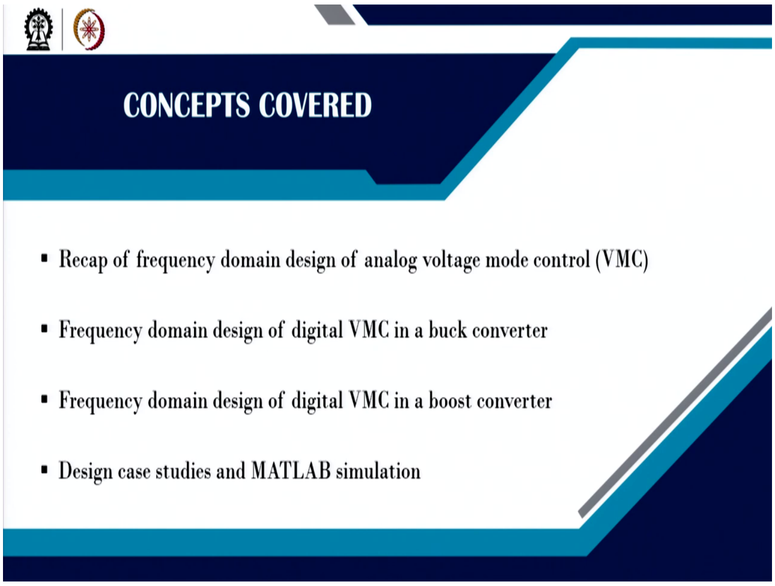
Digital Control in Switched Mode Power Converters and FPGA-based Prototyping

Prof. Santanu Kapat
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Module - 05
Frequency and Time Domain Digital Control Design Approaches
Lecture - 43
Design under Digital Voltage Mode Control – Frequency Domain Approaches

Welcome. In this class, we are going to talk about the design of digital voltage mode control using the frequency domain approach.

(Refer Slide Time: 00:31)

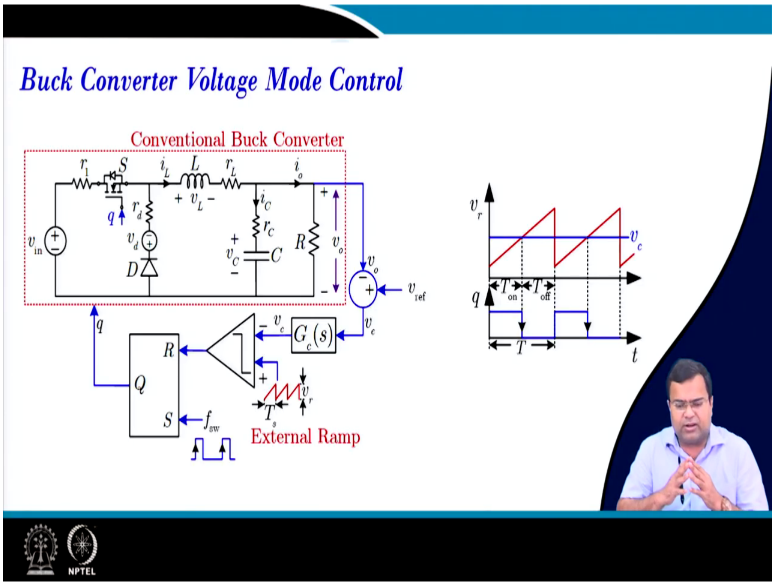


CONCEPTS COVERED

- Recap of frequency domain design of analog voltage mode control (VMC)
- Frequency domain design of digital VMC in a buck converter
- Frequency domain design of digital VMC in a boost converter
- Design case studies and MATLAB simulation

So, we will first recapitulate our frequency domain design of analog voltage mode control, then we want to design digital voltage mode control using a frequency domain approach in a buck converter and this is using you know the indirect method. Then we will also design digital voltage mode control in a boost converter using a frequency domain approach and finally, I want to show some design case studies and MATLAB simulation.

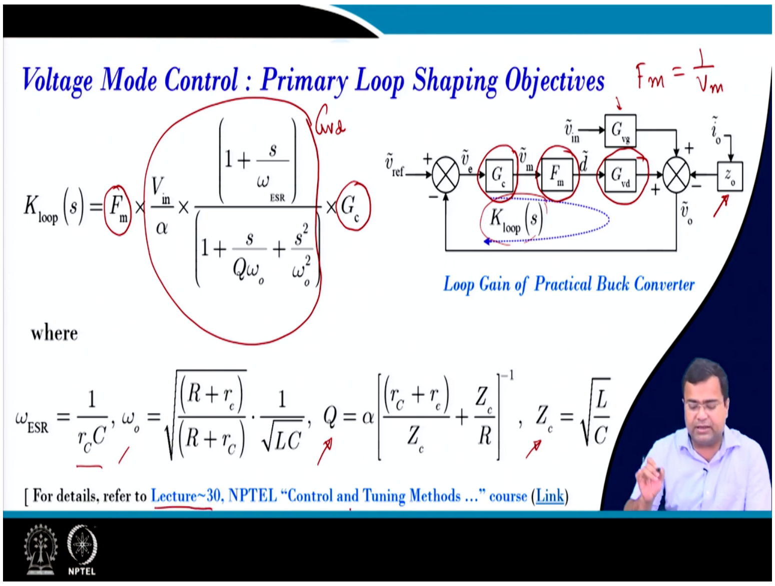
(Refer Slide Time: 00:58)



Buck Converter Voltage Mode Control

So, if we consider the buck converter voltage mode control, this is the analog voltage mode control that we have discussed multiple times.

(Refer Slide Time: 01:05)



Voltage Mode Control : Primary Loop Shaping Objectives

$F_m = \frac{1}{V_m}$

$$K_{loop}(s) = F_m \times \underbrace{\frac{V_{in}}{\alpha} \times \frac{\left(1+\frac{s}{\omega_{ESR}}\right)}{\left(1+\frac{s}{Q\omega_o}+\frac{s^2}{\omega_o^2}\right)}}_{G_{vd}} \times G_c$$

where

$$\omega_{ESR} = \frac{1}{r_C C},\ \omega_o = \sqrt{\frac{(R+r_c)}{(R+r_C)}}\cdot\frac{1}{\sqrt{LC}},\ Q = \alpha\left[\frac{(r_C+r_c)}{Z_c}+\frac{Z_c}{R}\right]^{-1},\ Z_c = \sqrt{\frac{L}{C}}$$

Loop Gain of Practical Buck Converter

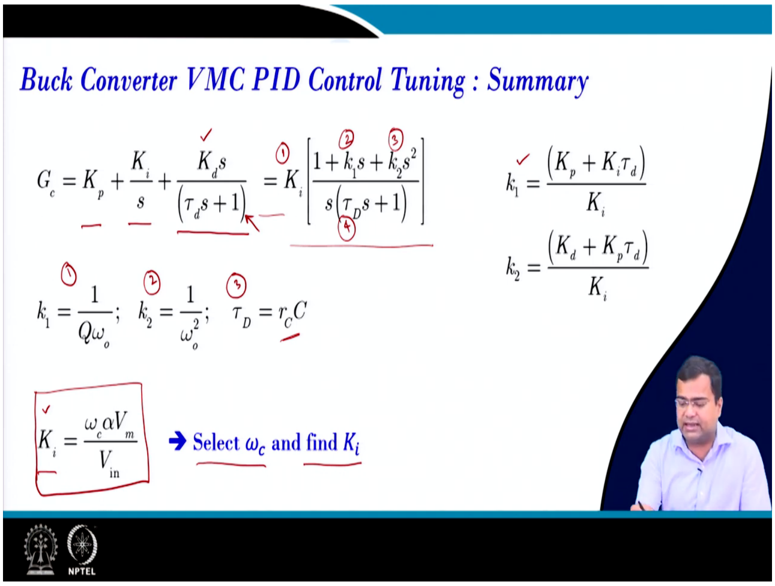
[For details, refer to Lecture~30, NPTEL “Control and Tuning Methods ...” course (Link)]

So, what are the primary loop-shaping objectives? So, if you take know small signal block diagram you know this small signal block diagram. This is our control of the output transfer function, this is our controller or compensator. This is a modulator gain and we know that F m is equal to 1 by V m. Where V m is the peak voltage of the sawtooth waveform and this is the open loop output impedance and this is the audio susceptibility.

So, we want to shape the loop transfer function, in such a way we can meet the desired crossover frequency as well as the phase margin. So, if we write the loop transfer function expression. So, this term is associated with this is nothing but our G vd. This is our modulator gain and this is the controller that we have to design. So, for the plane transfer function G vd, the ESR 0, we know all this you know the omega 0 which is the undamped natural frequency.

Then the Q factor as well as the characteristic impedance, all these expressions for the open loop practical synchronous buck converter that we have discussed in lecture 30 in our earlier NPTEL course which link is given here.

(Refer Slide Time: 02:23)



Buck Converter VMC PID Control Tuning : Summary

$$G_c = K_p + \frac{K_i}{s} + \frac{K_d s}{(\tau_d s + 1)} = K_i \left[\frac{1 + k_1 s + k_2 s^2}{s(\tau_D s + 1)} \right]$$

$$k_1 = \frac{(K_p + K_i \tau_d)}{K_i}$$

$$k_2 = \frac{(K_d + K_p \tau_d)}{K_i}$$

$$k_1 = \frac{1}{Q\omega_o}; \quad k_2 = \frac{1}{\omega_o^2}; \quad \tau_D = r_C C$$

$$\boxed{K_i = \frac{\omega_c \alpha V_m}{V_{in}}}$$

➔ Select ω_c and find K_i

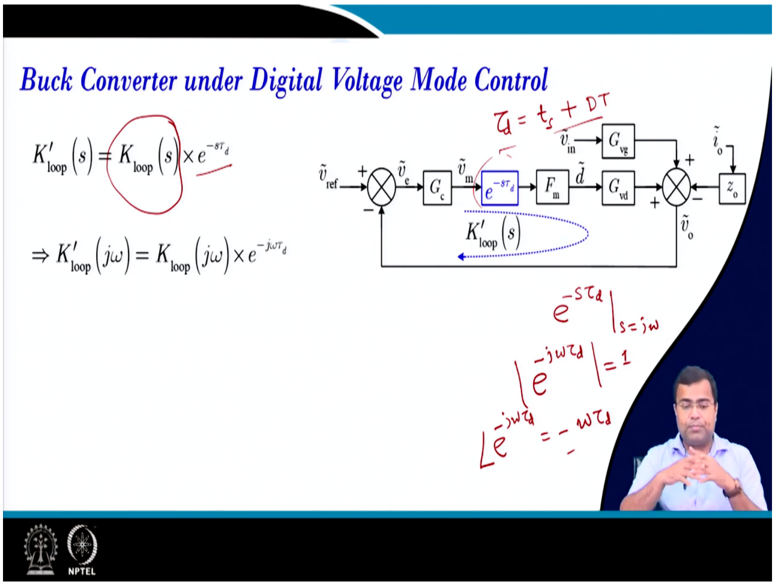
Now, our first objective is that we want to design a PID controller and if we take a practical PID controller, we know this is a proportional part. This is the integral term and this is the band-limited derivative where we have a derivative filter which is a low pass filter and this is a derivative gain, and this is a derivative gain. Then if we write you know the polynomial we can take K i out and these are the expression which is coming out from this left side.

And where k 1 can be and we are talking about the design of voltage mode control in a buck converter, where we are going to design using the conventional you know stable pole-zero cancellation. Where k 1 if we take k 1 equal to 1 by Q omega 0 and k 2 equal to 1 by omega 0 square and tau D which is nothing but you know r C into C that is the ESR you know the time constant due to the ESR of the capacitor.

Then the K i which I mean after this pole-zero cancellation compensation we will get that you know this K i term will be omega c alpha V m by V in and we need to select the crossover frequency and you know then we can find out the integral controller gain. So, in this controller, we have 1 unknown variable 2 3 and 4 tau D. Where we have one equation using pole-zero cancellation 1 will get 2 will get from here 3 will get from here. And these things are coming from our pole-zero after pole-zero compensation.

And the detailed steps we have again summarized in the previous lecture, in this course in lecture number 42; just the previous lecture. We are just taking that K i expression which we have already derived in the previous lecture. Now, k 1 and k 2 in this expression because we got the PID controller expression which in terms of numerator and denominator polynomial where k 1 equals in term it is a function of K p K i and tau d K d. So, these things are written.

(Refer Slide Time: 04:37)



Buck Converter under Digital Voltage Mode Control

$$K'_{\text{loop}}(s) = K_{\text{loop}}(s) \times e^{-s\tau_d}$$

$$\Rightarrow K'_{\text{loop}}(j\omega) = K_{\text{loop}}(j\omega) \times e^{-j\omega\tau_d}$$

$\tau_d = t_s + DT$

$e^{-s\tau_d}\Big|_{s=j\omega}$

$\left|e^{-j\omega\tau_d}\right| = 1$

$\angle e^{-j\omega\tau_d} = -\omega\tau_d$

Now, we want to under-voltage mode control we are talking about digital voltage mode control and we have discussed in I think in lecture number 40 that we have shown that if we incorporate that delay; that means, e to the power minus tau d and, this delay is due to our adhesive conversion time. That is our adhesive plus our DPWM delay which is D into T where D is a duty ratio and that we have discussed in lecture number 40.

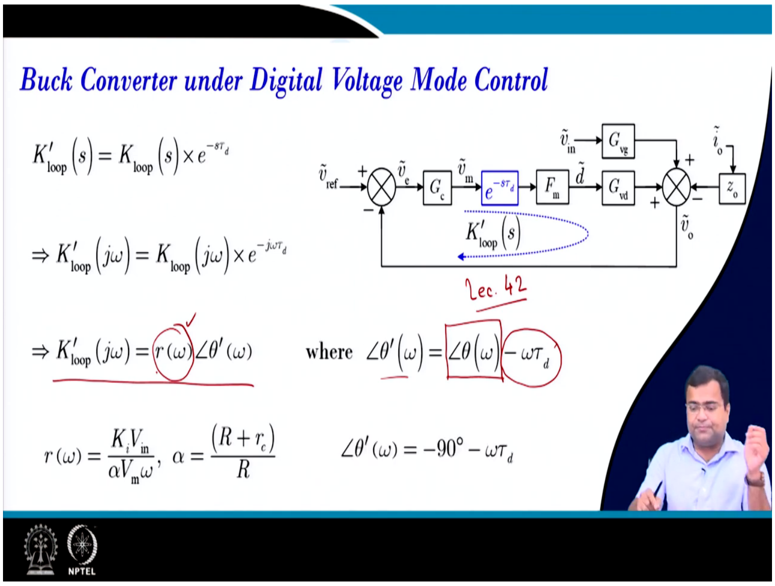
We have shown in lecture number 40 that if we incorporate this delay into the small-signal transfer function, then this small signal transfer function up to one-tenth of the switching frequency perfectly matches with the discrete-time small signal transfer function, which has been derived based on the exact you know I will say taking into account all the delay the discrete-time model.

But, we have shown if we derive the small signal transfer function using a continuous time approach and if we incorporate this delay, and then if we derive the discrete-time small signal model, they match reasonably accurately; almost accurate up to the one-tenth of the switching frequency which is our desired control bandwidth. Now if we consider the analog voltage mode control loop transfer function then if we incorporate this delay we can approximately get the loop transfer function under digital voltage mode control.

And we have shown that this delay amount is this. Next, this was the loop transfer function of the analog voltage mode control and we are just taking the delay term it can be shown it is well known that if you take e to the power minus s tau d this term you obtain the frequency response. That means, if you write s equal to j omega then e to the power minus j omega tau d, this term magnitude is 1. So, that means the delay will not affect the gain plot.

So, the gain plot remains unaltered it will be the same as the analog control gain plot but, the delay will have a phase because if you take e to the power minus j omega t tau d you will get simply minus; that means, minus omega tau d. So, it will give linear phase characteristics where the phase will proportionally vary with the frequency and the tau d will come into the picture and this can substantially reduce the phase margin compared to the analog control.

(Refer Slide Time: 07:14)



Buck Converter under Digital Voltage Mode Control

$$K'_{\text{loop}}\left(s\right) = K_{\text{loop}}\left(s\right) \times e^{-s\tau_d}$$

$$\Rightarrow K'_{\text{loop}}\left(j\omega\right) = K_{\text{loop}}\left(j\omega\right) \times e^{-j\omega\tau_d}$$

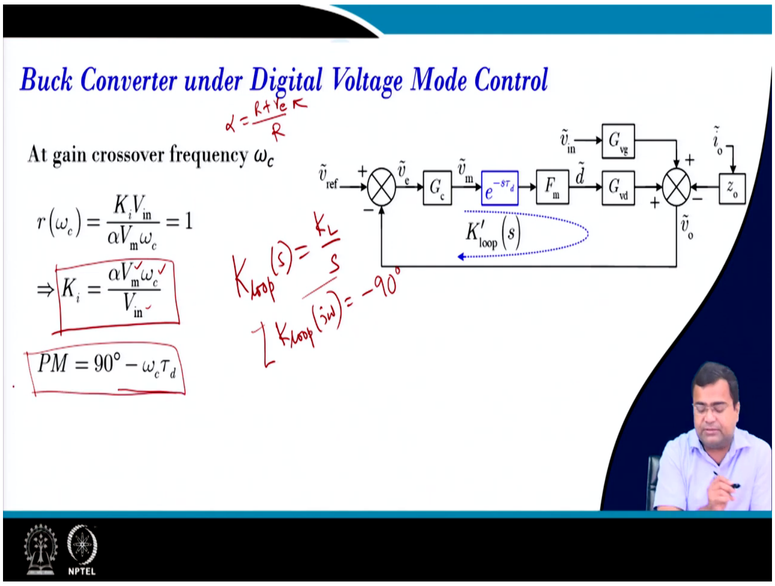
$$\Rightarrow \underline{K'_{\text{loop}}\left(j\omega\right) = r\left(\omega\right) \angle\theta'\left(\omega\right)} \quad \text{where } \angle\theta'\left(\omega\right) = \boxed{\angle\theta\left(\omega\right)} - \omega\tau_d$$

$$r\left(\omega\right) = \frac{K_i V_{\text{in}}}{\alpha V_{\text{m}} \omega}, \quad \alpha = \frac{\left(R + r_c\right)}{R} \qquad \angle\theta'\left(\omega\right) = -90^\circ - \omega\tau_d$$

So, now; that means, we got the loop transfer function with the voltage mode control with delay and this is the if written in terms of polar form. So, this is the magnitude expression and this is a phase expression. What is the magnitude expression? So, first of all, the phase expression will be the same as the phase expression of the analog controller which we have discussed in lecture number 42 in this course that we have discussed in the previous lecture.

In addition to that, we will get another phase lag which is minus omega tau d and r omega which is the same as the earlier which is K i V in by alpha V m omega, and this omega is now alpha equal to R equal to r equivalent. So, this is the r equivalent of r e or you can say r e. This is the re that expression was there. Now, at gain crossover frequency this magnitude will be 1. Because we are talking about the loop transfer function frequency response which will become 1 magnitude at gain crossover frequency and this is the phase expression.

(Refer Slide Time: 08:16)



Buck Converter under Digital Voltage Mode Control

$\alpha = \frac{R + r_e}{R}$

At gain crossover frequency ω_c

$$r(\omega_c) = \frac{K_i V_{in}}{\alpha V_m \omega_c} = 1$$

$$\Rightarrow \boxed{K_i = \frac{\alpha V_m \omega_c}{V_{in}}}$$

$$\boxed{PM = 90^\circ - \omega_c \tau_d}$$

$K_{loop}(s) = \frac{K_L}{s}$

$\angle K_{loop}(j\omega) = -90^\circ$

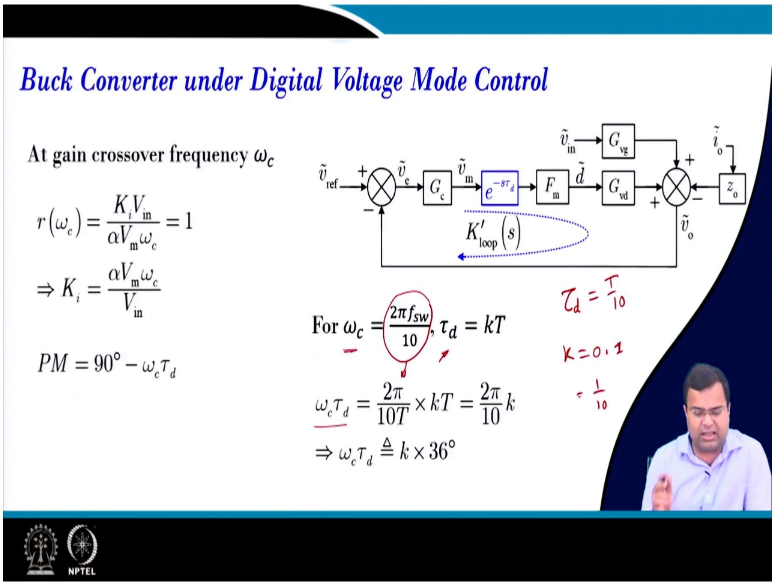
Now, how to design a gain crossover frequency? We know that r this omega c. That means if you write omega equal to omega c it will be 1. So, from this expression, we can get the integral gain in terms of alpha which is known. So, alpha is our R plus r e by R. It is known because r e is the equivalent resistance which is coming as a sum of the DC resistance of the inductor plus the RDS on and V m is the peak voltage of the sawtooth wave from.

Omega c is the crossover frequency and V in is the input voltage. Now the phase margin you know if we take digital voltage mode control is simply because we are talking about the PID controller and we are trying to obtain a first-order integral term. In analog control, we are trying to obtain the expression of the loop transfer function simply gain by s.

That means in the case of the analog loop transfer function we are trying to obtain some gain K L by S which is our desired objective using stable pole-zero cancellation.

So, in the case of the analog our phase of this you know if we take loop transfer function j omega, this phase will be simply minus 90 degree because it is just 1 by s and if you write the expression of phase margin it will be 180 minus 90. So, 90, but because of the delay you are getting an additional term. So, that will try to reduce the phase margin compared to analog control.

(Refer Slide Time: 09:52)



Buck Converter under Digital Voltage Mode Control

At gain crossover frequency ω_c

$$r(\omega_c) = \frac{K_i V_{in}}{\alpha V_m \omega_c} = 1$$

$$\Rightarrow K_i = \frac{\alpha V_m \omega_c}{V_{in}}$$

$$PM = 90^\circ - \omega_c \tau_d$$

For $\omega_c = \frac{2\pi f_{sw}}{10}$, $\tau_d = kT$

$$\omega_c \tau_d = \frac{2\pi}{10T} \times kT = \frac{2\pi}{10} k$$

$$\Rightarrow \omega_c \tau_d \triangleq k \times 36^\circ$$

$\tau_d = \frac{T}{10}$

$k = 0.1$

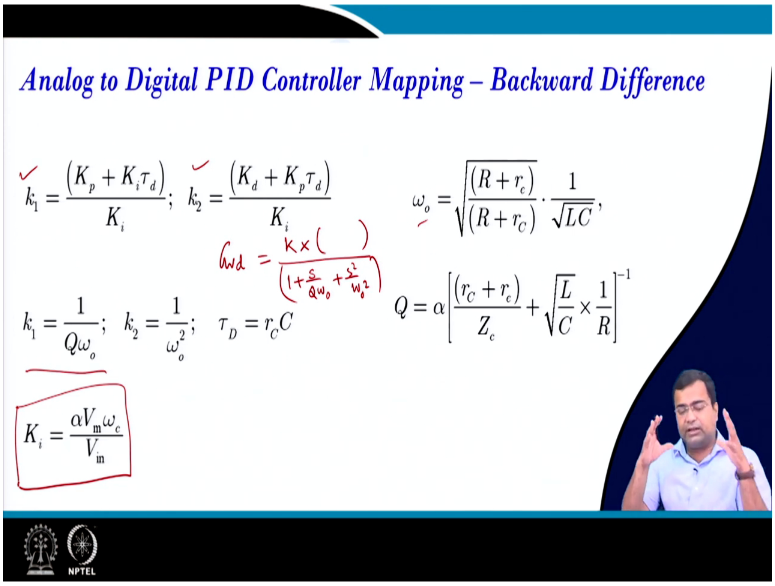
$= \frac{1}{10}$

So, now the question is if we compute let us say at; suppose if now it depends on the delay. So, if we take the crossover frequency as one-tenth of the switching frequency which typically we take, and if we take the delay as a fraction of the time period; that means, we can take a delay that may be one-tenth of the time period or it depends. So, if it is written k times T. Now then it can be shown that omega c tau d because omega c is this if you substitute then you will get the delay contribution due to that phase contribution due to the delay is k times thirty 36 degree.

That means if k is equal to 0.1. That means, if it is one-tenthnth of the time period, then it will contribute to 3.6-degree additional phase lag, and if the delay now slowly goes towards the time period. As we have discussed in the earlier architecture; that means, it can be possible that the delay can be a complete switching cycle. In that case, k will be 1 and you will get an additional phase lock of 36 degree.

So, you have to take into this account in the design process. Now once we design all the parameters because we know the K i value. We have obtained the other parameter of the PID controller.

(Refer Slide Time: 11:10)



Analog to Digital PID Controller Mapping – Backward Difference

✓ $k_1 = \frac{(K_p + K_i\tau_d)}{K_i}$; ✓ $k_2 = \frac{(K_d + K_p\tau_d)}{K_i}$

$\omega_o = \sqrt{\frac{(R+r_c)}{(R+r_C)}} \cdot \frac{1}{\sqrt{LC}}$,

$G_{vd} = \frac{k \times (\quad)}{\left(1+\frac{s}{Q\omega_0}+\frac{s^2}{\omega_0^2}\right)}$

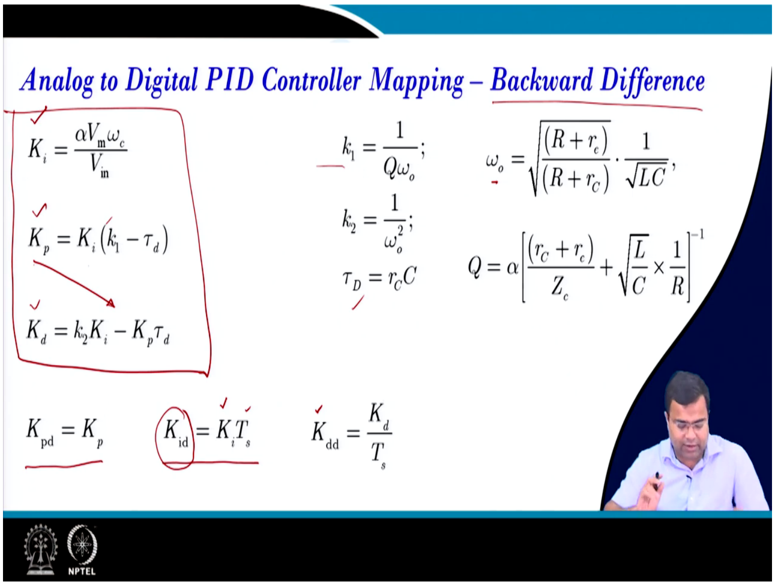
$\underline{k_1 = \frac{1}{Q\omega_o}}$; $k_2 = \frac{1}{\omega_o^2}$; $\tau_D = r_C C$

$Q = \alpha\left[\frac{(r_C + r_c)}{Z_c} + \sqrt{\frac{L}{C}} \times \frac{1}{R}\right]^{-1}$

$\boxed{K_i = \frac{\alpha V_m \omega_c}{V_{in}}}$

So, if we summarize first thing we know is k 1 k 2 which is a function of K p K i K d and we know k 1 using stable pole-zero cancellation. It is 1 by Q omega 0 and we know the denominator polynomial of the buck converter, what is there? If you take G vd you will get some gain into ES of 0 divided by you will get 1 plus s Q omega 0 plus a square by omega 0 square. So, I am talking about this Q omega 0 expressions using stable pole-zero cancellation. Then we know the expression of omega 0 and Q. We also found the K i from the loop transfer function at the gain crossover frequency.

(Refer Slide Time: noon)



Analog to Digital PID Controller Mapping – Backward Difference

$$K_i = \frac{\alpha V_m \omega_c}{V_{in}}$$

$$K_p = K_i\left(k_1 - \tau_d\right)$$

$$K_d = k_2 K_i - K_p \tau_d$$

$$k_1 = \frac{1}{Q\omega_o};$$

$$k_2 = \frac{1}{\omega_o^2};$$

$$\tau_D = r_C C$$

$$\omega_o = \sqrt{\frac{(R+r_c)}{(R+r_C)}} \cdot \frac{1}{\sqrt{LC}},$$

$$Q = \alpha\left[\frac{(r_C + r_c)}{Z_c} + \sqrt{\frac{L}{C}} \times \frac{1}{R}\right]^{-1}$$

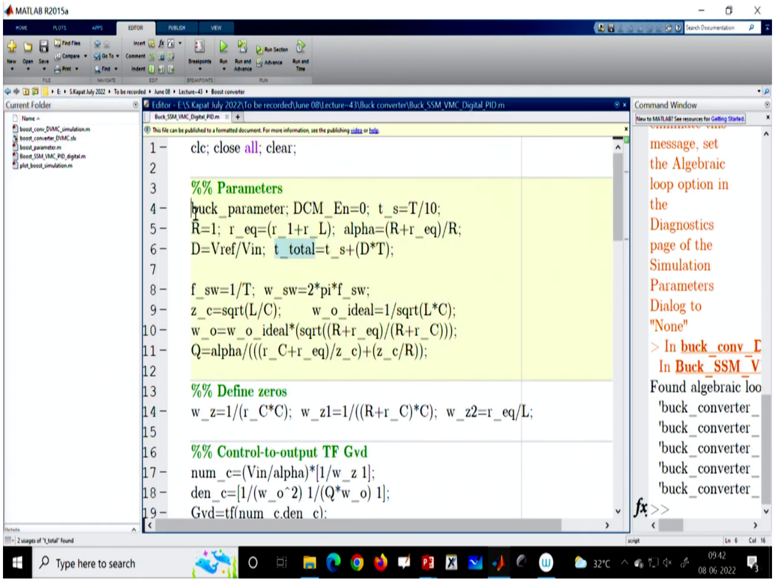
$$K_{pd} = K_p \qquad K_{id} = K_i T_s \qquad K_{dd} = \frac{K_d}{T_s}$$

Now, we have to map using the backward difference formula. So, we know K i and we know k 1 k 2, and tau d. We can also know how they are I mean this k 1 k 2 is a function of the power stage parameter. And then we want to calculate K p from k 1 that expression also we have shown, which will be K p equal to K i k 1 minus tau d an,d since everything is known here. So, you can get K p next K d again by solving k 1 k 2 we can get K d and this is also a function of K i and all.

So, since we have already found it here, we can find K d. Once we get the integral gain in the analog domain proportional and the derivative gain these are all in the analog domain. Then it is a matter of conversion you know how you convert that analog gain into digital gain, and we are using a backward difference formula and it can be shown the proportional gain will remain the same. The integral gain of the discrete domain will be simply the analog integral gain into the sampling time. In this case, it is nothing but the switching time period.

And for derivative gain in the digital domain will be derivative gain in the analog domain divided by T s ok. So, in that way, we can design the converter and we can design a compensator.

(Refer Slide Time: 13:26)



Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m

```
clc; close all; clear;

%% Parameters
buck_parameter; DCM_En=0;  t_s=T/10;
R=1;  r_eq=(r_1+r_L);  alpha=(R+r_eq)/R;
D=Vref/Vin;  t_total=t_s+(D*T);

f_sw=1/T;  w_sw=2*pi*f_sw;
z_c=sqrt(L/C);        w_o_ideal=1/sqrt(L*C);
w_o=w_o_ideal*(sqrt((R+r_eq)/(R+r_C)));
Q=alpha/(((r_C+r_eq)/z_c)+(z_c/R));

%% Define zeros
w_z=1/(r_C*C);  w_z1=1/((R+r_C)*C);  w_z2=r_eq/L;

%% Control-to-output TF Gvd
num_c=(Vin/alpha)*[1/w_z 1];
den_c=[1/(w_o^2) 1/(Q*w_o) 1];
Gvd=tf(num_c,den_c);
```

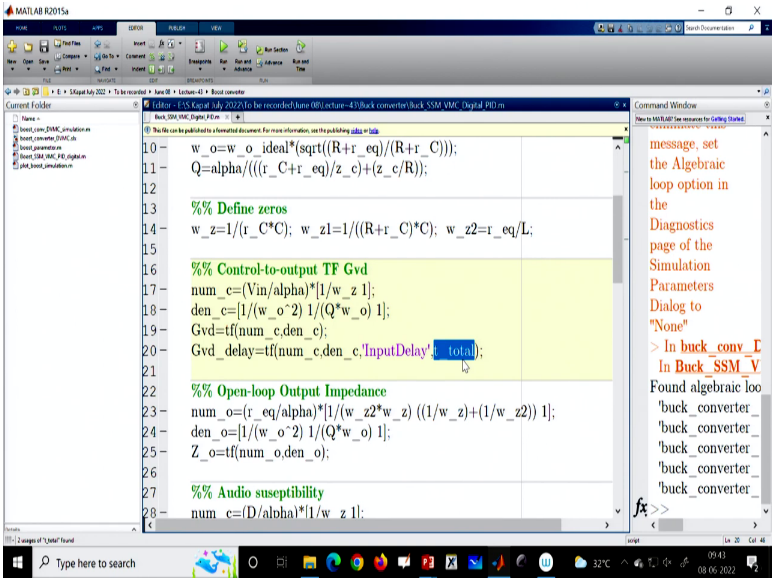
Command Window

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

Now, if we go to a MATLAB case study. So, I am showing a buck converter case study and here again, we have the power stage parameters we are talking about the R 1 is the initial load resistance, resistance is 1 and we have considered the delay due to a DC conversion is T by 10 and then all these things we have discussed in lecture number I think 35 and 36 where we have validated the model.

Now, here we have a total delay that t s plus D T which is the DPWM delay plus the EDC delay and we have discussed in our earlier NPTEL course how to design a voltage mode controller using stable pole-zero cancellation. So, we are following the same method.

(Refer Slide Time: 14:12)



Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m

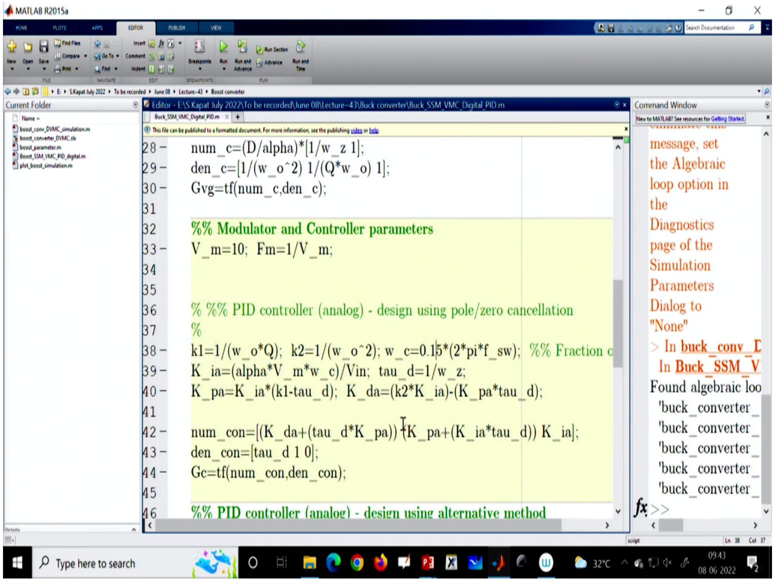
```
w_o=w_o_ideal*(sqrt((R+r_eq)/(R+r_C)));
Q=alpha/(((r_C+r_eq)/z_c)+(z_c/R));

%% Define zeros
w_z=1/(r_C*C);  w_z1=1/((R+r_C)*C);  w_z2=r_eq/L;

%% Control-to-output TF Gvd
num_c=(Vin/alpha)*[1/w_z 1];
den_c=[1/(w_o^2) 1/(Q*w_o) 1];
Gvd=tf(num_c,den_c);
Gvd_delay=tf(num_c,den_c,'InputDelay',t_total);

%% Open-loop Output Impedance
num_o=(r_eq/alpha)*[1/(w_z2*w_z) ((1/w_z)+(1/w_z2)) 1];
den_o=[1/(w_o^2) 1/(Q*w_o) 1];
Z_o=tf(num_o,den_o);

%% Audio suseptibility
num_c=(D/alpha)*[1/w_z 1];
```

Command Window

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

The only difference is in the case of the regular continuous strain transfer function. We will get Gvd which is the control to output transfer function as an expression of V in alpha ESR. I think that we have discussed the whole expression in lecture number 42. Now, we have a delayed version of this transfer function. That means, if we go back to our loop transfer function, in digital control we have an additional delay and that delay is e into t total; that means, the total delay which is a conversion time that is tau d conversion time plus our d DT that is a DPWM delay.

(Refer Slide Time: 14:51)



Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m

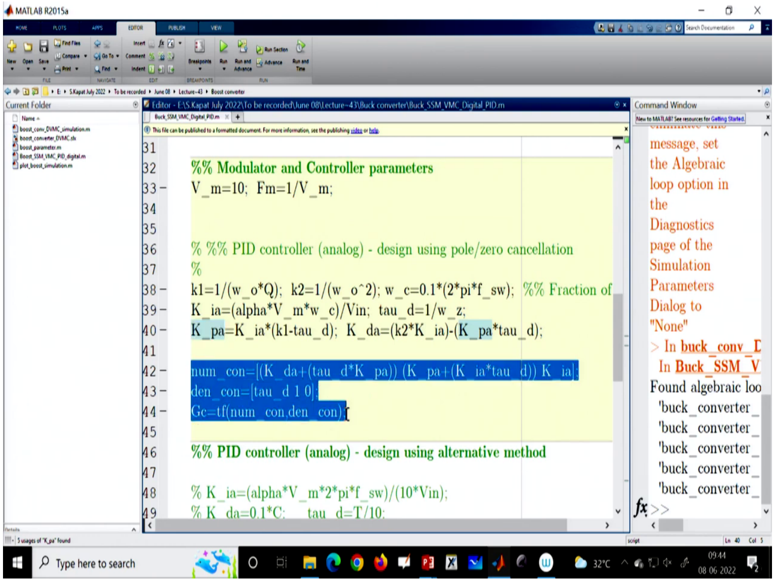
```
num_c=(D/alpha)*[1/w_z 1];
den_c=[1/(w_o^2) 1/(Q*w_o) 1];
Gvg=tf(num_c,den_c);

%% Modulator and Controller parameters
V_m=10;  Fm=1/V_m;


% %% PID controller (analog) - design using pole/zero cancellation
%
k1=1/(w_o*Q);  k2=1/(w_o^2); w_c=0.15*(2*pi*f_sw);  %% Fraction c
K_ia=(alpha*V_m*w_c)/Vin;  tau_d=1/w_z;
K_pa=K_ia*(k1-tau_d);  K_da=(k2*K_ia)-(K_pa*tau_d);

num_con=[(K_da+(tau_d*K_pa)) (K_pa+(K_ia*tau_d)) K_ia];
den_con=[tau_d 1 0];
Gc=tf(num_con,den_con);

%% PID controller (analog) - design using alternative method
```

Command Window

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

Now we want to design a controller. So, the first thing we are to design a controller, we have discussed just now the k 1 k 2 using exact pole-zero cancellation and then we can get omega c.

(Refer Slide Time: 15:05)



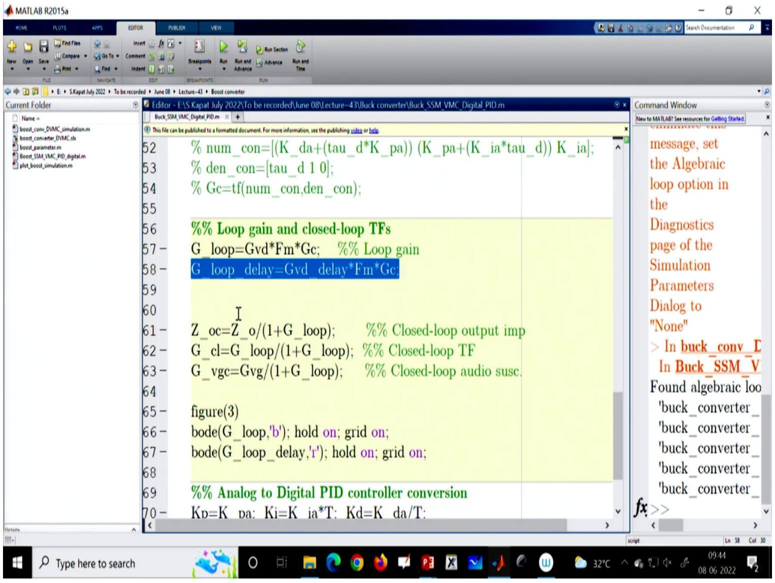
Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m

```
%% Modulator and Controller parameters
V_m=10;  Fm=1/V_m;


% %% PID controller (analog) - design using pole/zero cancellation
%
k1=1/(w_o*Q);  k2=1/(w_o^2); w_c=0.1*(2*pi*f_sw);  %% Fraction of
K_ia=(alpha*V_m*w_c)/Vin;  tau_d=1/w_z;
K_pa=K_ia*(k1-tau_d);  K_da=(k2*K_ia)-(K_pa*tau_d);

num_con=[(K_da+(tau_d*K_pa)) (K_pa+(K_ia*tau_d)) K_ia];
den_con=[tau_d 1 0];
Gc=tf(num_con,den_con);

%% PID controller (analog) - design using alternative method

% K_ia=(alpha*V_m*2*pi*f_sw)/(10*Vin);
% K_da=0.1*C;     tau_d=T/10;
```

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_I
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
```

Suppose we set the crossover frequency to one-tenth the switching frequency, then we can compute the analog integral gain we have just now shown. The tau d is the derivative filter time constant, then the analog proportional controller then the analog derivative controller. Then we can get the analog transfer function Gc.

(Refer Slide Time: 15:26)



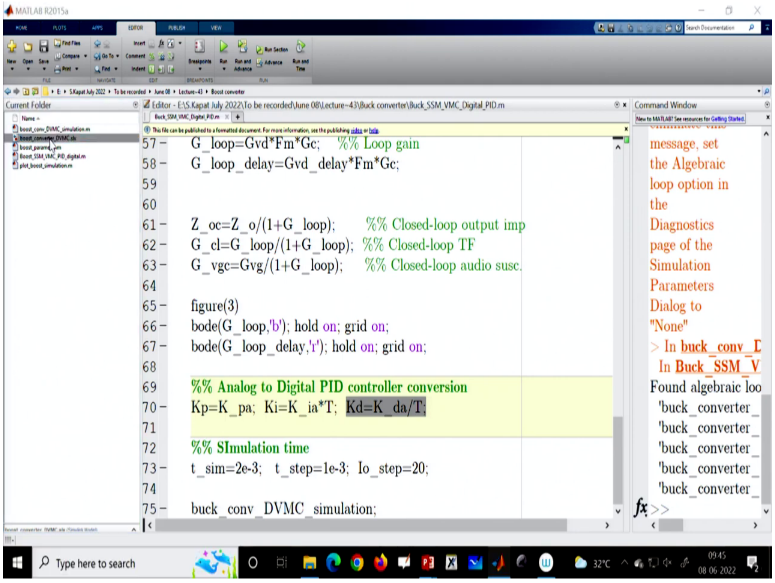
Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m

```
% num_con=[(K_da+(tau_d*K_pa)) (K_pa+(K_ia*tau_d)) K_ia];
% den_con=[tau_d 1 0];
% Gc=tf(num_con,den_con);

%% Loop gain and closed-loop TFs
G_loop=Gvd*Fm*Gc;    %% Loop gain
G_loop_delay=Gvd_delay*Fm*Gc;


Z_oc=Z_o/(1+G_loop);       %% Closed-loop output imp
G_cl=G_loop/(1+G_loop);  %% Closed-loop TF
G_vgc=Gvg/(1+G_loop);      %% Closed-loop audio susc.

figure(3)
bode(G_loop,'b'); hold on; grid on;
bode(G_loop_delay,'r'); hold on; grid on;

%% Analog to Digital PID controller conversion
Kp=K_pa;  Ki=K_ia*T;  Kd=K_da/T;
```

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_I
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
```

Now, once we have then we can draw the loop transfer function of the analog and the loop transfer function of the digital which is nothing but the analog loop transformation multiplied by the delay term, and then we can get the bode plot. We can compare to see how much phase margin is affected because of the delay, that you want to see at our gain crossover frequency.

(Refer Slide Time: 15:47)

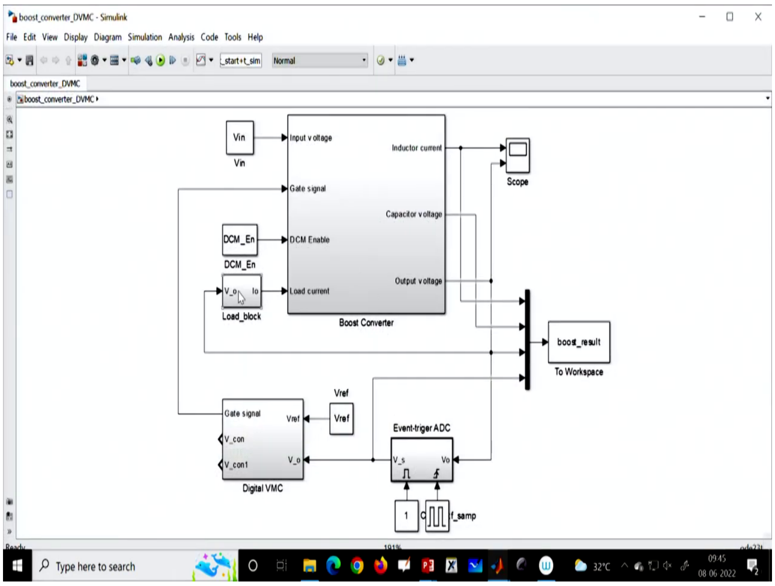

MATLAB R2015a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Current Folder
boost_conv_DVMC_simulation.m
boost_converter_DVMC.slx
boost_param...m
Boost_SSM_VMC_PID_digital.m
plot_boost_simulation.m
Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m
Buck_SSM_VMC_Digital_PID.m
Command Window
Type here to search
32°C
09:45
08 06 2022

```
G_loop=Gvd*Fm*Gc;    %% Loop gain
G_loop_delay=Gvd_delay*Fm*Gc;


Z_oc=Z_o/(1+G_loop);       %% Closed-loop output imp
G_cl=G_loop/(1+G_loop);  %% Closed-loop TF
G_vgc=Gvg/(1+G_loop);     %% Closed-loop audio susc.

figure(3)
bode(G_loop,'b'); hold on; grid on;
bode(G_loop_delay,'r'); hold on; grid on;

%% Analog to Digital PID controller conversion
Kp=K_pa;  Ki=K_ia*T;  Kd=K_da/T;

%% SImulation time
t_sim=2e-3;   t_step=1e-3;  Io_step=20;

buck_conv_DVMC_simulation;
```

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
 'buck_converter_
 'buck_converter_
 'buck_converter_
 'buck_converter_
 'buck_converter_
fx >>
```

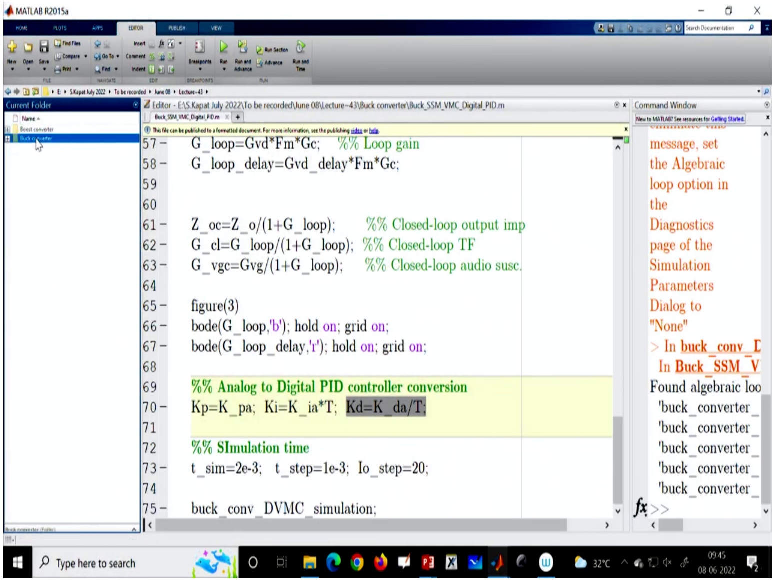
And finally, we want to convert all these gains into a digital discrete-time proportional controller, the discrete-time integral controller, and the discrete-time derivative controller and here we are simulating for 2 millisecond. We are applying a step transient at 1 millisecond and we are applying a load step of 20 ampere. And if you go to our Simulink diagram ok so Simulink diagram.

(Refer Slide Time: 16:13)



So, I want to show that this is a Simulink block diagram and here we are applying this diagram. We are applying a load step transient sorry this is the boost one.

(Refer Slide Time: 16:27)



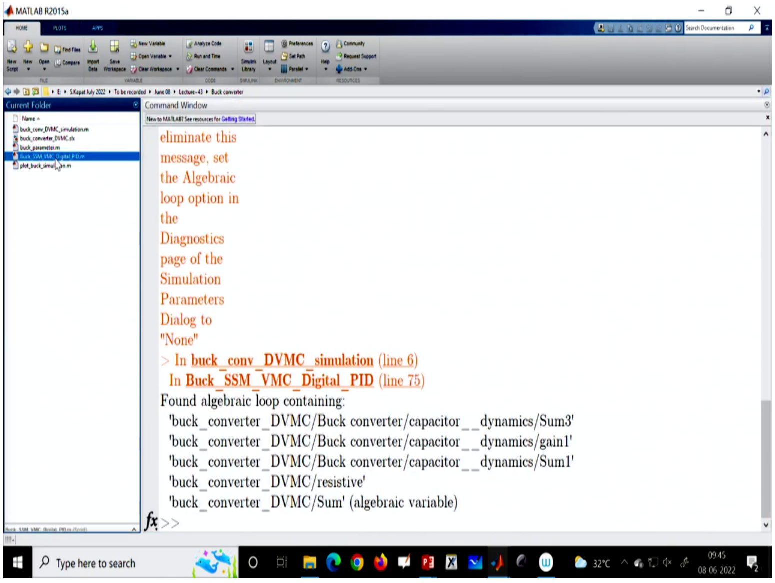
Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m

```
G_loop=Gvd*Fm*Gc;    %% Loop gain
G_loop_delay=Gvd_delay*Fm*Gc;


Z_oc=Z_o/(1+G_loop);       %% Closed-loop output imp
G_cl=G_loop/(1+G_loop);  %% Closed-loop TF
G_vgc=Gvg/(1+G_loop);      %% Closed-loop audio susc.

figure(3)
bode(G_loop,'b'); hold on; grid on;
bode(G_loop_delay,'r'); hold on; grid on;

%% Analog to Digital PID controller conversion
Kp=K_pa;  Ki=K_ia*T;  Kd=K_da/T;

%% SImulation time
t_sim=2e-3;   t_step=1e-3;  Io_step=20;

buck_conv_DVMC_simulation;
```

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

I want t,o go here.

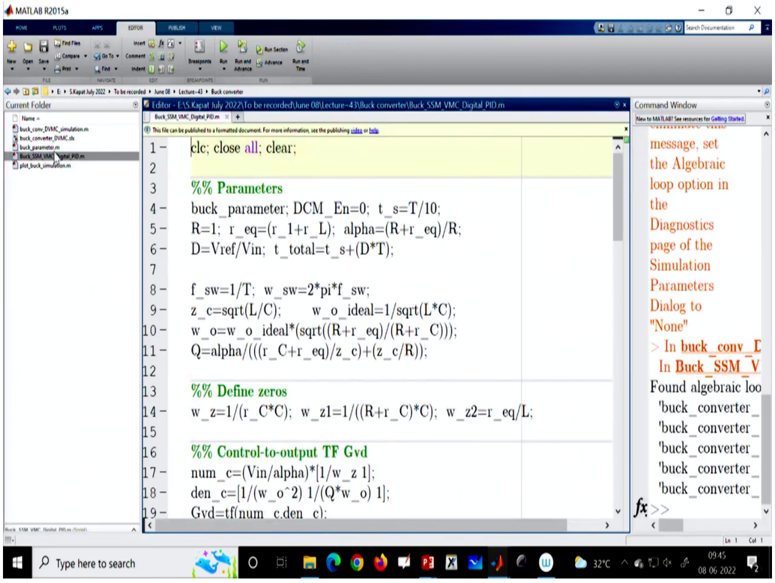
(Refer Slide Time: 16:33)



```
eliminate this
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_DVMC_simulation (line 6)
  In Buck_SSM_VMC_Digital_PID (line 75)
Found algebraic loop containing:
  'buck_converter_DVMC/Buck converter/capacitor__dynamics/Sum3'
  'buck_converter_DVMC/Buck converter/capacitor__dynamics/gain1'
  'buck_converter_DVMC/Buck converter/capacitor__dynamics/Sum1'
  'buck_converter_DVMC/resistive'
  'buck_converter_DVMC/Sum' (algebraic variable)
fx >>
```

Ok.

(Refer Slide Time: 16:34)



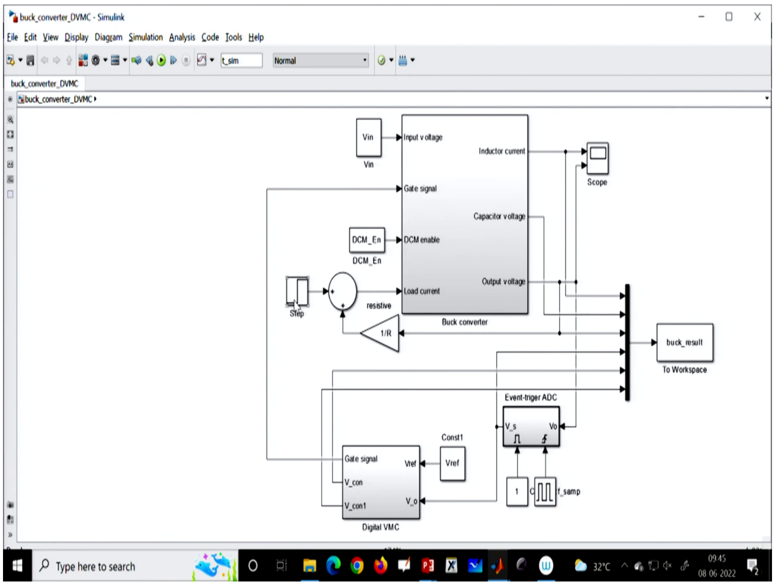
Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m

```
clc; close all; clear;

%% Parameters
buck_parameter; DCM_En=0;  t_s=T/10;
R=1;  r_eq=(r_1+r_L);  alpha=(R+r_eq)/R;
D=Vref/Vin;  t_total=t_s+(D*T);

f_sw=1/T;  w_sw=2*pi*f_sw;
z_c=sqrt(L/C);        w_o_ideal=1/sqrt(L*C);
w_o=w_o_ideal*(sqrt((R+r_eq)/(R+r_C)));
Q=alpha/(((r_C+r_eq)/z_c)+(z_c/R));

%% Define zeros
w_z=1/(r_C*C);  w_z1=1/((R+r_C)*C);  w_z2=r_eq/L;

%% Control-to-output TF Gvd
num_c=(Vin/alpha)*[1/w_z 1];
den_c=[1/(w_o^2) 1/(Q*w_o) 1];
Gvd=tf(num_c,den_c);
```

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

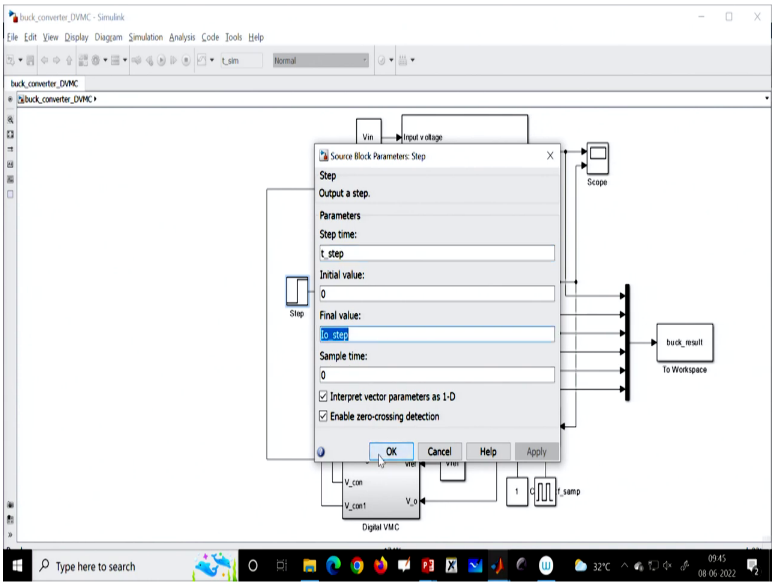
Yes.

(Refer Slide Time: 16:37)



So, here I want to show this diagram.

(Refer Slide Time: 16:42)


Source Block Parameters: Step

Step

Output a step.

Parameters

Step time:

t_step

Initial value:

0

Final value:

Io_step

Sample time:

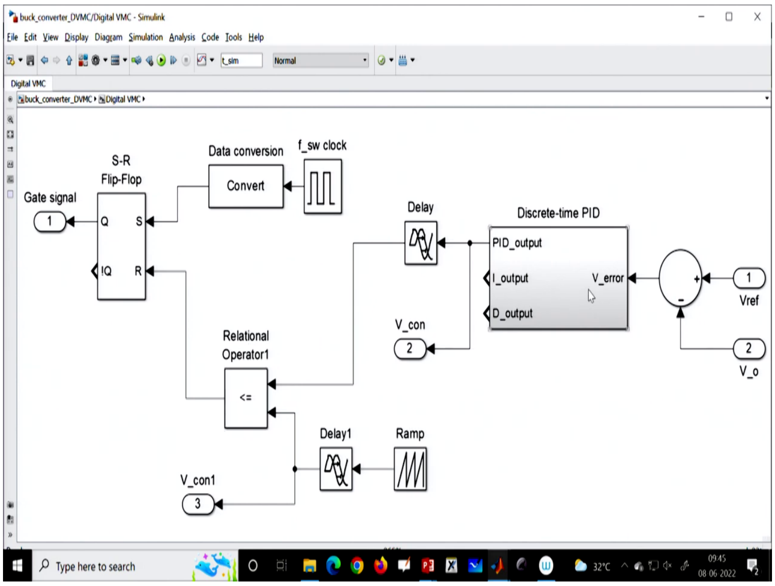
0

☑ Interpret vector parameters as 1-D

☑ Enable zero-crossing detection

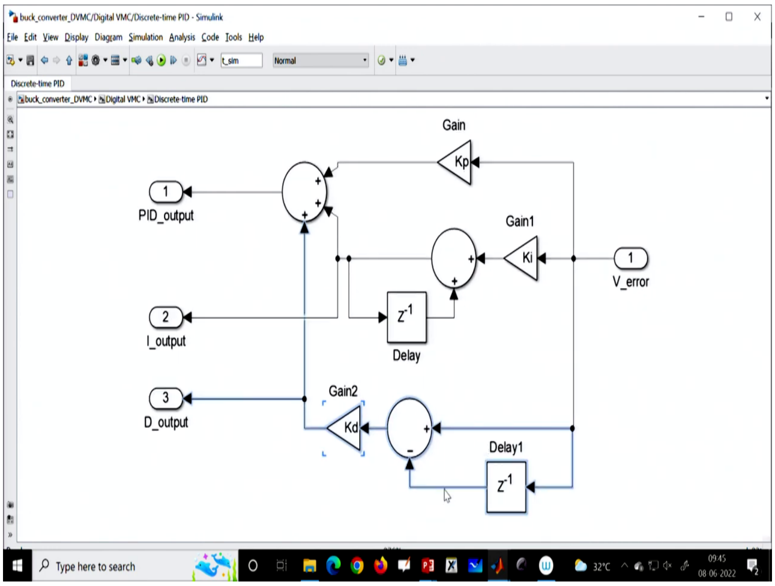
Here we are applying a load step of 20 ampere i 0 steps at t step time, where we have applied the load.

(Refer Slide Time: 16:49)



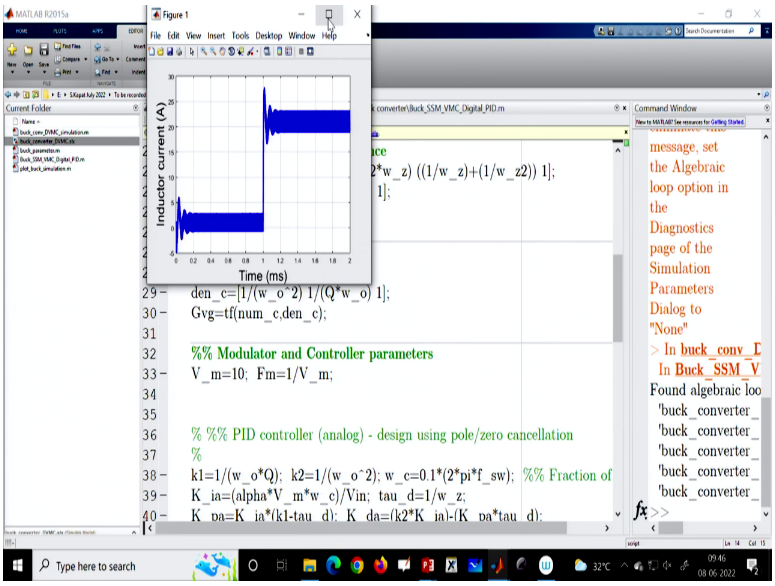
This is the digital voltage mode controller and we have explained to you know why these delays are incorporated in lecture number 35 and 36 we have discussed in detail.

(Refer Slide Time: 17:00)


buck_converter_DVMC/Digital VMC/Discrete-time PID - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
t_sim
Normal
Discrete-time PID
buck_converter_DVMC ▸ Digital VMC ▸ Discrete-time PID
Gain
Kp
1
PID_output
Gain1
Ki
1
V_error
2
I_output
z-1
Delay
3
D_output
Gain2
Kd
Delay1
z-1
Type here to search
32°C
09:45
08-06-2022

And this is our PID; a discrete time PID controller. This is the K p, this is the discrete-time integral part and this is a discrete-time derivative part that we have discussed. Now, we want to run a case study for a load step transient which is designed based on our traditional analog control and we want to check what happened.

(Refer Slide Time: 17:21)

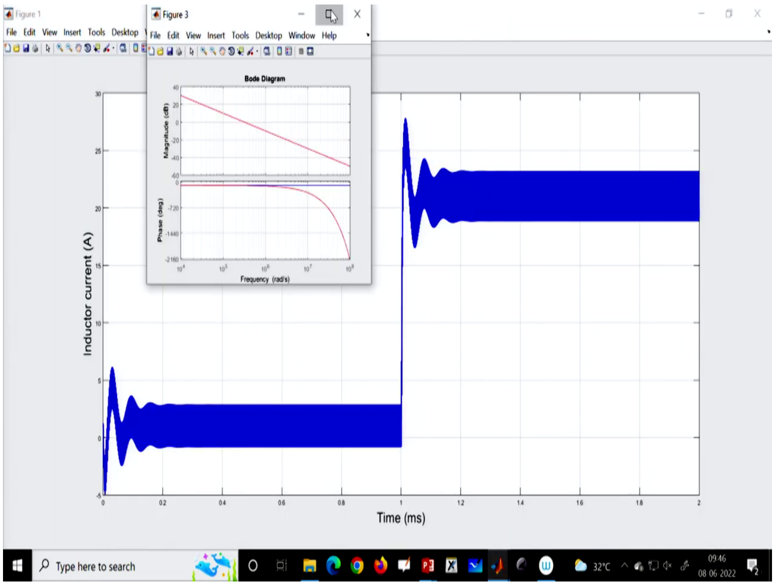


```
                                              ce
                                              2*w_z) ((1/w_z)+(1/w_z2)) 1];
                                              1];

den_c=[1/(w_o^2) 1/(Q*w_o) 1];
Gvg=tf(num_c,den_c);

%% Modulator and Controller parameters
V_m=10;  Fm=1/V_m;


% %% PID controller (analog) - design using pole/zero cancellation
%
k1=1/(w_o*Q);  k2=1/(w_o^2); w_c=0.1*(2*pi*f_sw);  %% Fraction of
K_ia=(alpha*V_m*w_c)/Vin;  tau_d=1/w_z;
K_pa=K_ia*(k1-tau_d);  K_da=(k2*K_ia)-(K_pa*tau_d);
```

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

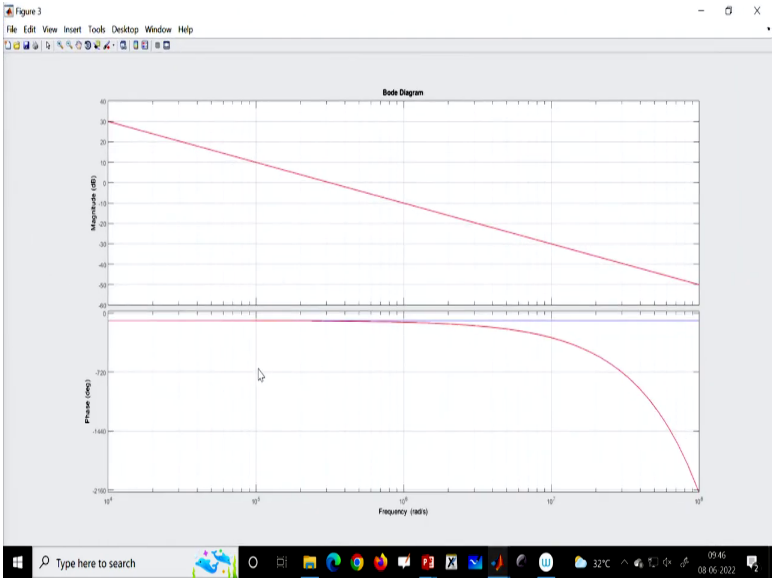
You know ok.

(Refer Slide Time: 17:25)

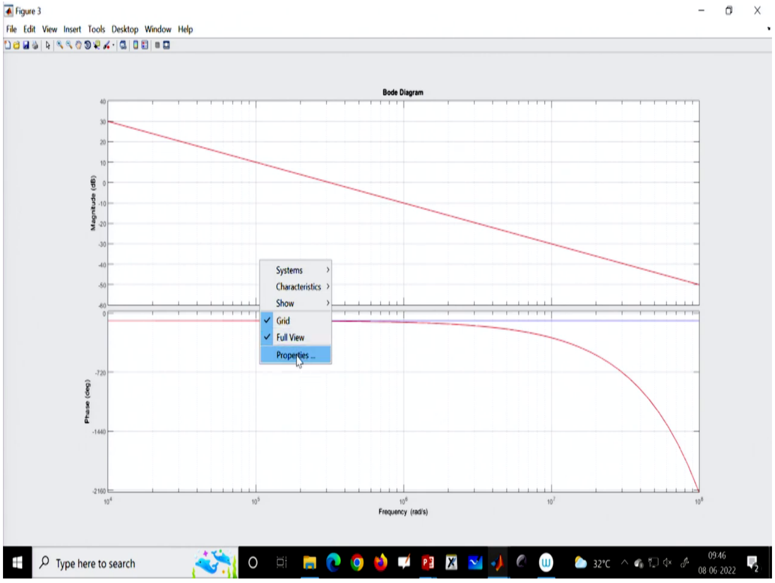


So, before we move first I want to show the bode plot.

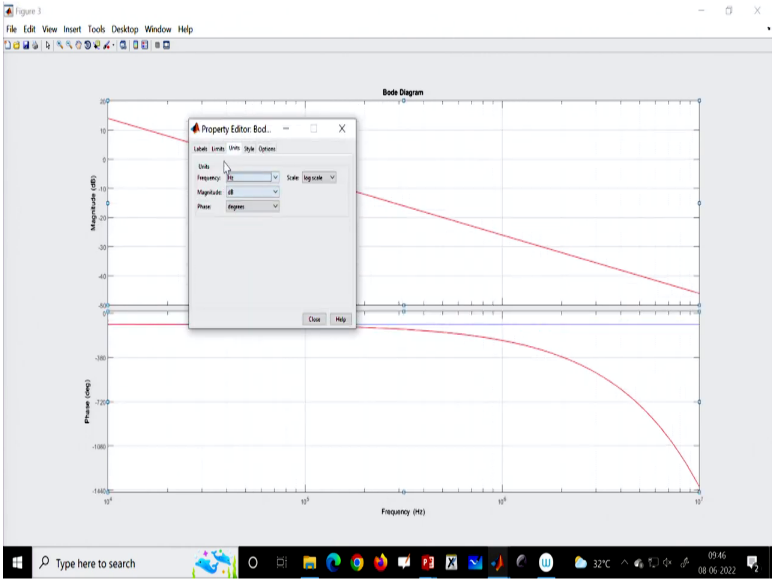
(Refer Slide Time: 17:28)



(Refer Slide Time: 17:29)



(Refer Slide Time: 17:30)



Property Editor: Bod...

Units

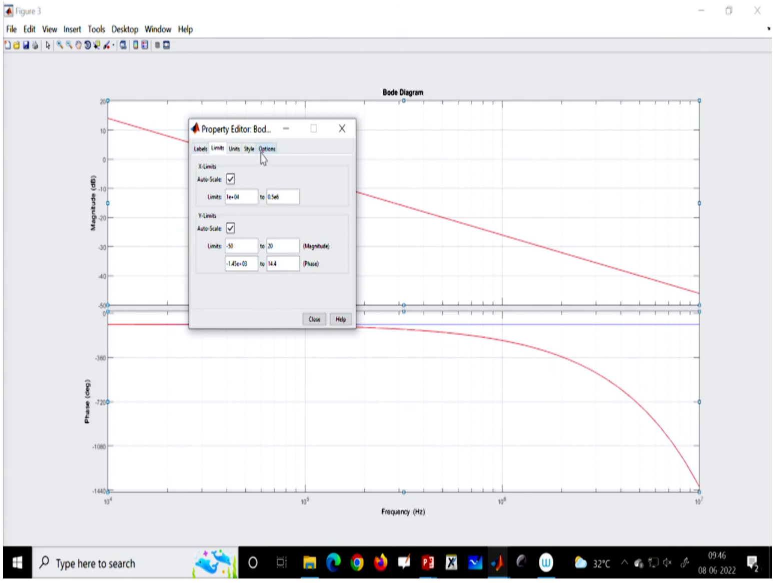
Frequency: Hz Scale: log scale

Magnitude: dB

Phase: degrees

So, let us go and I want to change some settings; that means, we have to hertz.

(Refer Slide Time: 17:34)



Property Editor: Bod...

X-Limits

Auto-Scale: ☑

Limits: 1e+04 to 0.5e6

Y-Limits

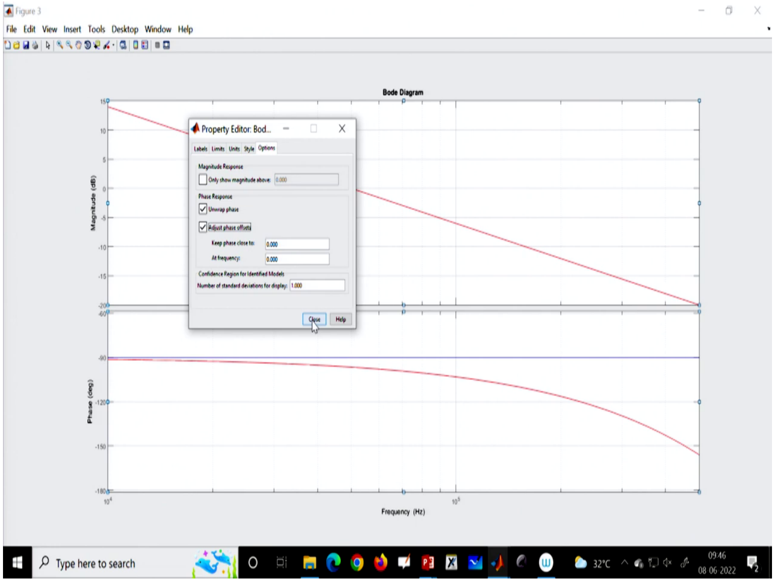
Auto-Scale: ☑

Limits: -50 to 20 (Magnitude)

-1.45e+03 to 14.4 (Phase)

The limit we have to limit up to half of the switching. You know up to the switching frequency you are limiting.

(Refer Slide Time: 17:43)



Property Editor: Bod...

Magnitude Response

□ Only show magnitude above: 0.000

Phase Response

☑ Unwrap phase

☑ Adjust phase offsets

Keep phase close to: 0.000

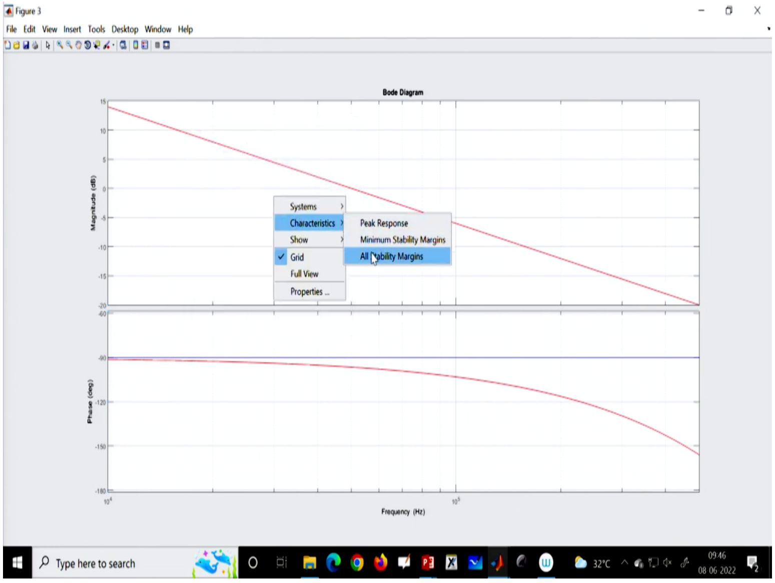
At frequency: 0.000

Confidence Region for Identified Models

Number of standard deviations for display: 1.000

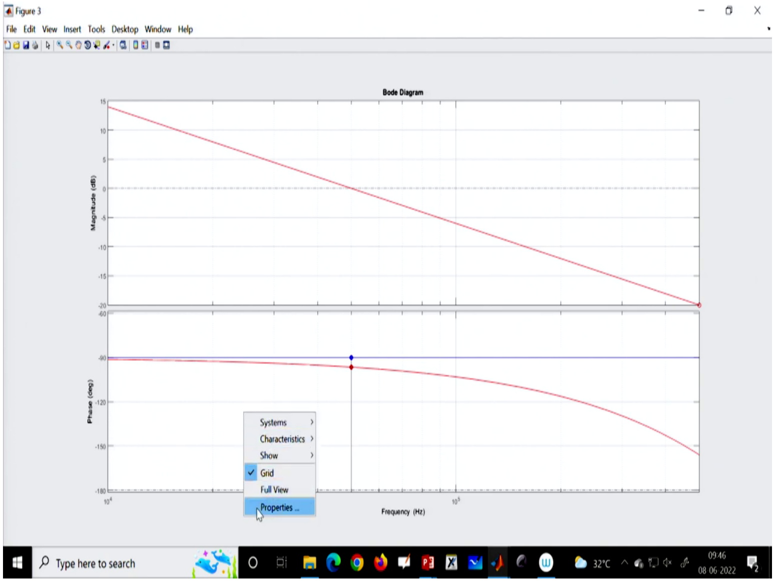
And phase also you are wrapping ok.

(Refer Slide Time: 17:48)



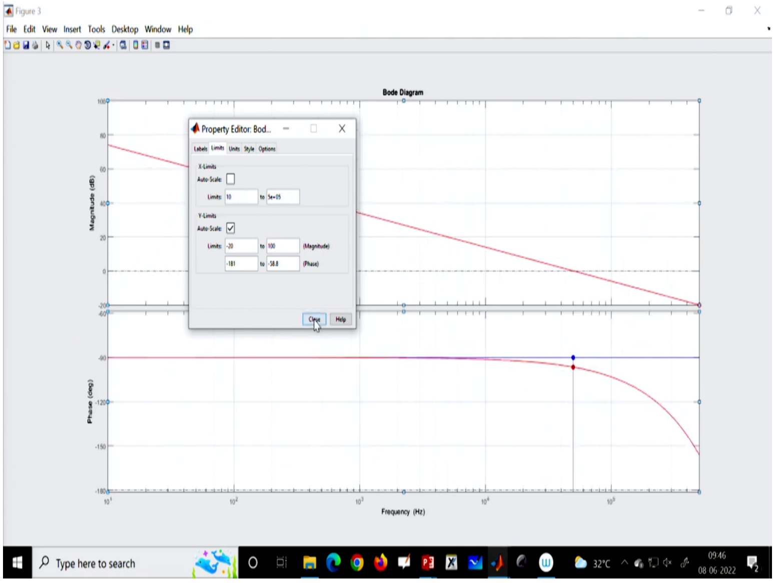
Now I want to show that our all ok.

(Refer Slide Time: 17:50)



So, here we want to achieve a desired crossover frequency of what? We want to achieve desired crossover frequency of one-tenth of the switching frequency. So, the switching frequency is 500 kilohertz and one-tenth of this is 50 kilohertz which is the case.

(Refer Slide Time: 18:06)



Property Editor: Bod...

Labels | Limits | Units | Style | Options

X-Limits

Auto-Scale: □

Limits: 10 to 5e+05

Y-Limits

Auto-Scale: ☑

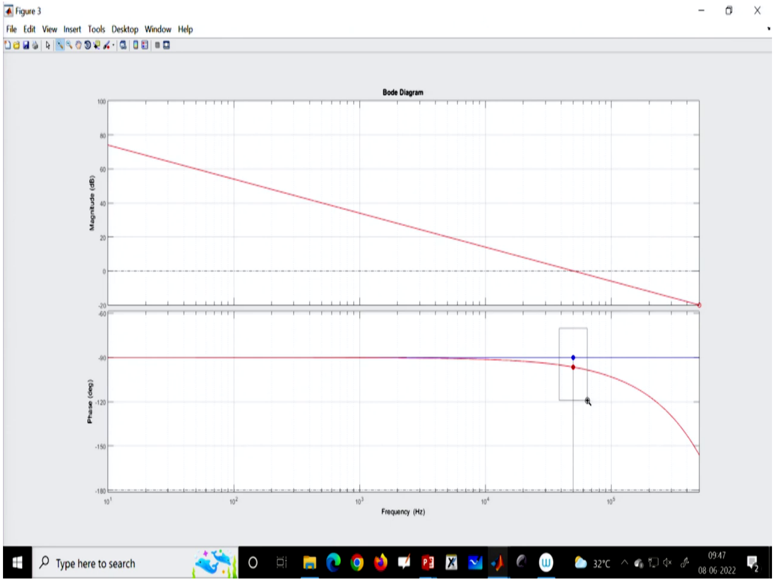
Limits: -20 to 100 (Magnitude)

-181 to -58.8 (Phase)

Close | Help

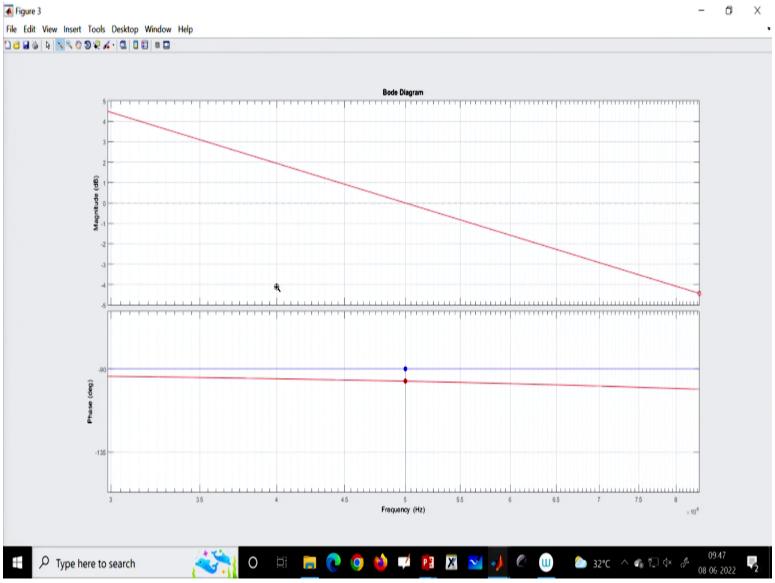
So, if we limit this to 1, yes. So, this is what I am showing.

(Refer Slide Time: 18:12)



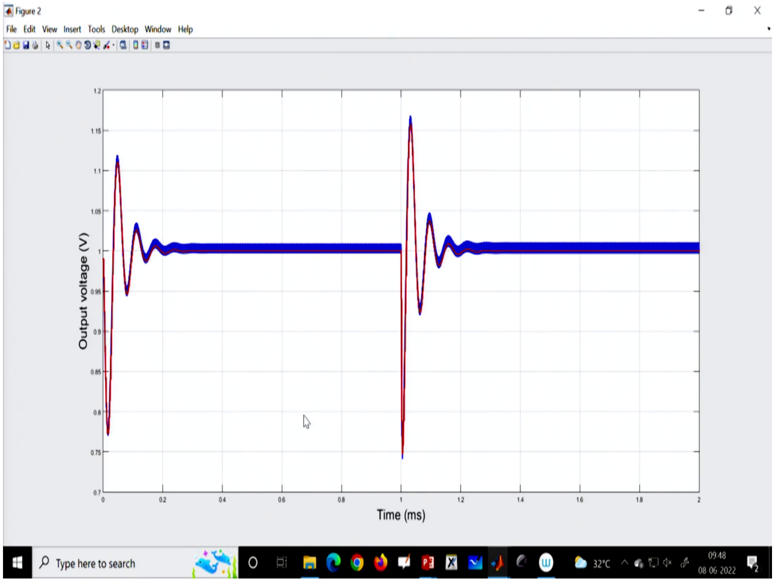
So, here is our one-tenth of the switching frequency and you can see there is an additional phase lag due to because we know for exact pole-zero cancellation in analog control the phase margin was always 90 degrees. But now, if we take the phase here for the digital you are getting a phase.

(Refer Slide Time: 18:42)



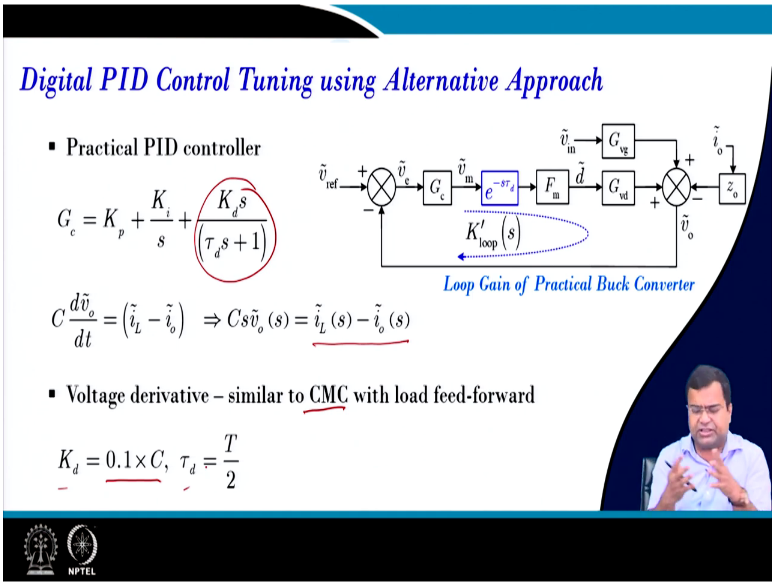
So, if you write it here you know if I zoom this part I want to show that you are losing some phase here and that may not be significant because we are losing around you know how much it may be some 20 degrees or so ok. And based on this phase margin criteria I mean whatever we have discussed we are designing this controller and this is the response of the load transient response.

(Refer Slide Time: 19:05)



And know earlier we also discussed that in the case of load tangent, if we simply shape the load transfer function, it is not enough. Because we have to shape the closed-loop output impedance and that means the loop transfer function does not include the open-loop output impedance. So, that is why this method may not be a good solution in terms of low transient response. Now we want to go next; that means, we want to consider the alternative approach.

(Refer Slide Time: 19:35)



Digital PID Control Tuning using Alternative Approach

- Practical PID controller

$$G_c = K_p + \frac{K_i}{s} + \frac{K_d s}{(\tau_d s + 1)}$$

Loop Gain of Practical Buck Converter

$$C\frac{d\tilde{v}_o}{dt} = \left(\tilde{i}_L - \tilde{i}_o\right) \Rightarrow Cs\tilde{v}_o(s) = \underline{\tilde{i}_L(s) - \tilde{i}_o(s)}$$

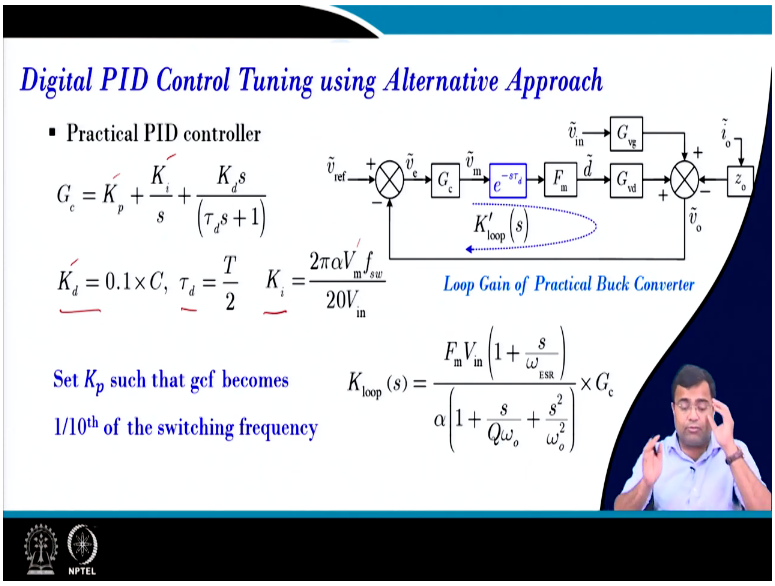
- Voltage derivative – similar to CMC with load feed-forward

$$K_d = \underline{0.1 \times C}, \ \tau_d = \frac{T}{2}$$

In the alternative approach, we have discussed the same loop transfer function everything remains the same, the same PID controller. But, we want to visualize the derivative gain as it is you take the derivative of a buck converter. Let us say if you ignore the effect of ESR it carries the information of the inductor current and the load current because it is the capacitor current C DV dt is equal to I c which is a capacitor current and it is i L minus i 0, which means it gives indirect information of the inductor current and the load feedforward.

So, here we can take K d equal to C directly, but it may be large and it may inject some noise. So, we are setting the derivative gain to just 0.1 times the output capacitor, and tau d you can set it to T by 10, T by 20, and T by 2, so it is not big; it is no problem.

(Refer Slide Time: 20:29)



Digital PID Control Tuning using Alternative Approach

- Practical PID controller

$$G_c = K_p + \frac{K_i}{s} + \frac{K_d s}{(\tau_d s + 1)}$$

$$K_d = 0.1 \times C, \quad \tau_d = \frac{T}{2} \quad K_i = \frac{2\pi\alpha V_m f_{sw}}{20 V_{in}}$$

Set K_p such that gcf becomes 1/10th of the switching frequency

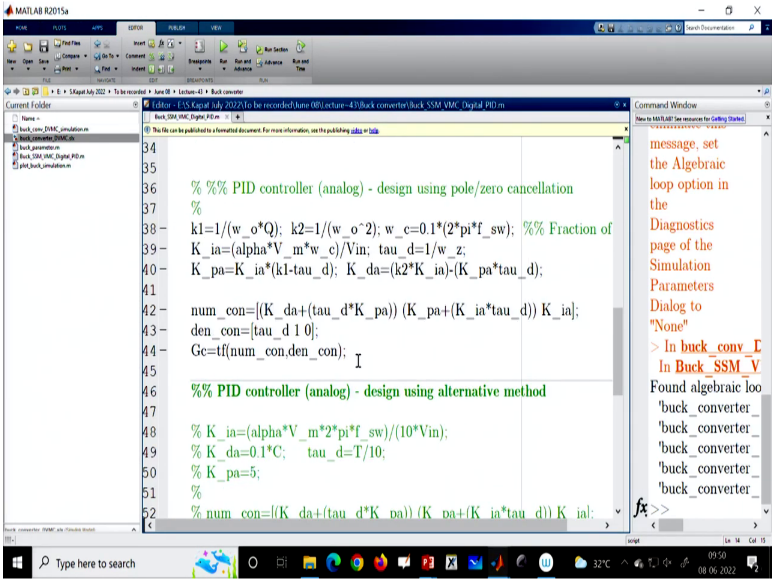
$$K_{loop}(s) = \frac{F_m V_{in}\left(1 + \frac{s}{\omega_{ESR}}\right)}{\alpha\left(1 + \frac{s}{Q\omega_o} + \frac{s^2}{\omega_o^2}\right)} \times G_c$$

Loop Gain of Practical Buck Converter

Then how do you design? So, in this case, we are first setting again the proportional control. We have taken K d equal to this tau d equal to this and we have taken the integral gain same as which we have discussed earlier. Because we just want to take integral gain to start with that is the one-tenth of the switching frequency and V m is the peak value of the saw tooth waveform. Now in this controller, we have four unknown K p K i K d and tau d. Here we already said three; that means, three parameters K d tau d and K i.

So, three parameters are already set. Now the fourth variable which is the K p that we have to select in such a way we can get the crossover frequency close to one-tenth of the switching frequency. So, where it is very difficult to analytically compute the expression of K p, you can always check using a bode plot what should be the value of K p, so that you can get the crossover frequency of nearly one-tenth of the switching frequency. So, we will summarize and we will take the case study again ok.

(Refer Slide Time: 21:34)



Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m

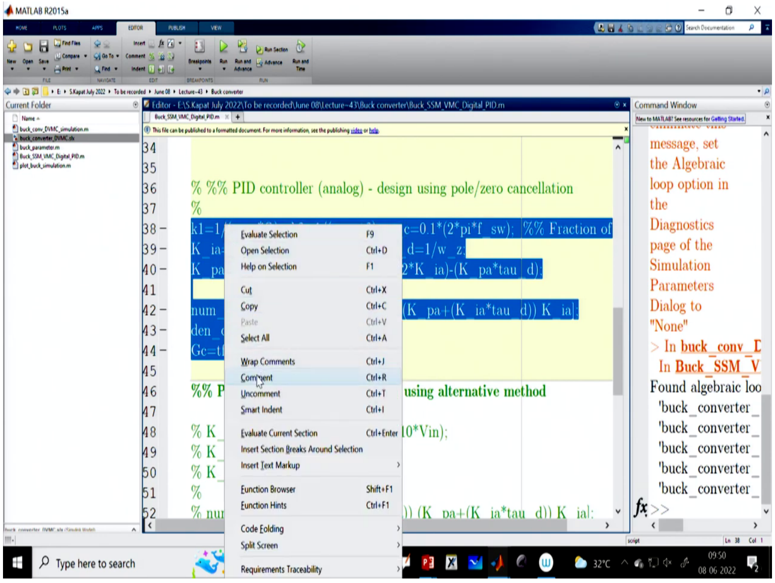
```
% %% PID controller (analog) - design using pole/zero cancellation
%
k1=1/(w_o*Q);  k2=1/(w_o^2); w_c=0.1*(2*pi*f_sw);  %% Fraction of
K_ia=(alpha*V_m*w_c)/Vin;  tau_d=1/w_z;
K_pa=K_ia*(k1-tau_d);  K_da=(k2*K_ia)-(K_pa*tau_d);

num_con=[(K_da+(tau_d*K_pa)) (K_pa+(K_ia*tau_d)) K_ia];
den_con=[tau_d 1 0];
Gc=tf(num_con,den_con);

%% PID controller (analog) - design using alternative method

% K_ia=(alpha*V_m*2*pi*f_sw)/(10*Vin);
% K_da=0.1*C;     tau_d=T/10;
% K_pa=5;
%
% num_con=[(K_da+(tau_d*K_pa)) (K_pa+(K_ia*tau_d)) K_ia];
```

Command Window

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

So, that means, again if we design, maybe we can go and check the case study. So, in this case, now we are designing the controller using the alternative wave.

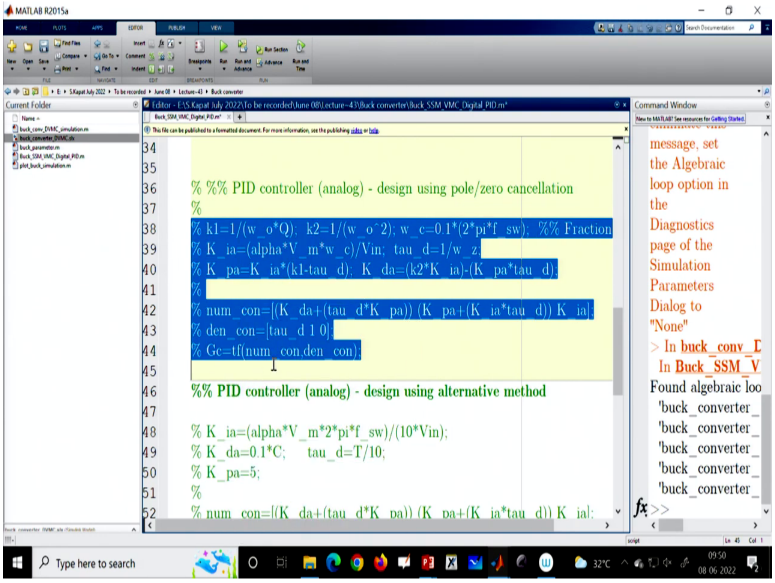
(Refer Slide Time: 21:45)



```
% %% PID controller (analog) - design using pole/zero cancellation
%
k1=1/                    c=0.1*(2*pi*f_sw);  %% Fraction of
K_ia                      d=1/w_z;
K_pa                     2*K_ia)-(K_pa*tau_d);

num                      (K_pa+(K_ia*tau_d)) K_ia];
den_c
Gc=t

%% P                     using alternative method

% K_                     10*Vin);
% K_
% K_
%
% nu                     )) (K_pa+(K_ia*tau_d)) K_ia];
```

That means we are commenting on this.

(Refer Slide Time: 21:47)



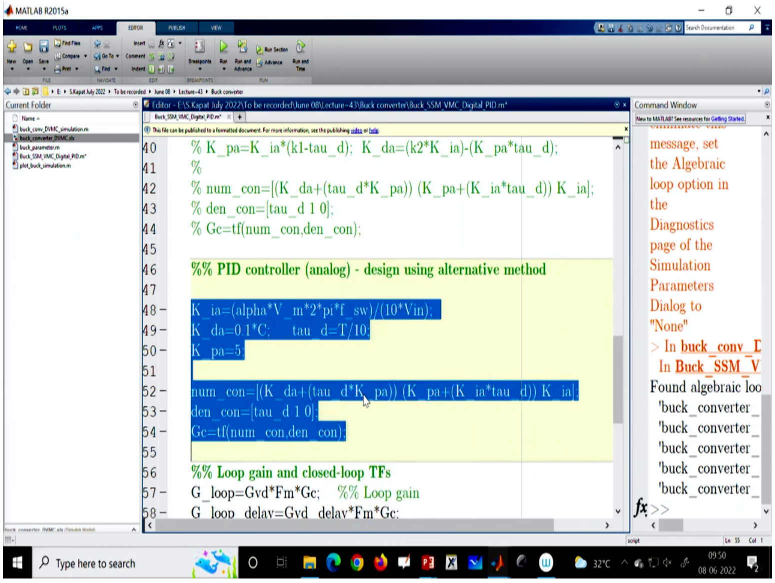
Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m*

```
% %% PID controller (analog) - design using pole/zero cancellation
%
% k1=1/(w_o*Q);  k2=1/(w_o^2); w_c=0.1*(2*pi*f_sw);  %% Fraction
% K_ia=(alpha*V_m*w_c)/Vin;  tau_d=1/w_z;
% K_pa=K_ia*(k1-tau_d);  K_da=(k2*K_ia)-(K_pa*tau_d);
%
% num_con=[(K_da+(tau_d*K_pa)) (K_pa+(K_ia*tau_d)) K_ia];
% den_con=[tau_d 1 0];
% Gc=tf(num_con,den_con);

%% PID controller (analog) - design using alternative method

% K_ia=(alpha*V_m*2*pi*f_sw)/(10*Vin);
% K_da=0.1*C;     tau_d=T/10;
% K_pa=5;
%
% num_con=[(K_da+(tau_d*K_pa)) (K_pa+(K_ia*tau_d)) K_ia];
```

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

And this is the alternative design approach that I have discussed.

(Refer Slide Time: 21:49)

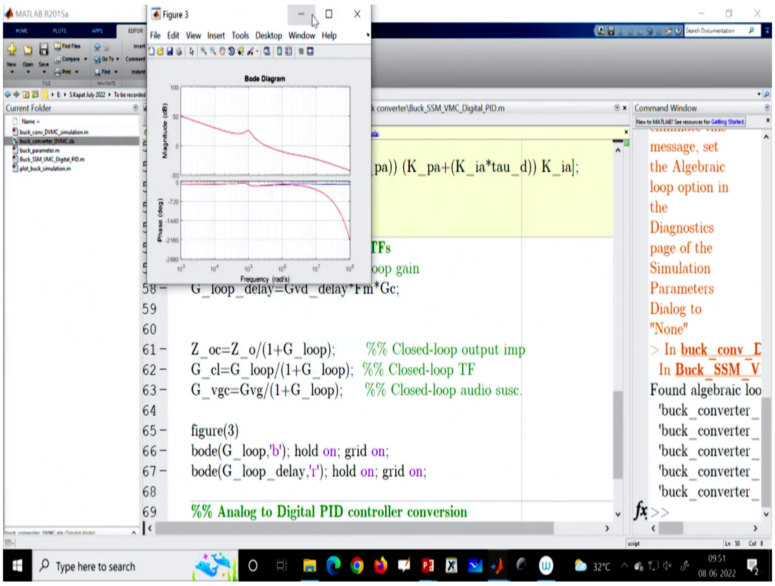


Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m*

```
% K_pa=K_ia*(k1-tau_d);  K_da=(k2*K_ia)-(K_pa*tau_d);
%
% num_con=[(K_da+(tau_d*K_pa)) (K_pa+(K_ia*tau_d)) K_ia];
% den_con=[tau_d 1 0];
% Gc=tf(num_con,den_con);

%% PID controller (analog) - design using alternative method

K_ia=(alpha*V_m*2*pi*f_sw)/(10*Vin);
K_da=0.1*C;     tau_d=T/10;
K_pa=5;

num_con=[(K_da+(tau_d*K_pa)) (K_pa+(K_ia*tau_d)) K_ia];
den_con=[tau_d 1 0];
Gc=tf(num_con,den_con);

%% Loop gain and closed-loop TFs
G_loop=Gvd*Fm*Gc;    %% Loop gain
G_loop_delay=Gvd_delay*Fm*Gc;
```

So, in this approach, we have set K i analog controller gain integral gain that I have just discussed. K d equal to this tau d you can take T by 10 T by 2 does not matter. And then we are taking the proportional gain to 5 and we have to check how much crossover frequency we are achieving and then we want to run and verify. So, let us do that.

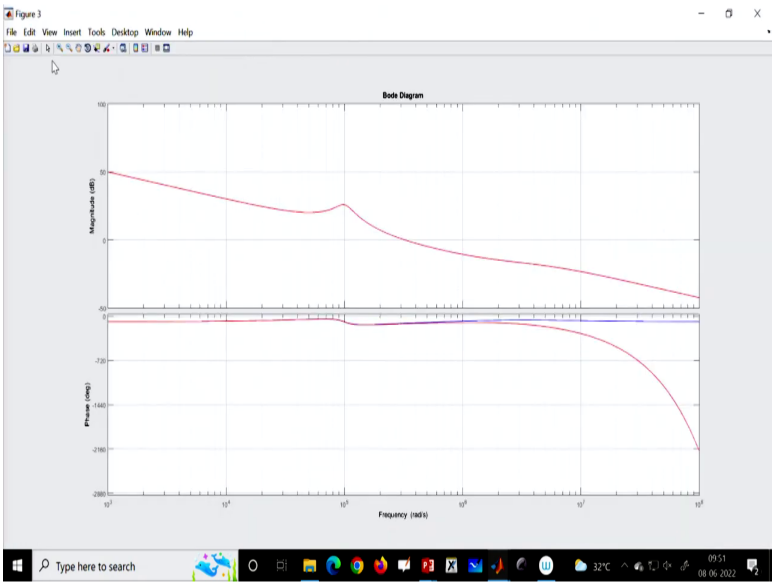
(Refer Slide Time: 22:15)



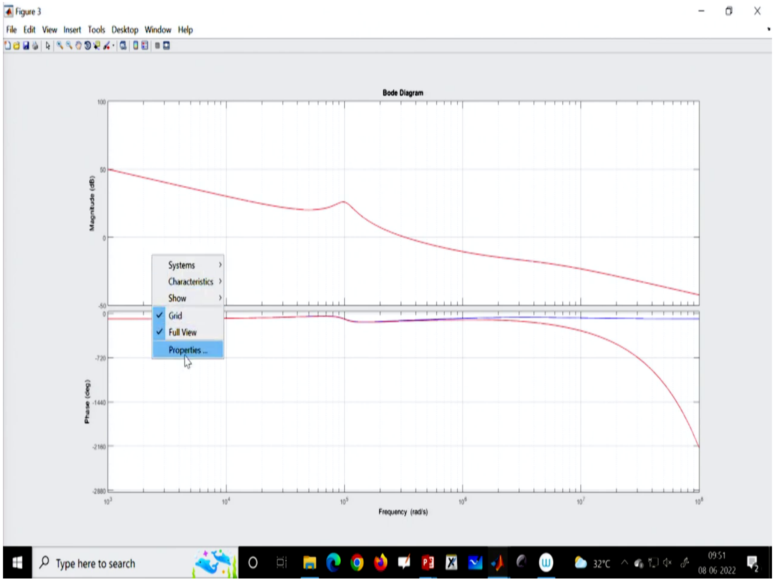
```
pa)) (K_pa+(K_ia*tau_d)) K_ia];

TFs
oop gain
G_loop_delay=Gvd_delay*Fm*Gc;


Z_oc=Z_o/(1+G_loop);        %% Closed-loop output imp
G_cl=G_loop/(1+G_loop);  %% Closed-loop TF
G_vgc=Gvg/(1+G_loop);       %% Closed-loop audio susc.

figure(3)
bode(G_loop,'b'); hold on; grid on;
bode(G_loop_delay,'r'); hold on; grid on;

%% Analog to Digital PID controller conversion
```

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
>>
```

(Refer Slide Time: 22:21)

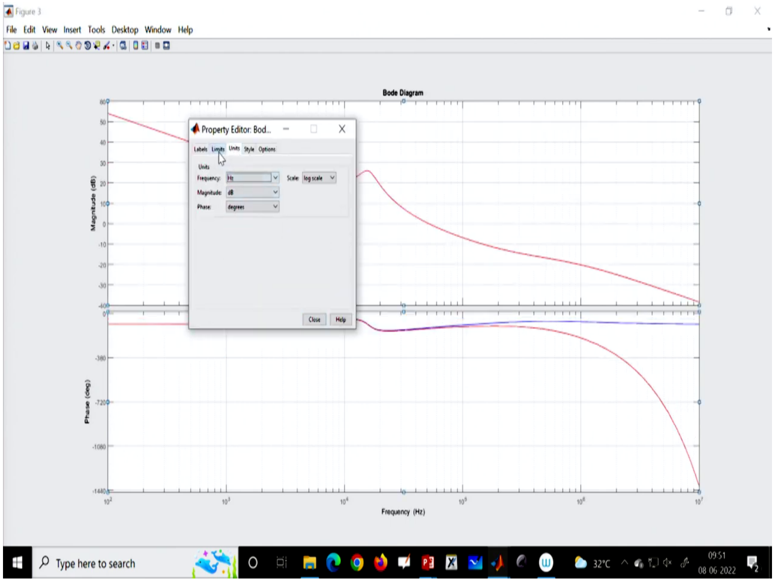


(Refer Slide Time: 22:22)



Now, we are running this and we want to check first our; so, we want to check.

(Refer Slide Time: 22:23)



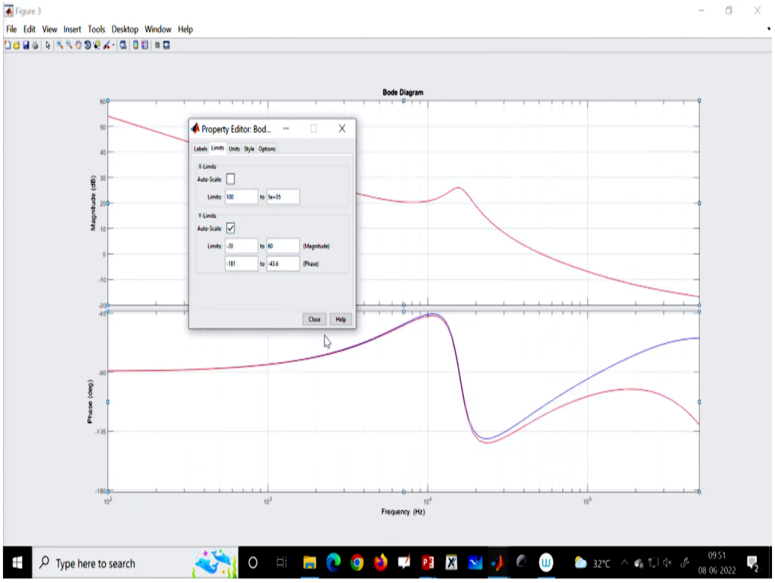
Units

Frequency: Hz   Scale: log scale

Magnitude: dB

Phase: degrees

(Refer Slide Time: 22:26)



Property Editor: Bod...

Labels | Limits | Units | Style | Options

X-Limits

Auto-Scale: □

Limits: 100 to 5e+05

Y-Limits

Auto-Scale: ☑

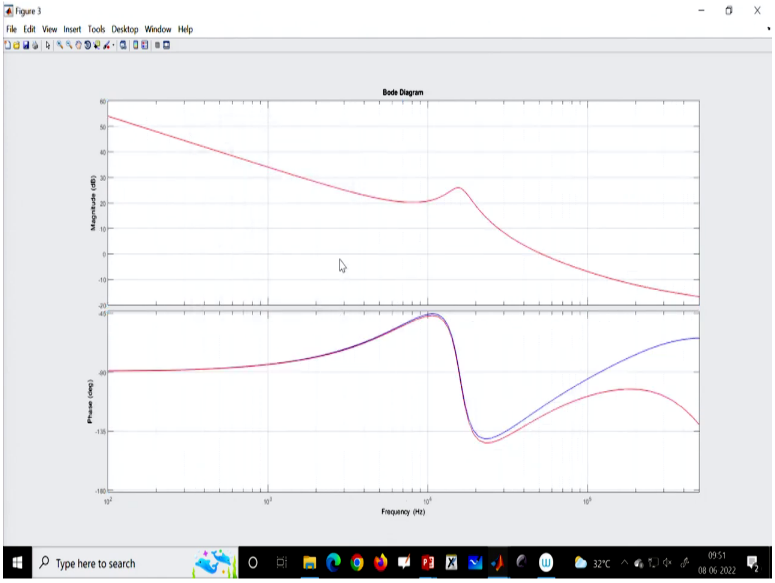
Limits: -20 to 60 (Magnitude)

-181 to -43.6 (Phase)

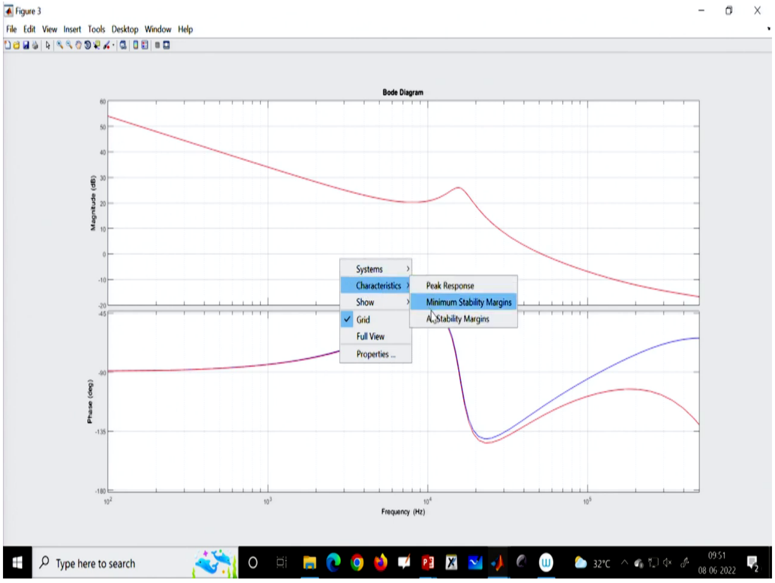
Close | Help

That means we want to use hertz and here also we want to limit r 6 0.5 OKs.

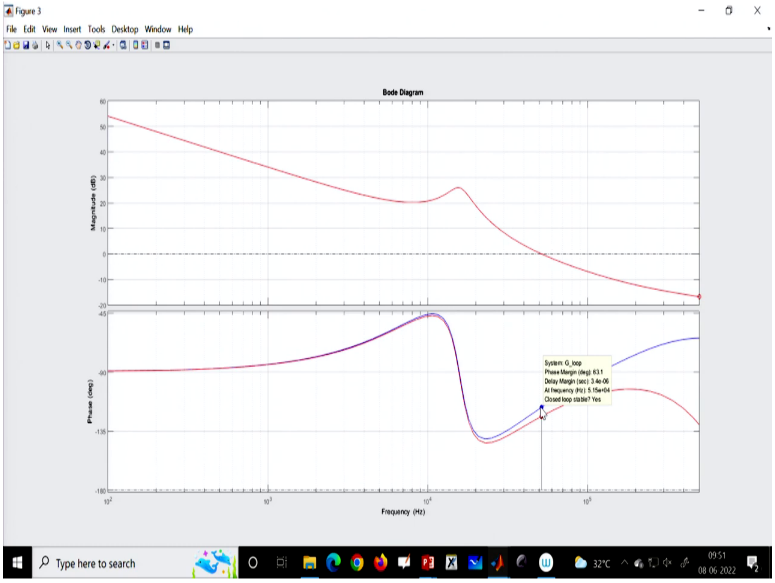
(Refer Slide Time: 22:36)



(Refer Slide Time: 22:37)



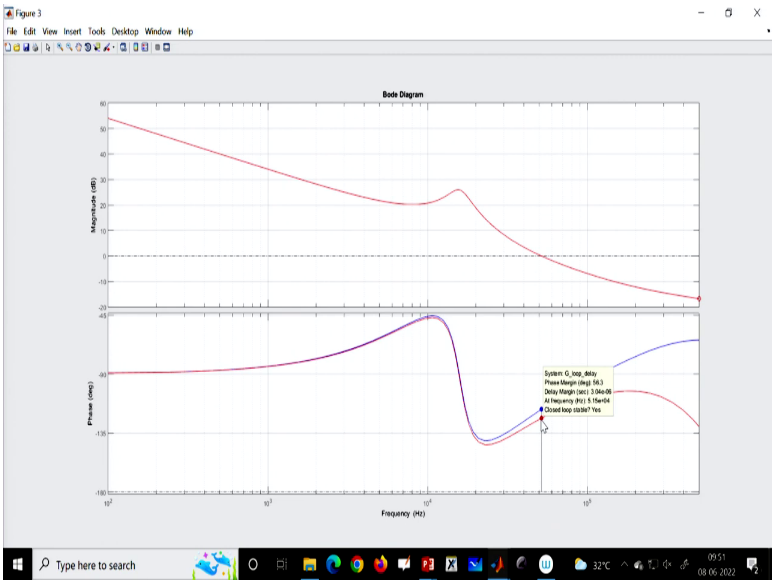
(Refer Slide Time: 22:38)



System: G_loop
Phase Margin (deg): 63.1
Delay Margin (sec): 3.4e-06
At frequency (Hz): 5.15e+04
Closed loop stable? Yes

And we want to check all the phase characteristics; that means, here it is like you know this is 10 kilohertz 20 30 40 50. So, it is roughly around 50; that means, it is 51.5, so nearly one-tenth of the switching frequency. That means, we have set the gain to achieve. Now in the case of analog control which is the blue line, you see the phase margin is 63.1 dB sorry degree.

(Refer Slide Time: 23:07)

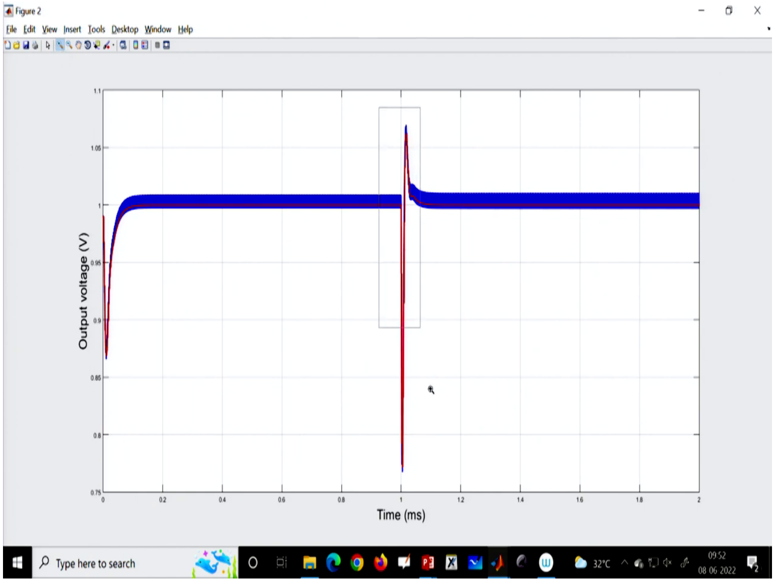


System: G_loop_delay
Phase Margin (deg): 56.3
Delay Margin (sec): 3.04e-06
At frequency (Hz): 5.15e+04
Closed loop stable? Yes

Whereas the digital control is 56; that means, there is just around an 8 to 10-degree phase r shift. That means, it is 63 and this is let us say 56. So, around 7-degree of phase degradation ok, and the phase margin is degraded by 7 degree. So, it is not a big deal because it is much higher than 45 degree. So, it should be ok. Now, we want to check here we are not doing any exact pole-zero cancellation, but another interesting point is you see the gain plot of the analog and digital are identical.

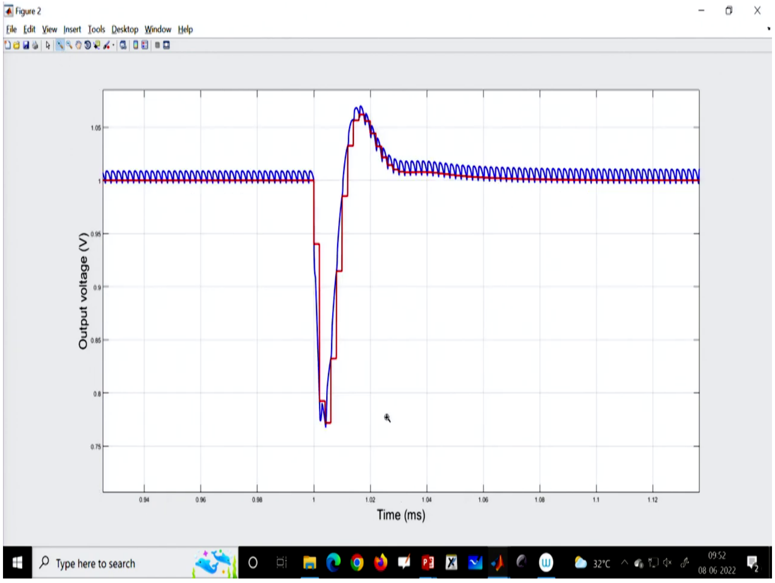
As I told you the delay will have no contribution to the gain because its magnitude is 1, but it will contribute to the phase and the phase will deviate drastically as we move further and further. Because it is omega into tau d as omega increases the phase lag also increases.

(Refer Slide Time: 24:02)



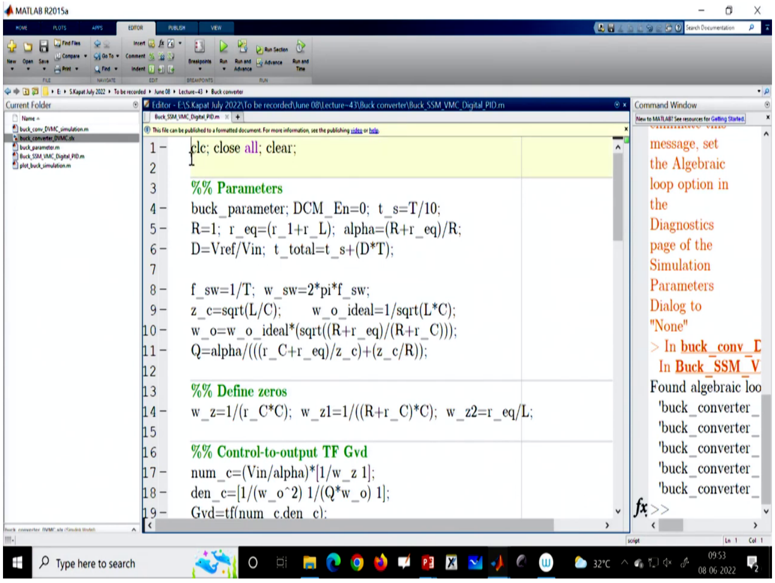
But interestingly if you check the load transient response in the output voltage, it is much better than the earlier solution.

(Refer Slide Time: 24:06)



But that means we are still using one-tenth of the switching frequency. We are achieving there as per the exact cancellation. Our loop transfer function shows close to more than 80 degree phase margin and with one-tenth of the switching figure. Here our phase margin is around 50 degree or plus. One-tenth of the switching frequency, but the transient response is much better in terms of undershoot overshoot. Now we want to compare this design with the earlier design. So, let us go and compare; that means, we have method one.

(Refer Slide Time: 24:39)



Editor - E:\S Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m

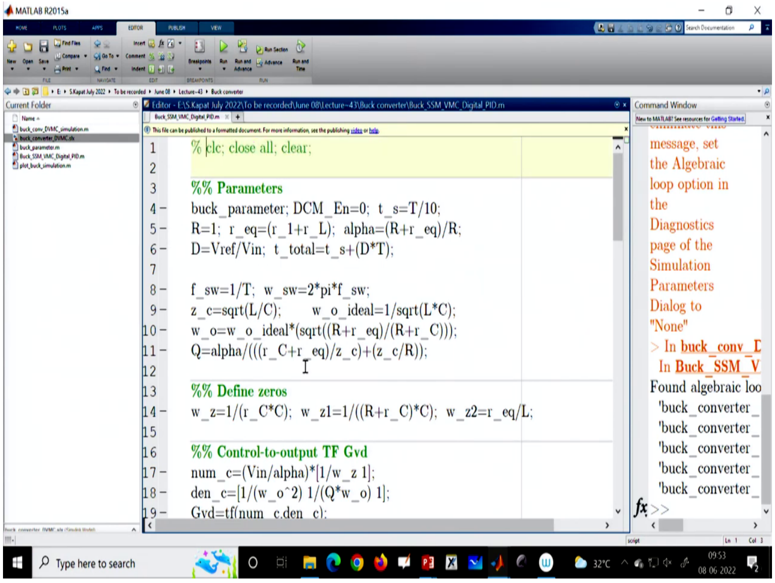
```
clc; close all; clear;

%% Parameters
buck_parameter; DCM_En=0; t_s=T/10;
R=1; r_eq=(r_1+r_L); alpha=(R+r_eq)/R;
D=Vref/Vin; t_total=t_s+(D*T);

f_sw=1/T; w_sw=2*pi*f_sw;
z_c=sqrt(L/C);        w_o_ideal=1/sqrt(L*C);
w_o=w_o_ideal*(sqrt((R+r_eq)/(R+r_C)));
Q=alpha/(((r_C+r_eq)/z_c)+(z_c/R));

%% Define zeros
w_z=1/(r_C*C); w_z1=1/((R+r_C)*C); w_z2=r_eq/L;

%% Control-to-output TF Gvd
num_c=(Vin/alpha)*[1/w_z 1];
den_c=[1/(w_o^2) 1/(Q*w_o) 1];
Gvd=tf(num_c,den_c);
```

Command Window

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

This is the alternative method.

(Refer Slide Time: 24:40)



Editor - E:\S Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m

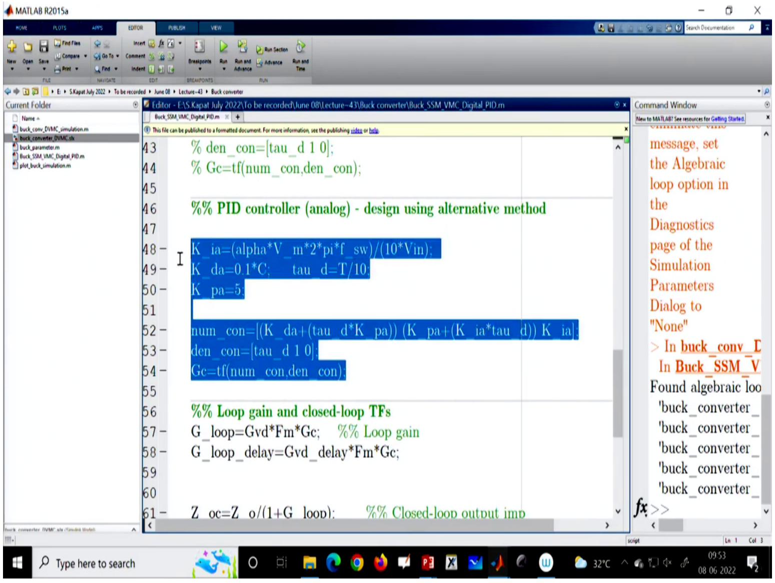
```
% clc; close all; clear;

%% Parameters
buck_parameter; DCM_En=0; t_s=T/10;
R=1; r_eq=(r_1+r_L); alpha=(R+r_eq)/R;
D=Vref/Vin; t_total=t_s+(D*T);

f_sw=1/T; w_sw=2*pi*f_sw;
z_c=sqrt(L/C);        w_o_ideal=1/sqrt(L*C);
w_o=w_o_ideal*(sqrt((R+r_eq)/(R+r_C)));
Q=alpha/(((r_C+r_eq)/z_c)+(z_c/R));

%% Define zeros
w_z=1/(r_C*C); w_z1=1/((R+r_C)*C); w_z2=r_eq/L;

%% Control-to-output TF Gvd
num_c=(Vin/alpha)*[1/w_z 1];
den_c=[1/(w_o^2) 1/(Q*w_o) 1];
Gvd=tf(num_c,den_c);
```

Command Window

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

And we want to continue with the earlier method.

(Refer Slide Time: 24:44)



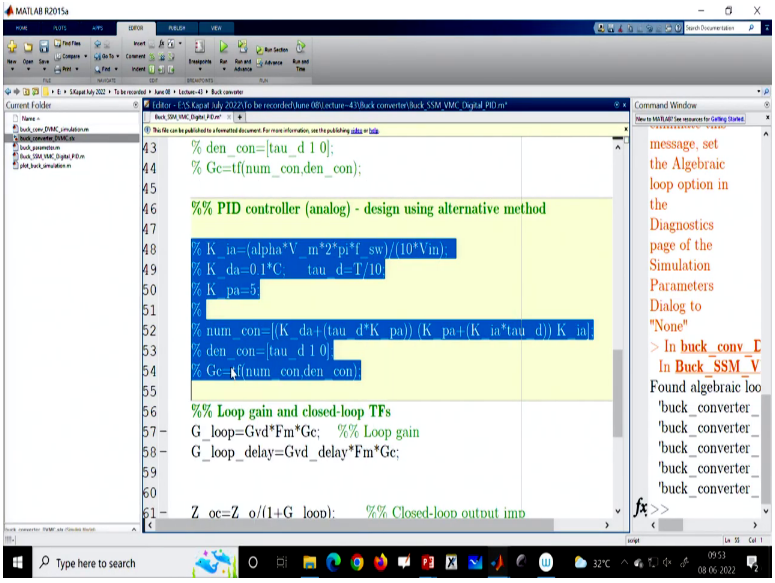
Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m

```
% den_con=[tau_d 1 0];
% Gc=tf(num_con,den_con);

%% PID controller (analog) - design using alternative method

K_ia=(alpha*V_m*2*pi*f_sw)/(10*Vin);
K_da=0.1*C;    tau_d=T/10;
K_pa=5;

num_con=[(K_da+(tau_d*K_pa)) (K_pa+(K_ia*tau_d)) K_ia];
den_con=[tau_d 1 0];
Gc=tf(num_con,den_con);

%% Loop gain and closed-loop TFs
G_loop=Gvd*Fm*Gc;    %% Loop gain
G_loop_delay=Gvd_delay*Fm*Gc;


Z_oc=Z_o/(1+G_loop);        %% Closed-loop output imp
```

Command Window

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

(Refer Slide Time: 24:49)



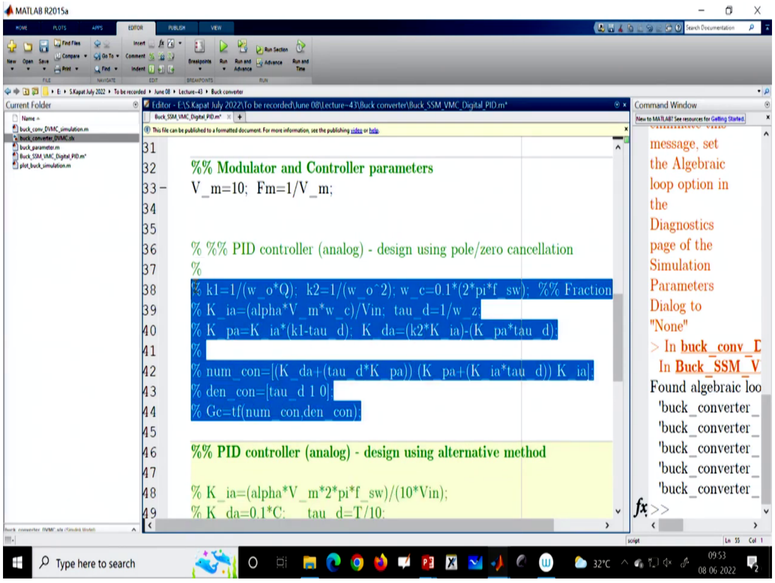
Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m*

```
% den_con=[tau_d 1 0];
% Gc=tf(num_con,den_con);

%% PID controller (analog) - design using alternative method

% K_ia=(alpha*V_m*2*pi*f_sw)/(10*Vin);
% K_da=0.1*C;    tau_d=T/10;
% K_pa=5;
%
% num_con=[(K_da+(tau_d*K_pa)) (K_pa+(K_ia*tau_d)) K_ia];
% den_con=[tau_d 1 0];
% Gc=tf(num_con,den_con);

%% Loop gain and closed-loop TFs
G_loop=Gvd*Fm*Gc;    %% Loop gain
G_loop_delay=Gvd_delay*Fm*Gc;


Z_oc=Z_o/(1+G_loop);        %% Closed-loop output imp
```

Command Window

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

(Refer Slide Time: 24:50)

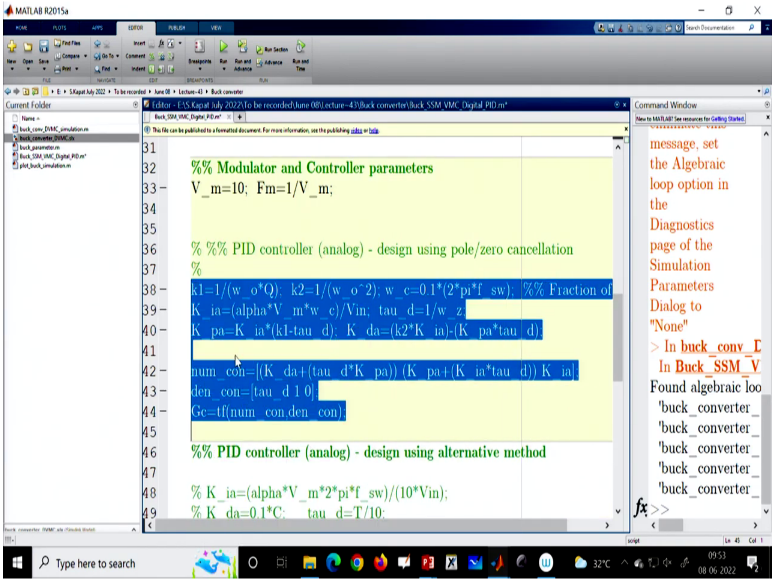


```
%% Modulator and Controller parameters
V_m=10;  Fm=1/V_m;


% %% PID controller (analog) - design using pole/zero cancellation
%
% k1=1/(w_o*Q);  k2=1/(w_o^2); w_c=0.1*(2*pi*f_sw);  %% Fraction
% K_ia=(alpha*V_m*w_c)/Vin;  tau_d=1/w_z;
% K_pa=K_ia*(k1-tau_d);  K_da=(k2*K_ia)-(K_pa*tau_d);
%
% num_con=[(K_da+(tau_d*K_pa)) (K_pa+(K_ia*tau_d)) K_ia];
% den_con=[tau_d 1 0];
% Gc=tf(num_con,den_con);

%% PID controller (analog) - design using alternative method

% K_ia=(alpha*V_m*2*pi*f_sw)/(10*Vin);
% K_da=0.1*C;    tau_d=T/10;
```

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
 'buck_converter_
 'buck_converter_
 'buck_converter_
 'buck_converter_
 'buck_converter_
fx >>
```

So, here we want to comment on this line comment and we want to discuss this alternative method.

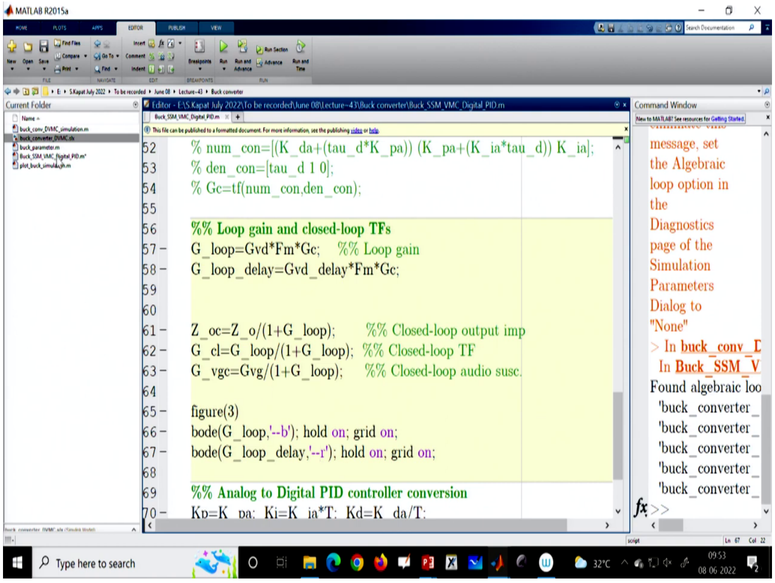
(Refer Slide Time: 24:53)



```
%% Modulator and Controller parameters
V_m=10;  Fm=1/V_m;


% %% PID controller (analog) - design using pole/zero cancellation
%
k1=1/(w_o*Q);  k2=1/(w_o^2); w_c=0.1*(2*pi*f_sw);  %% Fraction of
K_ia=(alpha*V_m*w_c)/Vin;  tau_d=1/w_z;
K_pa=K_ia*(k1-tau_d);  K_da=(k2*K_ia)-(K_pa*tau_d);

num_con=[(K_da+(tau_d*K_pa)) (K_pa+(K_ia*tau_d)) K_ia];
den_con=[tau_d 1 0];
Gc=tf(num_con,den_con);

%% PID controller (analog) - design using alternative method

% K_ia=(alpha*V_m*2*pi*f_sw)/(10*Vin);
% K_da=0.1*C;    tau_d=T/10;
```

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
 'buck_converter_
 'buck_converter_
 'buck_converter_
 'buck_converter_
 'buck_converter_
fx >>
```

(Refer Slide Time: 24:46)



Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\Buck_SSM_VMC_Digital_PID.m

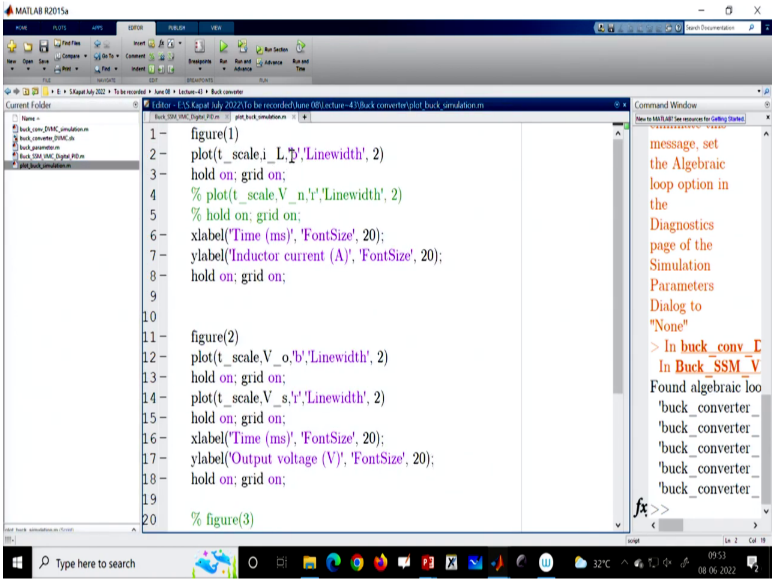
This file can be published to a formatted document. For more information, see the publishing video or help.

```
% num_con=[(K_da+(tau_d*K_pa)) (K_pa+(K_ia*tau_d)) K_ia];
% den_con=[tau_d 1 0];
% Gc=tf(num_con,den_con);

%% Loop gain and closed-loop TFs
G_loop=Gvd*Fm*Gc;    %% Loop gain
G_loop_delay=Gvd_delay*Fm*Gc;


Z_oc=Z_o/(1+G_loop);      %% Closed-loop output imp
G_cl=G_loop/(1+G_loop);  %% Closed-loop TF
G_vgc=Gvg/(1+G_loop);     %% Closed-loop audio susc.

figure(3)
bode(G_loop,'--b'); hold on; grid on;
bode(G_loop_delay,'--r'); hold on; grid on;

%% Analog to Digital PID controller conversion
Kp=K_pa; Ki=K_ia*T; Kd=K_da/T;
```

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

And we want to use a different color; that means, the bode plot also dotted all the dotted, and if we sorry dotted and we want to discuss I mean plot here.

(Refer Slide Time: 25:07)

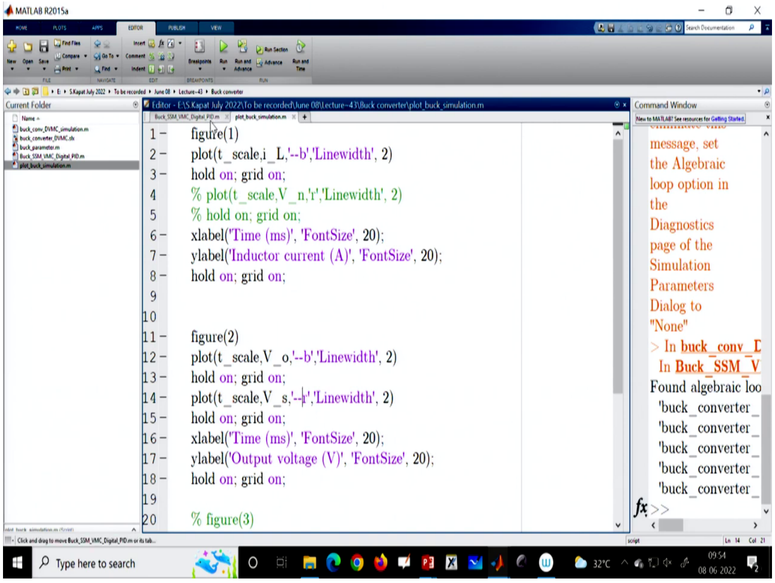


Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Buck converter\plot_buck_simulation.m

```
figure(1)
plot(t_scale,i_L,'b','Linewidth', 2)
hold on; grid on;
% plot(t_scale,V_n,'r','Linewidth', 2)
% hold on; grid on;
xlabel('Time (ms)', 'FontSize', 20);
ylabel('Inductor current (A)', 'FontSize', 20);
hold on; grid on;


figure(2)
plot(t_scale,V_o,'b','Linewidth', 2)
hold on; grid on;
plot(t_scale,V_s,'r','Linewidth', 2)
hold on; grid on;
xlabel('Time (ms)', 'FontSize', 20);
ylabel('Output voltage (V)', 'FontSize', 20);
hold on; grid on;

% figure(3)
```

Here also we want to use dotted.

(Refer Slide Time: 25:10)

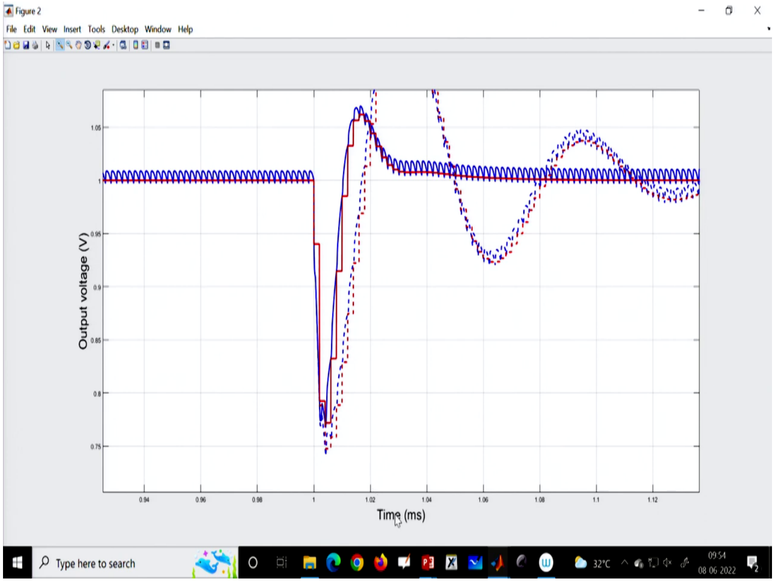


```
figure(1)
plot(t_scale,i_L,'--b','Linewidth', 2)
hold on; grid on;
% plot(t_scale,V_n,'r','Linewidth', 2)
% hold on; grid on;
xlabel('Time (ms)', 'FontSize', 20);
ylabel('Inductor current (A)', 'FontSize', 20);
hold on; grid on;


figure(2)
plot(t_scale,V_o,'--b','Linewidth', 2)
hold on; grid on;
plot(t_scale,V_s,'--r','Linewidth', 2)
hold on; grid on;
xlabel('Time (ms)', 'FontSize', 20);
ylabel('Output voltage (V)', 'FontSize', 20);
hold on; grid on;

% figure(3)
```

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
 'buck_converter_
 'buck_converter_
 'buck_converter_
 'buck_converter_
 'buck_converter_
fx >>
```

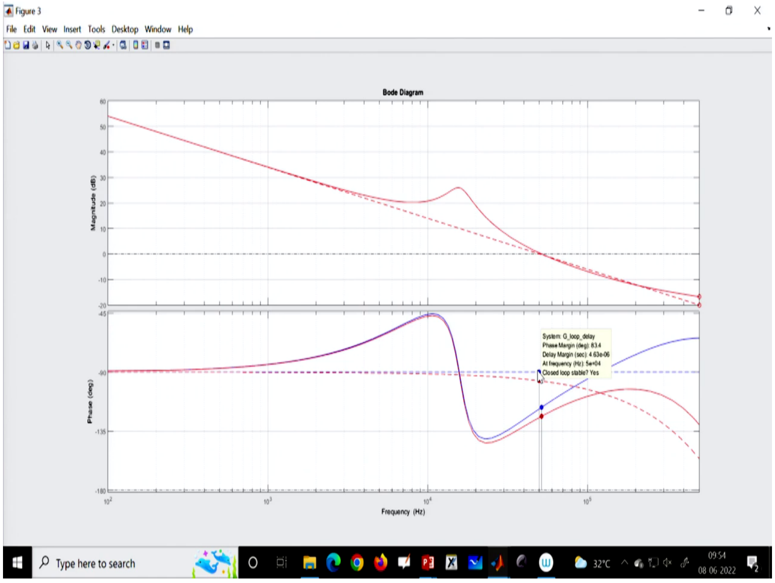
So, all dotted dash lines will be for the second design. So, let us run the simulation and we want to compare.

(Refer Slide Time: 25:20)



Now, I want to show before going to the transient response if the phase plot ok.

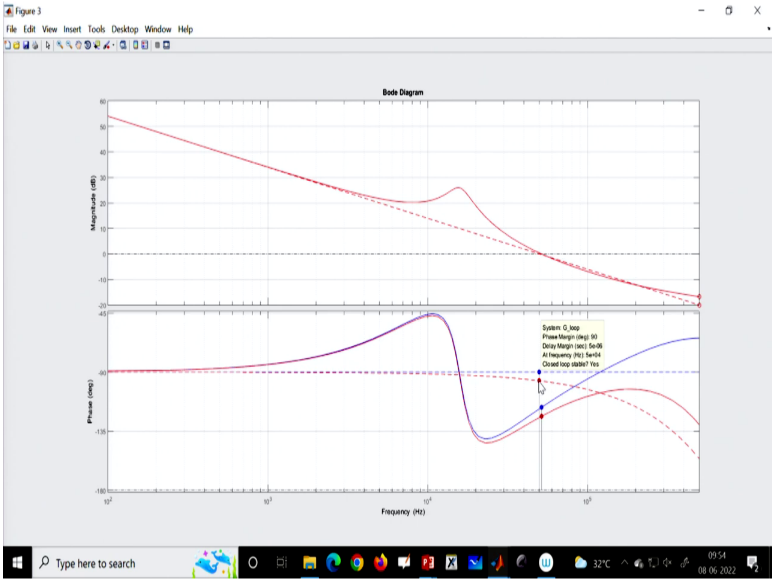
(Refer Slide Time: 25:24)



System: G_loop_delay
Phase Margin (deg): 83.4
Delay Margin (sec): 4.63e-06
At frequency (Hz): 5e+04
Closed loop stable? Yes

So, I want to show you know you can see that if we go by the traditional pole-zero cancellation the analog control we are getting 90 degree phase margin and which actually because of the exact cancellation and that is practically impossible,

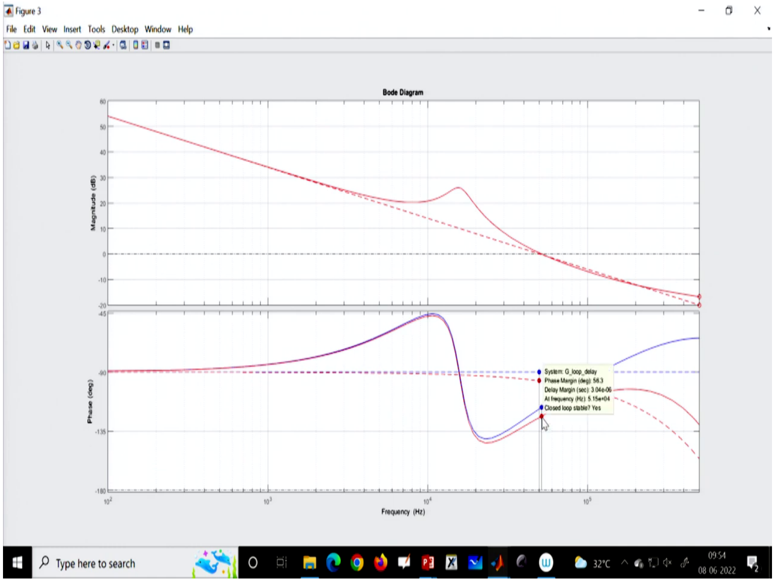
(Refer Slide Time: 25:38)



System: G_loop
Phase Margin (deg): 90
Delay Margin (sec): 5e-06
At frequency (Hz): 5e+04
Closed loop stable? Yes

But, if we incorporate digital delay then we are losing a phase and this phase is around 7 degree we are losing. So, 83.4, but the gain crossover frequency is 50 kilohertz it is.

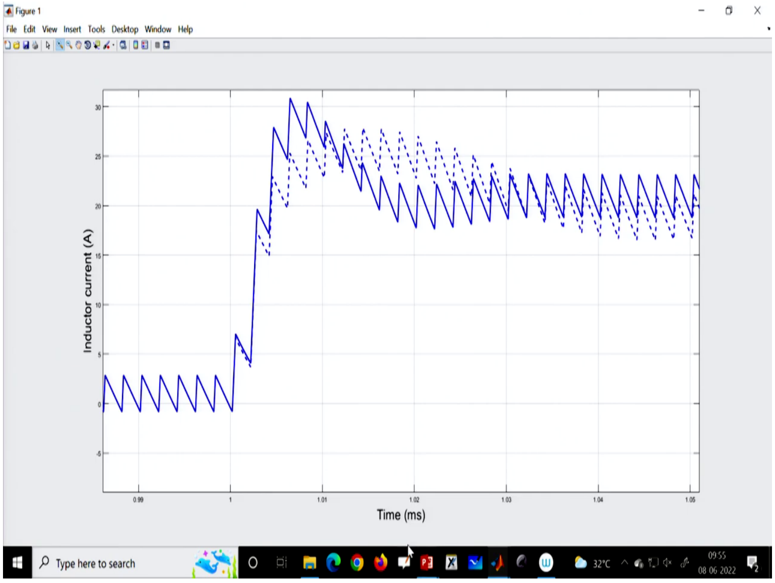
(Refer Slide Time: 25:59)



And the other design we got slightly higher and the phase margin is much lower and you can see since we are not doing any exact cancellation. So, there can be an lc pole effect. That means the phase you can say there is a phase swing, and also amplitude has a peaking effect. Now we want to compare what is our transient response. So, you want to see the transient response, and if you compare the transient response that you can see that the dash line is coming from the exact cancellation and that is much poor.

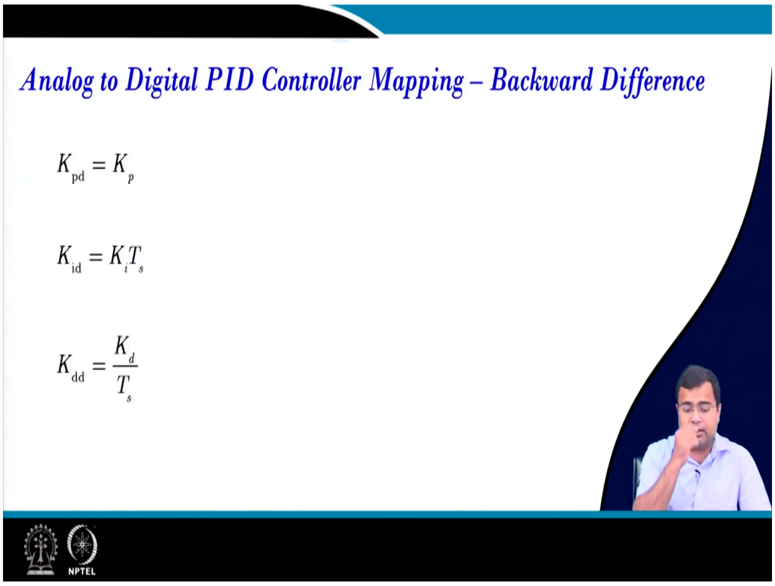
Because it has more like you know overshoot undershoot as well as the settling time is also long longer, whereas the alternative design approach is far better and it can settle much faster.

(Refer Slide Time: 26:49)



And if you take the inductor current waveform there also you can see you know not drastically different yeah. So, in summary, we want to discuss that this alternative approach is much better. Because we are visualizing this controller as if the derivative controller is carrying the current and the load information and that makes the whole design perspective different.

(Refer Slide Time: 27:10)



Analog to Digital PID Controller Mapping – Backward Difference

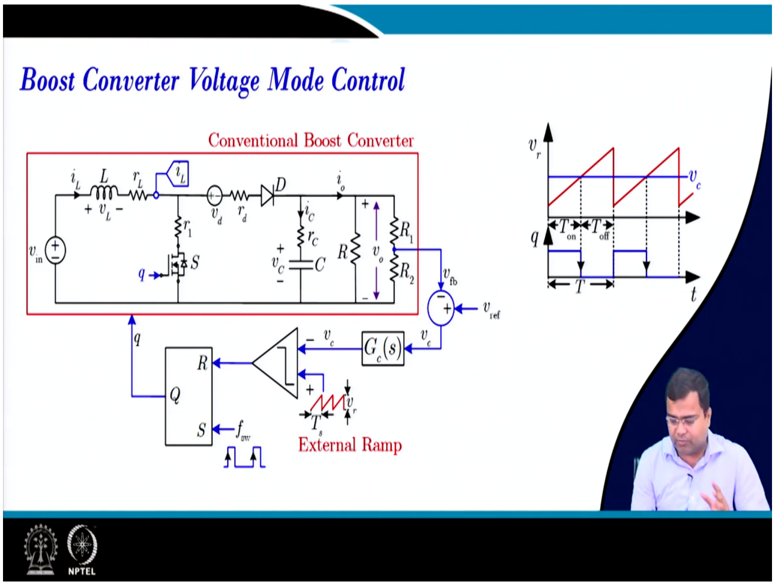
$$K_{pd} = K_p$$

$$K_{id} = K_i T_s$$

$$K_{dd} = \frac{K_d}{T_s}$$

Now, again for the same design if we get K p K i K d in analog then we know just we have to convert it into digital using the backward difference formula, and that we have already discussed.

(Refer Slide Time: 27:22)



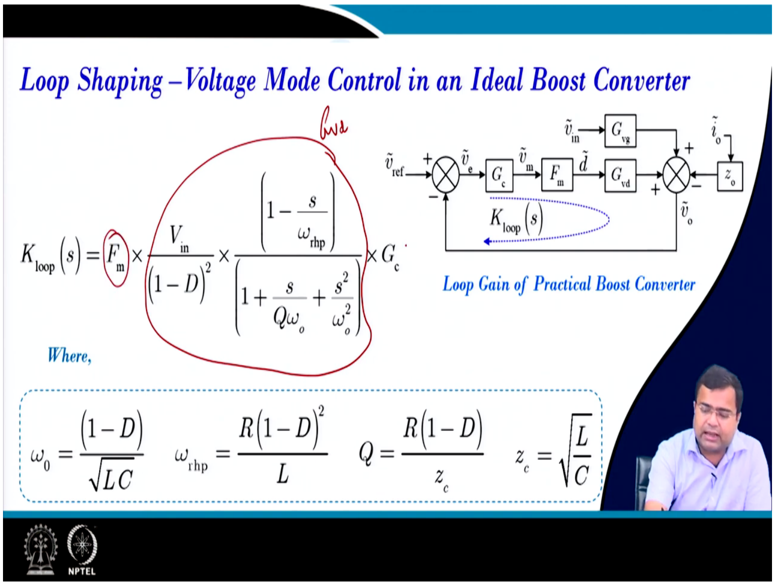
Boost Converter Voltage Mode Control

Conventional Boost Converter

External Ramp

Now, we are talking about a boost converter voltage mode control. So, this is the analog voltage mode control of a boost converter.

(Refer Slide Time: 27:28)



Loop Shaping – Voltage Mode Control in an Ideal Boost Converter

$$K_{loop}(s) = F_m \times \underbrace{\frac{V_{in}}{(1-D)^2} \times \frac{\left(1-\frac{s}{\omega_{rhp}}\right)}{\left(1+\frac{s}{Q\omega_o}+\frac{s^2}{\omega_o^2}\right)}}_{G_{vd}} \times G_c$$

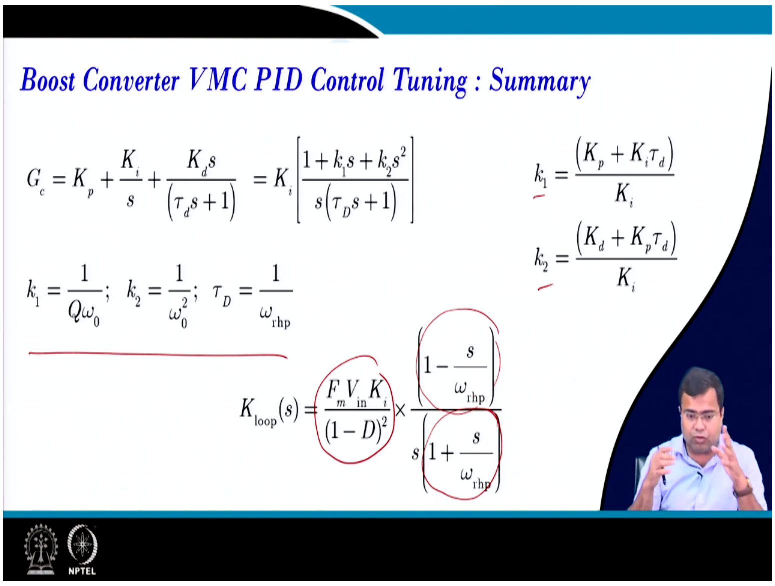
Where,

$$\omega_0 = \frac{(1-D)}{\sqrt{LC}} \qquad \omega_{rhp} = \frac{R(1-D)^2}{L} \qquad Q = \frac{R(1-D)}{z_c} \qquad z_c = \sqrt{\frac{L}{C}}$$

Loop Gain of Practical Boost Converter

And now this is the loop transfer function of the boost converter. Here we are talking about an ideal boost converter and this particular expression shows the boost converter G vd which is the control to the output transfer function. Again this is a modulator gain and this is a controller.

(Refer Slide Time: 27:47)



Boost Converter VMC PID Control Tuning : Summary

$$G_c = K_p + \frac{K_i}{s} + \frac{K_d s}{(\tau_d s+1)} = K_i\left[\frac{1+k_1 s+k_2 s^2}{s(\tau_D s+1)}\right]$$

$$k_1 = \frac{(K_p + K_i\tau_d)}{K_i}$$

$$k_2 = \frac{(K_d + K_p\tau_d)}{K_i}$$

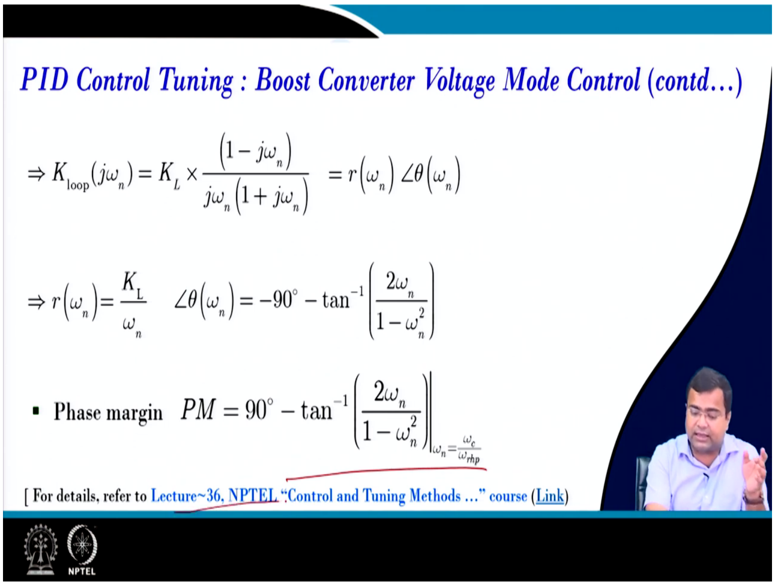
$$k_1 = \frac{1}{Q\omega_0};\; k_2 = \frac{1}{\omega_0^2};\; \tau_D = \frac{1}{\omega_{\text{rhp}}}$$

$$K_{\text{loop}}(s) = \frac{F_m V_{\text{in}} K_i}{(1-D)^2} \times \frac{\left(1-\frac{s}{\omega_{\text{rhp}}}\right)}{s\left(1+\frac{s}{\omega_{\text{rhp}}}\right)}$$

Now, we want to design a PID controller for a boost converter and we know that in a boost converter the exact stable pole-zero cancellation that we have discussed in lecture number 42. These are the parameter. And these parameters are a function of K p K i tau d and we can obtain K p K i d.

Now, if we do go by exact pole-zero cancellation stable pole-zero. We know in the boost converter we will get a loop transfer function like this. There will be some gain than 1 by s term and then we are placing the controller pole in coincidence with the rhp 0 of the control to the output transfer function.

(Refer Slide Time: 28:27)



PID Control Tuning : Boost Converter Voltage Mode Control (contd…)

$$\Rightarrow K_{\text{loop}}(j\omega_n) = K_L \times \frac{(1-j\omega_n)}{j\omega_n(1+j\omega_n)} = r(\omega_n)\angle\theta(\omega_n)$$

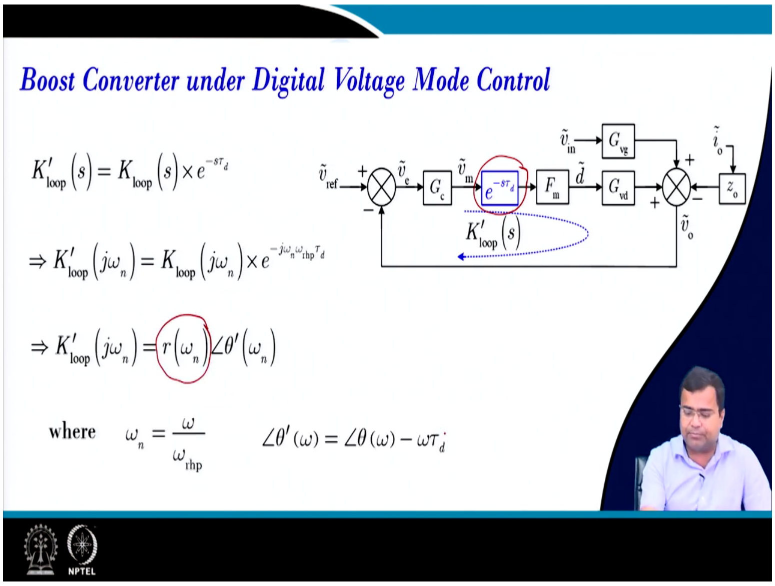
$$\Rightarrow r(\omega_n) = \frac{K_L}{\omega_n} \qquad \angle\theta(\omega_n) = -90^\circ - \tan^{-1}\left(\frac{2\omega_n}{1-\omega_n^2}\right)$$

- Phase margin $PM = 90^\circ - \tan^{-1}\left(\frac{2\omega_n}{1-\omega_n^2}\right)\Bigg|_{\omega_n=\frac{\omega_c}{\omega_{rhp}}}$

[For details, refer to Lecture~36, NPTEL "Control and Tuning Methods …" course (Link)]

Then we also know the traditional way of finding the loop frequency response of the loop transfer function and this expression we already know and this we have discussed in lecture number 42 in this course as well as 36 in the earlier NPTEL course.

(Refer Slide Time: 28:44)



Boost Converter under Digital Voltage Mode Control

$$K'_{\text{loop}}(s) = K_{\text{loop}}(s) \times e^{-s\tau_d}$$

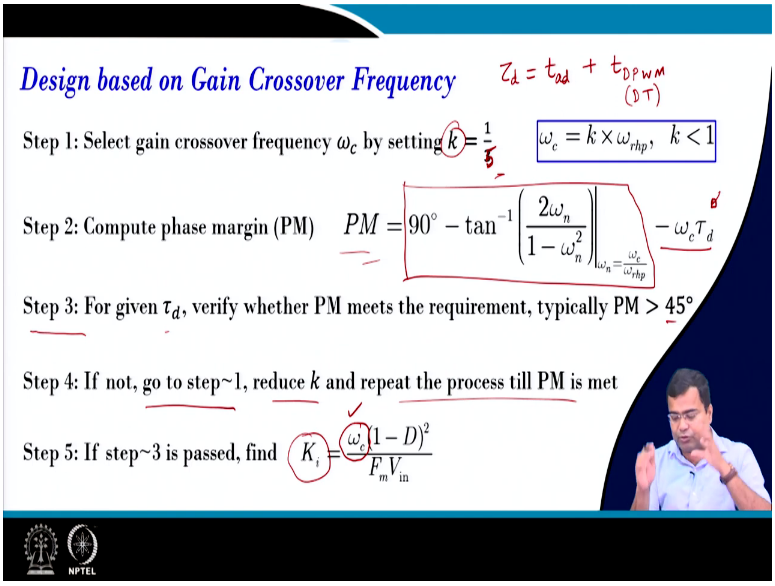
$$\Rightarrow K'_{\text{loop}}(j\omega_n) = K_{\text{loop}}(j\omega_n) \times e^{-j\omega_n \omega_{\text{rhp}} \tau_d}$$

$$\Rightarrow K'_{\text{loop}}(j\omega_n) = r(\omega_n) \angle\theta'(\omega_n)$$

where $\omega_n = \dfrac{\omega}{\omega_{\text{rhp}}}$ $\angle\theta'(\omega) = \angle\theta(\omega) - \omega\tau_d$

Now, if you take a digital voltage mode control again we are incorporating just a delay and we are continuing with the same continuous time small signal model again we can show that this gain plot will not be affected, but the phase will be affected because of this delay.

(Refer Slide Time: 29:00)



Design based on Gain Crossover Frequency

$\tau_d = t_{ad} + t_{DPWM}$ (DT)

Step 1: Select gain crossover frequency ω_c by setting $k = \frac{1}{5}$ $\boxed{\omega_c = k \times \omega_{rhp}, \quad k < 1}$

Step 2: Compute phase margin (PM) $PM = \boxed{90^\circ - \tan^{-1}\left(\dfrac{2\omega_n}{1-\omega_n^2}\right)\Bigg|_{\omega_n = \frac{\omega_c}{\omega_{rhp}}}} - \omega_c \tau_d$

Step 3: For given τ_d, verify whether PM meets the requirement, typically PM > 45°

Step 4: If not, go to step~1, reduce k and repeat the process till PM is met

Step 5: If step~3 is passed, find $K_i = \dfrac{\omega_c (1-D)^2}{F_m V_{\text{in}}}$

And then if we want to design based on the gain crossover frequency criteria. So, we need to set the gain crossover frequency, since it is a voltage mode control. So, typically it is better to use one-fifth of the rhp 0. It should be smaller and we know that we have discussed this in lecture 36 in our earlier NPTEL course. So, we set the gain crossover frequency to one-fifth of the rhp 0, then you can compute the phase margin.

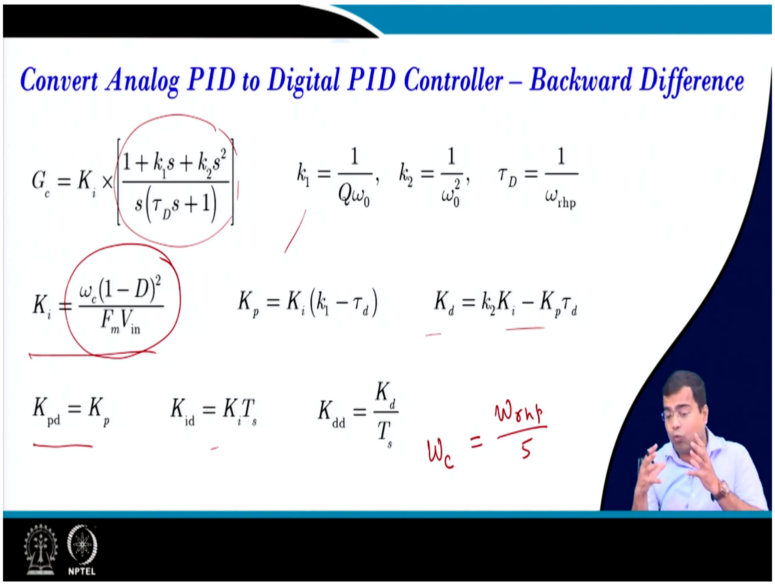
And this phase margin computation we know from the analog control voltage mode control in the case of boost converter, but because of the digital control the additional term will come due to the delay which is a linear phase. Then for given tau d because we have to find how much propagation and how much the delay is there in the digital contour. That means, this tau d again what is this tau d? It is the adhesive conversion time plus the DPWM delay.

And DPWM delay is nothing but D into T and this is whatever you take; that means, the conversion time. Now for given tau d, you have to check whether using this analytical computation can meet more than 45 degree phase margin, if it is not then you may have to reduce this factor so that your crossover frequency has to reduce further.

So, it will impose a constant in the gain crossover frequency which is nothing but the closed-loop bandwidth. Now if one; that means, if step 3 is not satisfied then we have to go to step 1 reduce K p and repeat the same process till we meet the required phase margin criteria. Once it is done then you are set with the phase margin. That means you can obtain what is my gain crossover frequency that will be known from this expression, this expression then you can find out K i by this formula; that means, the integral gain is known.

So, only integral gain was unknown other proportional derivatives everything was known from exact pole-zero stable pole-zero cancellation.

(Refer Slide Time: 31:04)



Convert Analog PID to Digital PID Controller – Backward Difference

$$G_c = K_i \times \left[\frac{1+k_1 s+k_2 s^2}{s(\tau_D s+1)}\right] \qquad k_1 = \frac{1}{Q\omega_0},\quad k_2 = \frac{1}{\omega_0^2},\quad \tau_D = \frac{1}{\omega_{\text{rhp}}}$$

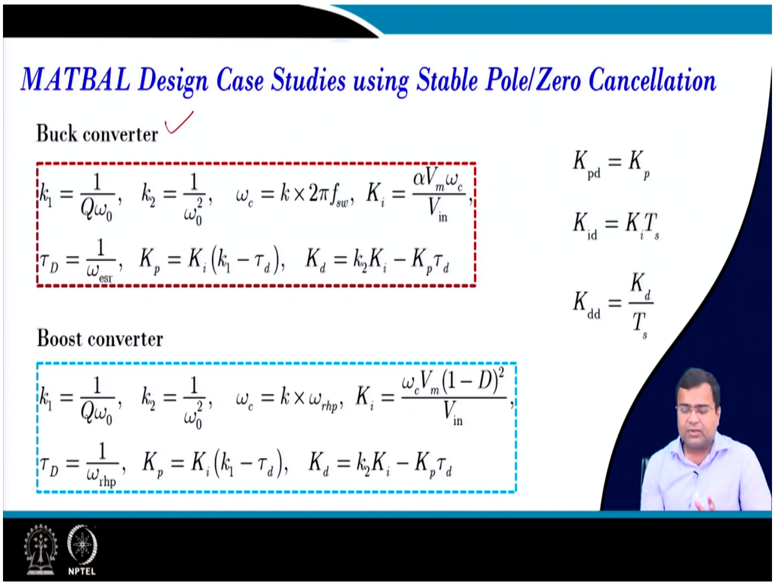
$$K_i = \frac{\omega_c(1-D)^2}{F_m V_{\text{in}}} \qquad K_p = K_i(k_1-\tau_d) \qquad K_d = k_2K_i - K_p\tau_d$$

$$K_{\text{pd}} = K_p \qquad K_{\text{id}} = K_i T_s \qquad K_{\text{dd}} = \frac{K_d}{T_s}$$

$$W_c = \frac{W_{rhp}}{5}$$

So, again we can convert; that means if we summarize k 1 k 2 k 3 and tau d here is the 0 k 1 K i is just now we got, K p can be written in terms of k 1 tau d K d can be written. So, if you take this expression and it is coming from a practical PID controller, then we can convert this gain into discrete time.

(Refer Slide Time: 31:29)



MATBAL Design Case Studies using Stable Pole/Zero Cancellation

Buck converter ✓

$$k_1 = \frac{1}{Q\omega_0},\quad k_2 = \frac{1}{\omega_0^2},\quad \omega_c = k\times 2\pi f_{sw},\quad K_i = \frac{\alpha V_m\omega_c}{V_{\text{in}}},$$

$$\tau_D = \frac{1}{\omega_{\text{esr}}},\quad K_p = K_i(k_1-\tau_d),\quad K_d = k_2K_i - K_p\tau_d$$

Boost converter

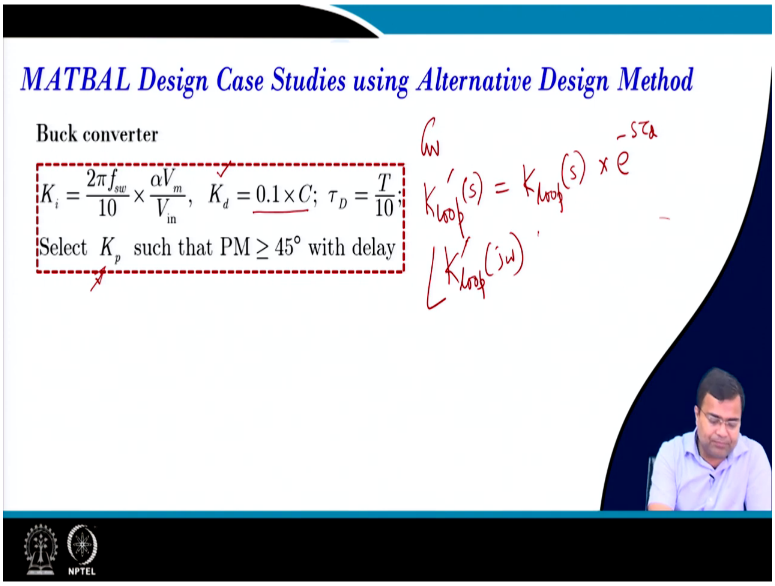
$$k_1 = \frac{1}{Q\omega_0},\quad k_2 = \frac{1}{\omega_0^2},\quad \omega_c = k\times \omega_{rhp},\quad K_i = \frac{\omega_c V_m(1-D)^2}{V_{\text{in}}},$$

$$\tau_D = \frac{1}{\omega_{\text{rhp}}},\quad K_p = K_i(k_1-\tau_d),\quad K_d = k_2K_i - K_p\tau_d$$

$$K_{\text{pd}} = K_p \qquad K_{\text{id}} = K_i T_s \qquad K_{\text{dd}} = \frac{K_d}{T_s}$$

So, we want to show a MATLAB design case study using stable pole-zero cancellation and for buck converter, we have already shown; that means, the case study, these are the design summary. For the boost converter, we may not want to go by this method we can go it, but we want to go by the alternative method.

(Refer Slide Time: 31:52)



MATBAL Design Case Studies using Alternative Design Method

Buck converter

$K_i = \frac{2\pi f_{sw}}{10}\times\frac{\alpha V_m}{V_{in}},\ K_d = 0.1\times C;\ \tau_D = \frac{T}{10};$

Select K_p such that PM $\geq 45°$ with delay

G_{vd}

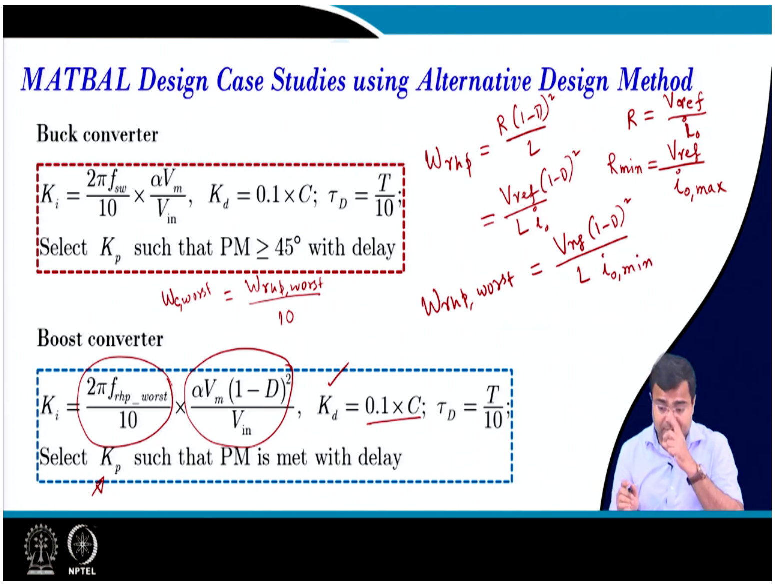
$K'_{loop}(s) = K_{loop}(s)\times e^{-s\tau_d}$

$\angle K'_{loop}(j\omega)$

That means the alternative method says that for the buck converter, we have to set K d equal to 0.1 time C tau d t by 10, and K i is this. Then we have to select K p in such a way; that means, our phase margin should be more than 45 degree and we know that G vd. That means the loop transfer function right which is for the analog-digital control is nothing but the loop transfer function of the analog control into e to the power minus s tau d.

So, for including this delay we have to set K p in such a way the loop transfer function if you take the loop this j omega this one. So, the phase margin of this should be; that means, what is the phase that if you compute the phase margin it should be greater than 45 degree ok.

(Refer Slide Time: 32:48)



MATBAL Design Case Studies using Alternative Design Method

Buck converter

$K_i = \frac{2\pi f_{sw}}{10} \times \frac{\alpha V_m}{V_{in}}, \quad K_d = 0.1 \times C; \; \tau_D = \frac{T}{10};$

Select K_p such that PM ≥ 45° with delay

$\omega_{rhp} = \frac{R(1-D)^2}{L}$ $\quad R = \frac{V_{ref}}{i_o}$

$= \frac{V_{ref}(1-D)^2}{L i_o}$ $\quad R_{min} = \frac{V_{ref}}{i_{o,max}}$

$\omega_{rhp,worst} = \frac{V_{ref}(1-D)^2}{L\, i_{o,min}}$

$\omega_{c,worst} = \frac{\omega_{rhp,worst}}{10}$

Boost converter

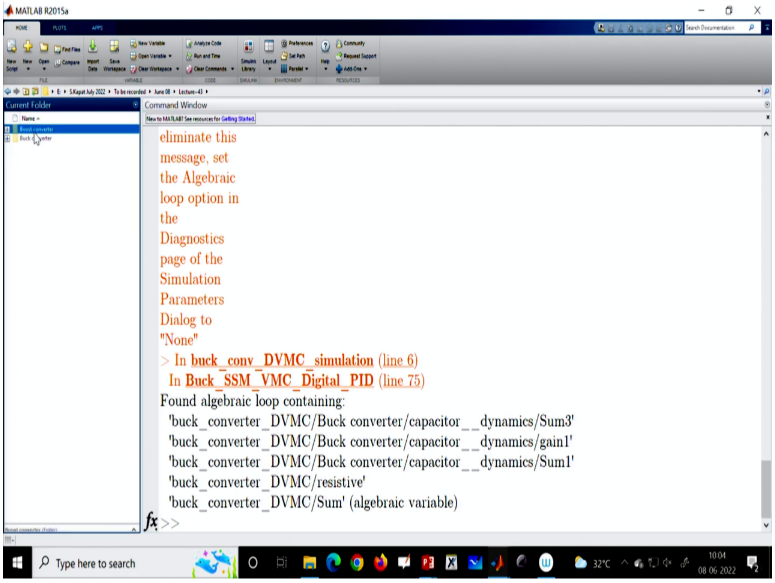
$K_i = \frac{2\pi f_{rhp\_worst}}{10} \times \frac{\alpha V_m (1-D)^2}{V_{in}}, \quad K_d = 0.1 \times C; \; \tau_D = \frac{T}{10};$

Select K_p such that PM is met with delay

So, then for the boost converter we have to take K i because we have discussed the K i if you go back to K i in the boost converter. K I was coming to this and we know what is omega c for a boost converter typically it is rhp 0 by 5 or even less. That means if you want to consider the worst case we have to take the worst case rhp 0 and omega c we are taking like one-tenth of the rhp. That means, for the integral gain selection; that means, we have selected here in this case this particular expression is nothing but omega c worst case; that means, the worst case divided by 10.

Sorry, omega c worst case which is nothing but omega c worst case which is nothing but we are talking about the rhp 0 worst divided by 10 just to be on the safe side. And that means, it is 2 pi and there is the expression, and this expression we already have from the boost converter design. So, that means, K i we know K d as we said we can set just simply 0.1 times C tau d T by 10. Now we have to play with K p such that we can meet certain phase margins including the delay due to the digital controller. So, let us go to the case study of this boost converter.

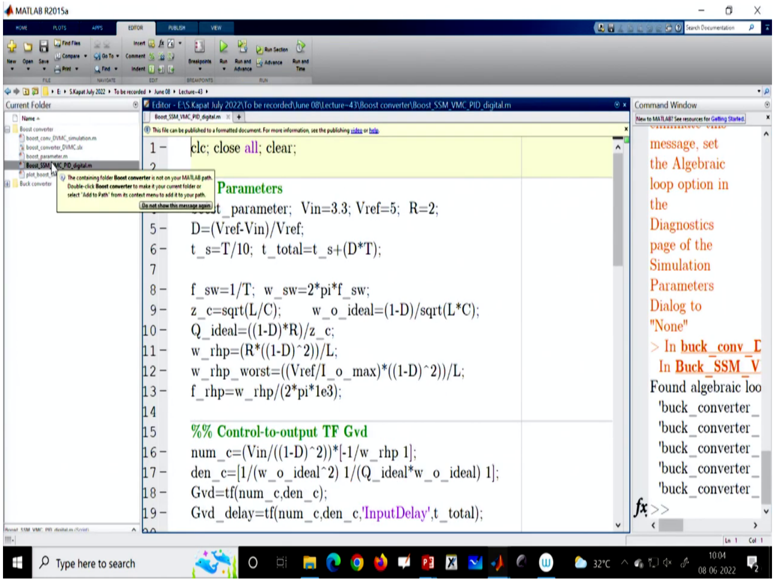
(Refer Slide Time: 34:21)



```
eliminate this
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_DVMC_simulation (line 6)
  In Buck_SSM_VMC_Digital_PID (line 75)
Found algebraic loop containing:
  'buck_converter_DVMC/Buck converter/capacitor__dynamics/Sum3'
  'buck_converter_DVMC/Buck converter/capacitor__dynamics/gain1'
  'buck_converter_DVMC/Buck converter/capacitor__dynamics/Sum1'
  'buck_converter_DVMC/resistive'
  'buck_converter_DVMC/Sum' (algebraic variable)
fx >>
```

Here we are talking about now if you go to the other folder. So, this is the boost converter case study.

(Refer Slide Time: 34:26)



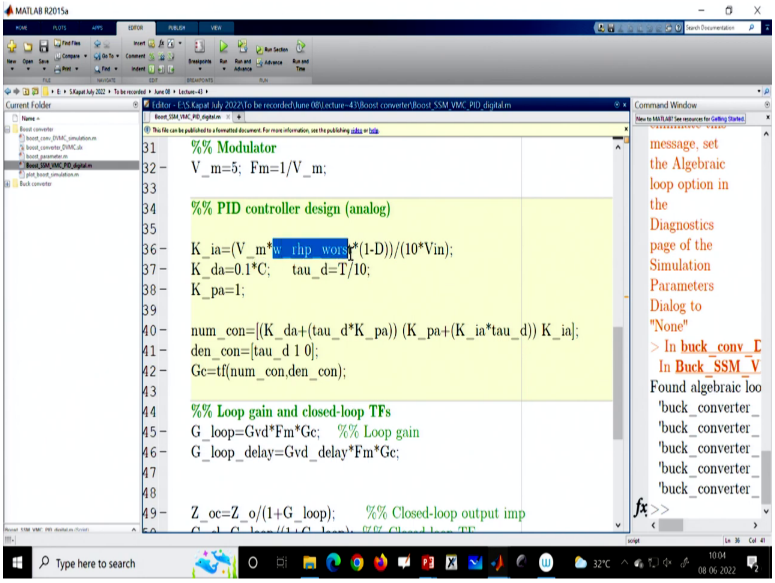
Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Boost converter\Boost_SSM_VMC_PID_digital.m

The containing folder Boost converter is not on your MATLAB path. Double-click Boost converter to make it your current folder or select 'Add to Path' from its context menu to add it to your path.

```
clc; close all; clear;

Parameters
t_parameter;  Vin=3.3; Vref=5;  R=2;
D=(Vref-Vin)/Vref;
t_s=T/10;  t_total=t_s+(D*T);

f_sw=1/T;  w_sw=2*pi*f_sw;
z_c=sqrt(L/C);       w_o_ideal=(1-D)/sqrt(L*C);
Q_ideal=((1-D)*R)/z_c;
w_rhp=(R*((1-D)^2))/L;
w_rhp_worst=((Vref/I_o_max)*((1-D)^2))/L;
f_rhp=w_rhp/(2*pi*1e3);

%% Control-to-output TF Gvd
num_c=(Vin/((1-D)^2))*[-1/w_rhp 1];
den_c=[1/(w_o_ideal^2) 1/(Q_ideal*w_o_ideal) 1];
Gvd=tf(num_c,den_c);
Gvd_delay=tf(num_c,den_c,'InputDelay',t_total);
```

Now, we are going to design this boost converter whatever I have discussed the G vd all these transfer functions are derived.

(Refer Slide Time: 34:34)



Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Boost converter\Boost_SSM_VMC_PID_digital.m

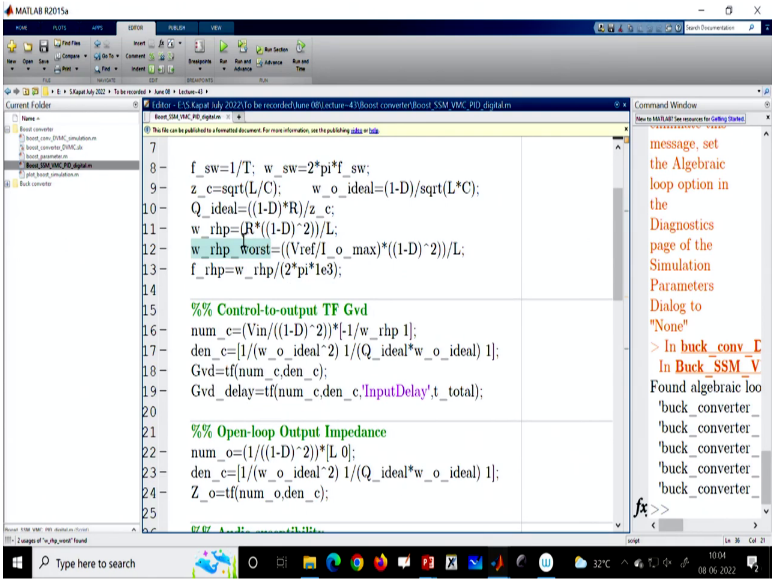
```
%% Modulator
V_m=5;  Fm=1/V_m;

%% PID controller design (analog)

K_ia=(V_m*w_rhp_worst*(1-D))/(10*Vin);
K_da=0.1*C;     tau_d=T/10;
K_pa=1;

num_con=[(K_da+(tau_d*K_pa)) (K_pa+(K_ia*tau_d)) K_ia];
den_con=[tau_d 1 0];
Gc=tf(num_con,den_con);

%% Loop gain and closed-loop TFs
G_loop=Gvd*Fm*Gc;    %% Loop gain
G_loop_delay=Gvd_delay*Fm*Gc;


Z_oc=Z_o/(1+G_loop);       %% Closed-loop output imp
```

Command Window

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

Here we are designing K i which is the integral gain in the analog domain I have just discussed. What is my rhp worst case?

(Refer Slide Time: 34:43)



Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Boost converter\Boost_SSM_VMC_PID_digital.m

```

f_sw=1/T;  w_sw=2*pi*f_sw;
z_c=sqrt(L/C);       w_o_ideal=(1-D)/sqrt(L*C);
Q_ideal=((1-D)*R)/z_c;
w_rhp=(R*((1-D)^2))/L;
w_rhp_worst=((Vref/I_o_max)*((1-D)^2))/L;
f_rhp=w_rhp/(2*pi*1e3);

%% Control-to-output TF Gvd
num_c=(Vin/((1-D)^2))*[-1/w_rhp 1];
den_c=[1/(w_o_ideal^2) 1/(Q_ideal*w_o_ideal) 1];
Gvd=tf(num_c,den_c);
Gvd_delay=tf(num_c,den_c,'InputDelay',t_total);

%% Open-loop Output Impedance
num_o=(1/((1-D)^2))*[L 0];
den_c=[1/(w_o_ideal^2) 1/(Q_ideal*w_o_ideal) 1];
Z_o=tf(num_o,den_c);

```

Command Window

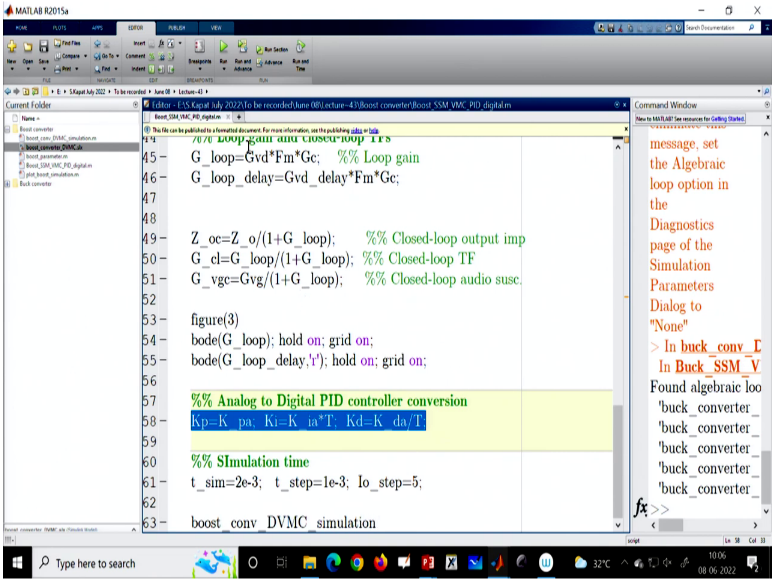
```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

So, the rhp 0 worst cases will come from that means we know that it is what is the rhp 0 worst case? So, let us go back and take. What is my rhp expression? It is r 1 minus D whole square by L. If you want to write in terms of load current. So, R is nothing but V 0 which is if we maintain the output voltage to V ref using integral. It is V ref by simply you know i 0, V ref by i 0 that is your R. That means, for worst case; that means, R has to be minimum which is nothing, but V ref divided by i 0 max. That means, we can write this one as nothing but V ref by L i 0 into 1 minus D whole square.

So, what is my rhp 0 worst case? So, it will be V ref 1 minus D whole square L i 0 minimum, that is the expression of the rhp 0 and that I have shown here in this code. That means, v ref i 0 max 1 by L ok. So, there should be 1 by d square yeah fine. Now we are designing. So, in this case, K d we have set 0.1 times C and tau d. So, I have set the proportional controller gain analog gain as just 1 very small gain, and let us check how does it look like.

That means, and this is the voltage mode controller of the boost converter and if you go inside again we are using a PID controller and finally, you have to get the digital PID controller gain.

(Refer Slide Time: 36:31)



Editor - E:\S.Kapat July 2022\To be recorded\June 08\Lecture-43\Boost converter\Boost_SSM_VMC_PID_digital.m

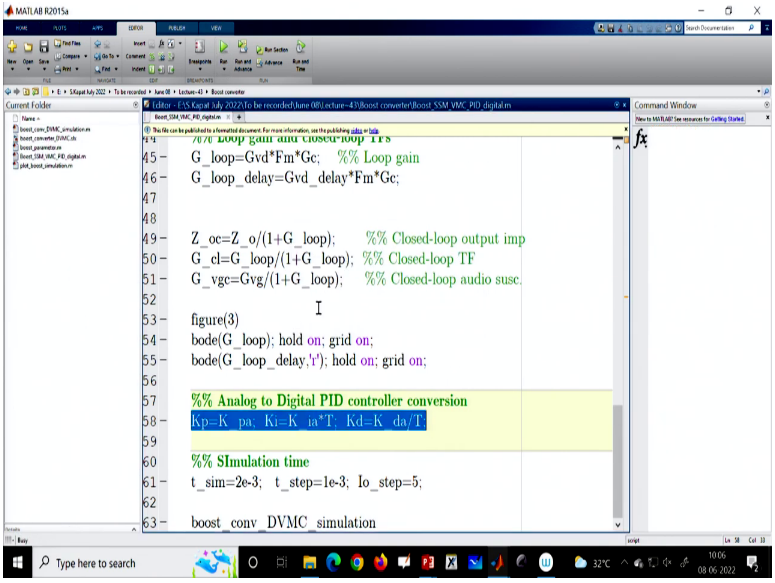
```
%% Loop gain and closed-loop TFs
G_loop=Gvd*Fm*Gc;    %% Loop gain
G_loop_delay=Gvd_delay*Fm*Gc;


Z_oc=Z_o/(1+G_loop);        %% Closed-loop output imp
G_cl=G_loop/(1+G_loop);  %% Closed-loop TF
G_vgc=Gvg/(1+G_loop);     %% Closed-loop audio susc.

figure(3)
bode(G_loop); hold on; grid on;
bode(G_loop_delay,'r'); hold on; grid on;

%% Analog to Digital PID controller conversion
Kp=K_pa;  Ki=K_ia*T;  Kd=K_da/T;

%% SImulation time
t_sim=2e-3;   t_step=1e-3;  Io_step=5;

boost_conv_DVMC_simulation
```

Command Window

```
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_conv_D
  In Buck_SSM_V
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
fx >>
```

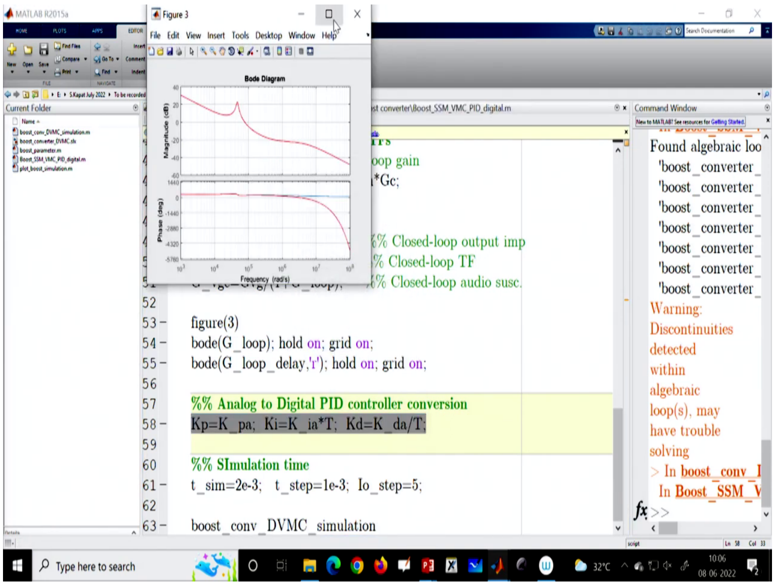
And whatever we obtain again will convert into analog to digital conversion and that will be plug-in into a Simulink model. So, let us run it and see what happened.

(Refer Slide Time: 36:40)



```
%% Loop gain and closed-loop TFs
G_loop=Gvd*Fm*Gc;   %% Loop gain
G_loop_delay=Gvd_delay*Fm*Gc;


Z_oc=Z_o/(1+G_loop);      %% Closed-loop output imp
G_cl=G_loop/(1+G_loop);  %% Closed-loop TF
G_vgc=Gvg/(1+G_loop);     %% Closed-loop audio susc.

figure(3)
bode(G_loop); hold on; grid on;
bode(G_loop_delay,'r'); hold on; grid on;

%% Analog to Digital PID controller conversion
Kp=K_pa;  Ki=K_ia*T;  Kd=K_da/T;

%% SImulation time
t_sim=2e-3;  t_step=1e-3;  Io_step=5;

boost_conv_DVMC_simulation
```

(Refer Slide Time: 36:41)

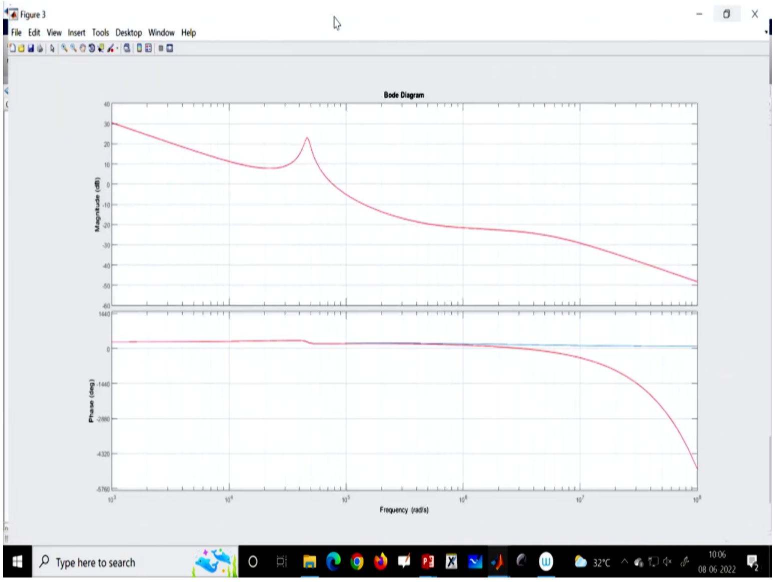


```
figure(3)
bode(G_loop); hold on; grid on;
bode(G_loop_delay,'r'); hold on; grid on;

%% Analog to Digital PID controller conversion
Kp=K_pa;  Ki=K_ia*T;  Kd=K_da/T;

%% SImulation time
t_sim=2e-3;  t_step=1e-3;  Io_step=5;

boost_conv_DVMC_simulation
```

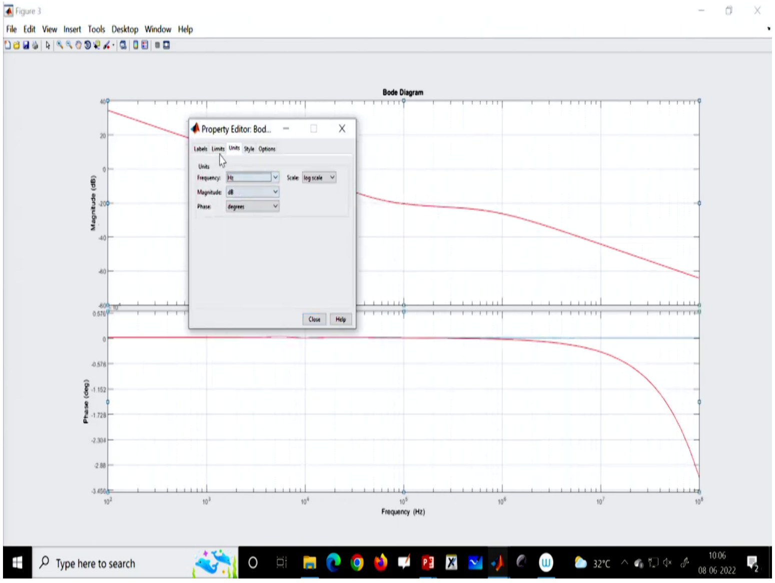
```
Found algebraic loo
  'boost_converter_
  'boost_converter_
  'boost_converter_
  'boost_converter_
  'boost_converter_
  'boost_converter_
  'boost_converter_
Warning:
Discontinuities
detected
within
algebraic
loop(s), may
have trouble
solving
> In boost_conv_I
  In Boost_SSM_V
fx >>
```

So, in this case, we are now running a boost converter.

(Refer Slide Time: 36:46)



(Refer Slide Time: 36:50)



Units

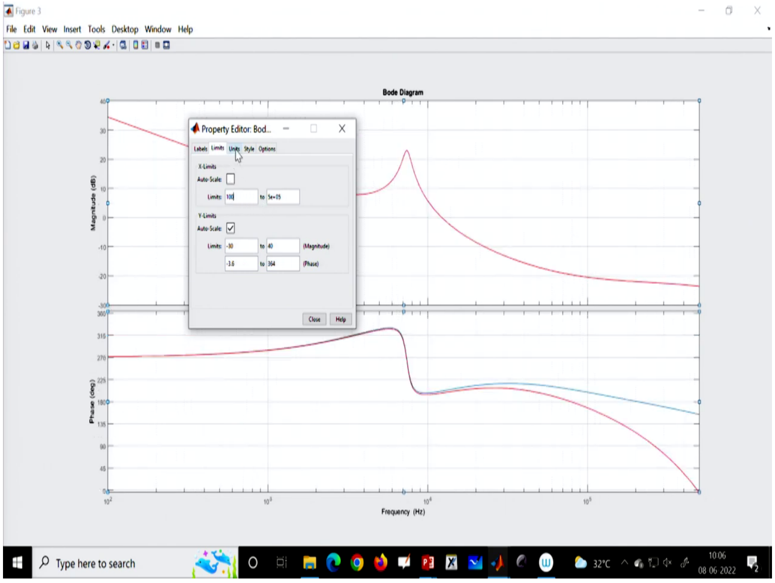
Frequency: Hz   Scale: log scale

Magnitude: dB

Phase: degrees

So, let us first consider the case study you know changing the time unit to hertz.

(Refer Slide Time: 36:52)



X-Limits

Auto-Scale: □

Limits: 100 to 5e+05

Y-Limits

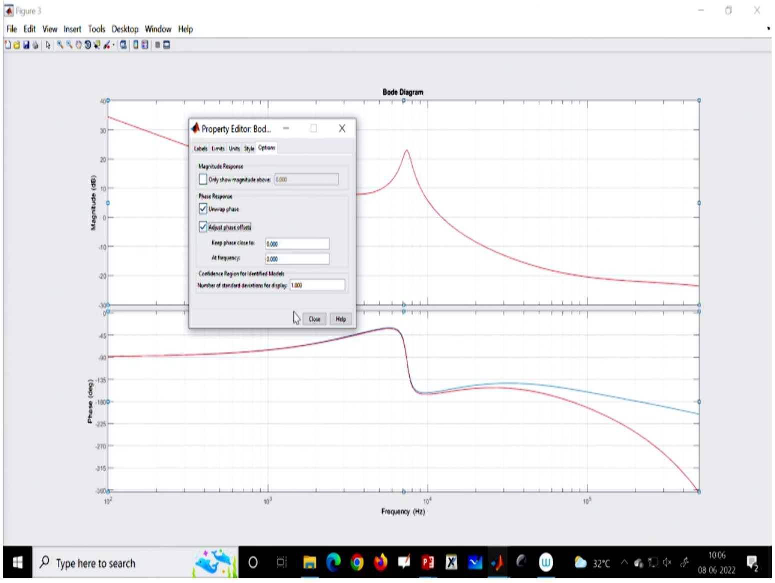
Auto-Scale: ☑

Limits: -30 to 40 (Magnitude)

-3.6 to 364 (Phase)

We want to limit to you know 0.5 e 6 which is the switching frequency.

(Refer Slide Time: 37:02)



Magnitude Response

□ Only show magnitude above: 0.000

Phase Response

☑ Unwrap phase

☑ Adjust phase offsets

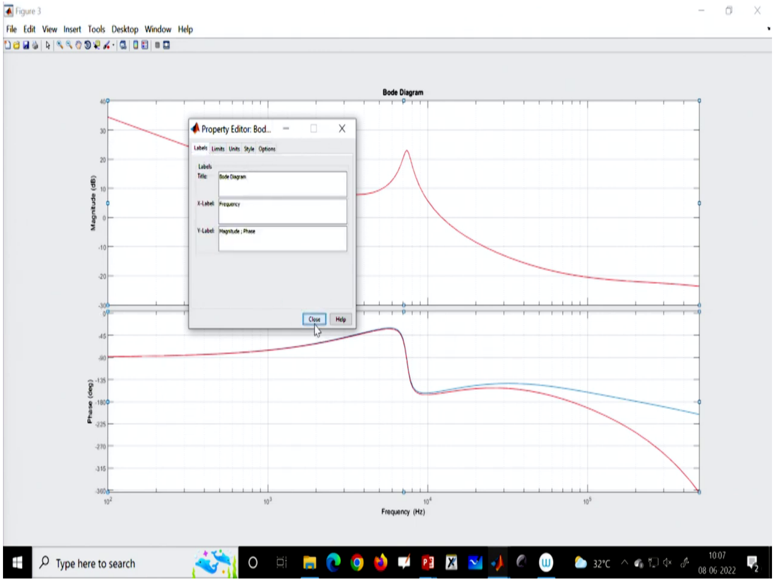
Keep phase close to: 0.000

At frequency: 0.000

Confidence Region for Identified Models

Number of standard deviations for display: 1.000

(Refer Slide Time: 37:05)



Labels

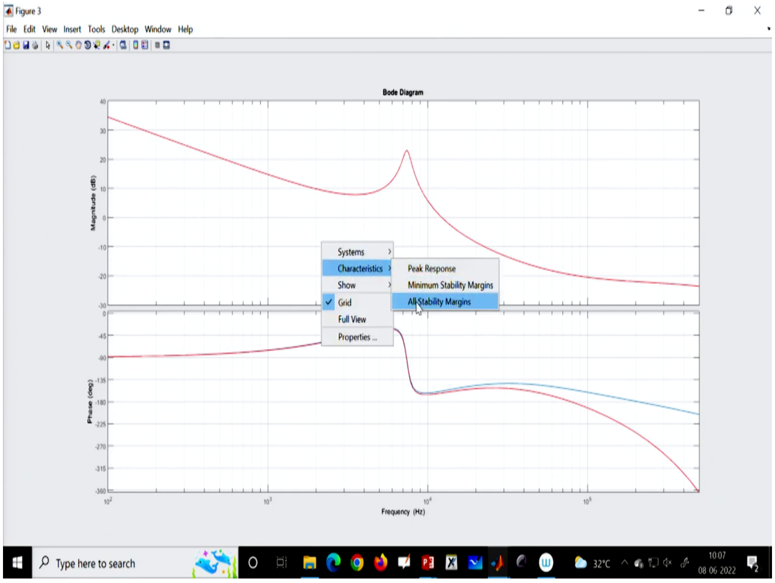
Title: Bode Diagram

X-Label: Frequency

Y-Label: Magnitude ; Phase

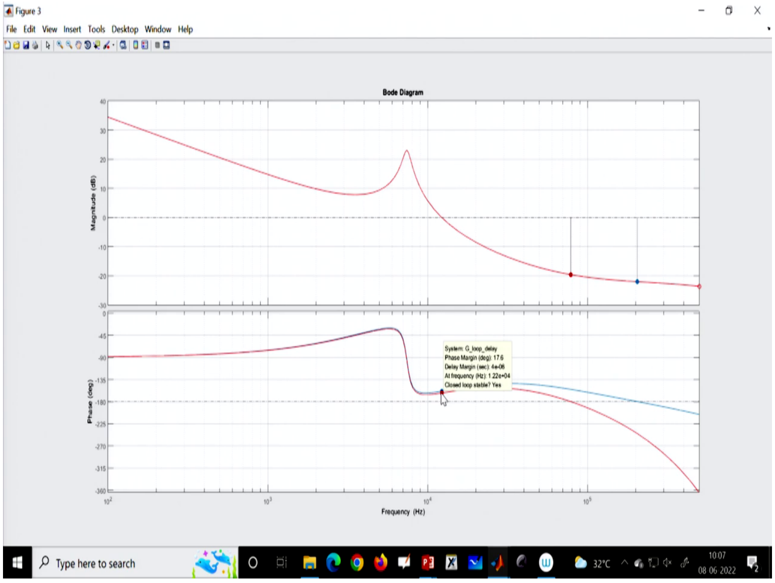
And we want to also wrap up the phase and now let us come back.

(Refer Slide Time: 37:09)



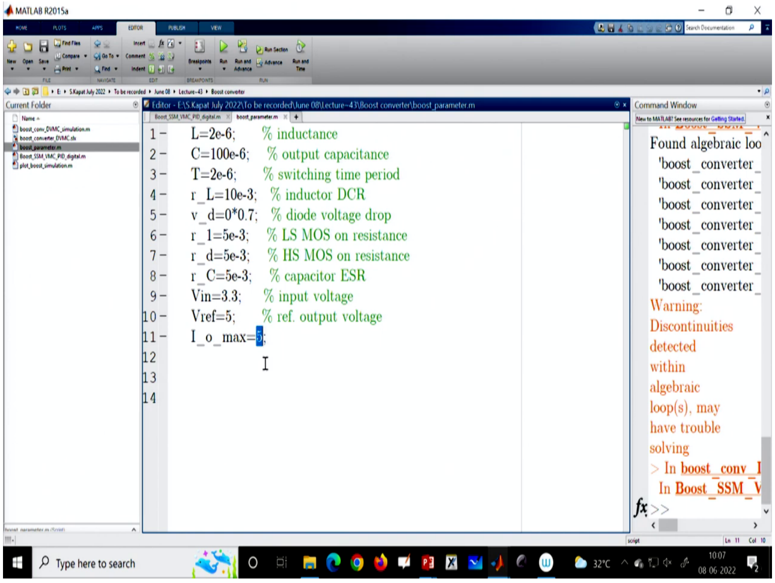
We want to see all the stability margins.

(Refer Slide Time: 37:11)



Now, you see the gain crossover frequency is coming to be slightly above. That means if we take here the gain crossover frequency is coming to be how much? Around 12.2 kilohertz, whereas, the switching frequency is 5 kilohertz. Because we are talking about the worst-case load current which is a 20 ampere load strip ok. So, at 20 ampere which is the maximum sorry I have to check not 20 what is mine in MATLAB? We have to check what is the worst case yeah.

(Refer Slide Time: 37:45)



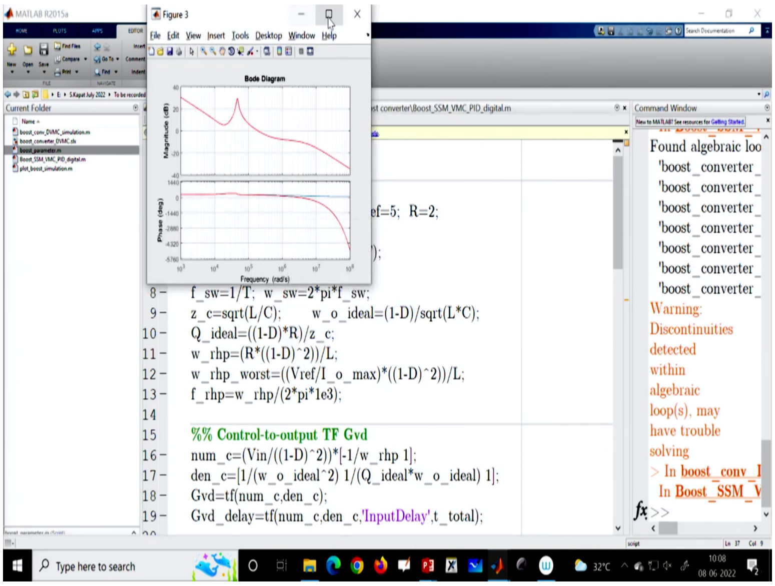
```
L=2e-6;       % inductance
C=100e-6;     % output capacitance
T=2e-6;       % switching time period
r_L=10e-3;    % inductor DCR
v_d=0*0.7;    % diode voltage drop
r_l=5e-3;     % LS MOS on resistance
r_d=5e-3;     % HS MOS on resistance
r_C=5e-3;     % capacitor ESR
Vin=3.3;      % input voltage
Vref=5;       % ref. output voltage
I_o_max=5;
```

```
Found algebraic loo
  'boost_converter_
  'boost_converter_
  'boost_converter_
  'boost_converter_
  'boost_converter_
  'boost_converter_
  'boost_converter_
Warning:
Discontinuities
detected
within
algebraic
loop(s), may
have trouble
solving
> In boost_conv_I
  In Boost_SSM_V
fx >>
```

So, worst case we have taken 5 ampere; that means, V ref is 5. So, 25 square, means 5 into 5; so we are talking about 25 watt you can always increase that with no issue. But I have taken 5 ampere as the maximum load current and that is why we are getting 12.2 as the crossover frequency, but here is the phase margin we are getting you to know is around 21.5 which is not very high.

So, you need to decrease you know I would say this gain to achieve 45 degree phase margin or we may have to increase the phase boost; that means, you know if we go back. If we go back I think we may have to increase this phase boost because the derivative will give some phase boost. So, let us again rerun this.

(Refer Slide Time: 38:41)



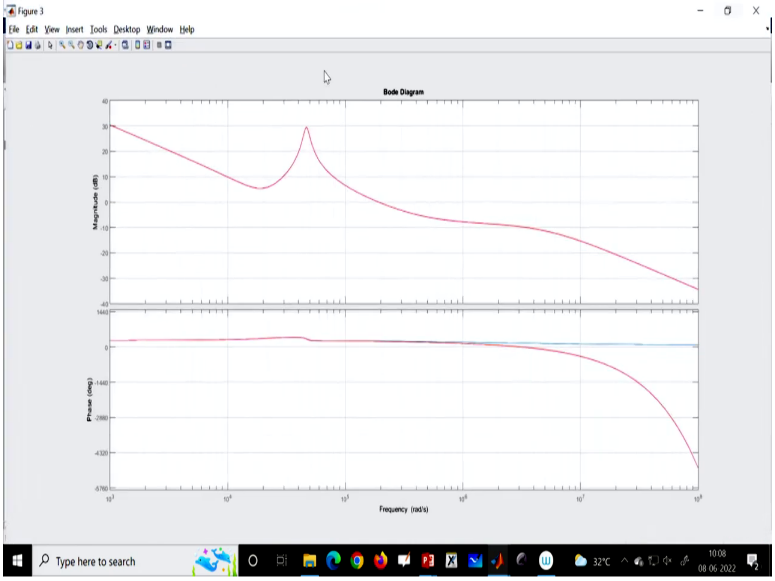
st converter\Boost_SSM_VMC_PID_digital.m

```
ef=5;  R=2;

);

f_sw=1/T;  w_sw=2*pi*f_sw;
z_c=sqrt(L/C);       w_o_ideal=(1-D)/sqrt(L*C);
Q_ideal=((1-D)*R)/z_c;
w_rhp=(R*((1-D)^2))/L;
w_rhp_worst=((Vref/I_o_max)*((1-D)^2))/L;
f_rhp=w_rhp/(2*pi*1e3);

%% Control-to-output TF Gvd
num_c=(Vin/((1-D)^2))*[-1/w_rhp 1];
den_c=[1/(w_o_ideal^2) 1/(Q_ideal*w_o_ideal) 1];
Gvd=tf(num_c,den_c);
Gvd_delay=tf(num_c,den_c,'InputDelay',t_total);
```

```
Found algebraic loo
  'boost_converter_
  'boost_converter_
  'boost_converter_
  'boost_converter_
  'boost_converter_
  'boost_converter_
  'boost_converter_
Warning:
Discontinuities
detected
within
algebraic
loop(s), may
have trouble
solving
> In boost_conv_I
  In Boost_SSM_V
fx >>
```

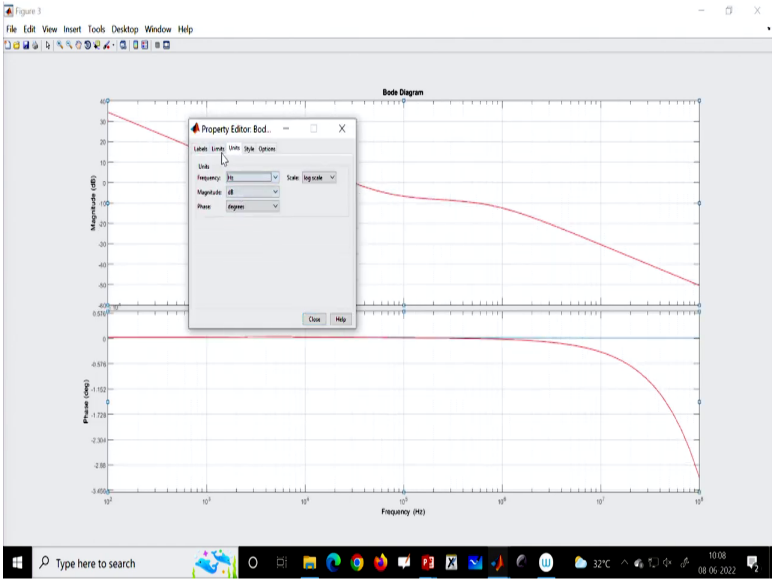
And let us check whether the derivative provides some additional phase boost or not.

(Refer Slide Time: 38:43)



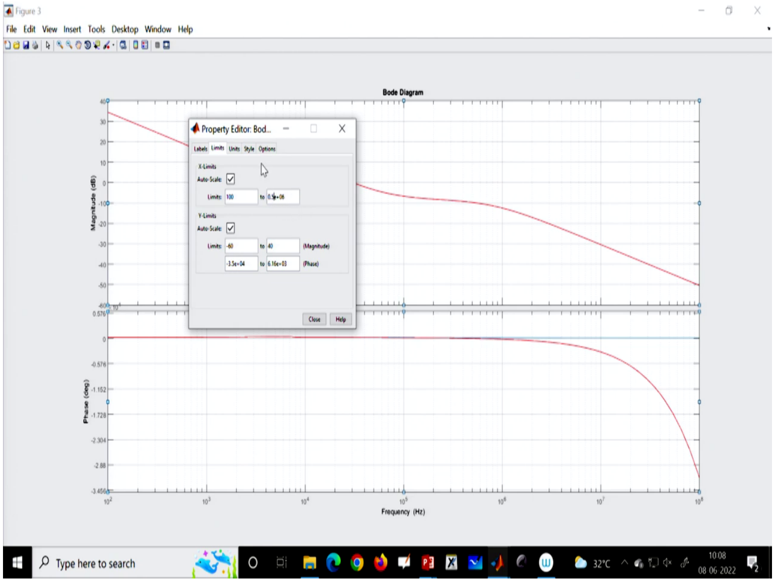
So, now again we are plotting it.

(Refer Slide Time: 38:47)



Units
Frequency: Hz
Scale: log scale
Magnitude: dB
Phase: degrees

(Refer Slide Time: 38:49)



Property Editor: Bod...

Labels | Limits | Units | Style | Options

X-Limits

Auto-Scale: ☑

Limits: 100 to 0.5e+06

Y-Limits

Auto-Scale: ☑

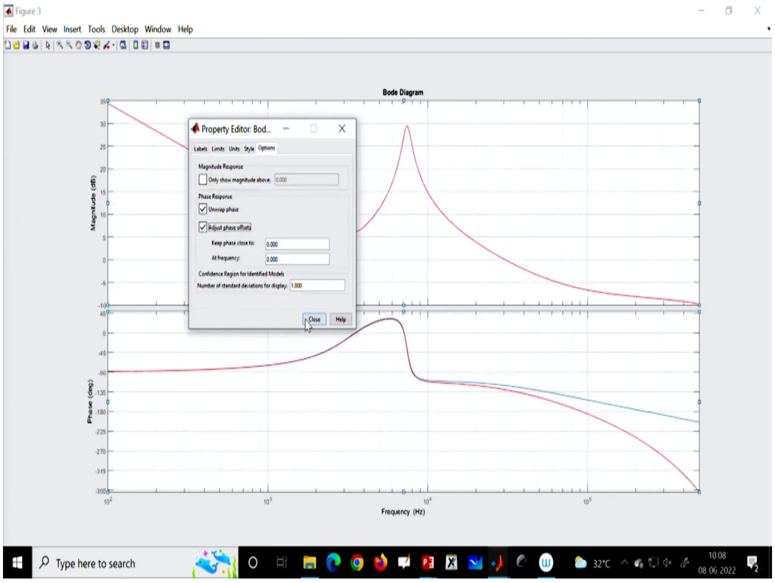
Limits: -60 to 40 (Magnitude)

-3.5e+04 to 6.16e+03 (Phase)

Close | Help

And unity gain and we are limiting to 0.6 to 0.5.

(Refer Slide Time: 38:57)



Property Editor: Bod...

Labels | Limits | Units | Style | Options

Magnitude Response

□ Only show magnitude above: 0.000

Phase Response

☑ Unwrap phase

☑ Adjust phase offsets

Keep phase close to: 0.000

At frequency: 0.000

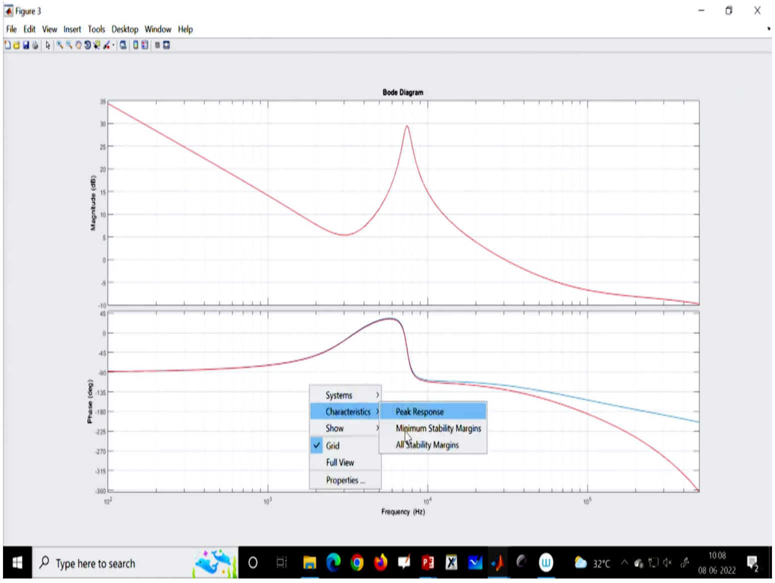
Confidence Region for Identified Models

Number of standard deviations for display: 1.000

Close | Help

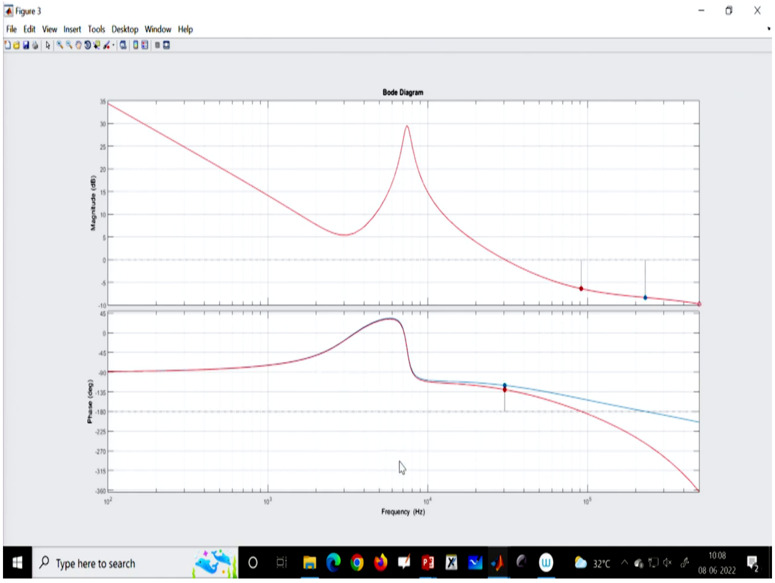
And we are making a phase adjustment.

(Refer Slide Time: 38:59)



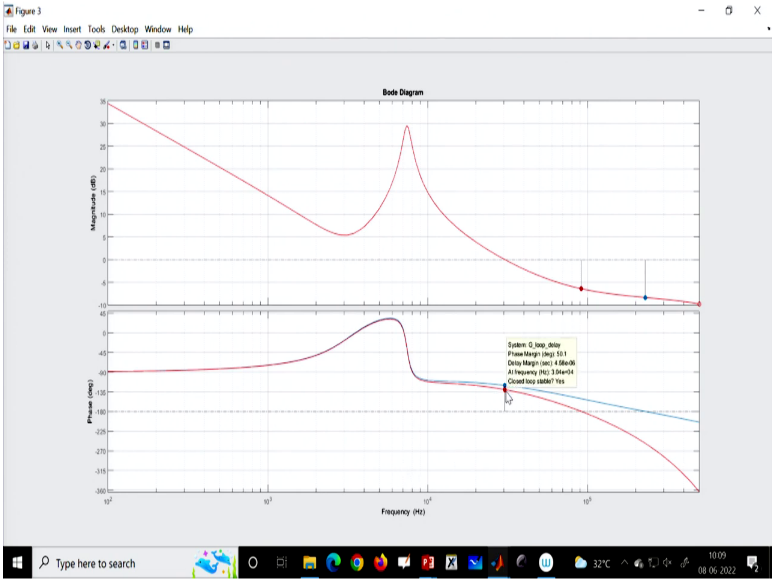
Yeah, and now we can get the phase characteristics.

(Refer Slide Time: 39:03)



So, now we have enough phase margin; that means, our derivative gain was smaller. So, we need to increase the derivative gain to get sufficient phase and here also we got a somewhat higher bandwidth.

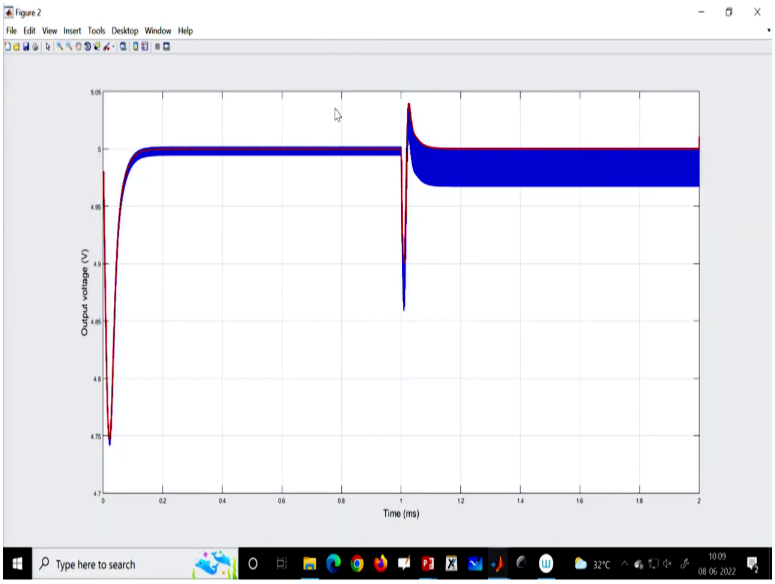
(Refer Slide Time: 39:13)



System: G_loop_delay
Phase Margin (deg): 50.1
Delay Margin (sec): 4.58e-06
At frequency (Hz): 3.04e+04
Closed loop stable? Yes

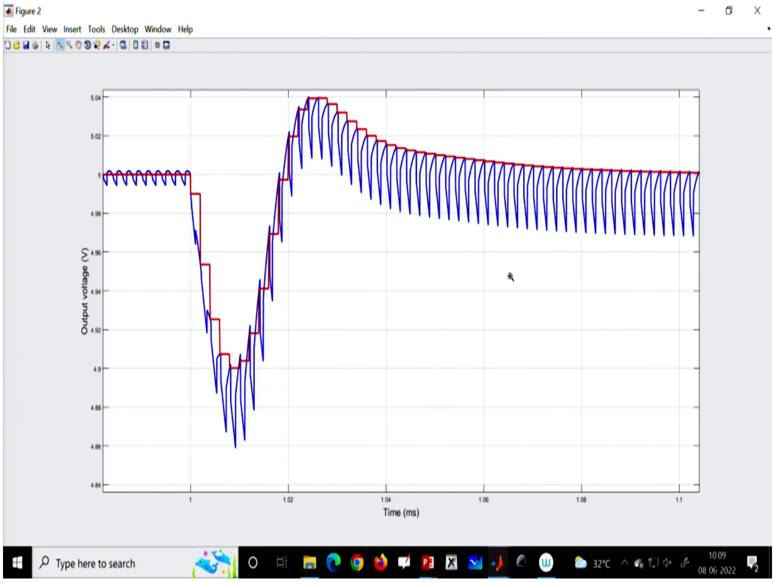
That means we got the bandwidth to be around 30 kilohertz compared to the earlier case and the phase margin is now 50 degree. So, you have a sufficient phase margin. Now you want to see the transient response.

(Refer Slide Time: 39:25)



So, if you see the transient response. So, this is a load transient response of the boost converter.

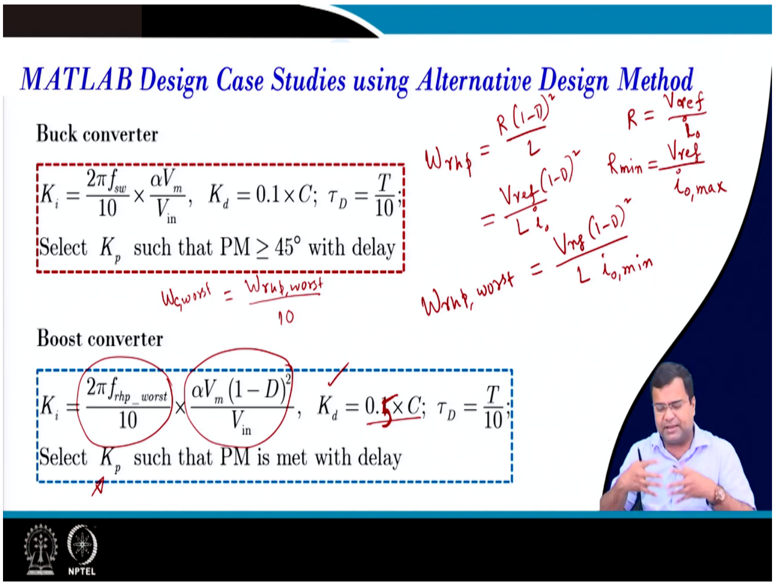
(Refer Slide Time: 39:31)



It is much better in the sense that you know we can get undershoot of around 140 millivolt and then the settling time is coming to be around 0.8 millisecond. That means 80 microsecond which means it is taking around 40 cycles because you know it has a right half plane 0. So, the bandwidth is limited, and here if you see the inductor current waveform you know this is the transient response.

So, this response is pretty nice. So, we can design the controller very effectively. That means if you want to get an additional phase I would say you can increase this derivative gain; that means, you can make the derivative gain somewhat higher.

(Refer Slide Time: 40:17)



MATLAB Design Case Studies using Alternative Design Method

Buck converter

$$K_i = \frac{2\pi f_{sw}}{10} \times \frac{\alpha V_m}{V_{in}}, \quad K_d = 0.1 \times C; \; \tau_D = \frac{T}{10};$$

Select K_p such that PM ≥ 45° with delay

$$\omega_{rhp} = \frac{R(1-D)^2}{L} = \frac{V_{ref}(1-D)^2}{L\, i_o} \qquad R = \frac{V_{ref}}{i_o} \qquad R_{min} = \frac{V_{ref}}{i_{o,max}}$$

$$\omega_{rhp,worst} = \frac{V_{ref}(1-D)^2}{L\, i_{o,min}}$$

$$\omega_{c,worst} = \frac{\omega_{rhp,worst}}{10}$$

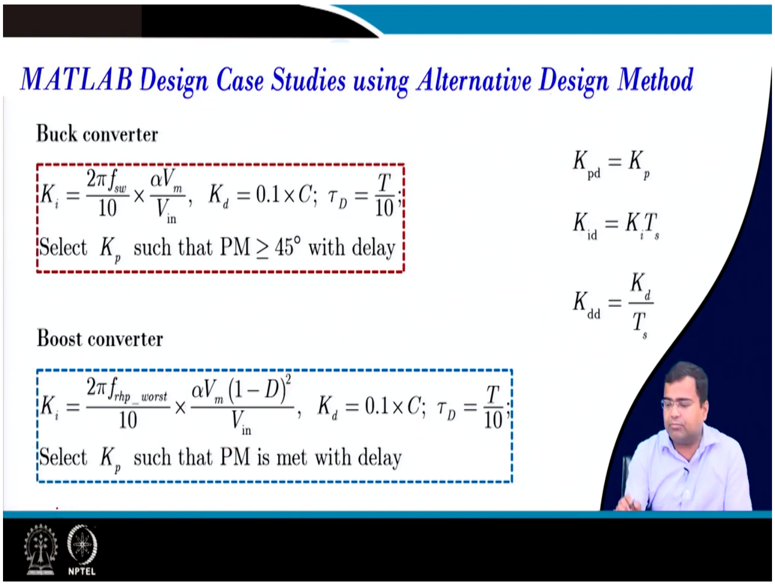
Boost converter

$$K_i = \frac{2\pi f_{rhp\_worst}}{10} \times \frac{\alpha V_m (1-D)^2}{V_{in}}, \quad K_d = 0.\overset{5}{\cancel{1}} \times C; \; \tau_D = \frac{T}{10};$$

Select K_p such that PM is met with delay

That means I can say for the boost converter you can set 0.5 instead of 0.1 you can set 0.5. So, we will get an additional phase boost, otherwise, the phase will be very very low.

(Refer Slide Time: 40:29)



MATLAB Design Case Studies using Alternative Design Method

Buck converter

$K_i = \frac{2\pi f_{sw}}{10} \times \frac{\alpha V_m}{V_{in}}, \quad K_d = 0.1 \times C; \; \tau_D = \frac{T}{10};$

Select K_p such that PM $\geq 45°$ with delay

Boost converter

$K_i = \frac{2\pi f_{rhp\_worst}}{10} \times \frac{\alpha V_m (1-D)^2}{V_{in}}, \quad K_d = 0.1 \times C; \; \tau_D = \frac{T}{10};$

Select K_p such that PM is met with delay

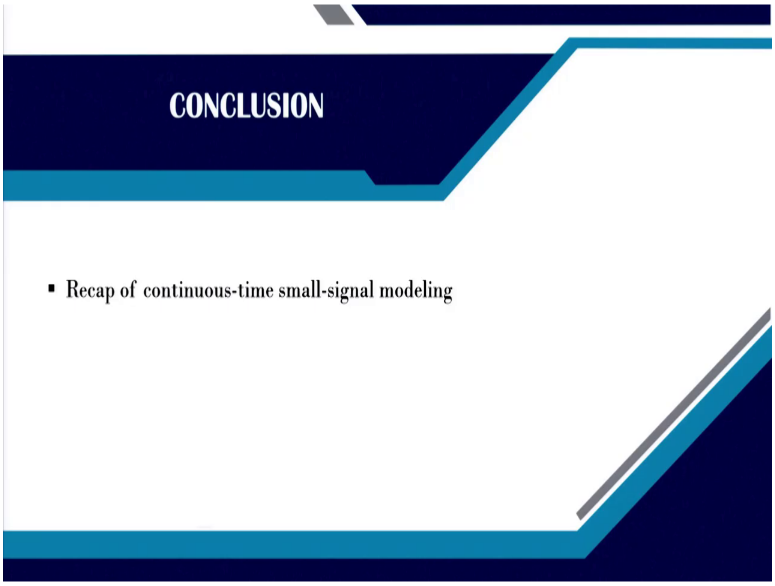
$K_{pd} = K_p$

$K_{id} = K_i T_s$

$K_{dd} = \frac{K_d}{T_s}$

So, now, we can convert all this into digital using this conversion.

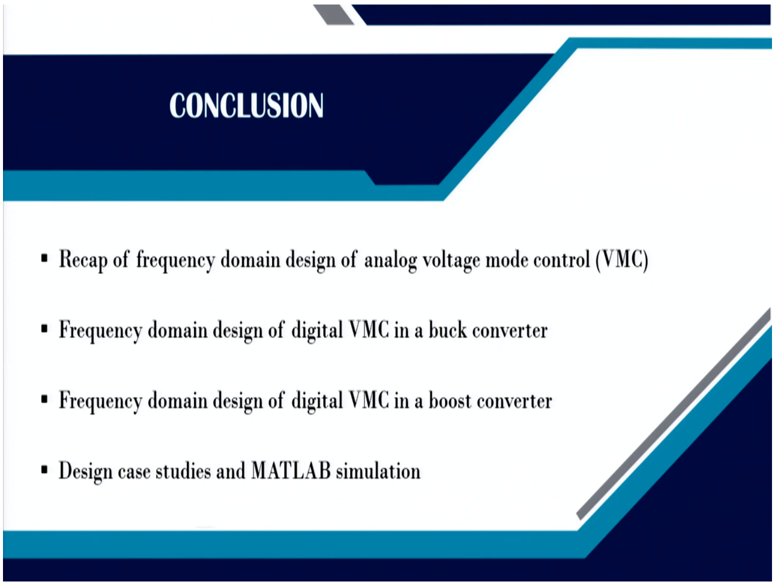
(Refer Slide Time: 40:31)



CONCLUSION

- Recap of continuous-time small-signal modeling

(Refer Slide Time: 40:32)



CONCLUSION

- Recap of frequency domain design of analog voltage mode control (VMC)
- Frequency domain design of digital VMC in a buck converter
- Frequency domain design of digital VMC in a boost converter
- Design case studies and MATLAB simulation

So, in summary, we have discussed we have recapitulated the frequency domain design of analog voltage mode control. We have discussed the frequency domain design of digital voltage mode control in a buck converter. We have also discussed the frequency domain design of digital voltage mode control in a boost converter and we have considered some design case studies and MATLAB simulation.

And I think this should be you know this design approach should be motivating enough. So, that you can design a voltage mode and digitally controlled buck and boost as well as the other converter. So, I have shown the approach by this frequency round approach we can still maintain the same continuous time approach and add the delay. But, in the subsequent lecture, I will be showing this continuous time model is nice when you talk about the small signal-based design, but as we want to push for a large signal design then this model cannot predict the different types of instability subharmonic instability.

Then we need to go for the actual discrete-time model and that will be taken in the subsequent lecture that is it for today.

Thank you very much.