complete Boost converter design process:

PART - 1)

Initial specification

Given :-

- Input voltage ($V_{in}$) = 10 V
- Output voltage ($V_{out}$) = 20 V.
- Output power ($P_{out}$) = 15 W.
- Battery charging application
- Switching frequency ($f_{SW}$) = 20 $KH_2$
- Current ripple = 5% of average inductor current
- voltage ripple = 2% of output voltage.

1) Duty cycle (D):

$$D = 1 - \left(\frac{V_{in}}{V_{out}}\right) = 1 - \frac{10}{20} = 0.5$$

2) Output current ($I_{out}$):

$$I_{out} = P_{out} / V_{out} = \frac{15\,W}{20\,V} = 0.75\,A$$

3) Average Inductor Current ($I_{L-avg}$):

$$I_{L-avg} = I_{out} / (1-D) = 0.75\,A / (1-0.5) = 1.5\,A.$$

4) Load resistance (R):

$$R = V_{out}^2 / P_{out} = \frac{20 \times 20}{15} = 26.67\,\Omega$$

5) current ripple ($\Delta I_L$)

$$\Delta I_L = 5\% \text{ of } I_{L\,avg} = 0.05 \times 1.5\,A = 0.075\,A.$$

6) voltage ripple ($\Delta V$):

$$\Delta V = 2\% \times V_{out} = 0.02 \times 20\,V = 0.4\,V$$

7) Inductor value (L):

$$L = \frac{(V_{in} \times D)}{\Delta I_L \times f_{sw}}$$

$$L = \frac{10V \times 0.5}{0.075A \times 20000Hz}$$

$$L = 3.33 \text{ mH}$$

8) Capacitor value (C):

$$C = \frac{(I_{out} \times D)}{\Delta V \times f_{sw}}$$

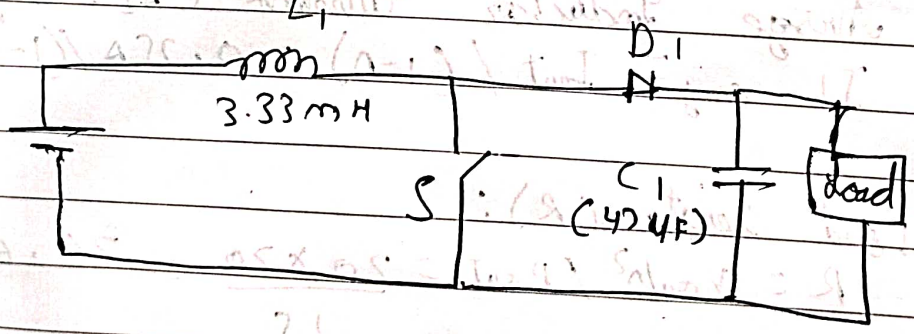
$$C = \frac{(0.75A \times 0.5)}{0.4 \times 20000 Hz}$$

$$C = 47 \mu F$$

## PART-2

Small signal analysis using state space modelling.

diagram



$0 = AX + BU$

$X \Rightarrow -A^{-1}BU$

$Y = CX = -CA^{-1}BU$

$\frac{Y}{U} = -CA^{-1}B$

$V_o = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} I_L \\ V_o \end{bmatrix}$

$\frac{V_o}{V_g}$ $V_o = C A^{-1} \begin{bmatrix} 1/L \\ 0 \end{bmatrix} V_g$ assume $I_2 = 0$ (Transfer relation when all other Input = 0)

$\frac{V_o}{I_2}$ $V_o = -CA^{-1} \begin{bmatrix} 0 \\ -\frac{1}{C} \end{bmatrix} I_2$

$A^{-1}$ $B$

$V_o = -\begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & x \\ -\frac{1}{(1-D)} & x \end{bmatrix} \begin{bmatrix} 1/L \\ 0 \end{bmatrix} V_g$

$= -1 \cdot \frac{L}{(1-D)} \cdot \frac{1}{L} = \frac{1}{1-D}$

$\boxed{\frac{V_o}{V_g} = \frac{1}{1-D}}$

# Small signal

$d = D + \hat{d}$

$V_o = V_o + \hat{V_o}$

$V_g = V_g + \hat{V_g}$

$i_2 = I_2 + \hat{i_2}$

$i_L = I_L + \hat{i_L}$

$\begin{bmatrix} I_L + \hat{i_L} \\ V_c + \hat{V_c} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{(1-D-\hat{d})}{L} \\ \frac{(1-D-\hat{d})}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} I_L + \hat{i_L} \\ V_c + \hat{V_c} \end{bmatrix}$

$+ \underbrace{\begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} \begin{bmatrix} V_g + \hat{V_g} \end{bmatrix}}_{BU} + \begin{bmatrix} 0 \\ -\frac{1}{C} \end{bmatrix} \begin{bmatrix} I_2 + \hat{i_2} \end{bmatrix}$

$\begin{bmatrix} V_o + \hat{V_o} \\ I_g + \hat{i_g} \end{bmatrix} = \begin{bmatrix} 0 & \phi \\ 1 & 0 \end{bmatrix} \begin{bmatrix} I_L + \hat{i_L} \\ V_c + \hat{V_c} \end{bmatrix}$

This small signal Model + steady state

Derivative of steady state = 0

$\dot{I_L} = 0 \quad , \dot{V_c} = 0$

We split more that into three part

$\begin{bmatrix} 0 & -\frac{(1-D)}{L} \\ \frac{1-D}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} I_L \\ V_c \end{bmatrix} + \begin{bmatrix} D & -\frac{(1-D)}{L} \\ \frac{1-D}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} \hat{i_L} \\ \hat{V_c} \end{bmatrix}$

$= 0$

$AX$ + $\begin{bmatrix} 0 & \frac{\hat{d}}{L} \\ -\frac{\hat{d}}{C} & 0 \end{bmatrix} \begin{bmatrix} I \\ V_o \end{bmatrix}$

(steady state part.)

$\Downarrow$

$\begin{bmatrix} \frac{V_o}{L} \\ -\frac{I}{C} \end{bmatrix} \cdot \hat{d}$

$AX + BU = 0$

$$\dot{\hat{x}} = A\hat{x} + B_1\hat{d} + B_2\hat{v}_g + B_3\hat{i}_g$$

$$\begin{bmatrix} \dot{\hat{i}}_L \\ \dot{\hat{v}}_c \end{bmatrix} = \begin{bmatrix} 0 & -\frac{(1-D)}{L} \\ \frac{1-D}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} \hat{i}_L \\ \hat{v}_c \end{bmatrix} + \begin{bmatrix} \frac{V_c}{L} \\ -\frac{I_L}{C} \end{bmatrix} \hat{d} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} \hat{v}_g + \begin{bmatrix} 0 \\ -\frac{1}{C} \end{bmatrix} \hat{i}_2$$

$$\hat{v}_o = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{i}_L \\ \hat{v}_c \end{bmatrix}$$

$$\hat{i}_g = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{i}_L \\ \hat{v}_c \end{bmatrix}$$

or

$$\Rightarrow \begin{bmatrix} \hat{v}_o \\ \hat{i}_g \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{i}_L \\ \hat{v}_c \end{bmatrix}$$

$$Y = CX$$

final small signal model

$$\begin{bmatrix} \dot{\hat{i}}_L \\ \dot{\hat{v}}_c \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -\frac{(1-D)}{L} \\ \frac{(1-D)}{C} & -\frac{1}{RC} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \hat{i}_L \\ \hat{v}_c \end{bmatrix}}_{\hat{x}} + \begin{bmatrix} \frac{1}{L} & 0 & \frac{V_0}{L} \\ 0 & -\frac{1}{C} & \frac{-I_L}{C} \end{bmatrix} \begin{bmatrix} \hat{v}_g \\ \hat{i}_2 \\ \hat{d} \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & -\frac{(1-D)}{L} \\ \frac{(1-D)}{C} & -\frac{1}{RC} \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{L} & 0 & \frac{V_0}{L} \\ 0 & -\frac{1}{C} & \frac{-I_L}{C} \end{bmatrix} \qquad D = 0$$

(represents direct feedback of ... converts)

1) Temporal operating mode,

$dT_s$, $(1-d)T_s$

$$\dot{x} = A_1 x + B_1 u$$
$$\dot{x} = A_2 x + B_2 u$$

2) Averaged large signal model

$$A_1 d + A_2 (1-d)$$
$$B_1 d + B_2 (1-d)$$

3) Derivatives of equilibrium variable $\Rightarrow 0$ all variables represent steady state quantities $d \Rightarrow D$, $x \Rightarrow X$, $y \Rightarrow Y$

$$0 = AX + BU$$

4) $d = D + \hat{d}$

$$x = X + \hat{x}$$
$$x = Y + \hat{y}$$

$\hat{d}\hat{x} \Rightarrow$ neglected.

take only first order

5) small signal model (used for controller design)

# Transfer function

$$G_1 = \frac{\text{Output}}{\text{control input}}$$

$$= \frac{\hat{V}_0(s)}{\hat{d}(s)} = C(sI - A)^{-1} B$$

Transfer

$$A = \begin{bmatrix} 0 & -\frac{(1-D)}{L} \\ \frac{(1-D)}{C} & -\frac{1}{RC} \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1 \\ \# & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{L} & 0 & \frac{V_0}{L} \\ 0 & -\frac{1}{C} & -\frac{I_L}{C} \end{bmatrix} \qquad D = 0$$

on substitution of values :-

$$A = \begin{bmatrix} 0 & -\frac{(1-0.5)}{3.33\times10^{-3}} \\ \frac{(1-0.5)}{47\times10^{-6}} & -\frac{1}{26.6\times10^{-6}} \end{bmatrix} = 1e^{4}\begin{bmatrix} 0 & -0.03 \\ 2.13 & -0.8 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{3.33\times10^{-3}} & 0 & \frac{20}{3.33\times10^{-3}} \\ 0 & -\frac{1}{47\times10^{-6}} & -\frac{1.5}{47\times10^{-6}} \end{bmatrix}$$

but we need only one column of B coz u is $\begin{bmatrix} v_g \\ i_2 \\ d \end{bmatrix}$ so only $\hat{d}$ is of interest in Transfer function hence

$$B = \begin{bmatrix} 3000 \\ 0 \end{bmatrix}$$

same as $\hat{V}_C$ that is our output get so its value from one value of C Matrix hence)

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

while $D = 0$

Now we will begin with transfer function:-

$$\frac{\hat{v}_o}{\hat{d}} = \frac{V_{in}}{(1-D)^2}\left(1 - \frac{sL_e}{R}\right)\cdot\frac{1 + sr_c C}{L_e C C s^2 + s\left(\frac{1}{RC} + \frac{r_c}{L_e}\right) + \frac{1}{L_e C}}$$

more simplified and some modifications give

$$G_{PS}(s) = \frac{V_{out}}{d(s)} = \frac{V_{in}}{L} \times \frac{R(1-D)^2}{s^2 + s^*\left(\frac{1}{(RC)}\right) + \frac{(1-D)^2}{LC}}$$

This is correct for boost ~~reg~~ because it has two poles ( from denominator $s^2 + 2\zeta\omega_n s + \omega_n^2$ ) has ~~two~~ one zero in RHP causing phase lag.

## PART-3

Now for designing K-factor controller we need

$$G_L(s) = G_C(s) \times G_{PS} \times G_{pwm} \times K_{fb}$$

or

$$G_{ol}$$

$G_C(s)$ = gain of control loop.

$G_{PS}$ = calculated transfer function

$G_{pwm}$ = pulse width modulator gain

$K_{fb}$ = feedback path gain.

Now at a frequency called crossover frequency $f_c$

$$|G_L(s)| \text{ at } f_c = 1$$

so we can calculate $G_c(s)$ from there by

$$G_c(s) = \frac{1}{G_{PWM}(f_c) \times G_{PS}(f_c) \times K_{fb}}$$

so now we will calculate each component one by one.

$$G_{PS} = \frac{6.389e07}{s^2 + 797.9s + 6.389e06}$$

from graph plotted ~~above~~ at last $G_{PS}$ and $\angle G_{PS}$ we get the $f_c$ and $\angle$ phase angle.

$f_c = 133$ Hz

Phase at $f_c = 5.99$ degree

$G_c$ at $f_c = 9.0917$

As phase of this $f_c$ is too less, we assumed $f_c$ as $\frac{1}{10}$ th of given frequency as standard to proceed.

$f_c = 2$ KHz

$G_{PS}$ at $f_c$

$G_{PS}(j\omega c) = |G_{PS}(j\omega c)| \angle\theta$

$|G_{PS}(j\omega c)| = 0.0891$ (should be close to 1)

$\angle G_{PS}(j\omega c) = -135.6°$ (should be between $-120$ at $-135$)

Now we proceed with

$G_{PWM}$ what $f_c$ should be $\frac{1}{V_r}$ ($V_{ramp}$ voltage)

I assume ramp voltage as 1.5 V
(should be between 1V and 2V)

so $G_{PWM} = 0.667 = \frac{1}{1.5}$

Now K for feedback again (assuming microcontroller implemented using ESP32 or Arduino with 3.3V ADC).

so $K_{fb} = \frac{3.3}{20} = 0.165$

that can be implemented by voltage divider

$R_1 = 100\ K\Omega$

$R_2 = 19.8\ K\Omega$

Now

final $G_c(j\omega_c) = 101.45$

Parameters from K-factor Method

$K = 10$ (standard value for good performance)

$f_z = \frac{f_c}{\sqrt{K}} = \frac{2000}{\sqrt{10}} = 632.5\ Hz$

$\omega_z = 2\pi * f_z = 3972\ \frac{rads}{s}$

This is zero frequency

Now we calculate pole frequency

$$f_p = \frac{f_c}{\sqrt{K}} * \sqrt{K} = 2000 * \sqrt{10} = 6325 \text{ Hz}$$

$$w_p = 2\pi * f_p = 39,720 \ \frac{\text{rad}}{s}$$

$$K_{boost} = K = 10$$

Now find controller gain ($K_c$)

$$K_c = G_c(s) \times \frac{w_z}{K_{boost}}$$

$$K_c = 101.45 \times \frac{3972}{10}$$

$$K_c = 40,296$$

Finally type 3 compensator Transfer function

$$G_c(s) = K_c * \frac{\left(\frac{s}{w_z} + 1\right)^2}{s * \left(\frac{s}{w_p} + 1\right)^2}$$

Verification:

1) At $f_c$:

$$|T(jw_c)| = |G_c(jw_c)| * |G_{PS}(jw_c)| * G_{DM} * K_{fb} = 1$$

2) Phase Margin

$$PM = 180 + \angle T(jw_c) = 60°$$

3) Gain Margin > 10dB

## PART-4

plant transfer function

$$G_{PZ} = \frac{0.07884(z) + 0.0779}{z^2 - 1.945(z) + 0.9608}$$

Controller transfer function

$$G_{CZ} = \frac{20.61 z^{\wedge 3} - 13.16(z)^2 - 19.93(z) + 13.}{z^{\wedge 3} - 1.007 z^{\wedge 2} + 0.006732 - 1.07}$$

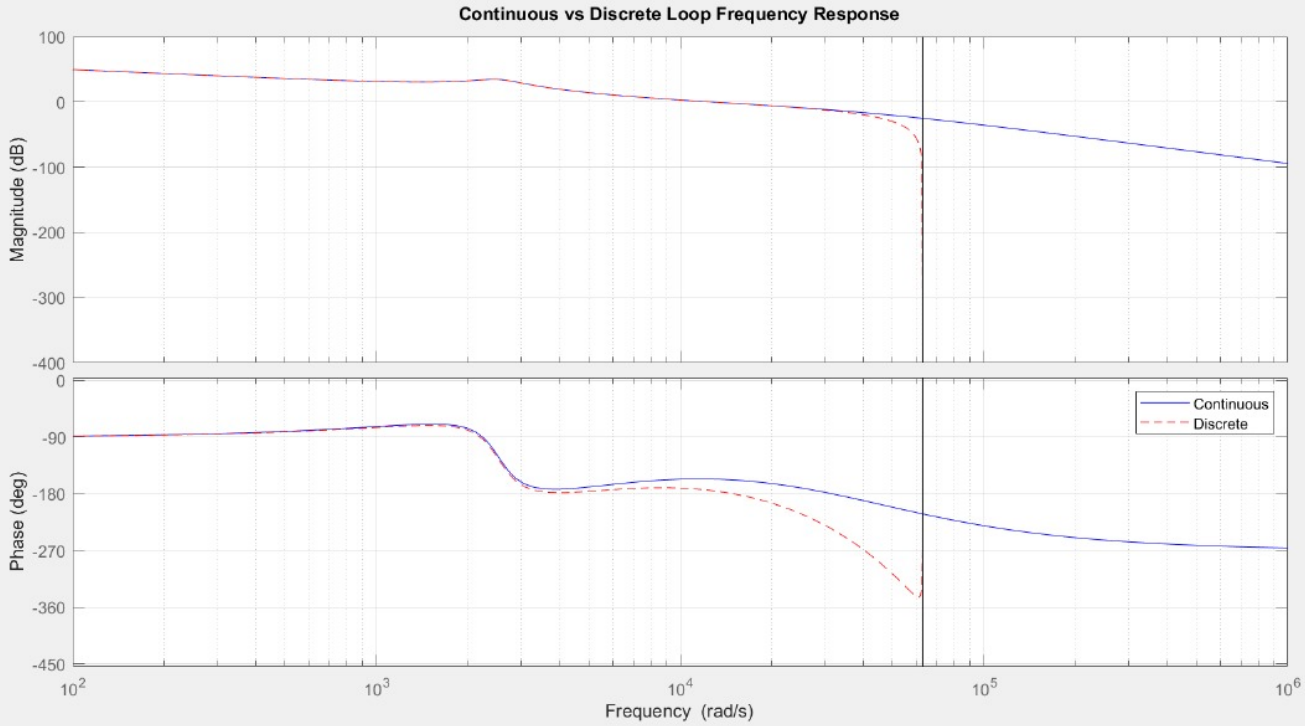
## PART-5

BOM

| Component | Specification | Quantity |
|---|---|---|
| **Inductor (L)** | Custom inductor using ETD44/22/15 core | 1 |
| **Capacitor (Cout)** | 47 µF, 25V Electrolytic | 1 |
| **Resistor (R1, R2)** | Voltage divider for feedback (e.g., 10kΩ, 2kΩ) | 2 |
| **MOSFET (Q1)** | IRF540N or equivalent | 1 |
| **Diode (D1)** | Schottky diode, 100V, 5A (e.g., 1N5822) | 1 |
| **Op-Amp (U1, U2, U3)** | TL081 / TLV2372 (for compensator) | 3 |
| **PWM Controller** | TL494 / SG3525 / Digital PWM via MCU | 1 |
| **Voltage Reference** | 2.5V Precision Reference (e.g., LM4040-2.5) | 1 |
| **Feedback Components** | Resistors (1kΩ, 10kΩ), Capacitors (100nF, 10nF) | As required |
| **Microcontroller** | ESP32 / STM32 (for digital control) | 1 |
| **ADC Module** | 12-bit ADC (if using external) | 1 |
| **DAC Module** | (Optional) 8-bit/12-bit for testing | 1 |

Figure 3

File Edit View Insert Tools Desktop Window Help



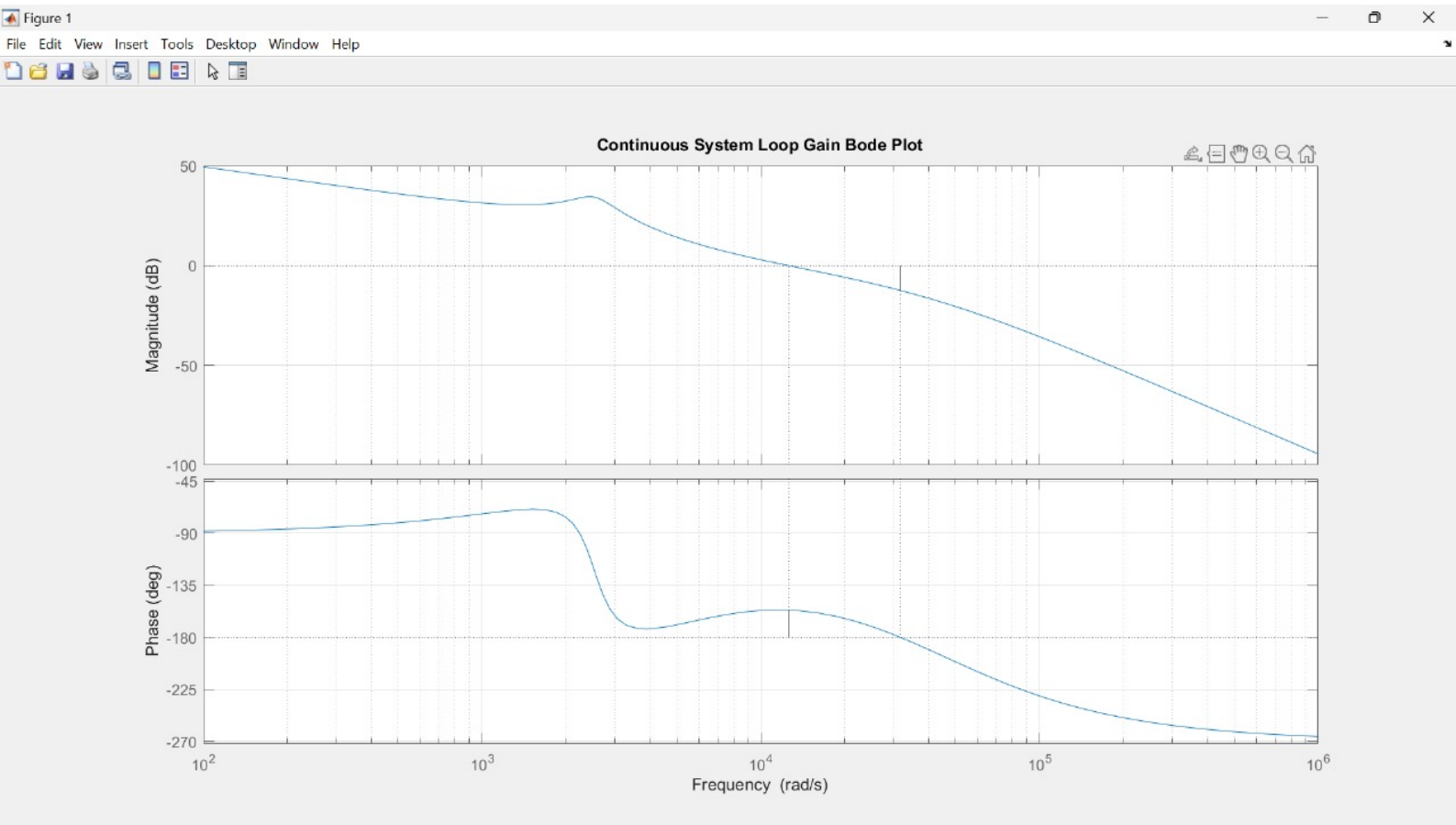


Figure 1
File Edit View Insert Tools Desktop Window Help
Continuous System Loop Gain Bode Plot
Magnitude (dB)
50
0
-50
-100
Phase (deg)
-45
-90
-135
-180
-225
-270
$10^2$
$10^3$
$10^4$
$10^5$
$10^6$
Frequency (rad/s)

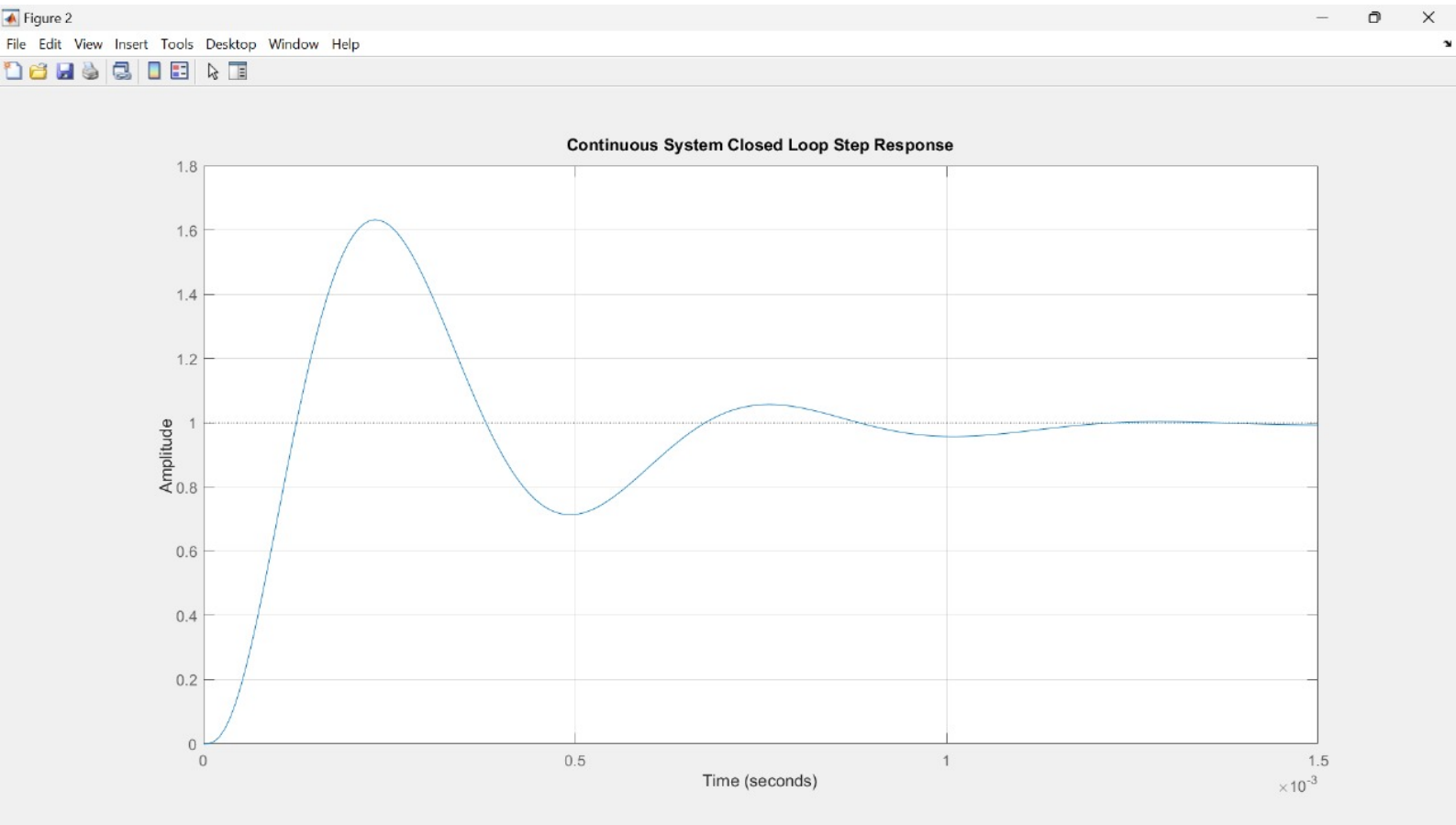


Figure 2
File Edit View Insert Tools Desktop Window Help
Continuous System Closed Loop Step Response
Amplitude
Time (seconds)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
0
0.5
1
1.5
×10-3

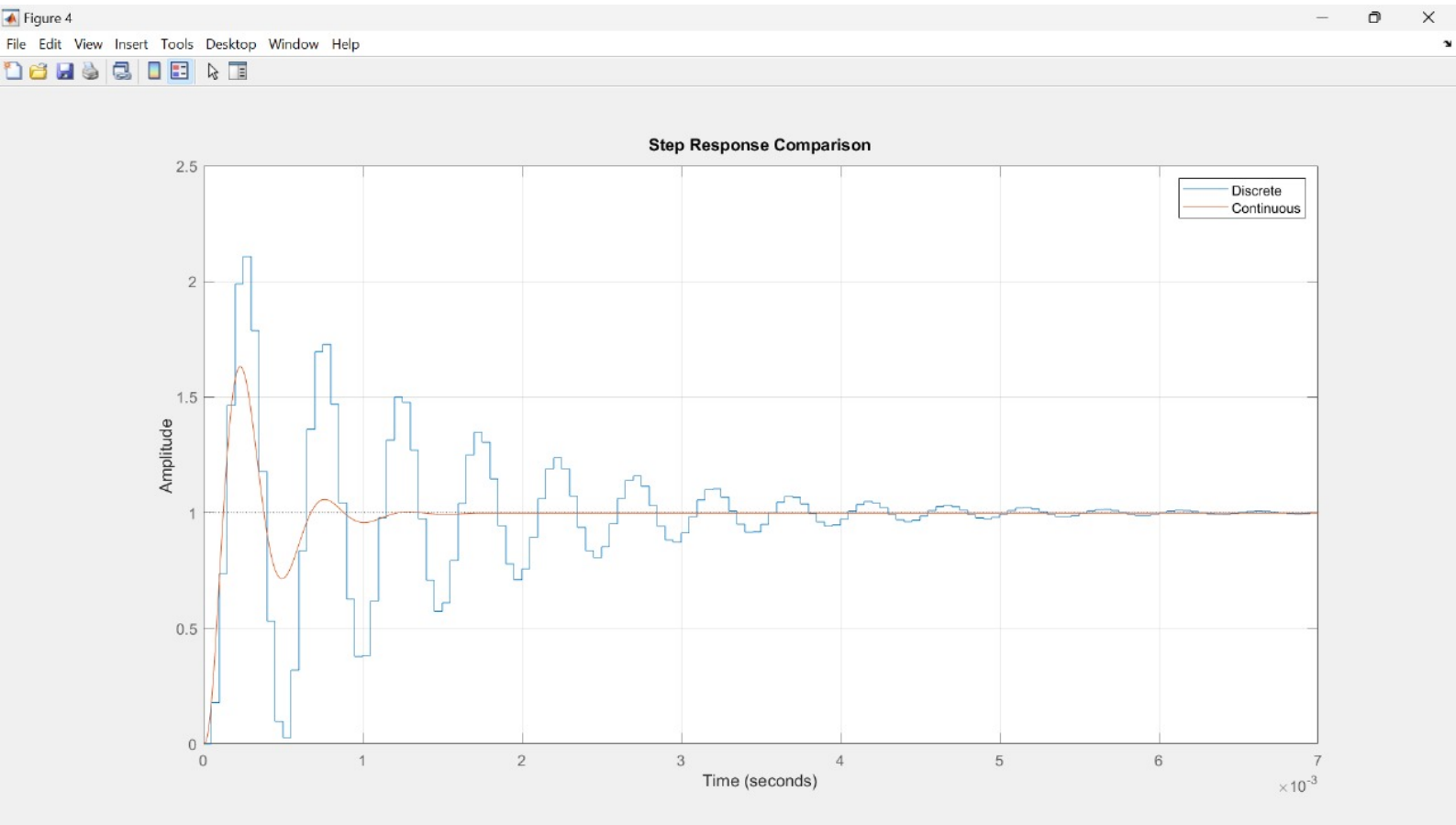


Figure 4
File Edit View Insert Tools Desktop Window Help
Step Response Comparison
Discrete
Continuous
Amplitude
Time (seconds)
0
0.5
1
1.5
2
2.5
0
1
2
3
4
5
6
7
×10-3

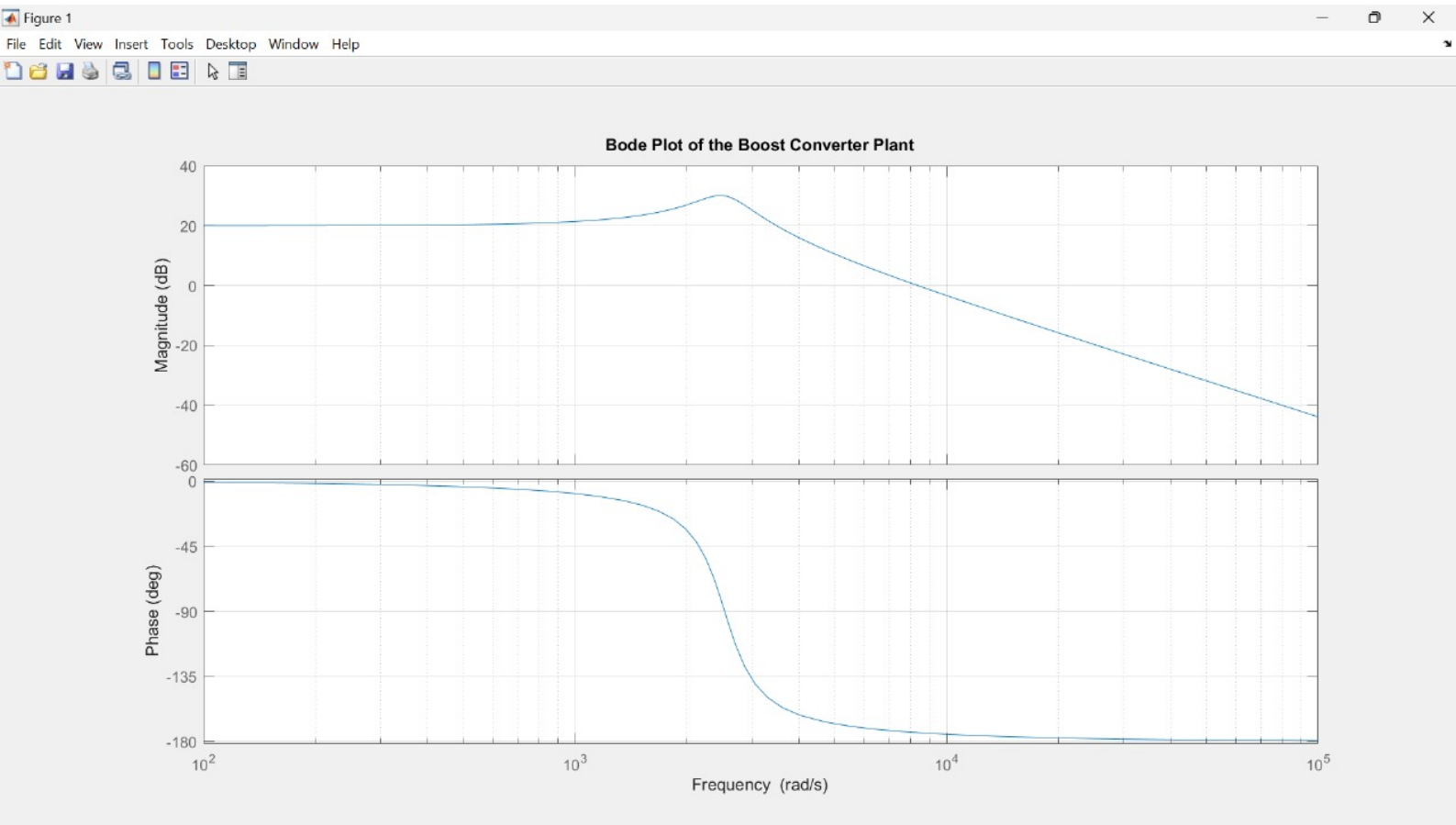
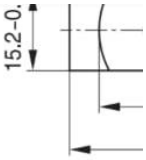


Figure 1
File Edit View Insert Tools Desktop Window Help
Bode Plot of the Boost Converter Plant
Magnitude (dB)
40
20
0
-20
-40
-60
Phase (deg)
0
-45
-90
-135
-180
$10^2$
$10^3$
$10^4$
$10^5$
Frequency (rad/s)

$V_e$ = 17800 mm³

**Approx. weight** 94 g/set




**Ungapped**

| Material | $A_L$ value nH | $\mu_e$ | $B_S$ mT | $P_V$ W/set |
|---|---|---|---|---|
| N27 | 3300 +30/–20% | 1560 | 320[1)] | < 3.48 (200 mT, , 1 |
| N87 | 3500 +30/–20% | 1650 | 320[1)] | < 9.40 (200 mT, 100 kHz, 100 °C) |
| N97 | 3600 +30/–20% | 1720 | 320[1)] | < 8.00 (200 mT, 100 kHz, 100 °C) |
| N95 | 4400 +30/–20% | 2085 | 330[1)] | < 8.85 (200 mT, 100 kHz, 100 °C)<br><10.20 (200 mT, 100 kHz, 25 °C) |



[1)] H = 250 A/m; f = 10 kHz; T = 100 °C

**Gapped** ($A_L$ values/air gaps examples)

| Material | g mm | $A_L$ value approx. nH | $\mu_e$ | Or ** |
|---|---|---|---|---|
| N27, N87 | 0.20 ±0.02 | 862 | 407 | B6 |
| | 0.50 ±0.05 | 438 | 207 | B6 |
| | 1.00 ±0.05 | 262 | 124 | B6 |
| | 1.50 ±0.05 | 194 | 92 | B6 |
| | 2.00 ±0.05 | 150 | 70 | B6 |

The $A_L$ value in the table applies to a core set comprising one ungapped core (
and one gapped core (dimension g > 0 mm).

Other $A_L$ values/air gaps and materials available on request — see Processing

Ans) I have selected 0.5 ± 0.05 mm

'438' is our $A_L$

So we will use this formula to find value of 'n'

$$L = n^2 A_L$$

$$n = \sqrt{L/A_L} = \sqrt{\frac{3.3mH}{438nH}} = 86.80$$

$$\approx 86$$

→ Here the number of turns is more so more resistant will be there

→ But this best compromise for winding

# INCOMPLETE TILL Gc

```
% Given parameters
Vin = 10;   % Input Voltage (V)
Vout = 20;  % Desired Output Voltage (V)
Pout = 15;  % Output Power (W)
R = (Vout^2) / Pout;  % Load Resistance (Ohm)
L = 333e-5;  % Inductor Value (H)
C = 47e-6;  % Capacitor Value (F)
D = 1 - (Vin / Vout); % Duty cycle (steady state)

% State-Space Matrices (A, B, C, D)
A = [0 -1/L; 1/C -1/(R*C)];
B = [Vin/L; 0];
C = [0 1]; % Measuring output voltage
D = 0;

% Create state-space system
sys = ss(A, B, C, D);

% Find Transfer Function Gps(s) = Vout(s) / D(s)
Gps = tf(sys);

% Display Transfer Function
disp('Transfer Function Gps(s):');
Gps

% Bode Plot for analysis
bode(Gps);
grid on;
title('Bode Plot of the Boost Converter Plant');
[gm, pm, wcg, wcp] = margin(Gps);
```

```
fc = wcp / (2 * pi); % Convert from rad/s to Hz


fprintf('Crossover Frequency (fc) = %.2f Hz\n', fc);
[gm, pm, wcg, wcp] = margin(Gps);
fc = wcp / (2 * pi); % Convert rad/s to Hz


fprintf('Crossover Frequency (fc) = %.2f Hz\n', fc);
fprintf('Phase at fc = %.2f degrees\n', pm);
[mag, phase, w] = bode(Gps, 2 * pi * fc);
Gps_at_fc = mag(:); % Magnitude at fc
G_PWM = 0.6667;
K_fb = 0.165;
Gc_at_fc = 1 / (Gps_at_fc * G_PWM * K_fb);
fprintf('Gc at fc = %.4f\n', Gc_at_fc);
```

## Complete code for controller (K factor )

```
% Complete Boost Converter Design, Analysis, and Digital Implementation
clear all;
close all;
clc;


%% 1. INITIAL PARAMETERS
```

```
% System Requirements
Vin = 10;         % Input Voltage (V)
Vout = 20;        % Desired Output Voltage (V)
Pout = 15;        % Output Power (W)
fsw = 20e3;       % Switching frequency (Hz)
Ts = 1/fsw;       % Switching period (s)

%% 2. CONVERTER DESIGN
% Operating Point Calculations
Iout = Pout/Vout;              % Output current (A)
D = 1 - (Vin/Vout);            % Duty cycle
IL_avg = Iout/(1-D);           % Average inductor current

% Ripple Specifications
di_L = 0.05 * IL_avg;          % 5% current ripple
dv_out = 0.02 * Vout;          % 2% voltage ripple

% Component Value Calculations
L = (Vin * D * Ts)/(di_L);       % Required inductance
Cout = (Iout * D * Ts)/(dv_out);  % Required capacitance
R = (Vout^2)/Pout;             % Load resistance

% Display Component Values
fprintf('\nConverter Design Parameters:\n');
fprintf('Duty Cycle: %.2f\n', D);
fprintf('Average Inductor Current: %.2f A\n', IL_avg);
fprintf('Required Inductance: %.2e H\n', L);
fprintf('Required Capacitance: %.2e F\n', Cout);
fprintf('Load Resistance: %.2f Ω\n', R);

%% 3. SMALL SIGNAL MODELING
```

```
% State Space Matrices
A = [0 -1/L; 1/Cout -1/(R*Cout)];
B = [Vin/L; 0];
C_matrix = [0 1];
D_matrix = 0;

% Create State Space System
sys = ss(A, B, C_matrix, D_matrix);

% Get Transfer Function
Gps = tf(sys);
fprintf('\nPlant Transfer Function:\n');
disp(Gps);

%% 4. CONTROLLER DESIGN (K-Factor Method)
% Frequency Analysis
fc = fsw/10;                % Crossover frequency
wc = 2*pi*fc;               % Angular crossover frequency

% Get Plant Characteristics at Crossover
[mag, phase] = bode(Gps, wc);
mag_db = 20*log10(mag);
phase_deg = phase;

% Control System Parameters
G_PWM = 0.6667;             % PWM gain
K_fb = 0.165;               % Feedback gain

% K-Factor Method Parameters
PM_desired = 60;            % Desired phase margin (degrees)
K = 10;                     % K-factor
```

```
% Calculate Type 3 Compensator
phi_boost = PM_desired - phase_deg - 180;
fz = fc/sqrt(K);              % Zero frequency
fp = fc*sqrt(K);              % Pole frequency

% Compensator Transfer Function
s = tf('s');
Gc = ((s/2/pi/fz + 1)^2)/((s/2/pi/fp + 1)^2) * (1/s);

% Adjust Gain
[mag_c, phase_c] = bode(Gc, wc);
Kc = 1/(mag*mag_c*G_PWM*K_fb);
Gc = Kc*Gc;

% Complete Loop Transfer Function
T = Gc*Gps*G_PWM*K_fb;

%% 5. CONTINUOUS SYSTEM ANALYSIS
figure(1);
margin(T);
grid on;
title('Continuous System Loop Gain Bode Plot');

figure(2);
closed_loop_sys = feedback(T, 1);
step(closed_loop_sys);
grid on;
title('Continuous System Closed Loop Step Response');

% Calculate Continuous System Metrics
```

```
[Gm, Pm, Wcg, Wcp] = margin(T);
fprintf('\nContinuous System Metrics:\n');
fprintf('Phase Margin: %.2f degrees\n', Pm);
fprintf('Gain Margin: %.2f dB\n', 20*log10(Gm));
fprintf('Crossover Frequency: %.2f Hz\n', Wcp/(2*pi));

% Step Response Metrics
S = stepinfo(closed_loop_sys);
fprintf('Settling Time: %.3f ms\n', S.SettlingTime*1000);
fprintf('Overshoot: %.2f%%\n', S.Overshoot);

%% 6. DISCRETIZATION
% Discretize Plant (ZOH method)
Gpz = c2d(Gps, Ts, 'zoh');
fprintf('\nDiscrete Plant Transfer Function:\n');
disp(Gpz);

% Discretize Controller (Tustin method)
Gcz = c2d(Gc, Ts, 'tustin');
fprintf('\nDiscrete Controller Transfer Function:\n');
disp(Gcz);

% Discrete Loop Transfer Function
Tz = Gcz * Gpz * G_PWM * K_fb;

%% 7. DISCRETE SYSTEM ANALYSIS
% Frequency Response Comparison
figure(3);
bode(T, 'b', Tz, 'r--');
grid on;
legend('Continuous', 'Discrete');
```

```
title('Continuous vs Discrete Loop Frequency Response');

% Step Response Comparison
figure(4);
discrete_closed_loop = feedback(Tz, 1);
step(discrete_closed_loop);
hold on;
step(closed_loop_sys);
grid on;
legend('Discrete', 'Continuous');
title('Step Response Comparison');

% Calculate Discrete Metrics
[Gm_d, Pm_d, Wcg_d, Wcp_d] = margin(Tz);
fprintf('\nDiscrete System Metrics:\n');
fprintf('Phase Margin: %.2f degrees\n', Pm_d);
fprintf('Gain Margin: %.2f dB\n', 20*log10(Gm_d));
fprintf('Crossover Frequency: %.2f Hz\n', Wcp_d/(2*pi));

Sd = stepinfo(discrete_closed_loop);
fprintf('Settling Time: %.3f ms\n', Sd.SettlingTime*1000);
fprintf('Overshoot: %.2f%%\n', Sd.Overshoot);

%% 8. DIGITAL IMPLEMENTATION
% Get Difference Equation Coefficients
[num_d, den_d] = tfdata(Gcz, 'v');

fprintf('\nDifference Equation Coefficients:\n');
fprintf('Numerator: ');
fprintf('%.4e ', num_d);
fprintf('\nDenominator: ');
```

```
fprintf('%.4e ', den_d);
fprintf('\n');

% Implementation Requirements
memory_words = length(num_d) + length(den_d) - 1;
fprintf('\nDigital Implementation Requirements:\n');
fprintf('Memory locations needed: %d words\n', memory_words);
fprintf('Multiplications per sample: %d\n', length(num_d) + length(den_d) - 1);
fprintf('Additions per sample: %d\n', length(num_d) + length(den_d) - 2);

% Generate C-style implementation code
fprintf('\nC Implementation Template:\n');
fprintf('// Previous inputs (x) and outputs (y)\n');
for i = 1:length(num_d)-1
   fprintf('float x_%d = 0.0f;\n', i);
end
for i = 1:length(den_d)-1
   fprintf('float y_%d = 0.0f;\n', i);
end

fprintf('\n// Controller implementation function\n');
fprintf('float compute_control(float error) {\n');
fprintf('    float output = %.4e * error;\n', num_d(1));
for i = 1:length(num_d)-1
   fprintf('    output += %.4e * x_%d;\n', num_d(i+1), i);
end
for i = 1:length(den_d)-1
   fprintf('    output -= %.4e * y_%d;\n', den_d(i+1), i);
end

fprintf('\n    // Update delay lines\n');
```

```
for i = length(num_d)-1:-1:2
   fprintf('    x_%d = x_%d;\n', i, i-1);
end
if length(num_d) > 1
   fprintf('    x_1 = error;\n');
end
for i = length(den_d)-1:-1:2
   fprintf('    y_%d = y_%d;\n', i, i-1);
end
if length(den_d) > 1
   fprintf('    y_1 = output;\n');
end

fprintf('    return output;\n');
fprintf('}\n');
```