University of Colorado
Boulder

# Photovoltaic Power Electronics
# ECEA 5717 Close-Loop Photovoltaic Power Electronics Laboratory
# Closed-Loop Voltage Regulator

Kai I. Tam

December 03, 2021

## 1. Documentation of system design

- Give a block diagram of your feedback system, and document the transfer function of each block in your loop.

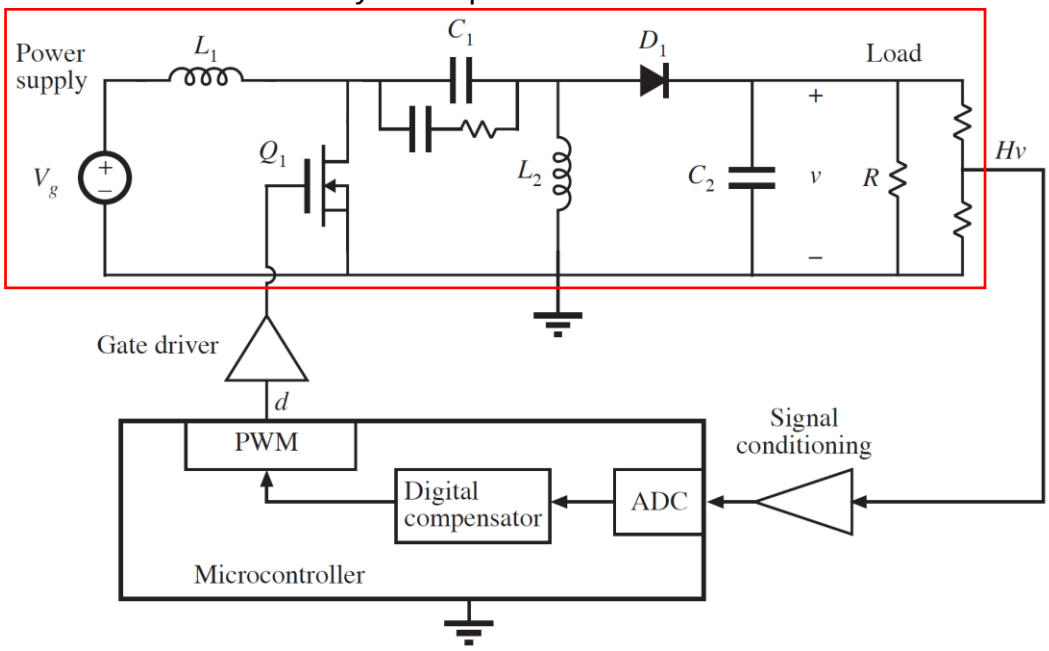



*Figure 1. General Block Diagram*

The circuit is effectively damped from Proj.4, we can utilize the Transfer function of Effective Buck Boost:

$$G_{vd-bb} = G_{vd0} \cdot \frac{(1-\frac{s}{\omega_z})}{(1+\frac{s}{Q\omega_o}+\frac{s^2}{{\omega_o}^2})} = \frac{V_g}{D'^2} \cdot \frac{1 - s\frac{L_1}{R}(\frac{D}{D'})^2}{1 + s\left(\frac{L_2 + (\frac{D}{D'})^2 L_1}{R}\right) + s^2 C_2 (L_2 + (\frac{D}{D'})^2 L_1)}$$

Effective Buck Boost DC Gain

$$20\log(\frac{V_g}{D'^2}) = 34.22dB$$

Effective Buck Boost Pole Frequency

$$f_o = \frac{\omega_o}{2\pi} = \frac{1}{2\pi\sqrt{C_2(L_2 + (\frac{D}{D'})^2 L_1)}} = 574Hz$$

Effective Buck Boost RHP Zero Frequency

$$f_z = \frac{\omega_z}{2\pi} = \frac{1}{2\pi \cdot [\frac{L_1}{R}\left(\frac{D}{D'}\right)^2]} = 17.8kHz$$

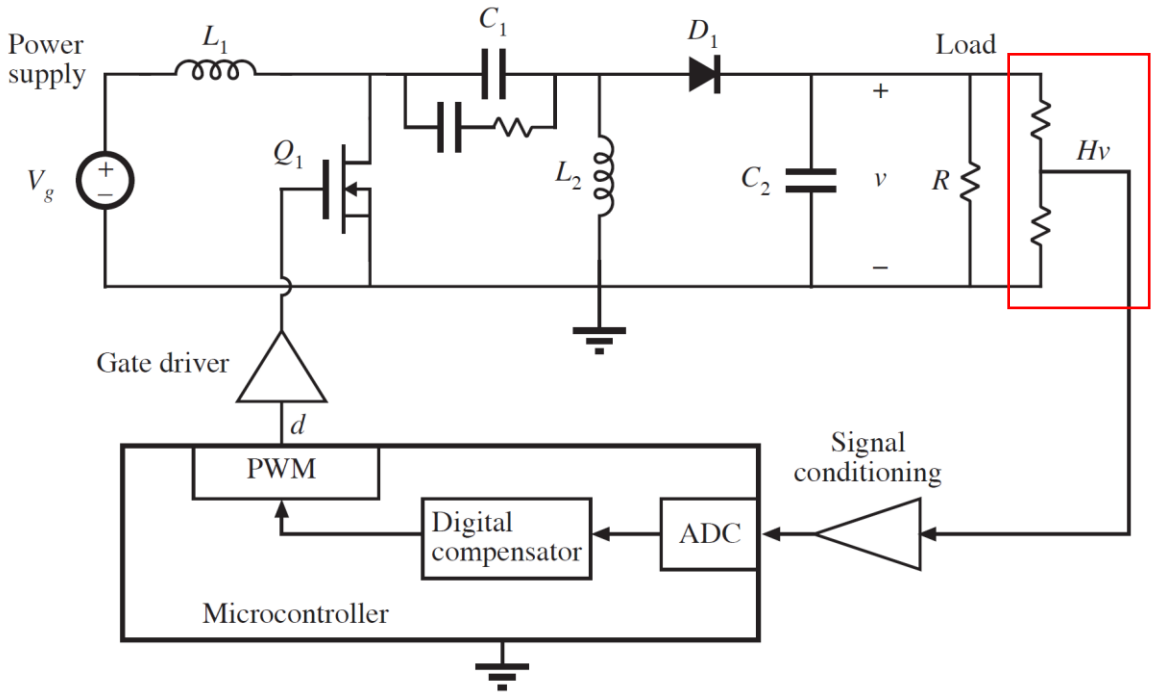



*Figure 2. H*

By choice, I chose the steady state ADC Voltage input to the microcontroller to be ½ of 3.3V at 12.5V, to get the best +/- range for the dynamic.

Desired ADC Voltage Feedback

$$ADC_{desire} = 12.5 \cdot H_v = \frac{3.3}{2} = 1.65V$$

Desired H gain

$$H_{desire} = \frac{1.65}{12.5} = 0.132$$

With available Closest Resistors values, I chose:

$$R_{h1} = 100k\Omega$$
$$R_{h2} = 12k\Omega$$

$$H_v = \frac{R_{h2}}{R_{h1} + R_{h2}} = \frac{12k}{100k + 12k} \approx 0.1072$$

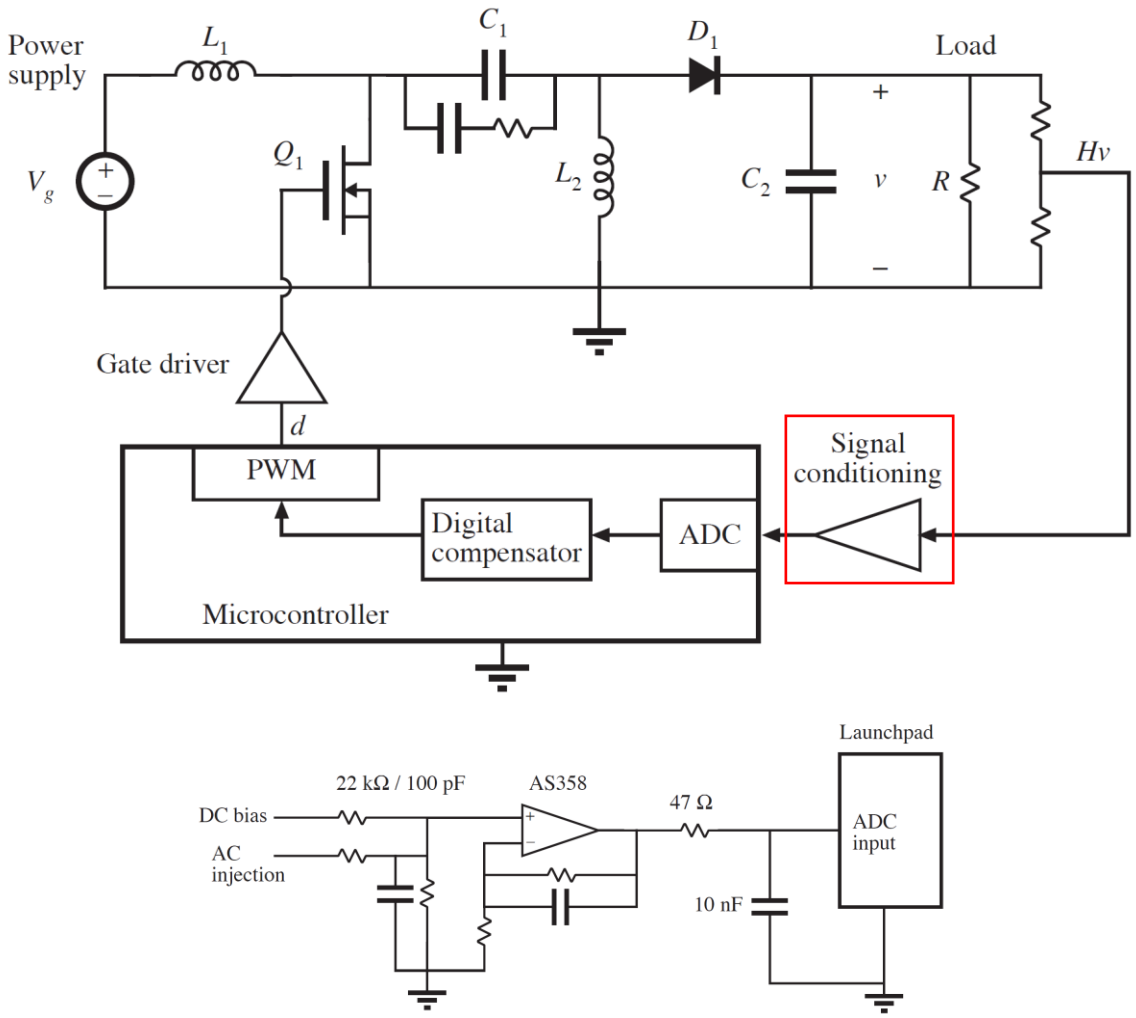



*Figure 3. Signal conditioning*

I decided to keep the 1:1 signal conditioning circuit from Project 4; it keeps the reading of AD2 smoother and cleaner. By connecting Feedback of H to DC bias, check Figure 9. for detail schematic. The 47Ω and 10nF will add another pole to the Loop gain but the pole frequency is all the way >300kHz which is out of our overall compensator frequencies and frequencies of interest. Therefore we will choose to ignore it and leave it out of the equation.

$$G_{signal\ condition} \approx 1$$

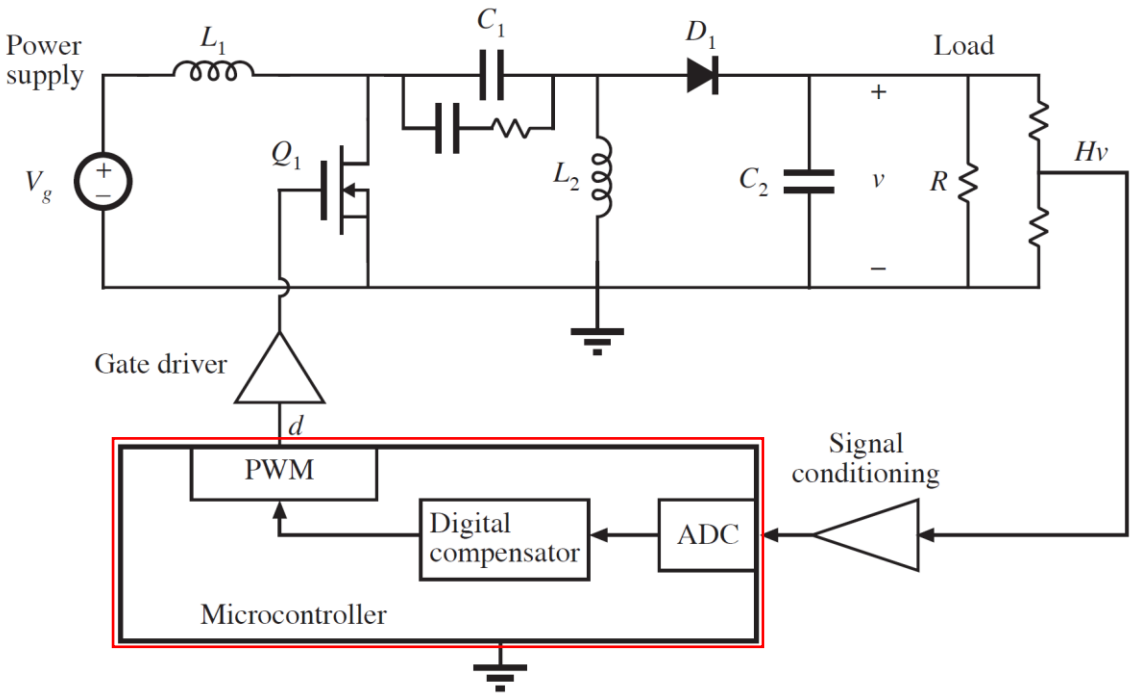



*Figure 4. Microcontroller*

The expected ADC value at steady state

$$ADC_{value,s.s} = 12.5 \cdot H_v \cdot \frac{2^n - 1}{3.3V} = 12.5 \cdot \frac{12}{112} \cdot \frac{2^{12} - 1}{3.3V} = 1662$$

Switching Frequency: 500kHz, therefore ePWM1 should be 120

$$ePWM1 = \frac{60MHz}{500kHz} = 120$$

With no compensator, steady state PWM output value should be:

$$PWM_{s.s.} = 120 \cdot D = 120 \cdot 0.425 \approx 51$$

With no compensator or compensator as 1, we should have a constant multiplier to archiving Vout = 12.5V:

$$Digital\ compensator\ as\ 1 = \frac{1}{Constant\ Multplier} = \frac{PWM_{s.s}}{ADC_{value,s.s}} = \frac{51}{1662} \approx \frac{1}{32.5882}$$

This can be used as a quick check for uncompensated Loop Gain. Actual ADC value can be obtained by injecting 12.5V to Vout, reading the real time average ADC value in CCS, reason is because the Vcc that the microcontroller it compares to does not sit on 3.3V perfectly. The actual ADC steady state value I obtained is 1360.

- Construct the magnitude and phase asymptotes of the loop gain of your system, and report the theoretical crossover frequency and phase margin.

Below are theoretical lossless values:
Uncompensated Loop Gain ($T_u$)

$$T_u(s) = \frac{H_v}{V_M} \cdot G_{vd}(s)$$

$$T_{u0} = \frac{H_v}{V_M} \cdot G_{vd0} = \frac{12}{112} \cdot \frac{1}{3.3} \cdot \frac{17}{(1-0.4237)^2} = 1.6621$$

$$T_{u0}(dB) = 20\log(T_{u0}) = 4.4129dB$$

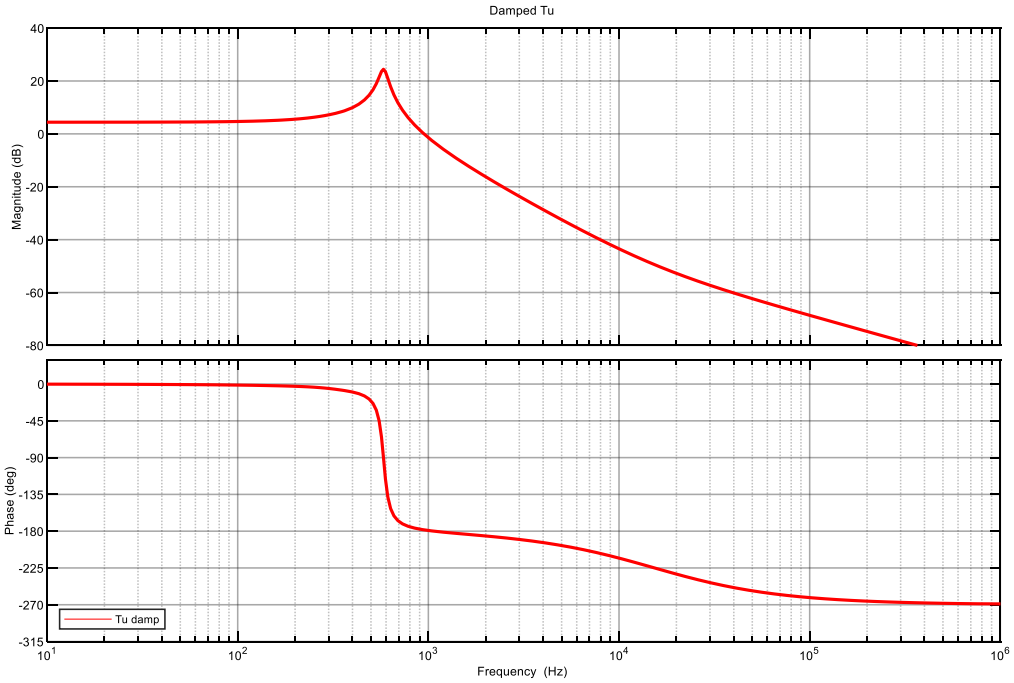



*Figure 5. MATLAB Uncompensated Loop Gain ($T_u$)*

| |
|---|
| Effective Buck Boost Pole Frequency: 574.2 Hz<br>Effective Buck Boost Zero Frequency: 17804.8 Hz<br>Damped Uncompensated Loop DC Gain : 1.6621<br>Damped Uncompensated Loop DC Gain (dB) : 4.4129 dB<br>Damped Uncompensated Loop Gain crossover: 948.0 Hz<br>Damped Uncompensated Loop Gain phase margin: 2 degrees |

Desired Crossover frequency ($f_c \geq 1kHz$)
Desired Crossover frequency ($\varphi_m \geq 52°$)

We want the crossover frequency as high as possible to achieve lower end gain, but since we have a RHP zero around 17.8kHz, we cannot go too high because compensator would not be able to achieve higher than 90º. By choice, I picked my Desired Crossover frequency $f_c = 2kHz$

Theoretical Compensator Zero ($f_{z\_cmp}$) and Compensator Pole ($f_{p\_cmp}$) at $\varphi_m \geq 52°$

$$f_{z\_cmp} = f_c\sqrt{\frac{1-\sin(\varphi_m)}{1+\sin(\varphi_m)}} = 2kHz\sqrt{\frac{1-\sin(52°)}{1+\sin(52°)}} \approx 631Hz$$

$$f_{p\_cmp} = f_c\sqrt{\frac{1+\sin(\varphi_m)}{1-\sin(\varphi_m)}} = 2kHz\sqrt{\frac{1+\sin(52°)}{1-\sin(52°)}} \approx 6343Hz$$

We want:

$$T_{uo}(\frac{f_o}{f_c})^2 G_{c0}\sqrt{\frac{f_{p\_cmp}}{f_{z\_cmp}}} = 1$$

The compensator DC gain ($G_{co}$) to achieve Crossover frequency $f_c = 2kHz$ is as follow:

$$G_{c0} = \frac{1}{T_{u0}}(\frac{f_c}{f_o})^2\sqrt{\frac{f_{z\_cmp}}{f_{p\_cmp}}} = \frac{1}{1.6621}(\frac{2000}{574})^2\sqrt{\frac{631}{6343}} = 2.3014$$

$$G_{c0}(dB) = 20\log(G_{c0}) = 7.2398dB$$

Also adding Compensator Inverted Zero to achieve higher lower frequency gain, we want it at least 10 times lower than $f_c = 2kHz$, so in would not interfere with the result, by choice,

$$f_{iz\_cmp} = 200Hz$$

Compensator

$$G_c(s) = G_{c0}\frac{(1+\frac{\omega_{L\_cmp}}{s})(1+\frac{s}{\omega_{z\_cmp}})}{(1+\frac{s}{\omega_{p\_cmp}})}$$

Compensated Loop gain

$$T(s) = \frac{H_v}{V_M}\cdot G_{vd}(s)\cdot G_c(s)$$

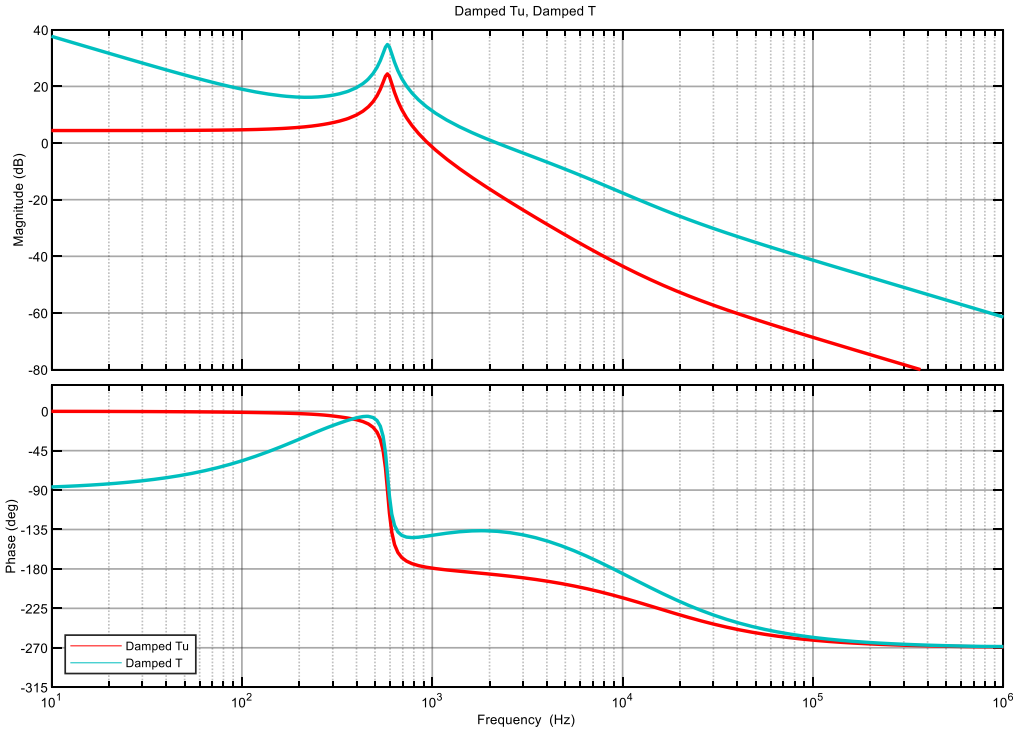



*Figure 6. Compensated and Uncompensated Loop Gain ($T_u$, T), non-optimized Phase Margin*

| |
|---|
| Damped Compensated Loop Gain crossover: 2191.4 Hz<br>Damped Compensated Loop Gain phase margin: 43 degrees |

The overall result deviated from the above predicted value, couple of reasons including the RHP zero is low which we can observe the phase start rolling off at 2kHz, we need to also take that into account. Adjusting Compensator: Decreasing Compensator Zero and Increasing Compensator Pole, to achieve the desired Phase Margin ($\varphi_m \geq 52°$), again, by choice:

$$f_{z\_cmp} = f_{z\_cmp} - 220Hz = 411Hz$$
$$f_{p\_cmp} = f_{p\_cmp} + 2000 = 8343Hz$$

Rework the Compensator DC gain

$$G_{c0} = \frac{1}{T_{u0}}\left(\frac{f_c}{f_o}\right)^2\sqrt{\frac{f_{z\_cmp}}{f_{p\_cmp}}} = \frac{1}{1.6621}\left(\frac{2000}{574}\right)^2\sqrt{\frac{411}{8343}} = 1.6192$$

$$G_{c0}(dB) = 20\log(G_{c0}) = 4.1862dB$$

Replotting the Resulting Loop Gain

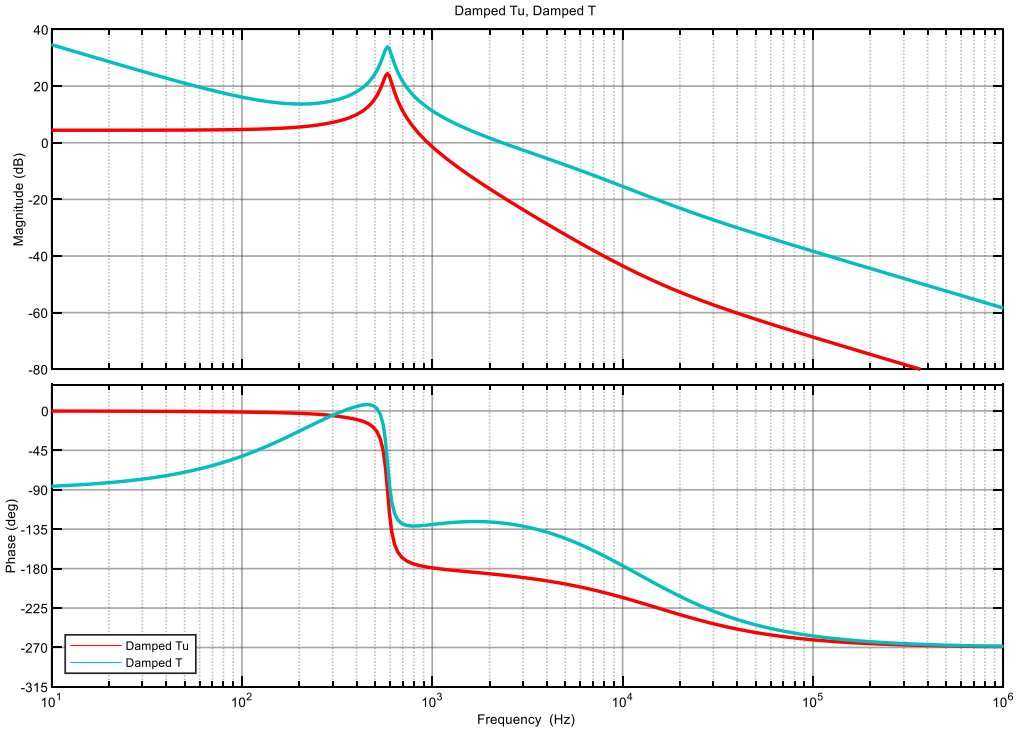



*Figure 7. Compensated and Uncompensated Loop Gain ($T_u$, T), optimized Phase Margin*

| |
|---|
| Damped Compensated Loop Gain crossover: 2335.0 Hz<br>Damped Compensated Loop Gain phase margin: 52 degrees |

We are now achieving $\varphi_m \geq 52°$, at Crossover Frequency $f_c \geq 1kHz = 2335Hz$, theoretically will need to slightly tune the result with actual readings because above are based on ideal value.

```
%% SEPIC Project 5 %%
s = tf('s'); % define s variable

%% Parameters (lossless)%%
Vg = 17;
V = 12.5;
R = 30;
D = V/(V+Vg);
L1 = 496e-06;
L2 = 485e-06;
C1 = 102e-06;
C2 = 102e-06;
fs = 500e+03;
Ts = 1/fs;
fsp = 22e+03;
Tsp = 1/fsp;
Vm = 3.3;
Rh1 = 100e+03;              %% Voltage divider for H
Rh2 = 12e+03;               %% Voltage divider for H
H = Rh2/(Rh1 + Rh2);
Hdb = mag2db(H);

%% Desired Parameters %%
fc = 2e+03;                                                 %% Desired Crossover Frequency
PhaseM = 55;                                                %% Desired Phase Margin

%% "Old" transfer function, with C1 open-circuited %%
wold = 1/sqrt(C2*(L2+L1*(D/(1-D))^2));                      %% Effective Buck-Boost Pole
Qold = R*sqrt(C2/(L2+L1*(D/(1-D))^2));                      %% Effective Buck-Boost Q
wz = (R/L1)*((1-D)/D)^2;                                    %% Effective Buck-Boost RHP zero
fo = wold/(2*pi);                                           %% Effective Buck-Boost Pole Frequency
fz = wz/(2*pi);                                             %% Effective Buck-Boost RHP zero Frequency
fprintf('Effective Buck Boost Pole Frequency: %4.1f Hz \n',fo);
fprintf('Effective Buck Boost Zero Frequency: %4.1f Hz \n',fz);

Gd0 = Vg/(1-D)^2;                                           %% Effective Buck-Boost DC gain
Gvd0_dB = mag2db(Gd0);                                      %% Effective Buck-Boost DC gain in dB
Gvdbb = Gd0*(1-s/wz)/(1 + (1/Qold)*(s/wold) + (s/wold)^2);  %% Effective Buck-Boost Gvd

%% Impedances %%
Z = 1/(s*C1);
L12 = L1*L2/(L1+L2);
wzN = (R/L12)*(1-D)^2/D;
ZN = s*(L1+L2)*(1-s/wzN)/(1-s/wz);
wozD = (1-D)/sqrt(C2*L12);
QzD = R*(1-D)*sqrt(C2/L12);
ZD = s*(L1+L2)*(1 + (1/QzD)*(s/wozD) + (s/wozD)^2)/(1 + (1/Qold)*(s/wold) + (s/wold)^2);

%% Damping %%
Cb = 220e-6;                        %% Damping Capacitance
Rb = 2.5;                           %% Damping Resistance
Zb = Rb + 1/s/Cb;
Zdamp = minreal(Z * Zb)/(Z + Zb);

%% Correction factor %%
Correction = minreal((1 + ZN/Z)/(1 + ZD/Z));
CorrectionDamped = minreal((1 + ZN/Zdamp)/(1 + ZD/Zdamp));

%% Transfer funciton with C1 %%
Gvd = minreal(Gvdbb * Correction);
Gvd_damp = minreal(Gvdbb * CorrectionDamped);

%% Uncompensated Loop Gain T %%
Tu_damp = H*(1/Vm)*Gvd_damp;
```

*MATLAB Code 1. Naturally Damped SEPIC*

```
%% Compensator %%
fp_cmp = fc*sqrt((1+sind(PhaseM))/(1-sind(PhaseM)))+(2e+03);    %% Desired compensator Pole Frequency
fz_cmp = fc*sqrt((1-sind(PhaseM))/(1+sind(PhaseM)))-220;        %% Desired compensator Zero Frequency
fiz_cmp = 200;                                                  %% Desired compensator Inverted Zero Freq
wp_cmp = 2*pi*fp_cmp;
wz_cmp = 2*pi*fz_cmp;
wiz_cmp = 2*pi*fiz_cmp;
fprintf('Compensator Pole Frequency: %4.1f Hz \n',fp_cmp);
fprintf('Compensator Zero Frequency: %4.1f Hz \n',fz_cmp);
fprintf('Compensator Inv. Zero Frequency: %4.1f Hz \n',fiz_cmp);

Tu0 = H*(1/Vm)*Gd0;
Tu0_dB = mag2db(Tu0);
fprintf('Damped Uncompensated Loop DC Gain : %2.4f \n',Tu0);
fprintf('Damped Uncompensated Loop DC Gain (dB) : %2.4f dB \n',Tu0_dB);

Gc0 = (1/Tu0)*((fc/fo)^2)*sqrt(fz_cmp/fp_cmp);
Gc0_dB = mag2db(Gc0);
fprintf('Compensator DC gain: %2.4f \n',Gc0);
fprintf('Compensator DC gain (dB): %2.4f dB \n',Gc0_dB);

Gc = Gc0*(1+s/wz_cmp)*(1+wiz_cmp/s)/(1+s/wp_cmp);

%% Compensated Loop gain %%
T_damp = Tu_damp*Gc;

%% Calculation of Crossover Frequency and Phase Margin %%
% Calculate Damped Uncompensated Loop Gain crossover frequency and phase margin
[Gmv,Pmv,Wcgv,Wcv] = margin(Tu_damp);
fcv = Wcv/2/pi;
fprintf('Damped Uncompensated Loop Gain crossover: %4.1f Hz \n',fcv);
fprintf('Damped Uncompensated Loop Gain phase margin: %4.0f degrees \n',Pmv);

% Calculate Damped Compensated Loop Gain crossover frequency and phase margin
[Gmv_d,Pmv_d,Wcgv_d,Wcv_d] = margin(T_damp);
fcv_d = Wcv_d/2/pi;
fprintf('Damped Compensated Loop Gain crossover: %4.1f Hz \n',fcv_d);
fprintf('Damped Compensated Loop Gain phase margin: %4.0f degrees \n',Pmv_d);

%% Plot %%
fmin=10; % minimum frequency = 10 Hz
fmax=100e4; % maximum frequency = 100 kHz

% Set Bode plot options
BodeOptions = bodeoptions;
BodeOptions.FreqUnits = 'Hz'; % we prefer Hz, not rad/s
BodeOptions.Xlim = [fmin fmax]; % frequency-axis limits
BodeOptions.Grid = 'on'; % include grid

% Frequency-response plot title and limits for G plots
BodeOptions.Title.String = 'Damped Tu, Damped T';
BodeOptions.Ylim = {[-80,40];[-315,30]}; % magnitude and phase axes limits
BodeOptions.PhaseMatching = 'on';
BodeOptions.PhaseMatchingFreq = 1;
BodeOptions.PhaseMatchingValue = 0;
% Plot magnitude and phase responses of Gold and G
Gfigure = figure(1);
bode(Tu_damp,BodeOptions,'r'); % plot Tu damp
hold on;
bode(T_damp,BodeOptions,'c'); % plot T damp
hold off;

% The lines below are just to make the frequency-response plots look nicer
h = findobj(gcf,'type','line');
set(h,'LineWidth',2); % thicker line width
Axis_handles=get(Gfigure,'Children');
axes(Axis_handles(3)); % magnitude plot, thicker grid lines
ax = gca;
ax.LineWidth = 1;
ax.GridAlpha = 0.4;
axes(Axis_handles(2)); % phase plot, thicker grid lines
ax = gca;
ax.LineWidth = 1;
ax.GridAlpha = 0.4;

legend('Damped Tu','Damped T','Location','southwest');
```

*MATLAB Code 2. Compensated and Uncompensated Loop Gain ($T_u$, T)*

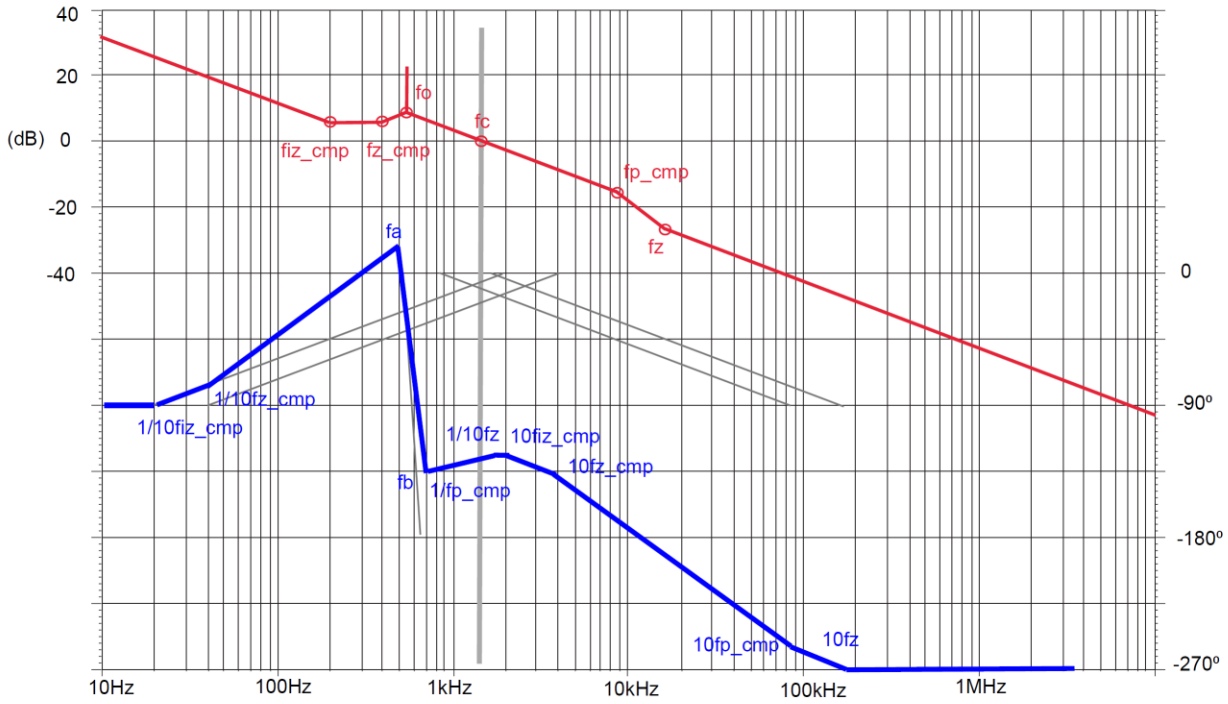



*Figure 8. Asymptotes of Compensated Loop Gain*

$$T(s) = \frac{H_v}{V_M} \cdot G_{vd}(s) \cdot G_c(s)$$

$$T(s) = T_{u0} \cdot \frac{(1 - \frac{s}{\omega_z})}{(1 + \frac{s}{Q\omega_o} + \frac{s^2}{{\omega_o}^2})} \cdot \frac{(1 + \frac{\omega_{L\_cmp}}{s})(1 + \frac{s}{\omega_{z\_cmp}})}{(1 + \frac{s}{\omega_{p\_cmp}})}$$

With lossless simulation, Q is relatively high.

$$Q \approx 11.0402 \approx 20.9dB$$
$$f_a = 10^{-\frac{1}{2Q}} \cdot f_o = 517.3Hz$$
$$f_b = 10^{\frac{1}{2Q}} \cdot f_o = 637.7Hz$$

By simple asymptotes assumptions:

$$f_c \approx 1.5kHz$$
$$\varphi_m \approx 55°$$

- Document the complete schematic of your final closed-loop circuit for Exp. 5.

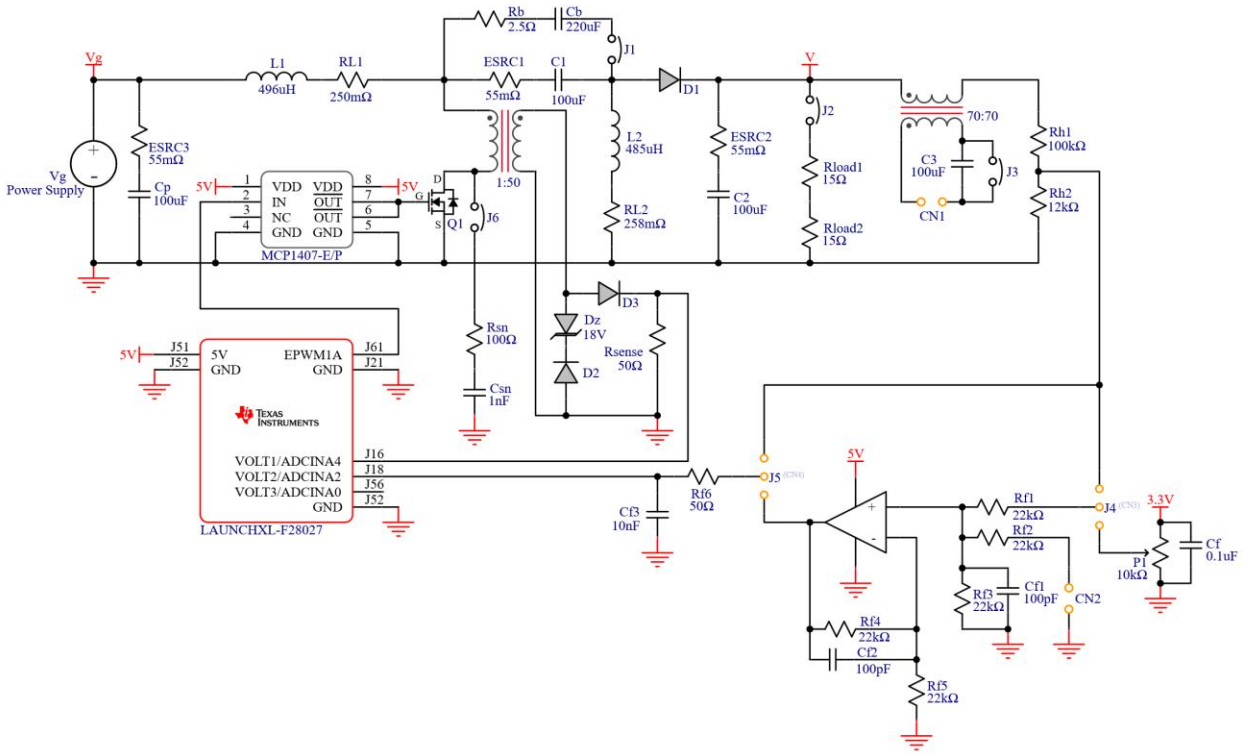



*Figure 9. Overall SEPIC Schematic*

I ended up rebuilding the whole circuit. To be much cleaner and tidier than the first prototype. Adding few jumpers for quick configurations.

J1: To configure with/without Natural Damped Circuit.
J2: Can be configure with default 30Ω Load for testing.
J3: Can be used to bypassing the DC blocking Capacitor for testing purpose.
J4: To configure the input of the signal conditioning circuit either using the potentiometer from Project 4 or directly feed from H.
J5: To configure either from the output of the signal conditioning circuit or directly feed from H.
J6: To configure with/without Snubber Circuit.

CN1: Connect to Network Analyzer for Loop Gain.
CN2: Connect to Network Analyzer for $G_{vd}$ in Project 4.

I also have the current sensing circuit from Project 3 connected to ADCINA4 for future development purpose. Isolation Transformer for Signal injection for Loop gain Measurement is built to have ratio 70:70.

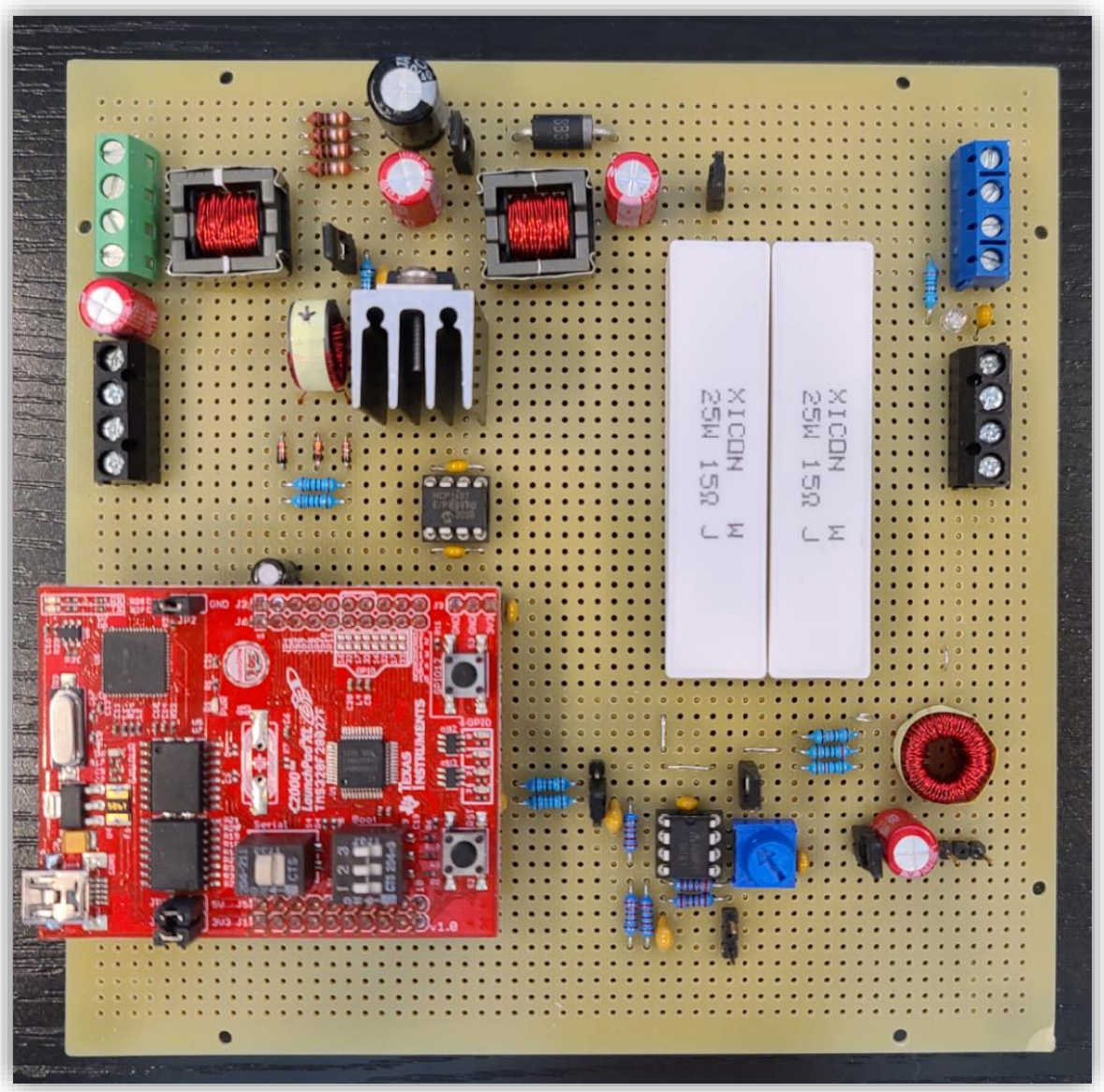



*Figure 10. My Circuit Build (TOP)*

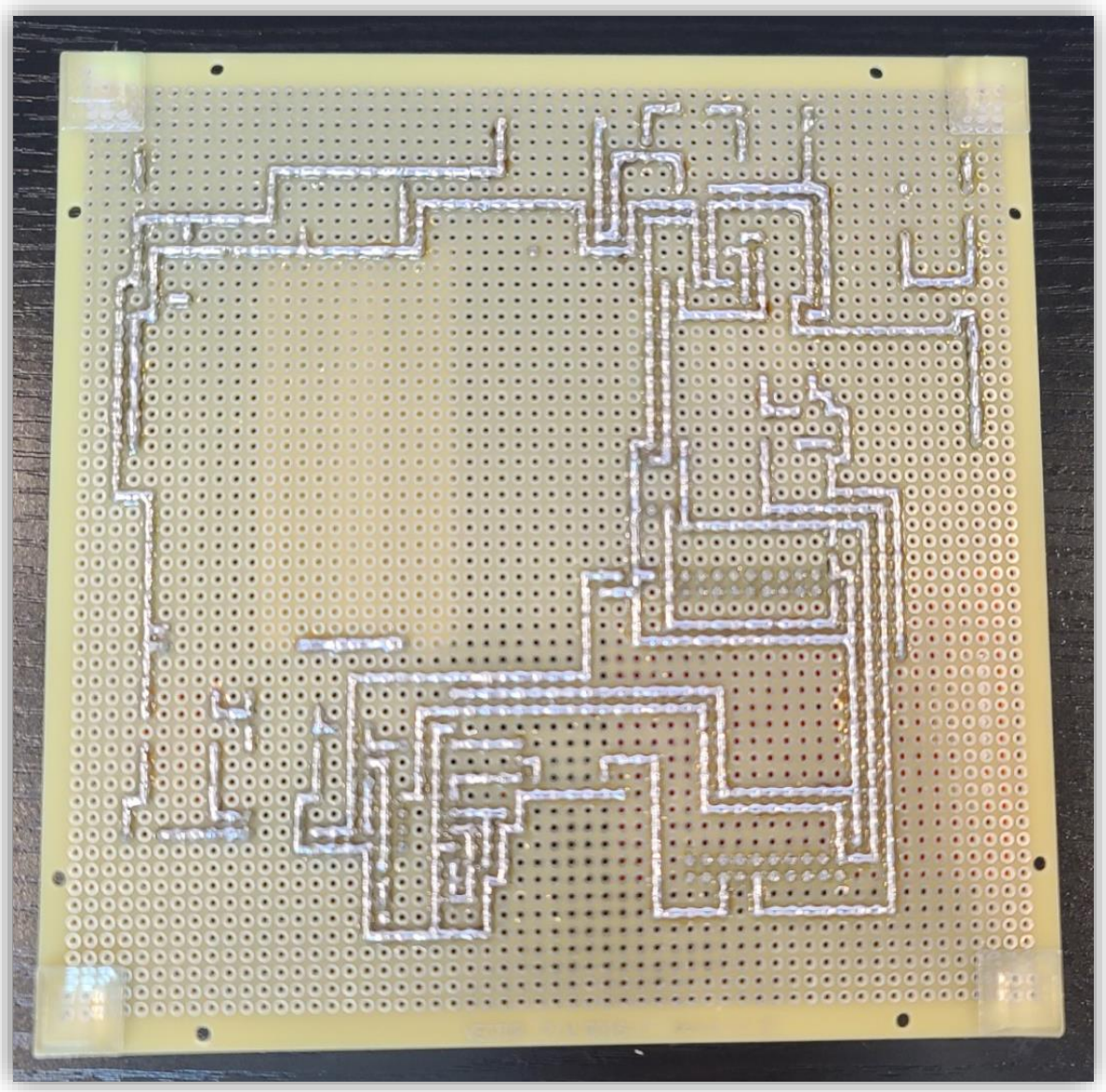


*Figure 11. My Circuit Build (BOTTOM)*

- Document the part of the microcontroller code that is relevant to your feedback controller (i.e., the interrupt service routine).

$$G_c(s) = G_{c0} \cdot \frac{(1 + \frac{\omega_{L\_cmp}}{s})(1 + \frac{s}{\omega_{z\_cmp}})}{(1 + \frac{s}{\omega_{p\_cmp}})}$$

Where from previously calculated:

$$\omega_{p\_cmp} = 2\pi f_{p\_cmp} = 2\pi \cdot 8343Hz = 52420.6Hz$$
$$\omega_{z\_cmp} = 2\pi f_{z\_cmp} = 2\pi \cdot 411Hz = 2582.4Hz$$
$$\omega_{L\_cmp} = 2\pi f_{L\_cmp} = 2\pi \cdot 200Hz = 1256.6Hz$$

Digital Compensator after z-transform

$$G_{cd}(z) = G_d \cdot \frac{(z - z_L)(z - z_z)}{(z - 1)(z - z_p)}$$

Note: $T_s$ from lecture is not switching period here, which is ADC sampling Period. In-order to find out Sampling Time, we want to toggle a GPIO high at the beginning of the ADC ISR and low at the end of the ADC ISR. I will relabel them as $T_{sp}$

$$T_{sp} \approx 45.41\mu S$$
$$f_{sp} \approx 22.02kHz$$

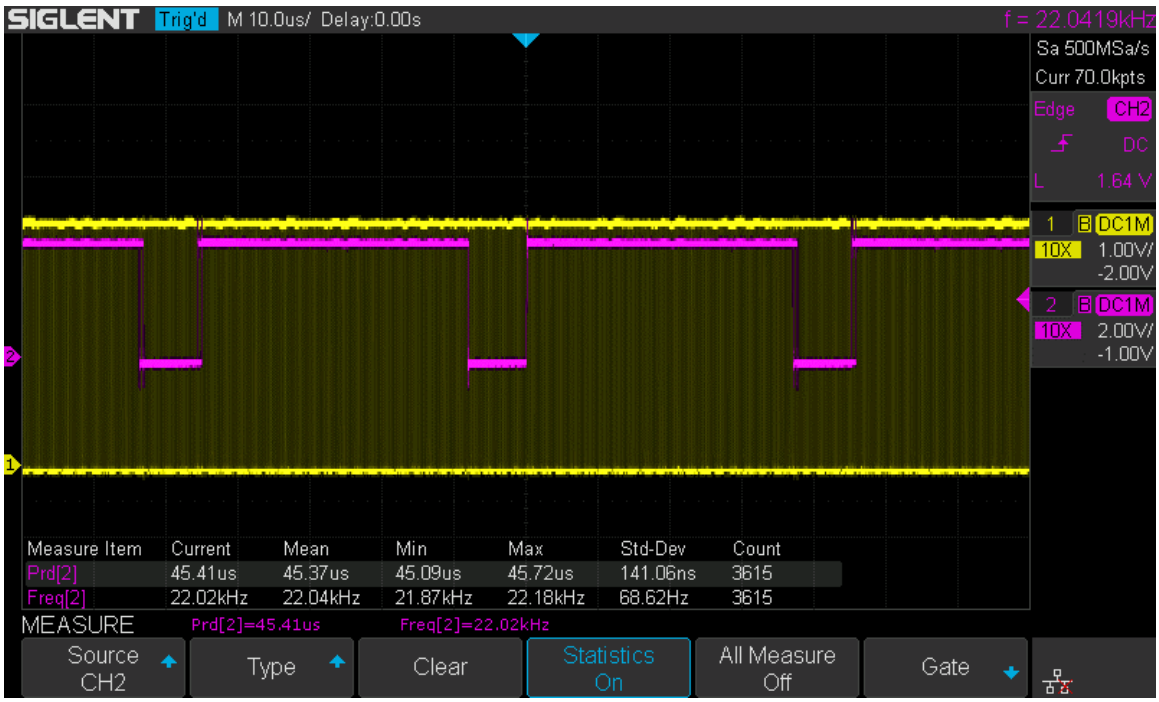



*Figure 12. Finding ADC Sampling Time*

```
%% Digital Compensator %%
Gd = Gc0*(wp_cmp/wz_cmp)*((1+wiz_cmp*Tsp/2)*(1+wz_cmp*Tsp/2))/(1+wp_cmp*Tsp/2);
zp = (1-wp_cmp*Tsp/2)/(1+wp_cmp*Tsp/2);
zz = (1-wz_cmp*Tsp/2)/(1+wz_cmp*Tsp/2);
zL = (1-wiz_cmp*Tsp/2)/(1+wiz_cmp*Tsp/2);

fprintf('Digital Compensator DC Gain: %4.4f \n',Gd);
fprintf('Digital Compensator Pole Gain: %4.4f \n',zp);
fprintf('Digital Compensator Zero Gain: %4.4f \n',zz);
fprintf('Digital Compensator Inv. Zero Gain: %4.4f \n',zL);


Digital Compensator DC Gain: 16.3485
Digital Compensator Pole Gain: -0.0873
Digital Compensator Zero Gain: 0.8892
Digital Compensator Inv. Zero Gain: 0.9445
```

*MATLAB Code 3. Digital Compensator MATLAB Computation Code*

$$G_d = G_{c0} \cdot \frac{\omega_{p\_cmp}}{\omega_{z\_cmp}} \cdot \frac{\left(1+\frac{\omega_{L\_cmp}T_{sp}}{2}\right)\left(1+\frac{\omega_{z\_cmp}T_{sp}}{2}\right)}{\left(1+\frac{\omega_{p\_cmp}T_{sp}}{2}\right)} = 16.3485$$

$$z_p = \frac{(1-\frac{\omega_{p\_cmp}T_{sp}}{2})}{(1+\frac{\omega_{p\_cmp}T_{sp}}{2})} = -0.0873$$

$$z_z = \frac{(1-\frac{\omega_{z\_cmp}T_{sp}}{2})}{(1+\frac{\omega_{z\_cmp}T_{s}}{2})} = 0.8892$$

$$z_L = \frac{(1-\frac{\omega_{L\_cmp}T_{sp}}{2})}{(1+\frac{\omega_{L\_cmp}T_{sp}}{2})} = 0.9445$$

Above value will then be scaled up 16 bits (65536) for Fixed Point Math.

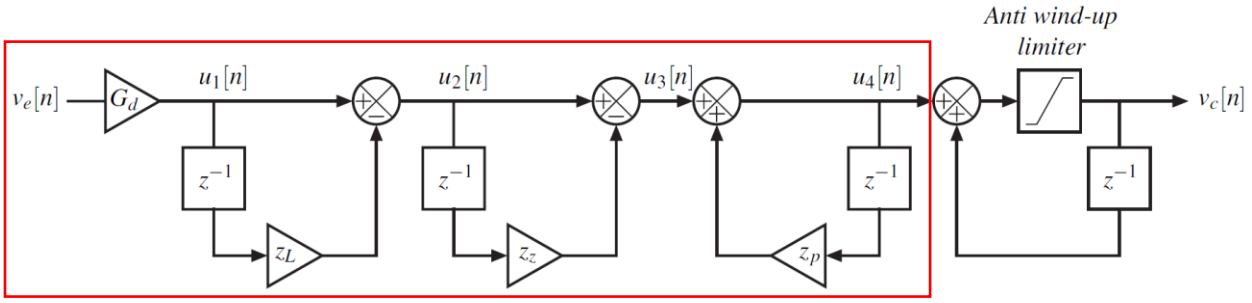
```
%% Digital Compensator Pre-scaled %%
zp_d = zp * 65536;
zz_d = zz * 65536;
zL_d = zL * 65536;

fprintf('Scaled Digital Compensator Pole Gain: %6.0f \n',zp_d);
fprintf('Scaled Digital Compensator Zero Gain: %6.0f \n',zz_d);
fprintf('Scaled Digital Compensator Inv. Zero Gain: %6.0f \n',zL_d);


Scaled Digital Compensator Pole Gain:  -5724
Scaled Digital Compensator Zero Gain:  58276
Scaled Digital Compensator Inv. Zero Gain:  61897
```

*MATLAB Code 4. Digital Compensator MATLAB Computation Code*

```
//
// Globals
//
        uint16_t                LoopCount;
        uint16_t                ConversionCount;
        uint16_t    volatile    Voltage2;
        uint16_t    volatile    DutyCycle;
        uint16_t    volatile    DutyCycle_p;

const   uint16_t    volatile    Vref    = 1360;             // Default output voltage reference

        int32_t     volatile    Dmax    = 156185000;        // Duty cycle maximum limit 0.75*120*26.4*65536
        int32_t     volatile    kscale  = 65536;            // Fixed-point math scale factor 2^16
        int32_t     volatile    ve      = 0;                // Error signal for period n
        int32_t     volatile    u1      = 65536;            // Default start state of u1[n],    pre-scaled
        int32_t     volatile    u1_o    = 65536;            // Default start state of u1[n-1],  pre-scaled
        int32_t     volatile    u2      = 65536;            // Default start state of u2[n],    pre-scaled
        int32_t     volatile    u2_o    = 65536;            // Default start state of u2[n-1],  pre-scaled
        int32_t     volatile    u3      = 65536;            // Default start state of u3[n],    pre-scaled
        int32_t     volatile    u3_o    = 65536;            // Default start state of u3[n-1],  pre-scaled
        int32_t     volatile    u4      = 65536;            // Default start state of u4[n],    pre-scaled
        int32_t     volatile    u4_o    = 65536;            // Default start state of u4[n-1],  pre-scaled
        int32_t     volatile    vc      = 65536;            // Default start state of vc[n],    pre-scaled
        int32_t     volatile    vc_o    = 65536;            // Default start state of vc[n-1],  pre-scaled

        int32_t     volatile    Gd      = 6;                // Digital Compensator DC gain Gd
        int32_t     volatile    zp      = -5724;            // Digital Compensator gain         pre-scaled
        int32_t     volatile    zz      = 58276;            // Digital Compensator gain         pre-scaled
        int32_t     volatile    zL      = 61897;            // Digital Compensator gain         pre-scaled
```

*Figure 13. Defining Global Variables*

For some reason if Gd set too high (ex. >8). The Bode Plot from Network Analyzer goes way off the chart, I leave it at 6 to get the optimum result and meeting both the Crossover Frequency and Phase Margin requirements. Which I will explain further on the upcoming pages.




$$u_1[n] = G_d v_e[n]$$
$$u_2[n] = u_1[n] - z_L u_1[n-1]$$
$$u_3[n] = u_2[n] - z_z u_2[n-1]$$
$$u_4[n] = u_3[n] + z_p u_4[n-1]$$
$$v_c[n] = u_4[n] + v_c[n-1]$$

```
//
//  adc_isr -
//
#pragma CODE_SECTION(adc_isr, "ramfuncs");
__interrupt void
adc_isr(void)
{
    GPIO_setHigh(myGpio, GPIO_Number_19);           // Toggle GPIO high to find out the Sampling Time.

    Voltage2 = ADC_readResult(myAdc, ADC_ResultNumber_2);          // Reading ADC

//  Digital Compensator

        ve = ((int32_t)Vref - (int32_t)Voltage2)* kscale;          // Compute error and Scale
        u1 = (Gd * ve);                                            // 32 bits
        u2 = u1 - ((int64_t)zL * (int64_t)u1_o) / kscale;          // 32 bits
        u3 = u2 - ((int64_t)zz * (int64_t)u2_o) / kscale;          // 32 bits
        u4 = u3 + ((int64_t)zp * (int64_t)u4_o) / kscale;          // 32 bits
        vc = u4 + vc_o;                                            // 32 bits

//  Setting up Maximum Duty Cycle Limit.

        if ((int32_t)vc > Dmax)
            {
                vc = Dmax;          // By default, Maximum Duty is limited to 75% conservatively
            }
        else
            {
                if (vc < 0)
                    {
                        vc = 0;     // Preventing if there is any negative number.
                    }
            }

//  Archive states

        u1_o = u1;      // Put u1[n] into u1[n-1] for next ISR
        u2_o = u2;      // Put u2[n] into u2[n-1] for next ISR
        u3_o = u3;      // Put u3[n] into u3[n-1] for next ISR
        u4_o = u4;      // Put u4[n] into u4[n-1] for next ISR
        vc_o = vc;      // Put vc[n] into vc[n-1] for next ISR

//  Set PWM duty cycle

        DutyCycle = (uint16_t)(vc/1735389);             // Scale vc down to 51 at steady state to achieve 12.5V
        PWM_setCmpA(myPwm, DutyCycle);                  // Setting PWM Duty Cycle
        PWM_setCmpAHr(myPwm, (unsigned int) 80 << 8);   // Value can be varied between 1-255

        GPIO_setLow(myGpio, GPIO_Number_19);            // Toggle GPIO low to find out the Sampling Time.

    ADC_clearIntFlag(myAdc, ADC_IntNumber_1);   //  Clear ADCINT1 flag reinitialize for next SOC
    PIE_clearInt(myPie, PIE_GroupNumber_10);    //  Acknowledge interrupt to PIE

    return;
}
```

*Figure 14. ADC ISR*

At Perfectly Regulated Steady State, pre-scaled vc should be:

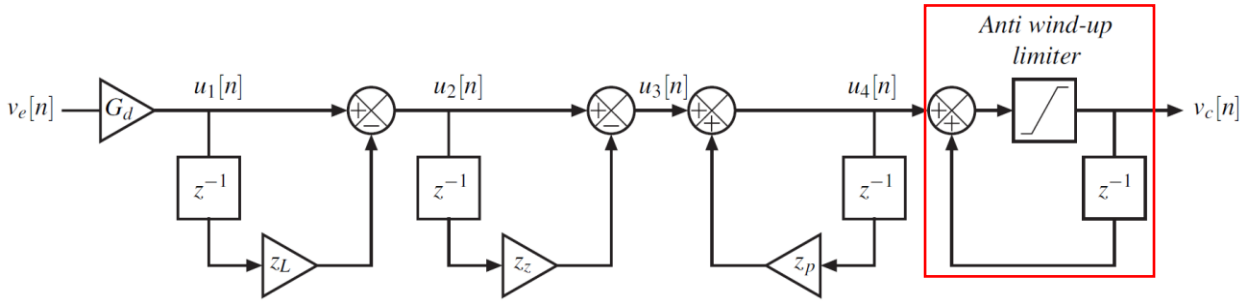
$$vc_{s.s} = ADC_{12.5V} \cdot kscale = 1360 \cdot 2^{16} = 89129000$$

Again, where 1360 is obtained by getting the ADC value at 12.5V. We need a Multiplier to scale vc back down to our Default Duty Cycle value to achieve 12.5Vout.

$$x = \frac{vc_{s.s.}}{PWM_{s.s.}} = \frac{8912900}{51.3597} \approx 1735389$$

## 2. Startup and large-signal stability

Describe what you did to make your closed-loop system start up. Document how you limited the maximum duty cycle, and the value of $D_{max}$ that you used. If you implemented soft start or current limiting, document what you did and the associated circuitry or code.




We can first make a DC sweep on LTSpice, see how D vs V(out) behaves.

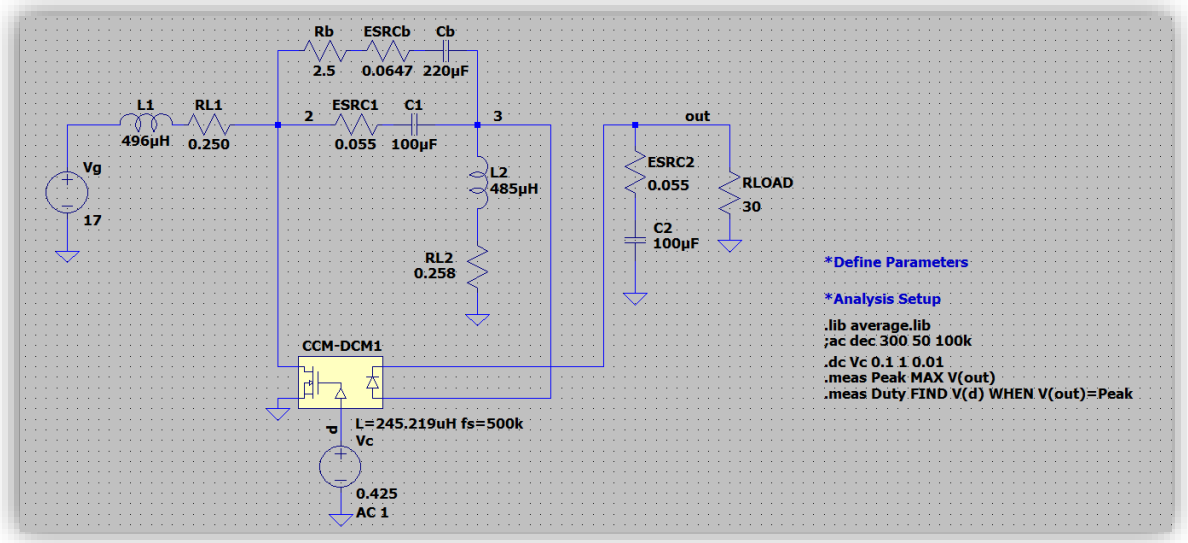



*Figure 15. SEPIC schematic in LTSpice*

```
peak: MAX(v(out))=92.4295 FROM 0.1 TO 1
duty: v(d)=0.92 at 0.92
```

By checking LTSpice Error Log, output has peaking at 0.92 Duty Cycle, then starting to roll off, therefore Duty Cycle shall not higher than 0.92. otherwise, the converter would not start if D were saturated.

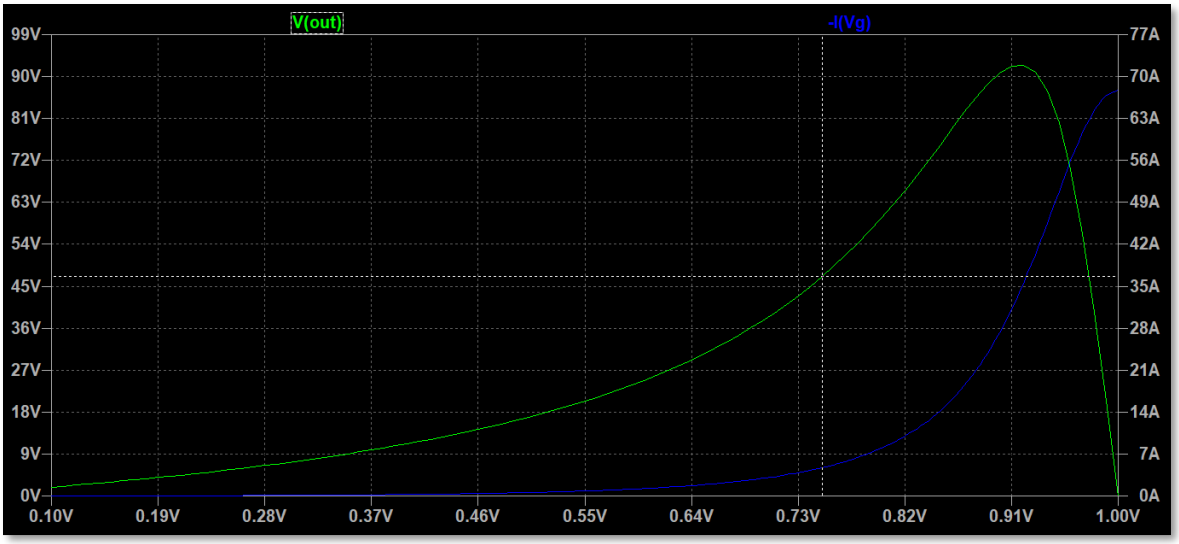
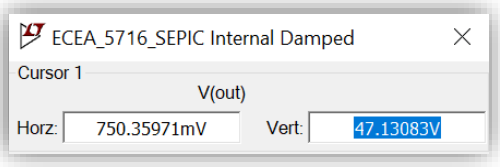



*Figure 16. D vs. V(out) in LTSpice*




Also, Since the output Capacitor C2 has a Voltage Rating of 50V, considering the worst-case scenario without damaging the circuit (ex, feedback open circuited), the output should not goes over 50V. With Duty Cycle around 0.75, output would be 47V.

```
      int32_t     volatile    Dmax    = 156185000;        // Duty cycle maximum limit 0.75*120*26.4*65536

//  Setting up Maximum Duty Cycle Limit.

       If ((int32_t)vc > Dmax)
           {
               vc = Dmax;          // By default, Maximum Duty is limited to 75% conservatively
           }
       else
           {
               if (vc < 0)
                   {
                       vc = 0;     // Preventing if there is any negative number.
                   }
           }
```

*Figure 17. Setting Maximum Duty Cycle in ADC ISR*

Inside ADC ISR, we set the limit to compare the output of the compensator vc, at this instant, where vc still pre-scaled, so Dmax should be:

$$D_{max} = 0.75 \cdot Period_{500kHz} \cdot \frac{ADC_{12.5V}}{PWM_{s.s.}} \cdot 2^{16} = 0.75 \cdot 120 \cdot \frac{1360}{51.3597} \cdot 65536 = 156185000$$

## 3. Loop gain measurement

Report your measured loop gain magnitude and phase. Construct magnitude and phase asymptotes of your loop gain, that follow all the standard rules for asymptote slopes and break frequency. Report the loop gain transfer function for your measurement, and give numerical values for the salient features (i.e., the corner and break frequencies, gains, Q, etc. as appropriate). Report the measured crossover frequency and phase margin. Compare your results with your theoretical predictions of part 1.

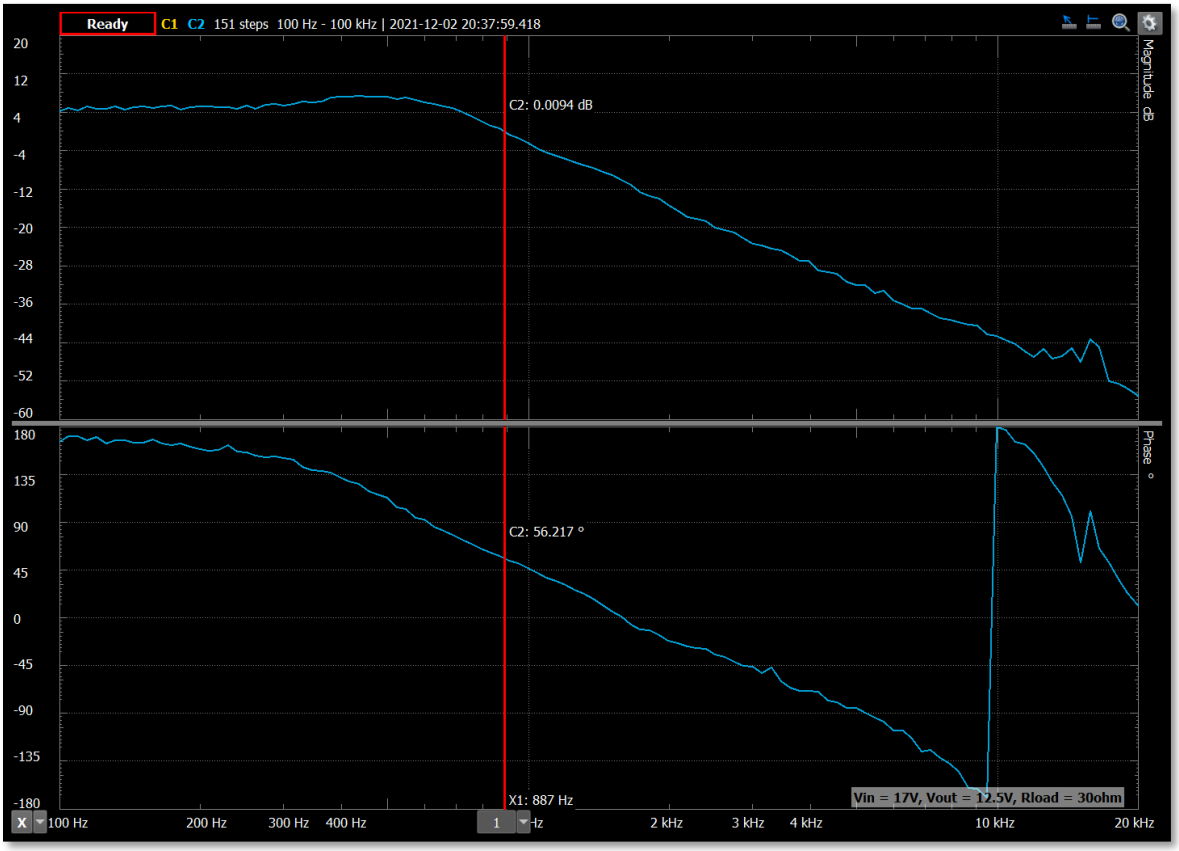



*Figure 18. Uncompensated Loop Gain ($T_u$)*

Previously predicted by MATLAB

| |
|---|
| Effective Buck Boost Pole Frequency: 574.2 Hz |
| Effective Buck Boost Zero Frequency: 17804.8 Hz |
| Damped Uncompensated Loop DC Gain (dB) : 4.4129 dB |
| Damped Uncompensated Loop Gain crossover: 948.0 Hz |
| Damped Uncompensated Loop Gain phase margin: 2 degrees |

By observation from Network Analyzer AD2

| |
|---|
| Effective Buck Boost Pole Frequency: ~580Hz |
| Effective Buck Boost Zero Frequency: ~20khz, the RHP Zero start rolling off at around 2kHz |
| Damped Uncompensated Loop DC Gain (dB) : ~4dB |
| Damped Uncompensated Loop Gain crossover: ~887.0 Hz |
| Damped Uncompensated Loop Gain phase margin: 56 degrees, because of lossy elements which makes the slope of -180º slower, which bringing the phase margin up. |

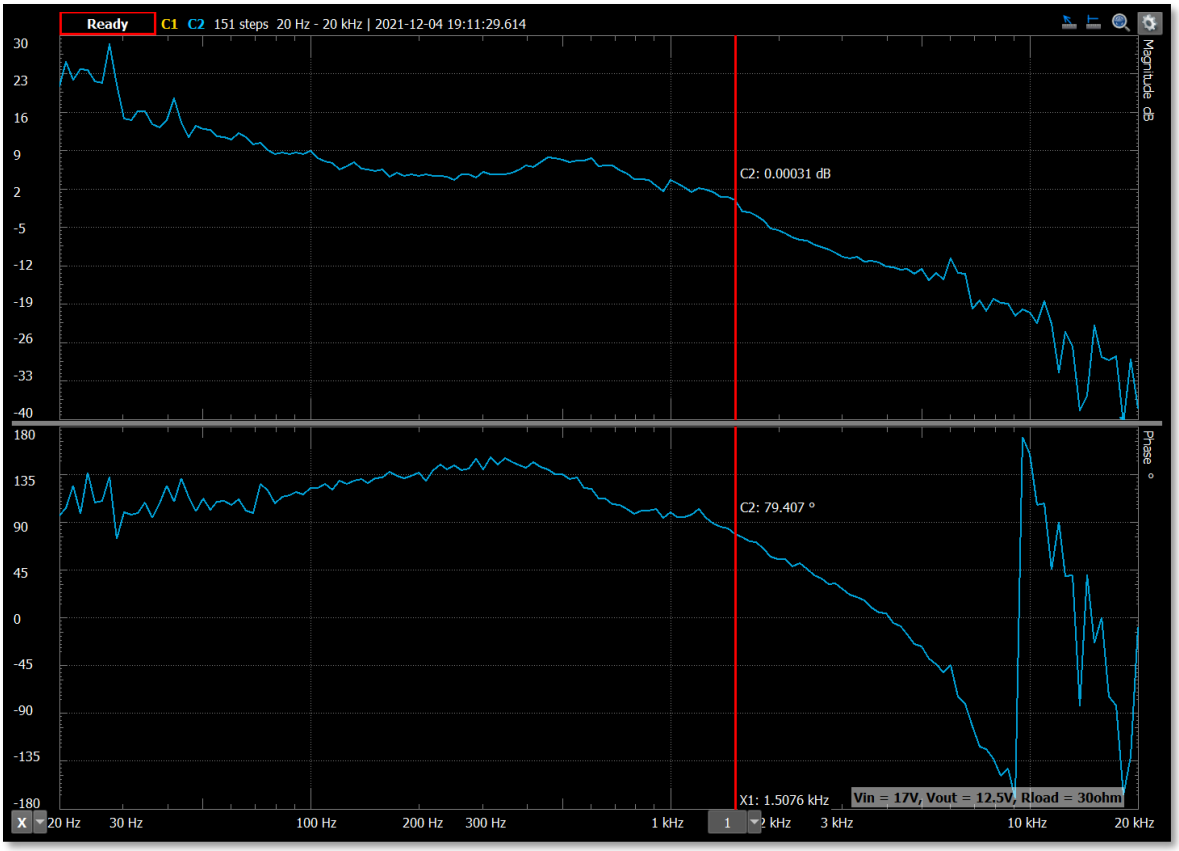



*Figure 19. Compensated Loop Gain (T)*

## Previously predicted MATLAB

Compensator Pole Frequency: 8343.2 Hz
Compensator Zero Frequency: 410.6 Hz
Compensator Inv. Zero Frequency: 200.0 Hz
Damped Compensated Loop Gain crossover: 2335.0 Hz
Damped Compensated Loop Gain phase margin: 52 degrees

## By observation from Network Analyzer AD2

Compensator Pole Frequency: ~10kHz
*We can observe it by combining the effect of pole and RHP zero (-90º) + (-90º) per decade, around 10kHz it appears to have a sharp roll off (-180º) on phase diagram.*

Compensator Zero Frequency: ~400Hz
*This one is hard to observe since the Original Pole of SEPIC and Compensated Inverted Zero and Pole all can have effect at this region because they are closely couple together. But we can observe a slight zero effect (+90º) around 1kHz on phase diagram.*

Compensator Inv. Zero Frequency: 200.0 Hz
*This one is easy to observe that the low frequency gain starts to roll back up around 200Hz, the jitter around low frequency might be due to the saturation of the isolation signal injection transformer.*

Damped Compensated Loop DC Gain (dB) : >25dB
Damped Compensated Loop Gain crossover: ~1.5k Hz
Damped Compensated Loop Gain phase margin: 79.4 degrees

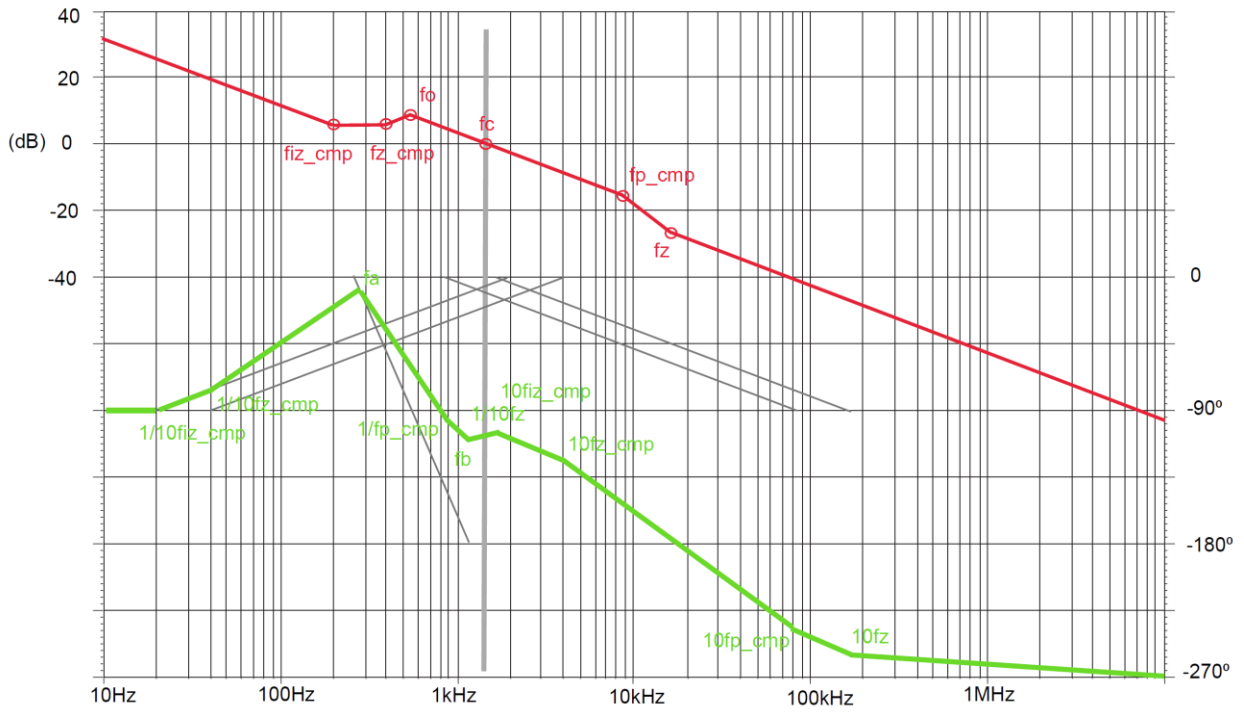



*Figure 20. Asymptotes of Compensated Loop Gain (Lower Q)*

$$T(s) = T_{u0} \cdot \frac{(1 - \frac{s}{\omega_z})}{(1 + \frac{s}{Q\omega_o} + \frac{s^2}{{\omega_o}^2})} \cdot \frac{(1 + \frac{\omega_{L\_cmp}}{s})(1 + \frac{s}{\omega_{z\_cmp}})}{(1 + \frac{s}{\omega_{p\_cmp}})}$$

Since the asymptotes obtained previously are based on lossless values, realize the real-life loop gain has lower Q because of its lossy elements, therefore, the (-180º) phase asymptotes would have a slower roll off slope, which theoretically brings up the phase margin even more, and the peaking of fa would be lower, (comparing to lossless which can slightly over 0º)

$$Q \approx 3.8dB \approx 1.55$$
$$f_a = 10^{-\frac{1}{2Q}} \cdot f_o \approx 275.8Hz$$
$$f_b = 10^{\frac{1}{2Q}} \cdot f_o \approx 1219.7Hz$$

$$T_{u0} \approx 4dB$$
$$f_{p_{cmp}} \approx 10kHz$$
$$f_{z\_cmp} \approx 400Hz$$
$$f_{L\_cmp} \approx 200Hz$$

The asymptotes assumptions are close to the real result once we have a lower Q value.

## 4. Regulation

For your voltage regulation with load current variation test, report the measured duty cycle, dc input voltage, dc output voltage, and load resistance for each step. Calculate the voltage regulation based on this data.

| Vg (V) | $V_{norminal}$ (V) | Load (Ω) | D (Mean) | Actual $V_{out}$ (V) |
|---|---|---|---|---|
| 17 | 12.5 | 15 | 0.450 | 12.553 |
| | | 30 | 0.437 | 12.550 |
| | | 60 | 0.429 | 12.545 |
| | | Open Circuit | 0.375 | 12.534 |

$$\frac{\Delta V}{V} = \frac{V_{max} - V_{min}}{V_{norminal}} \times 100\% = \frac{12.553 - 12.534}{12.550} \times 100\% = 0.151\%$$

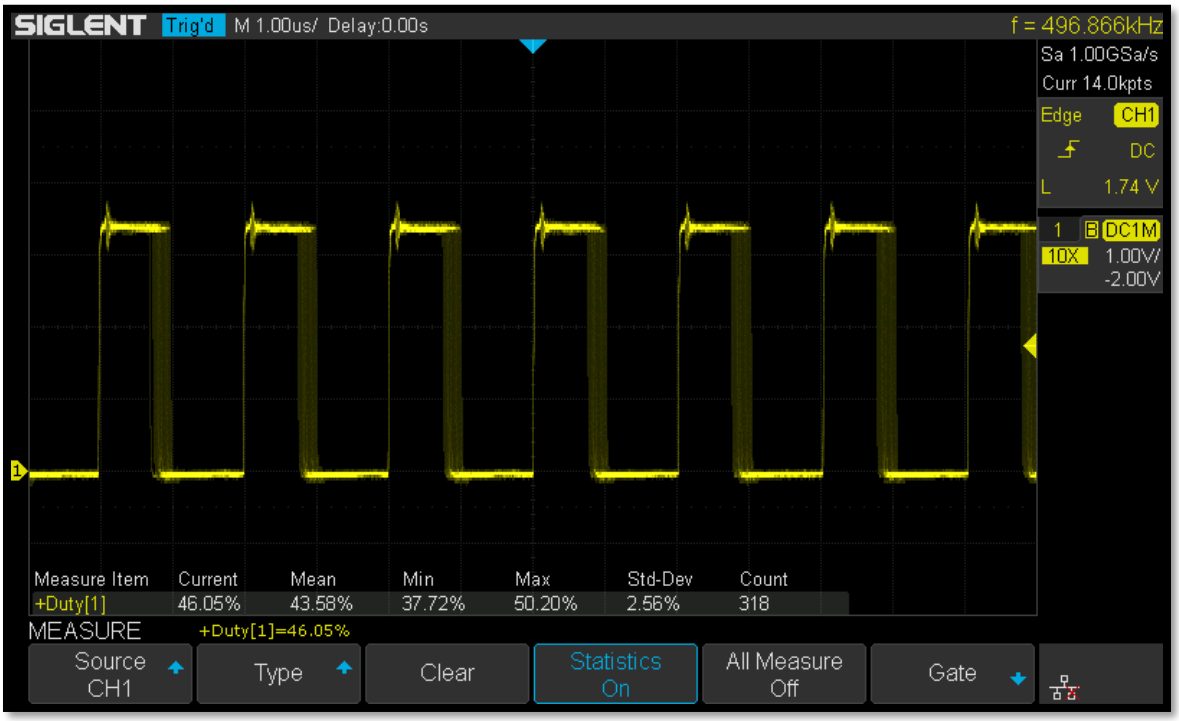



*Figure 21. Duty Cycle Measurement with Vg=17V Load = 30Ω*

For your voltage regulation with input voltage variation test, report the measured duty cycle, dc input voltage, and dc output voltage for each step. Calculate the voltage regulation based on this data.

| Vg (V) | $V_{norminal}$ (V) | Load (Ω) | D (Mean) | Actual $V_{out}$ (V) |
|---|---|---|---|---|
| 14.0 | 12.5 | 30 | 0.491 | 12.572 |
| 14.5 | | | 0.479 | 12.570 |
| 15.0 | | | 0.468 | 12.565 |
| 15.5 | | | 0.461 | 12.562 |
| 16.0 | | | 0.454 | 12.556 |
| 16.5 | | | 0.445 | 12.555 |
| 17.0 | | | 0.437 | 12.550 |
| 17.5 | | | 0.429 | 12.550 |
| 18.0 | | | 0.424 | 12.548 |
| 18.5 | | | 0.413 | 12.546 |
| 19.0 | | | 0.407 | 12.545 |
| 19.5 | | | 0.400 | 12.543 |
| 20.0 | | | 0.393 | 12.542 |
| 20.5 | | | 0.390 | 12.541 |
| 21.0 | | | 0.383 | 12.540 |

$$\frac{\Delta V}{V} = \frac{V_{max} - V_{min}}{V_{norminal}} \times 100\% = \frac{12.572 - 12.540}{V_{norminal}} \times 100\% = 0.255\%$$

By observing the Scope, the Duty Cycle changes constantly to compensate the Output variations. The control loop is doing its job.

Throughout the experiment, I observed there can be audible switching noise coming from the inductors. The loudness depends on the load condition, where when open circuit, it is nearly complete silence.

Another interesting found is if Digital Compensator Gd is more than 7 then Network Analyzer has lots of ripple, at calculated number 16.3485 is even worse, it is worth to go back and re-evaluate. Amplitude of the signal injections also affect the outcome. By default I set amplitude to 1V, graph will be heavily distorted lower than this value.

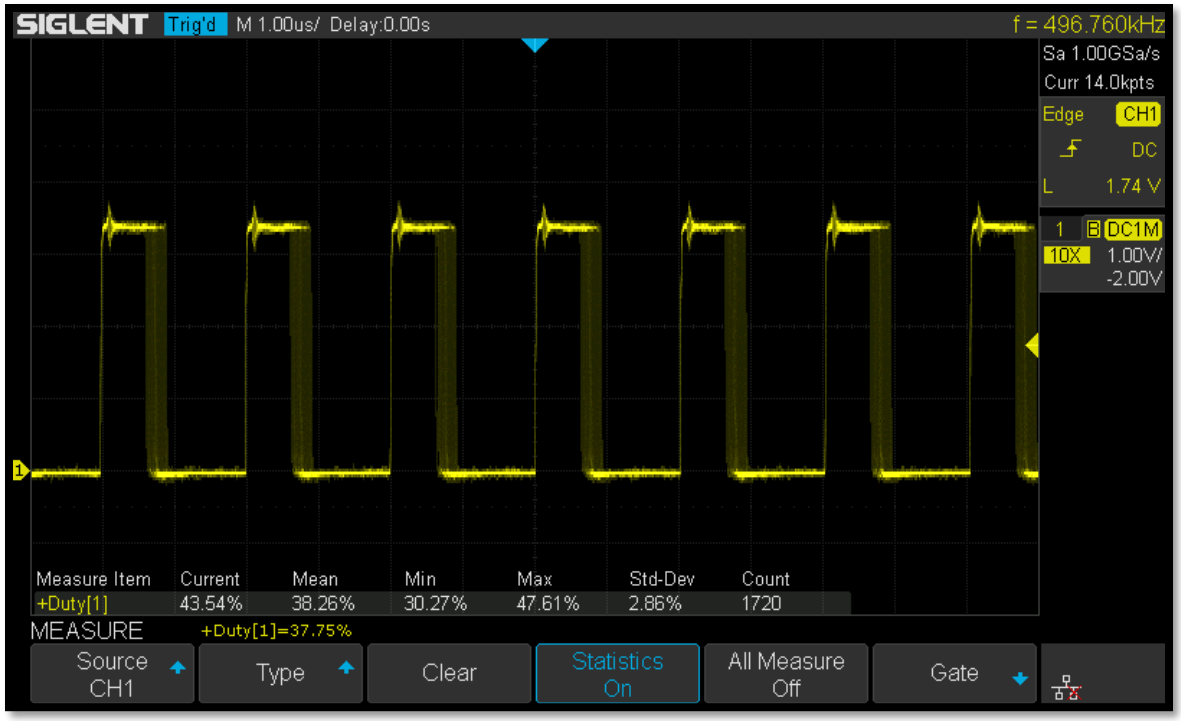



*Figure 22. Duty Cycle Measurement with Vg = 21V Load = 30Ω*

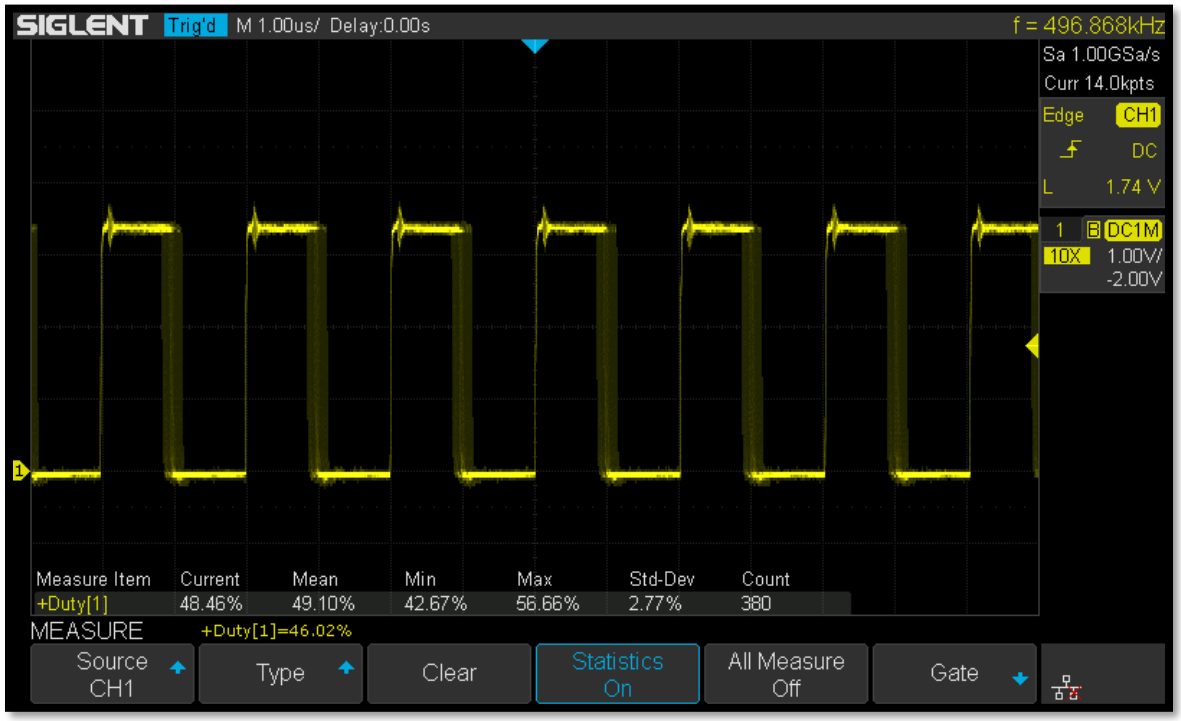



*Figure 23. Duty Cycle Measurement with Vg = 14V Load = 30Ω*

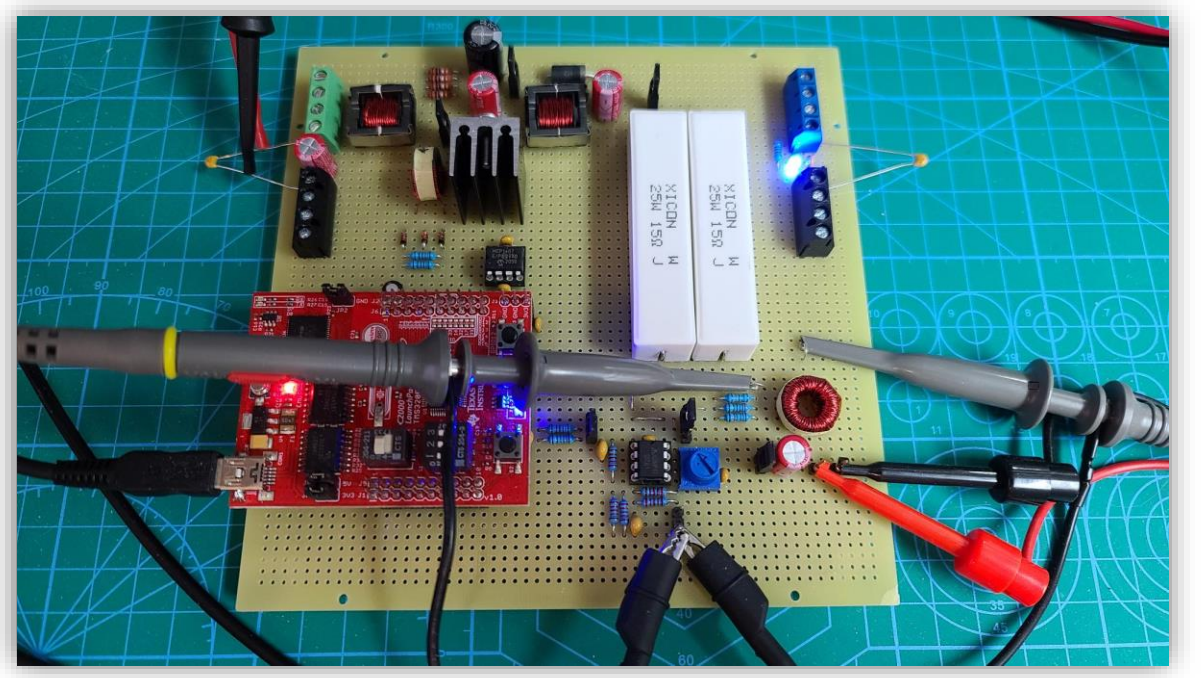



*Figure 24. Testing Set up*

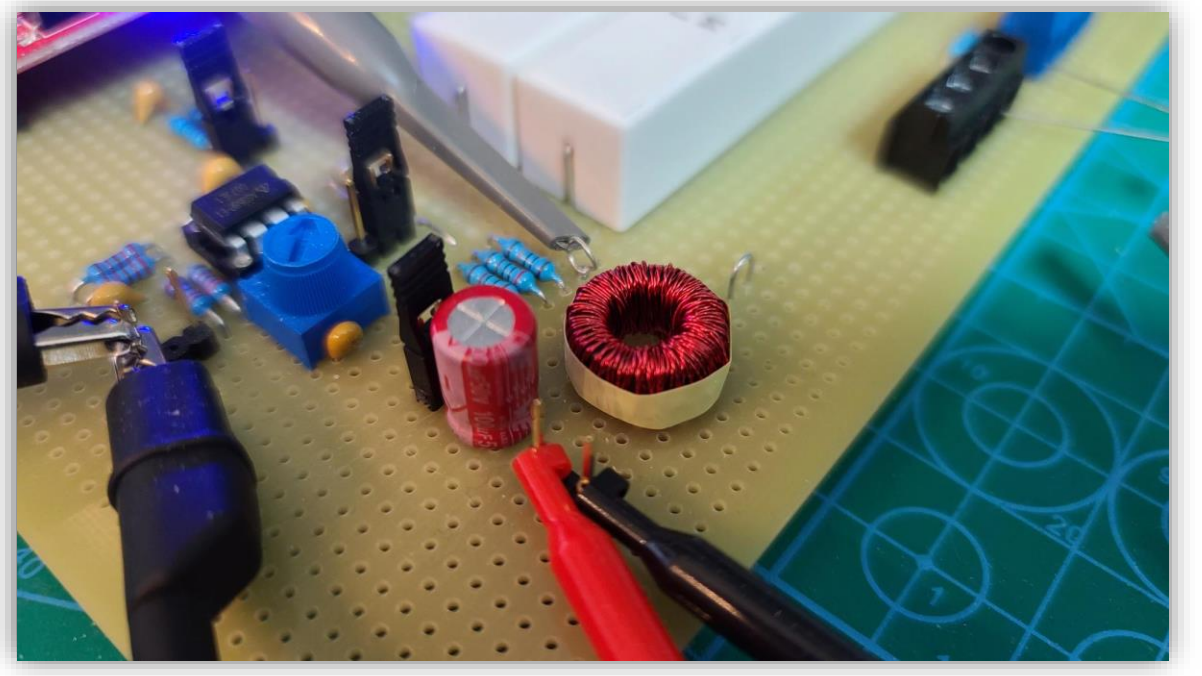


*Figure 25. 70:70 isolation signal injection transformer*

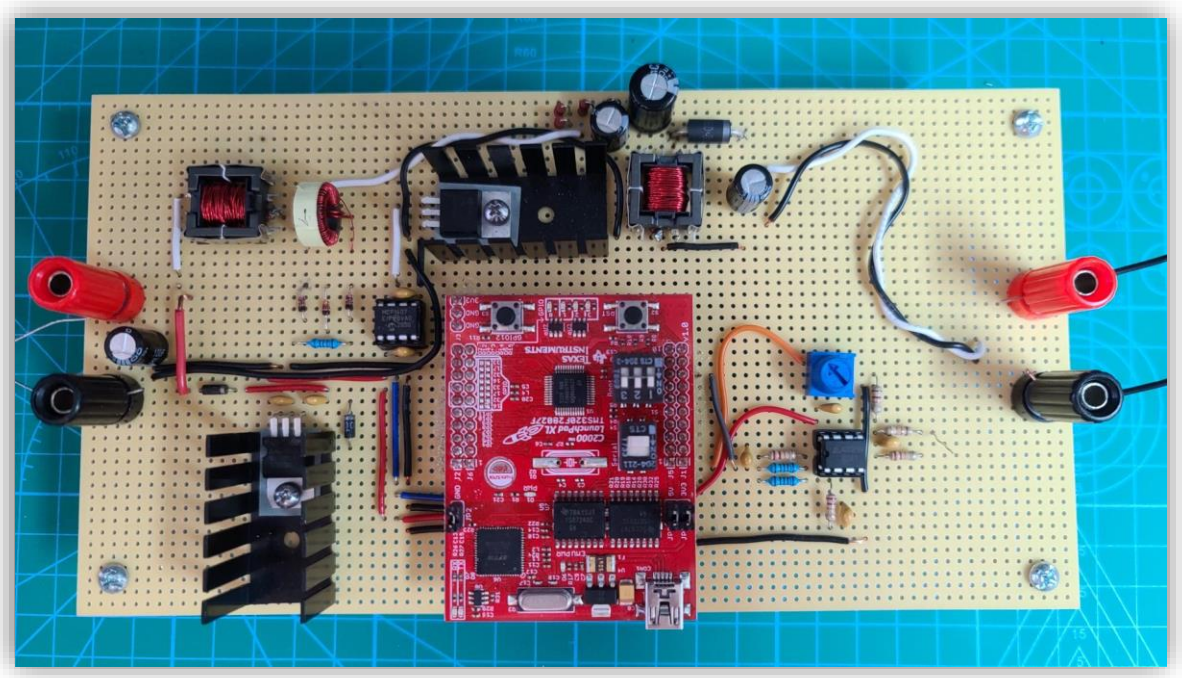


*Figure 26. First Prototype*

## Testing Equipment

- Fluke 289 True RMS Multimeter
- Siglent SDS1104X-E Digital Storage Oscilloscope 100MHz
- TekPower TP3005T DC Regulated Power Supply
- Digilent Analog Discovery 2 as Network Analyzer