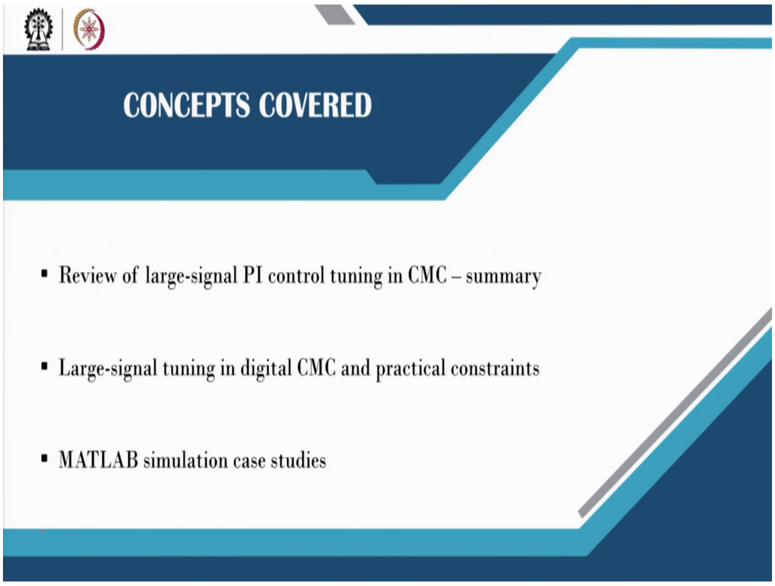
Digital Control in Switched Mode Power Converters and FPGA-based Prototyping
Prof. Santanu Kapat
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Module - 05
Frequency and Time Domain Digital Control Design Approaches
Lecture - 49
Trajectory-based Digital CMC Tuning and MATLAB Case Studies

Welcome back. So, in this lecture, we are going to talk about Trajectory based Digital Current Mode Control Tuning and MATLAB Case Study.

(Refer Slide Time: 00:31)



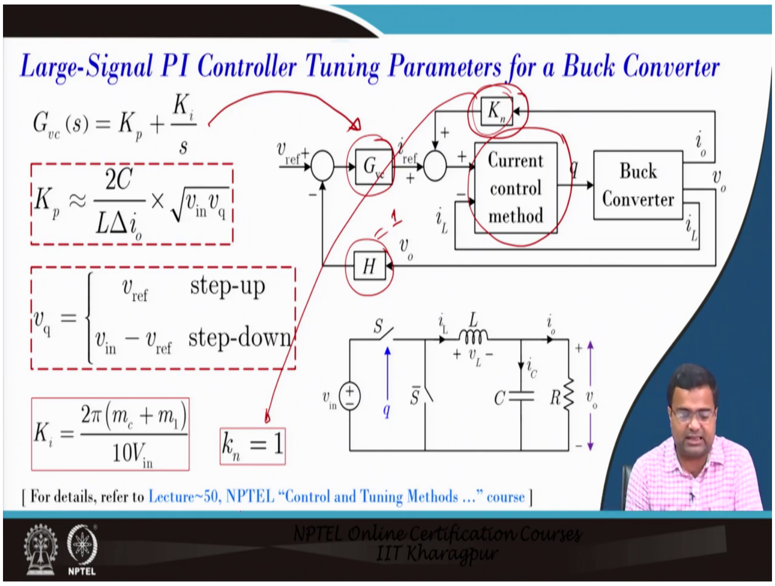
CONCEPTS COVERED

- Review of large-signal PI control tuning in CMC – summary
- Large-signal tuning in digital CMC and practical constraints
- MATLAB simulation case studies

So, this is the continuation of the previous lecture where we reviewed or we have discussed the large-signal PI controller tuning in current mode control. In this lecture, we will just summarize those results.

Then, we will talk about large-signal tuning in digital current mode control and practical constraints, and with MATLAB simulation case studies.

(Refer Slide Time: 00:54)



Large-Signal PI Controller Tuning Parameters for a Buck Converter

$$G_{vc}(s) = K_p + \frac{K_i}{s}$$

$$K_p \approx \frac{2C}{L\Delta i_o} \times \sqrt{v_{in} v_q}$$

$$v_q = \begin{cases} v_{ref} & \text{step-up} \\ v_{in} - v_{ref} & \text{step-down} \end{cases}$$

$$K_i = \frac{2\pi(m_c + m_1)}{10V_{in}}$$

$$k_n = 1$$

[For details, refer to Lecture~50, NPTEL "Control and Tuning Methods ..." course]

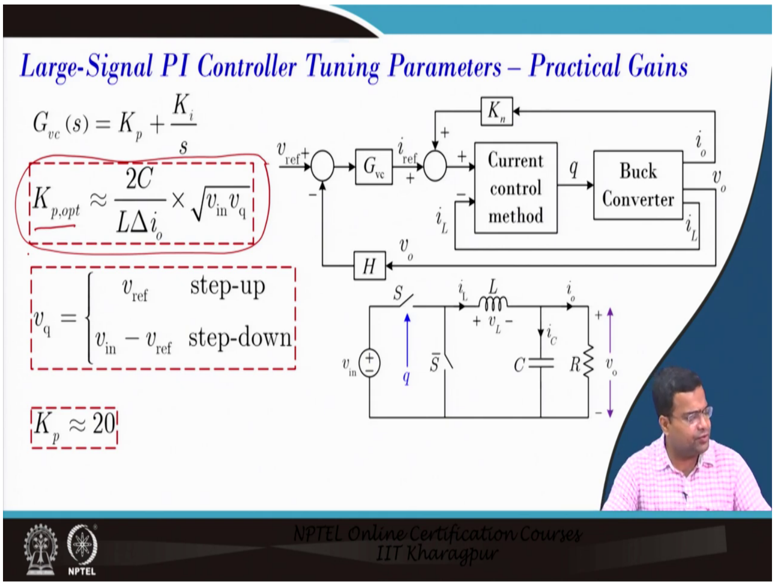
NPTEL Online Certification Courses
IIT Kharagpur

So, if we recollect our large-signal PI controller tuning in a buck converter. So, we have used this current mode control architecture, and then we have used a normalized load feed forward. And then what we did was that this gain is also that there is an outer voltage loop and here we have considered H equal to 1. So, for our case, we have considered, and the voltage loop as a PI controller in this case.

And then we discussed that K p can be calculated as 2 C by L delta i 0 square root of v into v q, where v q equals is to v ref for a step-up transient and v in minus v ref during the step-down transient. And the K i is nothing but 2 pi m c plus m 1 by 10 V in, where K n is a normalized gain, that is the feed-forward gain. So, this is this one, ok.

Now, we want to see what happens; we also discussed in the last class that you know this has been we have discussed this in lecture number 15 in our earlier course.

(Refer Slide Time: 02:07)



Large-Signal PI Controller Tuning Parameters – Practical Gains

$$G_{vc}(s) = K_p + \frac{K_i}{s}$$

$$K_{p,opt} \approx \frac{2C}{L\Delta i_o} \times \sqrt{v_{in} v_q}$$

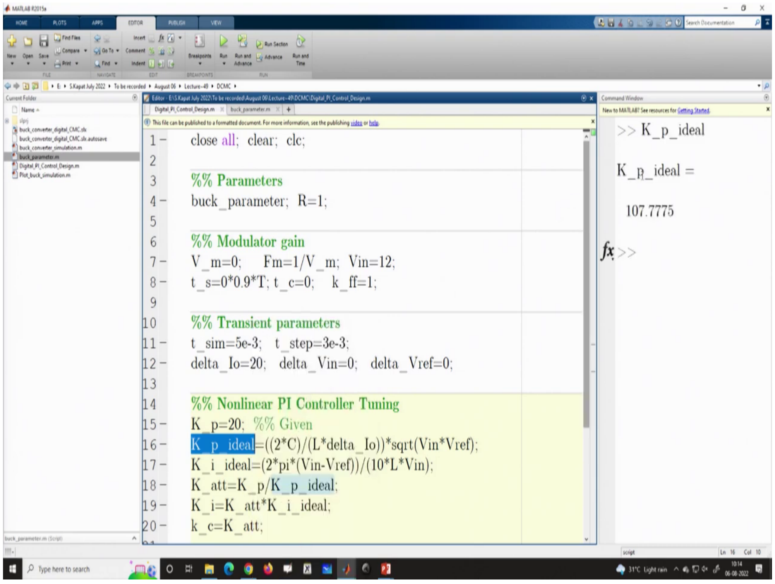
$$v_q = \begin{cases} v_{ref} & \text{step-up} \\ v_{in} - v_{ref} & \text{step-down} \end{cases}$$

$$K_p \approx 20$$

But now, we want to discuss if there is a limit in the controller gain which; means, we have calculated this optimal gain by analyanalyzing the calculation. But now if we want to limit this optimal gain, that means, K p equal 20, suppose that is my practical constant that we have discussed. Then how can we scale this other gain, so that we can get the near-optimal recovery?

Because we discussed in the last class that we are going to discuss using MATLAB case study. So, let us consider one scenario first. So, I just want to move to MATLAB.

(Refer Slide Time: 02:52)



Editor - E:\S.Kapat July 2022\To be recorded\August 06\Lecture-49\DCMC\Digital_PI_Control_Design.m

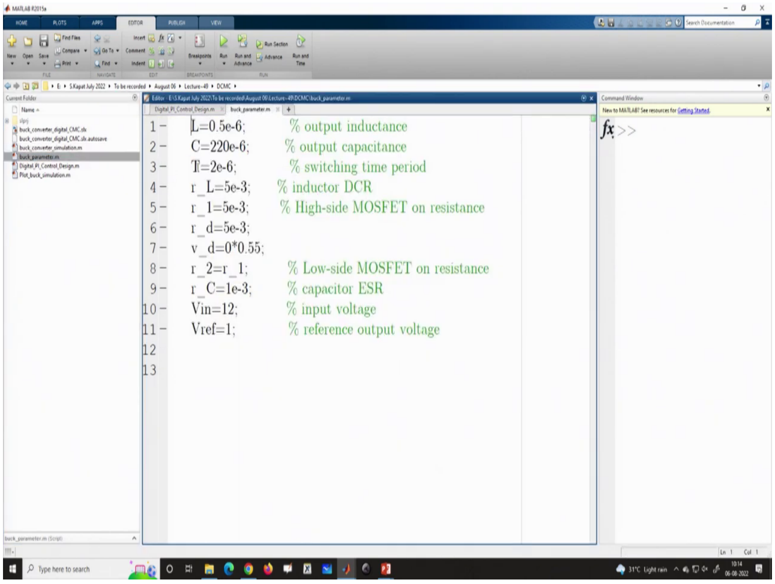
```
close all;  clear;  clc;

%% Parameters
buck_parameter;  R=1;

%% Modulator gain
V_m=0;     Fm=1/V_m;  Vin=12;
t_s=0*0.9*T; t_c=0;    k_ff=1;

%% Transient parameters
t_sim=5e-3;   t_step=3e-3;
delta_Io=20;   delta_Vin=0;   delta_Vref=0;

%% Nonlinear PI Controller Tuning
K_p=20;  %% Given
K_p_ideal=((2*C)/(L*delta_Io))*sqrt(Vin*Vref);
K_i_ideal=(2*pi*(Vin-Vref))/(10*L*Vin);
K_att=K_p/K_p_ideal;
K_i=K_att*K_i_ideal;
k_c=K_att;
```

Command Window

```
>> K_p_ideal

K_p_ideal =

  107.7775

fx >>
```

EDITOR
PUBLISH
VIEW
Digital_PI_Control_Design.m
buck_parameter.m
This file can be published to a formatted document. For more information, see the publishing video or help.
New to MATLAB? See resources for Getting Started.
Run Section
Run and Advance
Run and Time
Breakpoints
To be recorded ▸ August 06 ▸ Lecture-49 ▸ DCMC ▸
script
Ln 16 Col 10
31°C Light rain
10:14
06-08-2022

(Refer Slide Time: 02:58)

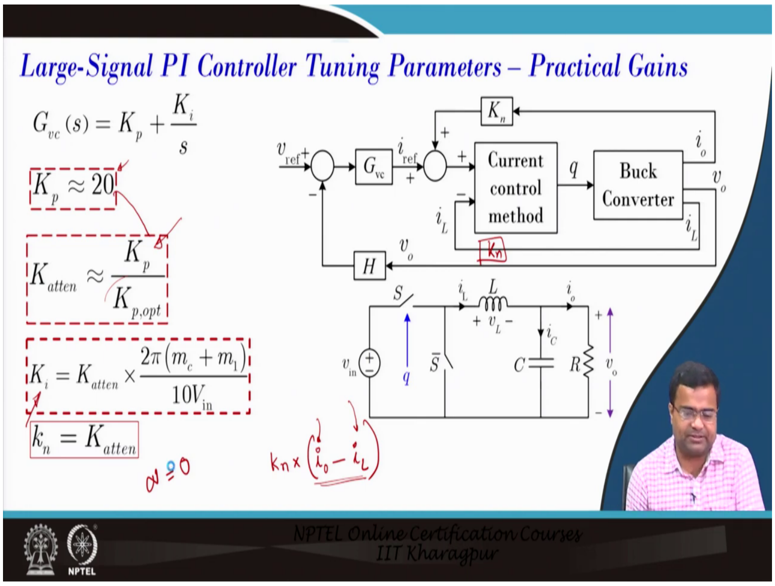


```
L=0.5e-6;       % output inductance
C=220e-6;       % output capacitance
T=2e-6;         % switching time period
r_L=5e-3;       % inductor DCR
r_1=5e-3;       % High-side MOSFET on resistance
r_d=5e-3;
v_d=0*0.55;
r_2=r_1;        % Low-side MOSFET on resistance
r_C=1e-3;       % capacitor ESR
Vin=12;         % input voltage
Vref=1;         % reference output voltage
```

And, let us say for 12-volt input I am talking about a buck converter where the inductor is microhenryenry, the capacitor is 220 microfarad and we are talking about 500 kilohertz and the input voltage is 12 volts and the output reference voltage is 1 volt.

Now, if we apply a 20-ampere load step then the K p ideal gain can be calculated, if you evaluate it is 107.7775 which is pretty large. But we only have 20 is the upper limit. Suppose you want to constrain this gain into 20, then what to do? We have discussed that you know.

(Refer Slide Time: 03:39)



Large-Signal PI Controller Tuning Parameters – Practical Gains

$$G_{vc}(s) = K_p + \frac{K_i}{s}$$

$$\boxed{K_p \approx 20}$$

$$\boxed{K_{atten} \approx \frac{K_p}{K_{p,opt}}}$$

$$\boxed{K_i = K_{atten} \times \frac{2\pi(m_c + m_1)}{10V_{in}}}$$

$$\boxed{k_n = K_{atten}}$$

$\alpha \geq 0$

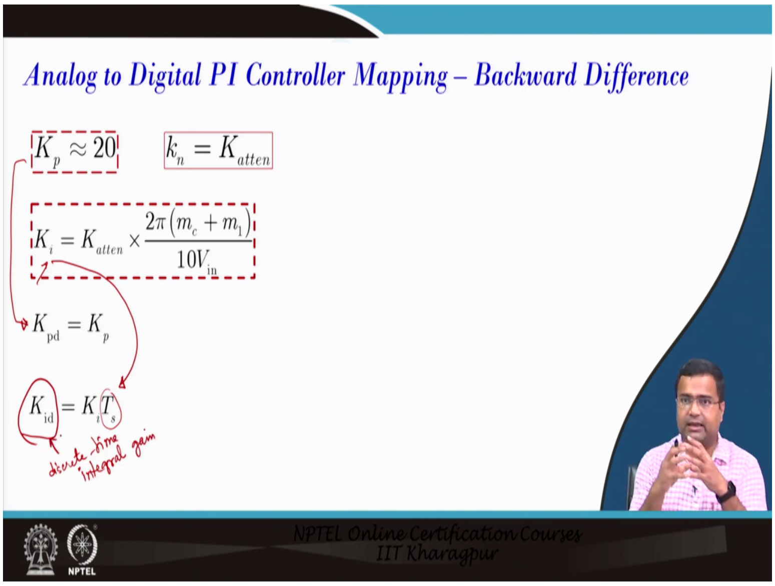
$k_n \times (i_o - i_L)$

So, if K p is a 20 is the upper limit, then what we will do? By taking K p is 20 we have to find out the attenuation factor. So, we have to attenuate K p, which is given 20. And what is the optimal gain that we have computed?

We have to take the ratio, and then we will get the attenuation factor. Then, the integral gain has to be attenuated by the same quantity and also this K n. Where is K n? This is K n and here also we are talking about K n. So, here also we are talking about K n; which means, the current loop gain; that means, this K n will come as K n into i 0 minus i L. So, that means, if we normalize if we reduce the gain we have to reduce both the gain of the i 0 as well as the inductor current.

So, both the inductor current and the load current will be on the same scale because it is the buck converter. The average inductor current is equal to the average load current. So, we have to find out. And we have discussed that this is because, under a sigma equal to 0, we can always normalize the switching surface.

(Refer Slide Time: 04:41)



Analog to Digital PI Controller Mapping – Backward Difference

$\boxed{K_p \approx 20}$ $\boxed{k_n = K_{atten}}$

$$\boxed{K_i = K_{atten} \times \frac{2\pi(m_c + m_1)}{10V_{in}}}$$

$K_{pd} = K_p$

$K_{id} = K_i T_s$

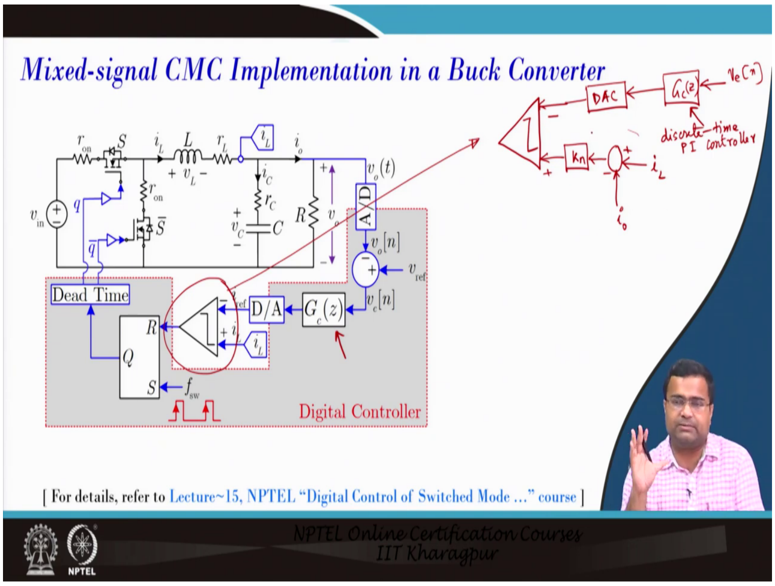
discrete-time integral gain

NPTEL Online Certification Courses
IIT Kharagpur

Now, once we get derived these normalized gains K n, K attenuation I mean in K integral gain, then we have to convert this into discrete time.

So, the proportional gain will remain the same and the integral gain will be multiplied by the time sampling time to get the discrete-time integral gain. So, this is my discrete-time integral gain. So, this is my discrete-time integral gain, which is the continuous-time integral gain into sampling time.

(Refer Slide Time: 05:23)



Mixed-signal CMC Implementation in a Buck Converter

[For details, refer to Lecture~15, NPTEL "Digital Control of Switched Mode ..." course]

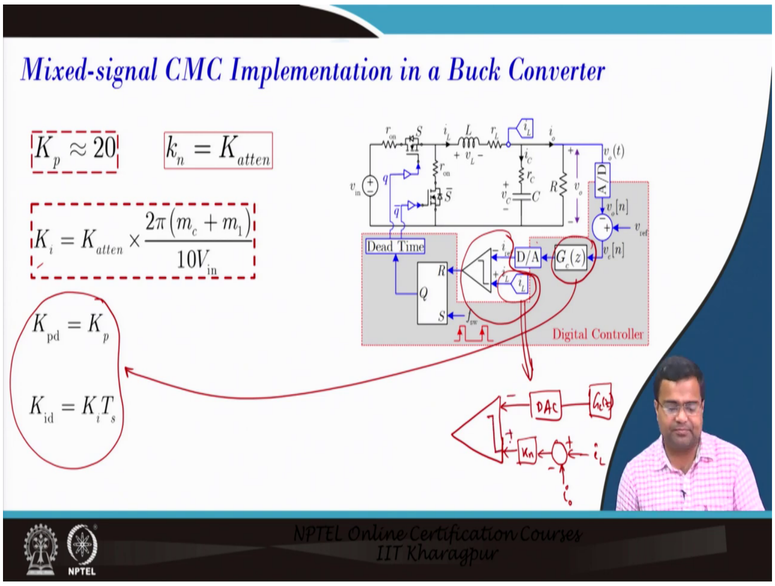
Now, we are talking about the mixed signal current mode control where we want to implement our large-signal tuning in the digital loop. That means, our controller will be sitting in the digital controller here. And we have discussed this in lecture number 15. But there will be a slight modification as per our requirement here.

If we take this out here, that means, we have this comparator, this comparator negative terminal, and this comparator positive terminal. So, the negative terminal is coming out from the DAC, it is coming out, DAC. And what is the DAC input? It is our controller G c z and the input is our error voltage in the discrete domain.

And what is this controller? We have discussed this as like a discrete time PI controller, ok. So, it is going out of the screen. So, let us write once more. So, it is a discrete-time PI controller. And as per all implementation, what we are going to do? We are considering one normalized gain K n, then this is our inductor current, which is added and this is our load current which is subtracted. So, this is the modified representation.

And we have discussed that if we scale down the proportional gain because we need to talk about it, we need to consider the realistic or the practical constraint. Then, we need to attenuate also this feedback loop and this K n will be replaced by the attenuation factor, that we have discussed in the previous slide.

(Refer Slide Time: 07:30)



Mixed-signal CMC Implementation in a Buck Converter

$\boxed{K_p \approx 20}$ $\boxed{k_n = K_{atten}}$

$$\boxed{K_i = K_{atten} \times \frac{2\pi(m_c + m_1)}{10V_{in}}}$$

$K_{pd} = K_p$

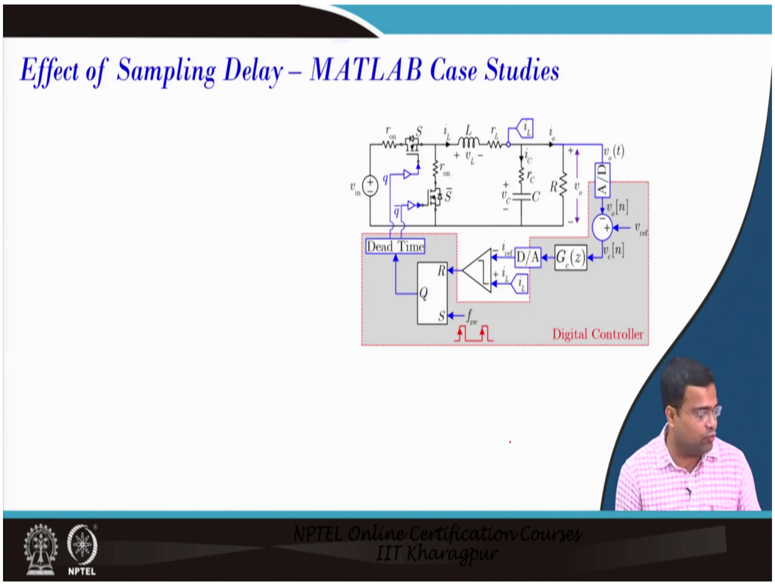
$K_{id} = K_i T_s$

NPTEL Online Certification Courses
IIT Kharagpur

So, that means, in the mixed signal current mode control, we have plugged in this digital controller K p equal to 20. K attenuation factor we have discussed this for this there is a modification in this loop, and the integral gain it is in the continuous domain, and the discrete domain this controller gain will be set as this.

And this part we will have to separately draw as if this will be your DAC, G c z, and this part will be K n will be there, this will be our I L, and this will be our i 0, ok. So, this is just the modification that we have to make.

(Refer Slide Time: 08:31)



Effect of Sampling Delay – MATLAB Case Studies

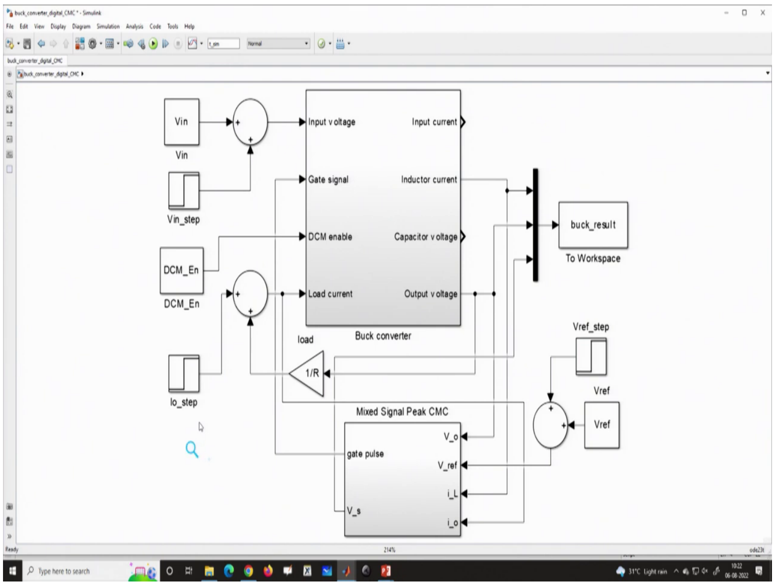
NPTEL Online Certification Courses
IIT Kharagpur

So, now we want to simulate this, ok. And we want to see what is the effect of sampling and how does it look like in real MATLAB implementation.

So, in this case, if you know we are talking about it again the same because we have discussed this MATLAB in like week 3 in detail where first we will clear the screen. We will take the buck converter parameter L equal to 0.5, all these things are given, 0.5 micro henry, C equal to 220 microfarad. We are initially talking about 12 inputs, but we can change it, reference voltage we set it as 1 volt.

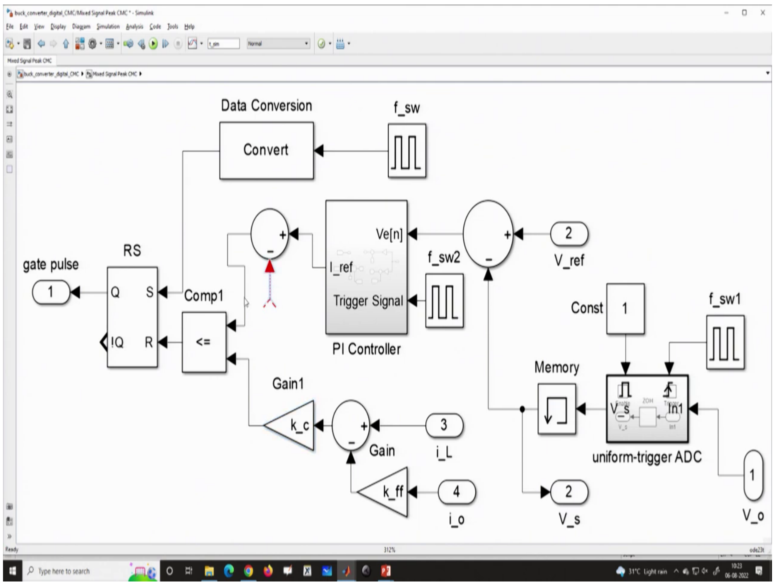
So, first, we will load this buck converter parameter which I have just discussed here.

(Refer Slide Time: 09:22)



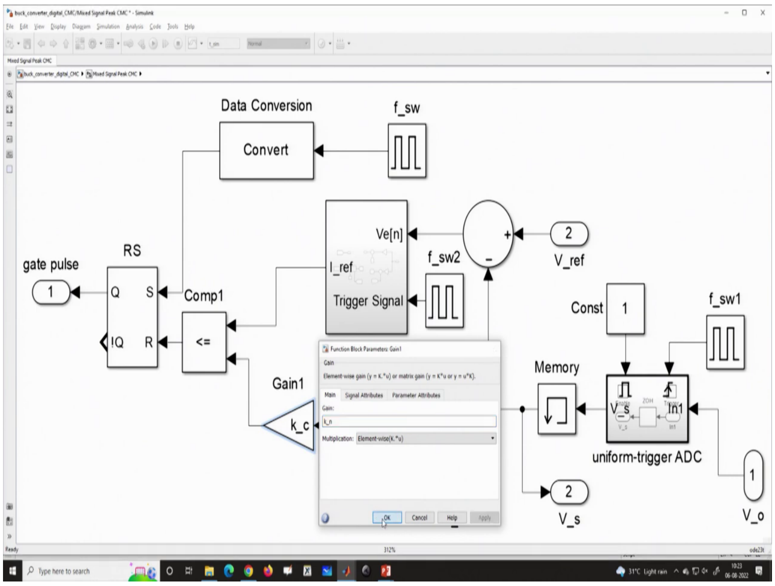
Then we will set load resistance equal to 1 because if you go to the Simulink block, we have a continuous load resistance here and then we are applying a load step, current load step, and of 20 ampere, ok.

(Refer Slide Time: 09:40)



So, that means, they are continuous load. We are not adding any ramp, but if you go here there is a provision for the ramp. You can subtract ramp add with this or subtract with this. So, ramp we are not considering it. So, we can just drop it so, this block can be directly connected here.

(Refer Slide Time: 09:52)



Function Block Parameters: Gain1

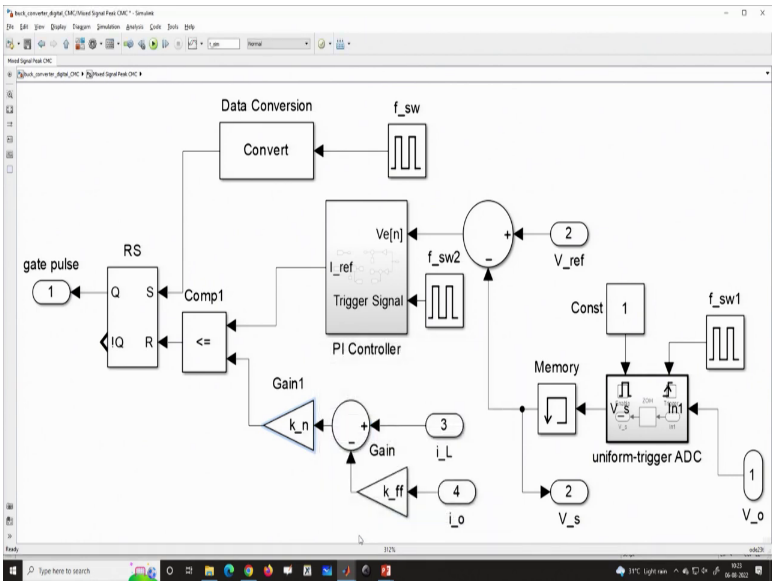
Gain

Element-wise gain (y = K.*u) or matrix gain (y = K*u or y = u*K).

Gain: k_n

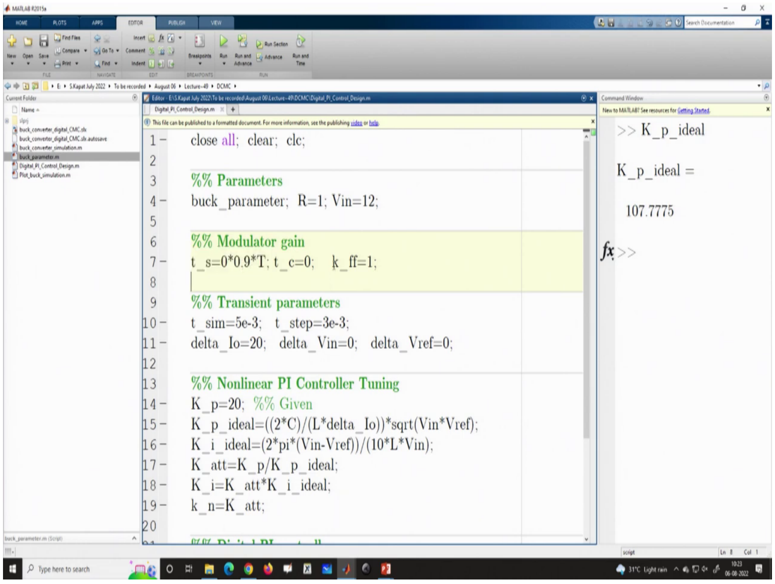
Multiplication: Element-wise(K.*u)

(Refer Slide Time: 10:04)



Now here is the K c; that means, the current loop gain which is the K n we say; we can also use K n here instead of you know avoid any confusion in the notation. And we have to accordingly say this is my K n.

(Refer Slide Time: 10:08)



```
close all;  clear;  clc;

%% Parameters
buck_parameter;  R=1; Vin=12;

%% Modulator gain
t_s=0*0.9*T; t_c=0;    k_ff=1;

%% Transient parameters
t_sim=5e-3;   t_step=3e-3;
delta_Io=20;   delta_Vin=0;   delta_Vref=0;

%% Nonlinear PI Controller Tuning
K_p=20;  %% Given
K_p_ideal=((2*C)/(L*delta_Io))*sqrt(Vin*Vref);
K_i_ideal=(2*pi*(Vin-Vref))/(10*L*Vin);
K_att=K_p/K_p_ideal;
K_i=K_att*K_i_ideal;
k_n=K_att;
```

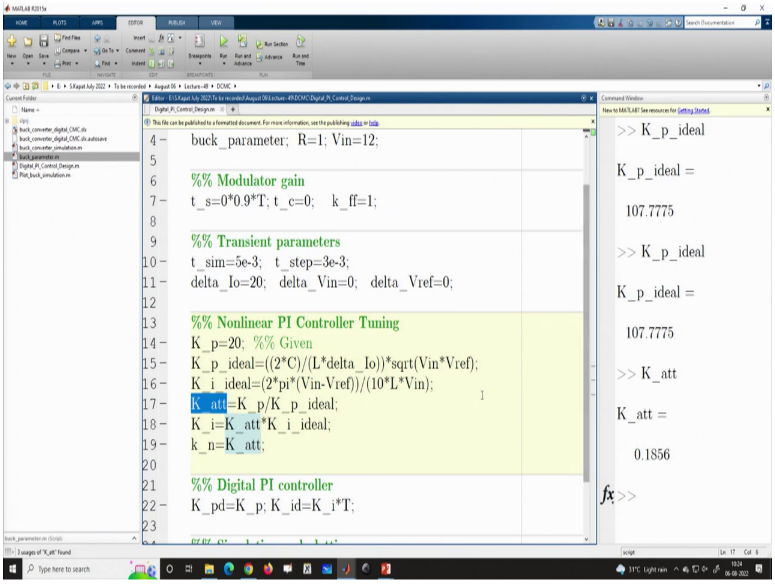
```
>> K_p_ideal

K_p_ideal =

  107.7775
```

So, there is no modulator gain. So, you can there is no point here. So, this part can be taken out. Input voltage we can change here, whatever we want we can change, whether we want 12 volt whatever we can do.

Sampling delay for the time being we are not considering because if we consider sampling delay then we will see. Now, we have a load current feed-forward k ff. If you go here there is a k ff, there is a load current feed-forward term. If it is 1, there is a load current feed-forward, and it is on the same scale as the inductor current.

We are simulating 5 millisecond, at 3 millisecond we are applying a load step of 20 ampere. Now, our practical gain is given as 20 that is given. Suppose that is the upper limit. But the ideal gain is computed and we have discussed earlier that the ideal that in if we calculate it will be too large; that means, we need to create an attenuation factor which will be 20 by 107.77. So, it will come to around 0.18.

(Refer Slide Time: 11:18)



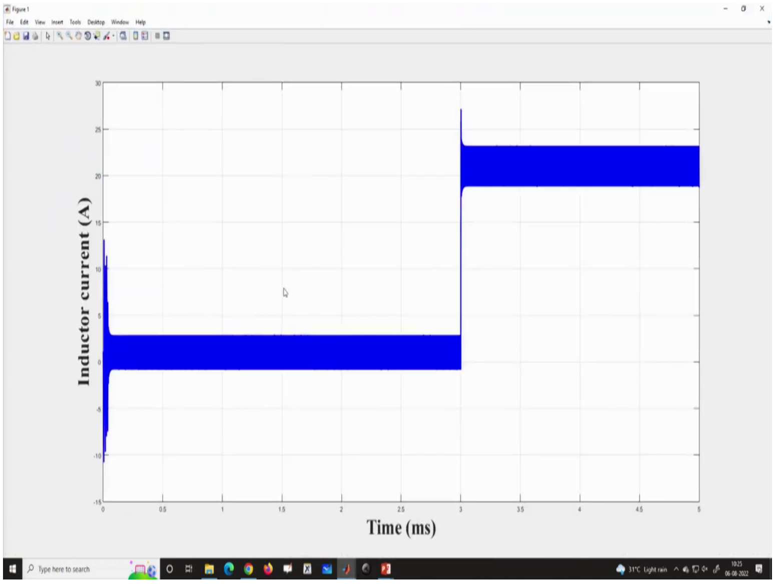
```
buck_parameter;  R=1; Vin=12;

%% Modulator gain
t_s=0*0.9*T; t_c=0;    k_ff=1;

%% Transient parameters
t_sim=5e-3;   t_step=3e-3;
delta_Io=20;   delta_Vin=0;   delta_Vref=0;

%% Nonlinear PI Controller Tuning
K_p=20;  %% Given
K_p_ideal=((2*C)/(L*delta_Io))*sqrt(Vin*Vref);
K_i_ideal=(2*pi*(Vin-Vref))/(10*L*Vin);
K_att=K_p/K_p_ideal;
K_i=K_att*K_i_ideal;
k_n=K_att;

%% Digital PI controller
K_pd=K_p; K_id=K_i*T;
```

```
>> K_p_ideal

K_p_ideal =

  107.7775

>> K_p_ideal

K_p_ideal =

  107.7775

>> K_att

K_att =

   0.1856

fx >>
```

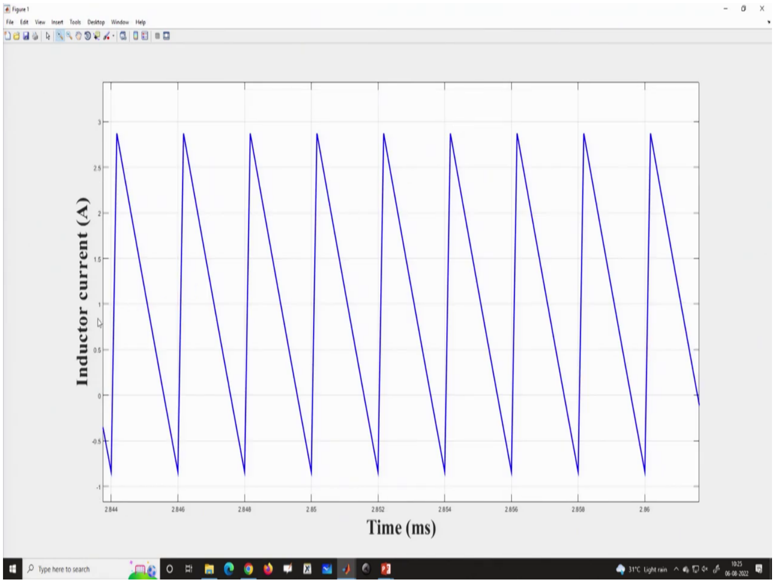
Then, what do we have to do? The actual integral gain will be the attenuation factor into the gain which is computed by our tuning formula. And the ideal proportional gain is also calculated using our tuning formula. And K n which is the normalized gain for the current loop consisting of inductor current minus load current, that is K n.

After all this, we have converted and the current loop is analog. So, proportional gain in discrete time is the same as the continuous-time proportional gain. And the discrete-time integral gain is nothing but the continuous-time integral gain into the sampling time which is the same as the switching period because we are taking 1 sample per cycle. Now, we want to run the simulation and see the effect.

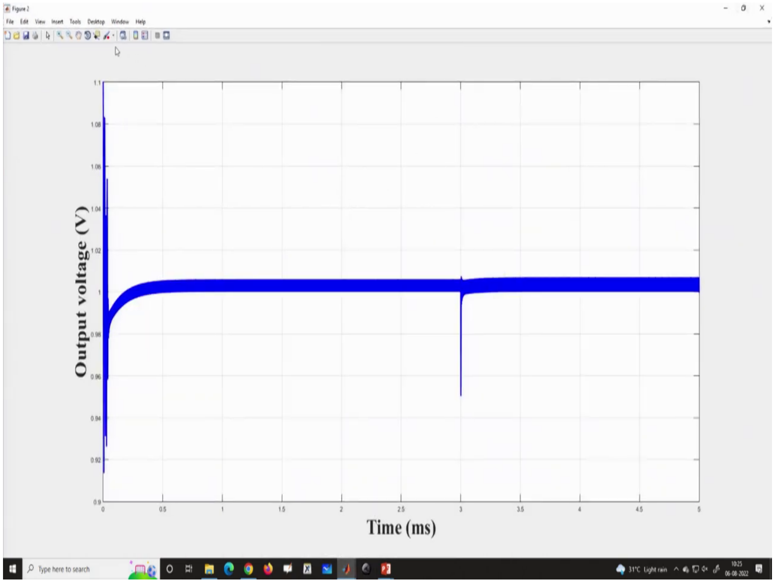
(Refer Slide Time: 12:20)



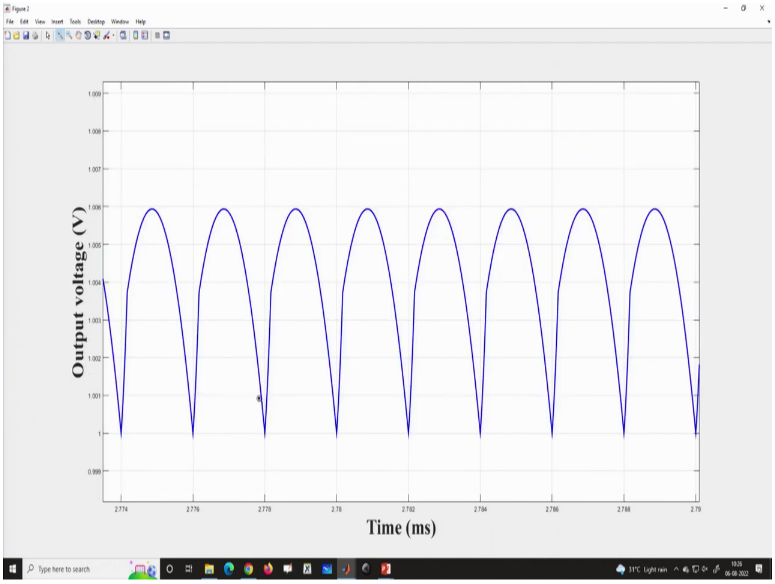
(Refer Slide Time: 12:28)



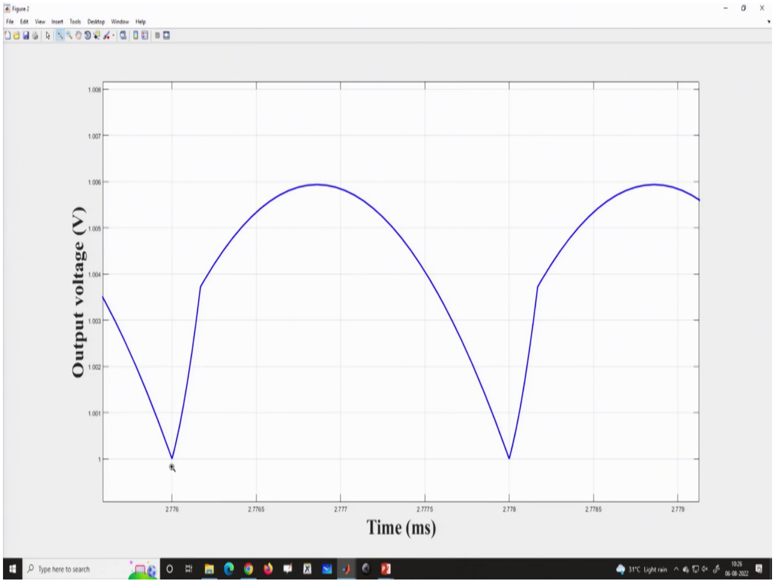
(Refer Slide Time: 12:37)



(Refer Slide Time: 12:40)



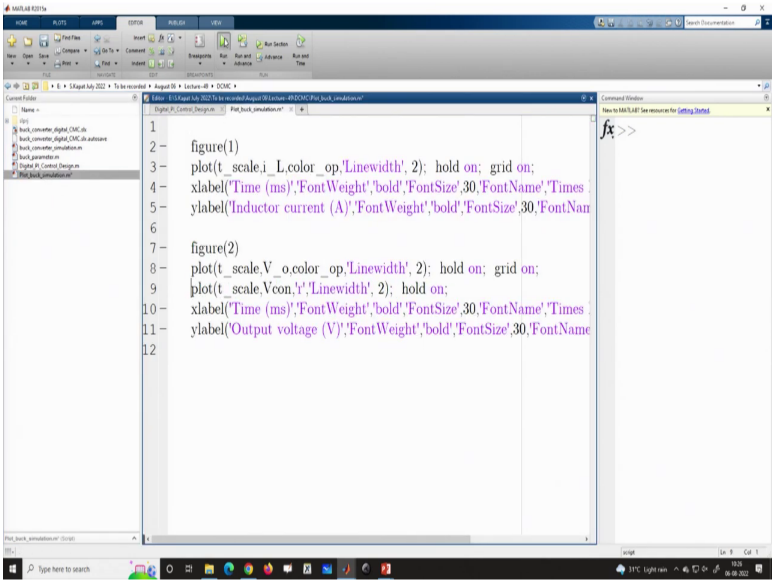
(Refer Slide Time: 12:47)



So, here we are considering 12 volt input and now and initially R was 1. So, if you go to the results, so initially if you consider the average inductor current was 1 ampere. So, because the output voltage is 1 and load resistance is 1 if you go to the output voltage waveform, the output voltage is regulated to 1 volt, more or less.

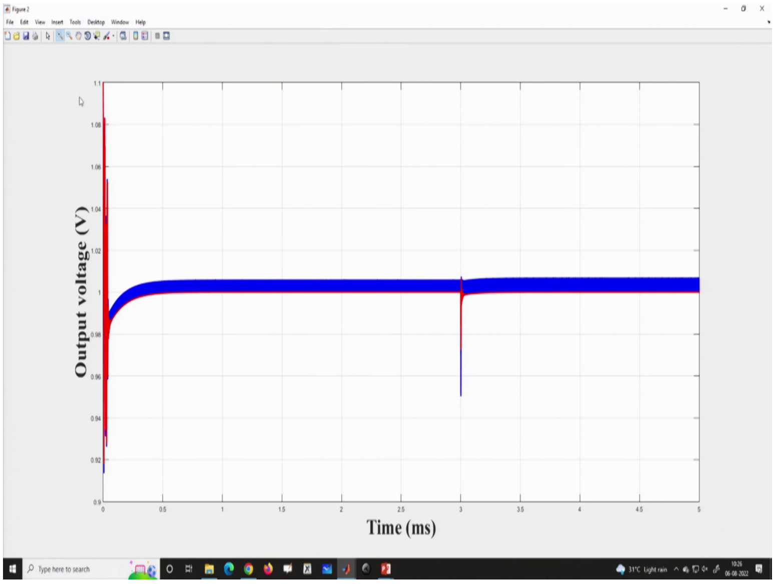
Why it is slightly above 1 volt? Because we are taking the sample right here because we are talking about digital control, it is not analog control, and we are taking one sample per cycle. So, if you want to see how does the sample look like, we can go to the plot and we can rerun this.

(Refer Slide Time: 13:00)

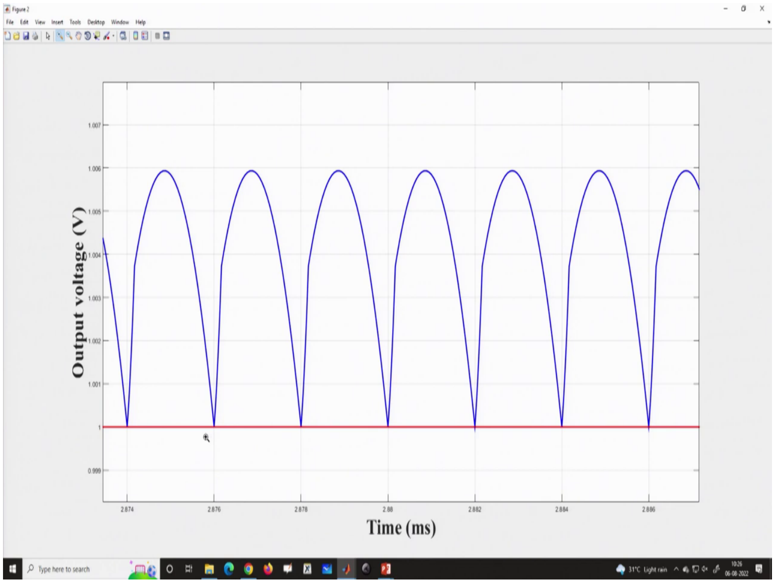


```
figure(1)
plot(t_scale,i_L,color_op,'Linewidth', 2);  hold on;  grid on;
xlabel('Time (ms)','FontWeight','bold','FontSize',30,'FontName','Times
ylabel('Inductor current (A)','FontWeight','bold','FontSize',30,'FontNam

figure(2)
plot(t_scale,V_o,color_op,'Linewidth', 2);  hold on;  grid on;
plot(t_scale,Vcon,'r','Linewidth', 2);  hold on;
xlabel('Time (ms)','FontWeight','bold','FontSize',30,'FontName','Times
ylabel('Output voltage (V)','FontWeight','bold','FontSize',30,'FontName
```

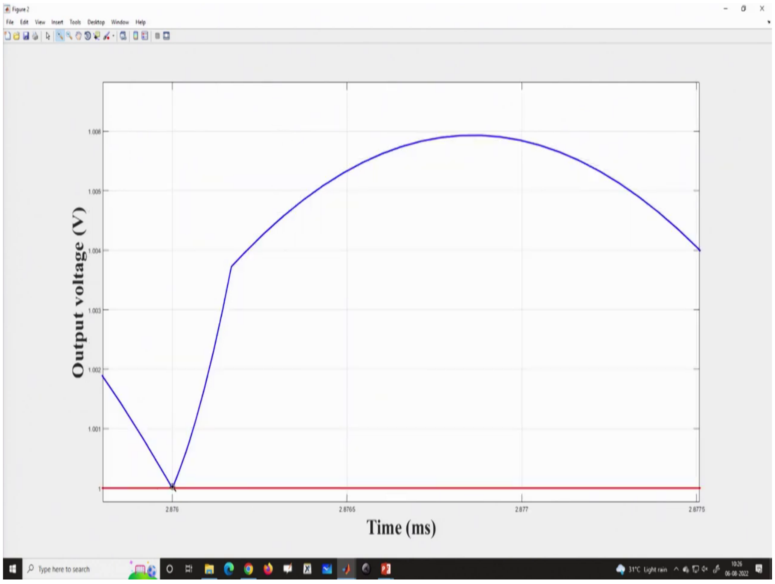
(Refer Slide Time: 13:04)



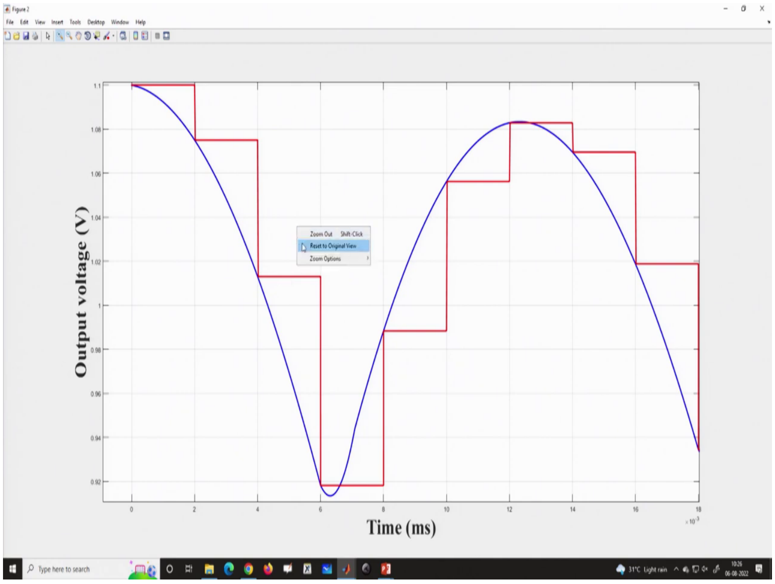
(Refer Slide Time: 13:11)



(Refer Slide Time: 13:11)



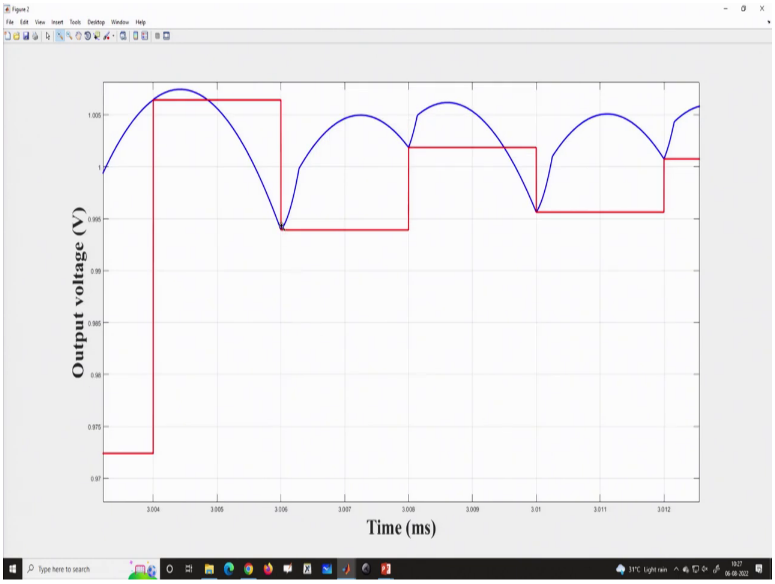
(Refer Slide Time: 13:30)



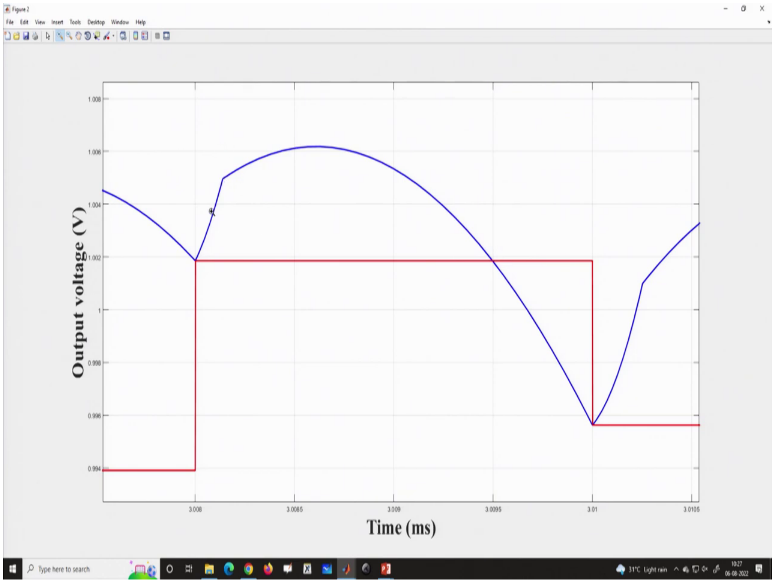
And so, that means if we see the sample that we are picking up the sample right the valley voltage, which means we are regulating this sample voltage, not the full voltage because it is digital control. And we are getting one sample per cycle. That means if you go through all this waveform you see we are taking the sample at every switching instant and which we are regulating.

That means, once it slowly settles down, we are slowly settling down, so this value is settling, which means we are settling the sample voltage, not the original output voltage. As a result, you can see there is a steady-state error in the actual voltage because. But if you take the sample somewhere here then you will get it. So, we will discuss so, first, we have not; that means, in this case, we have not considered any delay between the sampling and the switching.

(Refer Slide Time: 14:05)



(Refer Slide Time: 14:27)

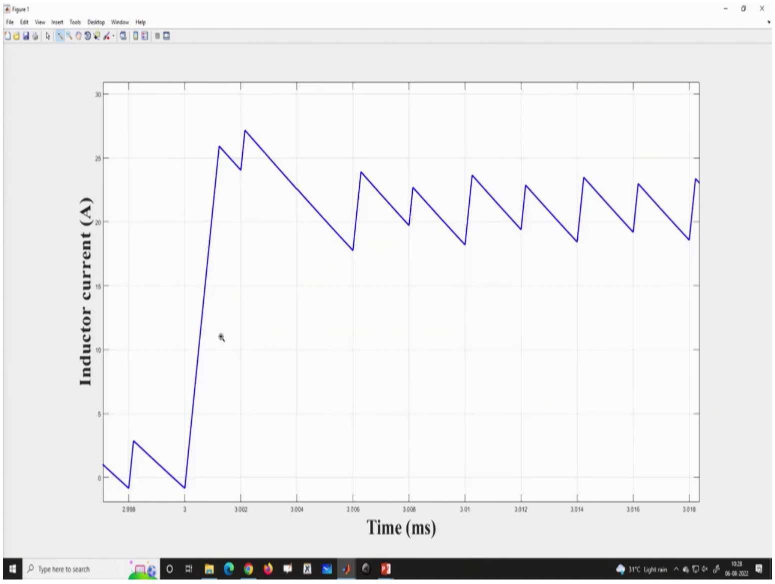


You see the switching is happening at every point and the sample is picked up at the same point. So, there is no delay considered here.

But there is a delay due to the modulator because if you talk about analog control you see the switching is happening here. That means, if you take the one full cycle waveform, the switching is happening here whereas, the sample is picked up here. So, in analog control, till the switching happen your output voltage will change and it will be reflected in the controller. So, the controller output will also get will get updated till the point of switching.

But in the case of digital control, the output voltage is locked; that means, inside the digital controller only the red waveform will be available. That means, all this ripple information is lost. It is not reflected inside. You are only taking the sample. That is why there is a subtle difference between the point of sampling and the point of switching, the voltages are different, and that will cause a difference in the response between analog control and digital control.

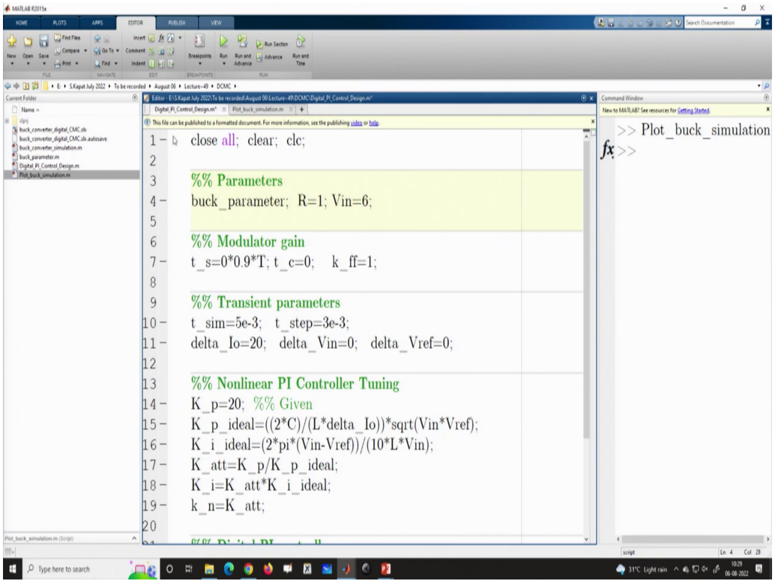
(Refer Slide Time: 15:28)



And this delay actually will look like a delay because there is a time difference between the sampling and the switching and this will cause a delay, ok.

And because of this delay, we will see; so, now, we are talking about the inductor current ripple. So, you see it is not perfect time optimal control, but it is fast enough because there is an in-between switching. That is why we are calling it not exactly time optimal, but we are trying to get close to the optimal response.

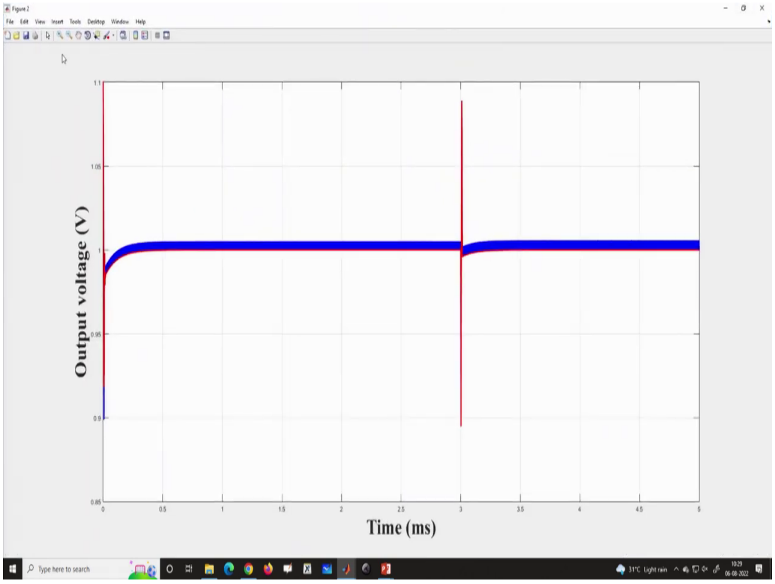
(Refer Slide Time: 15:55)



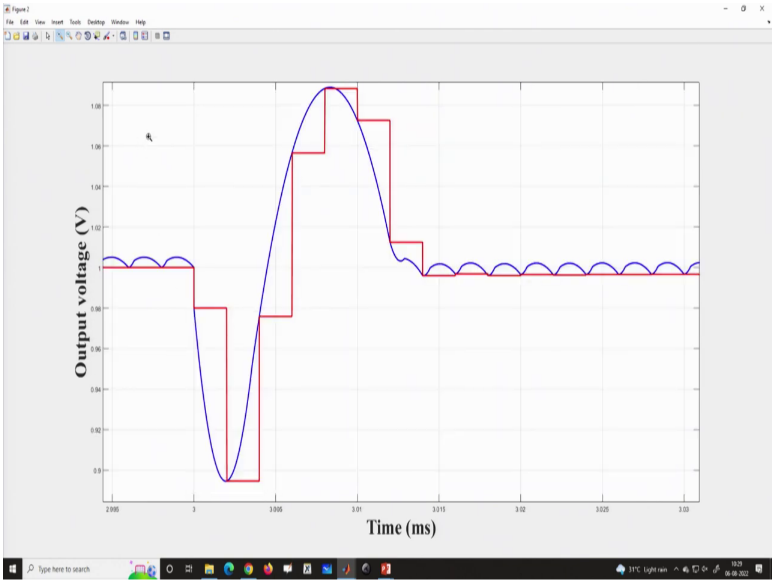
```
close all;  clear;  clc;

%% Parameters
buck_parameter;  R=1; Vin=6;

%% Modulator gain
t_s=0*0.9*T; t_c=0;    k_ff=1;

%% Transient parameters
t_sim=5e-3;   t_step=3e-3;
delta_Io=20;   delta_Vin=0;   delta_Vref=0;

%% Nonlinear PI Controller Tuning
K_p=20;  %% Given
K_p_ideal=((2*C)/(L*delta_Io))*sqrt(Vin*Vref);
K_i_ideal=(2*pi*(Vin-Vref))/(10*L*Vin);
K_att=K_p/K_p_ideal;
K_i=K_att*K_i_ideal;
k_n=K_att;
```

```
>> Plot_buck_simulation
>>
```

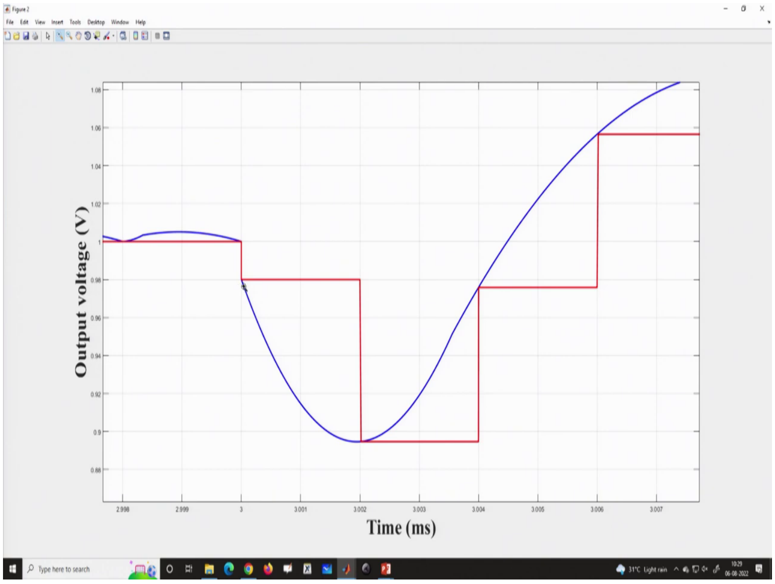
(Refer Slide Time: 16:01)



(Refer Slide Time: 16:10)



(Refer Slide Time: 16:14)

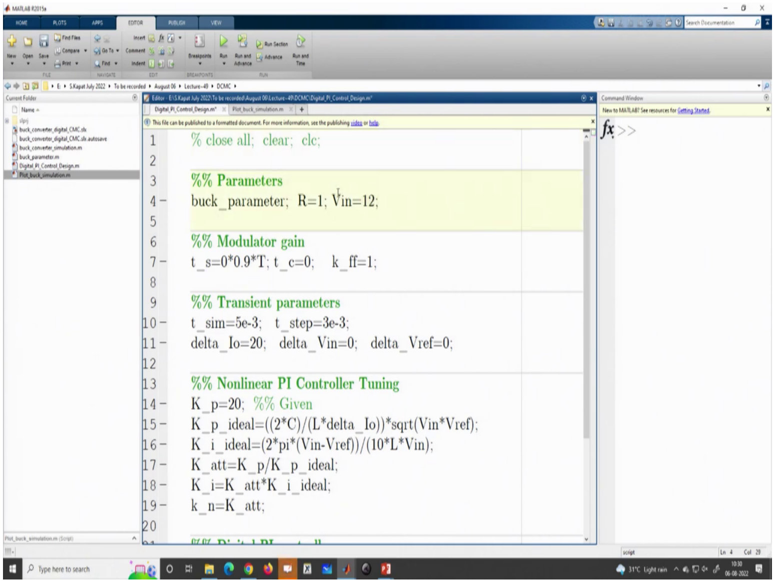


Now, what happens if we decrease the input voltage instead of this decrease the input voltage to 6 volts and rerun the simulation? Now, you will see that if you take the output voltage waveform, there is a difference. There is a subtle difference because here we are picking up the sample and there is a transient, there is an ESR jump it is captured, but at the time of switching this voltage is used where the actual output voltage is gone far below like you know 0.9 volt.

So, in the case of analog control, this output voltage will be reflected in the controller output continuously. But in digital control, you are not taking into account this. So, that means, one way to overcome you have to increase the sampling rate in order to get faster recovery because otherwise this delay is a huge delay and that is causing an additional voltage overshoot using our tuning formula.

That means, the delay between because why this delay has increased because we are using a lower input voltage; that means, if you go to the inductor current waveform, ok. Let us compare this with the earlier case; that means, we will hold on to this value sorry; we will hold on to this value and so this one.

(Refer Slide Time: 17:24)

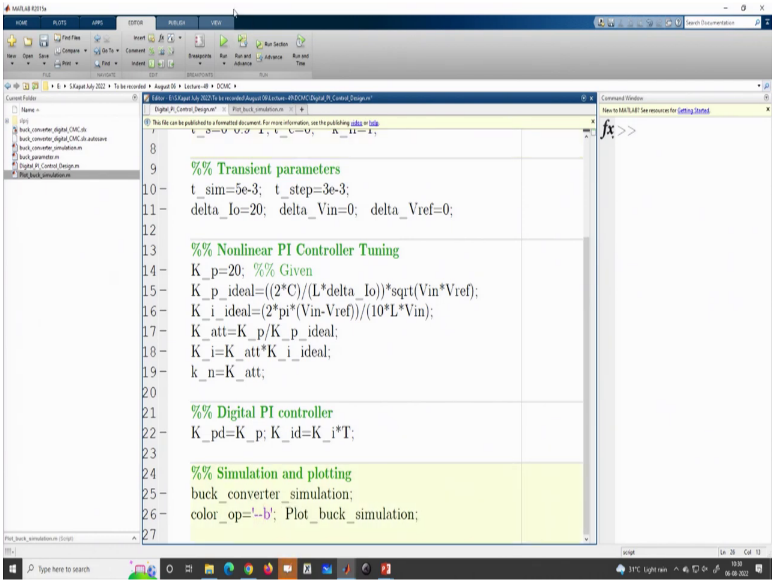


Editor - E:\S.Kapat July 2022\To be recorded\August 06\Lecture-49\DCMC\Digital_PI_Control_Design.m*

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

```
% close all;  clear;  clc;

%% Parameters
buck_parameter;  R=1; Vin=12;

%% Modulator gain
t_s=0*0.9*T; t_c=0;    k_ff=1;

%% Transient parameters
t_sim=5e-3;   t_step=3e-3;
delta_Io=20;   delta_Vin=0;   delta_Vref=0;

%% Nonlinear PI Controller Tuning
K_p=20;  %% Given
K_p_ideal=((2*C)/(L*delta_Io))*sqrt(Vin*Vref);
K_i_ideal=(2*pi*(Vin-Vref))/(10*L*Vin);
K_att=K_p/K_p_ideal;
K_i=K_att*K_i_ideal;
k_n=K_att;
```

(Refer Slide Time: 17:36)



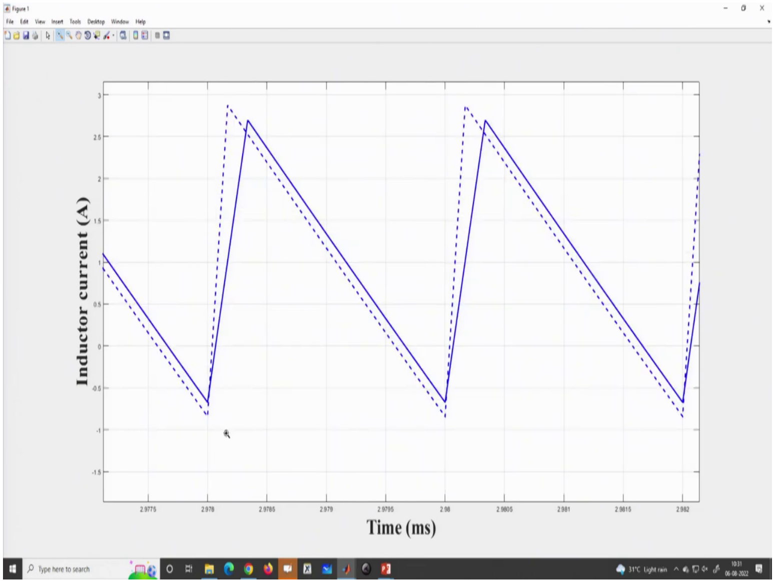
This file can be published to a formatted document. For more information, see the publishing video or help.

```
%% Transient parameters
t_sim=5e-3;  t_step=3e-3;
delta_Io=20;  delta_Vin=0;  delta_Vref=0;

%% Nonlinear PI Controller Tuning
K_p=20;  %% Given
K_p_ideal=((2*C)/(L*delta_Io))*sqrt(Vin*Vref);
K_i_ideal=(2*pi*(Vin-Vref))/(10*L*Vin);
K_att=K_p/K_p_ideal;
K_i=K_att*K_i_ideal;
k_n=K_att;

%% Digital PI controller
K_pd=K_p; K_id=K_i*T;

%% Simulation and plotting
buck_converter_simulation;
color_op='--b';  Plot_buck_simulation;
```

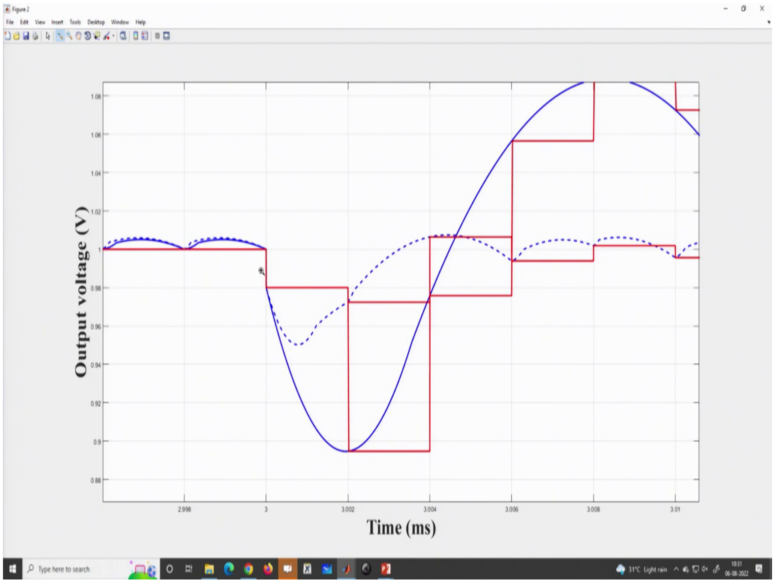
And now we will use 12 volt and we want to change the color. So, the plot is common; that means, there is a plot reference. So, we will use a dashed line and let us do that.

So, I want to show you two responses, and the difference between two responses. So, first, if you go to the inductor current waveform, I will show you that there is a change in the duty ratio.

(Refer Slide Time: 17:57)



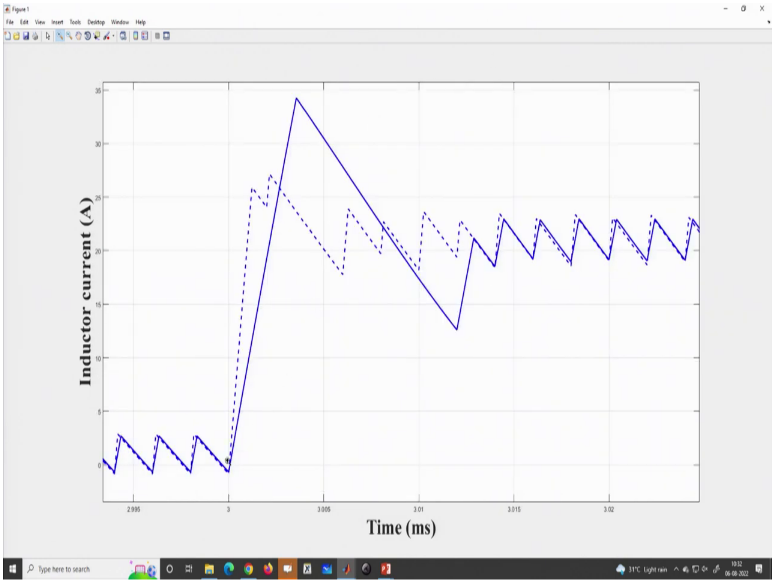
(Refer Slide Time: 18:20)



So, that means, the 12 volt will have a smaller duty ratio, as a result, the on time is small, but 6 volt has a larger duty ratio, so on time is large.

Now, next, go to the output voltage waveform. If you see the output voltage waveform the delay between switching under high input voltage; you see this is the sample voltage and this point is switching happening. So, it is not that much large.

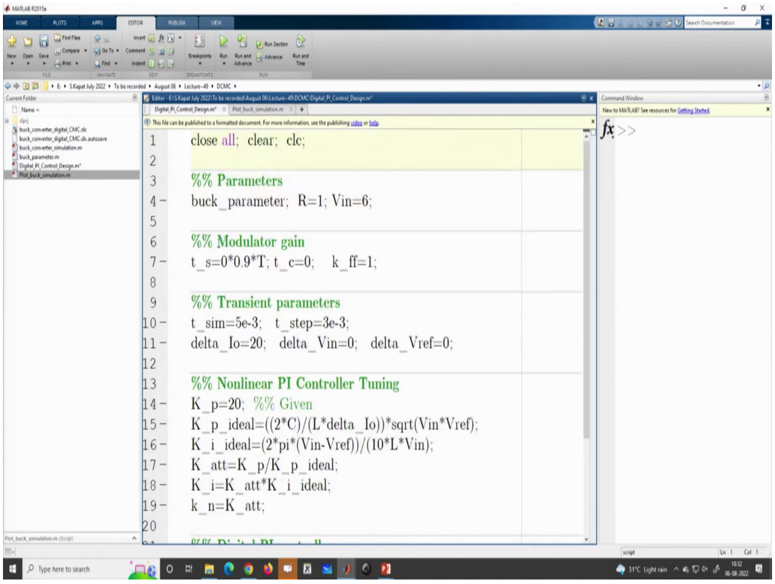
(Refer Slide Time: 18:45)



But in this case, it is large and the voltage is fast discharging because the slope of the inductor current is slower. If you go to the inductor current waveform, you will find the slope of the inductor current is slower because, at a high input voltage, your rising slope is higher. After all, it is v in minus v 0 by L. Whereas, for lower input voltage the slope is slower. As a result, the rate of discharge of the capacitor is faster in the case of the lower input voltage.

And that is why even a slight increase in the delay will cause a huge undershoot and that creates a mismatch between the sample voltage and the switching voltage where the actual switching taking place. And that is causing a mismatch and that is resulting in a large overshoot into this.

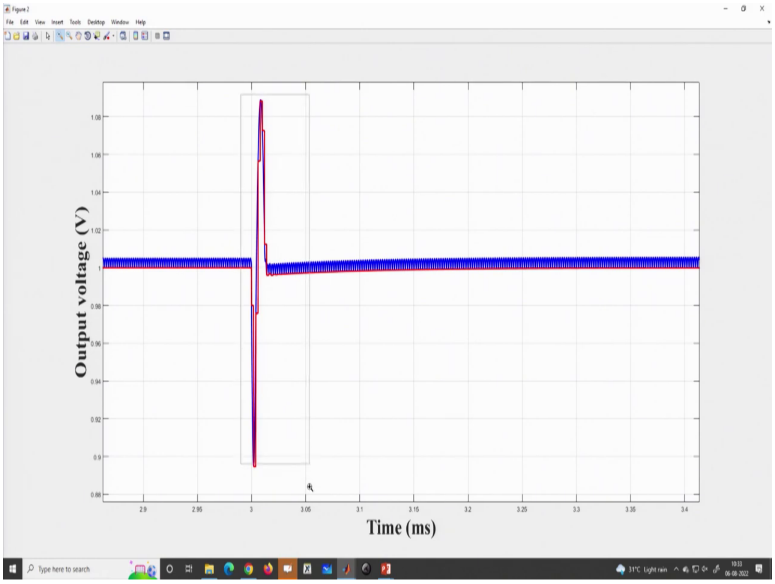
(Refer Slide Time: 19:31)


MATLAB R2015a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Editor - E:\S.Kapat July 2022\To be recorded\August 06\Lecture-49\DCMC\Digital_PI_Control_Design.m*
Digital_PI_Control_Design.m*
Plot_buck_simulation.m
buck_converter_digital_CMC.slx
buck_converter_digital_CMC.slx.autosave
buck_converter_simulation.m
buck_parameter.m
Digital_PI_Control_Design.m*
Plot_buck_simulation.m
Command Window
Type here to search

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

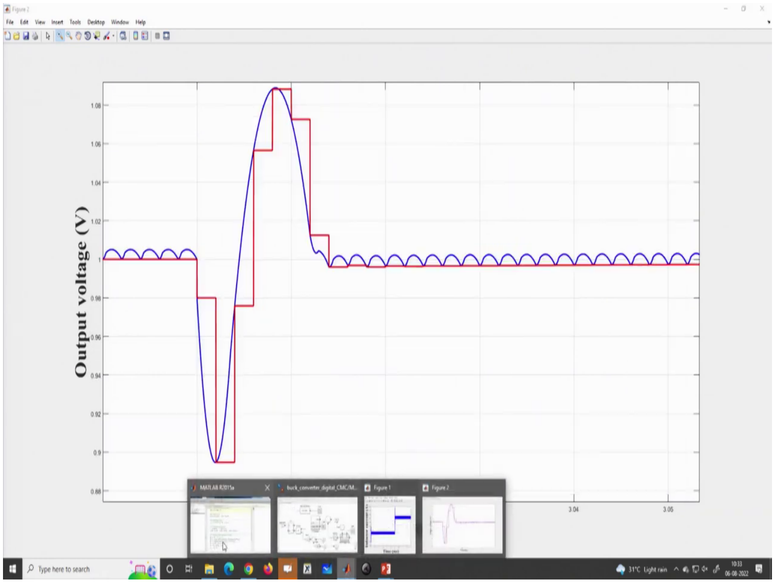
```
close all;  clear;  clc;

%% Parameters
buck_parameter;  R=1; Vin=6;

%% Modulator gain
t_s=0*0.9*T; t_c=0;    k_ff=1;

%% Transient parameters
t_sim=5e-3;   t_step=3e-3;
delta_Io=20;   delta_Vin=0;   delta_Vref=0;

%% Nonlinear PI Controller Tuning
K_p=20;  %% Given
K_p_ideal=((2*C)/(L*delta_Io))*sqrt(Vin*Vref);
K_i_ideal=(2*pi*(Vin-Vref))/(10*L*Vin);
K_att=K_p/K_p_ideal;
K_i=K_att*K_i_ideal;
k_n=K_att;
```

(Refer Slide Time: 19:44)



So, that means, if we go back to our code, so if we want to make changes here 6 volt. So, we want to just draw the plot and analyze that this is because of the delay. And during this delay time, the rate of discharge of capacities may much faster and that is causing a whole lot of issues.

(Refer Slide Time: 19:45)



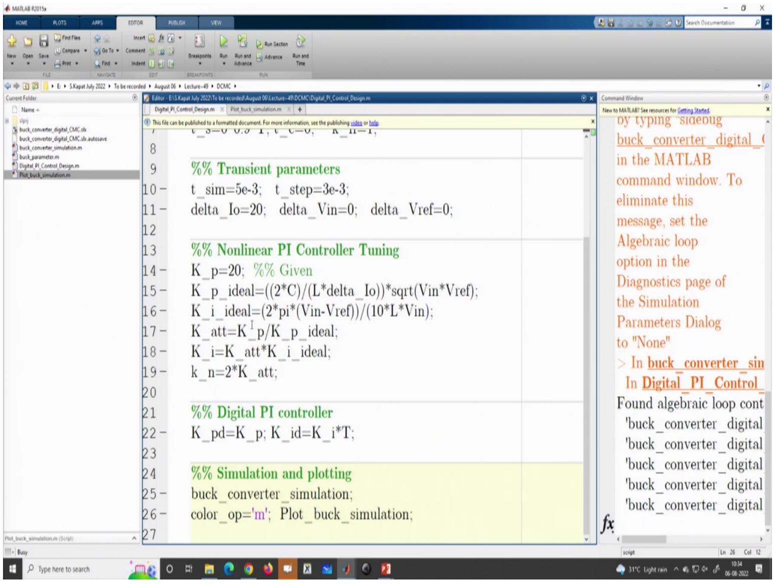
Then, what can we do? One way is because you need to you cannot do anything with the delay, but there is an additional overshoot. So, you need to increase the current loop gain, so that it will create some kind of damping. Because if you consider the inductor current minus load current which resembles the capacitor current, which gives some information about the derivative of the output voltage.

So, if you can increase this term, then the derivative action will come and it will add some phase boost. As a result, you can reduce some undershoot effects. So, that means, instead of this attenuation factor in the original derivation, we multiply 2 time; that means, we increase this gain. In an optimal sense, it should have been this, but due to the delay as well as you know due to the delay, due to the modulator delay, it is causing a deviation and that is resulting in an additional overshoot.

But now we have introduced an increase in the gain in the current loop so that it will give some phase boost action. And we want to see what will be the effect on top of this by changing the colour.

So, we will use you know maybe magenta colour and see what is the effect.

(Refer Slide Time: 21:05)



Editor - E:\S.Kapat July 2022\To be recorded\August 06\Lecture-49\DCMC\Digital_PI_Control_Design.m

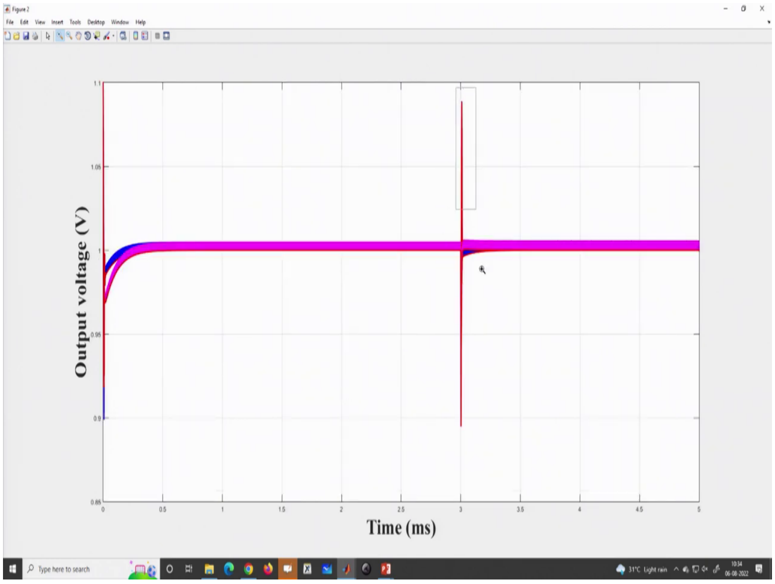
```
%% Transient parameters
t_sim=5e-3;  t_step=3e-3;
delta_Io=20;  delta_Vin=0;  delta_Vref=0;

%% Nonlinear PI Controller Tuning
K_p=20;  %% Given
K_p_ideal=((2*C)/(L*delta_Io))*sqrt(Vin*Vref);
K_i_ideal=(2*pi*(Vin-Vref))/(10*L*Vin);
K_att=K_p/K_p_ideal;
K_i=K_att*K_i_ideal;
k_n=2*K_att;

%% Digital PI controller
K_pd=K_p; K_id=K_i*T;

%% Simulation and plotting
buck_converter_simulation;
color_op='m';  Plot_buck_simulation;
```

Command Window

```
by typing 'sldebug
buck_converter_digital_(
in the MATLAB
command window. To
eliminate this
message, set the
Algebraic loop
option in the
Diagnostics page of
the Simulation
Parameters Dialog
to "None"
> In buck_converter_sin
  In Digital_PI_Control_
Found algebraic loop cont
 'buck_converter_digital
 'buck_converter_digital
 'buck_converter_digital
 'buck_converter_digital
 'buck_converter_digital
```

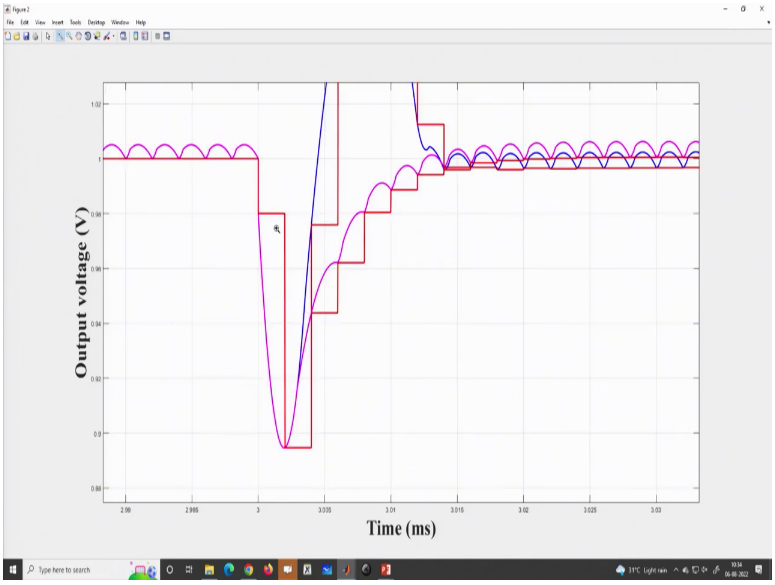
So, now, we have used a sort of similar to phase boost and want to see what is its effect on the transient response, whether can we reduce the current you know the overshoot, the additional overshoot can we reduce or not. That is what is the objective.

(Refer Slide Time: 21:32)

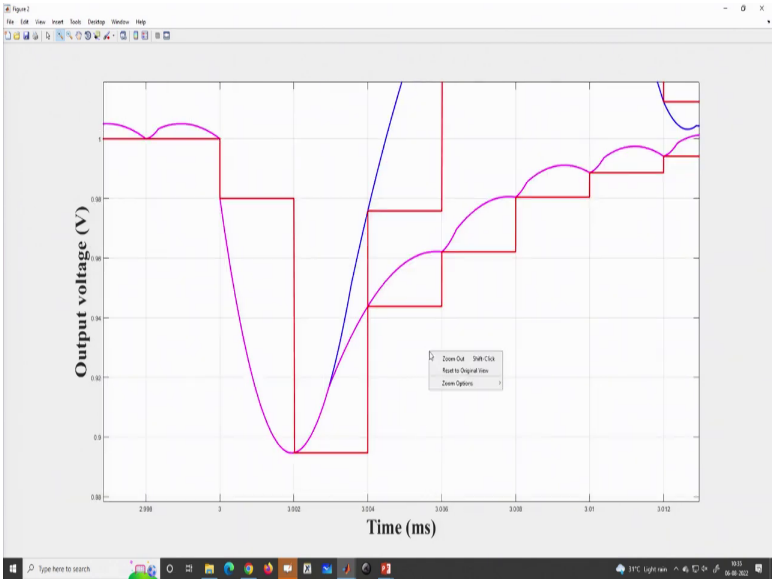


So, now let us go back and see the magenta color.

(Refer Slide Time: 21:37)

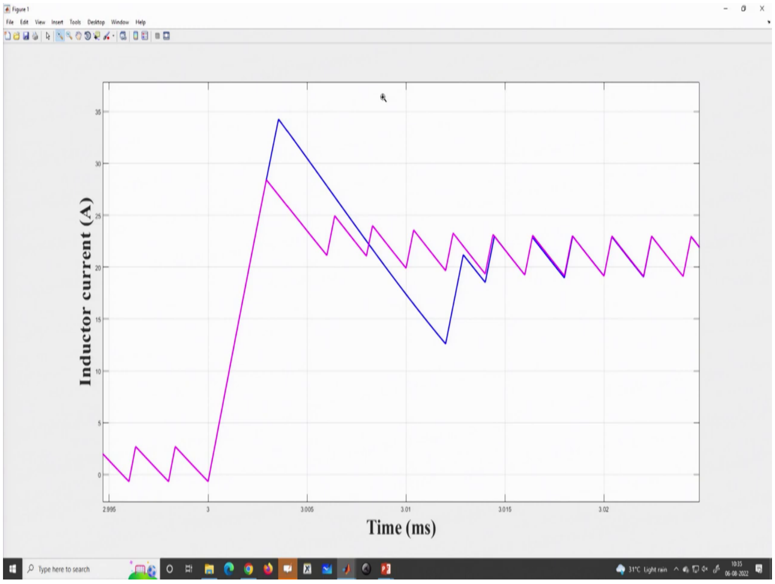


(Refer Slide Time: 21:38)



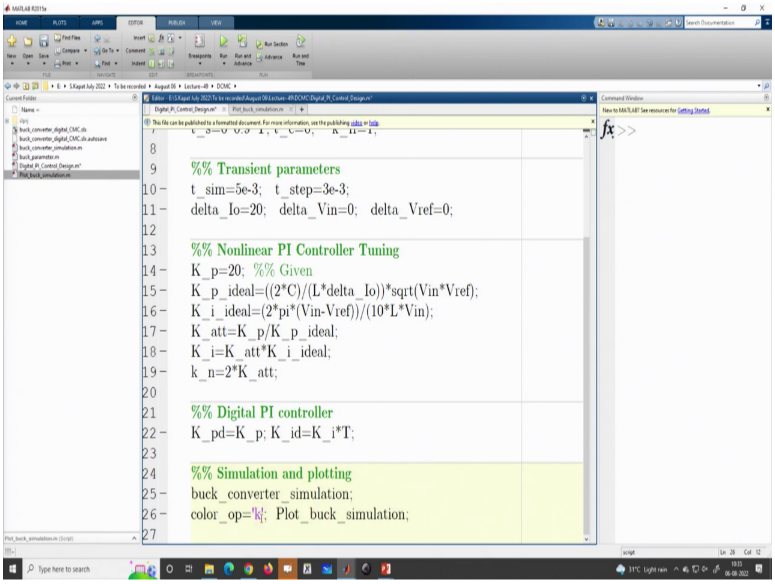
So, the magenta color can reduce the effect; that means, if we go to the inductor current waveform, you see the magenta color has reduced the current overshoot. Because we have introduced something similar to phase boost and that is also obvious here.

(Refer Slide Time: 21:51)



So, you can see between these two colors.

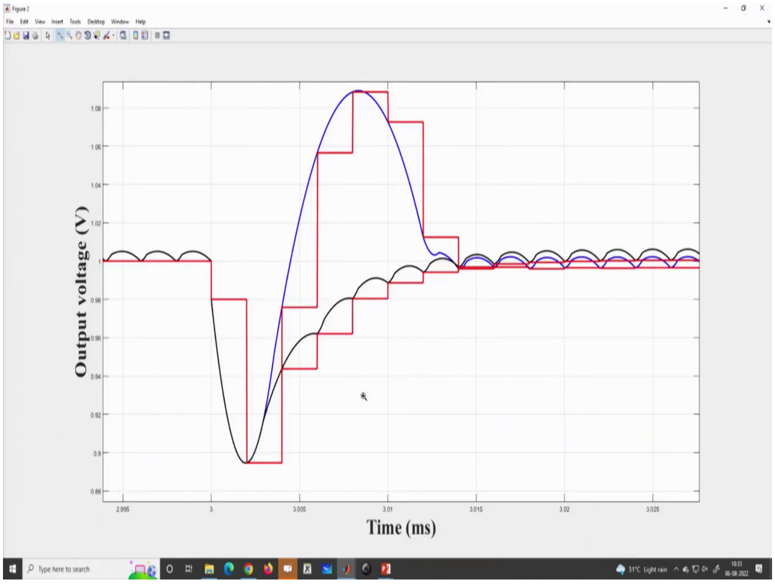
(Refer Slide Time: 22:12)


Editor - E:\S.Kapat July 2022\To be recorded\August 06\Lecture-49\DCMC\Digital_PI_Control_Design.m*
Digital_PI_Control_Design.m*
Plot_buck_simulation.m
This file can be published to a formatted document. For more information, see the publishing video or help.

```
%% Transient parameters
t_sim=5e-3;  t_step=3e-3;
delta_Io=20;  delta_Vin=0;  delta_Vref=0;

%% Nonlinear PI Controller Tuning
K_p=20; %% Given
K_p_ideal=((2*C)/(L*delta_Io))*sqrt(Vin*Vref);
K_i_ideal=(2*pi*(Vin-Vref))/(10*L*Vin);
K_att=K_p/K_p_ideal;
K_i=K_att*K_i_ideal;
k_n=2*K_att;

%% Digital PI controller
K_pd=K_p; K_id=K_i*T;

%% Simulation and plotting
buck_converter_simulation;
color_op='k'; Plot_buck_simulation;
```

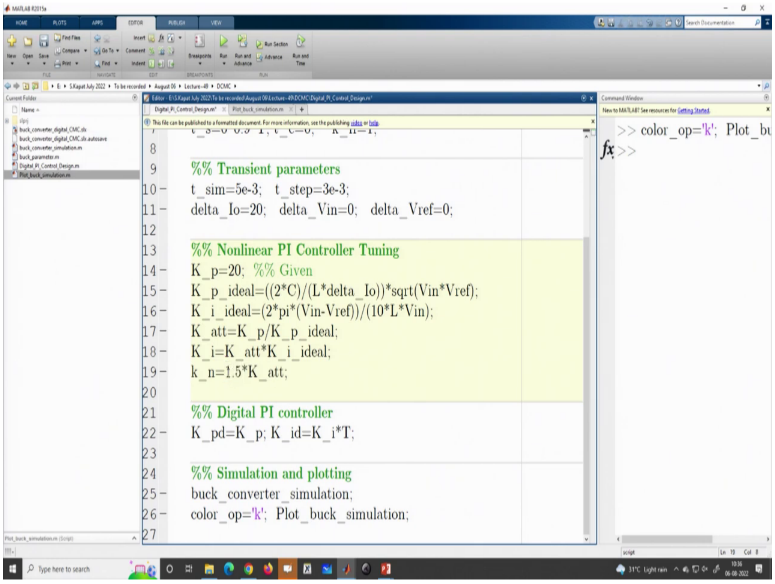
So, instead of magenta let us use another color. So, maybe we can use a black color that will be much more visible. So, we will rerun this, this particular, ok. So, now it is clear.

(Refer Slide Time: 22:20)



So, the blue one was the response by using the optimal tuning gain and that was giving additional voltage overshoot. But the black one is now the response by increasing the current loop gain from the optimal value, whatever we got. We have increased so that it gives a phase boost. But probably we have given an excess phase boost and that is why.

(Refer Slide Time: 22:46)



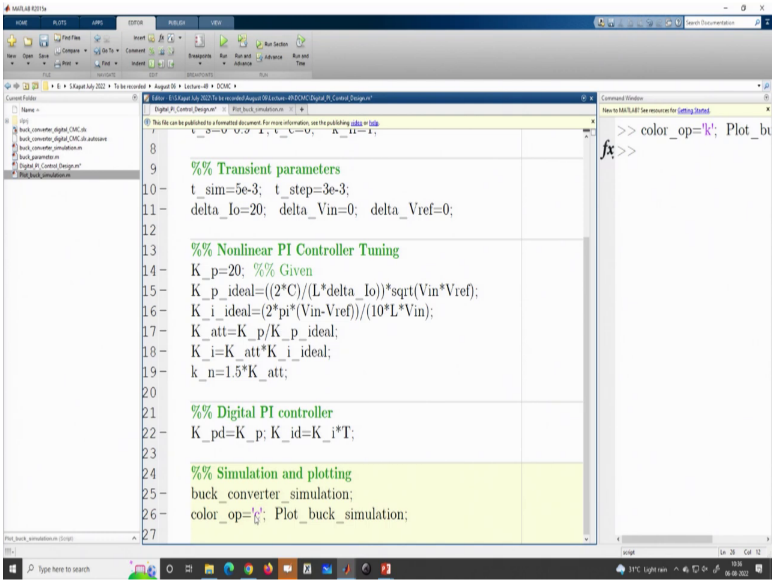
Editor - E:\S.Kapat July 2022\To be recorded\August 06\Lecture-49\DCMC\Digital_PI_Control_Design.m*

```
%% Transient parameters
t_sim=5e-3;   t_step=3e-3;
delta_Io=20;   delta_Vin=0;   delta_Vref=0;

%% Nonlinear PI Controller Tuning
K_p=20;  %% Given
K_p_ideal=((2*C)/(L*delta_Io))*sqrt(Vin*Vref);
K_i_ideal=(2*pi*(Vin-Vref))/(10*L*Vin);
K_att=K_p/K_p_ideal;
K_i=K_att*K_i_ideal;
k_n=1.5*K_att;

%% Digital PI controller
K_pd=K_p; K_id=K_i*T;

%% Simulation and plotting
buck_converter_simulation;
color_op='k';  Plot_buck_simulation;
```

Command Window

```
>> color_op='k';  Plot_bu
fx >>
```

(Refer Slide Time: 22:57)



Editor - E:\S.Kapat July 2022\To be recorded\August 06\Lecture-49\DCMC\Digital_PI_Control_Design.m*

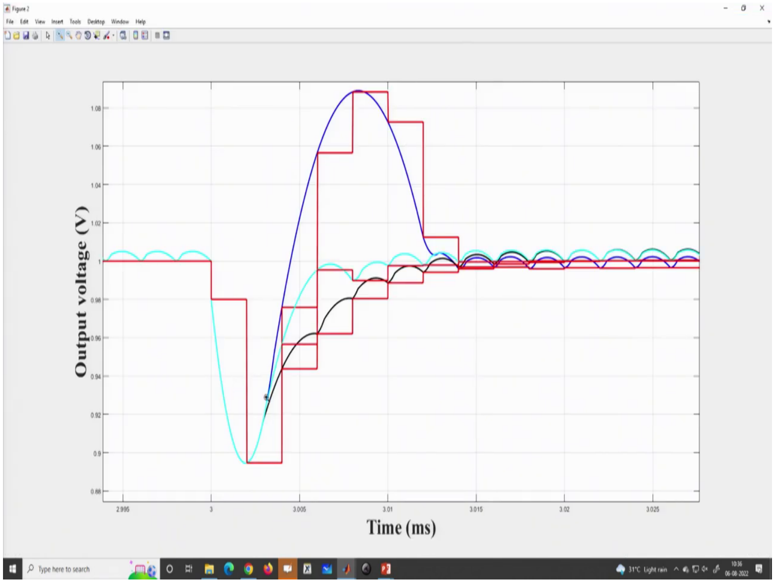
```
%% Transient parameters
t_sim=5e-3;   t_step=3e-3;
delta_Io=20;   delta_Vin=0;   delta_Vref=0;

%% Nonlinear PI Controller Tuning
K_p=20;  %% Given
K_p_ideal=((2*C)/(L*delta_Io))*sqrt(Vin*Vref);
K_i_ideal=(2*pi*(Vin-Vref))/(10*L*Vin);
K_att=K_p/K_p_ideal;
K_i=K_att*K_i_ideal;
k_n=1.5*K_att;

%% Digital PI controller
K_pd=K_p; K_id=K_i*T;

%% Simulation and plotting
buck_converter_simulation;
color_op='c';  Plot_buck_simulation;
```

Command Window

```
>> color_op='k';  Plot_bu
fx >>
```

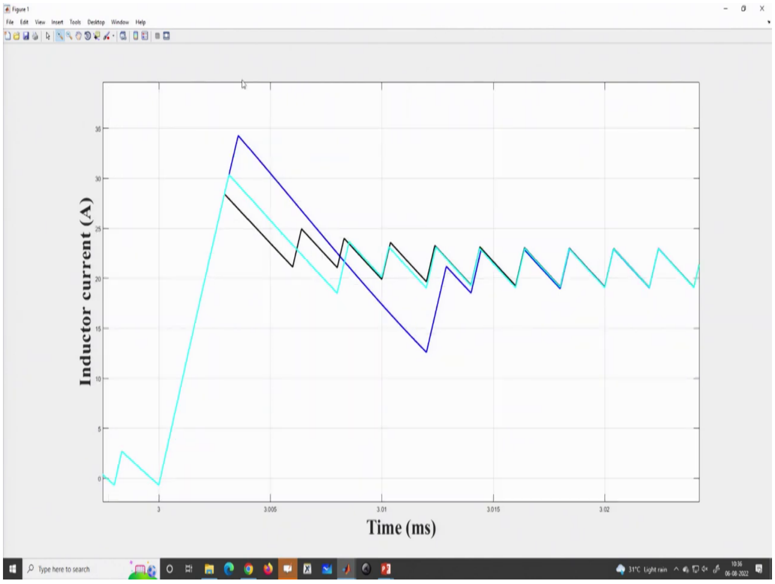
So, now the third response we want to get, instead of 2 if we put 1.5. And now we rerun the simulation, and we use a different colour, let us say we will use cyan colour and see what happens. So, here instead of 2, we have given 1.5.

(Refer Slide Time: 23:05)



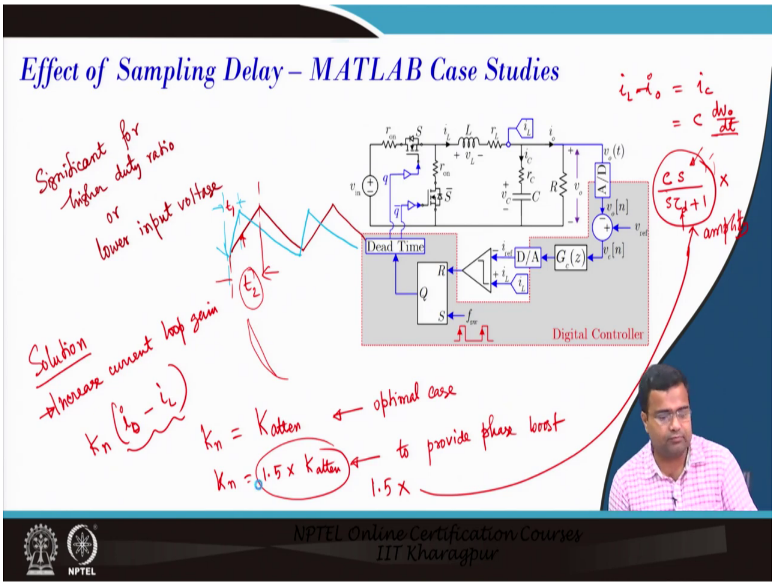
So, now using cyan color is somewhat better. It is somewhat close to the optimal. It is not exactly optimal, but somewhat close. If you go to the inductor current waveform it is somewhat close to the optimal behavior.

(Refer Slide Time: 23:20)



So, that means, we can play with this game by considering this sampling delay constant, ok. And these parts are important characteristics.

(Refer Slide Time: 23:37)



Effect of Sampling Delay – MATLAB Case Studies

Significant for higher duty ratio or lower input voltage

Solution → increase current loop gain

$K_n(i_0 - i_L)$

$K_n = K_{atten}$ ← optimal case

$K_n = 1.5 \times K_{atten}$ ← to provide phase boost

1.5 x

$i_L - i_o = i_c = C\frac{dv_o}{dt}$

$\frac{Cs}{s\tau_d+1}$ x

amplify

NPTEL Online Certification Courses
IIT Kharagpur

So, that means, we will let you know if we go back to our thing; that means, the effect of sampling delay what we saw; that this sampling delay effect, that effect is significant for higher duty ratio or in other sense or lower input voltage. And what was the cause? Because your delay amount was increasing; what I am saying is that if you take the inductor current waveform in one case like this and another case you have a smaller duty ratio. So, now, we are talking about the sampling point here.

So, this to this delay, I am talking about this delay and if you take this color, this to this delay. So, t 2 and this is t 1. So, t 2 is larger. And not only that this slope is smaller, so it will cause more undershoot. So, as a result, the impact of undershooting will be more prominent. It is a cascading effect. Both delays are increasing as well as the voltage rate of discharge is also increasing and that is causing a whole lot of problems.

So, then what is the solution? So, the solution is one way to increase we can increase the current loop gain. That means, we are talking about K n, which is i L, i 0 minus i L feedback loop. So, for optimal we choose the K attenuation factor, that is for the optimal case. But now we will choose K n maybe some 1.2 time K attenuation that is to provide phase boost.

Why will say phase boost? Because of this term, this i L minus i 0 in a buck converter is nothing but the capacitor current. And what is capacitor current? It is C dv C dt or dv 0 dt, ok. So, that means, this is a derivative action and if you take the Laplace it is like a C s. And if there is you know I would say if there is a practical derivative action, then we will have an S tau d plus 1. We talked about this practical derivative gain in the case of our PID controller.

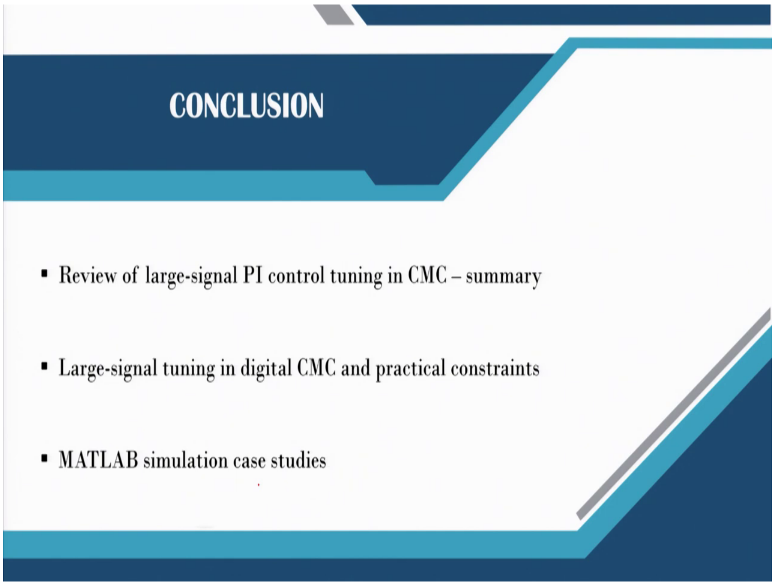
Now, this one since this time constant is very small, the 0 is coming and that is giving an additional phase; that means, the under excess is coming as if because it is a large-signal model. So, there is no concept of frequency response phase plane.

But if someone wants to think of some sort of phase boost, phase lag, because of that effect; that means if there is a lag; that means, a delay will cause a phase lag and the phase lag will cause more oscillation or more overshoot undershoot. But if you want to reduce the phase lag, then you have to provide some additional phase boost and that is provided by this additional term. So, we need to amplify this factor.

That means, we need to provide so, that means, when you are increasing this term, we are, that means, we are increasing this particular action by 1 time, ok. So, that means, what was there, we are providing this.

So, what I mean to say is that this attenuation factor was coming from the optimal gain, you know because we want to make sure that practical gain constants are there. But in addition to that we are giving some excess gain so that we provide some phase boost. So, it tries to anticipate whatever phase lag happens due to the sampling delay.

(Refer Slide Time: 28:19)



CONCLUSION

- Review of large-signal PI control tuning in CMC – summary
- Large-signal tuning in digital CMC and practical constraints
- MATLAB simulation case studies

So, in summary, we have discussed, we have reviewed large-signal PI controller tuning in current mode control. We have discussed large-signal tuning, PI controller tuning in digital current mode control, and their practical constraint.

And we have considered the MATLAB case study, and we tried to cover what is the real-time, and what are the issues; that means, when can you I mean because the sampling delay it can cause additional overshoot understood. Then, how to anticipate some aspect are discussed, but more aspect will be discussed when we will do hardware implementation in the future. That is it for today.

Thank you very much.