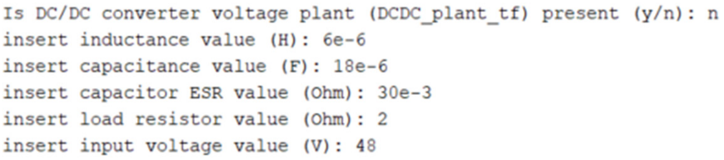
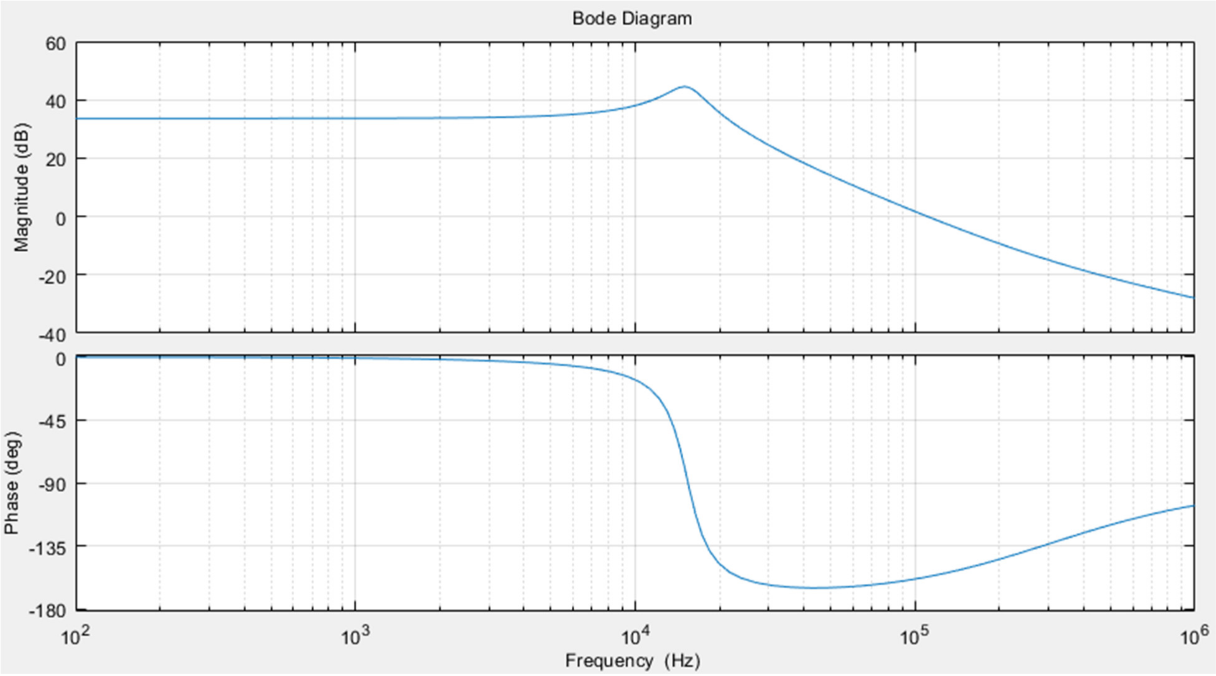
Matlab code to determine transfer function of the DC-DC converter and design Type III controller

This MATLAB scripts function is proposed to calculate analog Type III (3p2z) controller for DC-DC Buck converter in voltage control mode. At first the user is asked if DC-DC converter voltage plant is known. In case it is unknown (entering “n” in the script and pressing enter), the user is asked to enter the main parameters of the converter like inductance, capacitance, capacitor equivalent series resistance (ESR), load resistance and input voltage:



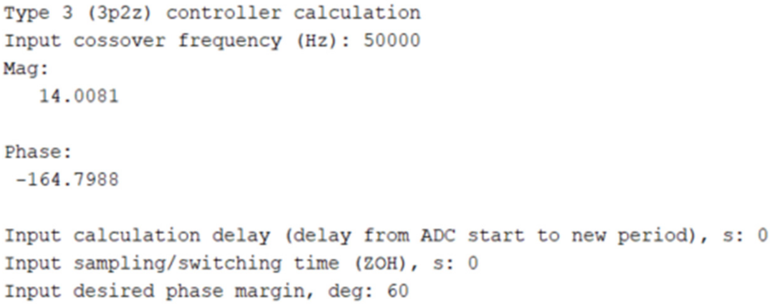
```
Is DC/DC converter voltage plant (DCDC_plant_tf) present (y/n): n
insert inductance value (H): 6e-6
insert capacitance value (F): 18e-6
insert capacitor ESR value (Ohm): 30e-3
insert load resistor value (Ohm): 2
insert input voltage value (V): 48
```

After this has been done, the script calculates and displays the transfer function of the DC-DC converter:



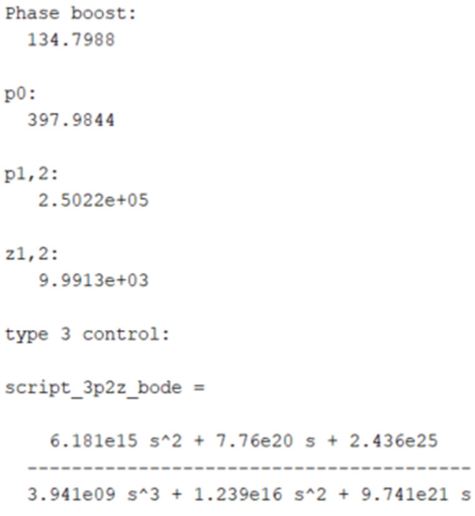
Otherwise, if the transfer function is known the user should enter letter “y”. The transfer function can be obtained by previously selecting case “n” or by other means like measurements and approximation. The transfer function is described with variable titled DCDC_plant_tf.

This transfer function further is used to calculate the Type III controller. The calculations are based on K factor method and equations that can be found in the paper. To design the controller user should enter crossover(cut-off) frequency after which the script calculates the gain and phase values at this frequency. Then the user is asked to insert delays if they are present in the system, like in case of digital control. If the control is analog, then the time can be inserted as 0 for both cases (calculation delay and sampling delay):



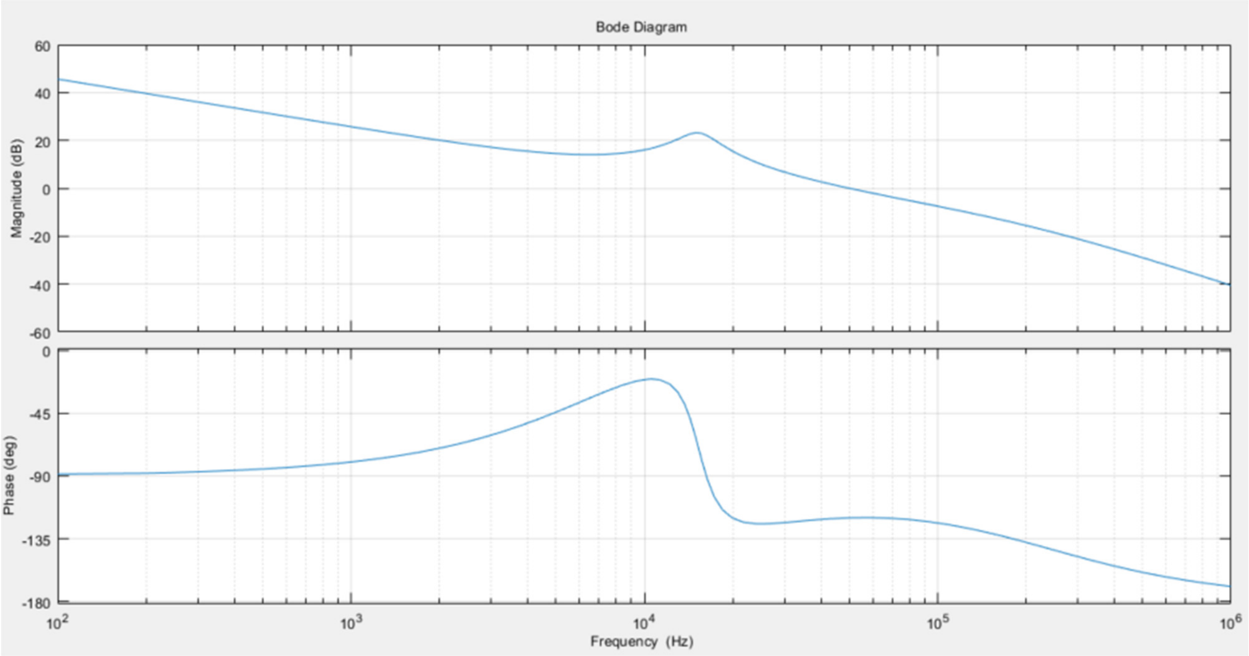
```
Type 3 (3p2z) controller calculation
Input cossover frequency (Hz): 50000
Mag:
   14.0081

Phase:
 -164.7988

Input calculation delay (delay from ADC start to new period), s: 0
Input sampling/switching time (ZOH), s: 0
Input desired phase margin, deg: 60
```

These values are then used for Type 3 controller design and the script outputs required phase boost value from the control and frequency at which the poles and zeros should be placed and outputs transfer function of analog Type III controller:

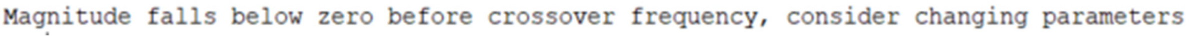


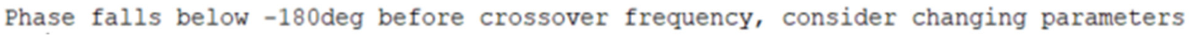
```
Phase boost:
  134.7988

p0:
  397.9844

p1,2:
   2.5022e+05

z1,2:
   9.9913e+03

type 3 control:

script_3p2z_bode =

    6.181e15 s^2 + 7.76e20 s + 2.436e25
  ---------------------------------------
  3.941e09 s^3 + 1.239e16 s^2 + 9.741e21 s
```

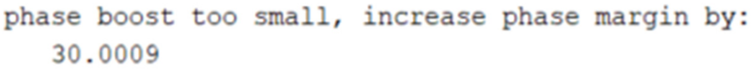
This result is then displayed as a combined transfer function of control and converter, that can be used to analyze the stability of the DC-DC converter:

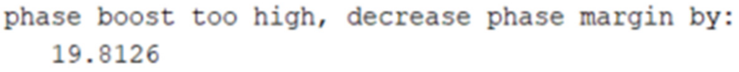


In case the gain falls below zero before crossover frequency or phase falls below -180°, the user is notified, that there needs to be an optimization. In case the phase boost can't be achieved with Type 3 control, the user is also notified that phase margin needs to be increased or decreased, in case that isn't possible, crossover frequency might need to be changed.


```
Magnitude falls below zero before crossover frequency, consider changing parameters
```


```
Phase falls below -180deg before crossover frequency, consider changing parameters
```


```
phase boost too small, increase phase margin by:
   30.0009
```


```
phase boost too high, decrease phase margin by:
   19.8126
```

The following code needs to be inserted in Matlab:

```
s = tf('s');
options = bodeoptions;
options.FreqUnits = 'Hz';
options.XLimMode = 'manual';
options.XLim = [100,1000000];
options.Grid = 'on';
disp('The output values are in Hz and Deg and resulting
transfer function is in Continuous-time, to obtain
discrete-time transfer function, use c2d function after
script')
prompt = input('Is DC/DC converter voltage plant
(DCDC_plant_tf) present (y/n): ','s');
if strcmp(prompt, 'n')
   prompt = 'insert inductance value (H): ';
   script_L = input(prompt);
   prompt = 'insert capacitance value (F): ';
   script_C = input(prompt);
   prompt = 'insert capacitor ESR value (Ohm): ';
   script_R_ESR = input(prompt);
   prompt = 'insert load resistor value (Ohm): ';
   script_R = input(prompt);
   prompt = 'insert input voltage value (V): ';
   script_Vin = input(prompt);
   script_wESR = 1/(script_C*script_R_ESR);
   script_wO = 1/(sqrt(script_C*script_L));
   script_Q = script_R*sqrt(script_C/script_L);
   DCDC_plant_tf =
script_Vin*(1+s/script_wESR)/(1+s/(script_Q*script_wO)+(s/s
cript_wO)^2);
end
bode(DCDC_plant_tf, options)
disp('Type 3 (3p2z) controller calculation')
prompt = 'Input cossover frequency (Hz): ';
script_fc = input(prompt);
[script_G,script_theta] =
bode(DCDC_plant_tf,script_fc*2*pi);
script_G = mag2db(script_G);
disp('Mag:');
disp(script_G);
disp('Phase:');
disp(script_theta);
```

```
script_G = db2mag(-script_G);
prompt = 'Input calculation delay (delay from ADC start to
new period), s: ';
script_ptd_time = input(prompt);
prompt = 'Input sampling/switching time (ZOH), s: ';
script_ZOH_time = input(prompt);
prompt = 'Input desired phase margin, deg: ';
script_PM = input(prompt);
script_boost = script_PM - script_theta - 90 - (-
360*script_ptd_time*script_fc -
180*script_fc*script_ZOH_time);
disp('Phase boost:');
disp(script_boost);

if script_boost<=180 && script_boost>0
    script_k_var = (tan(script_boost*pi/180/4 + pi/4));
    script_z1 = script_fc/script_k_var;
    script_p1 = script_fc*script_k_var;
    script_p0 =
script_G*script_z1*(script_p1^2+script_fc^2)/(script_p1^2*s
qrt((script_z1/script_fc)^2+1)*sqrt((script_fc/script_z1)^2
+1));
    script_3p2z_bode =
(script_p0*2*pi)/s*((1+s/(script_z1*2*pi))^2)/((1+s/(script
_p1*2*pi))^2);
    disp('p0:');
    disp(script_p0);
    disp('p1,2:');
    disp(script_p1);
    disp('z1,2:');
    disp(script_z1);
    disp('type 3 control:');
    script_3p2z_bode
    script_result = script_3p2z_bode*DCDC_plant_tf;
    for k = 0.1:100.0*2*pi:script_fc*2*pi
        [script_G,script_theta] = bode(script_result, k);
        script_G = mag2db(script_G);
        if script_G<0
            disp('Magnitude falls below zero before
crossover frequency, consider changing parameters')
            break;
        end
        if script_theta<-180
```

```
            disp('Phase falls below -180deg before
crossover frequency, consider changing parameters')
            break;
        end
    end
    bode(script_result, options)
else
    if script_boost<=0
        disp('phase boost too small, increase phase margin
by: ');
        disp(-script_boost);
    end
    if script_boost>180
        disp('phase boost too high, decrease phase margin
by: ');
        disp(script_boost - 180);
    end
end
```