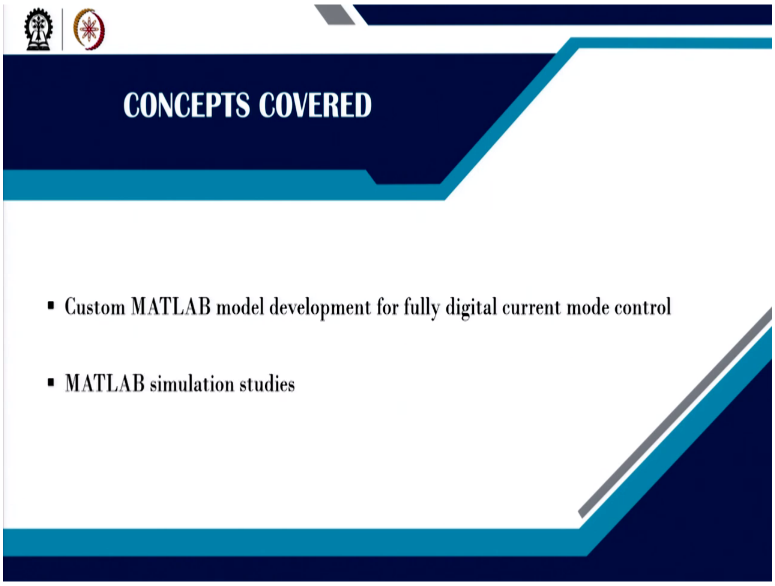
Digital Control in Switched Mode Power Converters and FPGA-based Prototyping

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

Module - 03
MATLAB Custom Model Development under Digital Control
Lecture - 27
MATLAB Model Development for Fully Digital Current Mode Control

Welcome back. So, in this lecture, we are going to talk about fully digital current mode control architecture.

(Refer Slide Time: 00:31)

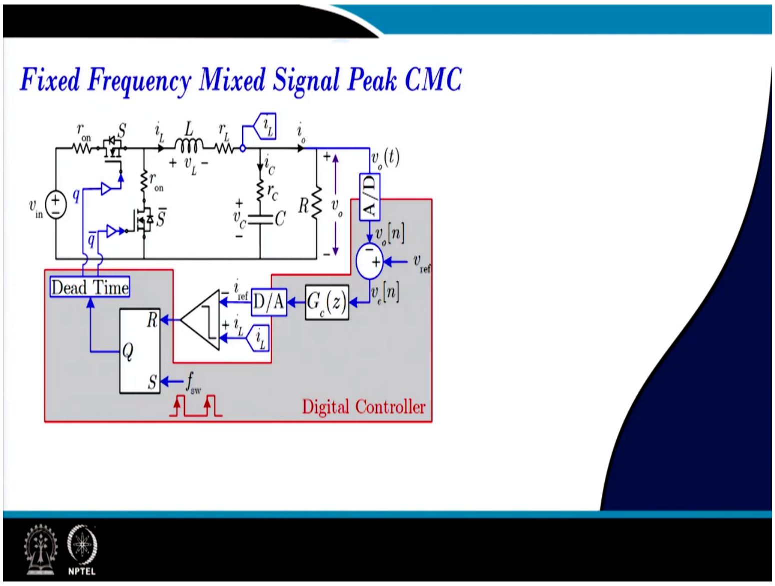


CONCEPTS COVERED

- Custom MATLAB model development for fully digital current mode control
- MATLAB simulation studies

So, first I will show you a custom MATLAB model development for fully digital current mode control and then I want to consider a few simulation case studies.

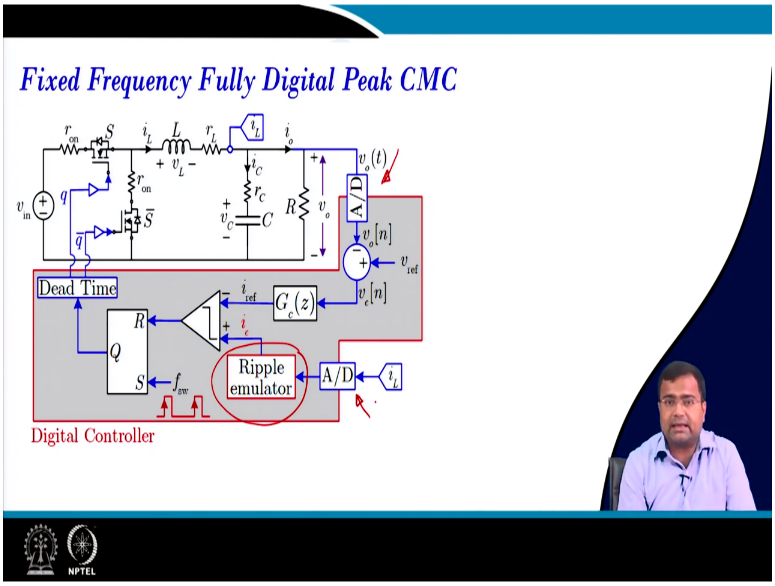
(Refer Slide Time: 00:41)



Fixed Frequency Mixed Signal Peak CMC

So, first, we will start from fixed frequency mixed signal peak current mode control, because we have discussed in the last class, where we have considered the analog current loop and digital voltage loop.

(Refer Slide Time: 00:55)



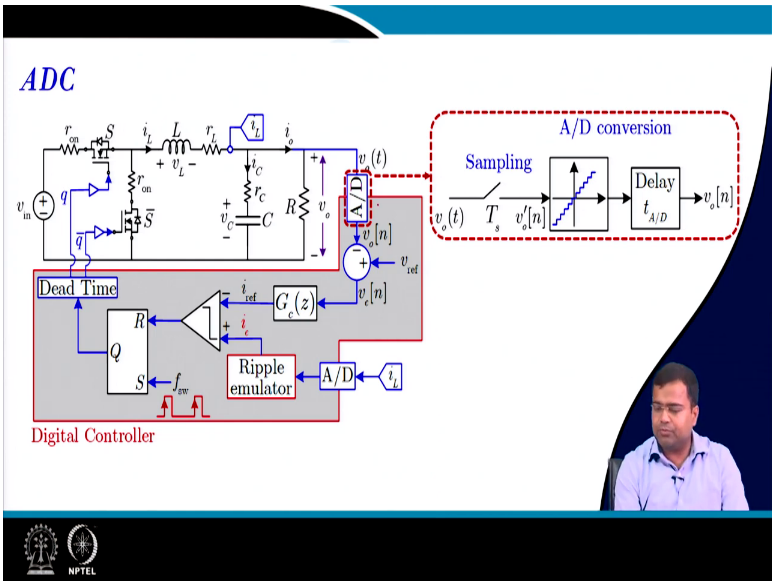
Fixed Frequency Fully Digital Peak CMC

Now, if you go to fully digital current mode control then we also need to sample the inductor current; that means, we have already sampled the output voltage here. But we also need to sample the inductor current and we have discussed there are two possibilities, if we sample the inductor current because we need to retain the ripple information of the inductor current.

So, one way you can use a very high sampling rate for the current loop, but that may not be preferable because that will incur more losses and you need a very high sampling or high-speed A to D converter. But, rather is more you know kind of effective way.

If you can sample once per switching cycle and emulate the ripple of the inductor current internally using the digital control platform that means, but it is not necessary that we need to generate the same rising slope of the inductor current. It may be different ok, but we are trying to emulate this current ripple inside the digital platform and we are taking samples of the inductor current once per switching cycle.

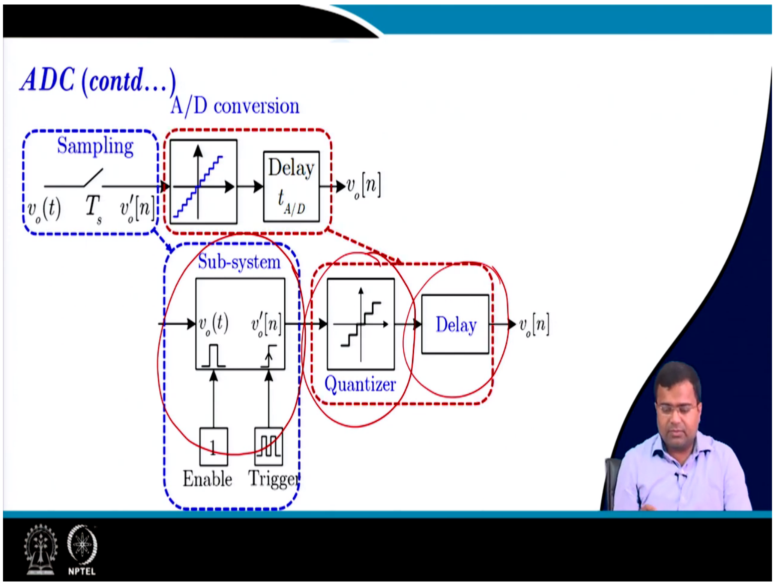
(Refer Slide Time: 02:05)



ADC

So, how to start with? So, we want to repeat the same process as the A to D converter that we have discussed for the voltage loop it has sampler quantization and the delay.

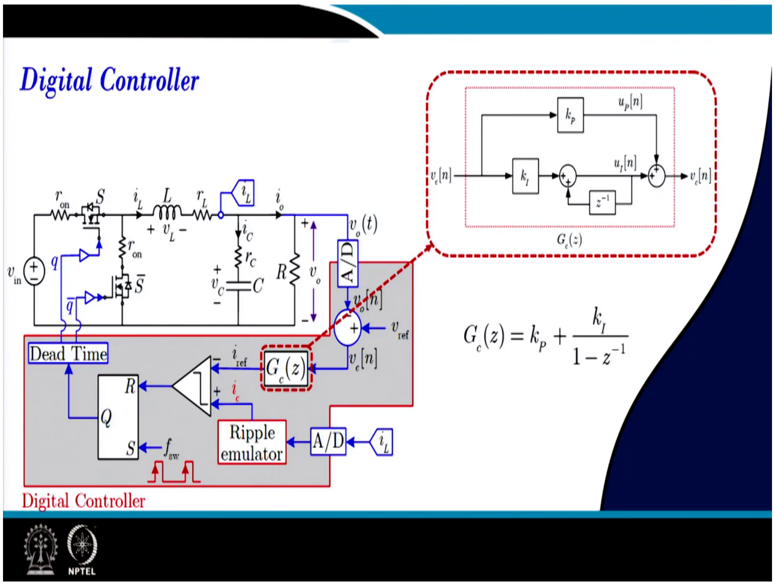
(Refer Slide Time: 02:15)



ADC (contd...)

And we have already discussed that the whole A to D process can be divided into this event trigger sampling followed by quantization and you can add one additional delay also.

(Refer Slide Time: 02:31)

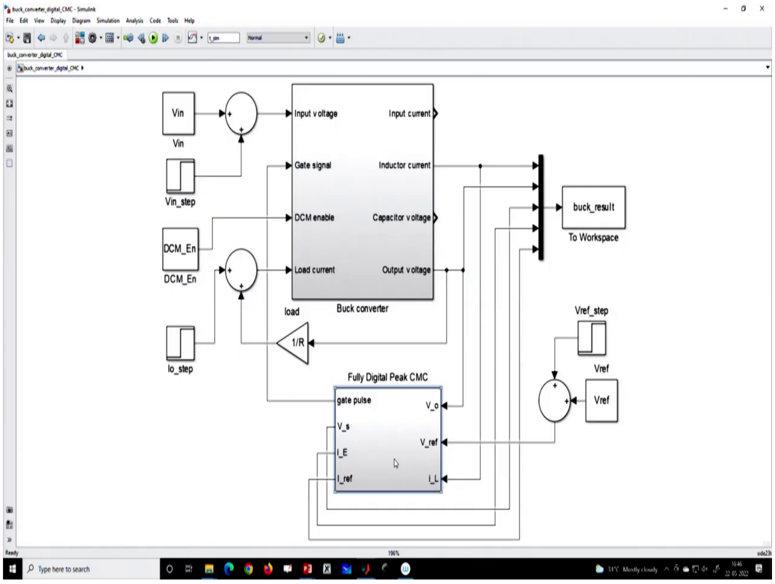


Digital Controller

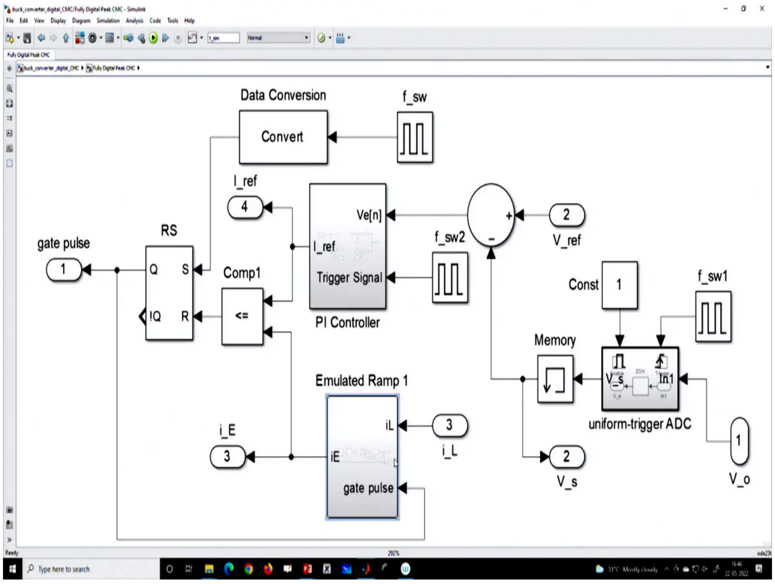
$$G_c(z) = k_p + \frac{k_I}{1-z^{-1}}$$

Next, this part we have discussed and we have discussed it for multiple lectures. We have also discussed how to implement a PI controller. In fact, in the previous lecture, we discussed that if you want to implement a PI controller using a difference equation, not by transfer function then it has to be clock synchronized.

(Refer Slide Time: 02:50)

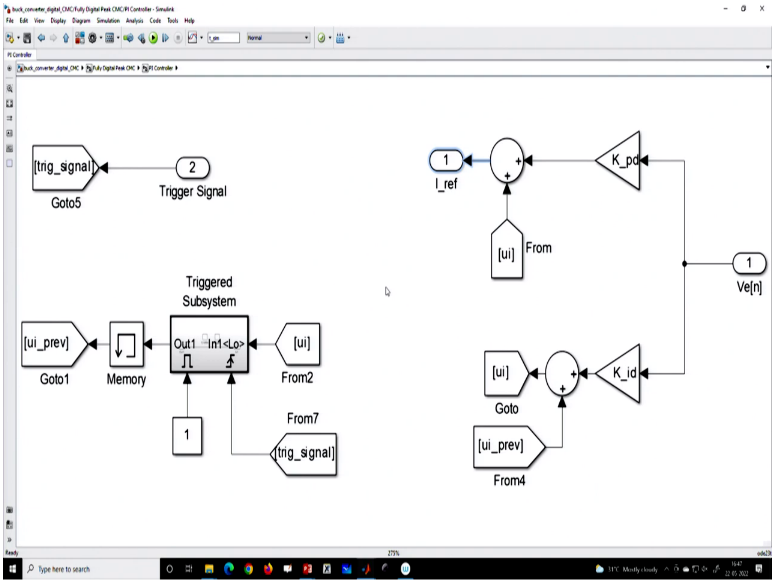


(Refer Slide Time: 02:54)



And if you go back to the MATLAB file; that means, this is a full digital peak current mode control.

(Refer Slide Time: 02:56)

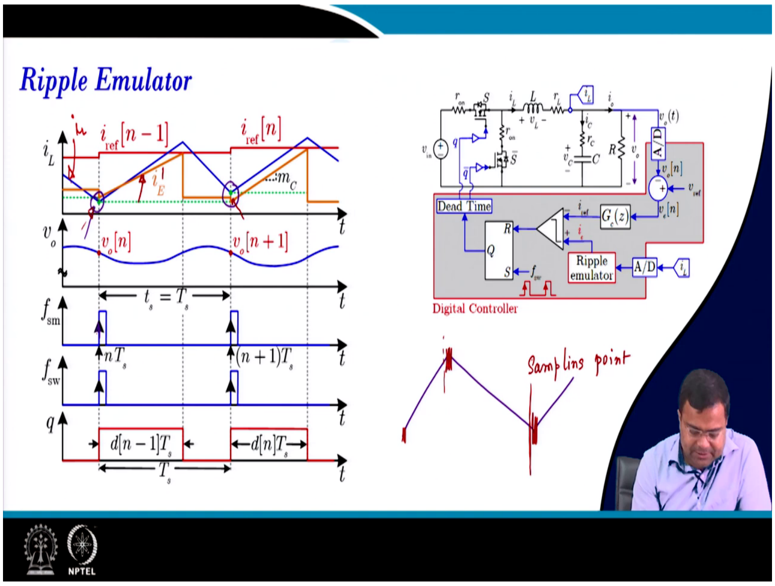


So, this part also we have discussed; means, we have used a trigger pulse to synchronize; that means, this is used to generate ui n minus 1 from ui n; that means, one sample 1 cycle delay of the integral output and that is synchronized with respect to the trigger signal. Similarly, after that trigger previous; that means, ui n minus 1 which is coming here will be added along with the K i into error voltage.

So, K id means discrete-time integral gain into error voltage. So, this is the overall integral controller; that means, ui n is equal to ui n minus 1 plus K id into v n. So, this is the integral controller implementation and this is the proportional control implementation and this output is used here.

So, that means, this is the output of the integral control. This is the output of the proportional control and if you sum them we will get the PI controller output. So, this part also we have discussed. That means, it is not a simple transfer function we are using a difference equation with a clock-synchronized operation.

(Refer Slide Time: 04:11)



Ripple Emulator

Next, this part is very important because here we are not using an analog current loop. We are using emulated current; that means, let us first start with the basic concept. If you sample the inductor current, let us say this is our blue one is the actual inductor current ok.

Now, this is the sample of the actual inductor current; that means, we are sending the sampling command that is the point we want to sample, and generally, this sampling point is slightly earlier than the actual switching takes place. Because if you go to any current sensor right? So, you will find the spike in the current sensor the switching spike. One the switch turns on or the switch turns off.

So, every time; that means, I am talking about this as the actual inductor current, but let us say if you go for any sense inductor current then you will have switching spikes here after the switch turns on and off. So, to avoid a such spike, so we want to sample just before the switch either turns on or turns off. So, in this case, we are talking about the trailing edge modulation. So, we are talking we are taking the sample here; that means, where this is the sampling point.

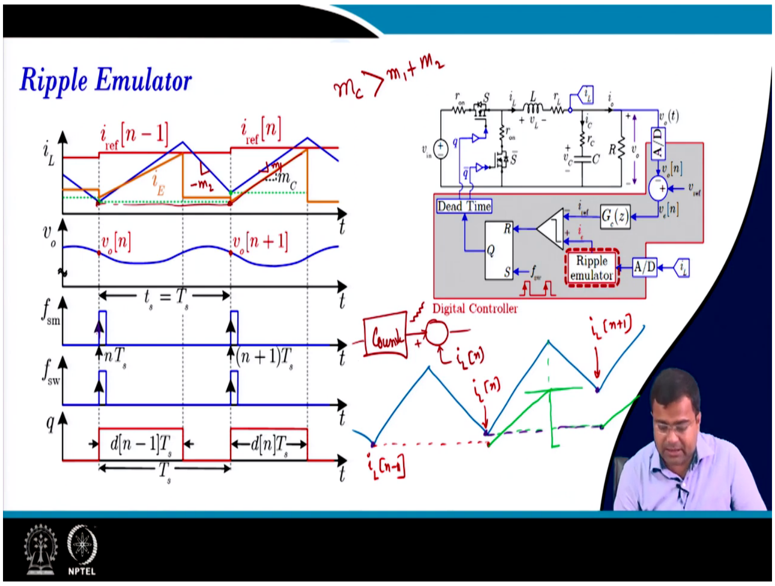
This is just below the switch that is about to turn on; that means, it is in the off state. That means you can say it is a valley current. Now that means, we are sending the sampling command here. But, as I said the A to D converter requires time for conversion; that means, we cannot get any immediate sample available for the controller, because we need to add this

Sawtooth waveform to the ramp I will discuss. But, we cannot use the current sample for the running cycle because we need to provide some time for A to D conversion.

So, that is why the sample command is sent here, but the actual current sample is used in the next cycle. So, there is an inherent 1-cycle delay. So, that means, this is the point of sampling here, but the availability of the sampling including we are forcefully delaying you know for the safe side because we need the information of the valley current. Because we cannot take any other point, because we need to take the valley point.

So, using that valley current information in the next cycle and you can see this one this particular waveform indicates the Sawtooth waveform that is used inside the digital controller; that means if I want to draw let us say this conceptual point.

(Refer Slide Time: 07:00)



Ripple Emulator

So, suppose I am just taking let us say I am just considering the inductor current like this ok. So, this is the inductor current waveform. Now we are sampling the current here; that means, we are just sampling just before the switch turns on.

But, we are using this sample in the next cycle; that means, this sample if we just interpolate this sample if we extend it here. Now we want to add a ramp; that means, we want to create a Sawtooth waveform here this is addition. That means if I take this is my i L n minus 1 and if this sample is i L n and if this sample is i L n plus 1 and so on. So, we want to add a ramp.

So, Sawtooth waveform means it is like a staircase implementation and if you take a very narrow staircase it will be like this.

That means if we want to that means, whenever we start switching at this point we are using a different color. Let us say we want to start this slope may be different, this slope may be different 1 minute. So, this slope should be different, and why we are going up to this is because here we have to turn off. And how it is turned off? As if there is a peak current reference that is coming here.

When this hit the peak current then turn it off and since we are using a counter we will simply reset this. Because we are using the counter and why we have reset it because we want to after this there is a latch operation, it is a trailing edge modulation the switch will cannot change state; that means, once the switch is turned off the switch cannot turn on throughout the rest of the interval.

So, it has to wait for the next clock edge. So, that means, it is not necessary to extend this waveform beyond that or we do not need to emulate the falling slope for peak current mode control because that information is simply redundant, we do not need that. So, we simply reset it.

Next time now this sample point is extended. So, this sample point is extended here; that means, we are using this sample point is extended here. But our Sawtooth waveforms our ramp signal; that means, this will start from here. Again it depends upon the peak current. So, that means, you can see this waveform that means if I rub up all these parts.

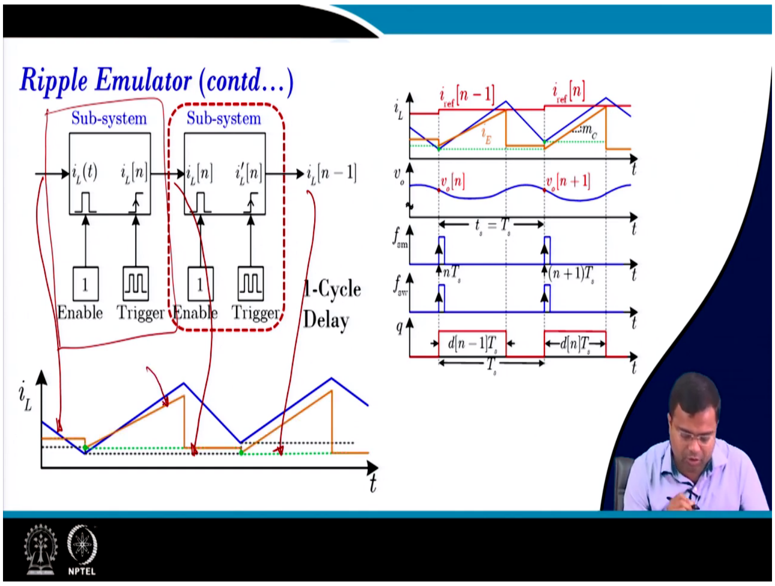
So, you can see the sample of the inductor current is used in the next cycle here and with an added ramp. Although it looks like the ramp is the same as the rising slope, it is not necessary, the ramp can be different. You know there is much research that this ramp should be different and we cannot keep this m 1 slope because it will be unstable even with this m c requirement when I go to the analysis part.

It can be shown that the m c has to be greater than m 1 plus m 2; that means, m 1 is the minus m 2 is the falling slope of the actual inductor current and m 1 is the rising slope of the inductor current. So, this m c has to be greater, otherwise, it will be inner current loop will be unstable and that can be proved. But what I am showing here is that you need to add a ramp ok to emulate this current.

Now, so I think the basic operation is clear. We have to take the sample of the inductor current we have to delay by 1 cycle and then we have to add the compensation. So, compensating ramp or the emulated ramp is always added in every cycle, but the only thing it base point depends because the if you say a counter let us say if say counter which generates this Sawtooth of waveform like this the output of the counter should be added with our sample current, whether it is i L n i L n plus 1 it will be updated.

So; that means, the base value will be changed according to the value of the inductor current sample ok.

(Refer Slide Time: 11:32)



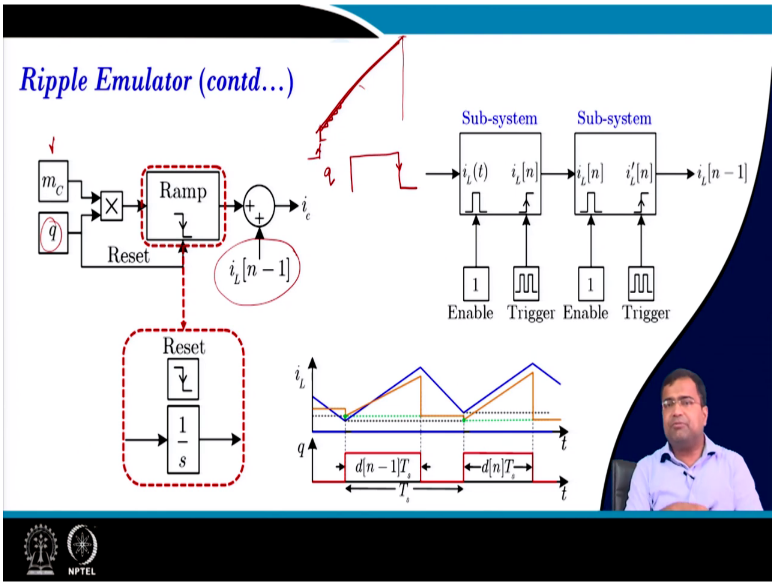
Ripple Emulator (contd…)

So, that means, now so how do you generate this sample inductor current? So, this is we know we can simply use an A to D converter block and that we know even trigger a DC, but we need to delay by one cycle, so you would another. So, if you simply use this current. This is the actual inductor current which is shown here and this current is shown here. This is a sample current.

You see it is taking the sample of the current just it is updating. But we need a realistic current sample for the controller, because this sample may not be available immediately. So, we need to use another such block which will delay by 1 cycle. And if you see 1 cycle delay, so this waveform is the green waveform here. So, that means, we are generating the first block will generate the sample and hold of the current valley point and the second block will delay by 1 cycle.

So, this is the overall implementation of the sample value, but it is not sufficient. Because we have to add this ramp because, after this sample point, we have to add this ramp; that means, we need an additional block of this.

(Refer Slide Time: 12:42)



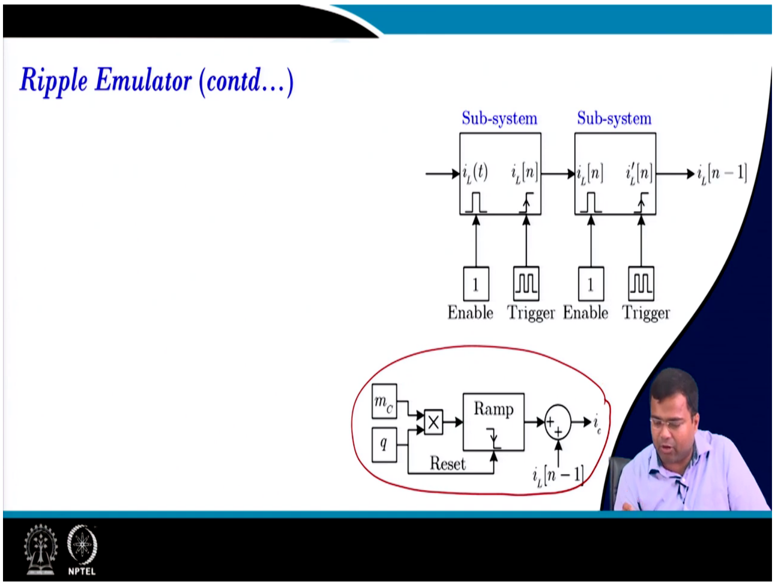
Ripple Emulator (contd...)

And how to generate a ripple emulator? So, that means, that means m c and we are only activating this m c because it is just for the MATLAB program, but if you use an FPGA or any digital controller you have to use this Sawtooth waveform right?

So, it can be a simple counter and it will start counting when the clock signal comes. That means that every rising edge of the switching clock; that means when the switch is turned on will keep on incrementing and whenever that means, this clock this ramp output plus the sample current is added when it hit the peak current then it is reset. So, that is why there is a reset signal. So, that means reset will happen when you take the q signal; that means, when the switch goes low that means, this is the gate signal.

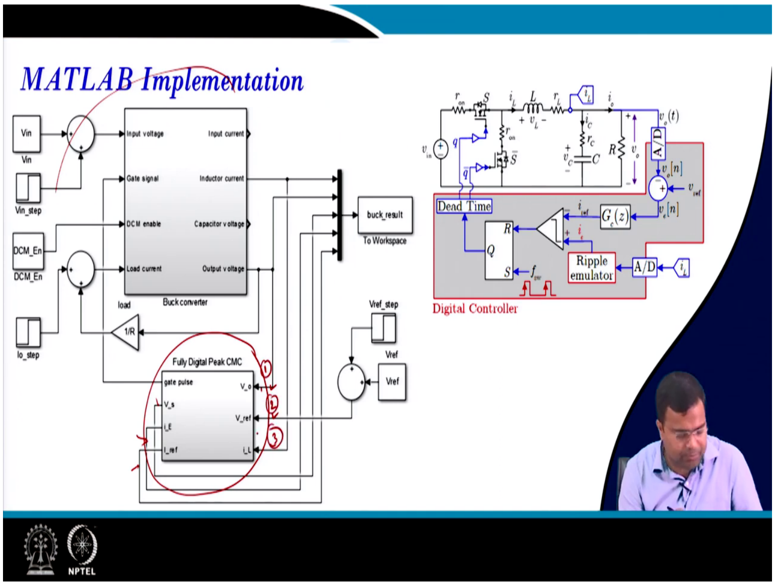
When it goes low that is resetting the theresettingter. So, it will just stop and this is the overall current controller that is used to; that means, and now this ramp here we are using simply integrator. It will integrate you can reset, but in the actual implementation, you can use a counter ok.

(Refer Slide Time: 13:52)



Ripple Emulator (contd…)

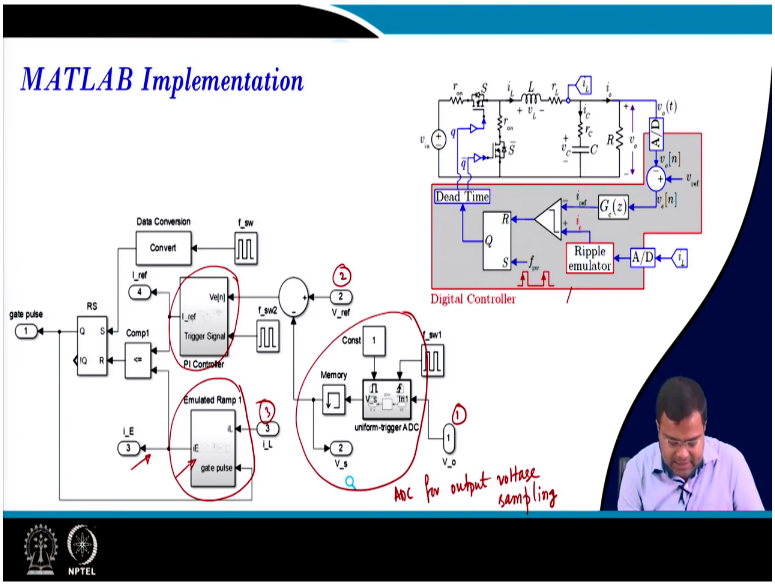
(Refer Slide Time: 13:52)



MATLAB Implementation

So, the actual ripple emulator; so is the overall diagram that realizes our error emulator block. So, this is the fully digital current mode control architecture and we want to go inside; that means, the MATLAB block and that I am going to consider. So, everything else is the same only this controller is different. Here again, we are taking the output voltage either the reference voltage or the inductor current.

(Refer Slide Time: 14:19)

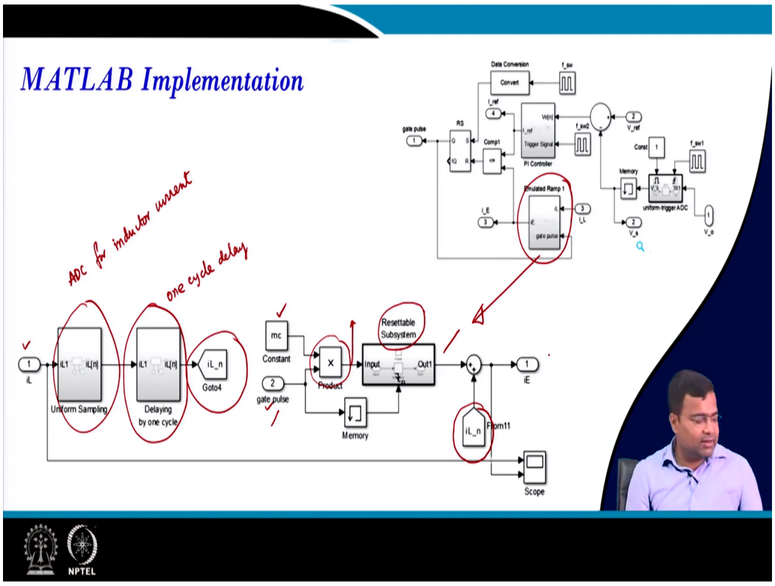


MATLAB Implementation

What we are doing here? And for simulation, we have to verify we want to check what the emulator signal emulated ramp looks like. What does the reference current look like? And you can also see the sample voltage. So, all are captured in the workspace. Next,f we zoom the whole block of this controller.

So, it starts with this you can see them; that means if you take this output voltage this is the number 1 number 2, and number 3 input port. So, the number 1 input port is this. Number 2 is the reference and number 3 is the inductor current. So, this block is our what is this block. This is your ADC for output voltage sampling ok. It is for output voltage sampling, ADC for output voltage sample. Next inductor current emulated ramp as well as the sampling block that should be inside and this is the PI controller ok.

(Refer Slide Time: 15:31)



MATLAB Implementation

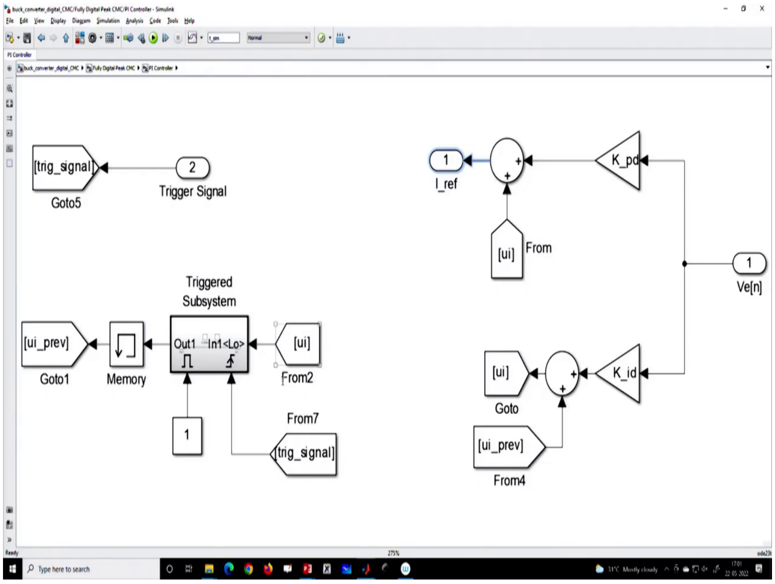
So, if you go to the next this is the inside error because what it is generating is I E which is an emulated current that includes the sample current plus the emulated current and that will be compared to the reference current and the reference current is coming from the PI controller. So, now we want to go inside this block; that means, we are talking about this block, and this block is shown here. What is the input to this block? Input to this block is the number 1.

So, you can see what are the input to this block. That means the input to this block is the inductor current. Yes, this is the inductor current and then the gate pulse. So, gate pulse these are 2 inputs ok next. So, this is the first conversion; that means, this is the ADC for inductor current and this is an event-triggered ADC and this is used for 1 cycle delay and now this is our sample inductor current. Once we get the sampled inductor current and this is our Sawtooth slope m c.

So, we can set or whatever we want to use that you can customize during runtime as well as you can customize in the dot m file. Then when the gate pulse is high only this will come and when this comes this is a scaling factor; that means, the output of this block will be m c when the gate pulse is high and this will be used for the resettable subsystem which has an integrator and I will go to the MATLAB block to show.

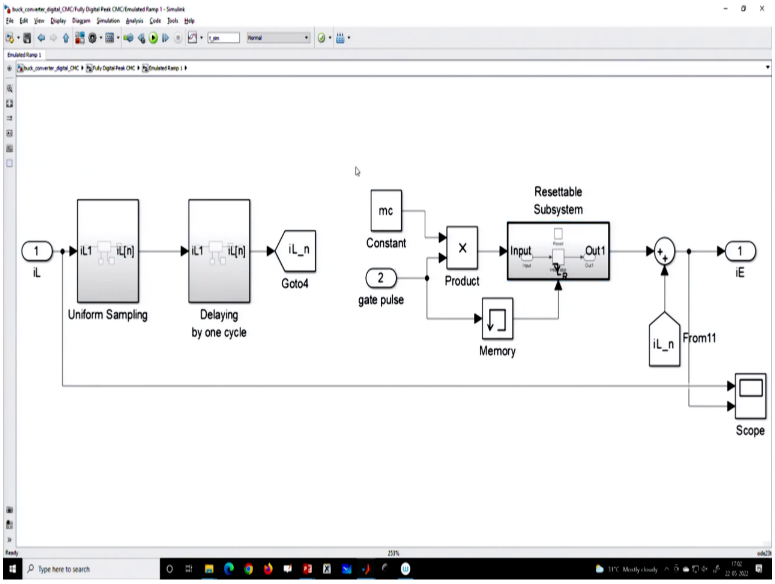
So, now this will generate our Sawtooth edge; that means, a continuous waveform with which we have to add the sampled inductor current sample plus delay. That means, the 1-cycle delay is the sample inductor current, and then the overall current is the emulated current.

(Refer Slide Time: 17:51)

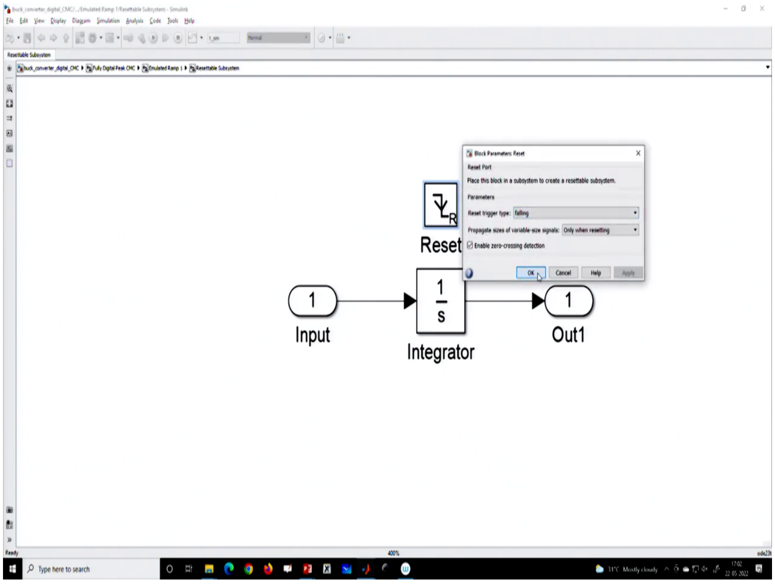

buck_converter_digital_CMC/Fully Digital Peak CMC/PI Controller - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
t_sim
Normal
PI Controller
buck_converter_digital_CMC ▸ Fully Digital Peak CMC ▸ PI Controller ▸
[trig_signal]
Goto5
2
Trigger Signal
1
I_ref
+
+
K_pd
[ui]
From
1
Ve[n]
Triggered
Subsystem
[ui_prev]
Goto1
Memory
Out1
In1<Lo>
[ui]
From2
1
From7
[trig_signal]
[ui]
Goto
+
+
K_id
[ui_prev]
From4
Type here to search

So, if you go to the MATLAB let us go to the MATLAB. Here you can see this is a controller block emulated ramp block that we are discussing.

(Refer Slide Time: 17:57)


buck_converter_digital_CMC/Fully Digital Peak CMC/Emulated Ramp 1 - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
t_sim
Normal
Emulated Ramp 1
buck_converter_digital_CMC ▸ Fully Digital Peak CMC ▸ Emulated Ramp 1 ▸
1
iL
iL1
iL[n]
Uniform Sampling
iL1
iL[n]
Delaying
by one cycle
iL_n
Goto4
mc
Constant
2
gate pulse
×
Product
Memory
Resettable
Subsystem
Input
Out1
+
+
1
iE
iL_n
From11
Scope
Type here to search

(Refer Slide Time: 17:59)


buck_converter_digital_CMC/.../Emulated Ramp 1/Resettable Subsystem - Simulink

buck_converter_digital_CMC ▸ Fully Digital Peak CMC ▸ Emulated Ramp 1 ▸ Resettable Subsystem

Block Parameters: Reset

Reset Port

Place this block in a subsystem to create a resettable subsystem.

Parameters

Reset trigger type: falling

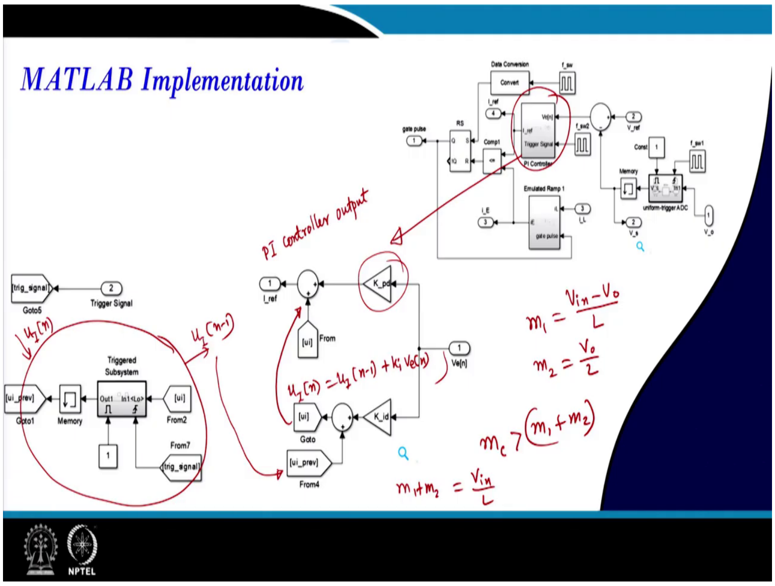
Propagate sizes of variable-size signals: Only when resetting

☑ Enable zero-crossing detection

So, if you go to the resettable block, there is a resettable integrator. That means, the integrator generally if it is reset otherwise the integrator will keep on increasing ok. And who will reset the integrator? The integrator reset happens; that means, you know based on the gate pulse. That means, the gate pulse has to there will be an edge you know is the edge detection circuit, and when you detect this edge.

So, this is an edge resettable edge. So, it will detect the falling edge and that falling edge is used to reset the integrator that is it. Next, the constant is the slope of the current that you want and output and this is the A to D converter for sampling ADC. This is for 1 cycle delay and this is added here. So, you can see this block and this block are the same go-to block and the form block and this will generate our actual emulated ramp. And I want to show you all the waveform, but before that, I want t,o explain the remaining part.

(Refer Slide Time: 19:03)

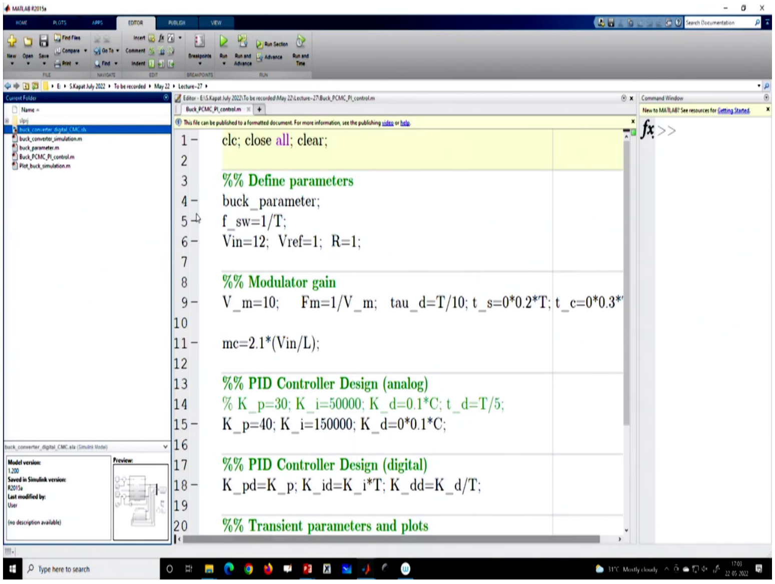


MATLAB Implementation

Next, this is a PI controller. So, I am talking about this block. This block is shown here and we have discussed is the same as mixed-signal current mode control, where it is a PI controller and it is a clock-driven PI controller which is implemented using a difference equation.

And again this is used to generate what? If the input is u I of n then the output will be u I of n minus 1 and that is actually added here and then this u i of n is generated which is u i of n minus 1 plus k i into v e n and this is a v n and this is your proportional controller and this output goes here by this block and this is overall. So, this is a PI controller output ok.

(Refer Slide Time: 20:13)



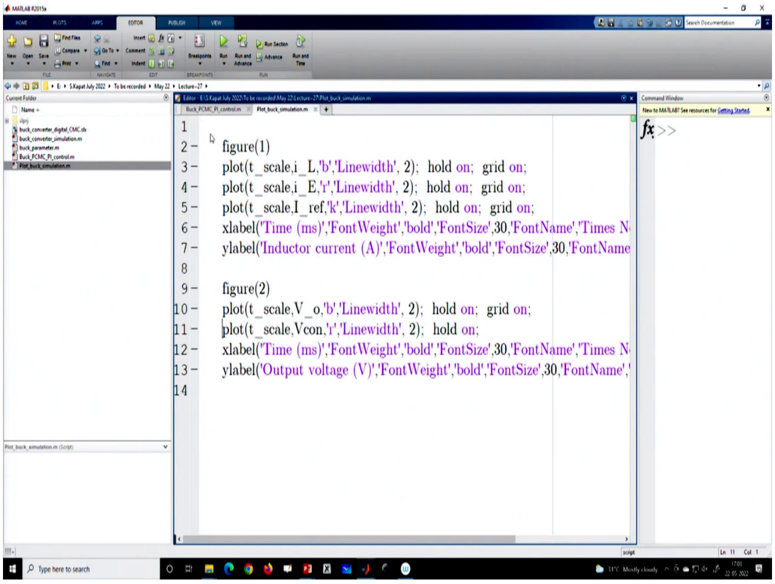
Editor - E:\S.Kapat July 2022\To be recorded\May 22\Lecture-27\Buck_PCMC_PI_control.m

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

```
clc; close all; clear;

%% Define parameters
buck_parameter;
f_sw=1/T;
Vin=12;  Vref=1;  R=1;

%% Modulator gain
V_m=10;     Fm=1/V_m;   tau_d=T/10; t_s=0*0.2*T; t_c=0*0.3*

mc=2.1*(Vin/L);

%% PID Controller Design (analog)
% K_p=30; K_i=50000; K_d=0.1*C; t_d=T/5;
K_p=40; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots
```

So, now let us go to the MATLAB and try to understand. So, we will simulate a few case studies.

(Refer Slide Time: 20:18)

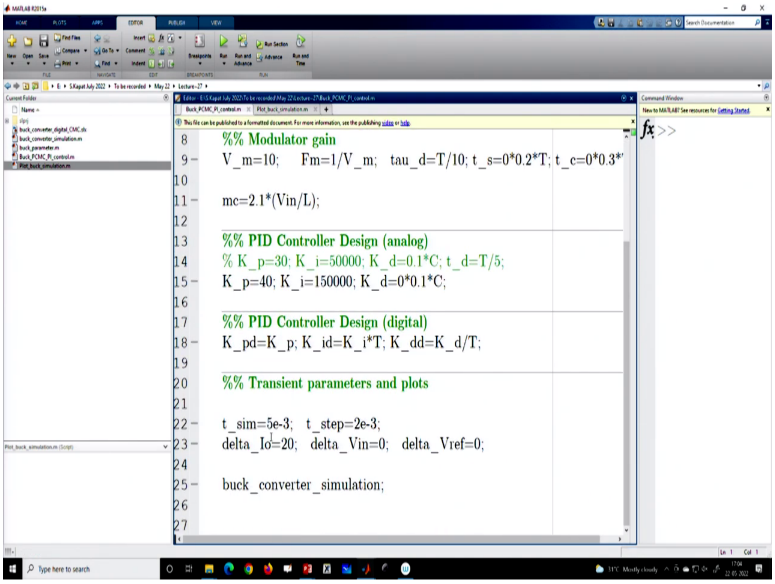


Editor - E:\S.Kapat July 2022\To be recorded\May 22\Lecture-27\Plot_buck_simulation.m

```

figure(1)
plot(t_scale,i_L,'b','Linewidth', 2);  hold on;  grid on;
plot(t_scale,i_E,'r','Linewidth', 2);  hold on;  grid on;
plot(t_scale,I_ref,'k','Linewidth', 2);  hold on;  grid on;
xlabel('Time (ms)','FontWeight','bold','FontSize',30,'FontName','Times N
ylabel('Inductor current (A)','FontWeight','bold','FontSize',30,'FontName

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

So, in this, I want to first you know if I go to the MATLAB I want to show a peak current mode control with a delay.

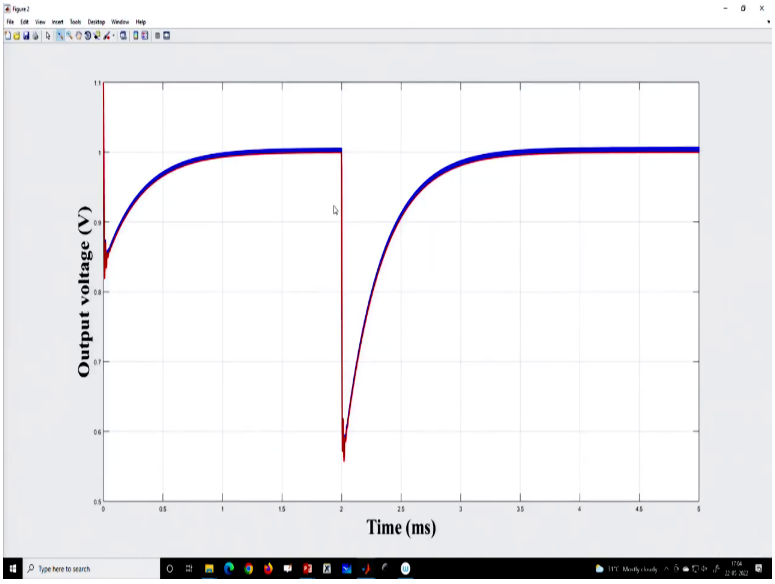
(Refer Slide Time: 20:25)



```
%% Modulator gain
V_m=10;     Fm=1/V_m;   tau_d=T/10; t_s=0*0.2*T; t_c=0*0.3*

mc=2.1*(Vin/L);

%% PID Controller Design (analog)
% K_p=30; K_i=50000; K_d=0.1*C; t_d=T/5;
K_p=40; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots

t_sim=5e-3;   t_step=2e-3;
delta_Io=20;   delta_Vin=0;   delta_Vref=0;

buck_converter_simulation;
```

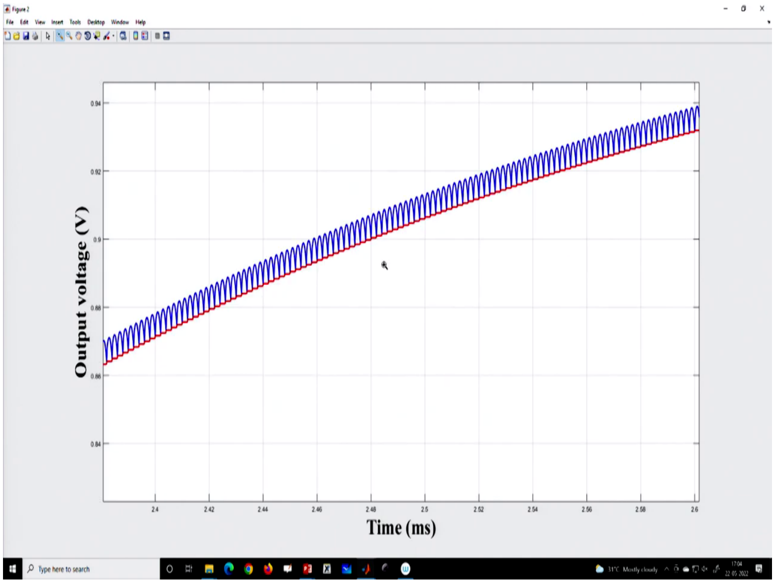
Where we are using again a low transient and we are not using a very large gain. So, let us first run the simulation and I want to show you all the signals ok.

(Refer Slide Time: 20:37)

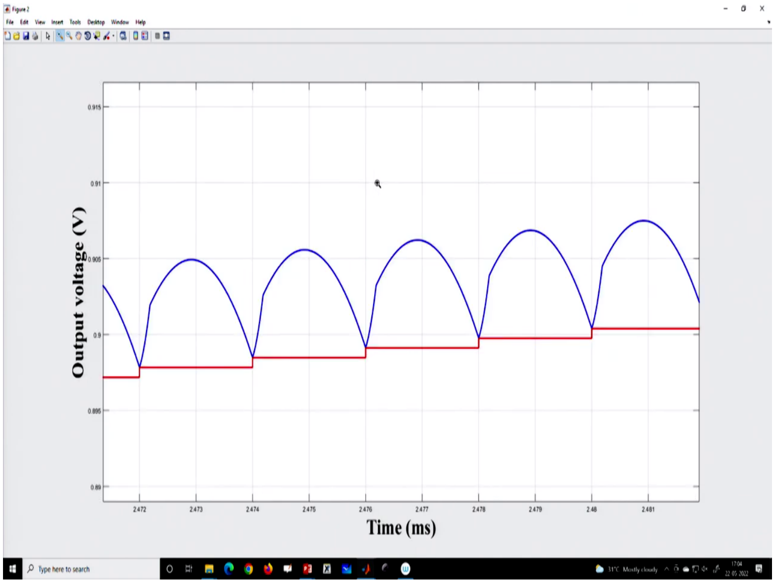


So, first I want to show the output voltage waveform.

(Refer Slide Time: 20:44)



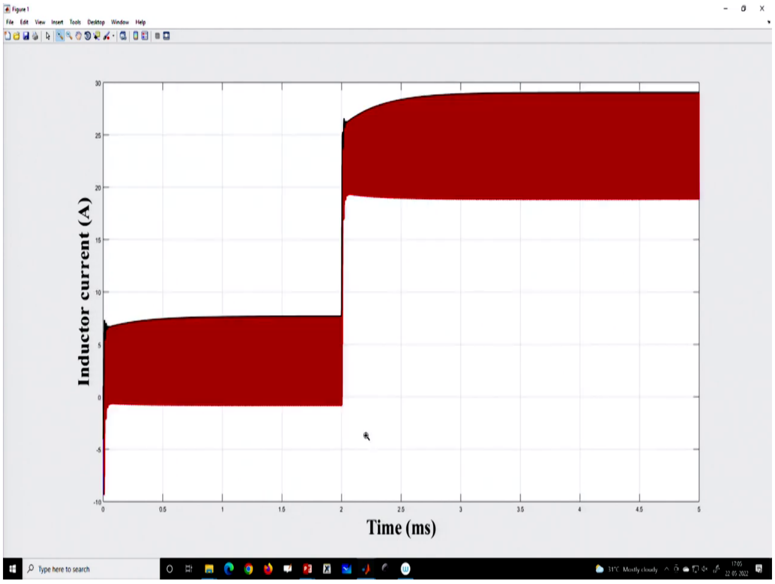
(Refer Slide Time: 20:45)



And if you see the output voltage waveform you can see. So, here I am not showing any delayed version of the output voltage. Because if you add delay you can add it you can customize it based on the requirement, but I want to first try to show the basic implementation of fully digital current mode control. Where you there is no need to consider a delay. So, you can assume that the output voltage is sampled exactly at the switching point or just below before that.

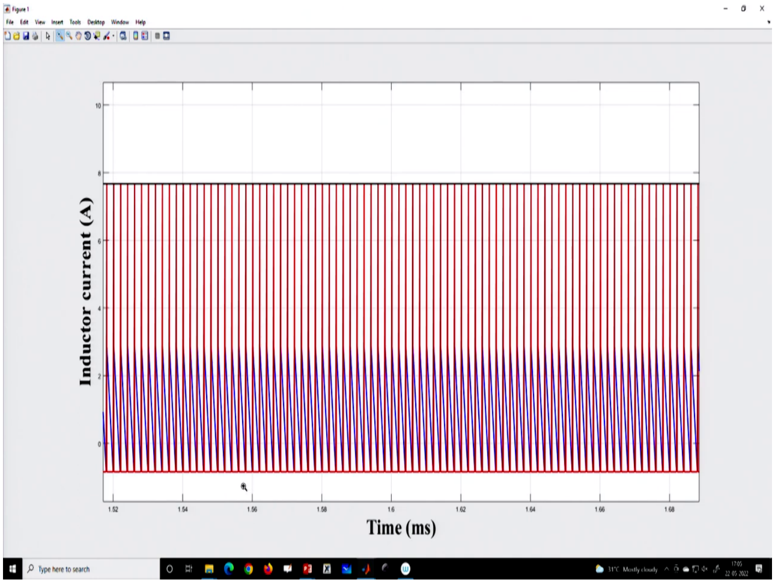
But one can go because if you go to delay, there will be a more advanced level. The analysis will be very complicated and is a research topic.

(Refer Slide Time: 21:23)

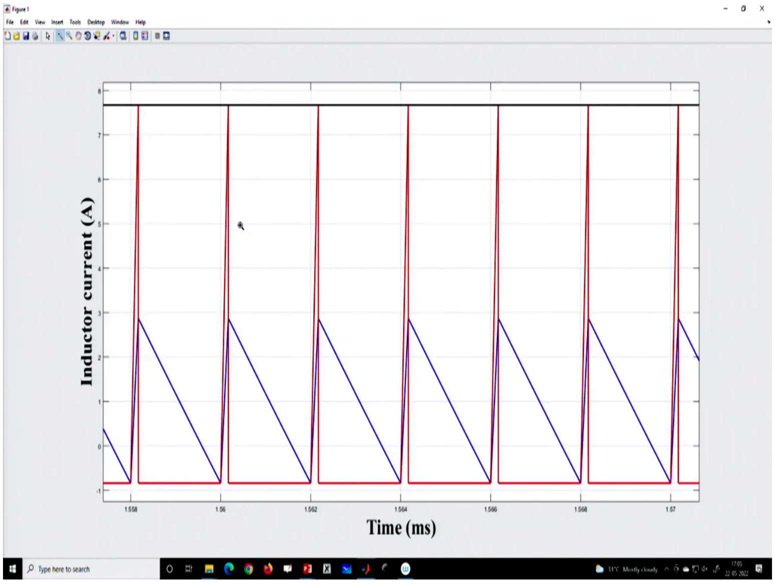


So, in the next part, I want to show the current waveform. So, let us go to the current waveform.

(Refer Slide Time: 21:30)

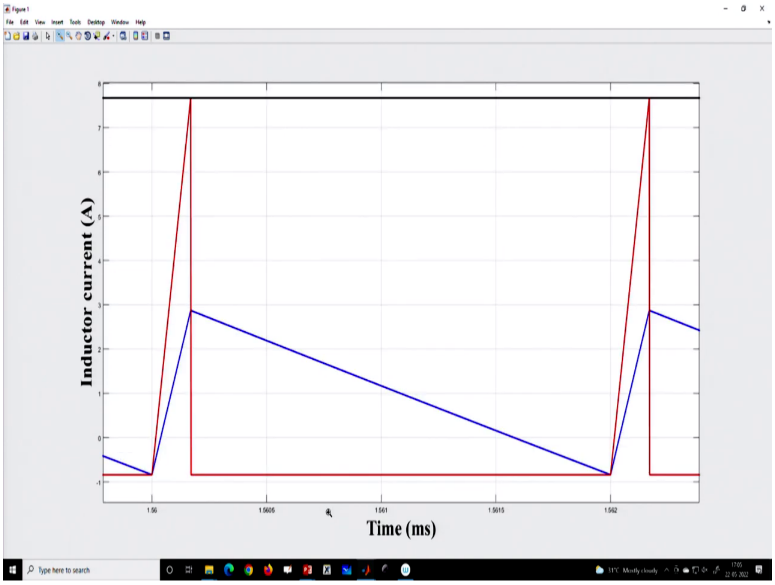


(Refer Slide Time: 21:32)



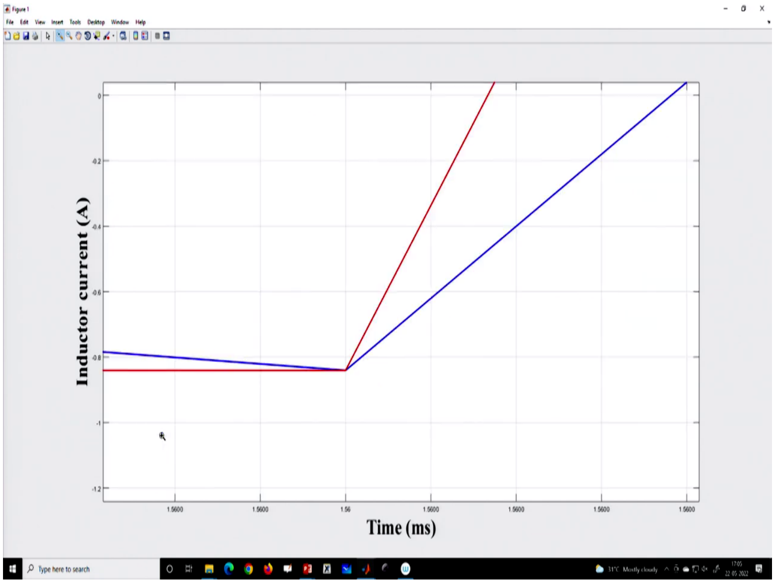
If you go to the current waveform the first thing I will show you now is that if I take one complete cycle.

(Refer Slide Time: 21:36)



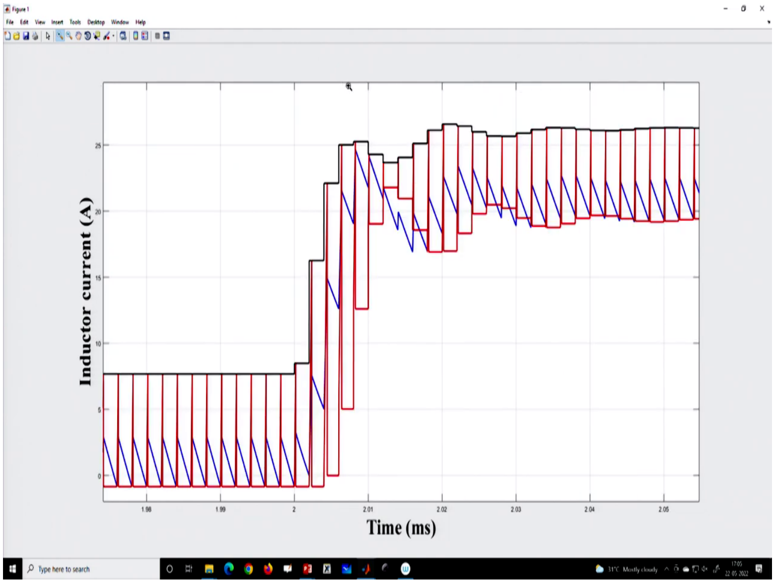
So, you can see the blue one is the actual inductor current which is like a sense inductor current and the red one is an emulated slope; that means, which is much higher. Otherwise, it will be unstable, I will show you.

(Refer Slide Time: 21:56)



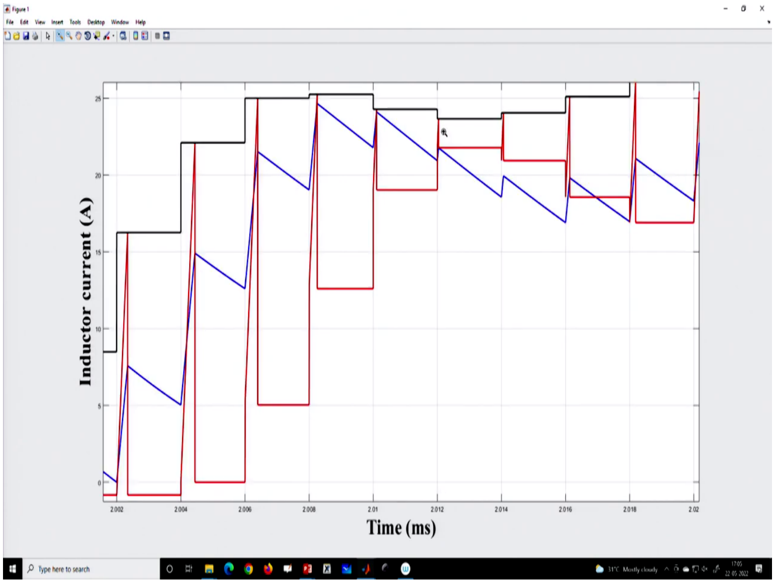
And after adding the sample with the emulated slope you can see here under steady state when it reaches under steady state since the valley point is exactly like repeating every cycle. Then it will look like as if like a sample and then added the ramp.

(Refer Slide Time: 22:15)



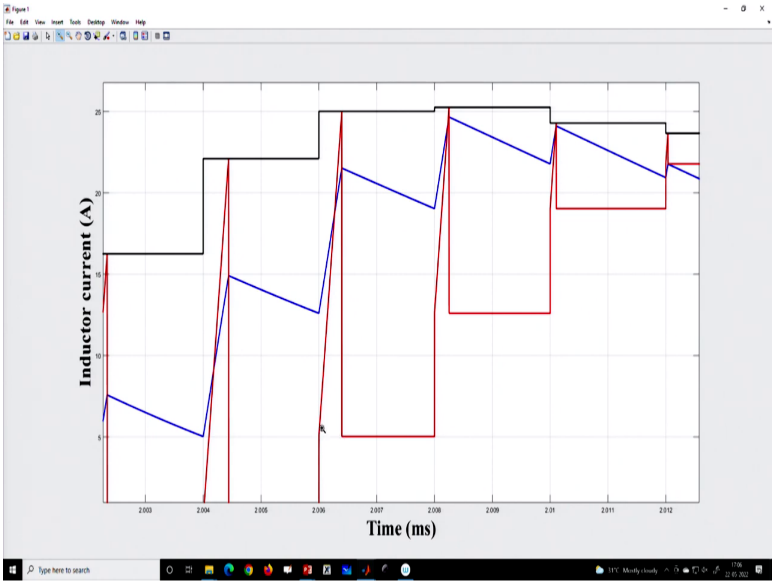
But I will show you the transient case also. But if you take a transient case I am just showing I just want to emphasize.

(Refer Slide Time: 22:18)



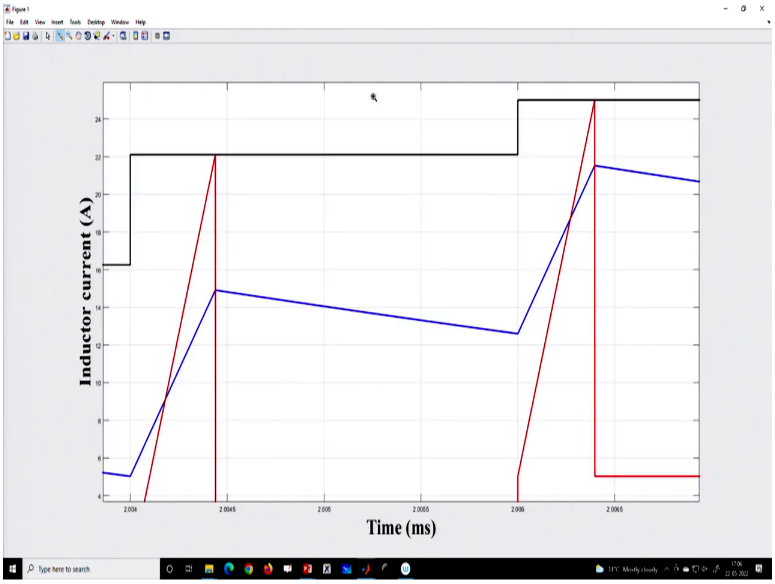
You can see that let us first consider maybe here yeah.

(Refer Slide Time: 22:26)



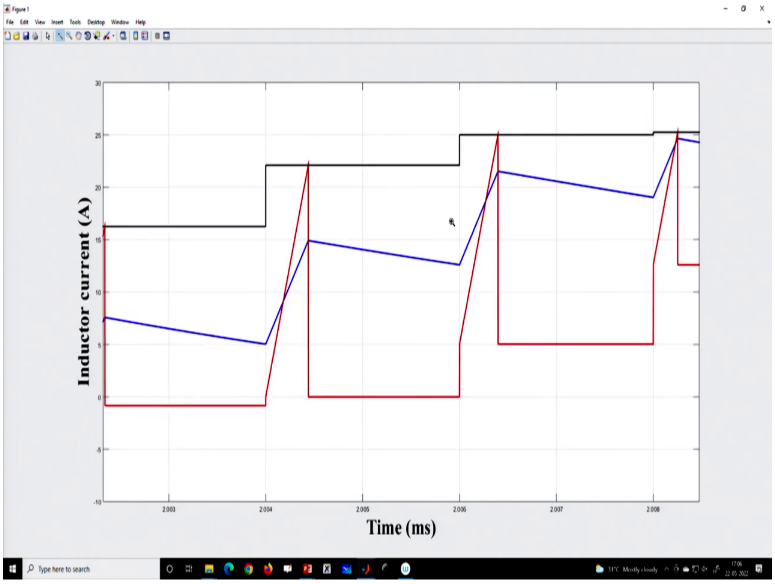
You can see this point this is the inductor current for this cycle. For this cycle, the valley point of the inductor current is here. So, you can interpolate and this is a valuable point.

(Refer Slide Time: 22:42)



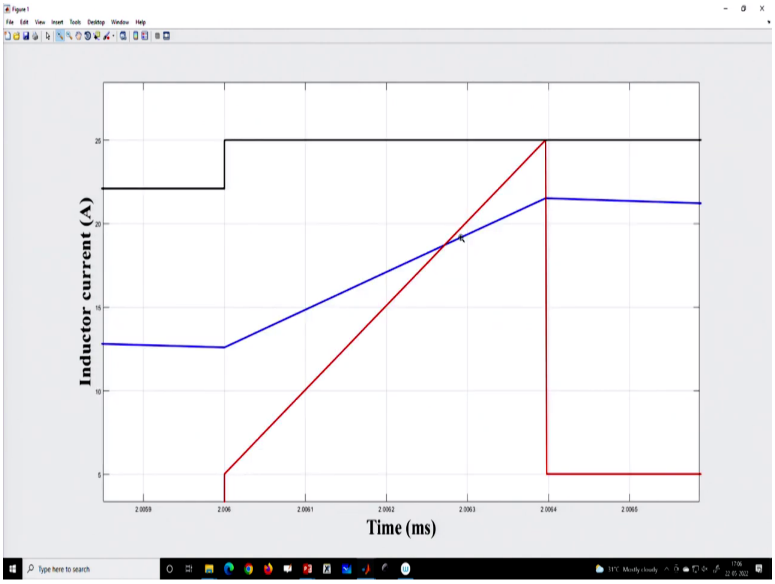
That means, your ramp is added which means if I further zoom this cycle this point is interpolated here, and then we have added this slope.

(Refer Slide Time: 22:49)



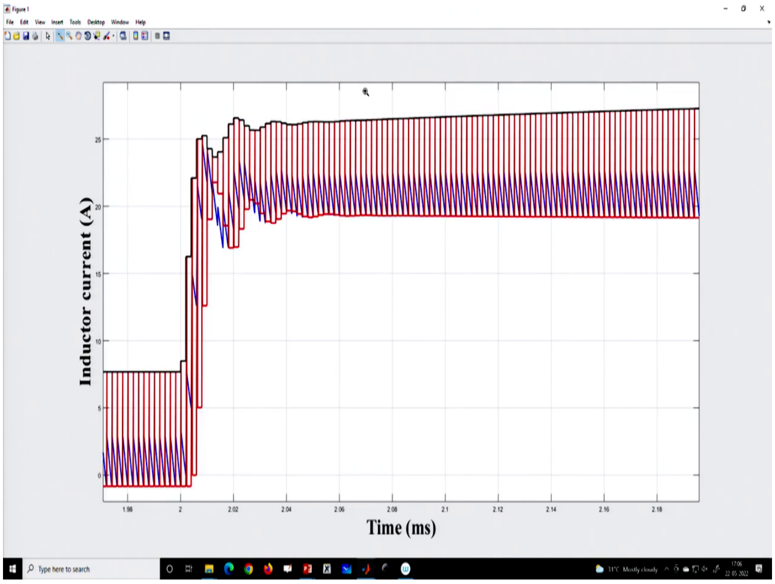
That means if you carefully observe we are making 1 cycle delay. This is your actual inductor current and its sample is used in the next cycle.

(Refer Slide Time: 23:05)

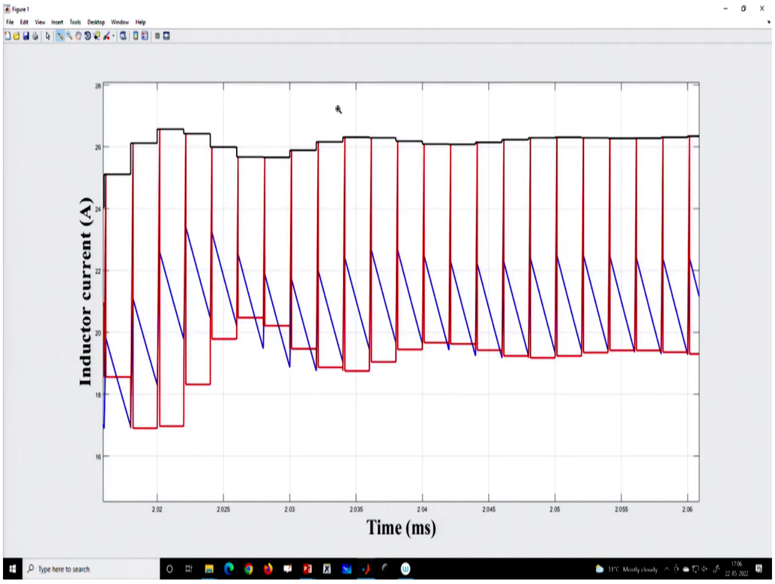


And then we added that m c and after m c this is an emulated ramp plus sample current when the overall current is compared with the peak current. When it hit the peak current the switch is turned off and that is exactly what we have explained.

(Refer Slide Time: 23:19)

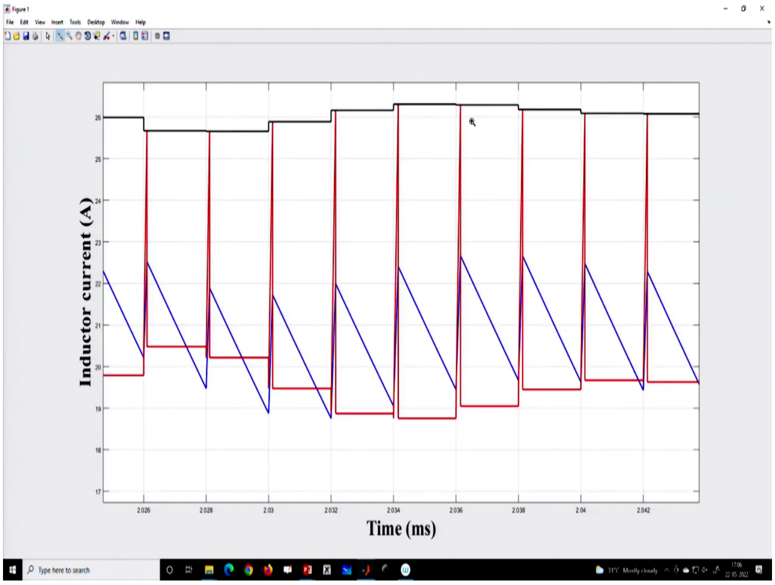


(Refer Slide Time: 23:22)



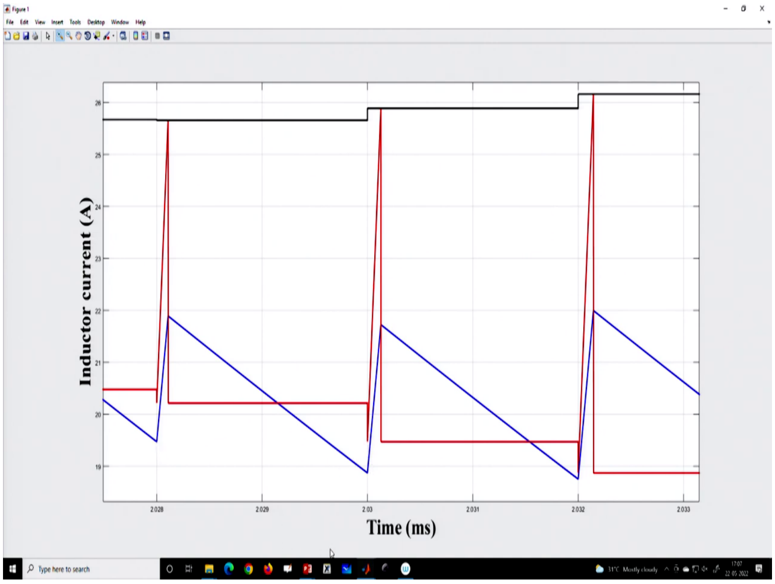
That means if you try to understand this part.

(Refer Slide Time: 23:24)



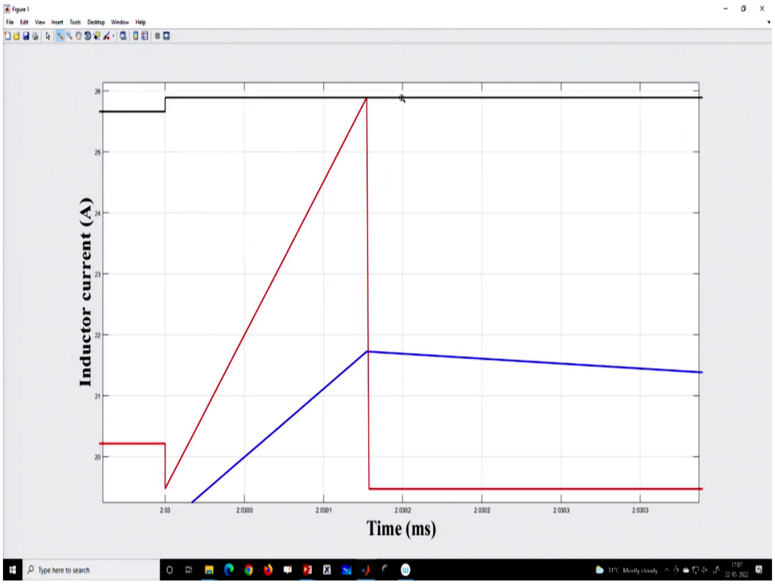
So, if you go to this block take a few cycles.

(Refer Slide Time: 23:27)



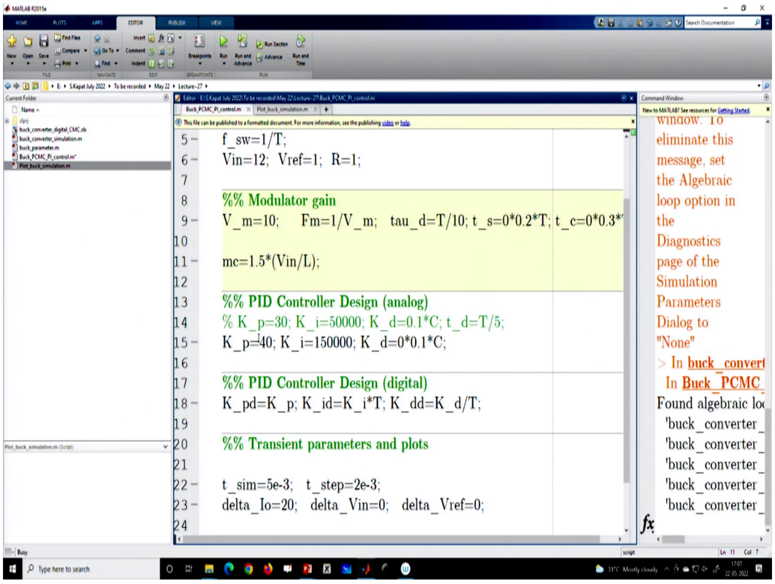
So, the inductor current in this sample is used here.

(Refer Slide Time: 23:34)



And we have taken a very large ramp, so otherwise, it will be unstable.

(Refer Slide Time: 23:42)

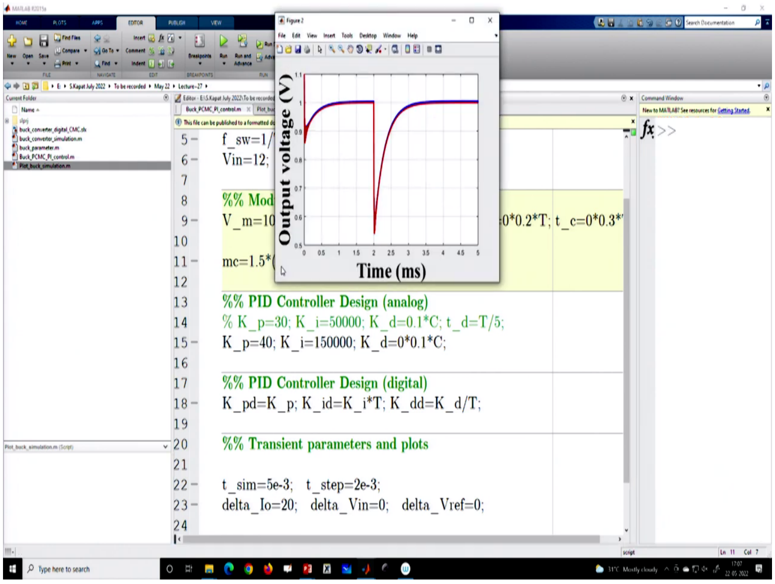


```
f_sw=1/T;
Vin=12;  Vref=1;  R=1;

%% Modulator gain
V_m=10;    Fm=1/V_m;   tau_d=T/10; t_s=0*0.2*T; t_c=0*0.3*

mc=1.5*(Vin/L);

%% PID Controller Design (analog)
% K_p=30; K_i=50000; K_d=0.1*C; t_d=T/5;
K_p=40; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots

t_sim=5e-3;  t_step=2e-3;
delta_Io=20;  delta_Vin=0;  delta_Vref=0;
```

```
window. To
eliminate this
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_convert
  In Buck_PCMC_
Found algebraic lo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
```

So, let us see if it is 2.1 type. So, if we use a ramp let us say 1.5.

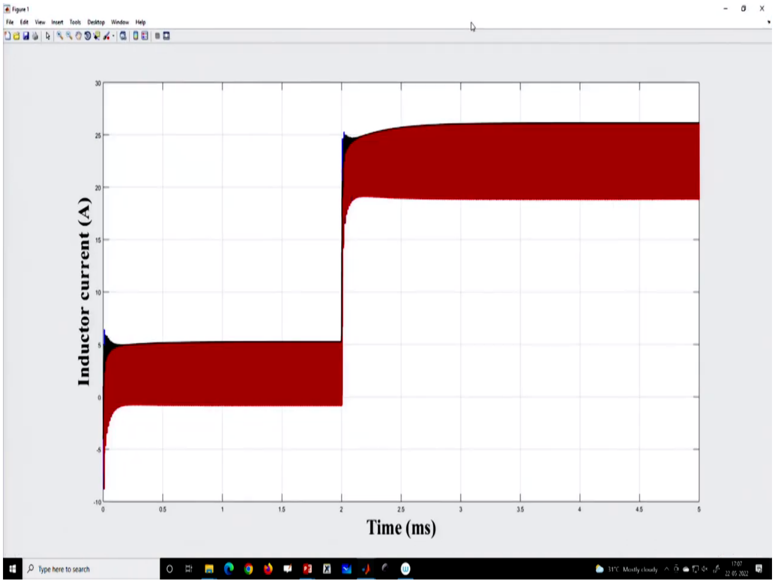
(Refer Slide Time: 23:49)



```
f_sw=1/
Vin=12;

%% Mod
V_m=10                                            0*0.2*T; t_c=0*0.3*

mc=1.5*(

%% PID Controller Design (analog)
% K_p=30; K_i=50000; K_d=0.1*C; t_d=T/5;
K_p=40; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots

t_sim=5e-3;  t_step=2e-3;
delta_Io=20;  delta_Vin=0;  delta_Vref=0;
```

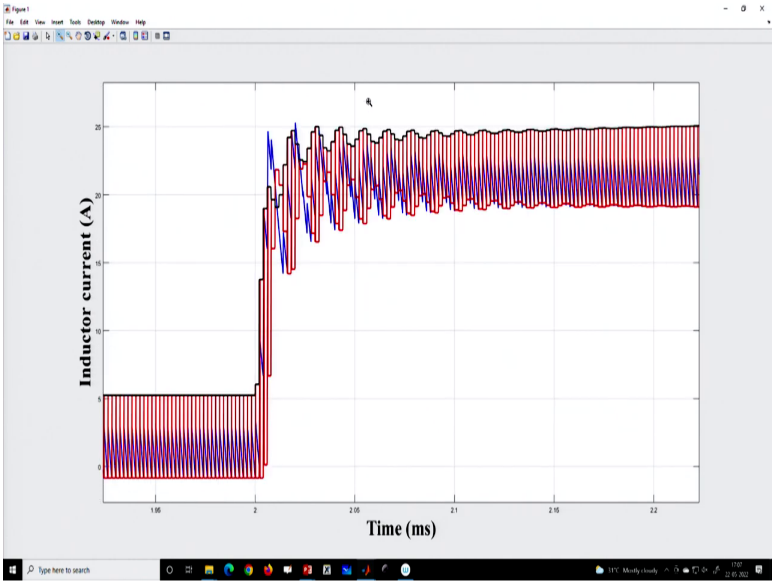
And we want to see it is the Vin by L.

(Refer Slide Time: 23:51)



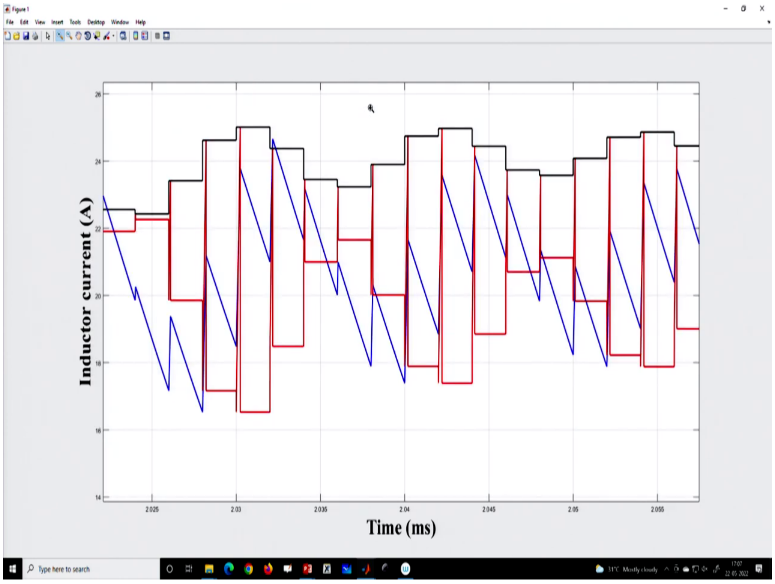
You know we have taken a more conservative slope much more conservative slope.

(Refer Slide Time: 23:56)

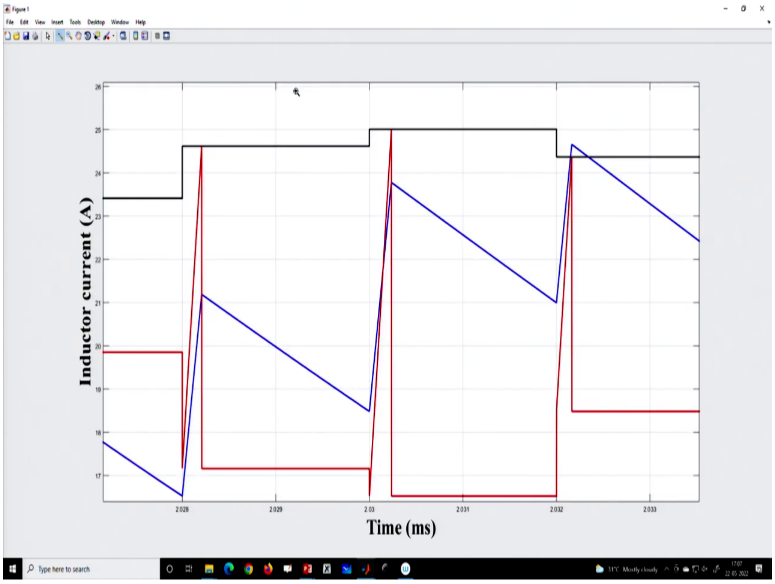


So, although it is stable now.

(Refer Slide Time: 23:59)

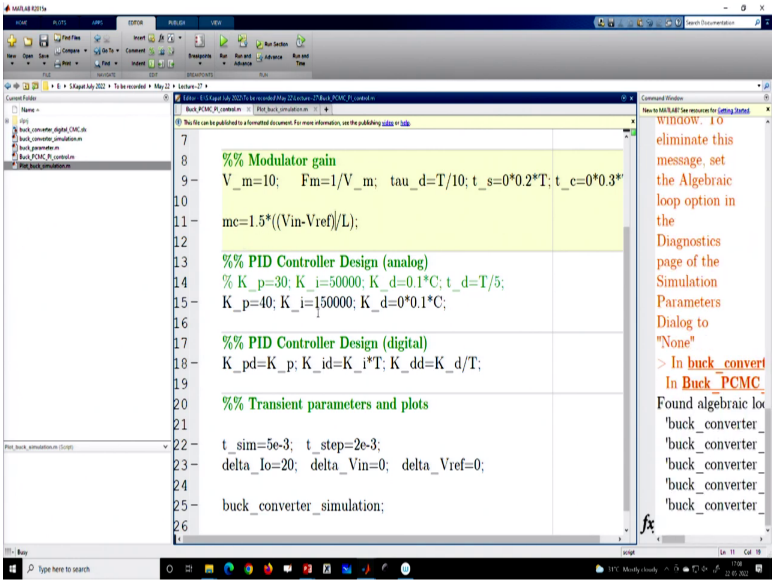


(Refer Slide Time: 24:01)



The slope of this is even higher, but again this operation I am showing. Now suppose if I want to say let us say what is the rising slope of the inductor current?

(Refer Slide Time: 24:14)



Editor - E:\S.Kapat July 2022\To be recorded\May 22\Lecture-27\Buck_PCMC_PI_control.m

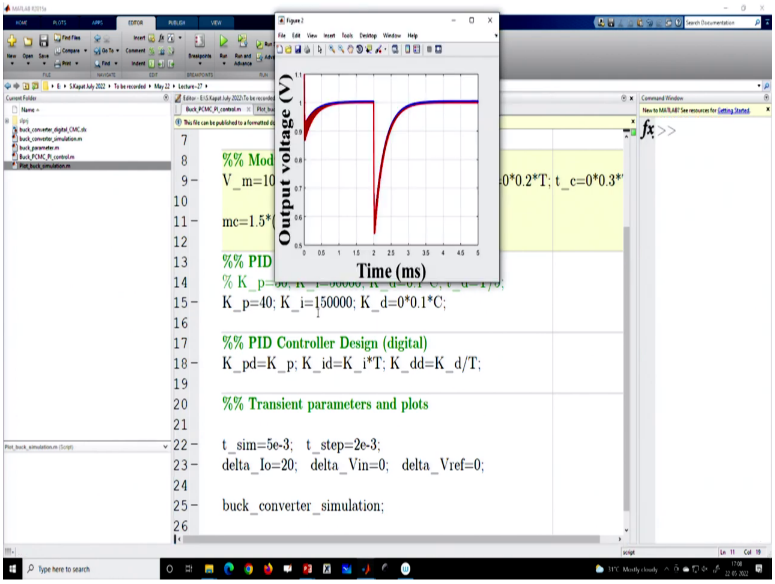
```
%% Modulator gain
V_m=10;    Fm=1/V_m;   tau_d=T/10; t_s=0*0.2*T; t_c=0*0.3*'

mc=1.5*((Vin-Vref)/L);

%% PID Controller Design (analog)
% K_p=30; K_i=50000; K_d=0.1*C; t_d=T/5;
K_p=40; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots

t_sim=5e-3;  t_step=2e-3;
delta_Io=20;  delta_Vin=0;  delta_Vref=0;

buck_converter_simulation;
```

Command Window

```
window. To
eliminate this
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_convert
  In Buck_PCMC_
Found algebraic lo
 'buck_converter_
 'buck_converter_
 'buck_converter_
 'buck_converter_
 'buck_converter_
```

It is something like a V in minus let us say V ref. So, let me check yeah it is V ref.

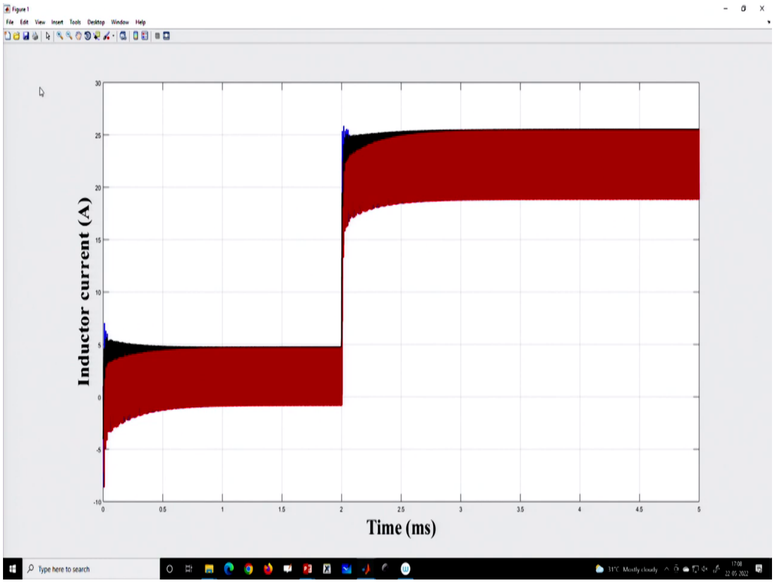
(Refer Slide Time: 24:36)



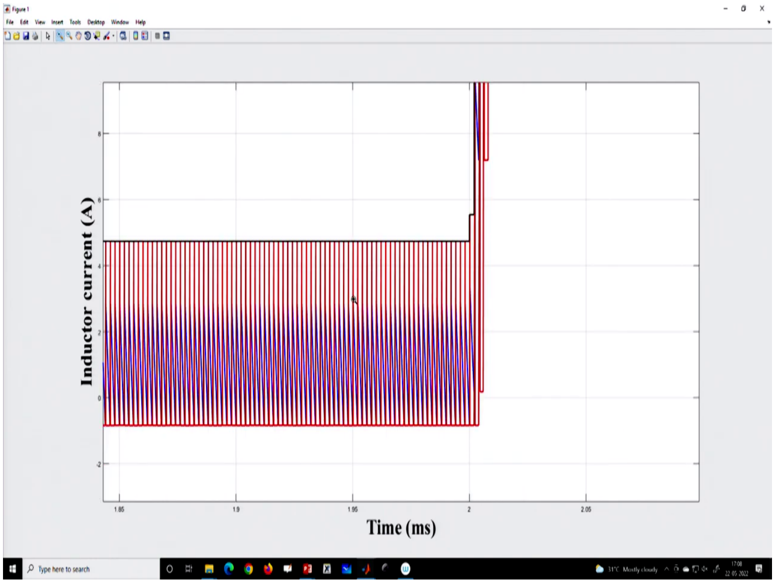
```
%% Mod
V_m=10                               0*0.2*T; t_c=0*0.3*'

mc=1.5*(

%% PID
% K_p=
K_p=40; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots

t_sim=5e-3;  t_step=2e-3;
delta_Io=20;  delta_Vin=0;  delta_Vref=0;

buck_converter_simulation;
```

So, suppose I am taking just V in my inductor rising slope 1.5 times. Will it work? Let us see.

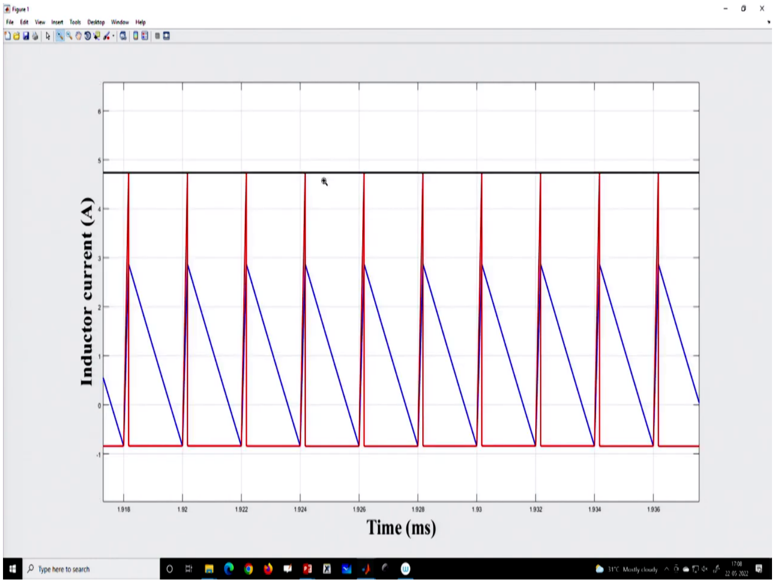
(Refer Slide Time: 24:37)



(Refer Slide Time: 24:42)

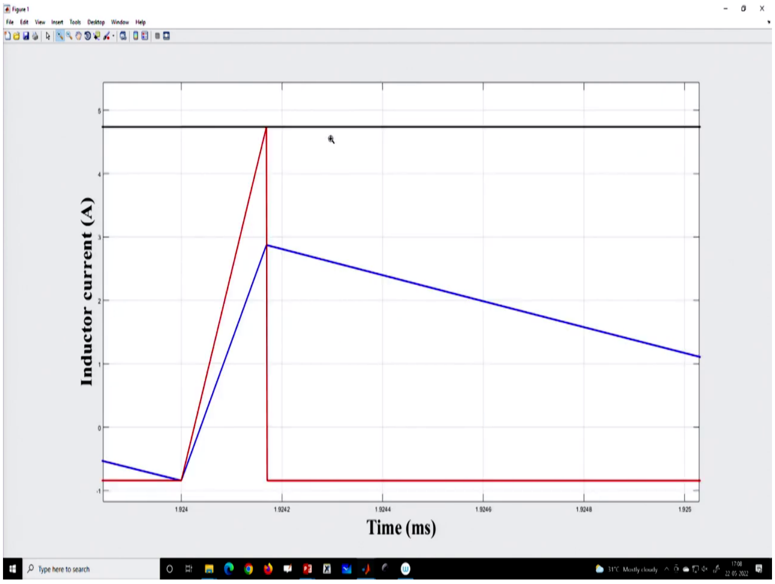


(Refer Slide Time: 24:43)

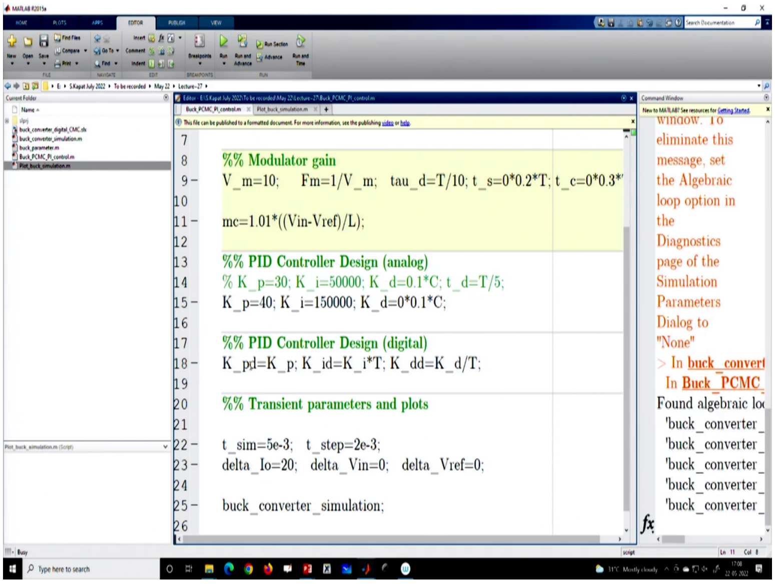


It is still working, but it is slowly going into sorry it is still working; that means, we have used the slope.

(Refer Slide Time: 24:47)



(Refer Slide Time: 24:57)

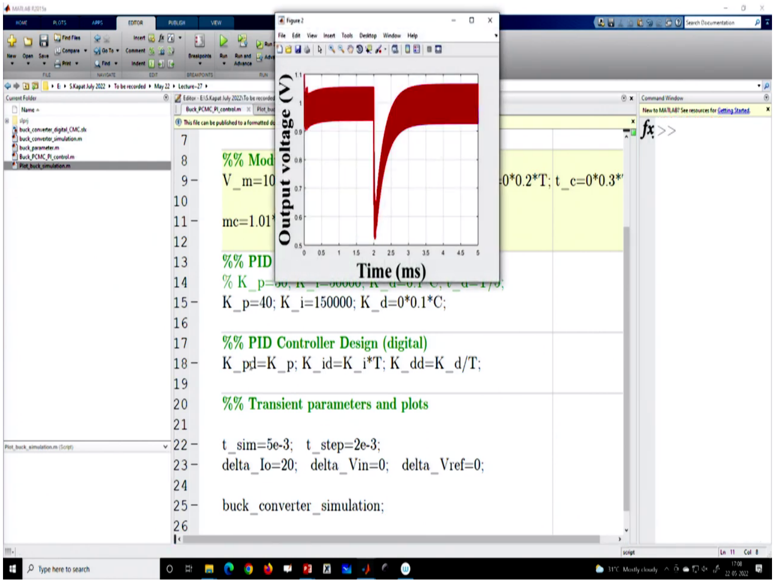


```
%% Modulator gain
V_m=10;    Fm=1/V_m;   tau_d=T/10; t_s=0*0.2*T; t_c=0*0.3*

mc=1.01*((Vin-Vref)/L);

%% PID Controller Design (analog)
% K_p=30; K_i=50000; K_d=0.1*C; t_d=T/5;
K_p=40; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots

t_sim=5e-3;   t_step=2e-3;
delta_Io=20;   delta_Vin=0;   delta_Vref=0;

buck_converter_simulation;
```

```
window. To
eliminate this
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_convert
  In Buck_PCMC_
Found algebraic loo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
```

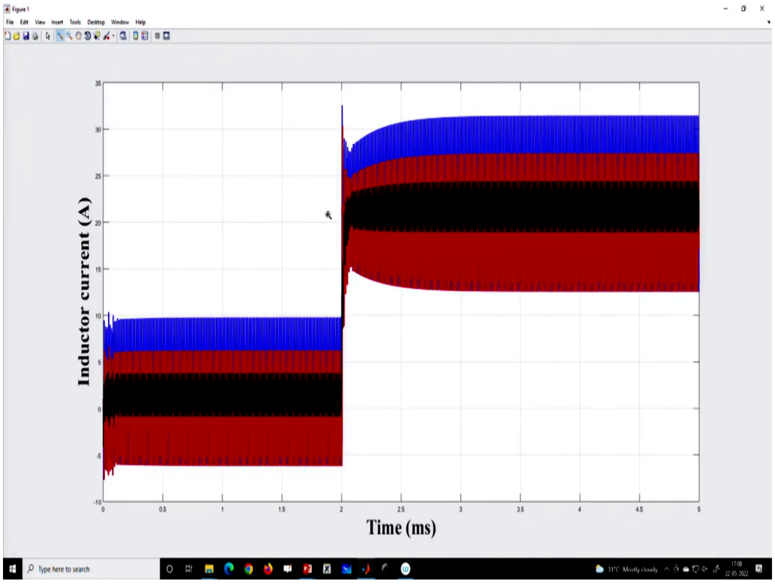
If you see just 1.5 times, but suppose we make it just the same or maybe 0.01 just slightly above it must be unstable that I can show.

(Refer Slide Time: 25:03)



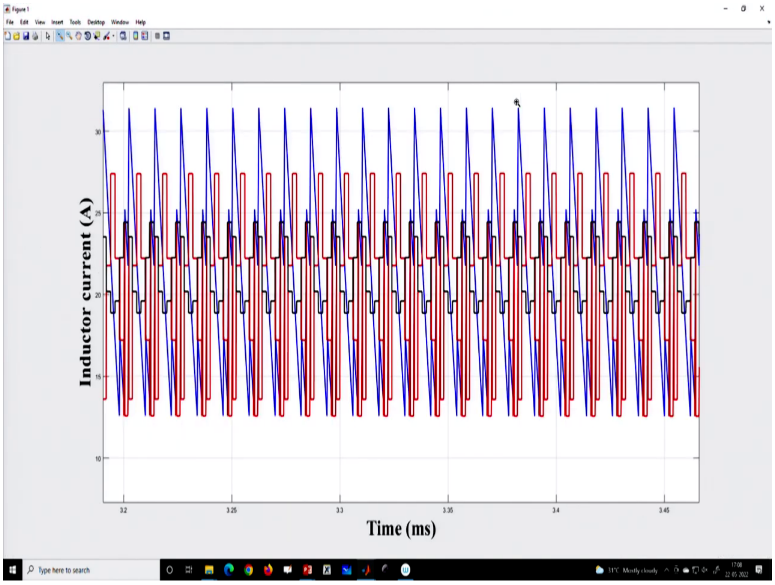
```
%% Mod
V_m=10                                        0*0.2*T; t_c=0*0.3*

mc=1.01*

%% PID
% K_p
K_p=40; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots

t_sim=5e-3;   t_step=2e-3;
delta_Io=20;   delta_Vin=0;   delta_Vref=0;

buck_converter_simulation;
```

(Refer Slide Time: 25:05)



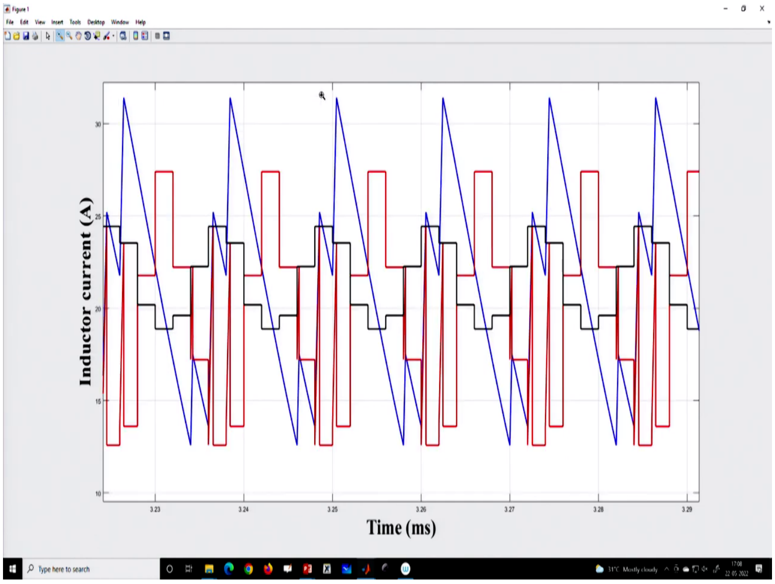
And it is unstable. You can see there is a subharmonic instability that is coming.

(Refer Slide Time: 25:09)



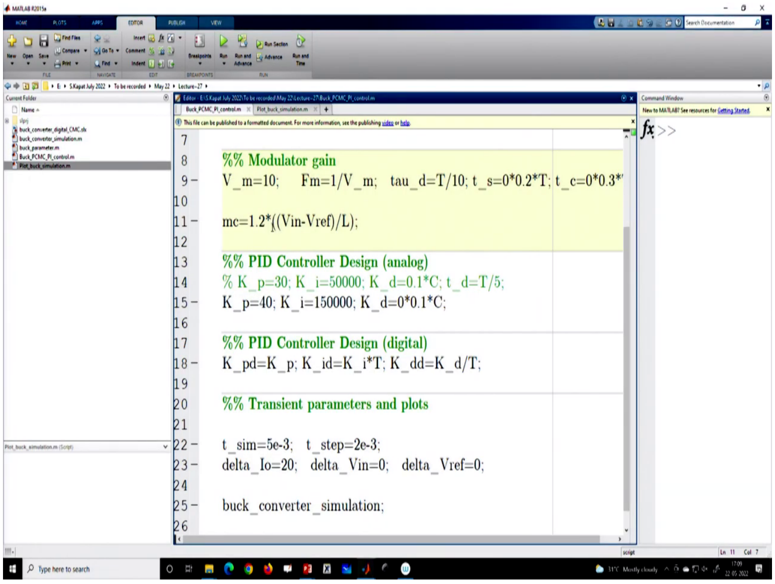
Because the current loop is unstable.

(Refer Slide Time: 25:11)



And when you go to the analysis part it will be shown that you need this is nothing, but m 1 plus m 2 plus; that means, the m c should be greater than the sum of m 1 plus m 2 m 1 is the rising slope.

(Refer Slide Time: 25:29)

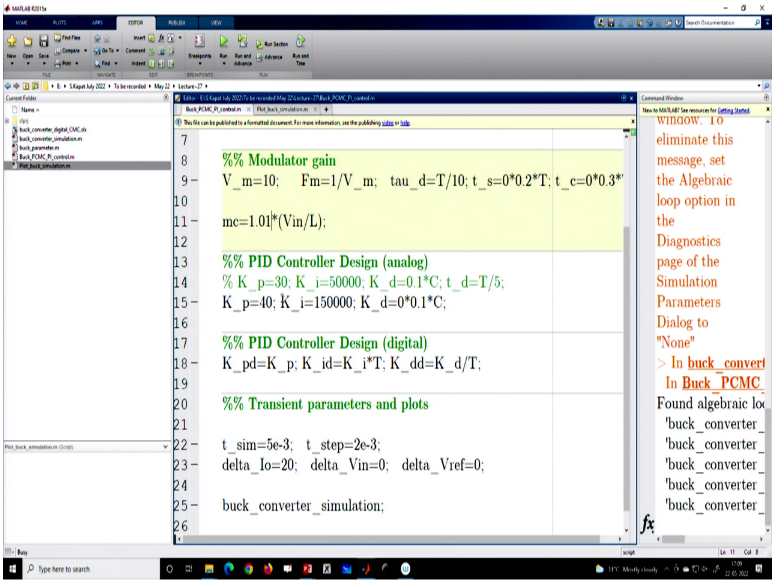


```
%% Modulator gain
V_m=10;    Fm=1/V_m;   tau_d=T/10; t_s=0*0.2*T; t_c=0*0.3*

mc=1.2*((Vin-Vref)/L);

%% PID Controller Design (analog)
% K_p=30; K_i=50000; K_d=0.1*C; t_d=T/5;
K_p=40; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots

t_sim=5e-3;   t_step=2e-3;
delta_Io=20;   delta_Vin=0;   delta_Vref=0;

buck_converter_simulation;
```

That means if you go to let us say what is; that means, that will be discussed in the theory; that means, I am saying the m c should be greater than m 1 plus m 2. And in a buck converter, what is m 1? It is V in minus V 0 by L.

And what is m 2? It is V 0 by L. Then what is m 1 plus m 2? It is simply V in by L. So, that is why we have taken. If we just m c is above V in by L, then it should be stable and we are going here.

(Refer Slide Time: 26:06)



Editor - E:\S.Kapat July 2022\To be recorded\May 22\Lecture-27\Buck_PCMC_PI_control.m

Buck_PCMC_PI_control.m | Plot_buck_simulation.m

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

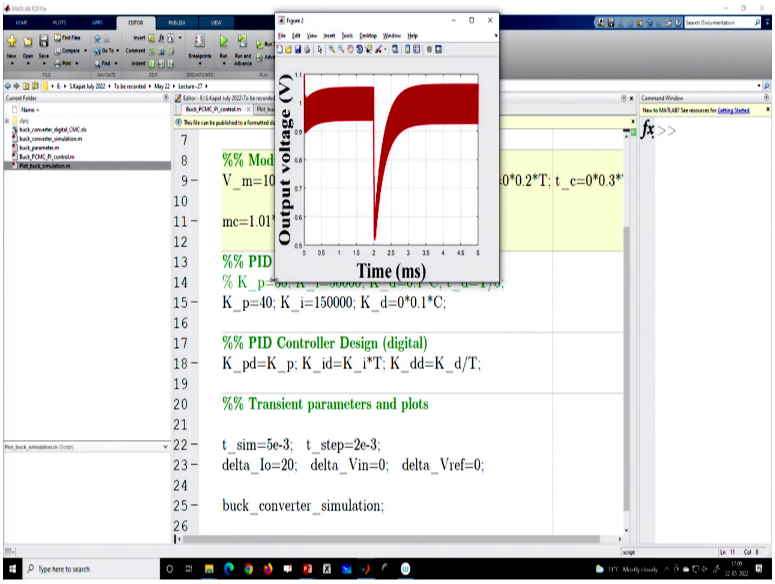
```
%% Modulator gain
V_m=10;    Fm=1/V_m;   tau_d=T/10; t_s=0*0.2*T; t_c=0*0.3*

mc=1.01*(Vin/L);

%% PID Controller Design (analog)
% K_p=30; K_i=50000; K_d=0.1*C; t_d=T/5;
K_p=40; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots

t_sim=5e-3;   t_step=2e-3;
delta_Io=20;   delta_Vin=0;   delta_Vref=0;

buck_converter_simulation;
```

Command Window

```
window. To
eliminate this
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_convert
  In Buck_PCMC_
Found algebraic lo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
```

That means, initially what we have considered is if we just take V in just 1.0 time. This is a verge of very close to the instability we are considering very close, it is even unstable.

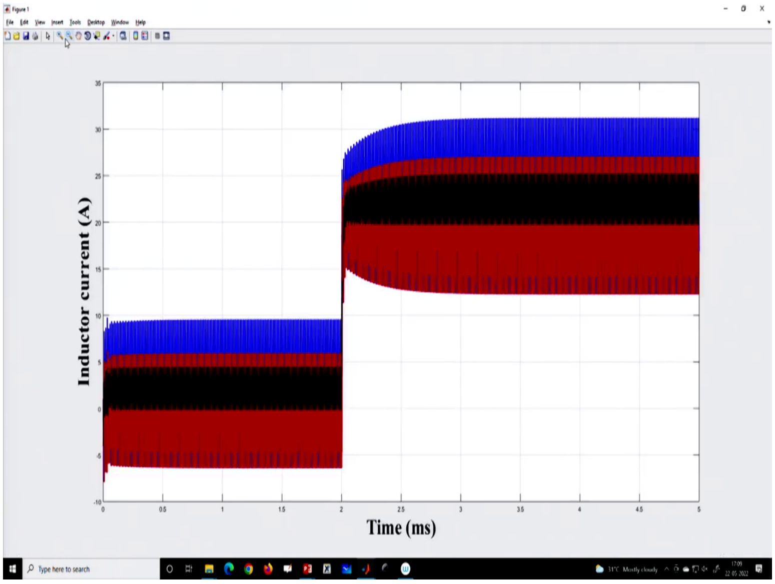
(Refer Slide Time: 26:18)



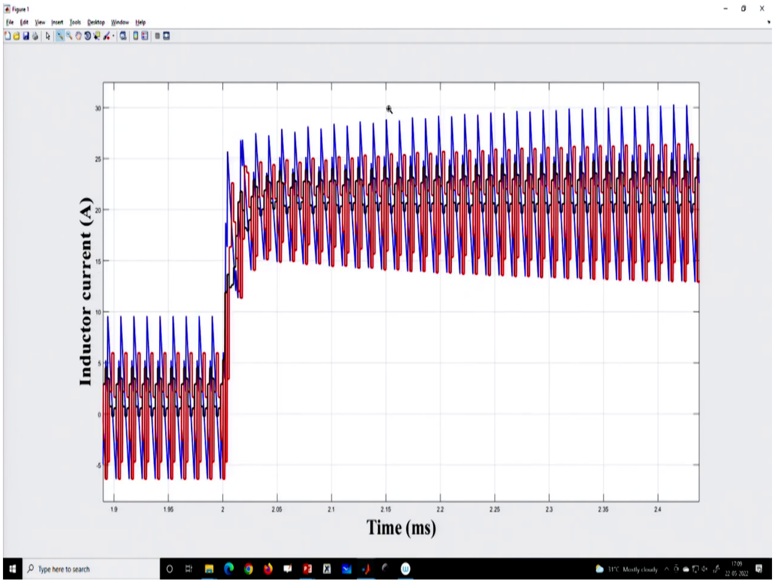
```
%% Mod
V_m=10                                   0*0.2*T; t_c=0*0.3*

mc=1.01

%% PID
% K_p
K_p=40; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots

t_sim=5e-3;   t_step=2e-3;
delta_Io=20;   delta_Vin=0;   delta_Vref=0;

buck_converter_simulation;
```

And I will show I mean it research paper there is a research paper.

(Refer Slide Time: 26:19)

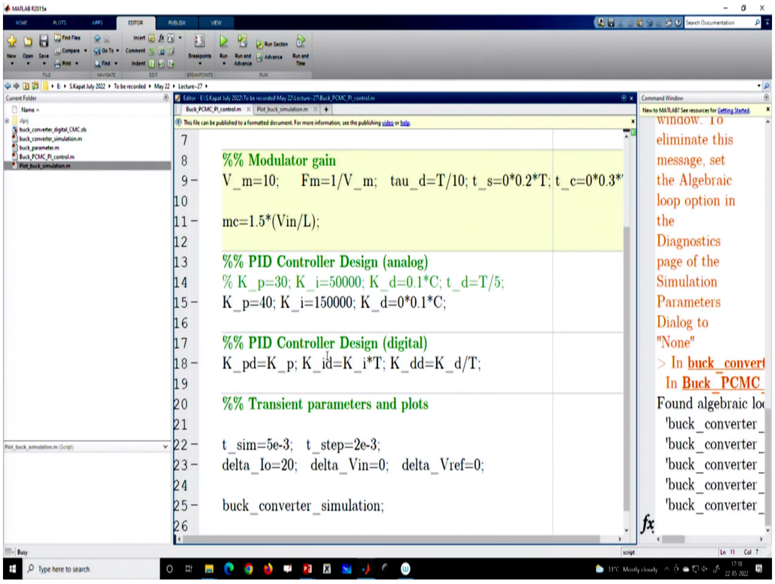


(Refer Slide Time: 26:24)



When we will go to analysis, even even if you make it higher it is not enough when you close the outer loop. Again these are research topics.

(Refer Slide Time: 26:36)

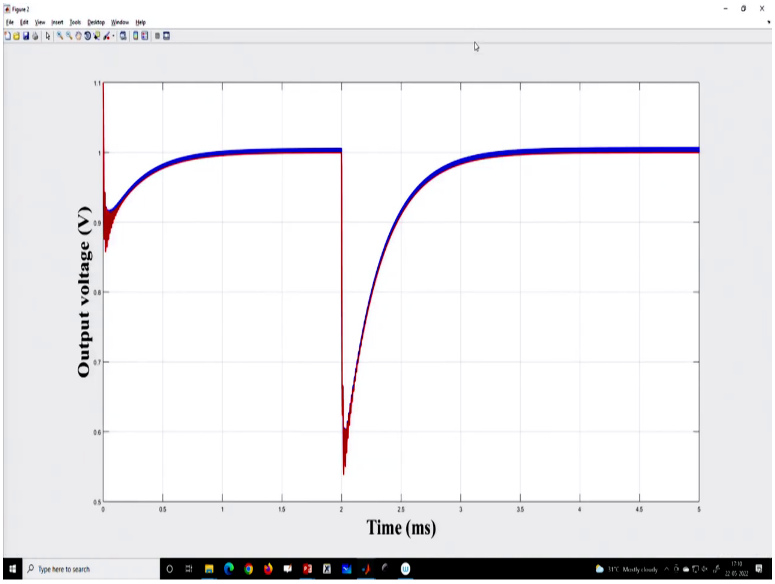


```
%% Modulator gain
V_m=10;    Fm=1/V_m;   tau_d=T/10; t_s=0*0.2*T; t_c=0*0.3*

mc=1.5*(Vin/L);

%% PID Controller Design (analog)
% K_p=30; K_i=50000; K_d=0.1*C; t_d=T/5;
K_p=40; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots

t_sim=5e-3;  t_step=2e-3;
delta_Io=20;  delta_Vin=0;  delta_Vref=0;

buck_converter_simulation;
```

```
window. To
eliminate this
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_convert
  In Buck_PCMC_
Found algebraic lo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
```

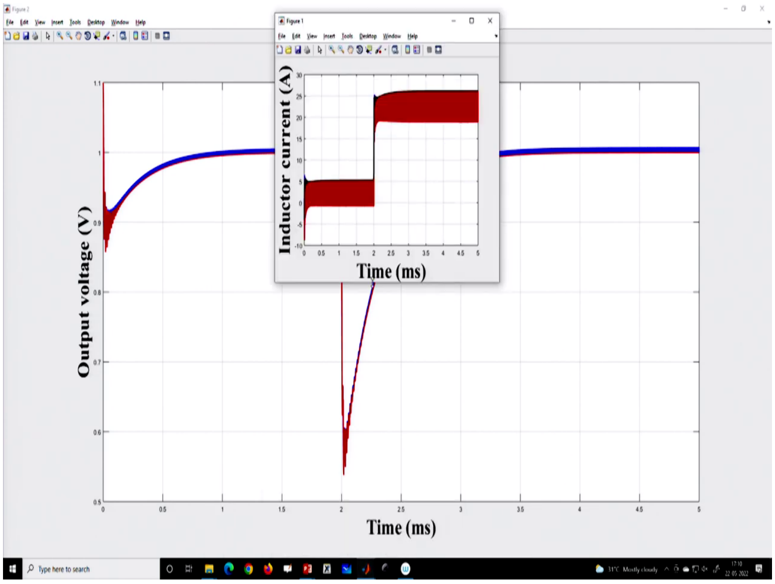
But I can say that we need to take sufficiently a conservative choice of this. So, that it will be stable. So, we have to make sure the current loop is closed.

(Refer Slide Time: 26:43)

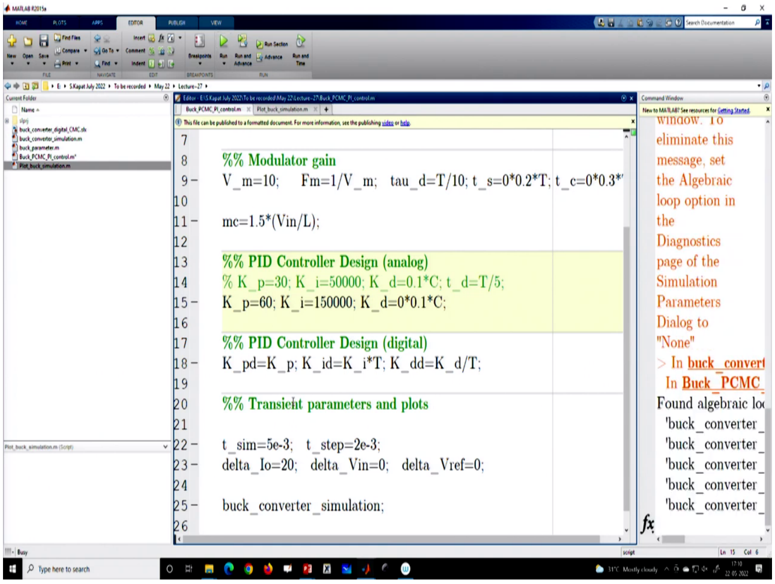


Because if we increase and you can see it is stable it is stable.

(Refer Slide Time: 26:49)



(Refer Slide Time: 26:56)



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

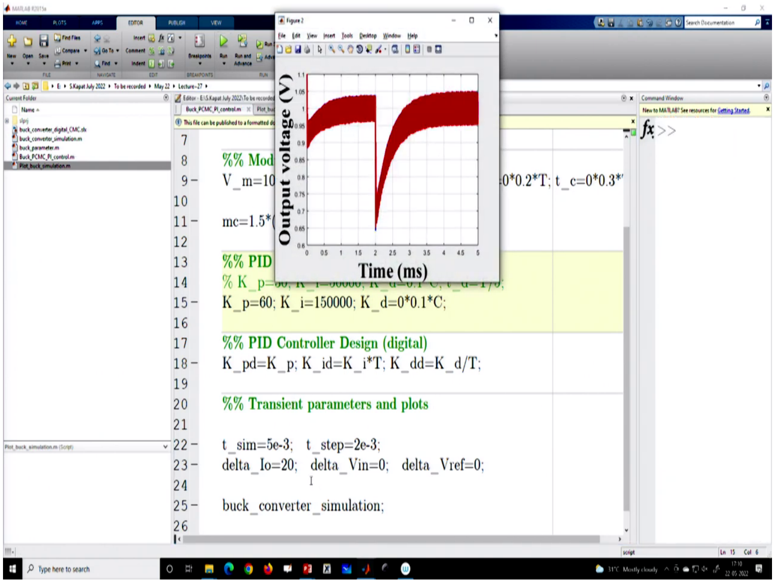
```
%% Modulator gain
V_m=10;    Fm=1/V_m;   tau_d=T/10; t_s=0*0.2*T; t_c=0*0.3*

mc=1.5*(Vin/L);

%% PID Controller Design (analog)
% K_p=30; K_i=50000; K_d=0.1*C; t_d=T/5;
K_p=60; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots

t_sim=5e-3;  t_step=2e-3;
delta_Io=20;  delta_Vin=0;  delta_Vref=0;

buck_converter_simulation;
```

New to MATLAB? See resources for Getting Started.

```
window. To
eliminate this
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_convert
  In Buck_PCMC_
Found algebraic lo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
```

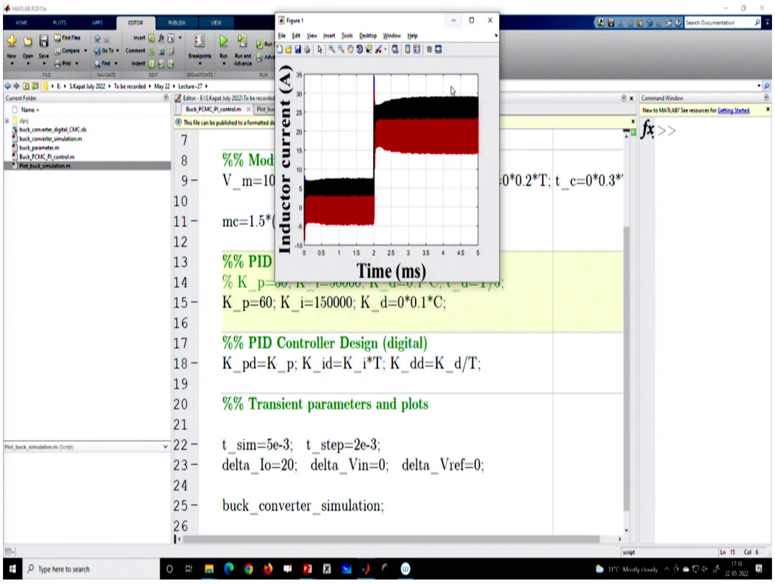
But suppose if we make this controller if we make 60; that means we have increased the controller gain and see what happened. Is it still really stable?

(Refer Slide Time: 27:01)



```
%% Mod
V_m=10                                  0*0.2*T; t_c=0*0.3*

mc=1.5*(

%% PID
% K_p=
K_p=60; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots


t_sim=5e-3;  t_step=2e-3;
delta_Io=20;  delta_Vin=0;  delta_Vref=0;

buck_converter_simulation;
```

(Refer Slide Time: 27:03)

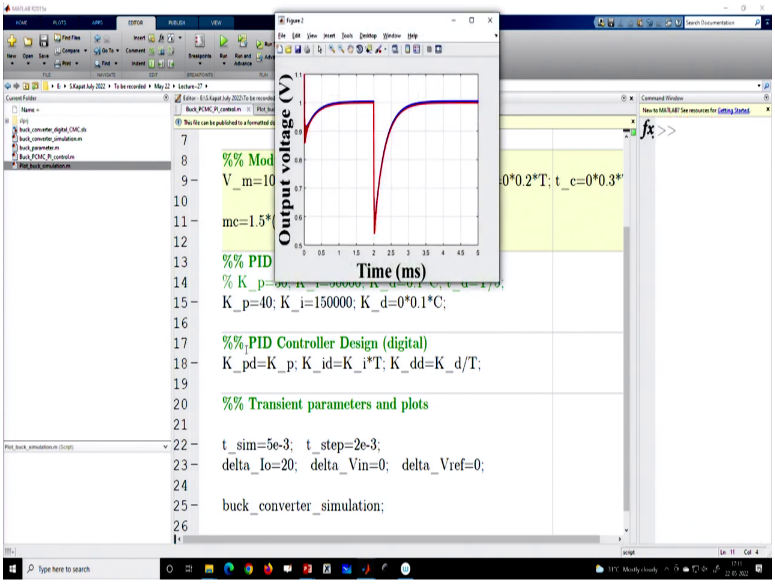


```
%% Mod
V_m=10                                  0*0.2*T; t_c=0*0.3*

mc=1.5*(

%% PID
% K_p=
K_p=60; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots


t_sim=5e-3;  t_step=2e-3;
delta_Io=20;  delta_Vin=0;  delta_Vref=0;

buck_converter_simulation;
```

So, it becomes unstable; that means, the choice of the slope also depends on the closed loop there is a research paper, but again I am not discussing it. I am just showing that you know you need to be careful about this design and we are not going to consider the design of full digital current because this is a more advanced topic, but it gives you some kind of motivation that you know this is a very interesting problem.

And you know you have to be careful about the selection of an outer voltage loop otherwise you will end up with instability.

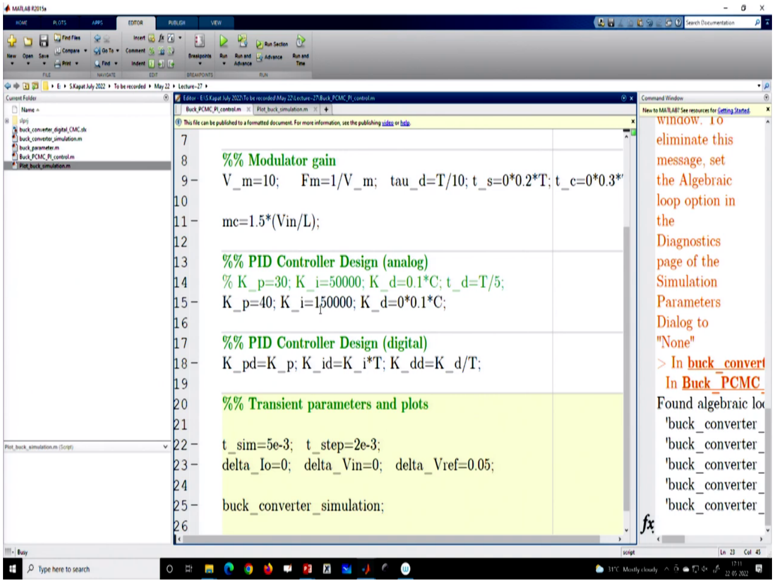
(Refer Slide Time: 27:31)



```
%% Mod
V_m=10                                              0*0.2*T; t_c=0*0.3*'

mc=1.5*(

%% PID
% K_p=
K_p=40; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots

t_sim=5e-3;   t_step=2e-3;
delta_Io=20;   delta_Vin=0;   delta_Vref=0;

buck_converter_simulation;
```

So, you can also perform here I did load transient.

(Refer Slide Time: 27:36)


Editor - E:\S.Kapat July 2022\To be recorded\May 22\Lecture-27\Buck_PCMC_PI_control.m

Buck_PCMC_PI_control.m | Plot_buck_simulation.m

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

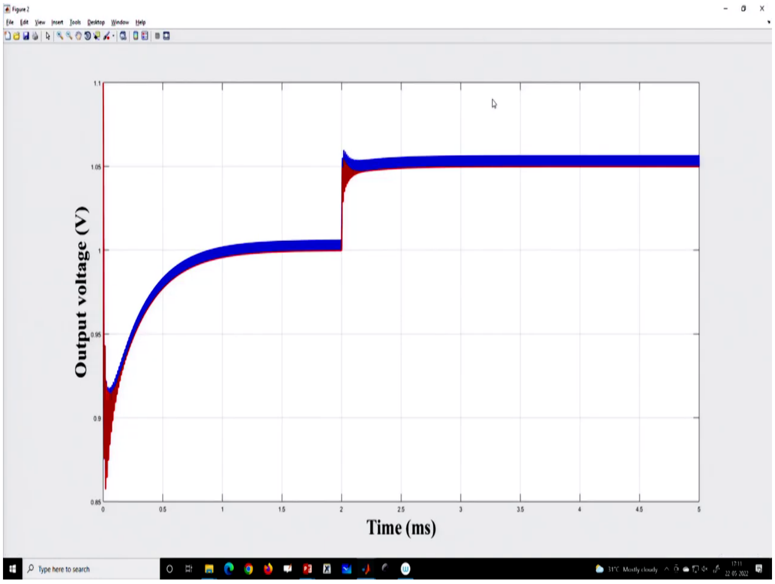
```
%% Modulator gain
V_m=10;     Fm=1/V_m;   tau_d=T/10; t_s=0*0.2*T; t_c=0*0.3*'

mc=1.5*(Vin/L);

%% PID Controller Design (analog)
% K_p=30; K_i=50000; K_d=0.1*C; t_d=T/5;
K_p=40; K_i=150000; K_d=0*0.1*C;

%% PID Controller Design (digital)
K_pd=K_p; K_id=K_i*T; K_dd=K_d/T;

%% Transient parameters and plots

t_sim=5e-3;   t_step=2e-3;
delta_Io=0;   delta_Vin=0;   delta_Vref=0.05;

buck_converter_simulation;
```

Command Window

```
window. To
eliminate this
message, set
the Algebraic
loop option in
the
Diagnostics
page of the
Simulation
Parameters
Dialog to
"None"
> In buck_convert
  In Buck_PCMC_
Found algebraic lo
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
  'buck_converter_
```

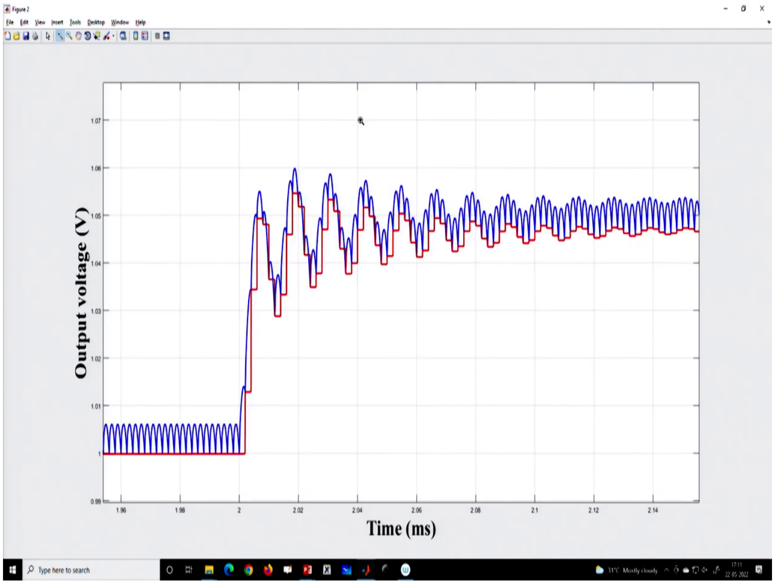
You can do reference transient and you can carry out the same reference transient that we have performed for mixed-signal current mode control ok.

(Refer Slide Time: 27:45)



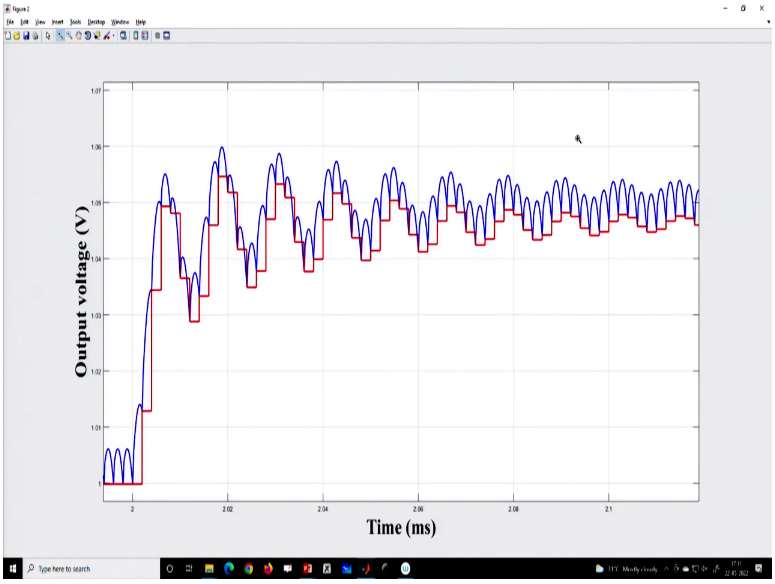
But, because this is on sample delay.

(Refer Slide Time: 27:50)



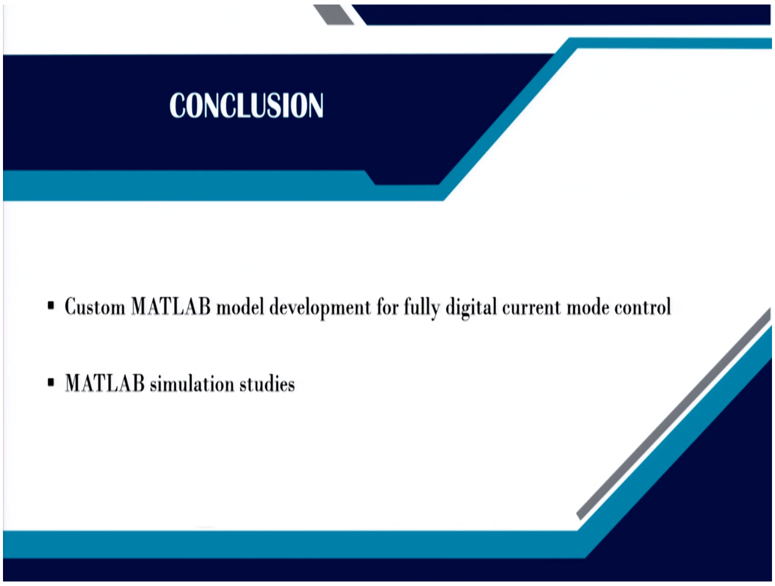
You can see there is some kind of oscillatory behavior.

(Refer Slide Time: 27:52)



Because any delay will cause the phase margin I mean to degrade the phase margin and because of that poor phase margin, it will also become more oscillatory ok.

(Refer Slide Time: 28:09)



CONCLUSION

- Custom MATLAB model development for fully digital current mode control
- MATLAB simulation studies

So, we need to be careful about this design process ok. So, in summary, we have discussed custom MATLAB model development for fully digital current mode control and we have discussed we have considered a few MATLAB simulation case studies.

So, here we want to finish the fixed frequency digital current mode control implementation. In the next lecture, we are going to talk about variable frequency digital current mode control. So, with this, I want to finish it here.

Thank you very much.