on
legend('Gcc')

% Loop gain of the controller Lv = Gcc*Gco_o
figure(3)
bode(Gco_o*Gcc,'k',P)
grid on
legend('Lv = GcoGcc')


% Check from Fig. 3 how large gain is at 10 kHz, if the required control
% bandwidth is 10 kHz.
% e.g. 10kHz -> gain 38.3dB -> Kf = 10^(38.3/20)
% Gain will be increased hence the required crossover frequency is achieved
```

```
Kf = 82.05
Gcc = Kf*(1+s/wz1)*(1+s/wz2)/(s*(1+s/wp1)*(1+s/wp2));

% Final loop gain of the buck converter
figure(4)
bode(Gco_o*Gcc,'k',P)
grid on
legend('Lv = GcoGcc')
h = findobj(gcf,'type','line');
set(h,'linewidth',1.5);
```

## Simulation exercises

1. What are the values of zeros, poles and the gain?
2. Present the Bode-plot of the final loop gain
3. Simulate the system with these control parameters. Use ***step***-block as the output voltage reference value. The voltage is e.g. at first 20 V during 15ms and increase to be 50V after that. Look at the output voltage reference value and measured value in the same scope (use mux-block).
4. Change the load current by using step-block to be at first 1A and later on 0.3 A. How the output voltage control behaves?