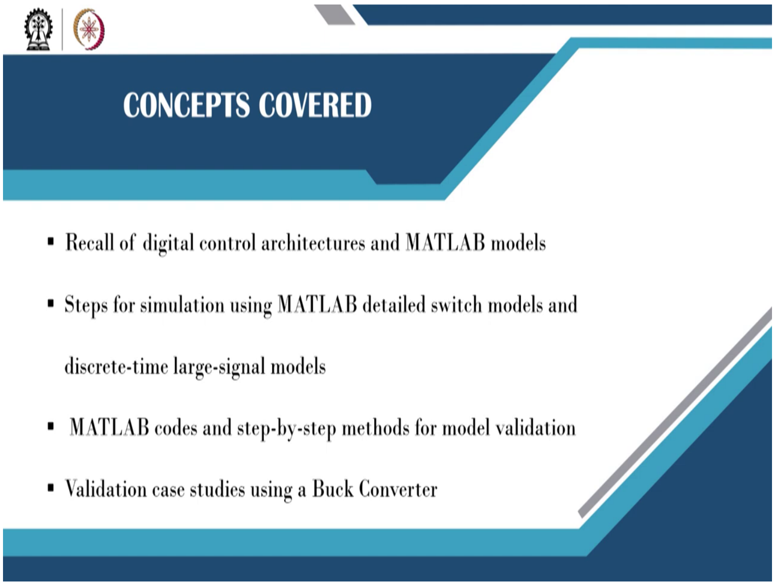
Digital Control in Switched Mode Power Converters and FPGA-based Prototyping

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

Module - 04
Modeling Techniques and Mode Validation using MATLAB
Lecture - 35
Validation of Discrete-Time Large-Signal Models using MATLAB - Part I

Welcome, this is lecture number 35. In this lecture, we are going to talk about the Validation of the Discrete-Time Large Signal Model using MATLAB and there will be 2 parts. So, this is part 1.

(Refer Slide Time: 00:35)



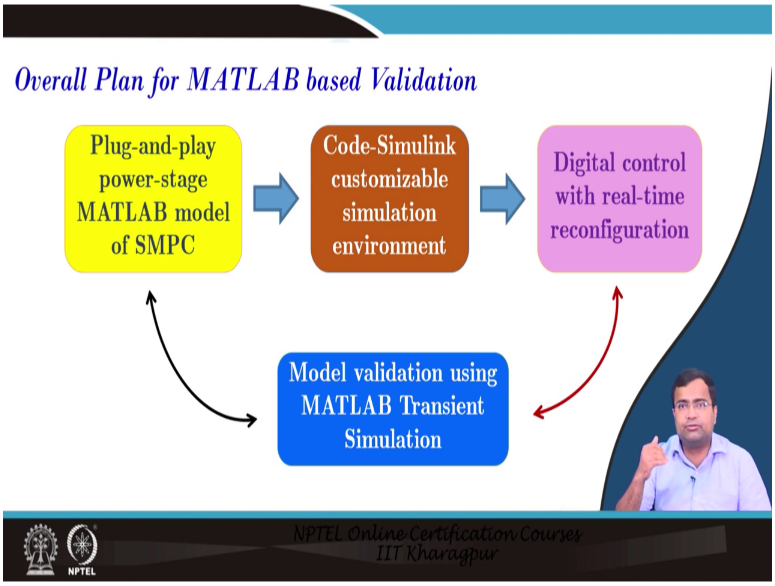
CONCEPTS COVERED

- Recall of digital control architectures and MATLAB models
- Steps for simulation using MATLAB detailed switch models and discrete-time large-signal models
- MATLAB codes and step-by-step methods for model validation
- Validation case studies using a Buck Converter

So, in this lecture first, we want to recall our digital control architecture and MATLAB models, then we want to show steps for simulation using MATLAB detailed switch models as well discrete time large signal model.

And then we will show MATLAB course codes as well as the step-by-step method for model validation. And finally, we want to show some validation case studies using a buck converter.

(Refer Slide Time: 00:59)



Overall Plan for MATLAB based Validation

Plug-and-play power-stage MATLAB model of SMPC

Code-Simulink customizable simulation environment

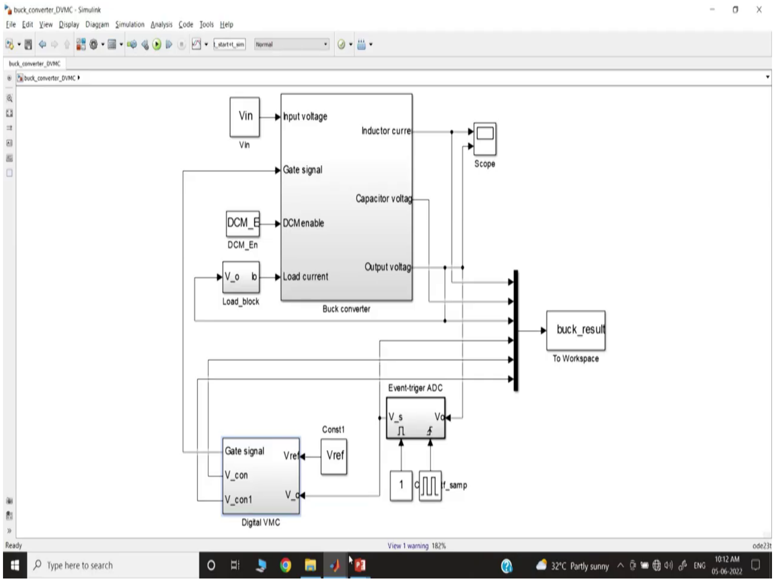
Digital control with real-time reconfiguration

Model validation using MATLAB Transient Simulation

NPTEL Online Certification Courses
IIT Kharagpur

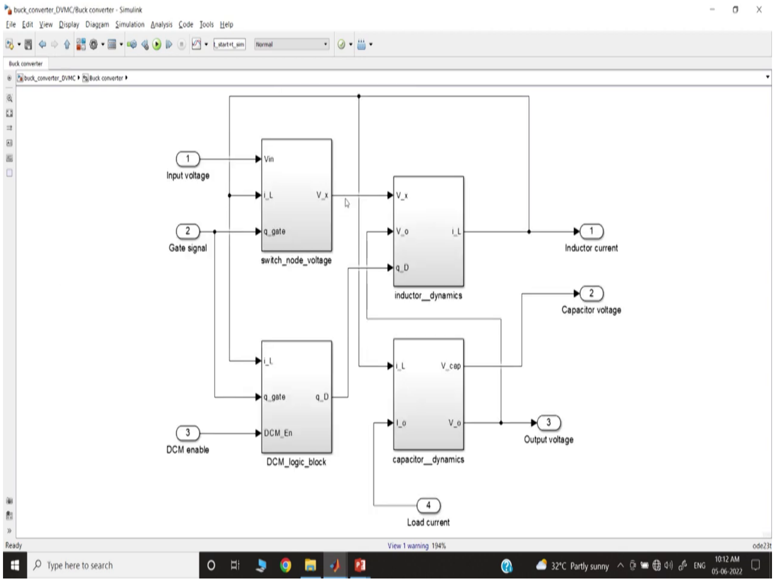
So, here I want to show the overall plan for MATLAB-based validation. So, the first step; is that means, we have a plug-and-play power stage MATLAB model of the switch mode power converter model.

(Refer Slide Time: 01:13)



And I just want to show you know if you go to our MATLAB model. So, this is the MATLAB model and this is a power stage model.

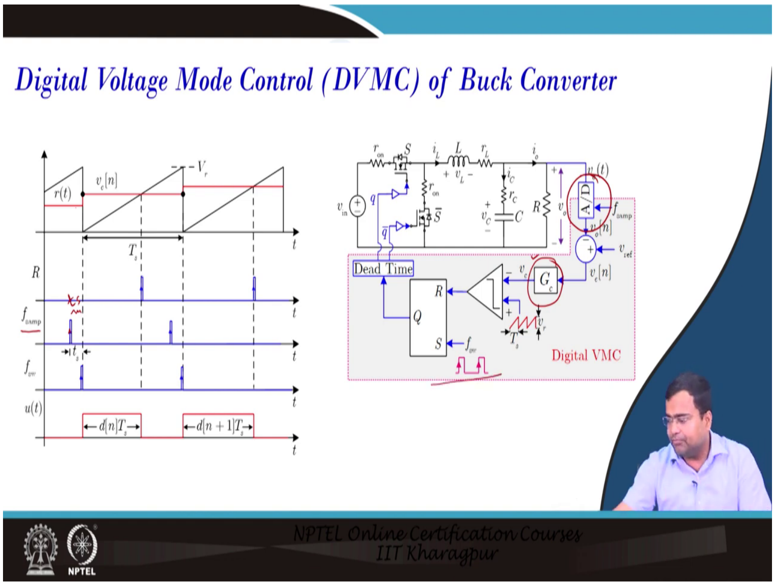
(Refer Slide Time: 01:18)



And I am talking about this model where we have plug and play MATLAB model. So, this is the power stage model, then we will be talking about the digital controller and then I will come back to the simulation again. So, the plug-and-play power stage model; then next we also have code, and Simulink customizes a customizable simulation environment, where we want to run the Simulink model from the dot and file itself. And there we want to plug in all the parameters of the buck DC-DC converter, I will say you know all these steps will be discussed in detail.

The third one, we also want to implement digital control where we can also keep some real-time reconfiguration possibility, but we want to implement digital control in Simulink. And finally, we want to validate our MATLAB model; that means, our MATLAB switch model as well as the discrete-time large signal model we want to validate. That means, we want to validate the accuracy of the discrete-time large signal model using MATLAB switch simulation.

(Refer Slide Time: 02:20)

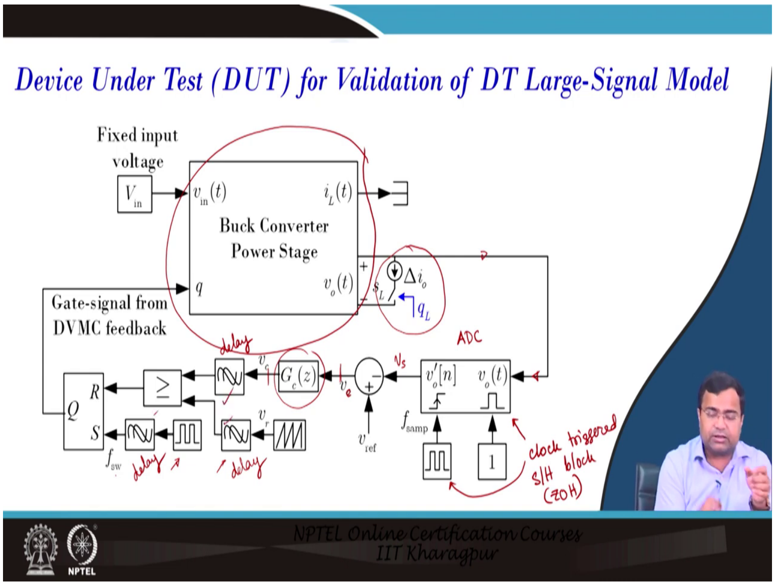


Digital Voltage Mode Control (DVMC) of Buck Converter

So, we will start with digital voltage mode control and which we have discussed multiple times in the previous week. That means, I think in week 3 we have discussed various architectures and their MATLAB model ok. And week 2 we discussed detail about the detail of the architecture.

So, here if you see that we have an A to D converter, then controller digital controller. So, we know that there will be a delay because we are talking about a sampling clock and generally the sample comes first and we need to provide some delay this is the delay, and for this delay, we said T s. This delay is used to accommodate the ADC conversion time and the controller computation time. And we will discuss how to insert such a delay in the MATLAB model so that we can sample fast and then we will switch as you know after taking the sample, then the control voltage which is the output of this controller will be ready after a certain amount of delay and when it is ready then we will start the PWM switching operation by a trailing edge modulator.

(Refer Slide Time: 03:29)



Device Under Test (DUT) for Validation of DT Large-Signal Model

So, here we want to show the MATLAB overall plan. So, this is a device under the stage; that means, I will show you the MATLAB Simulink model of the power converter, this is a power stage model and from the power stage model we are talking about a voltage mode control. So, we are sensing the output voltage, here also we can provide you know I would say a step-down ratio also, but here we are direct sensing. As you can see outside we can also put an external current sink load.

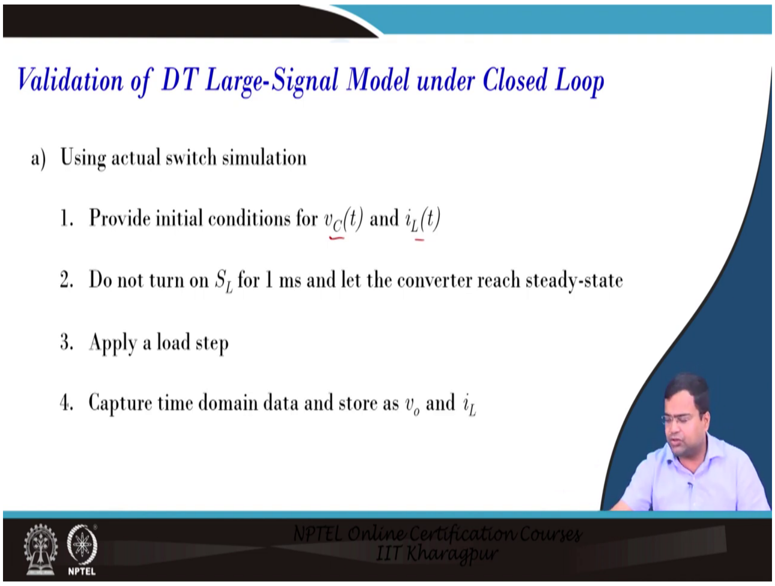
That means if you want to make a load transient and we will be talking about a load transient case study. Then once you take the output voltage you sense then we will represent our ADC block we discussed that this block consists oconsclock-triggeredd sample, a 0 order hold; sample and hold or I would say block 0 order hole block, which is nothing but 0 order hole block ok. So, sample and hole block.

Then after you get the sample voltage and this clock will be using the sampling clock which we have discussed. So, the first sample then we will get the output voltage sample here, this is going to the reference voltage, then we are getting an error voltage. The error voltage is going through a controller and since in MATLAB in the ideal environment we are not putting any delay till this point, then we are inserting an additional delay here.

So, this is my delay block, these are all delay blocks, this is also delay. So, here is what we are doing. We are delaying the control voltage because that should be updated. Similarly, the sawtooth is also delayed and the switching clock is also delayed. So that means, we are synchronizing the switching clock with the sampling clock and the switching clock is delayed by some T s amount of time.

And since the switching is starting after T s duration, the ramp should also start with respect to the switching clock, and also the controller should be ready, and the output of the controller should be ready at that time. So, all delays must be consistent, they must be the same.

(Refer Slide Time: 05:41)



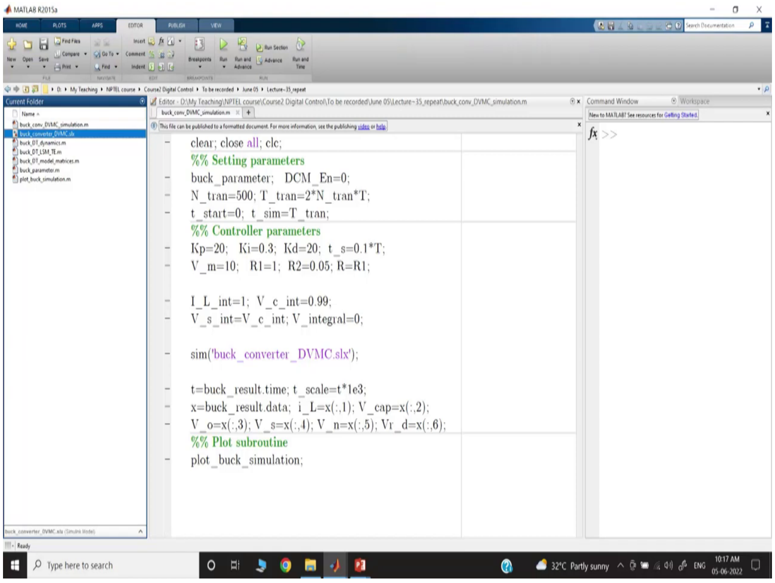
Validation of DT Large-Signal Model under Closed Loop

a) Using actual switch simulation

1. Provide initial conditions for $v_C(t)$ and $i_L(t)$
2. Do not turn on S_L for 1 ms and let the converter reach steady-state
3. Apply a load step
4. Capture time domain data and store as v_o and i_L

Now, what are the steps for the validation of the discrete-time large signal model? First, we have to run the switch simulation, the actual switch simulation where we need to provide an initial condition for capacitor voltage and the inductor current.

(Refer Slide Time: 06:03)

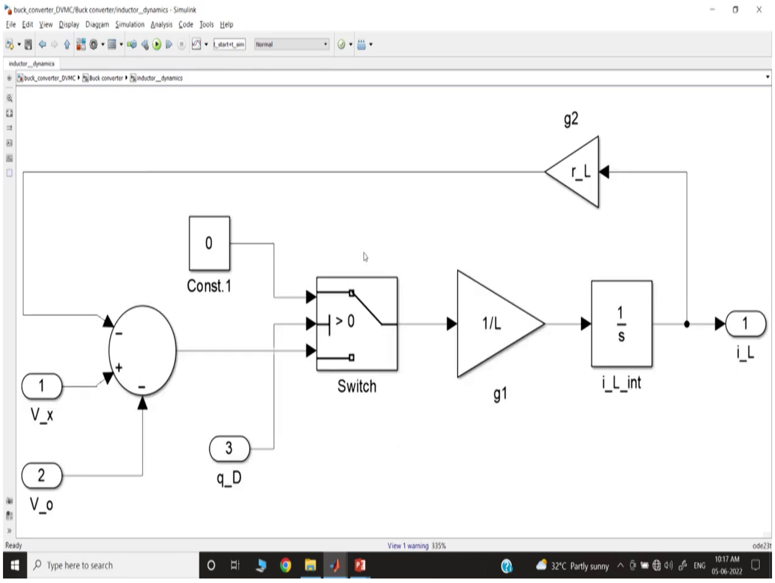


```
clear; close all; clc;
%% Setting parameters
buck_parameter;   DCM_En=0;
N_tran=500; T_tran=2*N_tran*T;
t_start=0;  t_sim=T_tran;
%% Controller parameters
Kp=20;   Ki=0.3;  Kd=20;  t_s=0.1*T;
V_m=10;   R1=1;  R2=0.05; R=R1;

I_L_int=1;  V_c_int=0.99;
V_s_int=V_c_int; V_integral=0;

sim('buck_converter_DVMC.slx');

t=buck_result.time; t_scale=t*1e3;
x=buck_result.data;  i_L=x(:,1); V_cap=x(:,2);
V_o=x(:,3); V_s=x(:,4); V_n=x(:,5); Vr_d=x(:,6);
%% Plot subroutine
plot_buck_simulation;
```

And then; so, let us we will map one by one; that means, let us go to this step. So, here I am showing the code, where you can see that we are first calling the buck converter parameter file. That means, we are loading the parameter file, then we are calling the DCM to enable 0; that means we are operating in a synchronous mode.

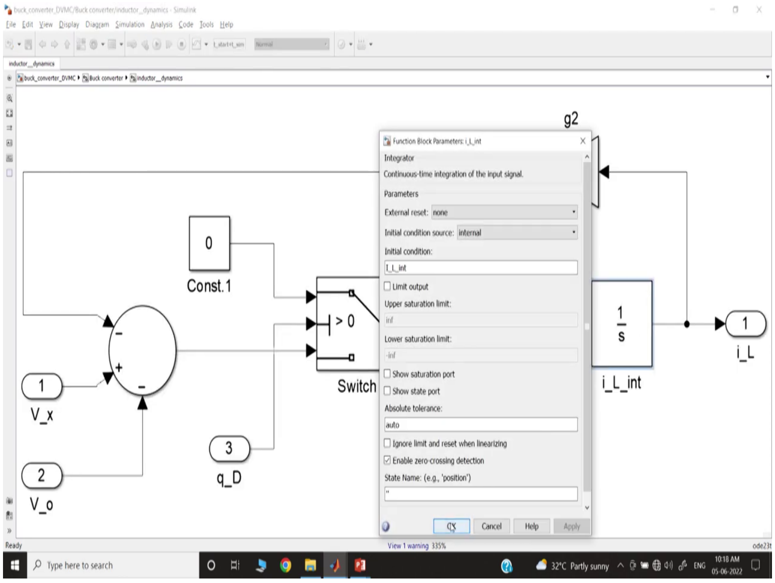
And we are also initializing that we are going to discuss, this is the initialization of the inductor current as well as this is the initialization of the capacitor voltage. And we have discussed the power stage, how to where is the initialization block.

(Refer Slide Time: 06:39)



Because if you go inside if you go to the current dynamics; so, you will find this initialization will be here, sitting here ok.

(Refer Slide Time: 06:42)


Function Block Parameters: i_L_int

Integrator

Continuous-time integration of the input signal.

Parameters

External reset: none

Initial condition source: internal

Initial condition:

I_L_int

□ Limit output

Upper saturation limit:

inf

Lower saturation limit:

-inf

□ Show saturation port

□ Show state port

Absolute tolerance:

auto

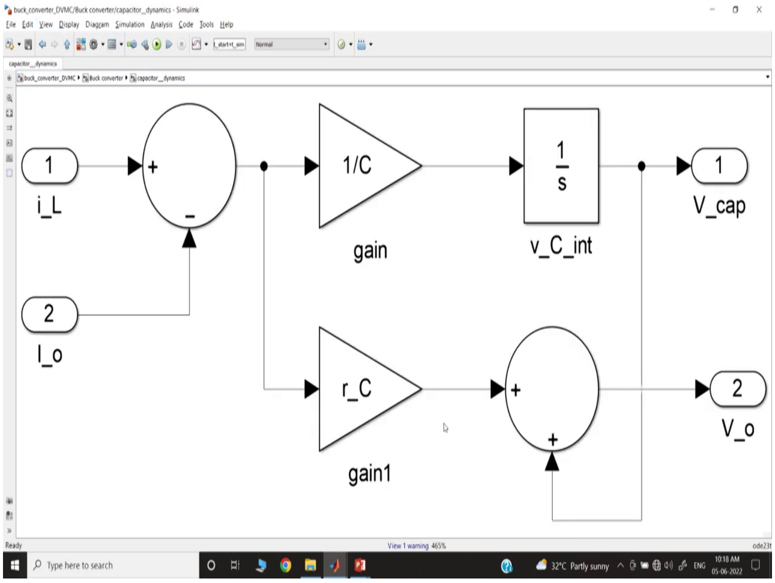
□ Ignore limit and reset when linearizing

☑ Enable zero-crossing detection

State Name: (e.g., 'position')

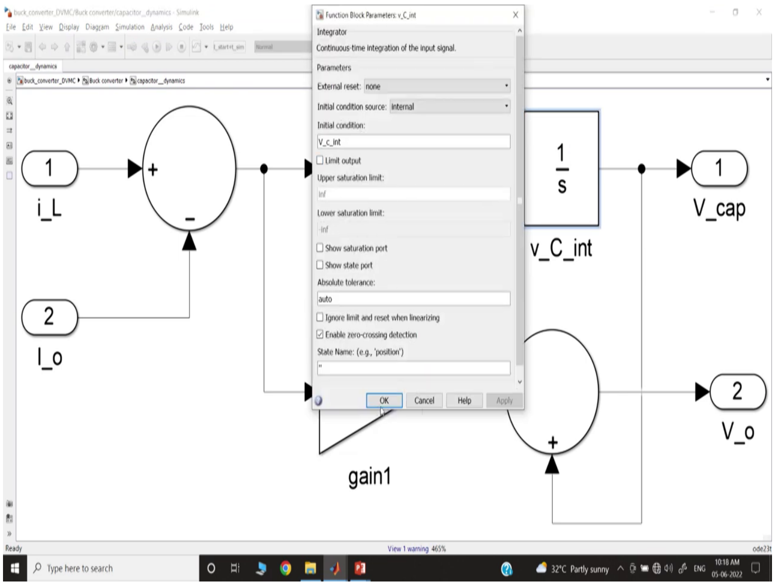
''

(Refer Slide Time: 06:47)



Then if you go to the voltage dynamics; that means, the capacitor dynamics.

(Refer Slide Time: 06:50)


Function Block Parameters: v_C_int

Integrator

Continuous-time integration of the input signal.

Parameters

External reset: none

Initial condition source: internal

Initial condition:

V_c_int

□ Limit output

Upper saturation limit:

inf

Lower saturation limit:

-inf

□ Show saturation port

□ Show state port

Absolute tolerance:

auto

□ Ignore limit and reset when linearizing

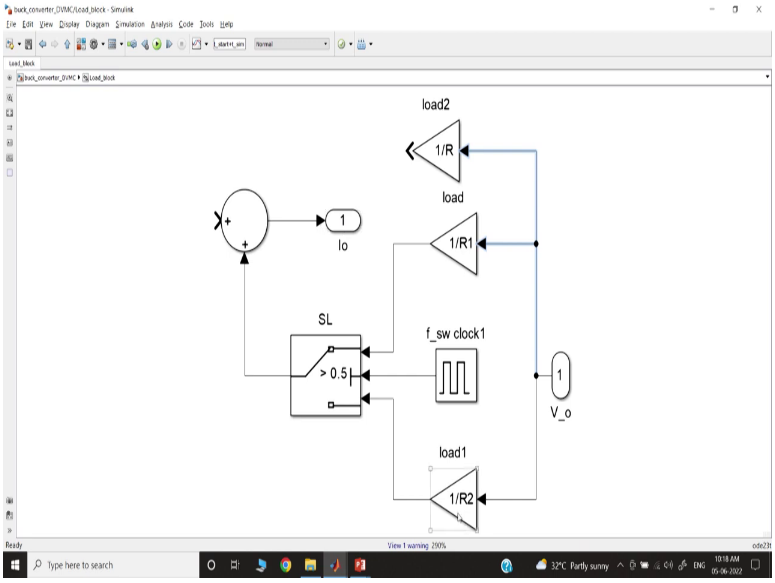
☑ Enable zero-crossing detection

State Name: (e.g., 'position')

''

Then you will get the initialization block here. So, this initialization voltage we are setting from the dot m file and that we have discussed. Next step; means, we have initialized and then, we will provide this.

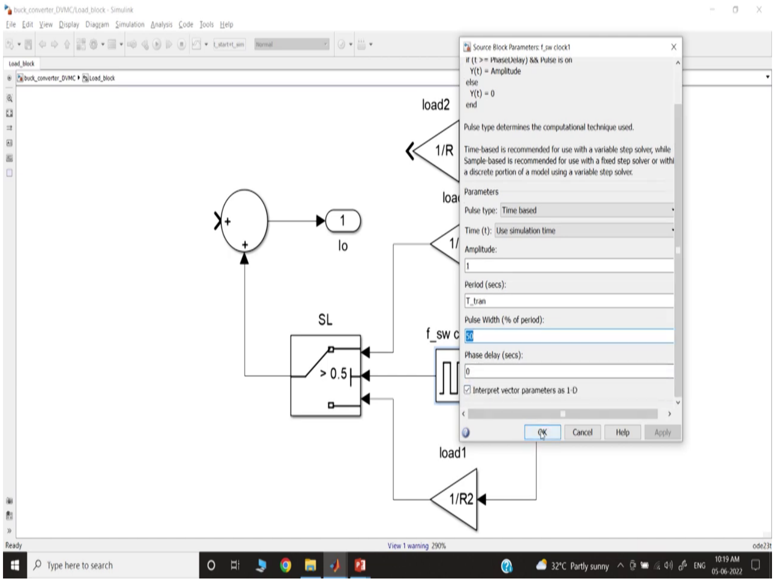
(Refer Slide Time: 07:15)



Next, we will initially run for one load condition and if you go to that load block; that means, I will be going this is load block, this is a load block inside you can see there are two load resistance R 1 and R 2.

So, R is a continuous load resistance, but we are not using it for the time being. So; that means, what we will do initially will run R 1 resistance because see this is the output voltage and this is the load current. So, output voltage by load current is nothing but; that means, the load resistance; that means, your load current is nothing but V 0 by R 1 and in the second case it will be V 0 by R2.

(Refer Slide Time: 07:46)



Source Block Parameters: f_sw clock1

```
if (t >= PhaseDelay) && Pulse is on
  Y(t) = Amplitude
else
  Y(t) = 0
end
```

Pulse type determines the computational technique used.

Time-based is recommended for use with a variable step solver, while Sample-based is recommended for use with a fixed step solver or withi a discrete portion of a model using a variable step solver.

Parameters

Pulse type: Time based

Time (t): Use simulation time

Amplitude: 1

Period (secs): T_tran

Pulse Width (% of period): 50

Phase delay (secs): 0

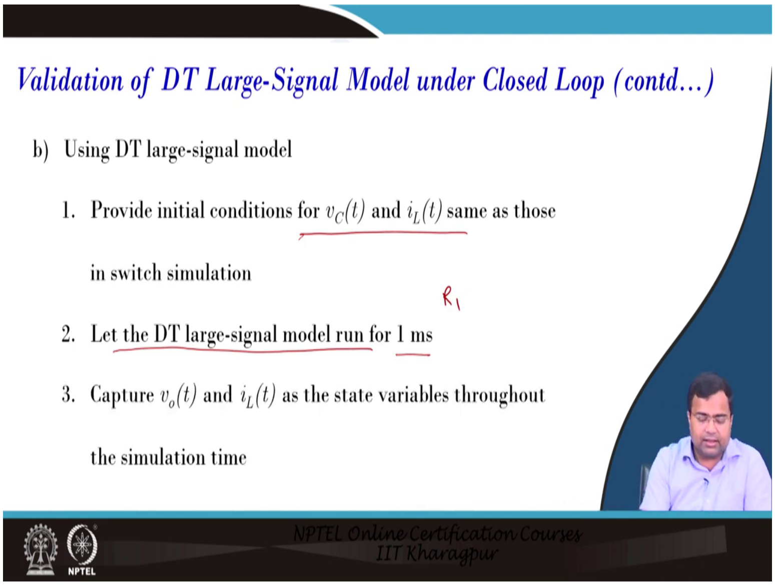
☑ Interpret vector parameters as 1-D

And if you go to the clock, you know I will discuss in detail, there is a clock which will take either 1 or 0 and it has a fixed frequency clock with time period T tran, I will discuss what is T tran is and it will have 50 percent duty ratio and there is no phase delay here. And that means, after 50 percent time of the T tran, the first 50 percent time it will run R 1 resistance, and then another 50 percent it will connect R 2 resistance ok.

So that means, we have a resistance and we are making a load triangle. So, we want to check, we want to show the validation for both steadthstatestate as well as the transient case because we are talking about a large signal model.

Then we will apply the load step. I have told you that it will change to another load resistance. So, that will implicate a load resistance, then we want to capture the time domain data and we will store it.

(Refer Slide Time: 08:37)



Validation of DT Large-Signal Model under Closed Loop (contd…)

b) Using DT large-signal model

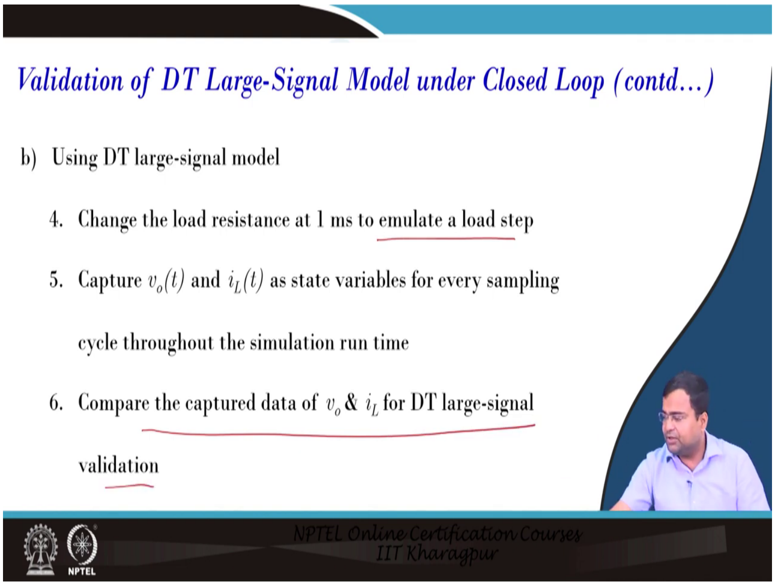
1. Provide initial conditions for $v_C(t)$ and $i_L(t)$ same as those in switch simulation
2. Let the DT large-signal model run for 1 ms
3. Capture $v_o(t)$ and $i_L(t)$ as the state variables throughout the simulation time

And I will show you all the step, then once you store the switch simulation data which include initially one load resistance, then after some time another load resistance with transient as well as the two steady-state condition we store this data in the workspace.

Then, we will use a discrete-time large signal model and we will provide an initial condition for v C and I L, the same as those in switch simulation. So, the initial condition must match; that means, the large signal model should take the same initial condition of the switch simulation. Then we will run the discrete-time large signal model for the first millisecond and you can also change this, it doesn't need to be 1 millisecond, you can take a shorter duration or longer duration.

But initially, we should take R 1 resistance that is for the first time. Then when there is a load transient, you change to R 2 resistance and will run for some more time. Then we will capture all the variables.

(Refer Slide Time: 09:34)



Validation of DT Large-Signal Model under Closed Loop (contd…)

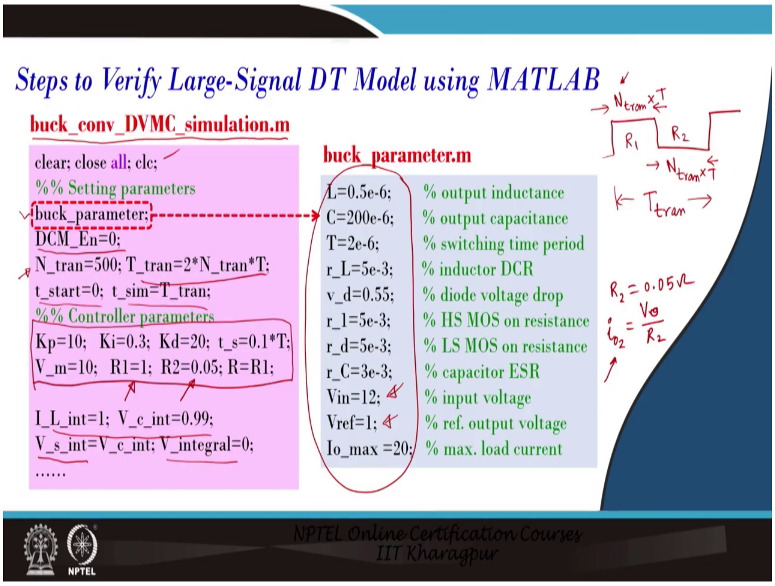
b) Using DT large-signal model

4. Change the load resistance at 1 ms to emulate a load step
5. Capture $v_o(t)$ and $i_L(t)$ as state variables for every sampling cycle throughout the simulation run time
6. Compare the captured data of v_o & i_L for DT large-signal validation

Then what we will do? Now, you change the load resistance to two emulated load steps that we have discussed. So, R 1 was connected earlier and it will be R 2 resistance, load resistance and virtually it looks like a load step, then again we will capture all the variables like voltage and current data during the simulation runtime.

And then we compare the capture discrete-time model and we will compare with the actual switch simulation. That means you know the last step will be, will compare the discrete-time large signal model as well as the switch simulation data.

(Refer Slide Time: 10:09)



Steps to Verify Large-Signal DT Model using MATLAB

buck_conv_DVMC_simulation.m

```
clear; close all; clc;
%% Setting parameters
buck_parameter;
DCM_En=0;
N_tran=500; T_tran=2*N_tran*T;
t_start=0; t_sim=T_tran;
%% Controller parameters
Kp=10;   Ki=0.3;  Kd=20;  t_s=0.1*T;
V_m=10;   R1=1;  R2=0.05; R=R1;

I_L_int=1;  V_c_int=0.99;
V_s_int=V_c_int; V_integral=0;
……
```

buck_parameter.m

```
L=0.5e-6;       % output inductance
C=200e-6;       % output capacitance
T=2e-6;         % switching time period
r_L=5e-3;       % inductor DCR
v_d=0.55;       % diode voltage drop
r_1=5e-3;       % HS MOS on resistance
r_d=5e-3;       % LS MOS on resistance
r_C=3e-3;       % capacitor ESR
Vin=12;         % input voltage
Vref=1;         % ref. output voltage
Io_max =20;     % max. load current
```

So, let us go and check. So, this is the code that we are going to demonstrate in MATLAB. So, the first code we are writing is a digital voltage mode control in a buck converter and there is a dot m file.

So, we will first clear the screen, then we will load the buck parameter and again all these files are here. So, the buck parameter we are considering is the inductor, capacitance tor values and we are initially talking about a reference voltage of 1 volt, input voltage you can change in the code itself, you can use 12 volts or something else.

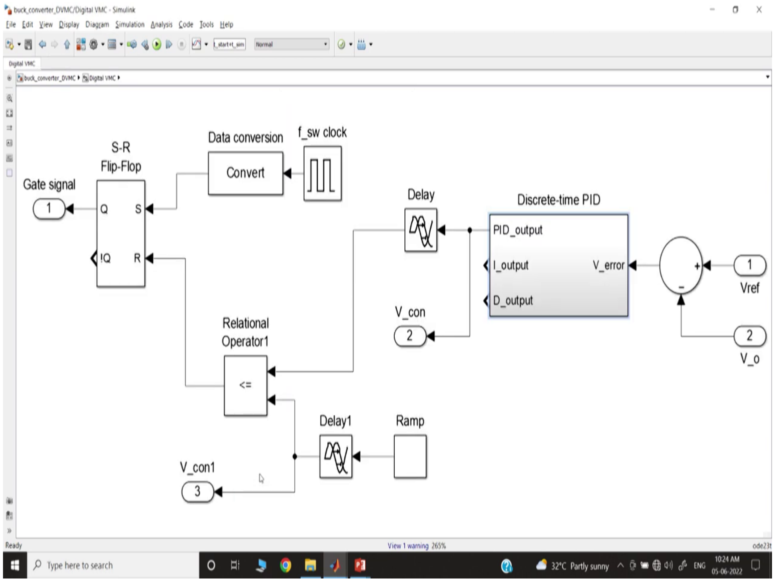
Then we are putting DCM enables 0; that means, it is in the synchronous mode; which means, the current can go in the negative direction. Then initially N tran is several cycles. So, first 500 cycles it will run N 1 R 1 resistance; that means if I talk like this. So, I would say this duration is the N tran into T, where here it is the number of cycles and this time your R 1 resistance was connected and this time your R 2 resistance is connected and we are taking 50 percent.

So, this duration is also N tran into T so; that means, the total duration from here to here will be twice N tran into 2, into T. Then we will start the simulation and then we will run for the total T tran time, which is the sum of this is my T tran time ok, we run this case. Now, we are setting some controller values, but again we have not yet discussed the controller design. So, we want to first show for a given set of controllers how can we validate the model.

Initially, we are considering load resistance R 1 to be 1 ohm, since Vref is 1 volt. So, 1 volt by 1 ohm is the 1-ampere load step for the initial case. Then we are making we are changing the load resistance from R 1 to R 2 and R 2 is 0.05 ohm; that means, f you divide; that means, R 2 is we are taking 0.05 ohm. So, then what will be the load i 0 2? It will be Vref or I say V 0 by R 2 and if the voltage is regulated at 1 volt, it represents a 20-ampere load change ok.

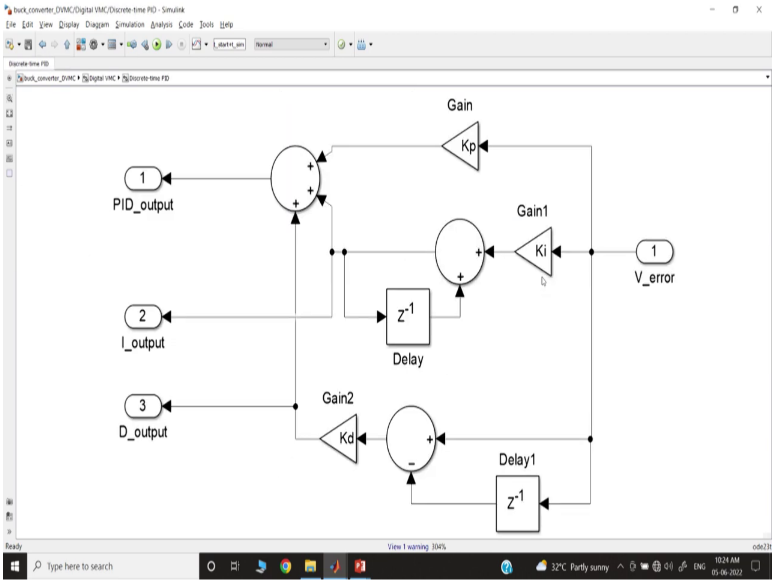
We can also use a load step like a constant current load you can use. So, here we are just objective is to validate and we are again setting the initial condition. Now, there are two more initial conditions, for example, the sample voltage initial condition, and another is the integral initial condition. So, let us go to the MATLAB block and I will show you.

(Refer Slide Time: 12:59)



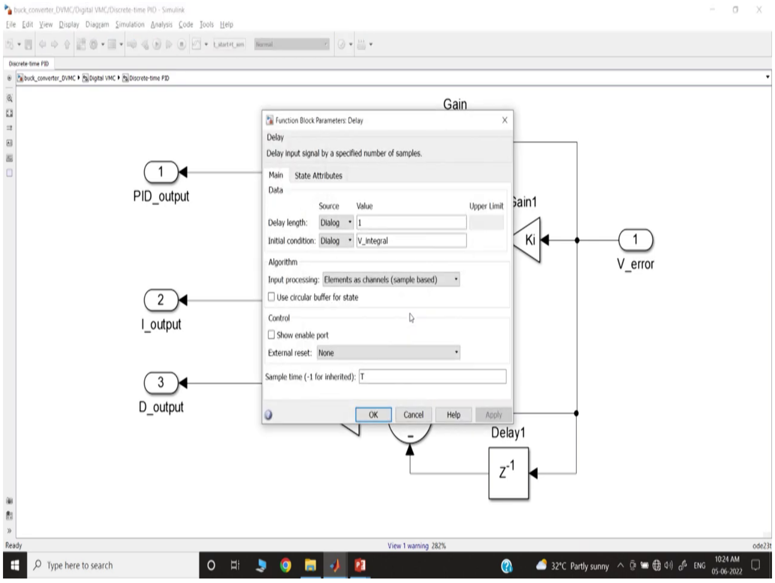
If you go to the MATLAB block here we have a digital controller.

(Refer Slide Time: 13:01)


buck_converter_DVMC/Digital VMC/Discrete-time PID - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
Discrete-time PID
Gain
Kp
1
PID_output
Gain1
Ki
1
V_error
2
I_output
z^-1
Delay
3
D_output
Gain2
Kd
Delay1
z^-1
Ready

And if you go inside, this is the block because we are talking about this is a PI controller, a sorry integral part of the discrete-time integral control. This is the proportional control and this is the discrete-time derivative control, ok.

(Refer Slide Time: 13:19)


buck_converter_DVMC/Digital VMC/Discrete-time PID - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
Discrete-time PID
buck_converter_DVMC ▸ Digital VMC ▸ Discrete-time PID
Gain
1
PID_output
2
I_output
3
D_output
Gain1
Ki
1
V_error
Delay1
z^{-1}
Ready
View 1 warning 282%
ode23t
Type here to search
32°C Partly sunny
ENG
10:24 AM
05-06-2022

Function Block Parameters: Delay

Delay

Delay input signal by a specified number of samples.

Main | State Attributes

Data

	Source	Value	Upper Limit
Delay length:	Dialog	1	
Initial condition:	Dialog	V_integral	

Algorithm

Input processing: Elements as channels (sample based)

□ Use circular buffer for state

Control

□ Show enable port

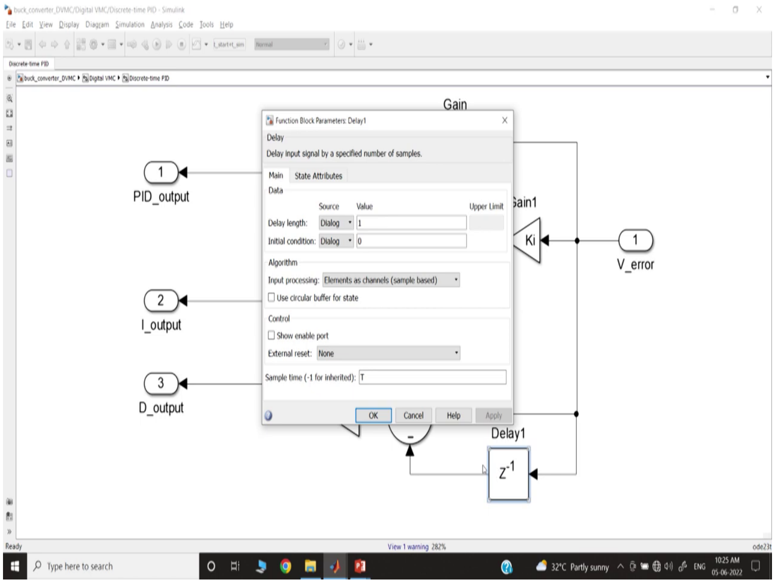
External reset: None

Sample time (-1 for inherited): T

OK | Cancel | Help | Apply

So, if you go inside this block, you will find something like a V integral, because you see this part is u n plus u n. This is the u n that is the output of the integral control u I of n, that is integral with nth interval, that value is equal to its previous value. That means, u i n minus 1 plus k I into error voltage ok. That means, we will discuss this, this is a discrete-time integral gain and then this is our discrete-time derivative gain, what is that? That will be Kd into v e n which is the running sample error voltage minus its previous value.

(Refer Slide Time: 14:04)


buck_converter_DVMC/Digital VMC/Discrete-time PID - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
Discrete-time PID
buck_converter_DVMC ▸ Digital VMC ▸ Discrete-time PID
Gain
1
PID_output
2
I_output
3
D_output
Gain1
Ki
1
V_error
Delay1
z^{-1}
Ready
View 1 warning 282%
ode23t
Type here to search
32°C Partly sunny
ENG
10:25 AM
05-06-2022

Function Block Parameters: Delay1

Delay

Delay input signal by a specified number of samples.

Main | State Attributes

Data

	Source	Value	Upper Limit
Delay length:	Dialog	1	
Initial condition:	Dialog	0	

Algorithm

Input processing: Elements as channels (sample based)

□ Use circular buffer for state

Control

□ Show enable port

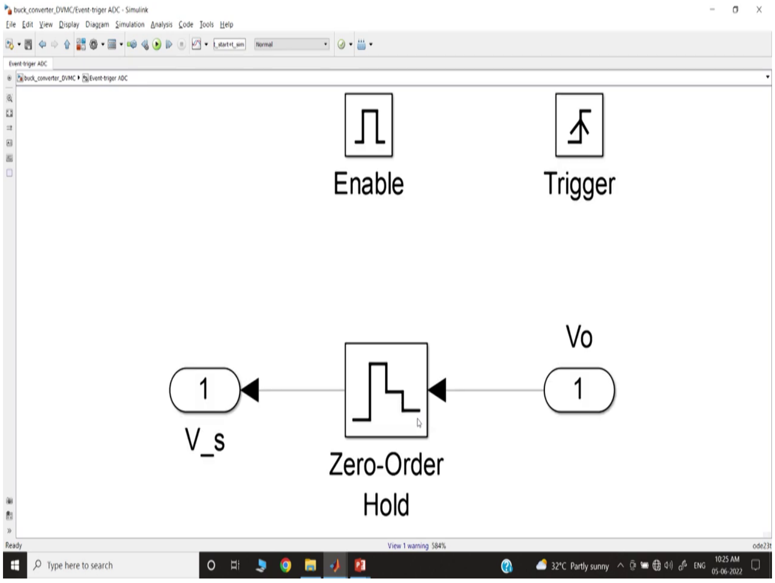
External reset: None

Sample time (-1 for inherited): T

OK | Cancel | Help | Apply

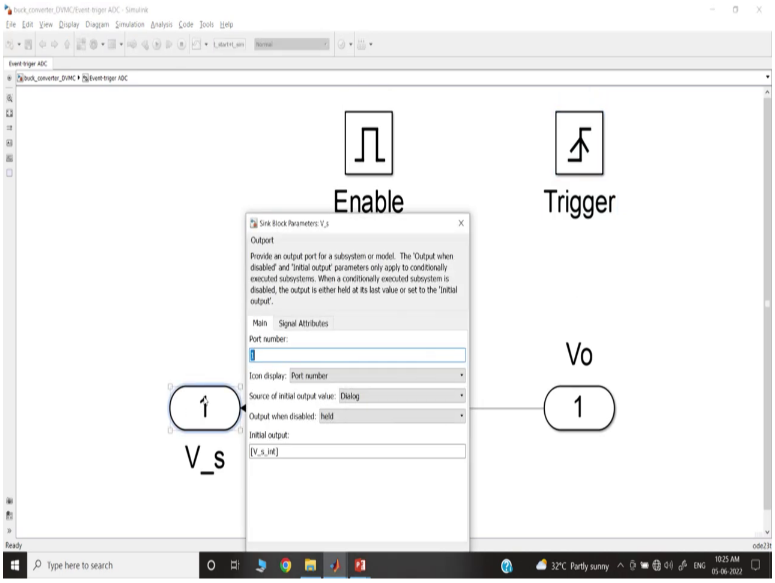
And if you go inside we can also initialize, but here we are setting simply 0.

(Refer Slide Time: 14:14)


buck_converter_DVMC/Event-triger ADC - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
Event-triger ADC
Enable
Trigger
Vo
1
1
V_s
Zero-Order
Hold
Ready
Type here to search

Another I want to say is if you go to the ADC block; that means, this block inside this block if you go you can see there is a 0 order hold, but it will only enable whenever the enable it should be enabled and it will capture the signal at the rising edge of the trigger signal ok.

(Refer Slide Time: 14:29)


buck_converter_DVMC/Event-triger ADC - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
Enable
Trigger
Vo
1
1
V_s
Type here to search

Sink Block Parameters: V_s

Outport

Provide an output port for a subsystem or model. The 'Output when disabled' and 'Initial output' parameters only apply to conditionally executed subsystems. When a conditionally executed subsystem is disabled, the output is either held at its last value or set to the 'Initial output'.

Main | Signal Attributes

Port number: 1

Icon display: Port number

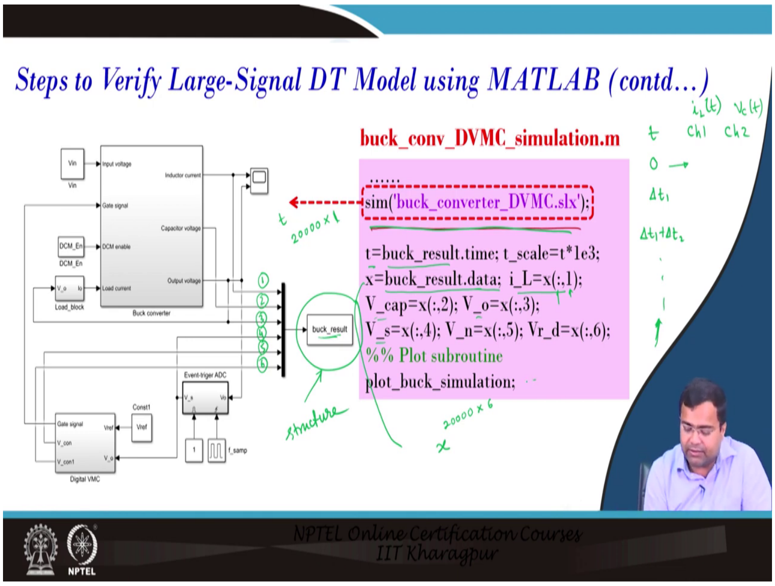
Source of initial output value: Dialog

Output when disabled: held

Initial output: [V_s_int]

And if you go inside the output of the 0 order hold, this is like a register and here we are initializing V s initializer ok. So, we have explained all the model parts here, and let us go to the presentation ok.

(Refer Slide Time: 14:41)



Steps to Verify Large-Signal DT Model using MATLAB (contd…)

buck_conv_DVMC_simulation.m

```
……
sim('buck_converter_DVMC.slx');

t=buck_result.time; t_scale=t*1e3;
x=buck_result.data; i_L=x(:,1);
V_cap=x(:,2); V_o=x(:,3);
V_s=x(:,4); V_n=x(:,5); Vr_d=x(:,6);
%% Plot subroutine
plot_buck_simulation;
```

NPTEL Online Certification Courses
IIT Kharagpur

So, here next we have discussed that this is our overall you know I would say the Simulink block and we are running the Simulink file from the dot m file here; that means, you know if you look at the dot m file. So, this is the dot m file, we are running through the simulation file.

Then once you run, this will all the result time domain results will be stored in the workspace you say there are many channels; channel 1, channel 2, channel 3, channel 4, channel 5, and channel 6 there are 6 channels there and for each channel the time is common. So, this is like you know channel 1, channel 2 like that.

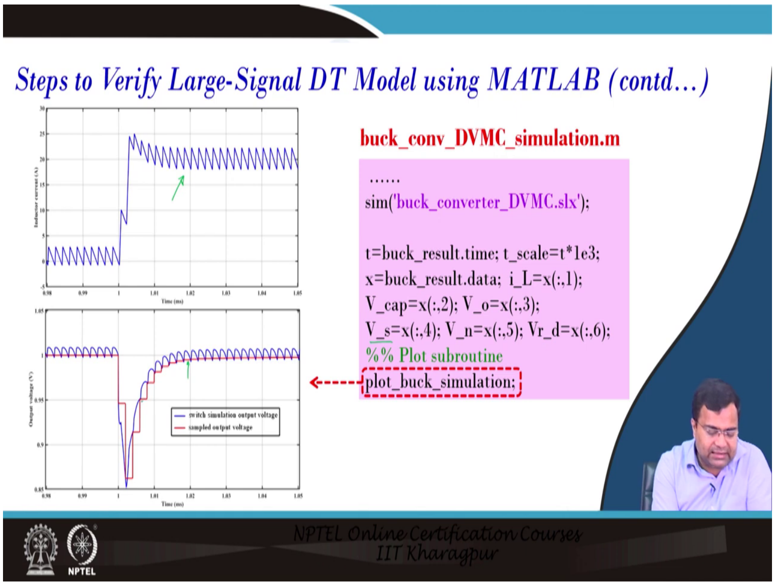
So, time will be 0, then there can be first step delta t 1, and the second can be delta t 1 plus delta t 2 because if you are using a variable time solver each step can vary, so like it will go. Then channel 1 if you take this to be inductor current for example. So, it will show the values of the inductor current at that time instant. Similarly, if you take let us say here we are taking capacitor voltage. So, it will be V c t.

So, all this value so; that means, in this vector, you will get first the time vector, that is why it is written; that means, that t is equal to buck result, which is the file name; that means, the name of the workspace that where you are storing. It is a structure, it is a structure in time. So, you will get a structure, in this structure if you first access dotted time. So, it will take the time vector, so time vector.

Then you have a structure that will be dot data and this data depends on what is the size. So, if there is like it runs for a total of 20000 let us say data point; that means, the vector; that means, I am telling the length of time let us the time length is 20001. Then since there are 6 channels what is the data vector length x? So, x here in this case which is here will be dimensionally first; that means, 20000 into 6 channels

So, you have to take one channel and then you will get the whole data of current whatever you are coming ok. So, like that way you will get and then once you take all the separately if you call that output voltage, capacitor voltage, sample voltage, inductor current. So, the inductor current is the first channel. So, it will take the first channel and take all the data here that it can capture then we will have a plot command.

(Refer Slide Time: 17:23)



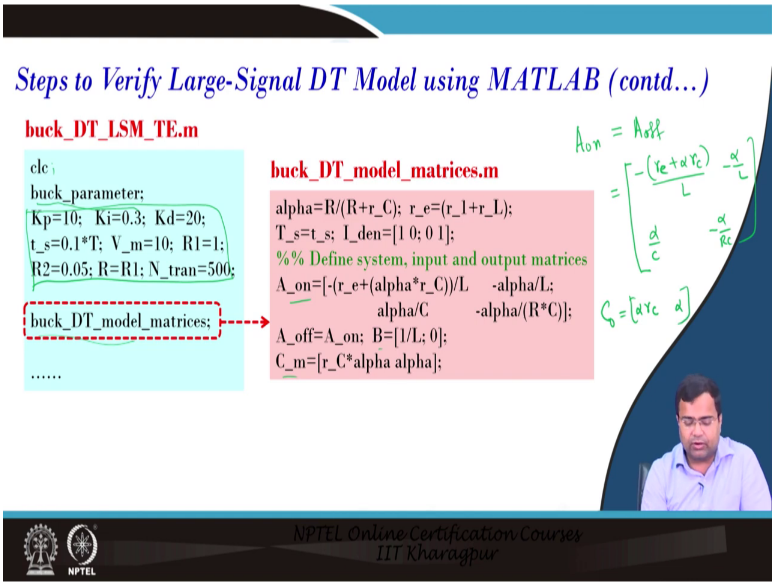
Steps to Verify Large-Signal DT Model using MATLAB (contd…)

buck_conv_DVMC_simulation.m

```
……
sim('buck_converter_DVMC.slx');

t=buck_result.time; t_scale=t*1e3;
x=buck_result.data;  i_L=x(:,1);
V_cap=x(:,2); V_o=x(:,3);
V_s=x(:,4); V_n=x(:,5); Vr_d=x(:,6);
%% Plot subroutine
plot_buck_simulation;
```

Now, the next step is the simulation result I just showed this is coming from a switch simulation and the same thing we are talking about is the sample voltage here, sample voltage output voltage samples because this is a sample voltage.

(Refer Slide Time: 17:42)



Steps to Verify Large-Signal DT Model using MATLAB (contd…)

buck_DT_LSM_TE.m

```
clc;
buck_parameter;
Kp=10;  Ki=0.3;  Kd=20;
t_s=0.1*T;  V_m=10;  R1=1;
R2=0.05; R=R1; N_tran=500;

buck_DT_model_matrices;

……
```

buck_DT_model_matrices.m

```
alpha=R/(R+r_C);  r_e=(r_l+r_L);
T_s=t_s;  I_den=[1 0; 0 1];
%% Define system, input and output matrices
A_on=[-(r_e+(alpha*r_C))/L     -alpha/L;
                alpha/C        -alpha/(R*C)];
A_off=A_on;  B=[1/L; 0];
C_m=[r_C*alpha alpha];
```

$A_{on} = A_{off}$

$= \begin{bmatrix} -\frac{(r_e+\alpha r_c)}{L} & -\frac{\alpha}{L} \\ \frac{\alpha}{c} & -\frac{\alpha}{Rc} \end{bmatrix}$

$C_0 = [\alpha r_c \quad \alpha]$

NPTEL Online Certification Courses
IIT Kharagpur

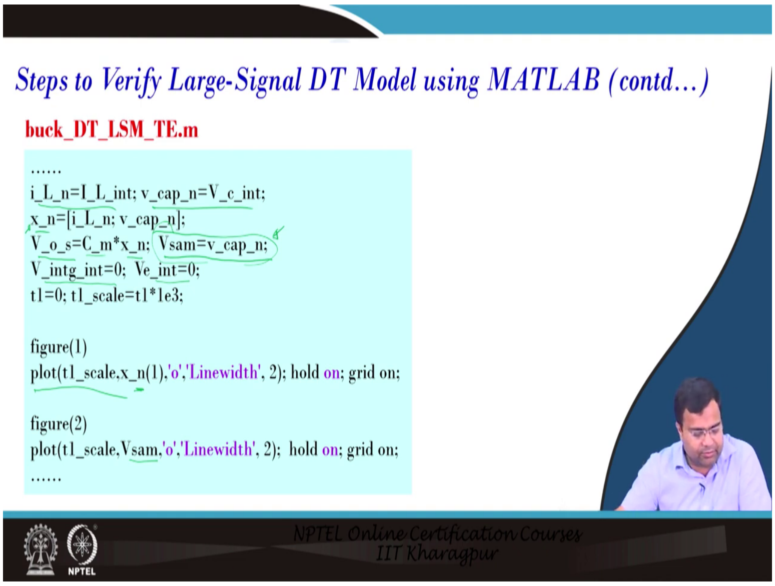
So, you can see there is a slight delay and I will be showing the case study, then we have to match the discrete-time large signal model.

So, how what are the steps? In the discrete-time large signal model, we will not close or clear the file, we may simply clear the I would say just the window command from, but we should not clear anything on the workspace. Then we will call the buck parameter, this part you can disable this because this part will be common it is taken from the switch simulation ok.

And then the next step I will go is it will take discrete time model matrices because you know any discrete time model you need on state A matrix of state. For buck converter synchronous buck if you assume that RDS 1 for both the high side and low side is the same. So, we are taking these two are equal.

And what is this? We have discussed that it is minus r e plus alpha r c divided by L minus alpha L alpha C minus alpha by R C and this is exactly what is the R n. And R of equal to R n for this case and the B matrix when the switch is on is 1 by L and 0 and when the switch is off it is 0. And then C m is the output matrix, which is the output matrix it is alpha r C and alpha ok.

(Refer Slide Time: 19:13)



Steps to Verify Large-Signal DT Model using MATLAB (contd…)

buck_DT_LSM_TE.m

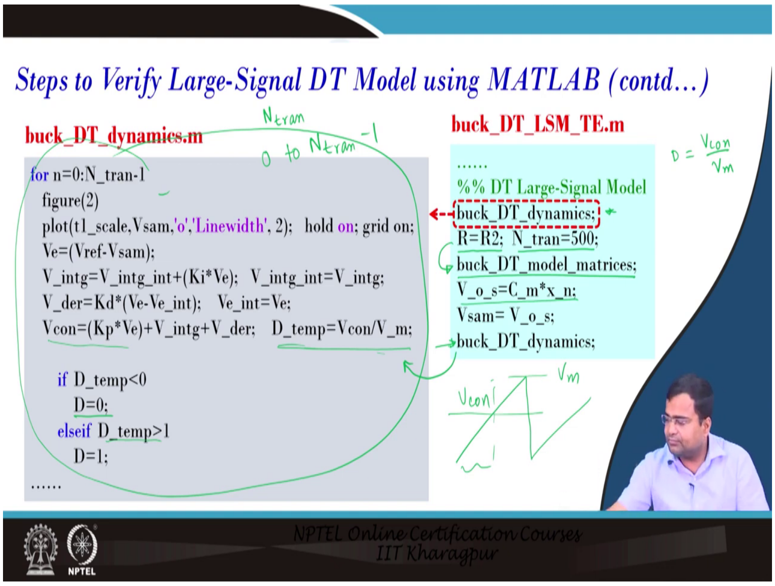
```
……
i_L_n=I_L_int; v_cap_n=V_c_int;
x_n=[i_L_n; v_cap_n];
V_o_s=C_m*x_n; Vsam=v_cap_n;
V_intg_int=0;  Ve_int=0;
t1=0; t1_scale=t1*1e3;

figure(1)
plot(t1_scale,x_n(1),'o','Linewidth', 2); hold on; grid on;

figure(2)
plot(t1_scale,Vsam,'o','Linewidth', 2);  hold on; grid on;
……
```

Next, it is continuing. So, we are also giving the initial step, it is taking from the previous value because we have defined this in our switch simulation. So, it has to take the same initial condition, then it will create a vector, then the sample voltage is nothing but the output matrix into the x n state vector and x n state vector initially loading with the initial condition. So, it is a sample voltage.

So, the sample voltage can also be like a v cap, another sample is the initialization of the sample voltage. This is the output sample instantaneous value and this is the initialization which can take v cap dot n and then we are setting the integral that we are setting 0 there. So, again the error voltage can also be defined, and so on scaling.

So, I am not going to explain each and every step, but now we are plotting time versus the first vector which is the inductor current and the sample is that, which is the output voltage, so here.

(Refer Slide Time: 20:21)



Steps to Verify Large-Signal DT Model using MATLAB (contd…)

buck_DT_dynamics.m

```
for n=0:N_tran-1
  figure(2)
  plot(t1_scale,Vsam,'o','Linewidth', 2);   hold on; grid on;
  Ve=(Vref-Vsam);
  V_intg=V_intg_int+(Ki*Ve);   V_intg_int=V_intg;
  V_der=Kd*(Ve-Ve_int);   Ve_int=Ve;
  Vcon=(Kp*Ve)+V_intg+V_der;   D_temp=Vcon/V_m;

    if D_temp<0
      D=0;
    elseif D_temp>1
      D=1;
......
```

buck_DT_LSM_TE.m

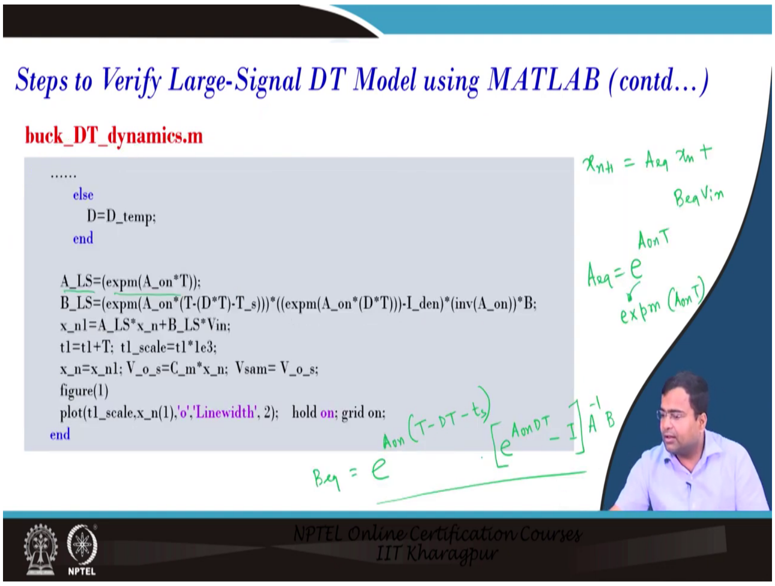
```
......
%% DT Large-Signal Model
buck_DT_dynamics;
R=R2;  N_tran=500;
buck_DT_model_matrices;
V_o_s=C_m*x_n;
Vsam= V_o_s;
buck_DT_dynamics;
```

So, here we can take, and then what will happen? Once the matrix is defined and is based on the load resistance then it will run the loop, which is a buck discrete-time dynamics. So, this is the model and the detail code is there, it will enter into a loop and it will run from; that means, a total N tran number of the cycle it has to execute.

So, it will start from 0 to N tran minus 1 ok. So, that is what exactly, then it will compute all these value control parameters, everything it will compute from there it will compute the duty ratio which is the V controlled by the V m, V m is the sawtooth that V m ok. So, this is the V m and this is the control voltage Vcon, then we can compute the duty ratio right. Because Vcon is to V m is our, the ratio if you take it will be the duty ratio, ok.

Because we know that duty ratio expression is nothing but Vcon by V m here. Then we have to check whether the temporary duty ratio is greater than or less than because of the f running cycle, if it is less than 0 then we have to set the actual duty ratio as 0, 0. The duty ratio cannot be negative and if it tends to become larger than 1, then you have to set it to 1 its maximum value can be 1.

(Refer Slide Time: 21:38)



Steps to Verify Large-Signal DT Model using MATLAB (contd…)

buck_DT_dynamics.m

```
……
    else
       D=D_temp;
    end

  A_LS=(expm(A_on*T));
  B_LS=(expm(A_on*(T-(D*T)-T_s)))*((expm(A_on*(D*T)))-I_den)*(inv(A_on))*B;
  x_n1=A_LS*x_n+B_LS*Vin;
  t1=t1+T; t1_scale=t1*1e3;
  x_n=x_n1; V_o_s=C_m*x_n; Vsam= V_o_s;
  figure(1)
  plot(t1_scale,x_n(1),'o','Linewidth', 2);   hold on; grid on;
end
```

$x_{n+1} = A_{eq} x_n + B_{eq} V_{in}$

$A_{eq} = e^{A_{on}T}$

expm $(A_{on}T)$

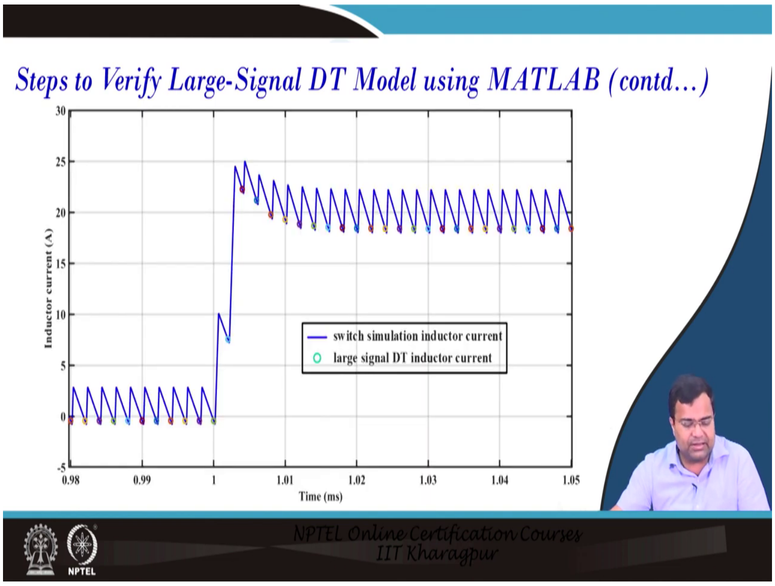
$B_{eq} = e^{A_{on}(T-DT-t_s)} \left[e^{A_{on}DT} - I\right] A^{-1} B$

NPTEL Online Certification Courses
IIT Kharagpur

And then the run, the loop will continue and then it will this is the model we want to show the large signal; that means, we know that A x n plus 1 is that A equivalent into x n plus B equivalent into V in right. What is an equivalent for the buck converter? We know that A equivalent is equal to e to the power A on T and this is exactly what for an exponential case study it will be a written exam, A on T in the MATLAB code. So, it will represent this in MATLAB.

Similarly, the B equivalent, you know if you see here; what is the B equivalent here? So, B equivalent will be e to the power A on T minus DT minus t s, this here inside you can see e to the power A on DT minus identity matrix into A inverse into B. So, this is the B equivalent matrix.

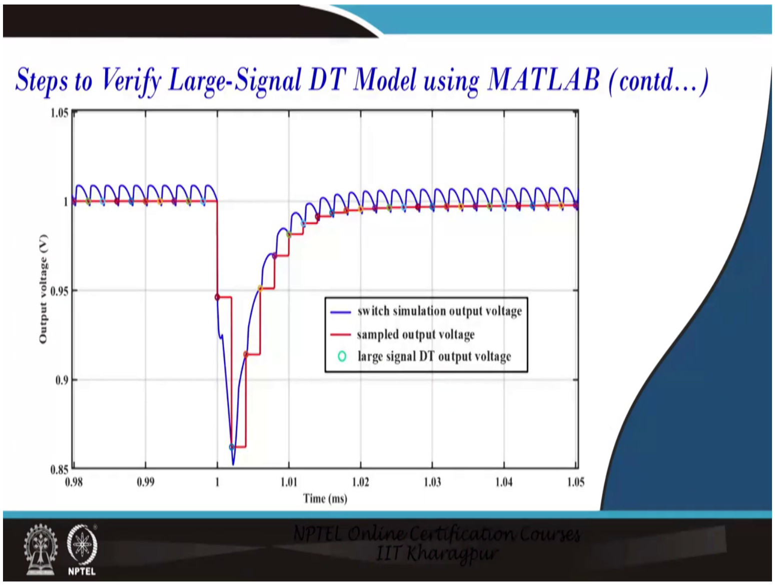
(Refer Slide Time: 22:53)


Steps to Verify Large-Signal DT Model using MATLAB (contd…)

NPTEL Online Certification Courses
IIT Kharagpur

So, we are plugged in and then it is running and this is the demonstration of the switch simulation.

(Refer Slide Time: 22:55)


Steps to Verify Large-Signal DT Model using MATLAB (contd…)

NPTEL Online Certification Courses
IIT Kharagpur

You can see, if you see once it runs for 500 cycles then we are changing the load resistance from R to R 2; that means, initially it was R 1 now it is R 2.

Then it will again run for 500 cycles and we are calling again the model matrix because the model matrix has to be updated with the load new load resistance. Then we are again taking the sample value and then we are calling the buck function, a dynamic function which is again the same loop that will run, but only with an updated A matrix. Then the same thing we will we have discussed and then it will validate the model.

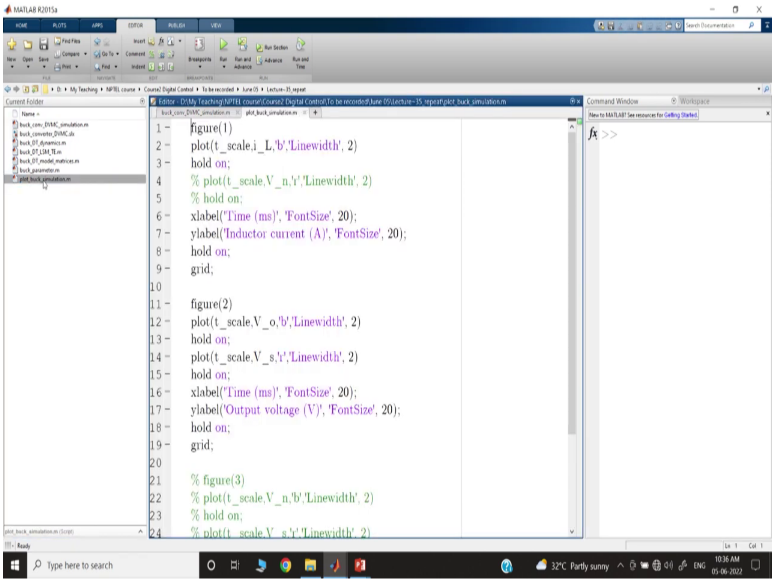
So, here we want to first show the MATLAB case study, let us go to the MATLAB model. So, again we have discussed the overall MATLAB model, this is a load block that we have discussed. Now, we want to first show that this is our switch simulation. That means, the first thing we are loading is the buck parameter, this set the synchronous mode synchronous configuration running for 500 cycles, and the total run time of the simulation time is 2 times 500 cycles into 2 into time period T.

So, it is running for so time period we have taken 2 microsecond. So, the total number of cycles is 500 into 2 so 1000. So, 1000 into 2 microsecond we are running for 2 millisecond. So, the total simulation time is 2 millisecond for this case, we have set some value Kp to be 20, you can say 10 whatever you want no problem.

So, you can set 10 also then we are setting a delay time of 10; that means, 10 percent of the time period. That means, it is T by 10 and we want to see the ramp voltage magnitude peak value we have taken 10 volt, initial rotation 1 ohm then the final rotation is 0.05.

And then we are initializing the inductor current and the capacitor voltage and then we are initializing the sample value as the same as you know which is the capacitor voltage, and initial voltage. And the integral term initial condition we are setting to 0, then we are simulating Simulink that code, and then we are storing the data that we have discussed and the plot command.

(Refer Slide Time: 25:09)

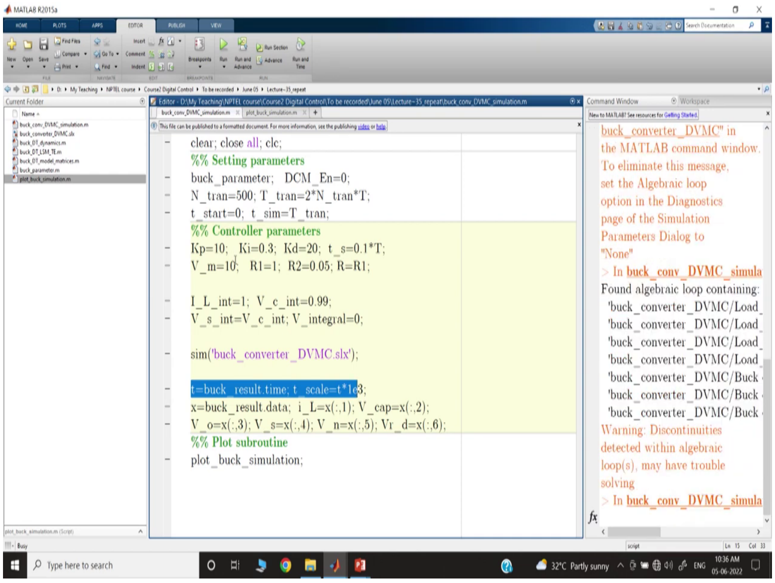


Editor - D:\My Teaching\NPTEL course\Course2 Digital Control\To be recorded\June 05\Lecture-35_repeat\plot_buck_simulation.m

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

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

% figure(3)
% plot(t_scale,V_n,'b','Linewidth', 2)
% hold on;
% plot(t_scale,V_s,'r','Linewidth', 2)
```

And if you go to the plot command, you will see that in figure 1 we are plotting the inductor current, we are just labeling it. Then figure 2, it is plotting the output voltage as well as the sample voltage. So, this is a sample voltage and we are leveling this.

(Refer Slide Time: 25:26)



Editor - D:\My Teaching\NPTEL course\Course2 Digital Control\To be recorded\June 05\Lecture-35_repeat\buck_conv_DVMC_simulation.m

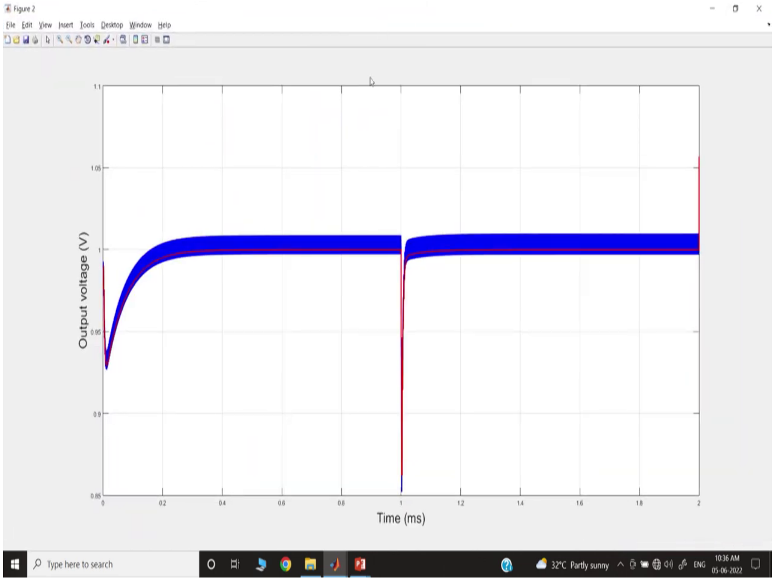
```
clear; close all; clc;
%% Setting parameters
buck_parameter;   DCM_En=0;
N_tran=500; T_tran=2*N_tran*T;
t_start=0; t_sim=T_tran;
%% Controller parameters
Kp=10;  Ki=0.3;  Kd=20;  t_s=0.1*T;
V_m=10;  R1=1;  R2=0.05; R=R1;

I_L_int=1;  V_c_int=0.99;
V_s_int=V_c_int; V_integral=0;

sim('buck_converter_DVMC.slx');

t=buck_result.time; t_scale=t*1e3;
x=buck_result.data;  i_L=x(:,1); V_cap=x(:,2);
V_o=x(:,3); V_s=x(:,4); V_n=x(:,5); Vr_d=x(:,6);
%% Plot subroutine
plot_buck_simulation;
```

Command Window

```
buck_converter_DVMC" 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_conv_DVMC_simula
Found algebraic loop containing:
  'buck_converter_DVMC/Load_
  'buck_converter_DVMC/Load_
  'buck_converter_DVMC/Load_
  'buck_converter_DVMC/Load_
  'buck_converter_DVMC/Buck
  'buck_converter_DVMC/Buck
  'buck_converter_DVMC/Buck
Warning: Discontinuities
detected within algebraic
loop(s), may have trouble
solving
> In buck_conv_DVMC_simula
```

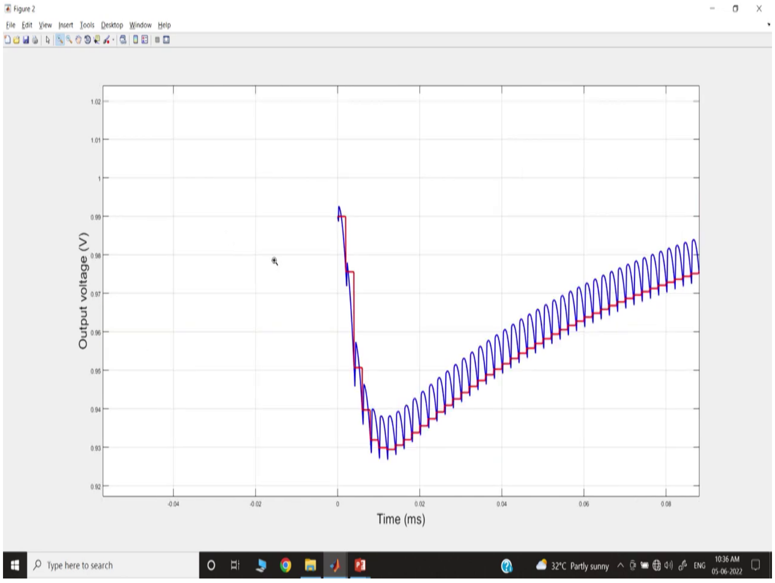
So, let us run the case study you know this is the closed-loop digital voltage mode control and we want to see what happened.

(Refer Slide Time: 25:29)

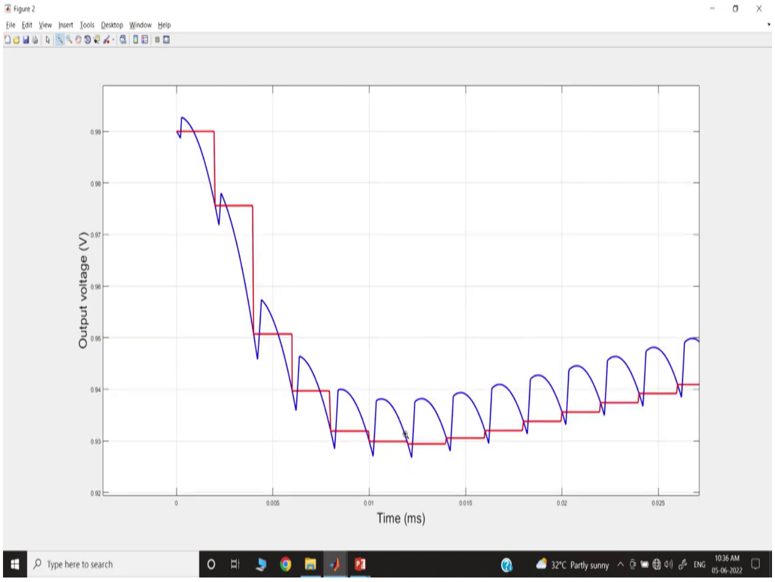


So, here I have shown it is just a switch simulation.

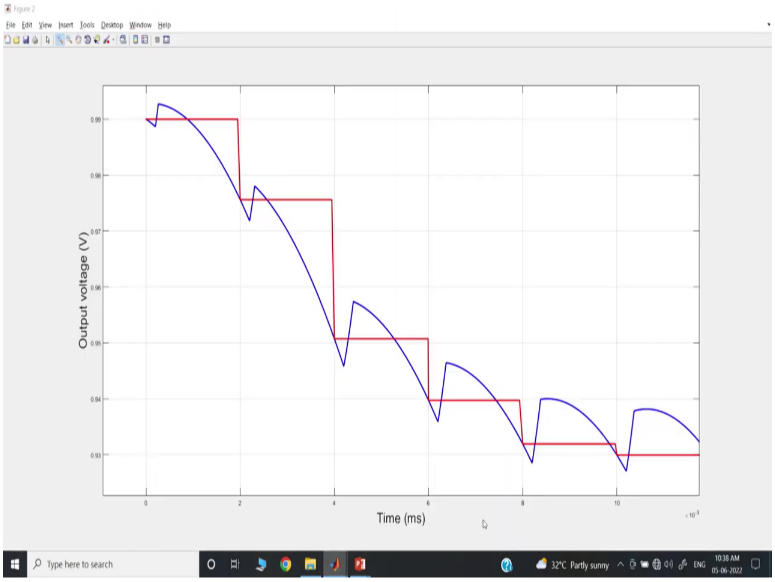
(Refer Slide Time: 25:37)



(Refer Slide Time: 25:39)

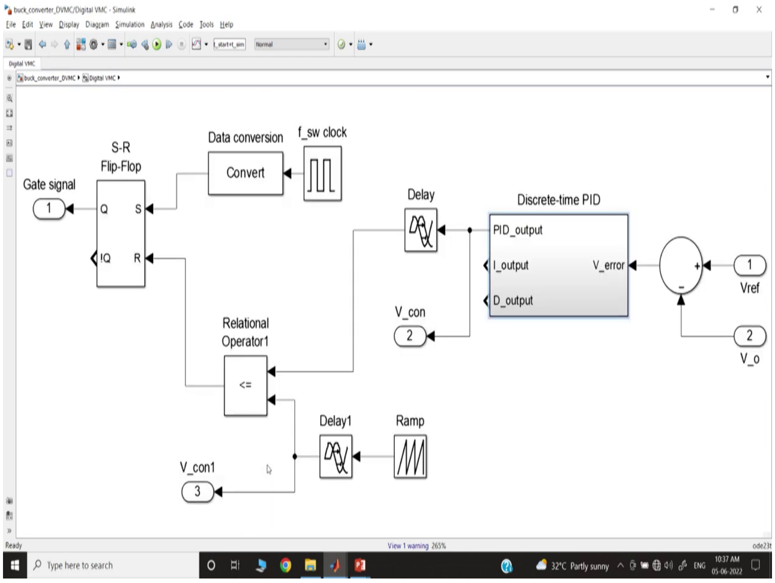


(Refer Slide Time: 25:41)

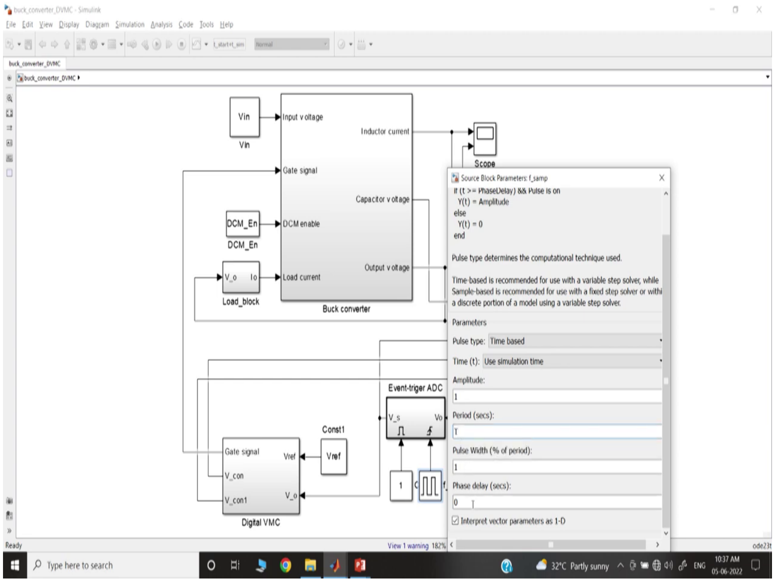


And I just want to show that from the very beginning that we are first running the simulation and the simulation starts with a sample. That means, if we first sample and then we will go to, I would say you know the switching action.

(Refer Slide Time: 25:56)



(Refer Slide Time: 26:03)



Source Block Parameters: f_samp

```
if (t >= PhaseDelay) && Pulse is on
  Y(t) = Amplitude
else
  Y(t) = 0
end
```

Pulse type determines the computational technique used.

Time-based is recommended for use with a variable step solver, while Sample-based is recommended for use with a fixed step solver or withi a discrete portion of a model using a variable step solver.

Parameters

Pulse type: Time based

Time (t): Use simulation time

Amplitude: 1

Period (secs): T

Pulse Width (% of period): 1

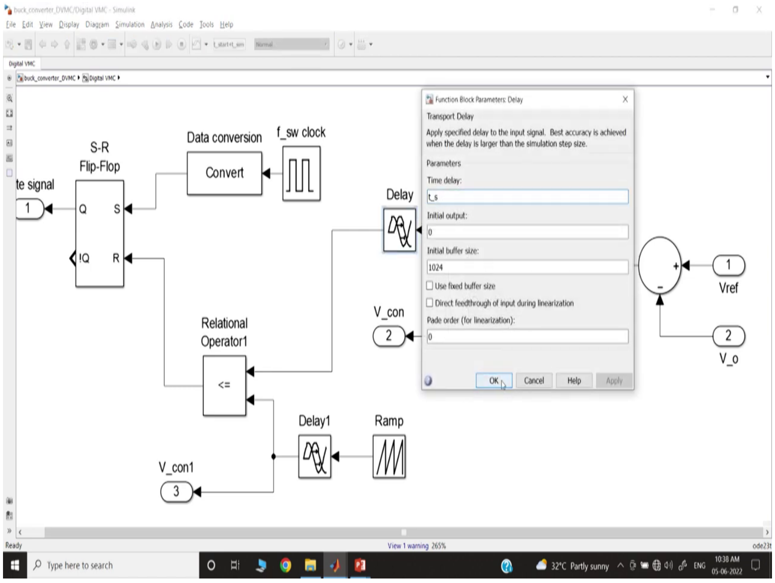
Phase delay (secs): 0

☑ Interpret vector parameters as 1-D

That means, if you go to the Simulink model, I want to show you that you can see here that since it is a sample so, it is using this sampling clock there is no delay. That means, the clock is the period T, which is the time period switching period and we are taking the uniform sampling the sampling rate is the same as the switching frequency. So, the time period is T and there is no delay here.

The next part is if you go inside. So, up to this point, there is no delay; that means, it is computing the error voltage as per the running sample, then if you go inside it is the discrete-time integral control. And this is the discrete-time derivative control and we have already discussed this and this is a discrete-time proportional control. And then if you add it up you will get a PID controller output and this output is going.

(Refer Slide Time: 26:44)



Function Block Parameters: Delay

Transport Delay

Apply specified delay to the input signal. Best accuracy is achieved when the delay is larger than the simulation step size.

Parameters

Time delay: t_s

Initial output: 0

Initial buffer size: 1024

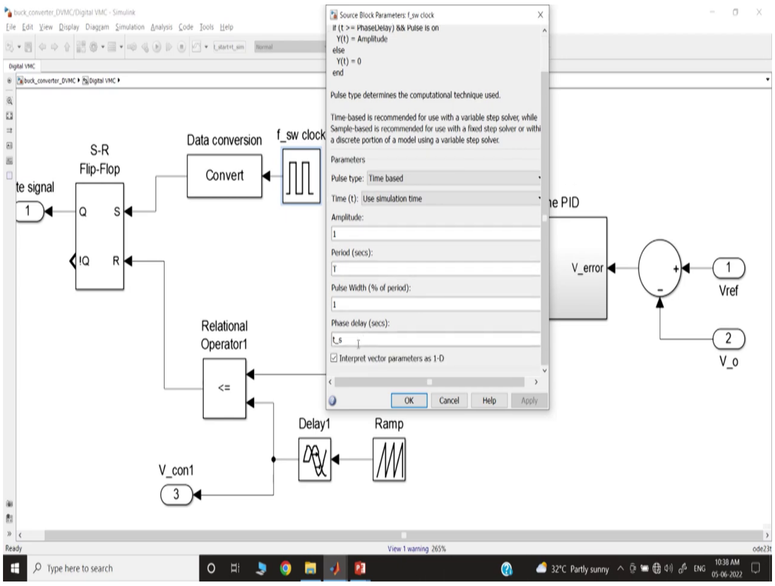
□ Use fixed buffer size

□ Direct feedthrough of input during linearization

Pade order (for linearization): 0

Now, we are delaying, if you go inside we are delaying by T s amount of time and I told you the same delay should be applicable for the ramp.

(Refer Slide Time: 26:52)



Source Block Parameters: f_sw clock

```
if (t >= PhaseDelay) && Pulse is on
  Y(t) = Amplitude
else
  Y(t) = 0
end
```

Pulse type determines the computational technique used.

Time-based is recommended for use with a variable step solver, while Sample-based is recommended for use with a fixed step solver or withi a discrete portion of a model using a variable step solver.

Parameters

Pulse type: Time based

Time (t): Use simulation time

Amplitude: 1

Period (secs): T

Pulse Width (% of period): 1

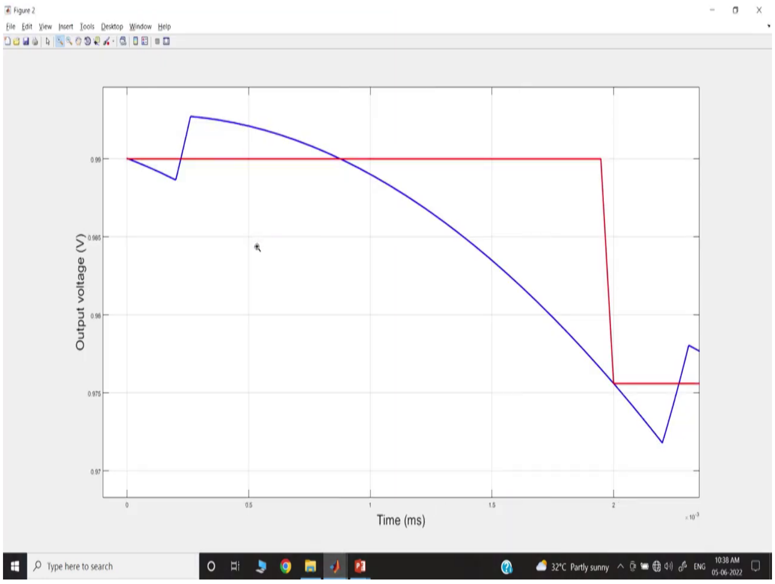
Phase delay (secs): t_s

☑ Interpret vector parameters as 1-D

And for the clock the delay is set here; so you can see the delay here. So that means, all the switching clock is a delayed versions of the sampling clock, which is the time period is the same, but the T s amount of delay. The ramp signal is also delayed and then the output of the PID controller is also delayed.

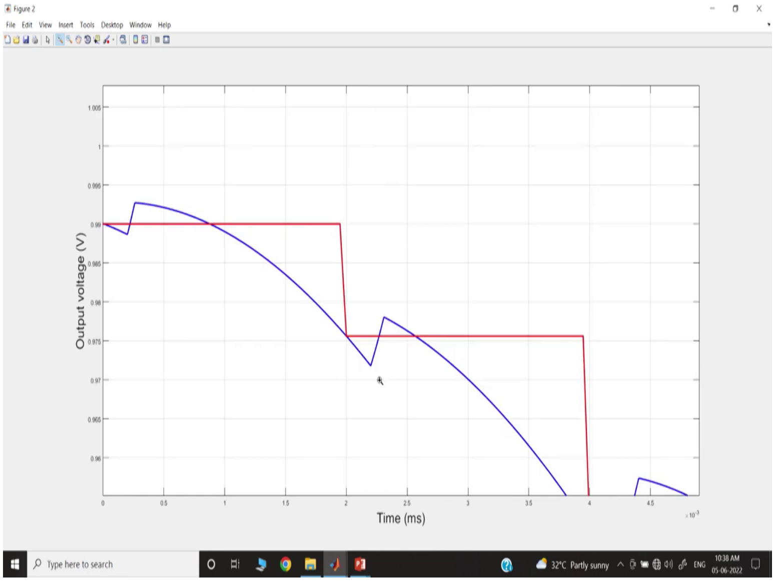
Now, we have shown the case study. So, in this case, a study is what is reflected. So, if you initially sample then it is coming.

(Refer Slide Time: 27:17)



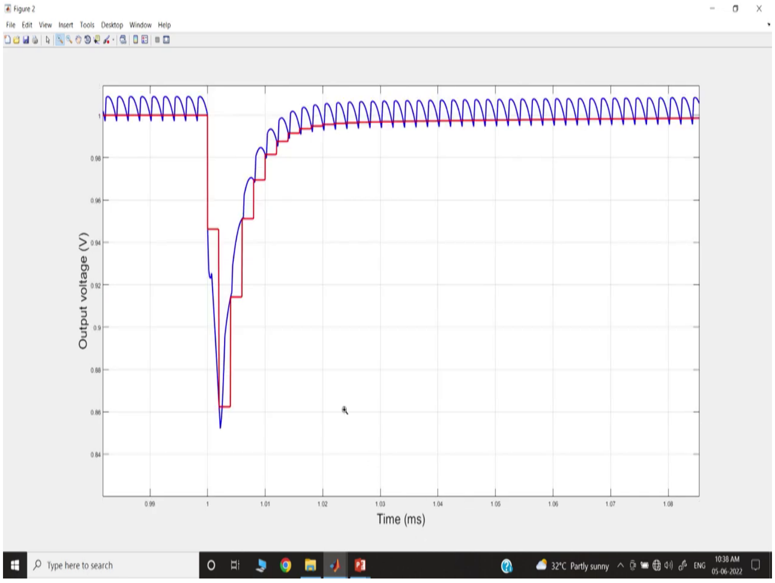
So, it is getting updated every two microsecond, you see 0 2. The samples are updated.

(Refer Slide Time: 27:23)

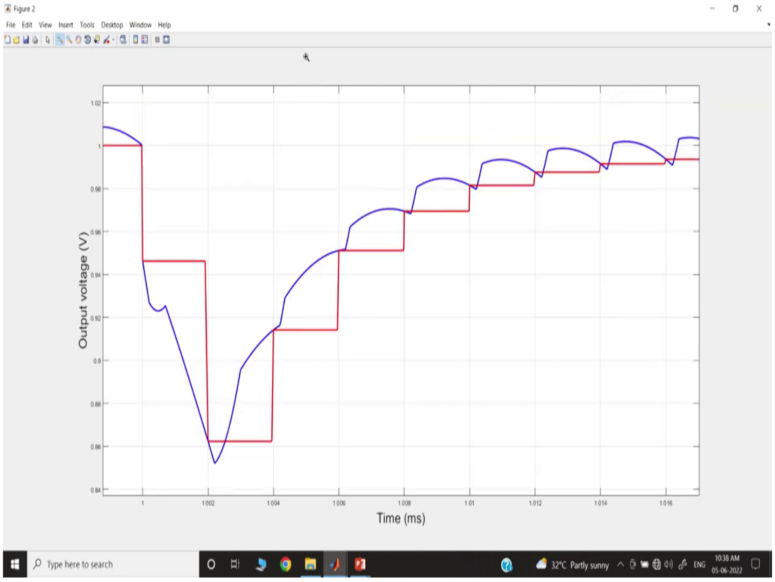


But the switching is happening a little later because we are switching.

(Refer Slide Time: 27:30)

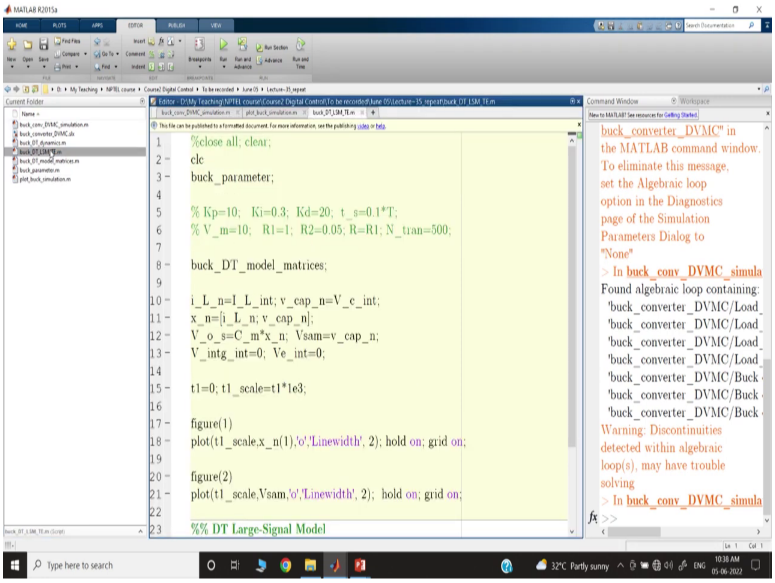


(Refer Slide Time: 27:32)



Then, if you go to the transient case you can see that this is where the load transient is happening ok.

(Refer Slide Time: 27:40)

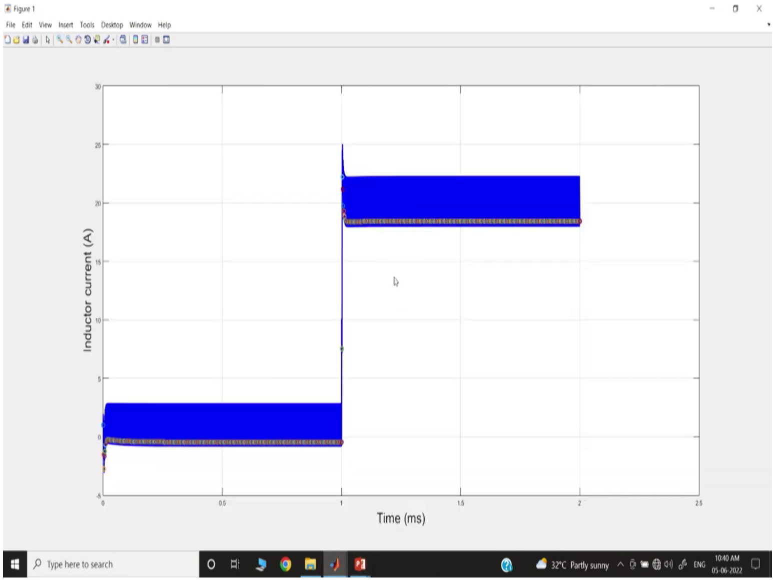


```
%close all; clear;
clc
buck_parameter;

% Kp=10;   Ki=0.3;  Kd=20;  t_s=0.1*T;
% V_m=10;   R1=1;  R2=0.05; R=R1; N_tran=500;

buck_DT_model_matrices;

i_L_n=I_L_int; v_cap_n=V_c_int;
x_n=[i_L_n; v_cap_n];
V_o_s=C_m*x_n;  Vsam=v_cap_n;
V_intg_int=0;  Ve_int=0;

t1=0; t1_scale=t1*1e3;

figure(1)
plot(t1_scale,x_n(1),'o','Linewidth', 2); hold on; grid on;

figure(2)
plot(t1_scale,Vsam,'o','Linewidth', 2);  hold on; grid on;

%% DT Large-Signal Model
```

```
buck_converter_DVMC" 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_conv_DVMC_simula
Found algebraic loop containing:
  'buck_converter_DVMC/Load_
  'buck_converter_DVMC/Load_
  'buck_converter_DVMC/Load_
  'buck_converter_DVMC/Load_
  'buck_converter_DVMC/Buck
  'buck_converter_DVMC/Buck
  'buck_converter_DVMC/Buck
Warning: Discontinuities
detected within algebraic
loop(s), may have trouble
solving
> In buck_conv_DVMC_simula
fx >>
```

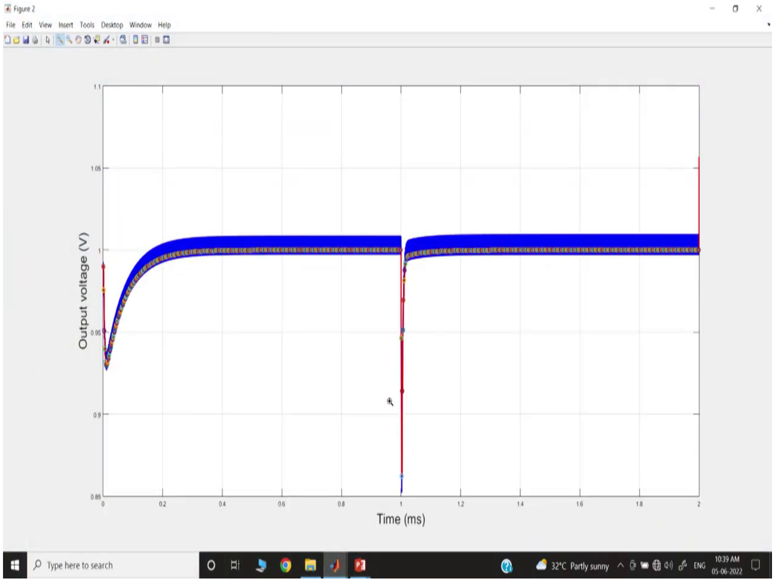
Next, we want to run the discrete-time model. So, you go to the discrete-time large signal model, and since we are using the same controller gain. So, you just commented on this line and let us run the case study.

(Refer Slide Time: 27:51)

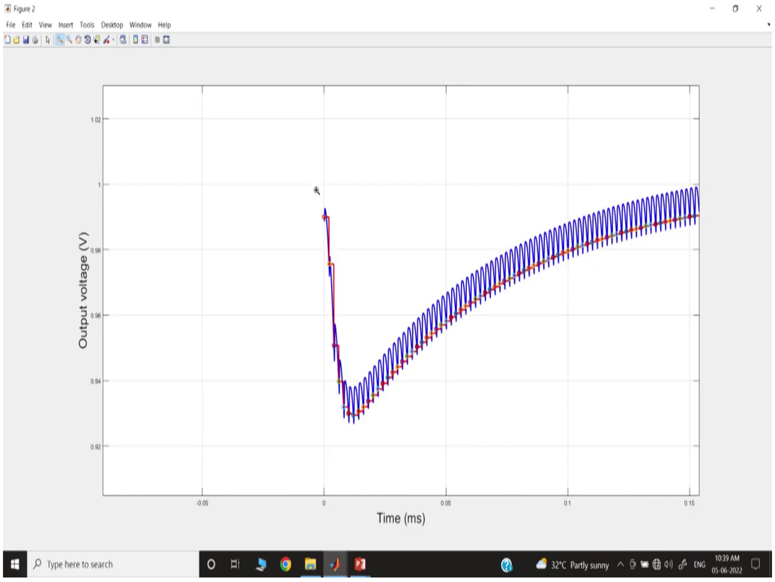


So, if you run it, then I want to show that let us go to the output voltage plot first.

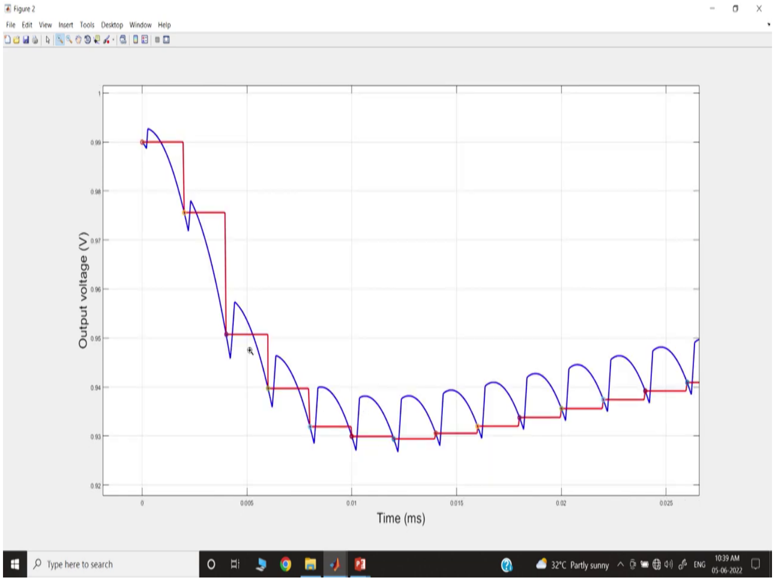
(Refer Slide Time: 27:56)



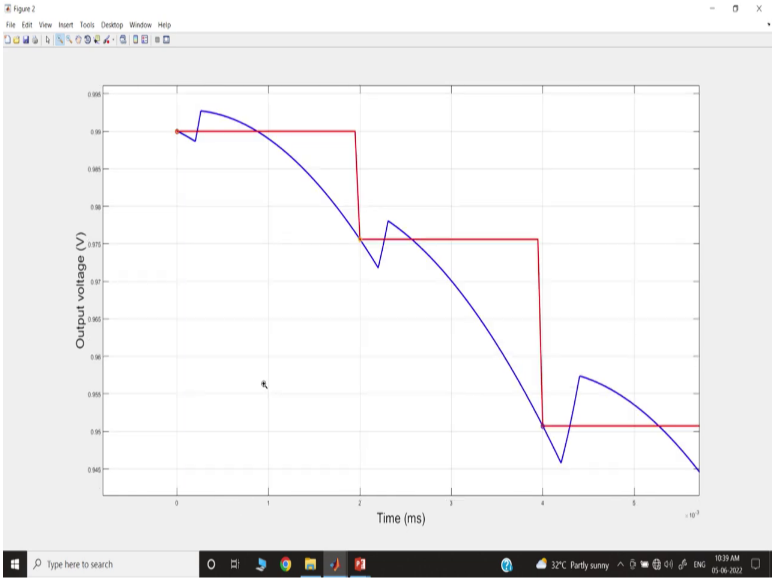
(Refer Slide Time: 27:59)



(Refer Slide Time: 28:01)

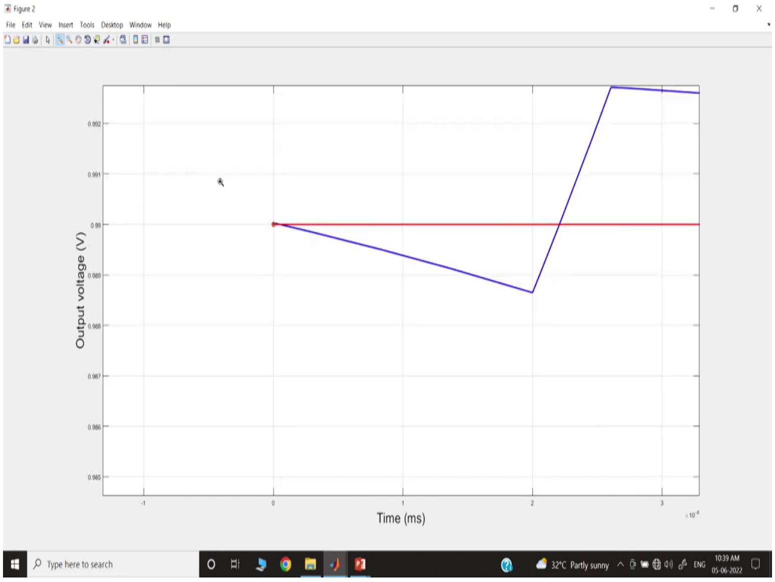


(Refer Slide Time: 28:03)

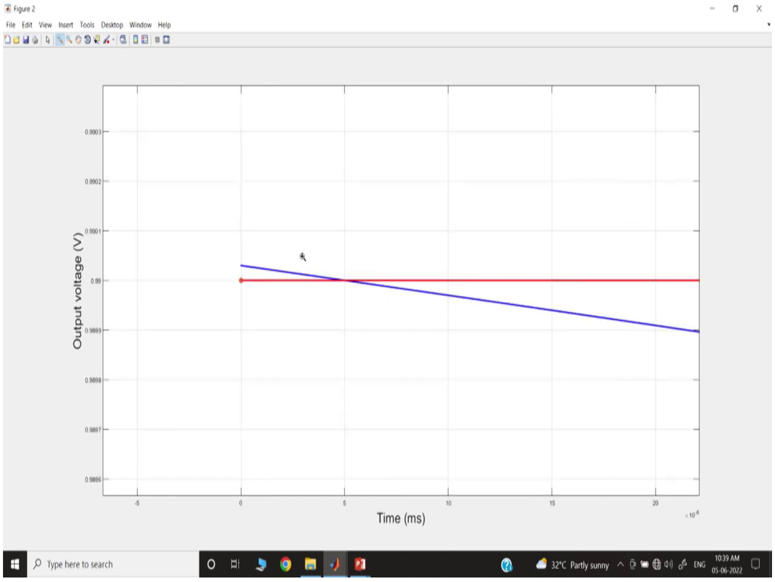


So, output voltage plot if you go, you see our discrete-time models are derived between two subsequent samples.

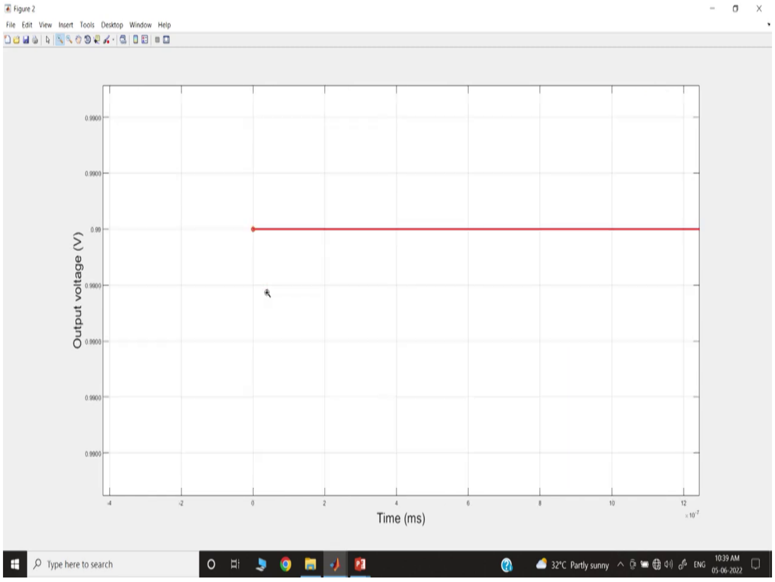
(Refer Slide Time: 28:08)



(Refer Slide Time: 28:10)

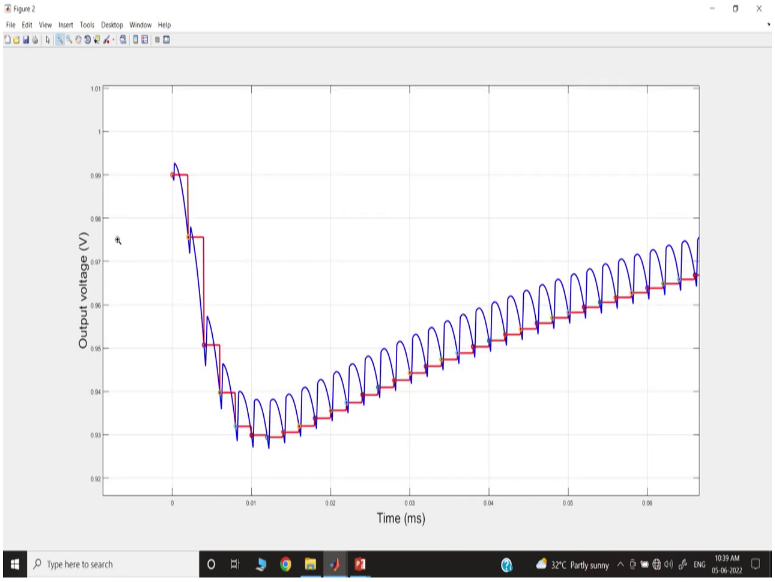


(Refer Slide Time: 28:12)



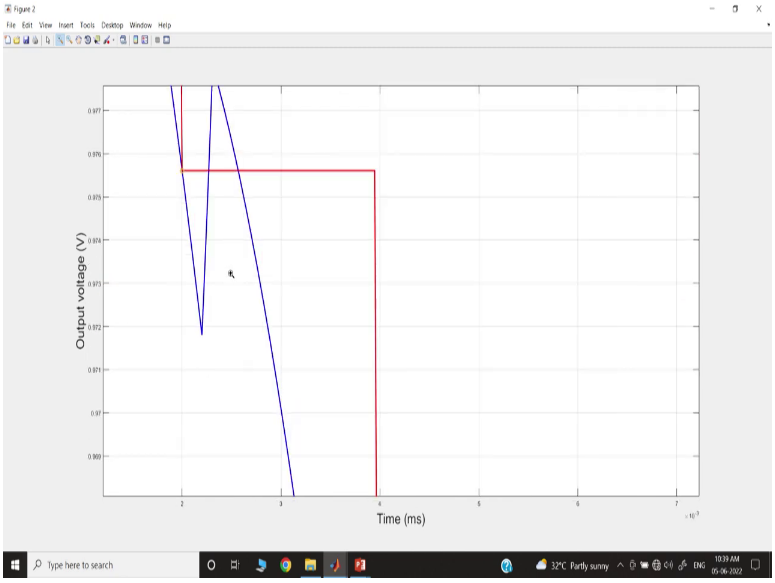
So, every time, whenever we are sampling it, it exactly matches with the discrete-time model and the sample values are matching perfectly, ok.

(Refer Slide Time: 28:16)



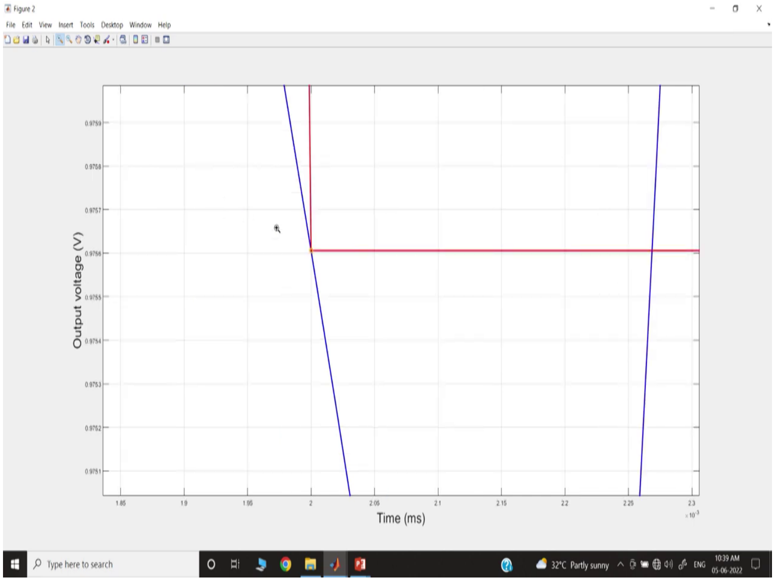
That means, they are matching perfectly.

(Refer Slide Time: 28:19)

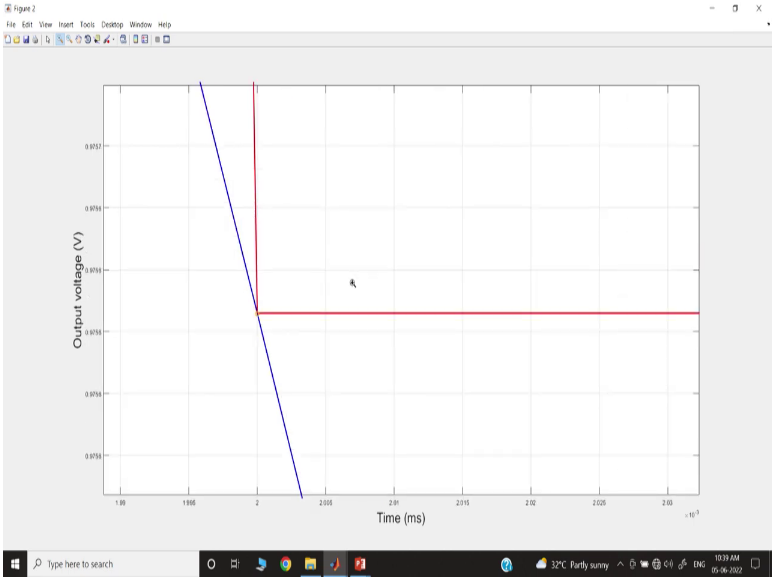


And you can see that it is taking an identical voltage.

(Refer Slide Time: 28:20)

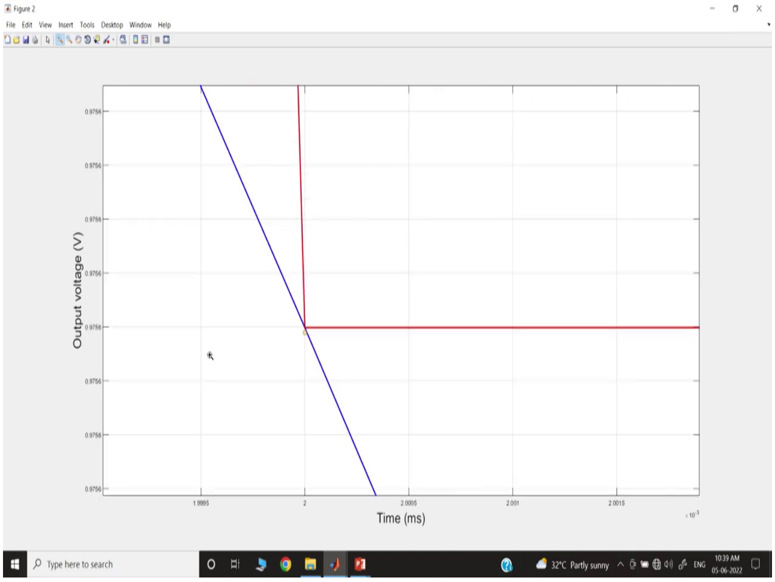


(Refer Slide Time: 28:22)

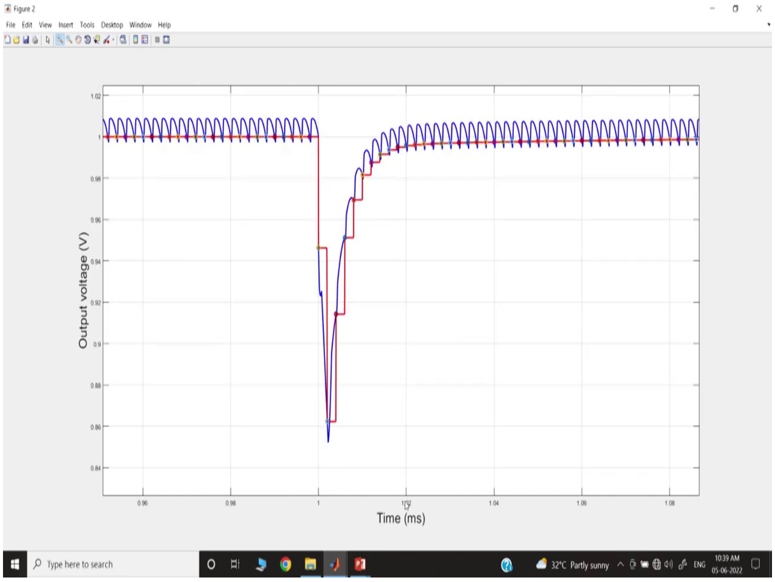


So, they are perfectly matching ok.

(Refer Slide Time: 28:23)

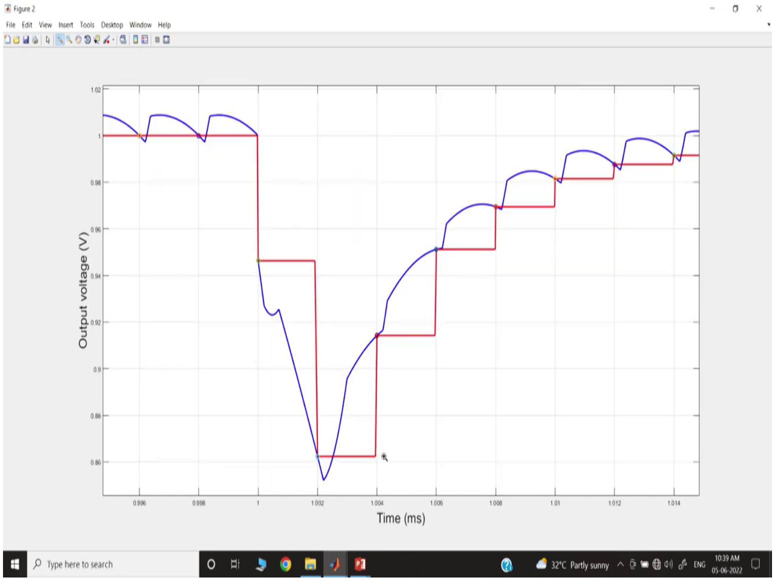


(Refer Slide Time: 28:27)



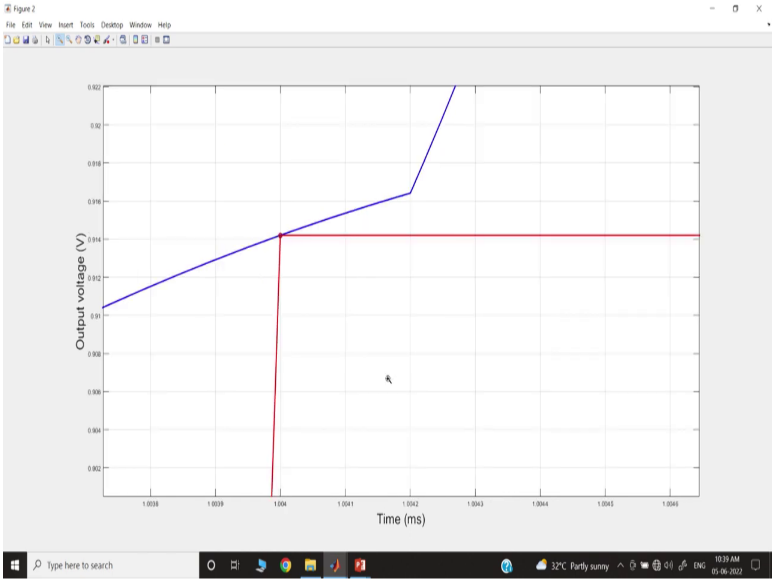
The same thing is happening even during the transient.

(Refer Slide Time: 28:28)



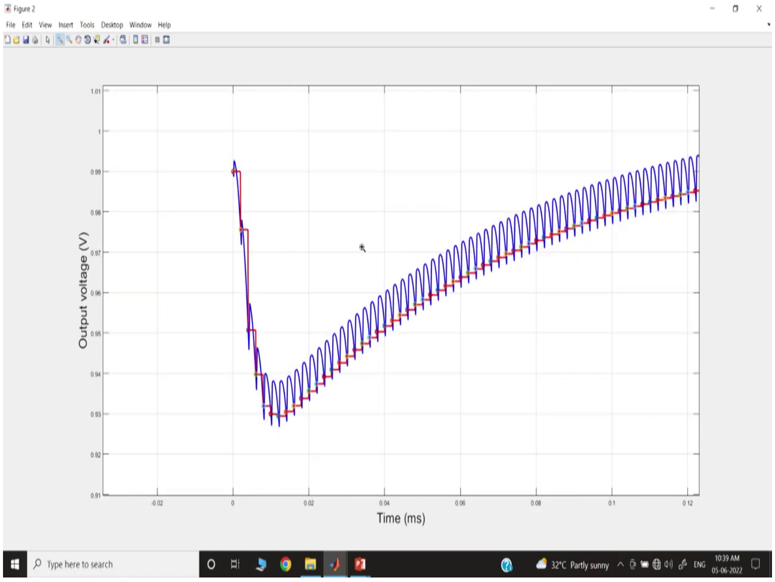
You see during transient, they are exactly matching.

(Refer Slide Time: 28:33)

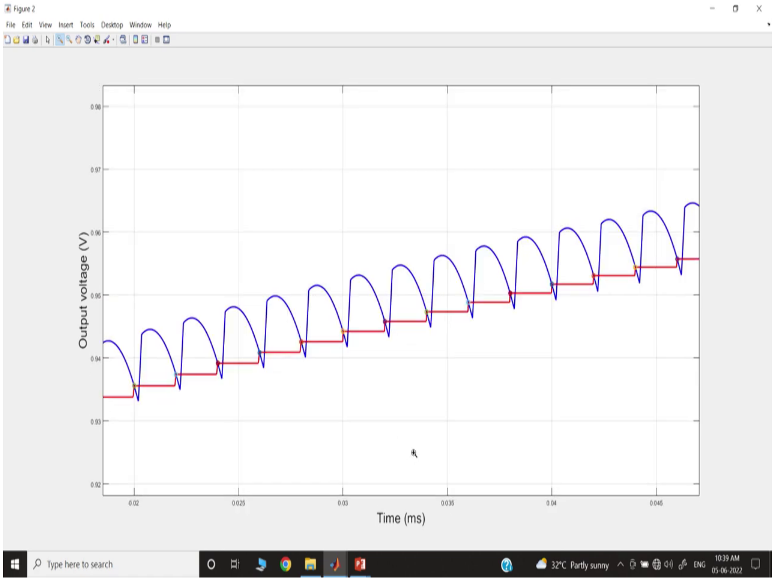


That means the sample which is captured through switch simulation and the discrete-time model at every sampling instant, are identically matching, ok.

(Refer Slide Time: 28:41)

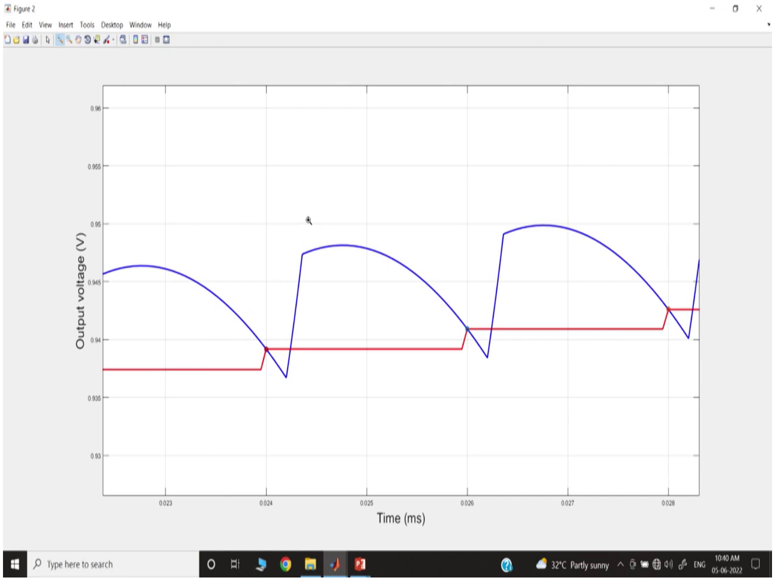


(Refer Slide Time: 28:43)



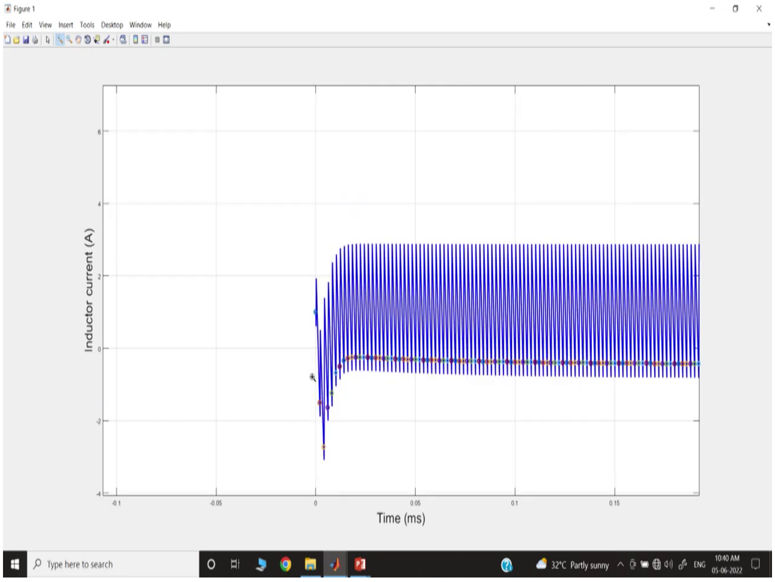
That means the model is perfectly valid.

(Refer Slide Time: 28:45)

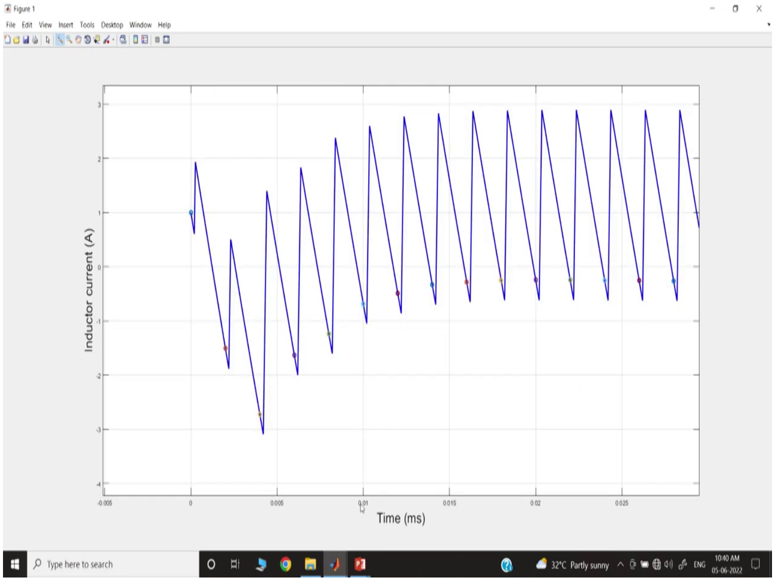


And you can see there is a; so if you go to the inductor current waveform, I want to show that if you take the inductor current.

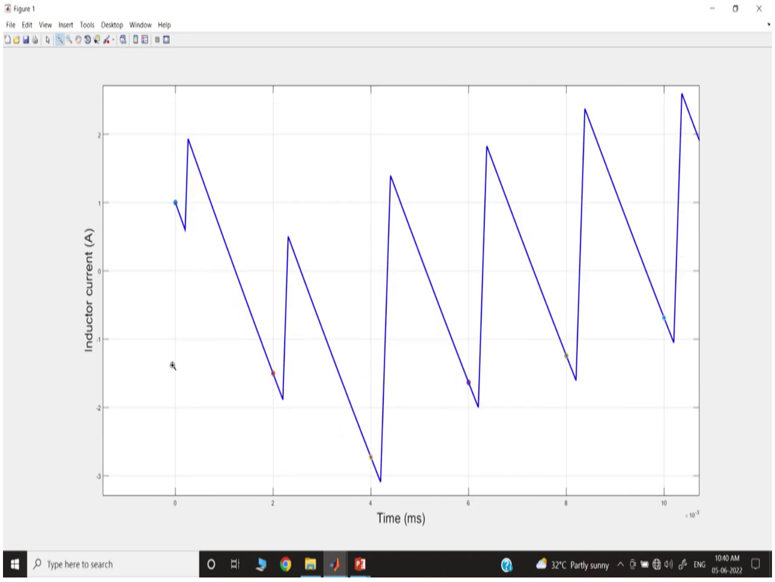
(Refer Slide Time: 28:52)



(Refer Slide Time: 28:53)

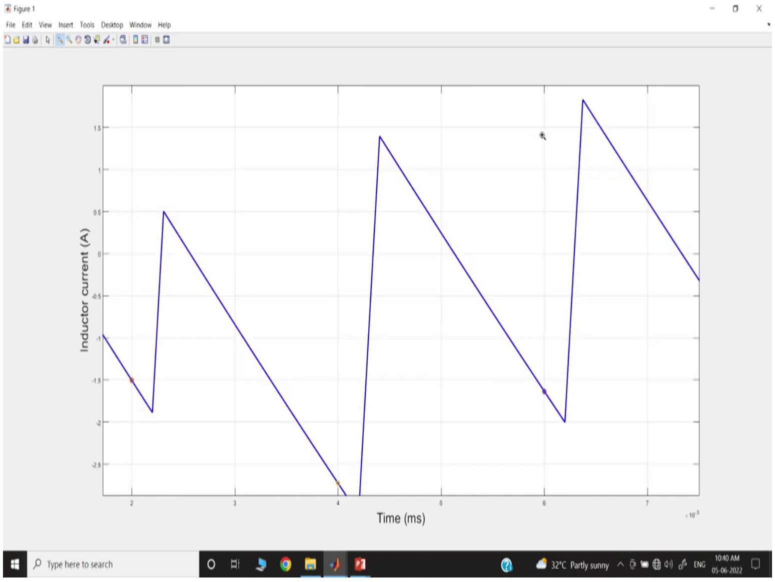


(Refer Slide Time: 28:56)



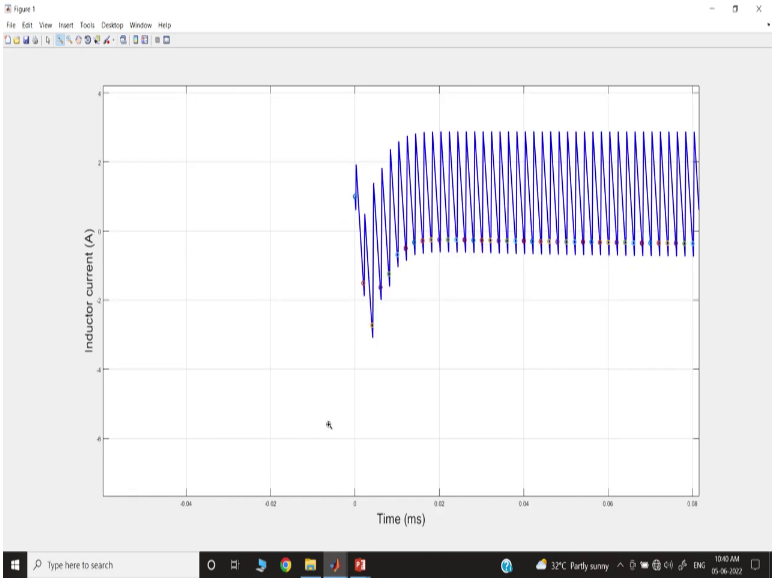
Again if you run through this, you see it is matching from the very beginning; that means, it is matching like this every cycle.

(Refer Slide Time: 29:01)

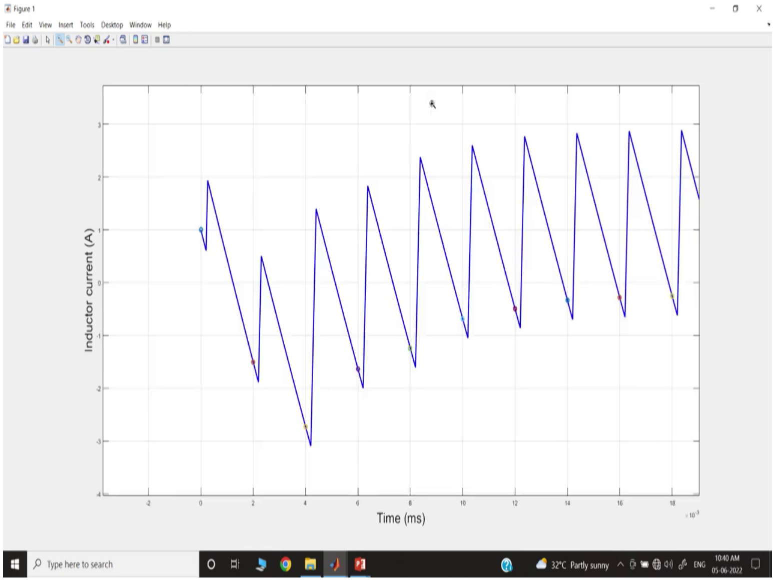


And the switching is happening after some delay.

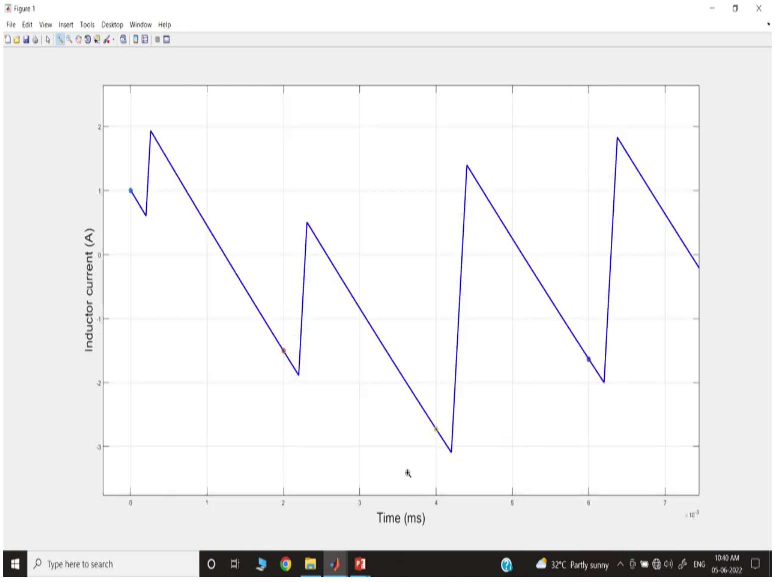
(Refer Slide Time: 29:04)



(Refer Slide Time: 29:06)

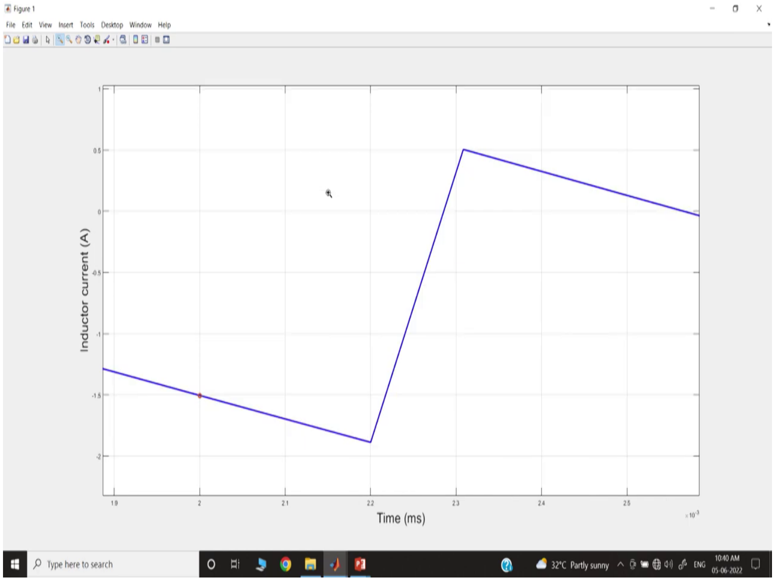


(Refer Slide Time: 29:08)



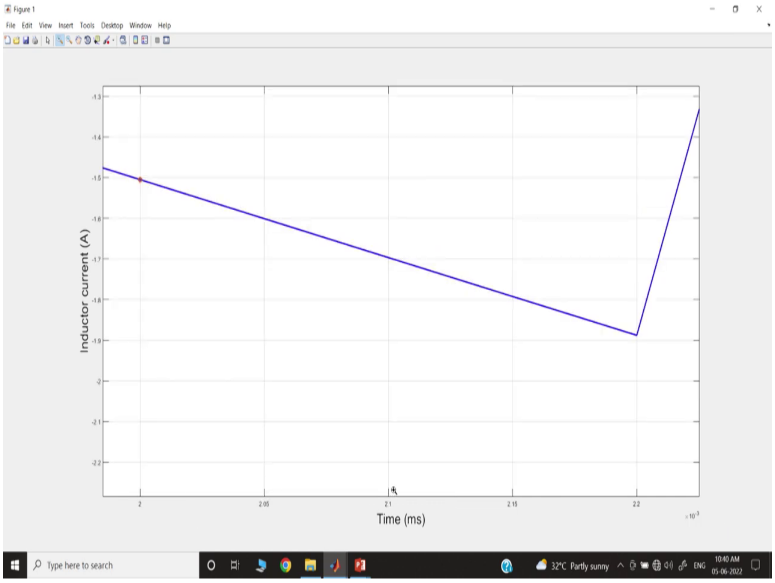
That means, you know we have discussed that if you consider interval two sampling with the T s amount of delay.

(Refer Slide Time: 29:11)

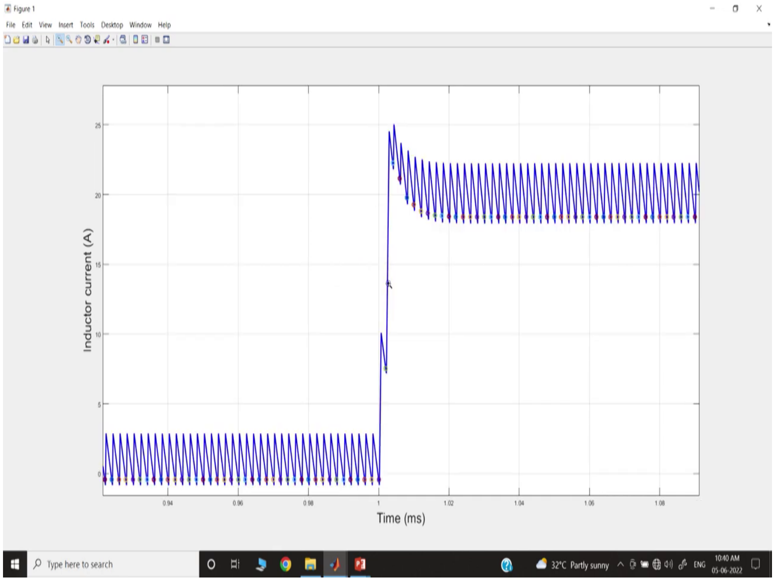


So, it is delayed and this delay amount we have set to 20 nanoseconds, 200 nanoseconds. So, you can see 0.2 millisecond means 200 nanosecond, sorry here we have taken sorry our time period as the actual time period is 2 microsecond. So, that by 10 200, it is yeah point 2.2 microsecond. So, it is 0.2 microsecond; that means, 200 nanosecond, yes.

(Refer Slide Time: 29:34)

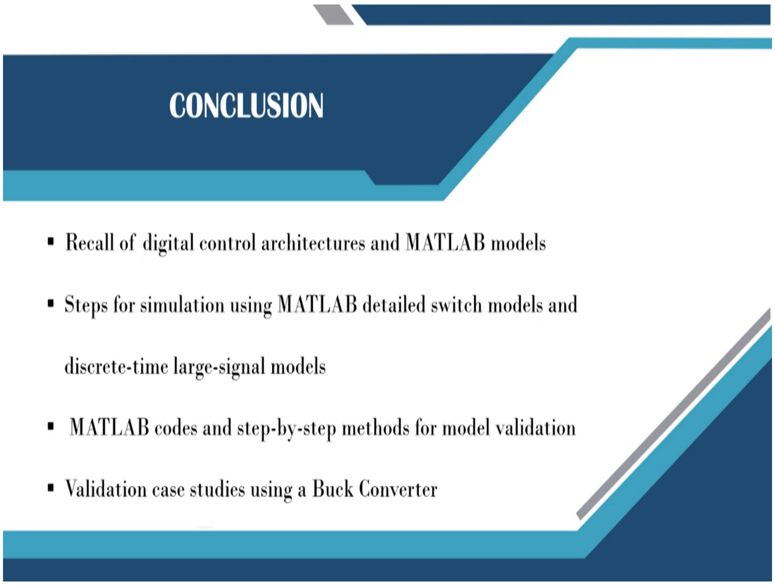


(Refer Slide Time: 29:37)



So, it is that perfectly matching the inductor current is matching perfectly. So, that means, what we can say is that our you know investigation; that means, the matching is perfectly happening.

(Refer Slide Time: 29:48)



CONCLUSION

- Recall of digital control architectures and MATLAB models
- Steps for simulation using MATLAB detailed switch models and discrete-time large-signal models
- MATLAB codes and step-by-step methods for model validation
- Validation case studies using a Buck Converter

So, in summary, we have you know recall we have discussed the digital control architecture, particularly voltage mode control architecture, and MATLAB model. We have discussed the step for simulation using MATLAB detailed switch model and discrete time large signal model.

Then MATLAB course and step-by-step method for model validations are discussed and we have also considered some simulation case studies using a buck converter; that is it for today.

Thank you very much.