

Control and Tuning Methods in Switched Mode Power Converters
Prof. Santanu Kapat
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Module - 12
Performance Comparison and Simulation
Lecture - 55

Small-Signal vs. Large-signal Tuning: Comparison Using MATLAB Simulation

Welcome, this is lecture number 55. In this lecture we are going to talk about Small-signal and Large-signal based Tuning and their Comparison Using MATLAB Simulation.

(Refer Slide Time: 00:36)

Concepts Covered

- Small-signal based tuning of VMC and CMC in a buck converter
- Large-signal based tuning of VMC and CMC in a buck converter
- Small-signal based tuning of VMC and CMC in a boost converter
- Large-signal tuning of CMC in a boost converter

NPTEL

So, here we will first talk about small-signal based design and tuning of voltage mode and current mode control in a buck converter. Then we will talk about large-signal based tuning of voltage mode current mode control; that means, both are in buck converter. Then we will talk about small-signal based tuning of voltage mode and current mode control in a boost converter and then, finally, large-signal tuning in a current mode control boost converter.

(Refer Slide Time: 00:59)

Small-Signal and Large-Signal based Tuning of a Buck Converter

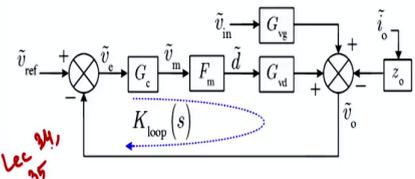
- Small-signal based tuning of VMC in a buck converter
- Small-signal based tuning of CMC in a buck converter
- Large-signal based tuning of CMC in a buck converter
- Large-signal based tuning of VMC in a buck converter



So, if we recall I am not going to again because we have already discussed the tuning methodology small-signal as well as large-signal in sufficient detail. So, first in the buck converter we will take a voltage mode control, current mode control both are designed using small-signal model. Then, in a large-signal model, we have designed both current mode and voltage mode control.

(Refer Slide Time: 01:22)

Type-III Compensator Design in VMC Buck Converter: Summary



Lec 34, 35



So, if you recall our voltage mode control design you know. So, we this is the loop transfer function that we have discussed multiple time you know you will get more detail in lecture number if you go to lecture number 34, 35 both have like an extensive design.

(Refer Slide Time: 01:45)

Type-III Compensator Design in VMC Buck Converter: Summary

Lec. 35

$$G_{vd} \approx V_{in} \times \frac{\left(1 + \frac{s}{\omega_{ESR}}\right)}{\left(1 + \frac{s}{Q\omega_o} + \frac{s^2}{\omega_o^2}\right)},$$

$$\omega_o \approx \frac{1}{\sqrt{LC}} \quad \omega_{ESR} = \frac{1}{rC},$$

$$Q \approx \left[\frac{(r_c + r_e)}{z_c} + \frac{z_c}{R}\right]^{-1}$$

$$G_c = k_c \times \frac{(1 + k_1 s + k_2 s^2)}{s \left(1 + \frac{s}{w_{cp1}}\right) \left(1 + \frac{s}{w_{cp2}}\right)}$$

$\alpha \approx 1$

For type 3 compensator I think you have to go to lecture number lecture 35. So, details are available. So, where the G_{vd} is given and if you are considering you know alpha term, if you are considering alpha to be almost equal to 1, then it simplified to ESR zero. And if we take ω_o to be 1 by square root of LC because the r plus r_c by r plus r_l re both are more or less same. So, we have approximated and that we have discussed and Q factor as well.

(Refer Slide Time: 02:20)

Type-III Compensator Design in VMC Buck Converter: Summary

$$G_c = k_c \times \frac{(1 + k_1 s + k_2 s^2)}{s \left(1 + \frac{s}{w_{cp1}}\right) \left(1 + \frac{s}{w_{cp2}}\right)}$$

$$k_1 = \frac{\sqrt{LC}}{Q}, \quad k_2 = LC$$

$$w_{cp1} = \frac{1}{r_c C}, \quad w_{cp2} \triangleq w_p$$

$$w_p = \frac{w_c}{\tan(90^\circ - PM)}$$

$$k_c = \frac{\alpha w_c}{F_m V_{in}} \times \sqrt{1 + \left(\frac{w_c}{w_p}\right)^2}$$

So, this is a transfer function and we need a type 3 compensator that we have discussed. And then how to change you know for exact you know perfect compensation by analytical pole zero compensation. We know we can cancel one controller pole to cancel ESR zero one controller another controller pole is flexible.

And the controller zeros are placed to cancel the control to output transformation pole ok. And then you know we know that how to obtain this omega p which is nothing but our flexible controller pole and that required two information. We need to provide what is my crossover frequency.

And generally we take up to one tenth of the switching frequency and then we need to specify the phase margin and we have discussed detail model matching and all. And we know the controller gain will turn out to be the ratio of omega c by omega p; where omega c is the crossover frequency and omega p is the pole frequency that we will obtain from this equation.

(Refer Slide Time: 03:53)

CMC Design in a Buck Converter

$$K_{\text{loop}}(s) = \frac{R \left(1 + \frac{s}{\omega_{\text{esr}}} \right)}{\left(1 + \frac{s}{\omega_p} \right)} \times G_c$$

Lec. 38

Type-II

$$G_c = \frac{k_c \left(1 + \frac{s}{\omega_{cz}} \right)}{s \left(1 + \frac{s}{\omega_{cp}} \right)}$$

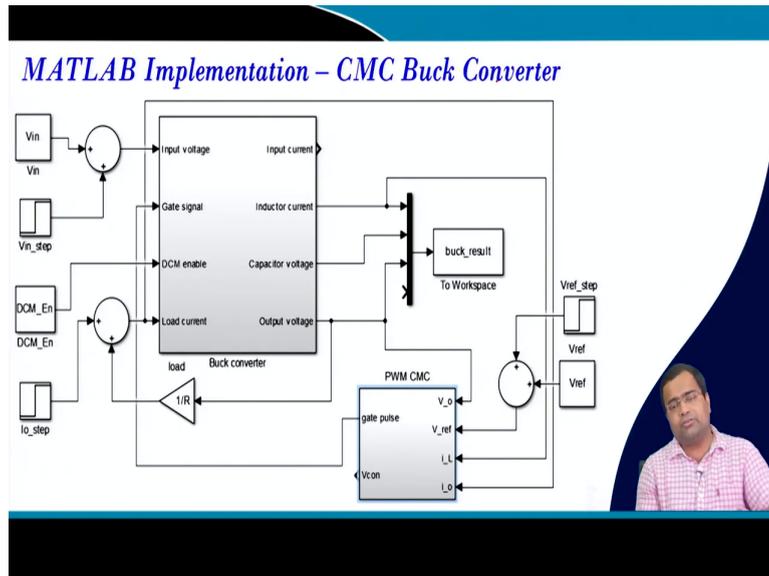
$\omega_{cz} = \omega_p, \omega_{cp} = \omega_{\text{esr}}$

$k_c = \frac{\omega_c}{R}$

Then, for the current mode controller design in a buck converter, if we recapitulate our loop transfer function takes like a first order system, then we need to take the controller, which is a type 2 compensator. It is like a type 2 compensator that we have discussed in I think it is lecture number 38 we discussed in sufficient detail.

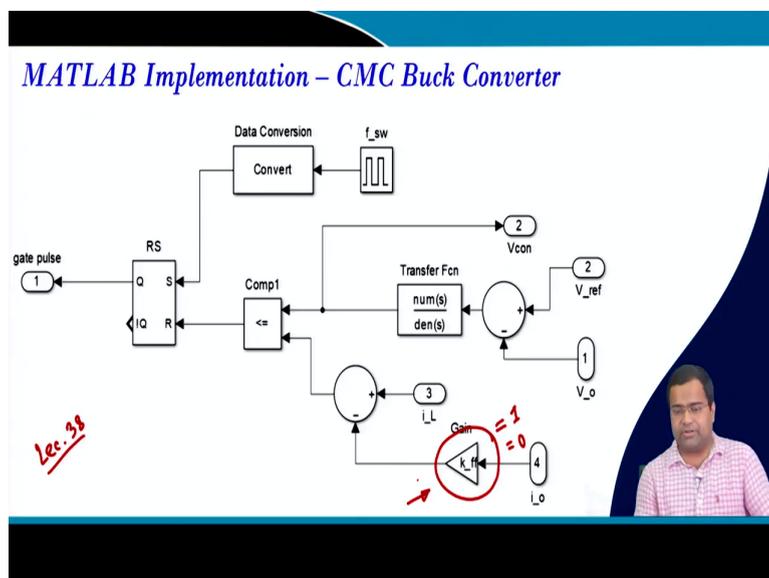
So, you will find lecture 38 ok where ω_{cz} , ω_p , ω_{cp} , ω_{esr} all are discussed how to place the controller zero and controller pole; because we have one controller zero, one controller pole that we have to cancel and another pole is at origin. So, you will get a first order system right and how to get k_c the controller gain that we also discussed.

(Refer Slide Time: 04:43)



And we have also discussed the MATLAB model under current mode control. This has been you know we have discussed multiple time how to implement current mode control buck converter.

(Refer Slide Time: 04:54)



And you know here we are also considering an additional load current feedforward, either you can put it 1, then it will be a load current feedback and if you set to 0 then there is no load current feedforward ok. So, you know in regular current mode control we have discussed

in lecture 38 what we have discussed that if you do not consider load feedforward then the current mode control performance is sluggish.

But, if you consider load current feedforward then you can make the output impedance close to 0, but there will be some finite slew rate due to the characteristic impedance. But, if you talk about reference transient because you are talking about the bandwidth then you will find during the reference transient there will be hardly you know this gain even if you do not consider it does not matter ok. So, in a buck converter we are going to consider load transient performance ok.

(Refer Slide Time: 05:49)

Large-Signal PI Controller Tuning Parameters for a Buck Converter

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

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

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

$$K_i = \frac{\pi(1-D)}{10L} \quad k_n = 1$$

And then we also discussed in lecture number I think lecture number 50 the PI controller tuning parameter of a buck converter where K_p we discussed where we have two separate gains for step up and step down that we have discussed. And then we have said we have also discussed that how to design this integral gain by means of small-signal model that also we have discussed in sufficient detail.

And for load current feedforward we need to consider normally load current gain to be 1 for buck converter right. And we have also discussed and how to implement it is a standard current mode control that is just now we have discussed and we are considering the load current feedforward.

(Refer Slide Time: 06:31)

Large-Signal PID Controller Tuning Parameters

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

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

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

$$K_i = \frac{\pi(1-D)}{10L}$$

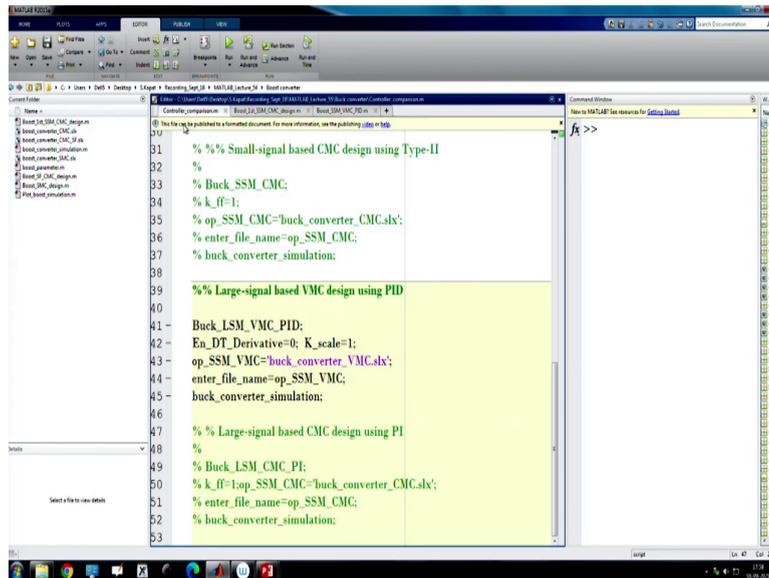
$$K_d \approx C$$

$$\tau_D \approx \frac{T}{20}$$

And if you take the PID controller for our you know large-signal tuning, then we have to discuss what should be my proportional gain and what should be my v_q for step up and step down we have discussed in sufficient detail in lecture number 50. And then how to set the integral gain from the small-signal model.

Then the derivative gain is simply the capacitor value and then we need to put a band limited derivative. Because of this band limited derivative, the time constant should be much smaller than the switching period that we have also discussed. And we know the voltage mode control implementation output voltage reference voltage. So, it is exactly same as the regular voltage mode control where we are only applying the large-signal tuning ok.

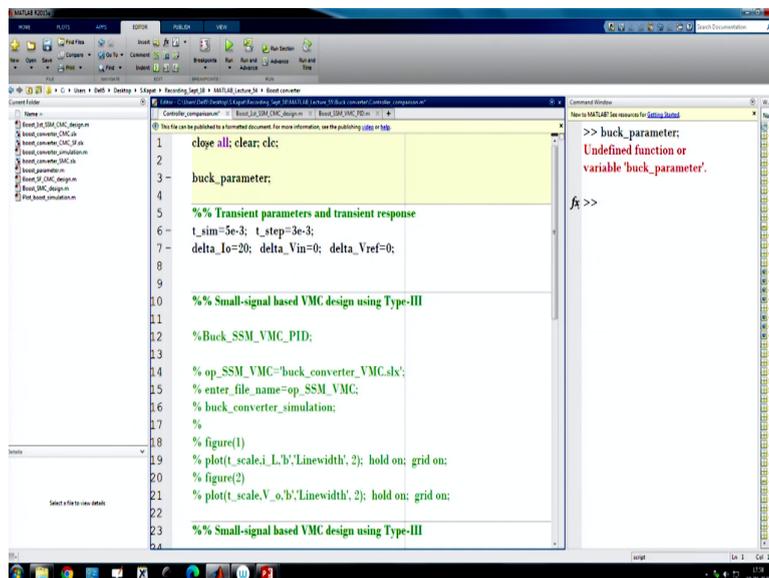
(Refer Slide Time: 07:17)



```
31 %% Small-signal based CMC design using Type-II
32 %
33 % Buck_SSM_CMC;
34 % k_ff=1;
35 % op_ssm_cmc='buck_converter_CMC.slx';
36 % enter_file_name='op_ssm_cmc';
37 % buck_converter_simulation;
38
39 %% Large-signal based VMC design using PID
40
41 Buck_LSM_VMC_PID;
42 En_DT_Derivative=0; K_scale=1;
43 op_ssm_vmc='buck_converter_VMC.slx';
44 enter_file_name='op_ssm_vmc';
45 buck_converter_simulation;
46
47 %% Large-signal based CMC design using PI
48 %
49 % Buck_LSM_CMC_PI;
50 % k_ff=1; op_ssm_cmc='buck_converter_CMC.slx';
51 % enter_file_name='op_ssm_cmc';
52 % buck_converter_simulation;
53
```

So, let us go to the simulation model and check out the performance comparison.

(Refer Slide Time: 07:23)

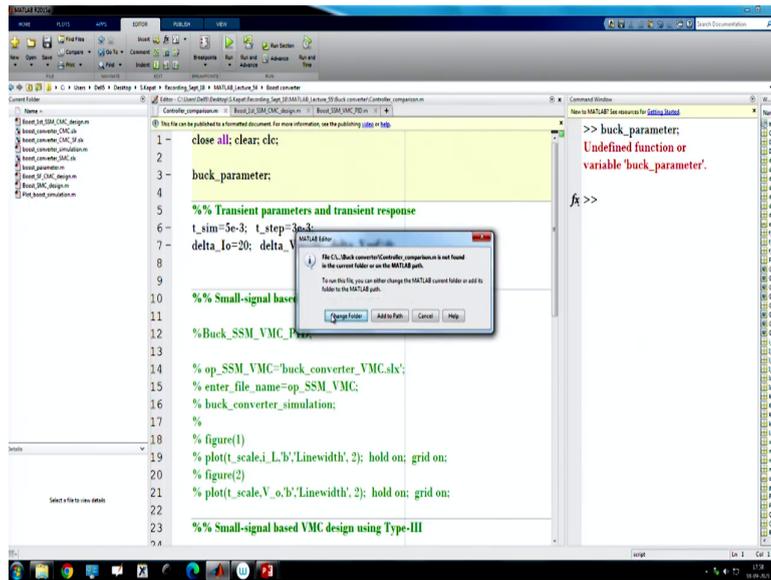


```
1 close all; clear; clc;
2
3 buck_parameter;
4
5 %% Transient parameters and transient response
6 t_sim=5e-3; t_step=3e-3;
7 delta_Io=20; delta_Vin=0; delta_Vref=0;
8
9
10 %% Small-signal based VMC design using Type-III
11
12 %Buck_SSM_VMC_PID;
13
14 % op_ssm_vmc='buck_converter_VMC.slx';
15 % enter_file_name='op_ssm_vmc';
16 % buck_converter_simulation;
17 %
18 % figure(1)
19 % plot(t_scale,i_L,'b','Linewidth',2); hold on; grid on;
20 % figure(2)
21 % plot(t_scale,V_o,'b','Linewidth',2); hold on; grid on;
22
23 %% Small-signal based VMC design using Type-III
```

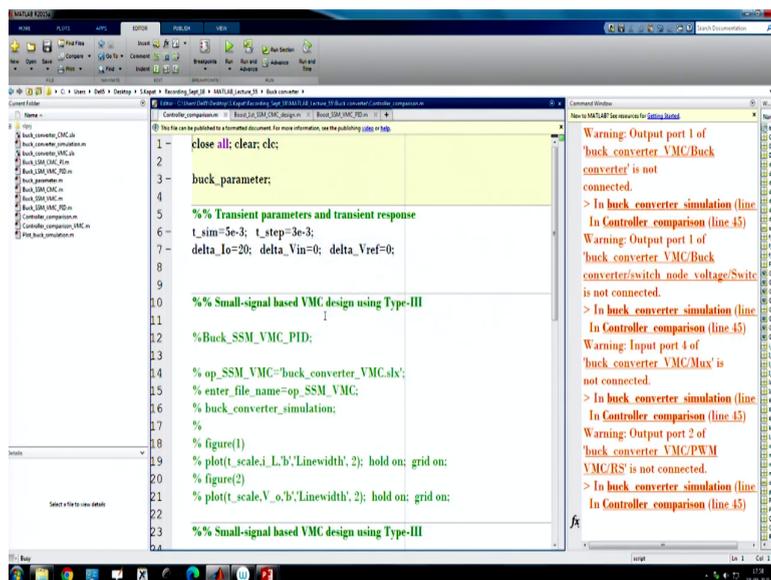
>> buck_parameter;
Undefined function or variable 'buck_parameter'.

So, if we go to the simulation model, we are applying a load step of 20 ampere ok and there is no. So, we already discussed in our how to interact develop interactive simulation case study and if we run this file. Suppose if you want to run it then undefined so, yes.

(Refer Slide Time: 07:46)

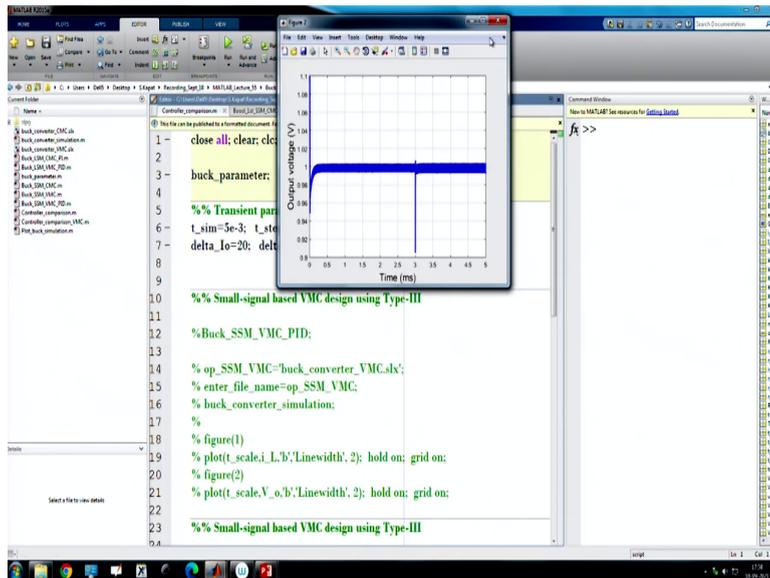


(Refer Slide Time: 07:47)

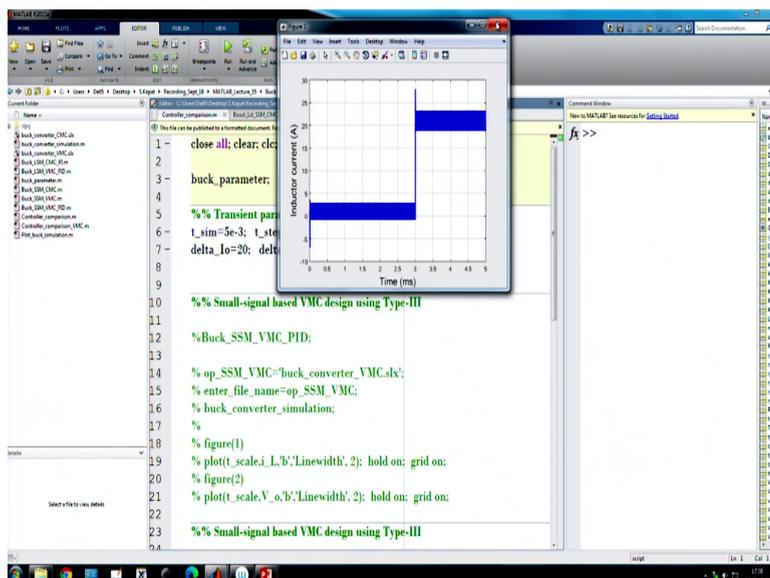


So, it will ask for you need to change the MATLAB ok.

(Refer Slide Time: 07:51)

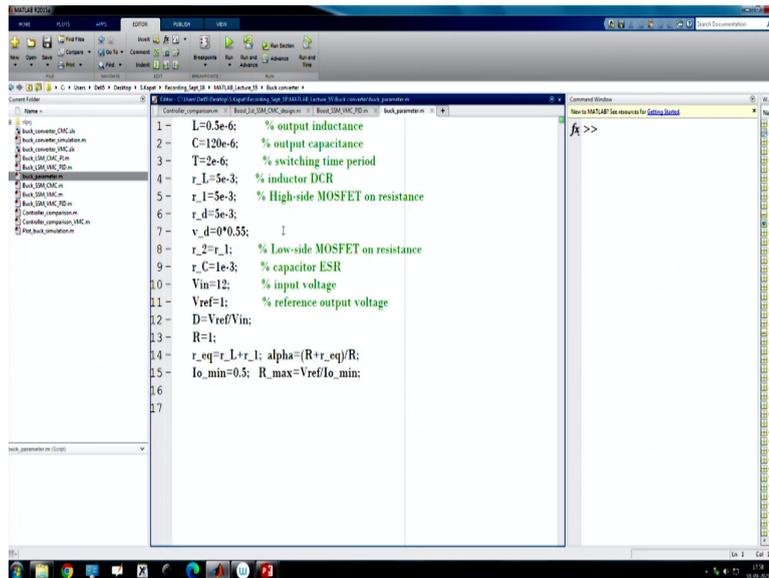


(Refer Slide Time: 07:52)



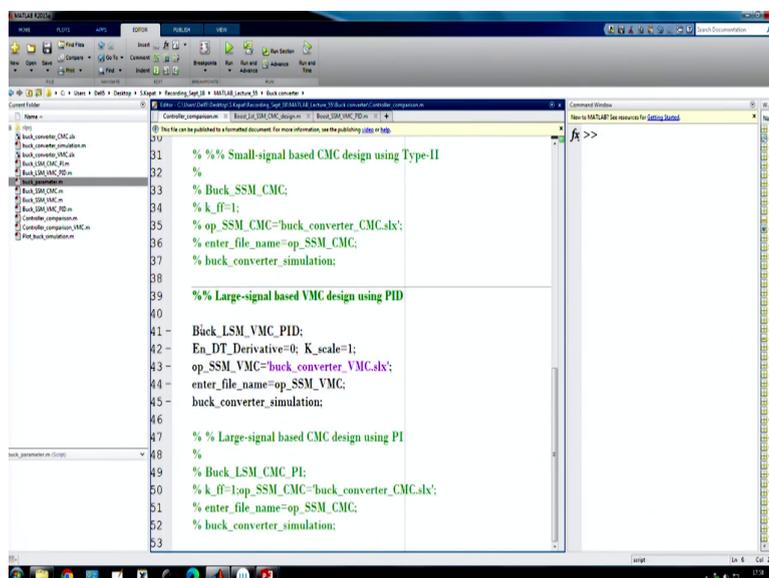
So, let us first wait because we will go one by one.

(Refer Slide Time: 07:58)



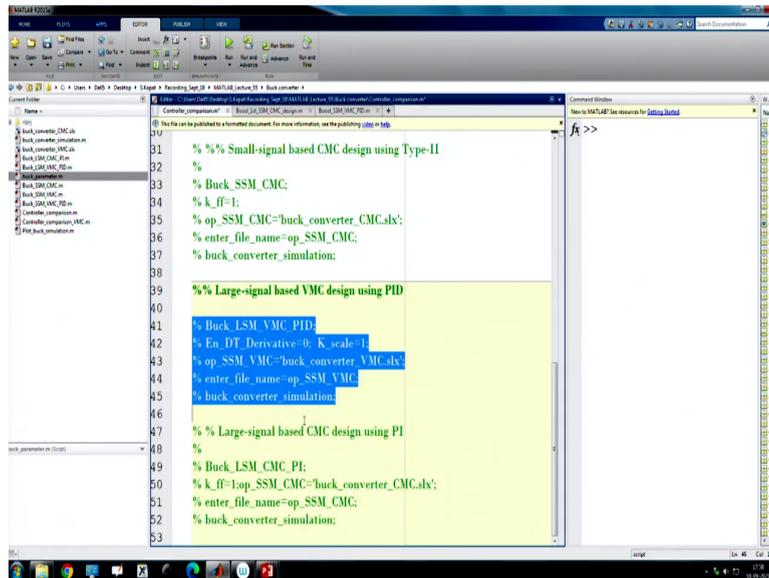
The first thing it will buck converter parameter. It is the same parameter file, which we have been discussing throughout this lecture. There is no change. Next we are considering 5 millisecond total simulation time, 3 millisecond time we are applying a load step transient.

(Refer Slide Time: 08:12)



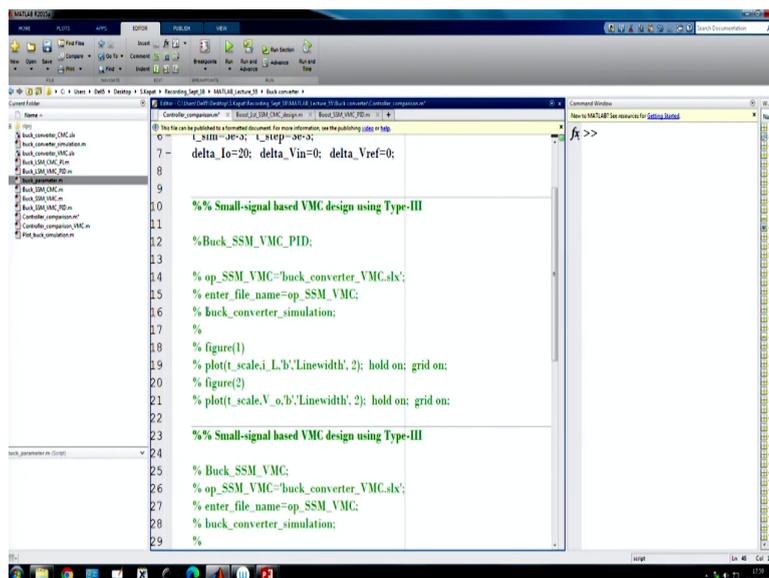
Now, initially what we are considering, ok.

(Refer Slide Time: 08:17)



```
31 %% Small-signal based CMC design using Type-II
32 %
33 % Buck_SSM_CMC;
34 % k_ff=1;
35 % op_SSM_CMC='buck_converter_CMC.slx';
36 % enter_file_name='op_SSM_CMC;
37 % buck_converter_simulation;
38
39 %% Large-signal based VMC design using PID
40
41 % Buck_LSM_VMC_PID;
42 % En_DT_Derivative=0; K_scale=1;
43 % op_SSM_VMC='buck_converter_VMC.slx';
44 % enter_file_name='op_SSM_VMC;
45 % buck_converter_simulation;
46
47 %% Large-signal based CMC design using PI
48
49 % Buck_LSM_CMC_PI;
50 % k_ff=1; op_SSM_CMC='buck_converter_CMC.slx';
51 % enter_file_name='op_SSM_CMC;
52 % buck_converter_simulation;
53
```

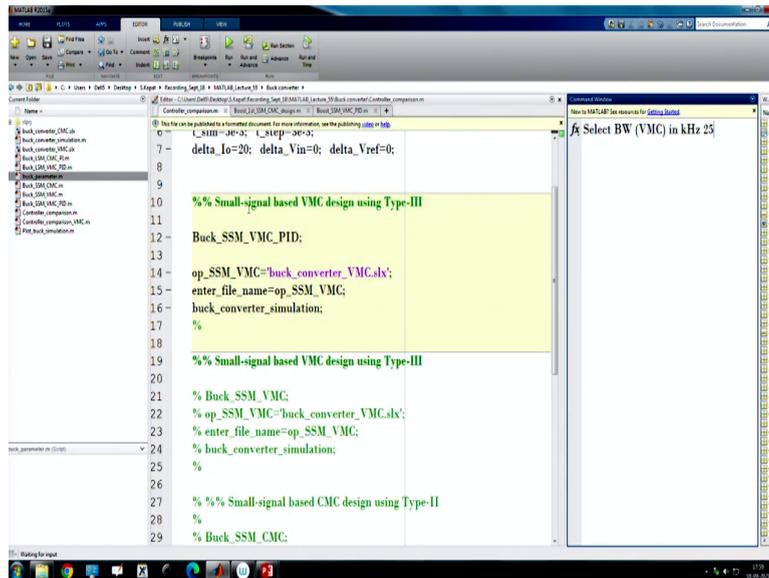
(Refer Slide Time: 08:20)



```
7 - delta_Io=20; delta_Vin=0; delta_Vref=0;
8
9
10 %% Small-signal based VMC design using Type-III
11
12 %Buck_SSM_VMC_PID;
13
14 % op_SSM_VMC='buck_converter_VMC.slx';
15 % enter_file_name='op_SSM_VMC;
16 % buck_converter_simulation;
17 %
18 % figure(1)
19 % plot(t_scale,i_L,'b','Linewidth',2); hold on; grid on;
20 % figure(2)
21 % plot(t_scale,V_o,'b','Linewidth',2); hold on; grid on;
22
23 %% Small-signal based VMC design using Type-III
24
25 % Buck_SSM_VMC;
26 % op_SSM_VMC='buck_converter_VMC.slx';
27 % enter_file_name='op_SSM_VMC;
28 % buck_converter_simulation;
29 %
```

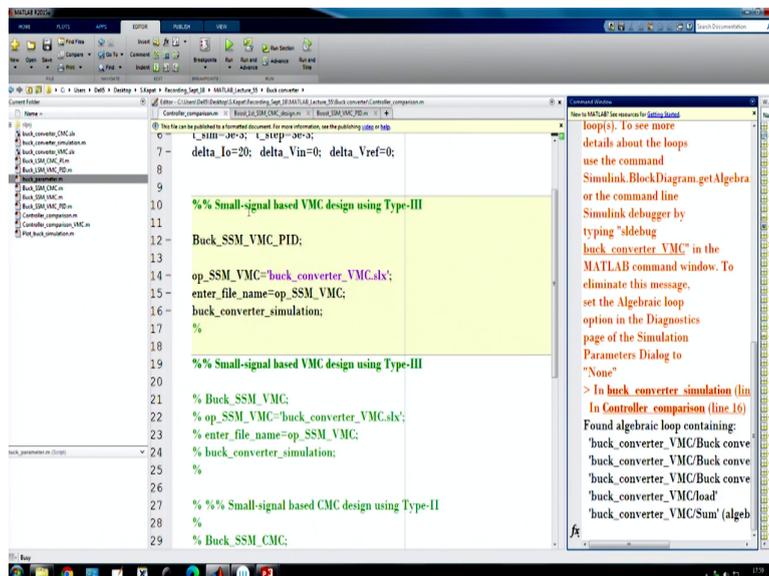
Comment, our traditional we can go for a PID controller also or traditional PID controller.

(Refer Slide Time: 08:24)



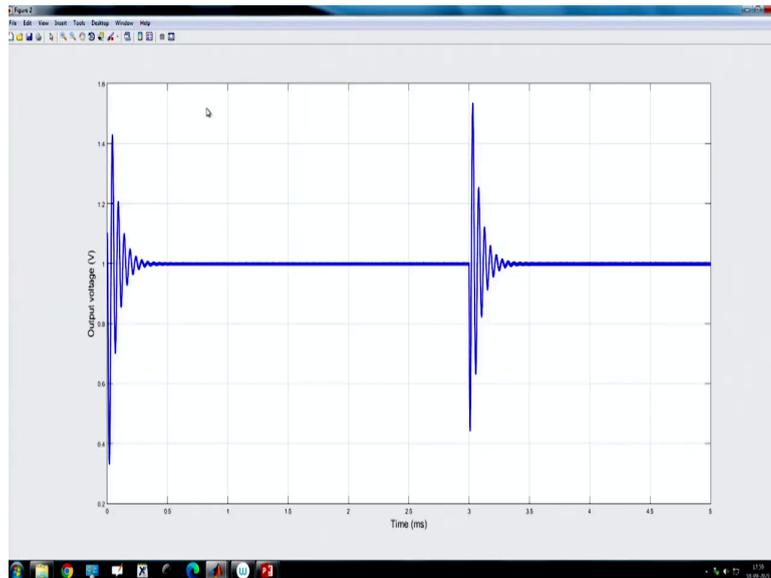
Let us go for a traditional PID controller, where if we run it; it will ask for bandwidth. So, let us say we know that PID controller if you go for one tenth of the switching frequency, then the model may not be valid. So, you can make it like you know, like a 500 kilohertz is the switching frequency.

(Refer Slide Time: 08:52)



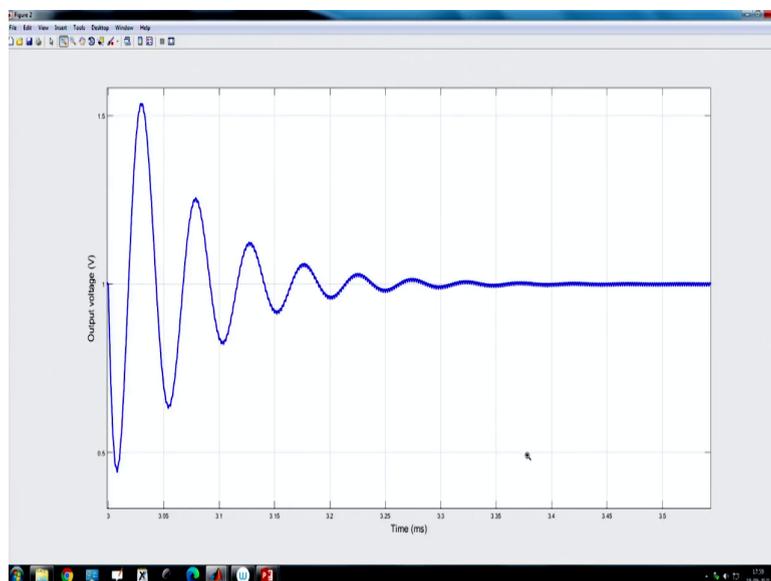
So, you can set let us say 25 kilo, one twentieth of the switching frequency.

(Refer Slide Time: 08:55)



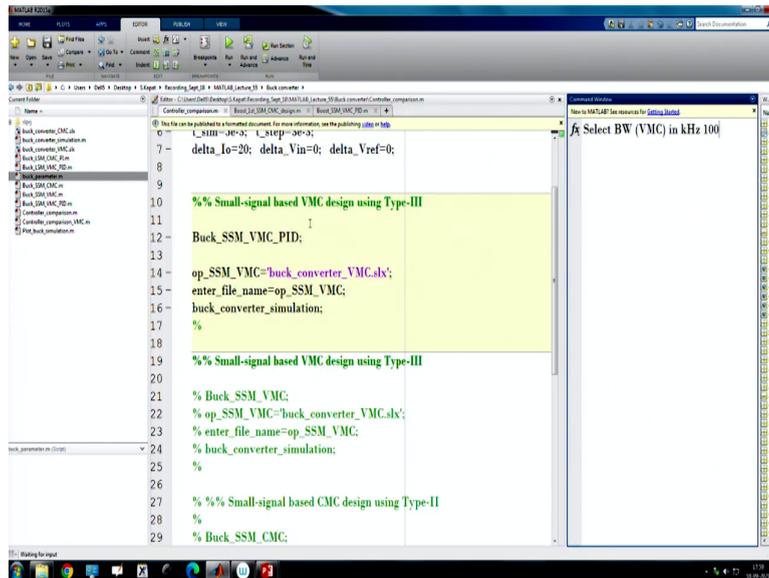
And let us check how does the response looks like. So, response is not at all good because you know it is a large overshoot undershoot and which is not acceptable ok.

(Refer Slide Time: 09:03)



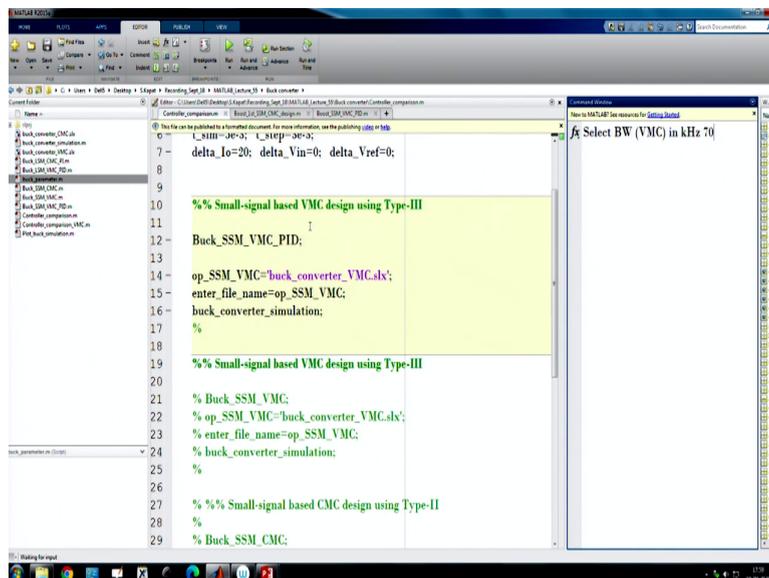
We can improve the response by increasing the bandwidth.

(Refer Slide Time: 09:08)



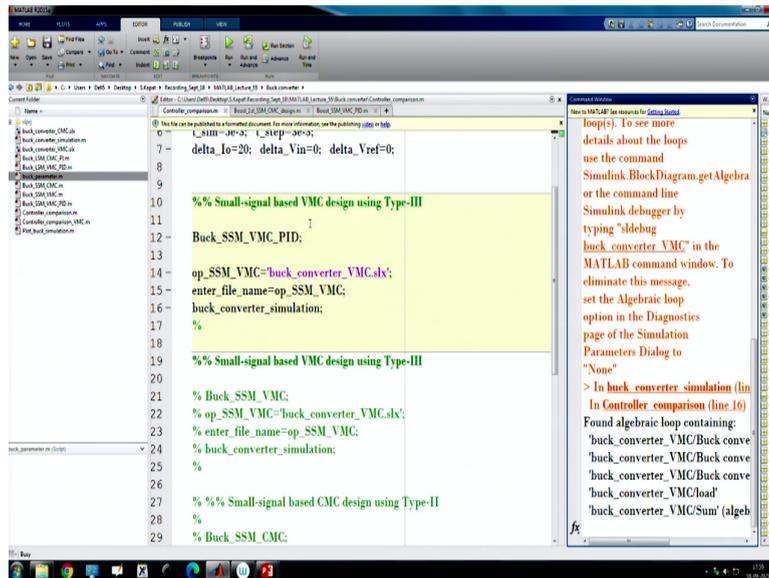
So, you can set. Let us say you know 100 kilohertz.

(Refer Slide Time: 09:13)



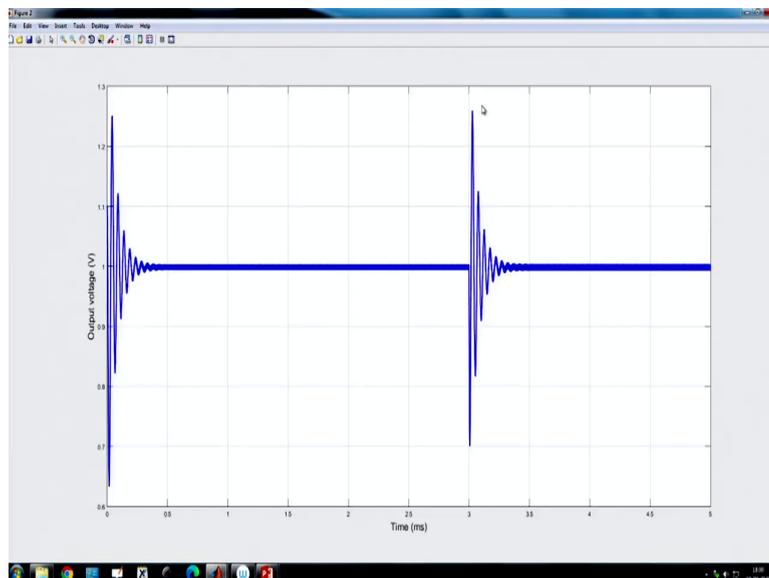
Or 80, 70 kilohertz.

(Refer Slide Time: 09:14)



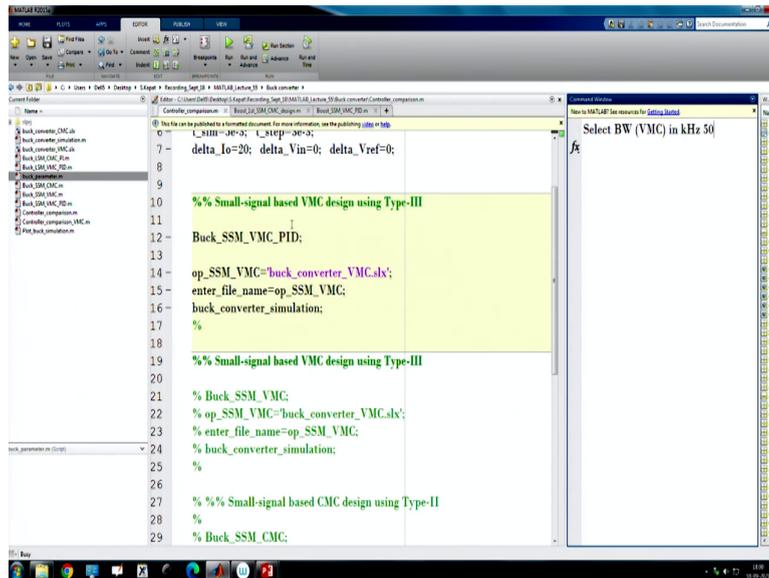
But the model may not be valid.

(Refer Slide Time: 09:17)



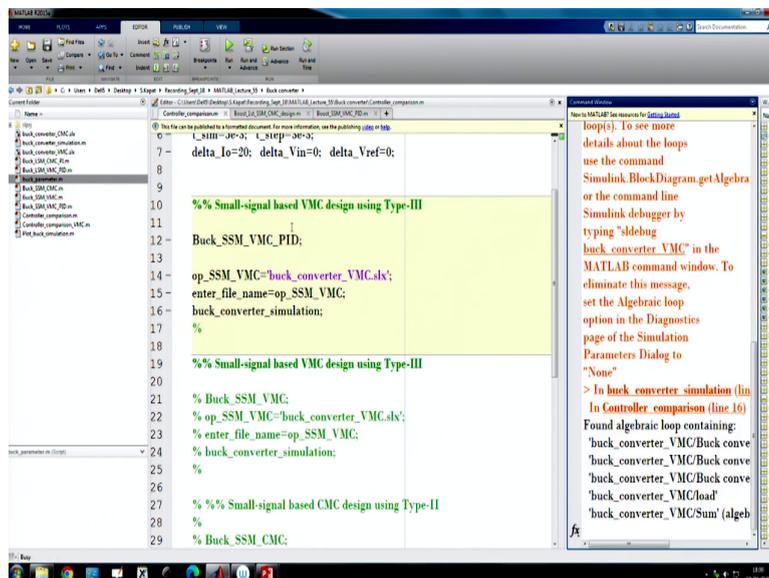
But we can you know. So, we can reduce it, but the question is the model validity remains a question. So, that is why the PID controller we will not talk because the very sluggish response.

(Refer Slide Time: 09:28)



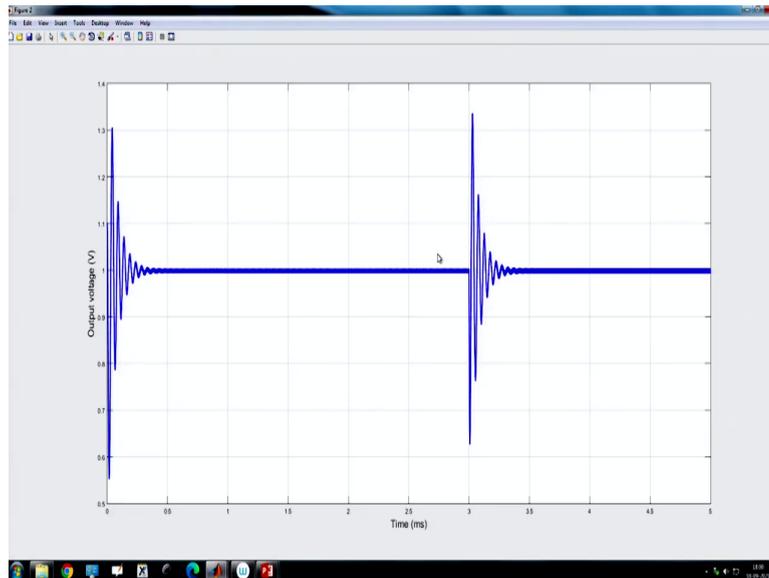
Even if you go up to one tenth of the switching frequency.

(Refer Slide Time: 09:31)



If you go then even the response is not satisfactory.

(Refer Slide Time: 09:34)



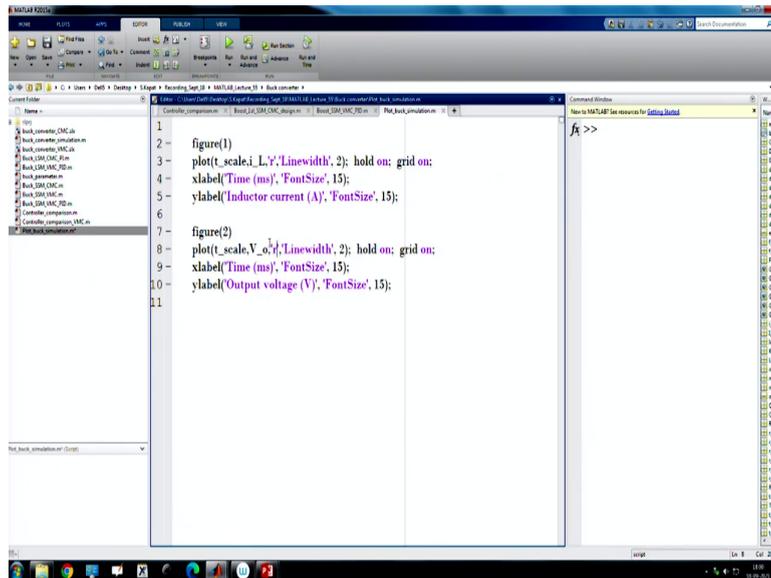
We will hold it the result this is using PID controller in small-signal model.

(Refer Slide Time: 09:44)

```
1  
2- figure(1)  
3- plot(t_scale,i_L,'LineWidth', 2); hold on; grid on;  
4- xlabel('Time (ms)', 'FontSize', 15);  
5- ylabel('Inductor current (A)', 'FontSize', 15);  
6  
7- figure(2)  
8- plot(t_scale,V_o,'LineWidth', 2); hold on; grid on;  
9- xlabel('Time (ms)', 'FontSize', 15);  
10- ylabel('Output voltage (V)', 'FontSize', 15);  
11
```

Now, on top of this let us say.

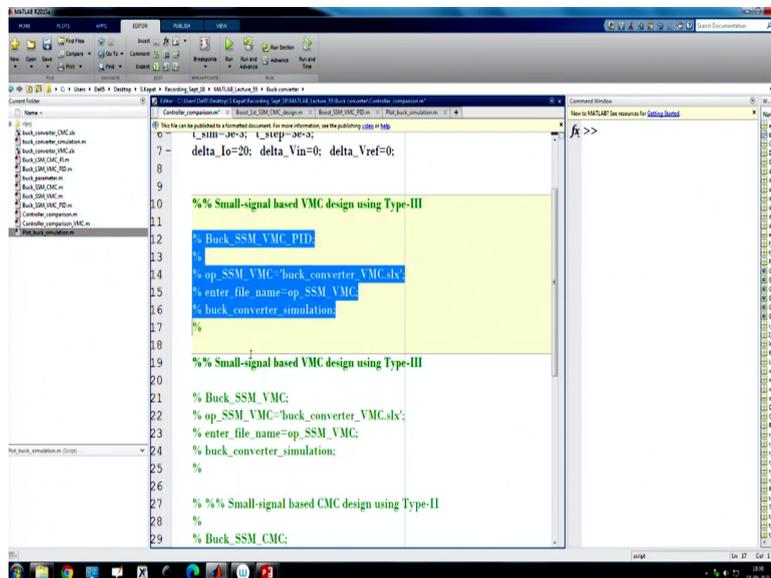
(Refer Slide Time: 09:46)



```
1 figure(1)
2 plot(t_scale,i_L,'Linewidth', 2); hold on; grid on;
3 xlabel('Time (ms)', 'FontSize', 15);
4 ylabel('Inductor current (A)', 'FontSize', 15);
5
6
7 figure(2)
8 plot(t_scale,V_o,'Linewidth', 2); hold on; grid on;
9 xlabel('Time (ms)', 'FontSize', 15);
10 ylabel('Output voltage (V)', 'FontSize', 15);
11
```

If we use red color, we are going for a type 3 compensator ok. So, type 3 compensator where we are considering.

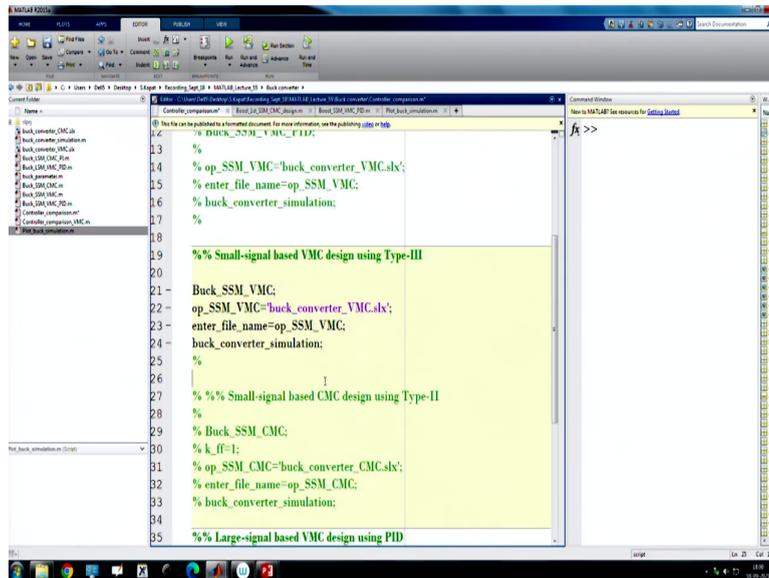
(Refer Slide Time: 10:00)



```
7- delta_Io=20; delta_Vin=0; delta_Vref=0;
8
9
10 %% Small-signal based VMC design using Type-III
11
12 % Buck_SSM_VMC_PTD
13 %
14 % op_SSM_VMC='buck_converter_VMC.slx';
15 % enter_file_name='op_SSM_VMC';
16 % buck_converter_simulation;
17 %
18
19 %% Small-signal based VMC design using Type-III
20
21 % Buck_SSM_VMC;
22 % op_SSM_VMC='buck_converter_VMC.slx';
23 % enter_file_name='op_SSM_VMC';
24 % buck_converter_simulation;
25 %
26
27 %% Small-signal based CMC design using Type-II
28 %
29 % Buck_SSM_CMC;
```

Now, this we have already run comment type 3.

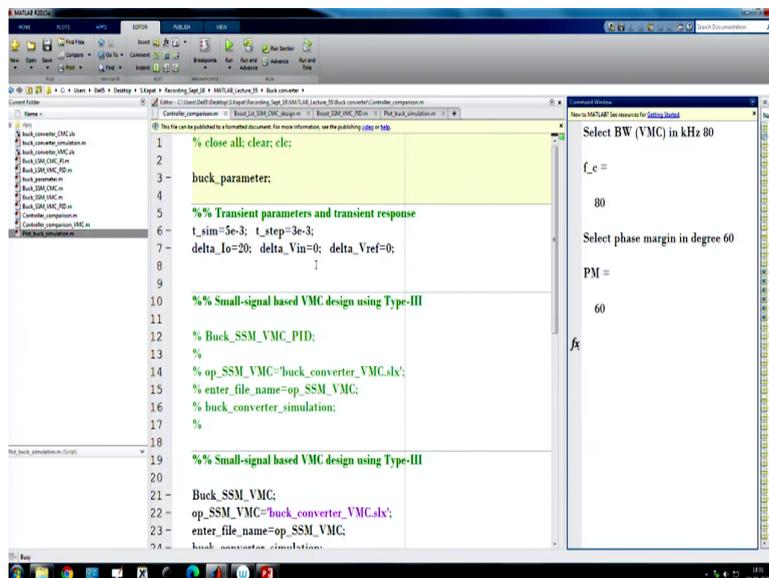
(Refer Slide Time: 10:05)



```
12 % DHDC_2SM_VMC_PID;
13 %
14 % op_SSM_VMC='buck_converter_VMC.slx';
15 % enter_file_name=op_SSM_VMC;
16 % buck_converter_simulation;
17 %
18
19 %% Small-signal based VMC design using Type-III
20
21 Buck_SSM_VMC;
22 op_SSM_VMC='buck_converter_VMC.slx';
23 enter_file_name=op_SSM_VMC;
24 buck_converter_simulation;
25 %
26
27 %% Small-signal based CMC design using Type-II
28 %
29 % Buck_SSM_CMC;
30 % k_ff=1;
31 % op_SSM_CMC='buck_converter_CMC.slx';
32 % enter_file_name=op_SSM_CMC;
33 % buck_converter_simulation;
34
35 %% Large-signal based VMC design using PID
```

This is our type 3 compensator. So, now we are running.

(Refer Slide Time: 10:11)

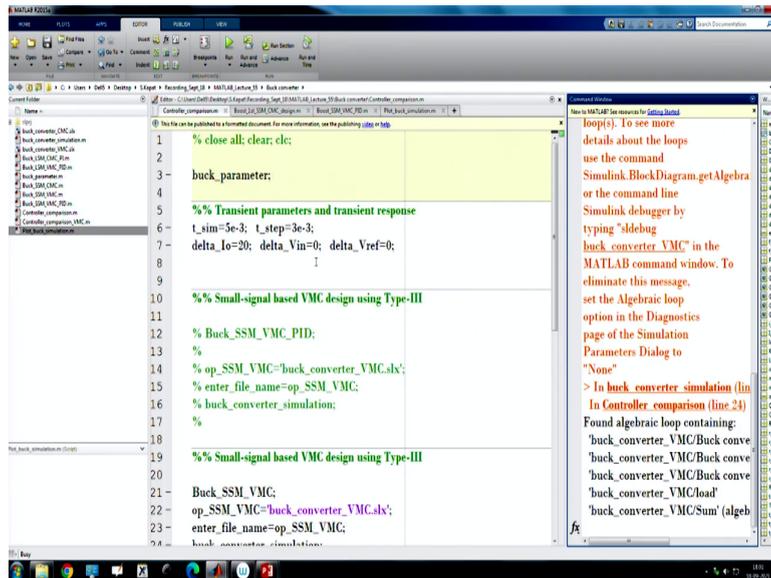


```
1 % close all; clear; clc;
2
3 buck_parameter;
4
5 %% Transient parameters and transient response
6 t_sim=5e-3; t_step=3e-3;
7 delta_Io=20; delta_Vin=0; delta_Vref=0;
8
9
10 %% Small-signal based VMC design using Type-III
11
12 % Buck_SSM_VMC_PID;
13 %
14 % op_SSM_VMC='buck_converter_VMC.slx';
15 % enter_file_name=op_SSM_VMC;
16 % buck_converter_simulation;
17 %
18
19 %% Small-signal based VMC design using Type-III
20
21 Buck_SSM_VMC;
22 op_SSM_VMC='buck_converter_VMC.slx';
23 enter_file_name=op_SSM_VMC;
24 buck_converter_simulation;
```

Select BW (VMC) in kHz 80
f_c = 80
Select phase margin in degree 60
PM = 60
f_x

So, we need to hold this ok. So, run it here. We are going to like it can go up to 70 to 80 kilohertz for load transient response. The model still matches fine and band you know gain phase margin is 60 dB.

(Refer Slide Time: 10:23)



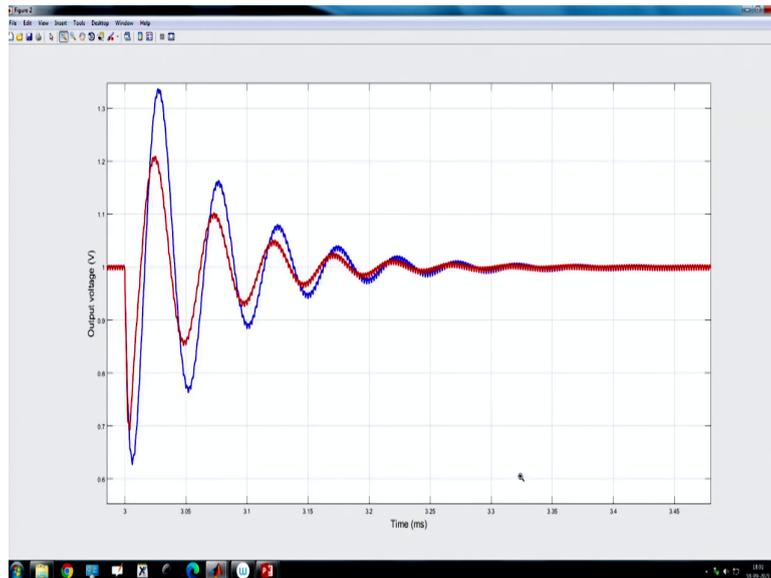
So, let us see how does the response looks like.

(Refer Slide Time: 11:26)



Is this response is much faster because model validity works even for a wider frequency range?

(Refer Slide Time: 10:31)



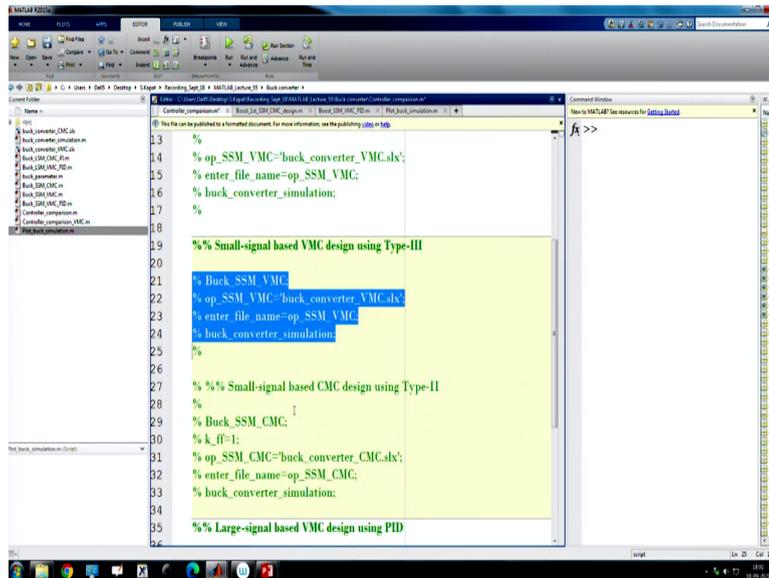
But, still, this may not be sufficient. Now, we are going for large-signal based tuning.

(Refer Slide Time: 10:40)

```
1 figure(1)
2 plot(t_scale,i_L,'LineWidth', 2); hold on; grid on;
3 xlabel('Time (ms)', 'FontSize', 15);
4 ylabel('Inductor current (A)', 'FontSize', 15);
5
6
7 figure(2)
8 plot(t_scale,V_o,'LineWidth', 2); hold on; grid on;
9 xlabel('Time (ms)', 'FontSize', 15);
10 ylabel('Output voltage (V)', 'FontSize', 15);
11
```

And we are using another color. Let us say we are using magenta or maybe we can use green color for the large-signal based tuning.

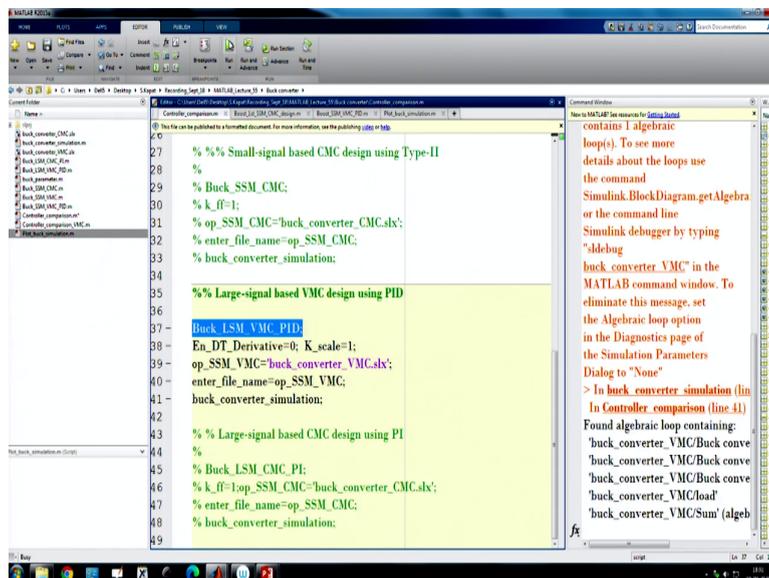
(Refer Slide Time: 10:50)



```
13 %  
14 % op_SSM_VMC='buck_converter_VMC.slx';  
15 % enter_file_name=op_SSM_VMC;  
16 % buck_converter_simulation;  
17 %  
18  
19 %% Small-signal based VMC design using Type-III  
20  
21 % Buck_SSM_VMC  
22 % op_SSM_VMC='buck_converter_VMC.slx';  
23 % enter_file_name=op_SSM_VMC;  
24 % buck_converter_simulation  
25 %  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```

Where we will comment on this particular file, we will comment.

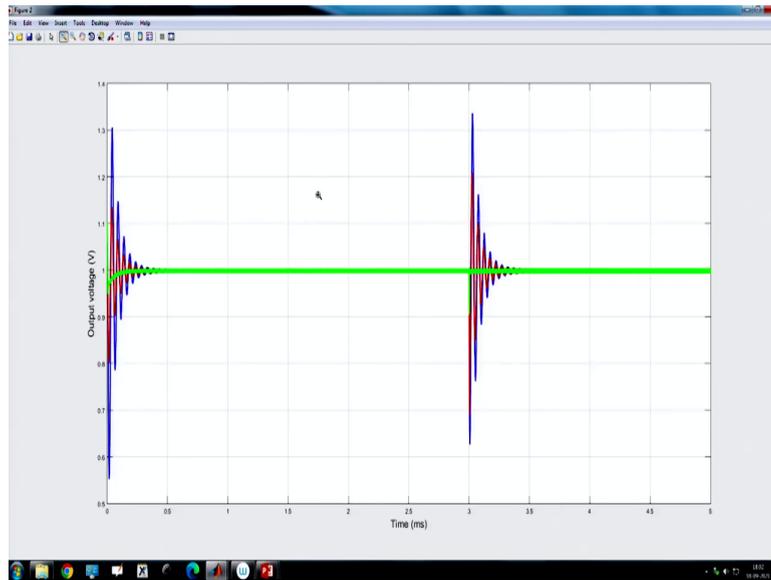
(Refer Slide Time: 10:55)



```
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```

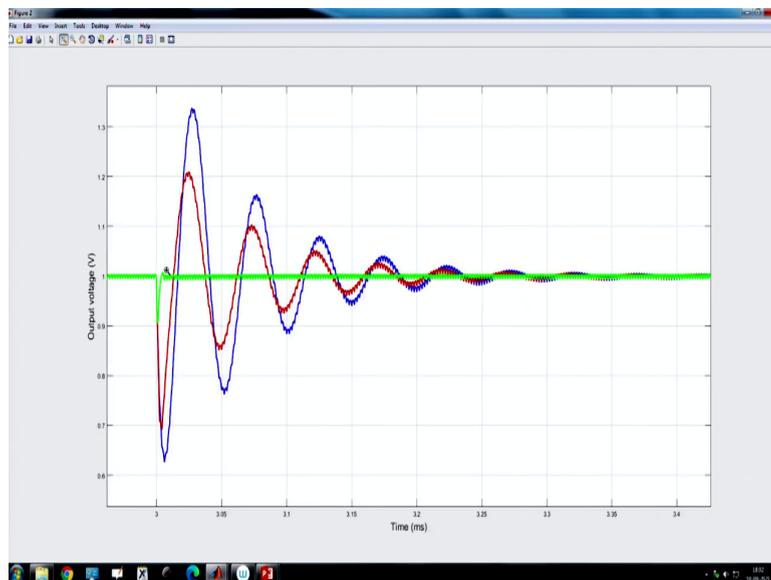
And we are going for large-signal based PID controller tuning uncommet. So, here this function is calling PID controller where all the gains are set accordingly and we have I have shown you all the expression as well as the block diagram. Let us run it and check how does the response looks like.

(Refer Slide Time: 11:21)

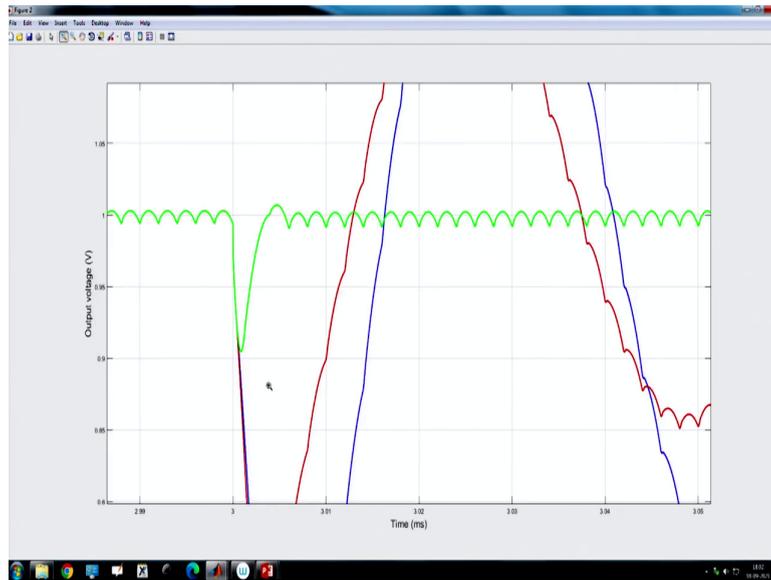


You can see the response is drastically improved.

(Refer Slide Time: 11:24)

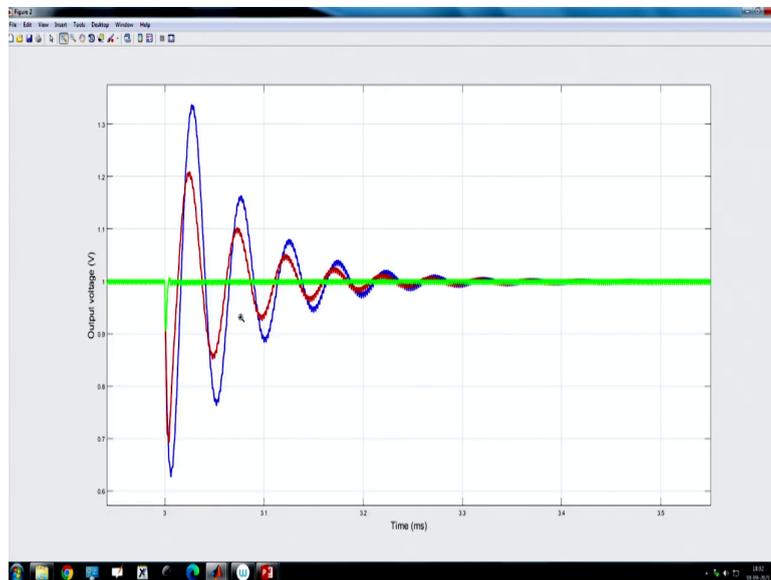


(Refer Slide Time: 11:26)



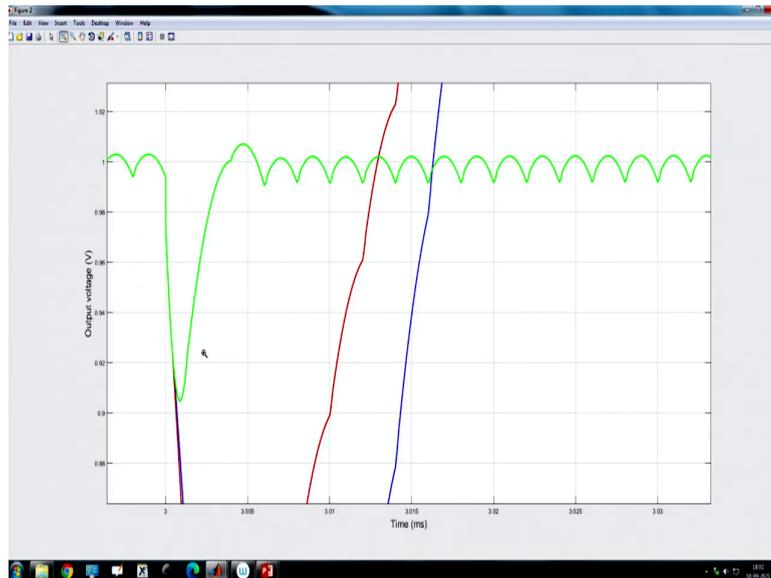
Because the green color it is just around 100 milli volt is the transient response your undershoot less than 100 millivolt.

(Refer Slide Time: 11:35)



And the recovery time is just a few cycle.

(Refer Slide Time: 11:37)



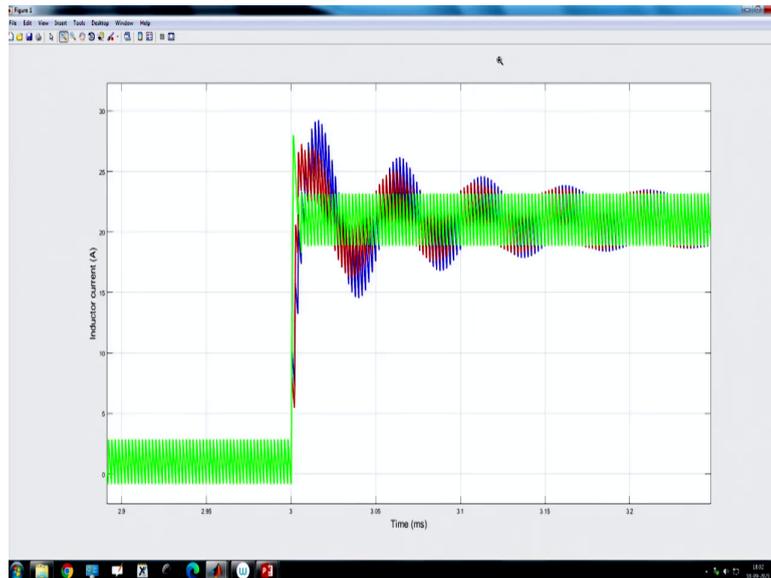
I mean, it is like you know because it is just a two switching cycle whereas, this takes much longer duration.

(Refer Slide Time: 11:46)



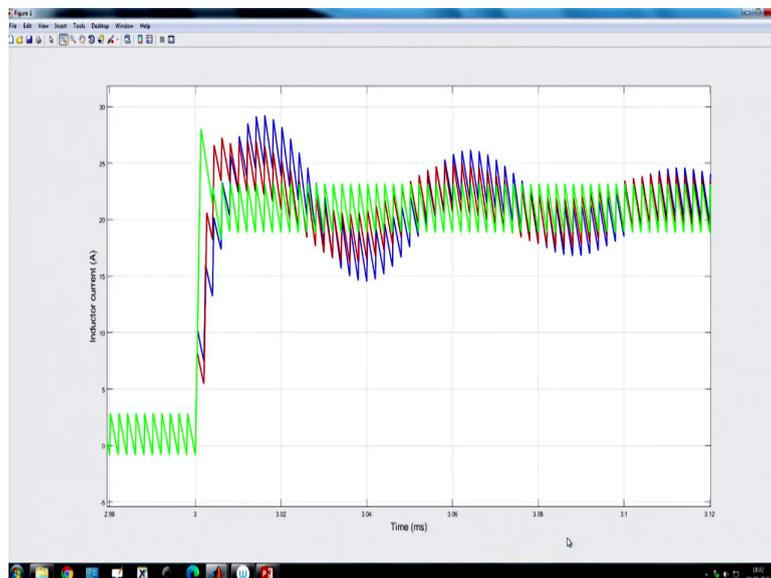
And if you go to inductor current you know.

(Refer Slide Time: 11:51)



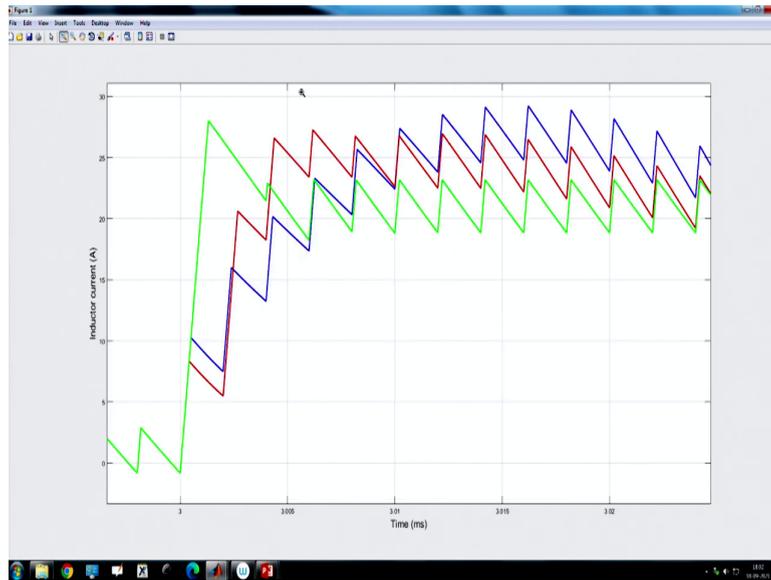
You see, the inductor current is not very significantly high.

(Refer Slide Time: 11:54)



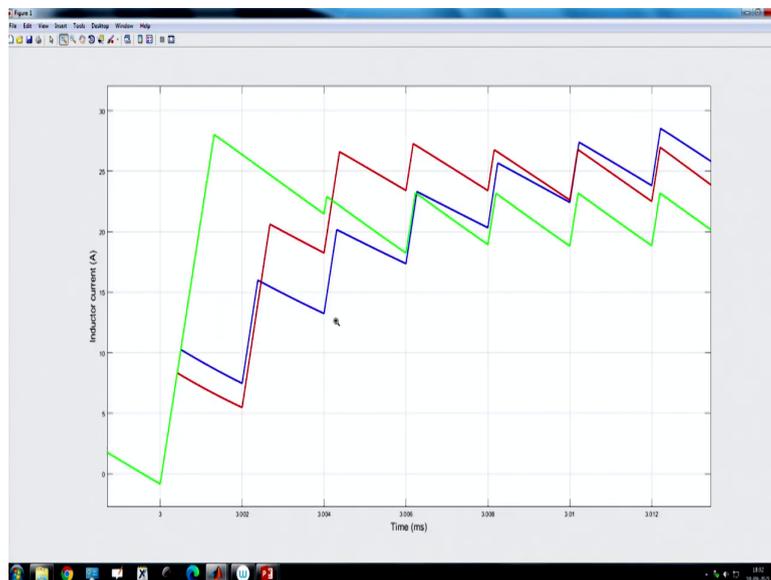
But, our traditional approach of designing because of the model validity and you know small load duty ratio.

(Refer Slide Time: 12:02)



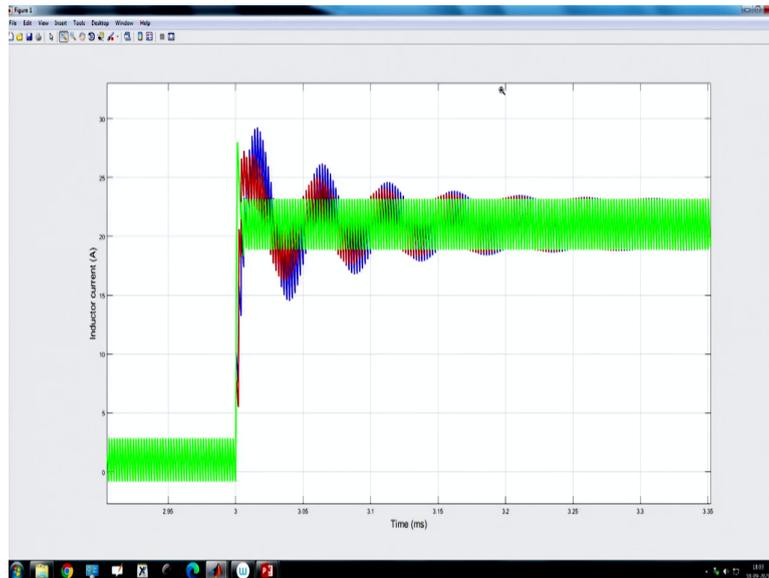
So, you will see the difference will come at the fast recovery time.

(Refer Slide Time: 12:10)



Because since the small-signal requires low duty ratio perturbation. So, during the slew up process, you know there is a turn on of happening and because of that, it will slow down the transient response. Whereas, the large-signal tuning it allow duty ratio saturation. you see it is turning on for a quite some time and then turning off and it is coming to next operating point in almost one switching cycle. So, it achieved almost time optimal recovery.

(Refer Slide Time: 12:32)



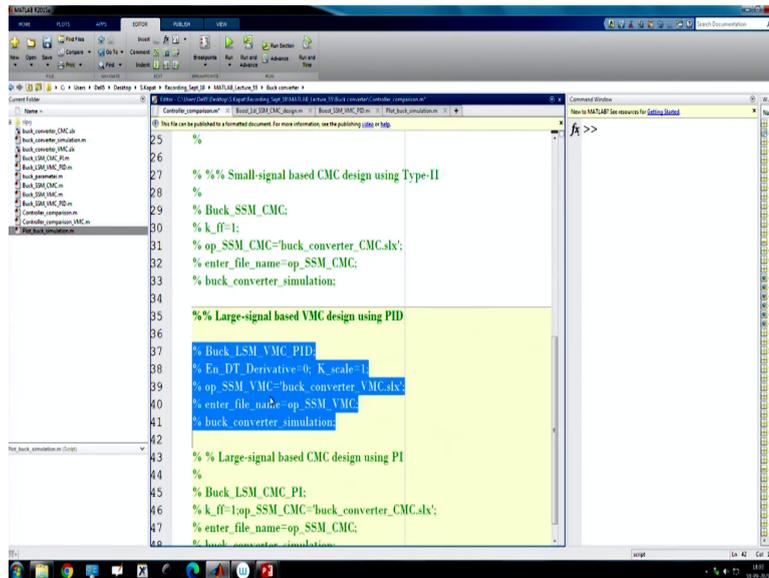
So, now, this large-signal tuning can drastically improve the transient performance compared to the four small-signal tuning. Now, we are going for current mode control. So, let us you know close all the files and we will again clear once more.

(Refer Slide Time: 12:49)

```
close all; clear; clc;
buck_parameter;
%% Transient parameters and transient response
L_sim=5e-3; L_step=3e-3;
delta_Io=20; delta_Vin=0; delta_Vref=0;
%% Small-signal based VMC design using Type-III
% Buck_SSM_VMC_PID;
%
% op_SSM_VMC='buck_converter_VMCsk';
% enter_file_name='op_SSM_VMC;
% buck_converter_simulation;
%
%% Small-signal based VMC design using Type-III
% Buck_SSM_VMC;
% op_SSM_VMC='buck_converter_VMCsk';
% enter_file_name='op_SSM_VMC;
% buck_converter_simulation;
```

Now, we are going for current mode control.

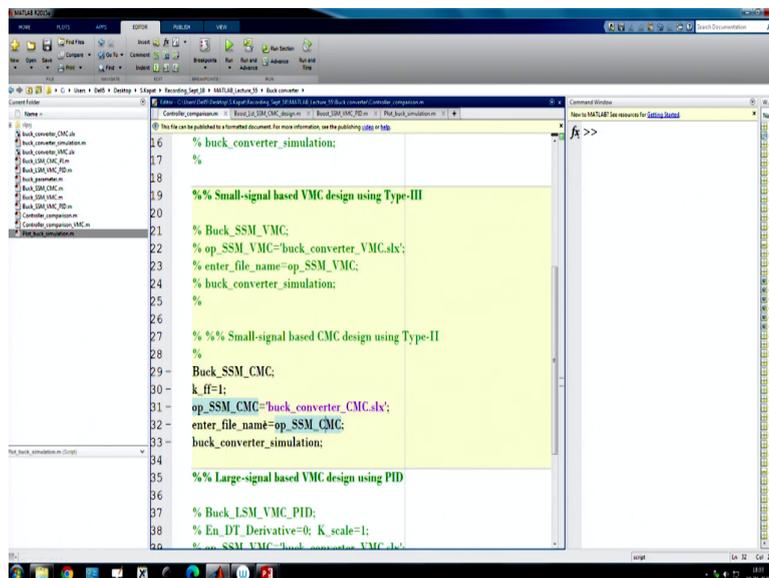
(Refer Slide Time: 12:56)



```
25 %  
26 %  
27 % % Small-signal based CMC design using Type-II  
28 %  
29 % Buck_SSM_CMC;  
30 % k_ff=1;  
31 % op_SSM_CMC='buck_converter_CMC.slx';  
32 % enter_file_name='op_SSM_CMC';  
33 % buck_converter_simulation;  
34 %  
35 % % Large-signal based VMC design using PID  
36 %  
37 % Buck_LSM_VMC_PID;  
38 % En_DT_Derivative=0; K_scale=1;  
39 % op_SSM_VMC='buck_converter_VMC.slx';  
40 % enter_file_name='op_SSM_VMC';  
41 % buck_converter_simulation;  
42 %  
43 % % Large-signal based CMC design using PI  
44 %  
45 % Buck_LSM_CMC_PI;  
46 % k_ff=1; op_SSM_CMC='buck_converter_CMC.slx';  
47 % enter_file_name='op_SSM_CMC';  
48 % buck_converter_simulation;
```

So, in current mode control you know we will see two things.

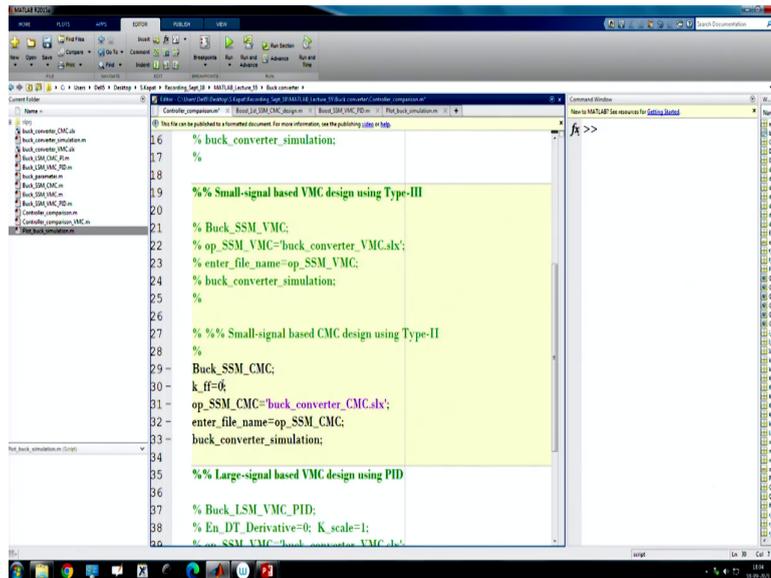
(Refer Slide Time: 13:03)



```
16 % buck_converter_simulation;  
17 %  
18 %  
19 % % Small-signal based VMC design using Type-III  
20 %  
21 % Buck_SSM_VMC;  
22 % op_SSM_VMC='buck_converter_VMC.slx';  
23 % enter_file_name='op_SSM_VMC';  
24 % buck_converter_simulation;  
25 %  
26 %  
27 % % Small-signal based CMC design using Type-II  
28 %  
29 % Buck_SSM_CMC;  
30 % k_ff=1;  
31 % op_SSM_CMC='buck_converter_CMC.slx';  
32 % enter_file_name='op_SSM_CMC';  
33 % buck_converter_simulation;  
34 %  
35 % % Large-signal based VMC design using PID  
36 %  
37 % Buck_LSM_VMC_PID;  
38 % En_DT_Derivative=0; K_scale=1;  
39 % op_SSM_VMC='buck_converter_VMC.slx';
```

The first thing first we will go for small-signal based current mode control. So, here k_{ff} I have shown you if you go to the k_{ff} the k_{ff} if you go to current mode control. So, this is a load current feedforward. So, if you set k_{ff} to be 0 then you are not considering load feedforward ok. So, we will consider cases first k_{ff} equal to 0.

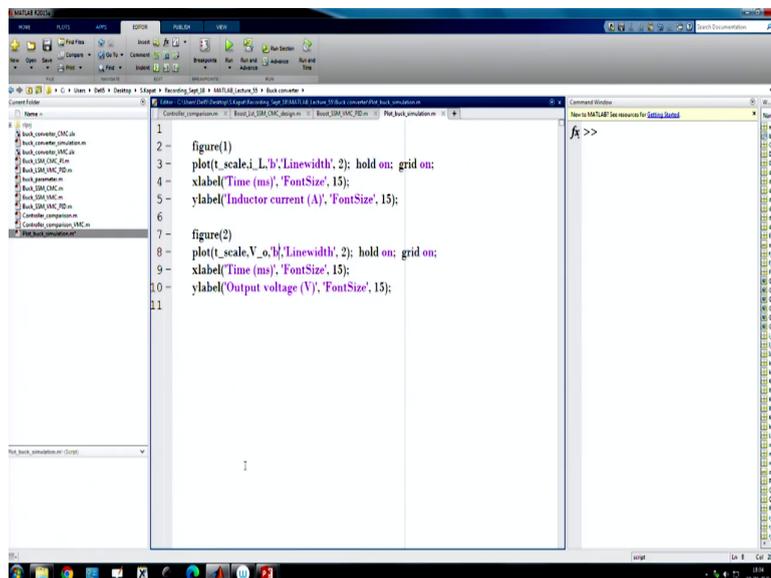
(Refer Slide Time: 13:24)



```
16 % buck_converter_simulation;
17 %
18
19 %%% Small-signal based VMC design using Type-III
20
21 % Buck_SSM_VMC;
22 % op_SSM_VMC='buck_converter_VMC.slk';
23 % enter_file_name='op_SSM_VMC;
24 % buck_converter_simulation;
25 %
26
27 %%% Small-signal based CMC design using Type-II
28 %
29 - Buck_SSM_CMC;
30 - k_ff=0;
31 - op_SSM_CMC='buck_converter_CMC.slk';
32 - enter_file_name='op_SSM_CMC;
33 - buck_converter_simulation;
34
35 %%% Large-signal based VMC design using PID
36
37 % Buck_LSM_VMC_PID;
38 % En_DT_Derivative=0; K_scale=1;
39 % op_SSM_VMC='buck_converter_VMC.slk';
```

So, in this case, we are talking about let us say we are using blue color.

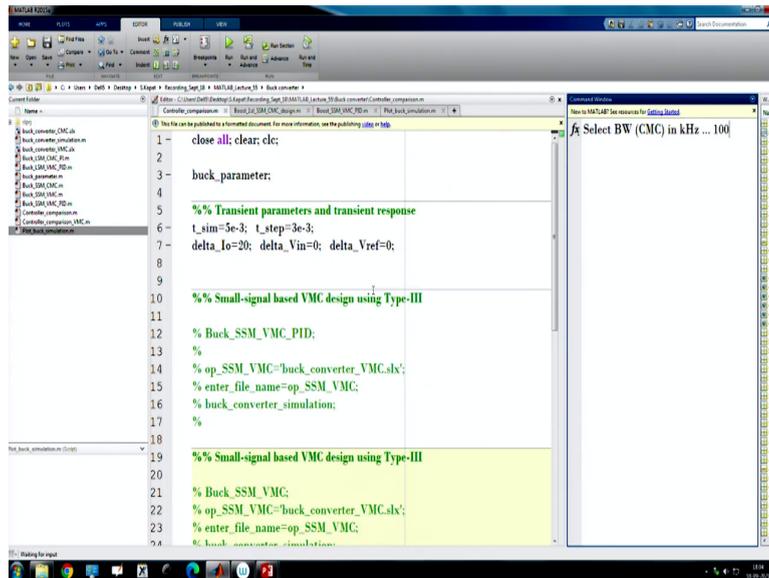
(Refer Slide Time: 13:29)



```
1
2 - figure(1)
3 - plot(t_scale,i_L,'LineWidth', 2); hold on; grid on;
4 - xlabel('Time (ms)', 'FontSize', 15);
5 - ylabel('Inductor current (A)', 'FontSize', 15);
6
7 - figure(2)
8 - plot(t_scale,V_o,'LineWidth', 2); hold on; grid on;
9 - xlabel('Time (ms)', 'FontSize', 15);
10 - ylabel('Output voltage (V)', 'FontSize', 15);
11
```

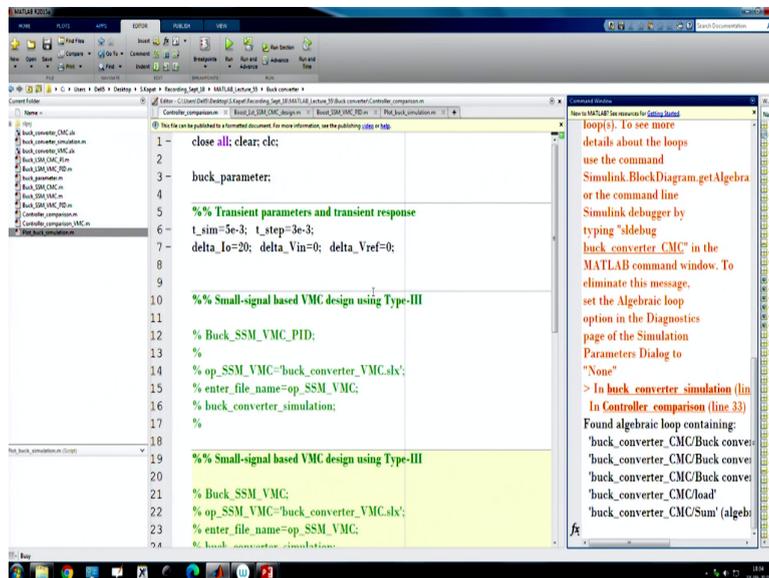
So, the regular current mode control design ok. So, let us run it using the regular current mode control design.

(Refer Slide Time: 13:41)



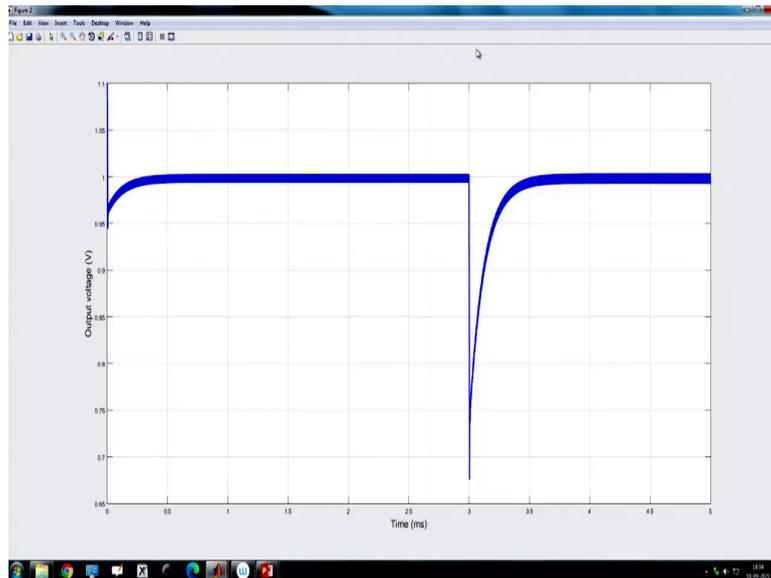
So, if you run it; it will ask for bandwidth. So, let us say we can go up to let us say 100 kilohertz.

(Refer Slide Time: 13:46)

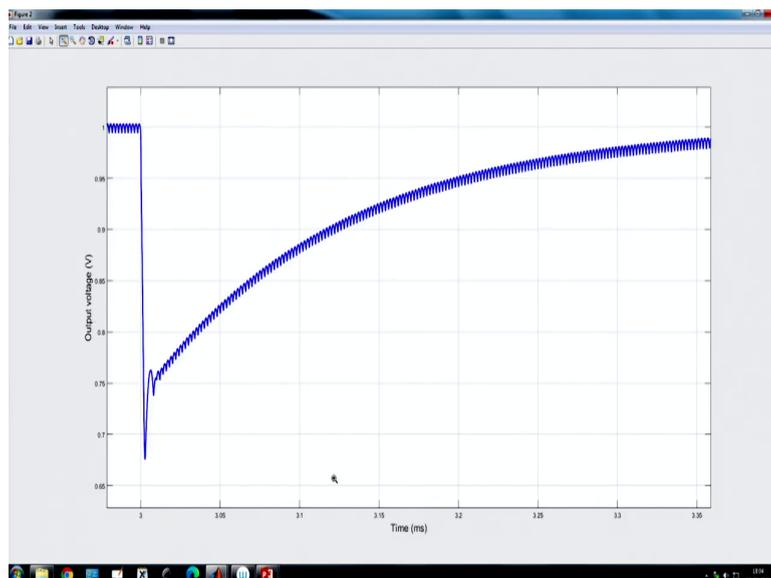


Close trying to go close to one fifth for load transient somewhat we can go, but I think this still model matching will be a concern, but we are trying to go close to one fifth. Let us see what happen. So, this is a load transient response using current mode control.

(Refer Slide Time: 14:04)

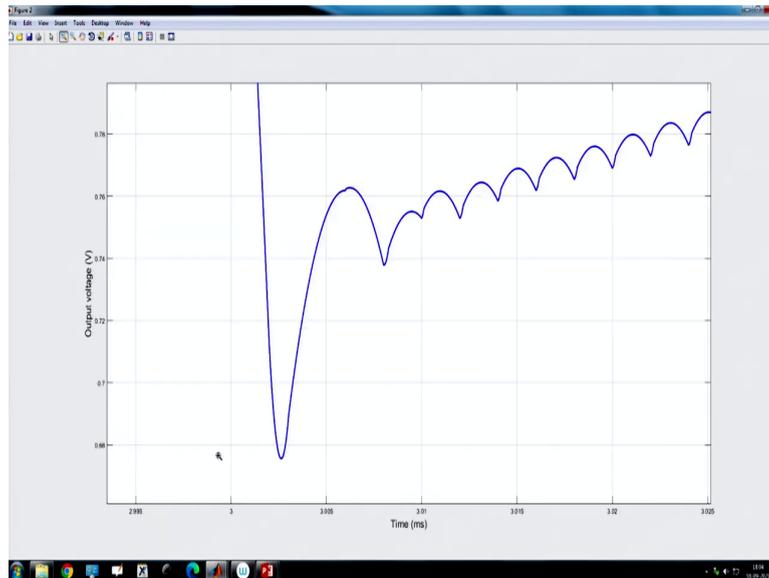


(Refer Slide Time: 14:10)



And this is a response and you see this we have discussed the model matching will not happen due to the jump.

(Refer Slide Time: 14:12)



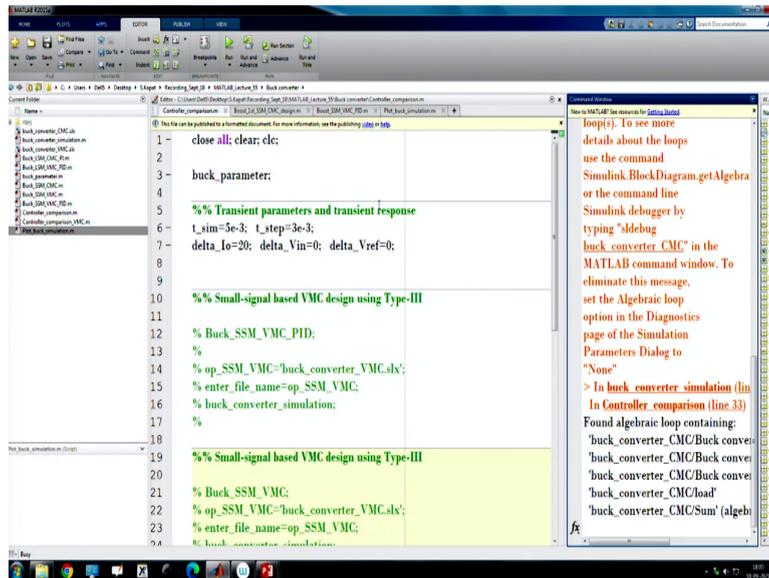
So, it should not go to one fifth.

(Refer Slide Time: 14:19)

```
1 close all; clear; clc;
2
3 buck_parameter;
4
5 %% Transient parameters and transient response
6 t_sim=5e-3; t_step=3e-3;
7 delta_Io=20; delta_Vin=0; delta_Vref=0;
8
9
10 %% Small-signal based VMC design using Type-III
11
12 % Buck_SSM_VMC_PID;
13 %
14 % op_SSM_VMC='buck_converter_VMC.slx';
15 % enter_file_name=op_SSM_VMC;
16 % buck_converter_simulation;
17 %
18
19 %% Small-signal based VMC design using Type-III
20
21 % Buck_SSM_VMC;
22 % op_SSM_VMC='buck_converter_VMC.slx';
23 % enter_file_name=op_SSM_VMC;
24 % buck_converter_simulation;
```

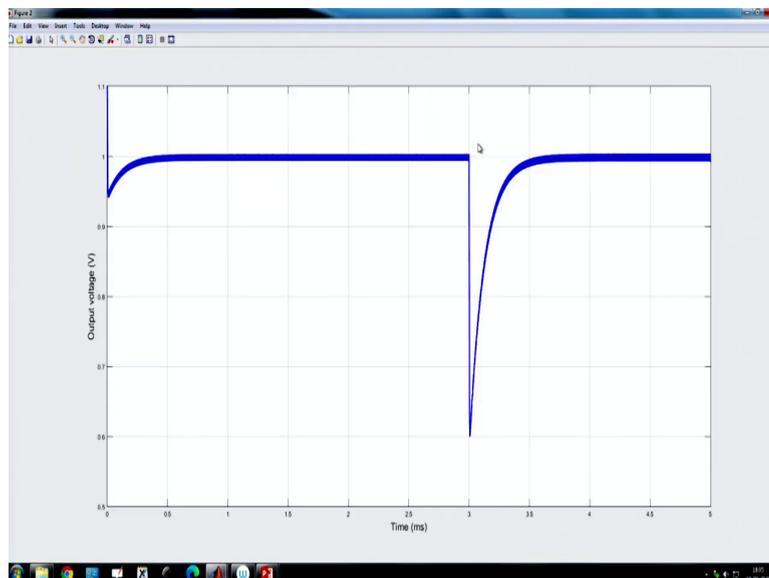
So, let us limit our whole thing to one eighth.

(Refer Slide Time: 14:27)



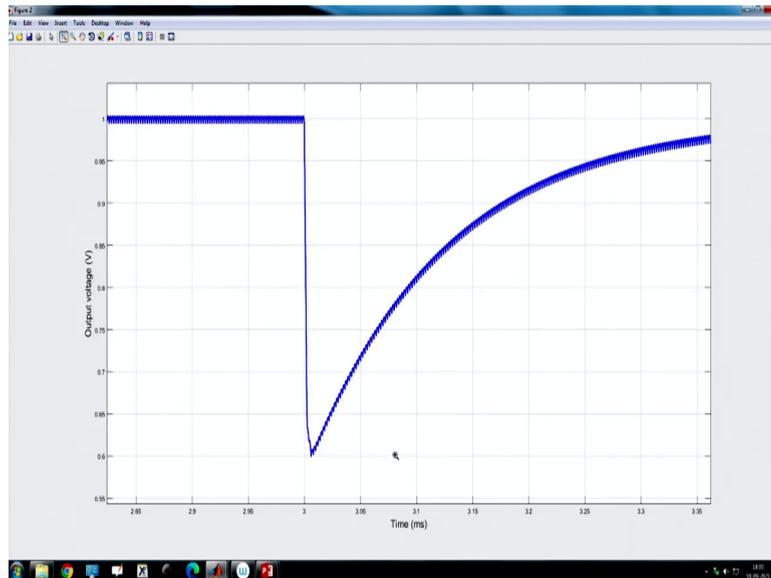
That means we will get 500 by, let us say one eighth.

(Refer Slide Time: 14:30)



And if we say the response; so, now, there is no as such very large overshoot fine.

(Refer Slide Time: 14:36)



So, it is a sluggish response ok, but unlike in voltage mode, there is an overshoot undershoot. It has only undershot, but it is slow. Now, if you consider on top of that load current feedforward.

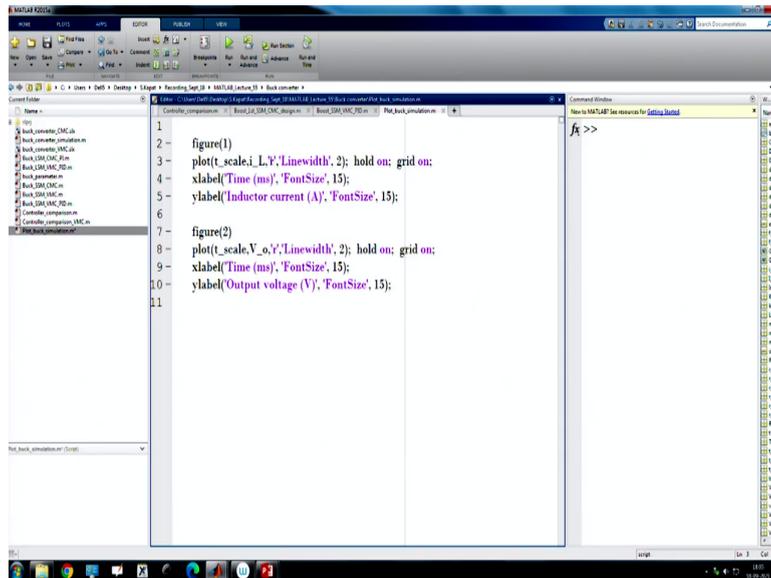
(Refer Slide Time: 14:50)

The screenshot shows the MATLAB Editor interface. The main window displays a script with the following content:

```
13 %  
14 % op_SSM_VMC='buck_converter_VMC.slx';  
15 % enter_file_name=op_SSM_VMC;  
16 % buck_converter_simulation;  
17 %  
18  
19 %% Small-signal based VMC design using Type-III  
20  
21 % Buck_SSM_VMC;  
22 % op_SSM_VMC='buck_converter_VMC.slx';  
23 % enter_file_name=op_SSM_VMC;  
24 % buck_converter_simulation;  
25 %  
26  
27 %% Small-signal based CMC design using Type-II  
28 %  
29 Buck_SSM_CMC;  
30 k_H=1;  
31 op_SSM_CMC='buck_converter_CMC.slx';  
32 enter_file_name=op_SSM_CMC;  
33 buck_converter_simulation;  
34  
35 %% Large-signal based VMC design using PID
```

That means we are considering 1 and we are using a different color.

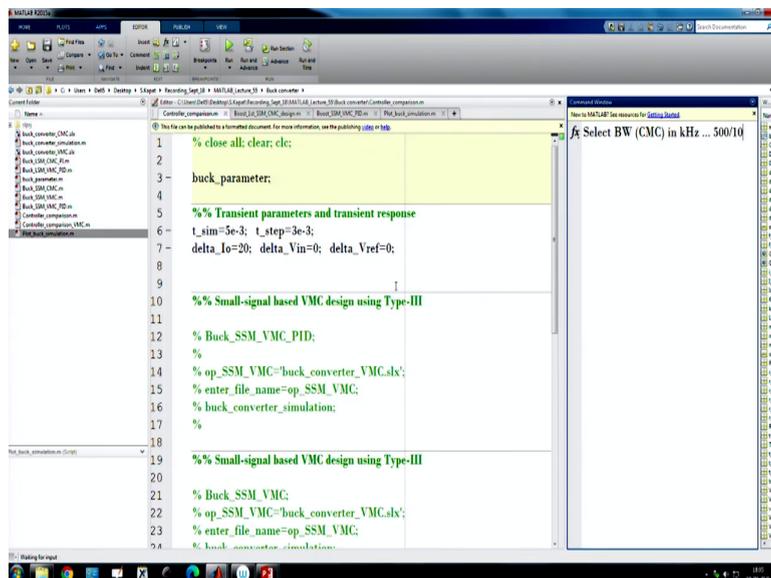
(Refer Slide Time: 14:55)



```
1 figure(1)
2
3 plot(t_scale,i_L,'Linewidth', 2); hold on; grid on;
4 xlabel('Time (ms)', 'FontSize', 15);
5 ylabel('Inductor current (A)', 'FontSize', 15);
6
7 figure(2)
8 plot(t_scale,V_o,'Linewidth', 2); hold on; grid on;
9 xlabel('Time (ms)', 'FontSize', 15);
10 ylabel('Output voltage (V)', 'FontSize', 15);
11
```

Let us say we are using red color and let us run it. So, if you run it.

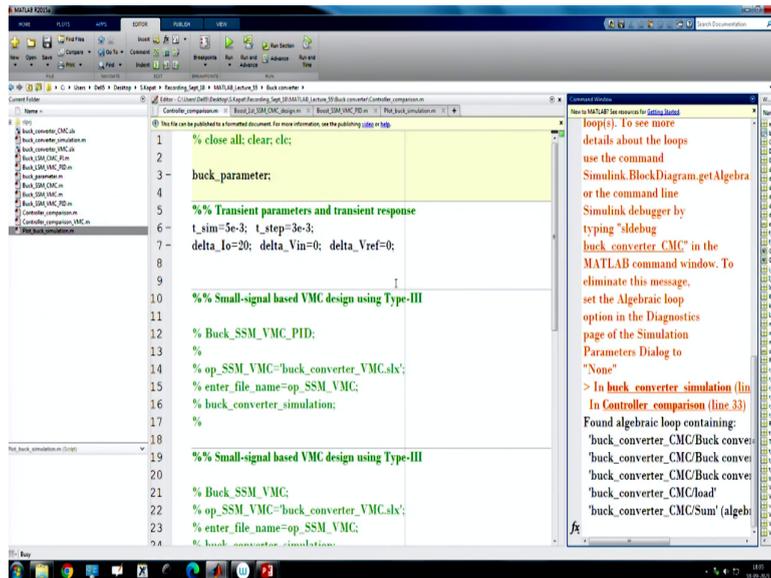
(Refer Slide Time: 15:03)



```
1 % close all; clear; clc;
2
3 buck_parameter;
4
5 %% Transient parameters and transient response
6 t_sim=5e-3; t_step=3e-3;
7 delta_Io=20; delta_Vin=0; delta_Vref=0;
8
9
10 %% Small-signal based VMC design using Type-III
11
12 % Buck_SSM_VMC_PID;
13 %
14 % op_SSM_VMC='buck_converter_VMC.slx';
15 % enter_file_name=op_SSM_VMC;
16 % buck_converter_simulation;
17 %
18
19 %% Small-signal based VMC design using Type-III
20
21 % Buck_SSM_VMC;
22 % op_SSM_VMC='buck_converter_VMC.slx';
23 % enter_file_name=op_SSM_VMC;
24 % buck_converter_simulation;
```

We want to hold it then. So, here also we are you know, 500 divided by so, let us go on 10.

(Refer Slide Time: 15:14)

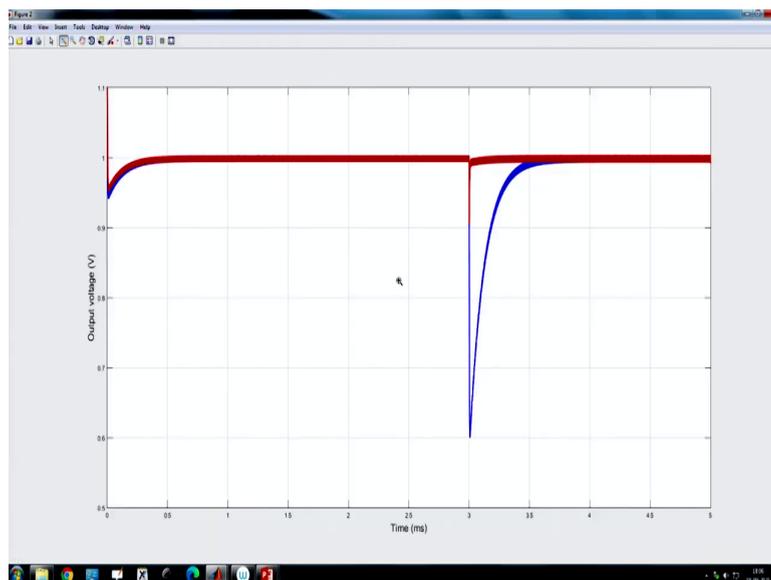


```
1 % close all; clear; clc;
2
3 buck_parameter;
4
5 %% Transient parameters and transient response
6 t_sim=5e-3; t_step=3e-3;
7 delta_Io=20; delta_Vin=0; delta_Vref=0;
8
9
10 %% Small-signal based VMC design using Type-III
11
12 % Buck_SSM_VMC_PID;
13 %
14 % op_SSM_VMC='buck_converter_VMC.slx';
15 % enter_file_name='op_SSM_VMC;
16 % buck_converter_simulation;
17 %
18
19 %% Small-signal based VMC design using Type-III
20
21 % Buck_SSM_VMC;
22 % op_SSM_VMC='buck_converter_VMC.slx';
23 % enter_file_name='op_SSM_VMC;
24 % buck_converter_simulation;
```

loop(s). To see more details about the loops use the command Simulink.BlockDiagram.getAlgebraic or the command line Simulink debugger by typing "sdebug buck_converter_CMC" in the MATLAB command window. To eliminate this message, set the Algebraic loop option in the Diagnostics page of the Simulation Parameters Dialog to "None"

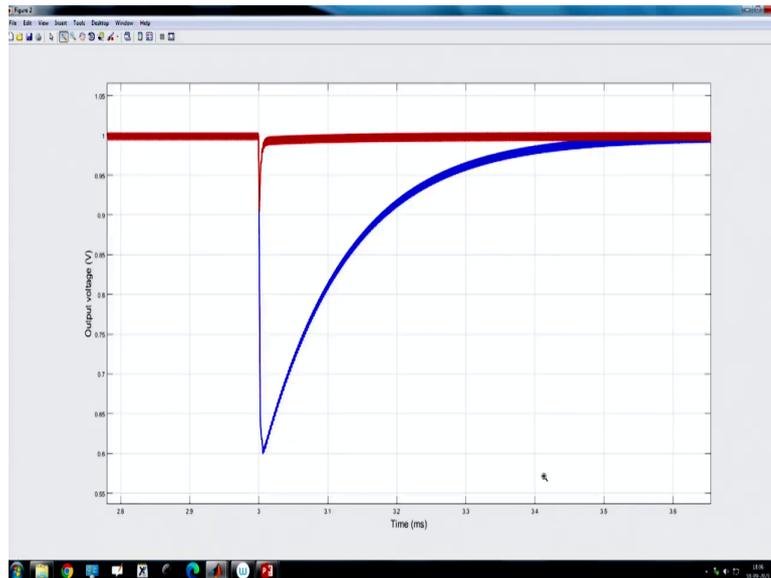
```
> In buck_converter_simulation (lin
In Controller_comparison (line 33)
Found algebraic loop containing:
'buck_converter_CMC/Buck conver
'buck_converter_CMC/Buck conver
'buck_converter_CMC/load
'buck_converter_CMC/Sum' (algeb
```

(Refer Slide Time: 15:17)



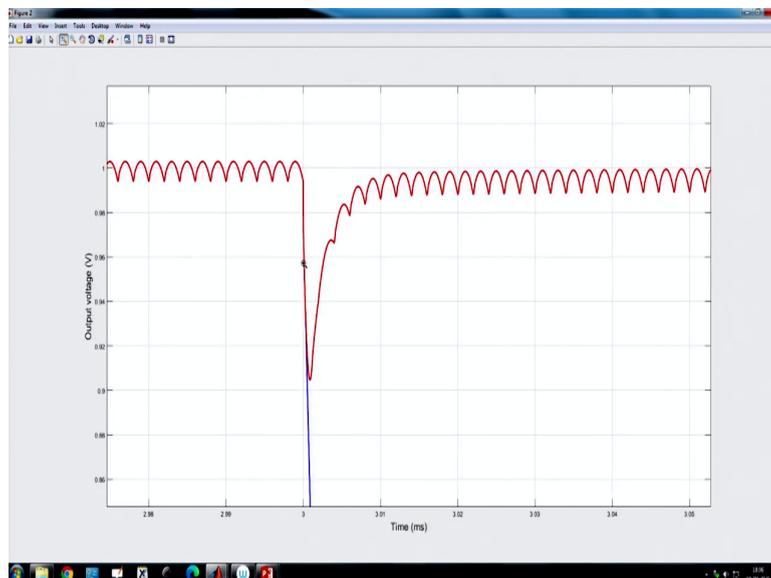
And see it is much faster.

(Refer Slide Time: 15:20)



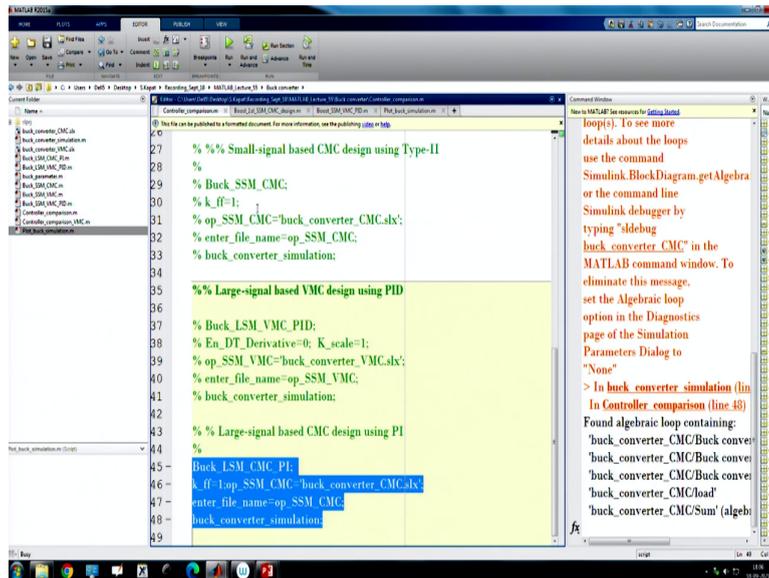
Because we have incorporated the load current feedforward.

(Refer Slide Time: 15:22)



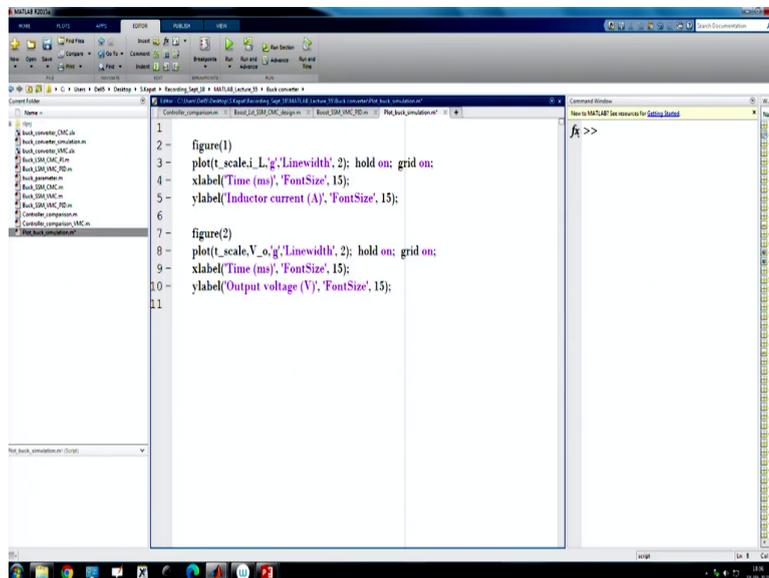
And we have discussed the load current feedforward can drastically improve the performance right. Now, we are going to large-signal tuning ok for current mode.

(Refer Slide Time: 15:35)



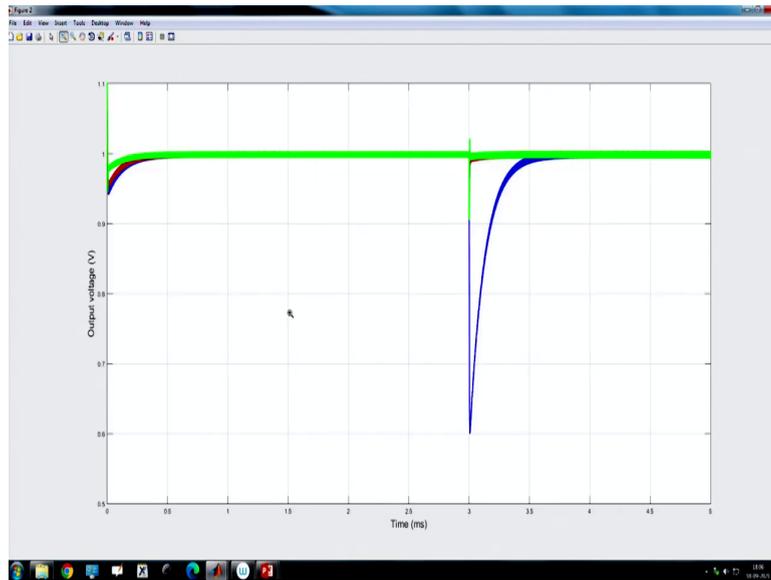
Where we need to incorporate loads, load current feedforward and we discuss that we can use load you know we can estimate the load current or we can use a physical sensing right.

(Refer Slide Time: 15:47)



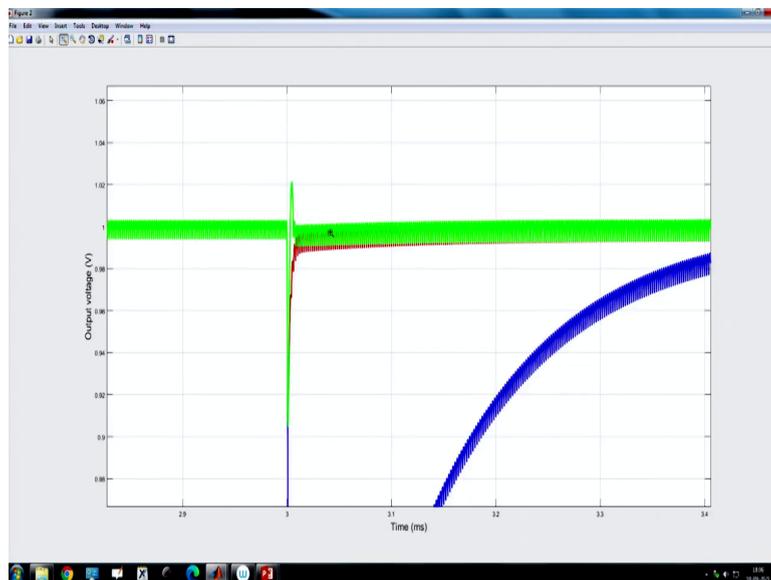
So, estimating load current is a good solution in digital control solution. So, let us do that and we will run it now using green color.

(Refer Slide Time: 15:59)

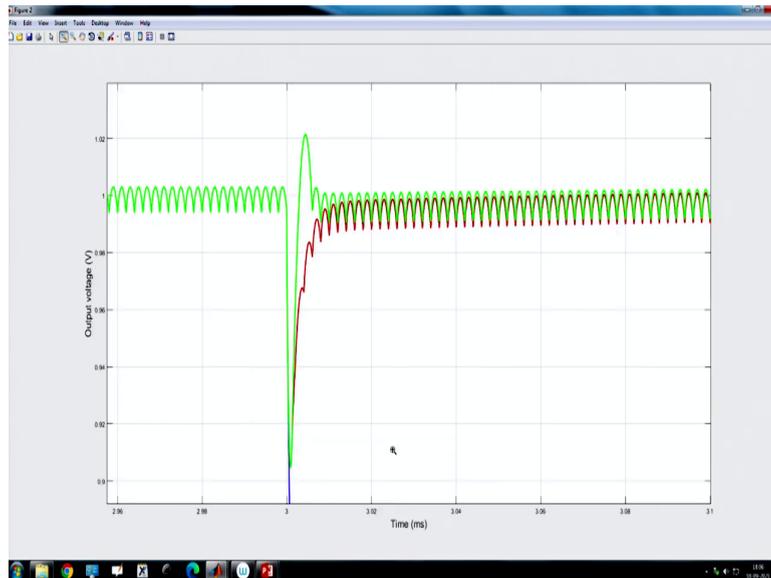


And see what happens yes. So, you see.

(Refer Slide Time: 16:01)

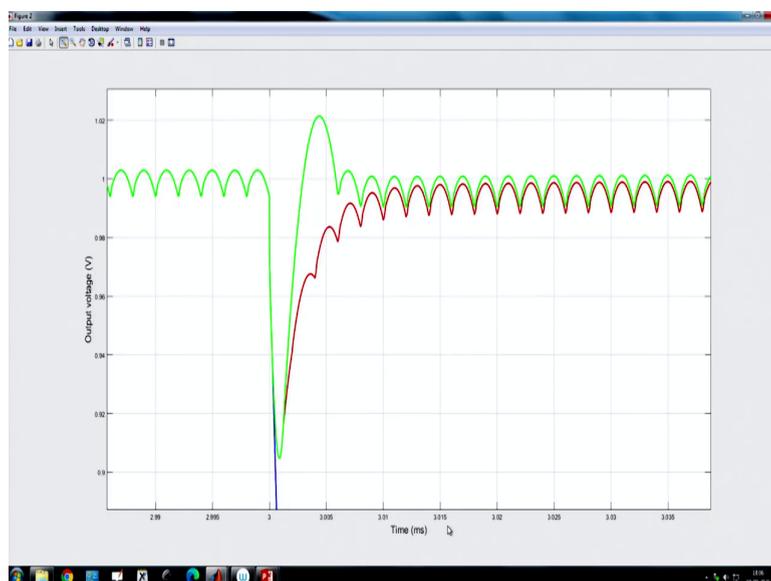


(Refer Slide Time: 16:02)



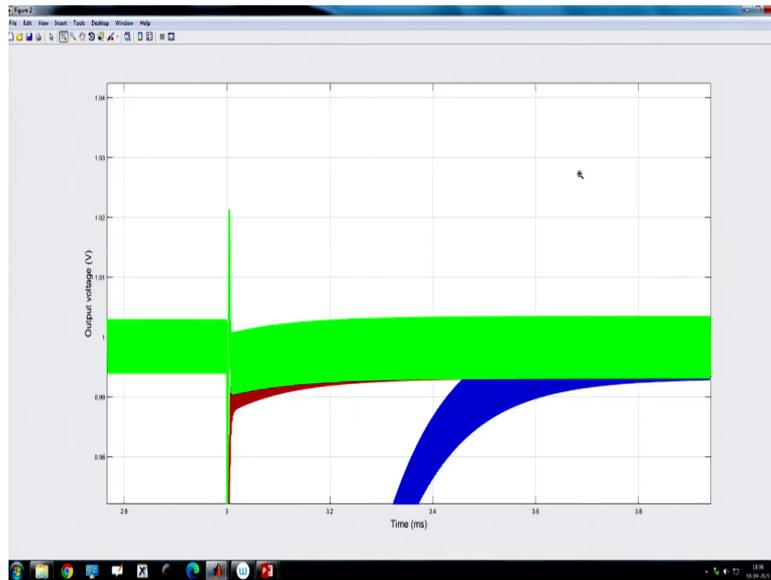
When we consider our tuning.

(Refer Slide Time: 16:06)



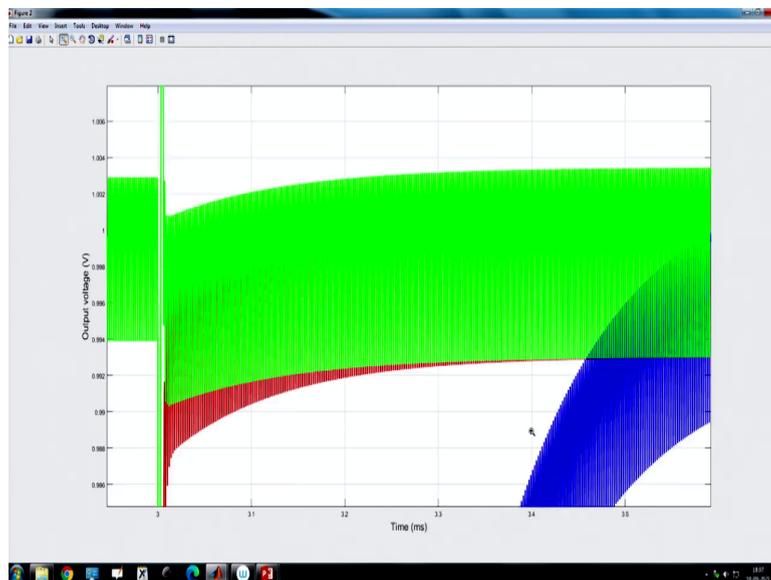
Large-signal tuning it is recovering in almost three cycles whereas, load feedforward is taking some more time using the regular current mode control with load feedforward.

(Refer Slide Time: 16:17)

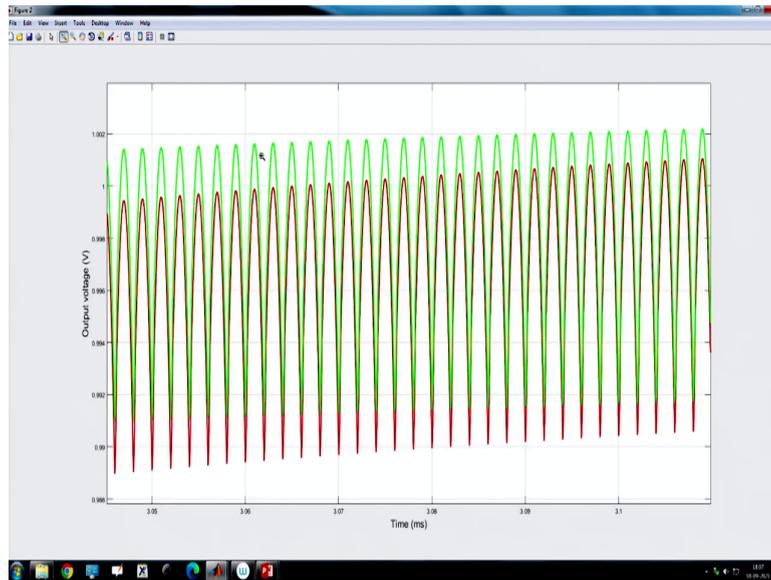


And it is somewhat sluggish response.

(Refer Slide Time: 16:19)

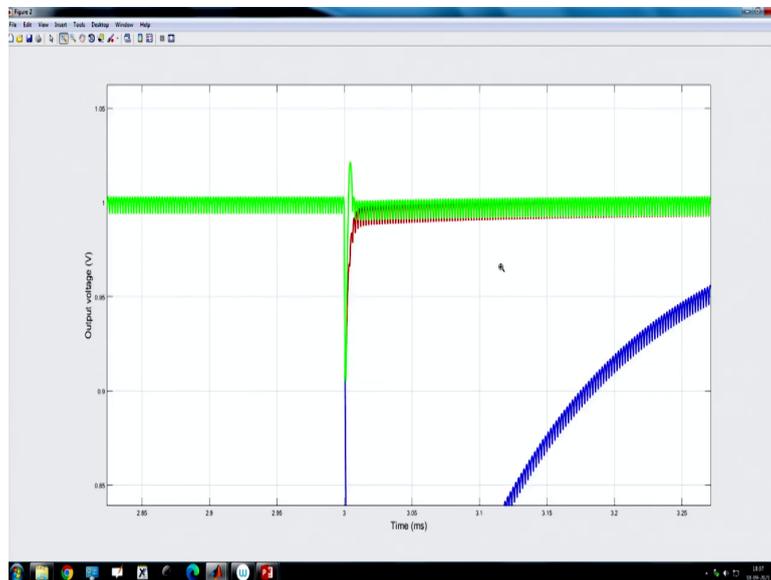


(Refer Slide Time: 16:21)



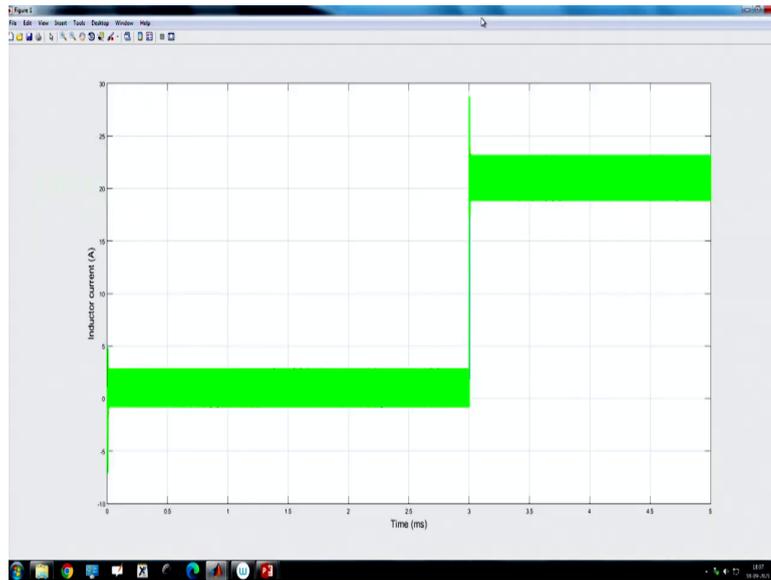
So, it takes few more cycles to reach steady state ok. But, it is good that both are giving fine with load feedforward.

(Refer Slide Time: 16:25)



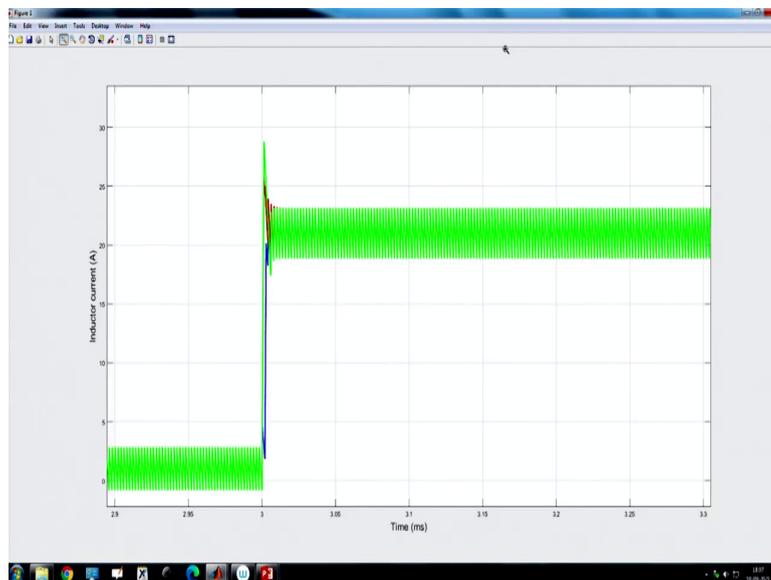
But, optimal gain makes it even faster right.

(Refer Slide Time: 16:31)

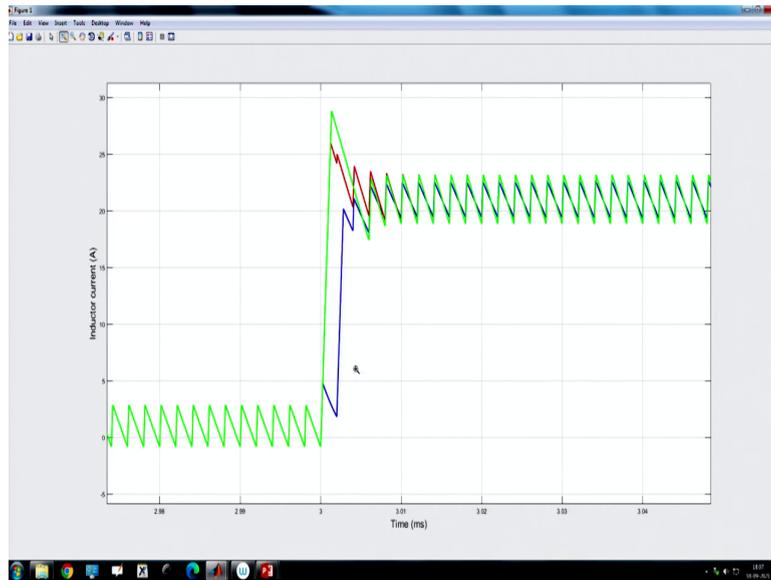


So, you can just check it here.

(Refer Slide Time: 16:36)

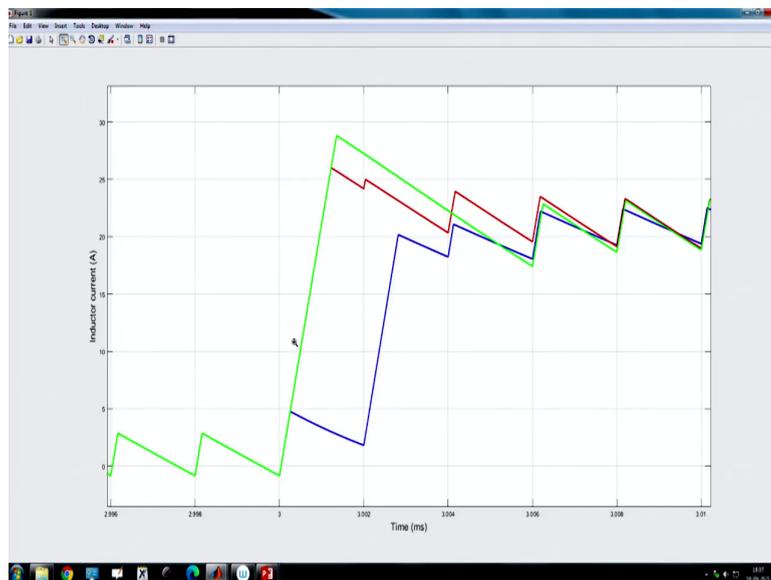


(Refer Slide Time: 16:38)



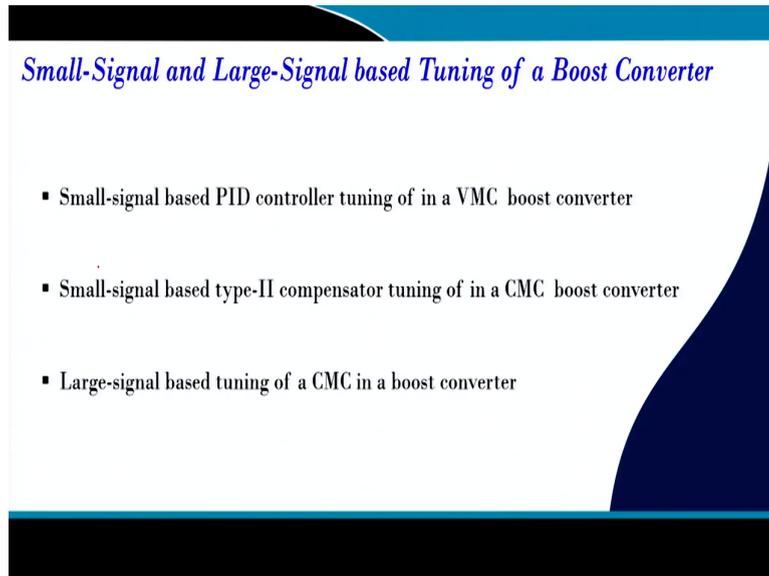
So, these two are the responses.

(Refer Slide Time: 16:40)



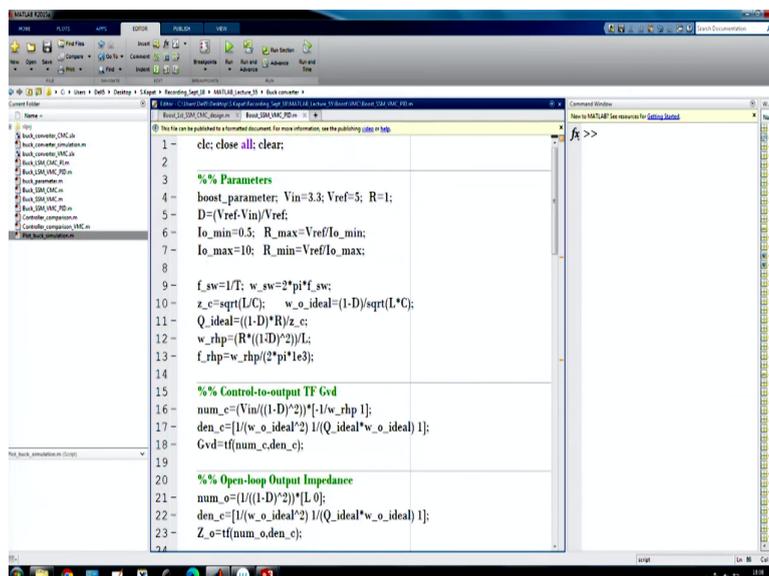
But, because of load feedforward it is allowing the duty ratio to saturate. So, that, but in regular current mode control because of on off operation it is slowing down the transient performance and which is quite obvious and that we have discussed. So, we have discussed PID tuning, large-signal, small-signal for voltage mode current mode in all case studies.

(Refer Slide Time: 16:58)



Now, we are going to consider a boost converter. So, in a boost converter, we are going to consider reference transient using small-signal as well as large-signal ok. So, let us go now we will just stop all this waveform. So, let us go to our boost converter case study, ok. So, we have to consider buck.

(Refer Slide Time: 17:27)



Now, in boost converter first we will consider a voltage mode control technique where we are considering 1 ohm load resistance, 3.3 volt input, 5 volt output.

(Refer Slide Time: 17:36)

```

64 - G_vgc=Gvg/(1+G_loop);    %% Closed-loop audio susc.
65 -
66 - %% Frequency response
67 - figure(3)
68 - % bode(Z_o,'b');
69 - % hold on;
70 - % bode(Z_oc,'-b');
71 - % hold on;
72 - bode(G_loop,'r');
73 - hold on; grid on;
74 - [Gm,Pm,Wcg,Wcp] = margin(G_loop);
75 -
76 -
77 - %% Transient parameters and transient response
78 - t_sim=5e-3; t_step=3e-3;
79 - delta_Io=0; delta_Vin=0; delta_Vref=0.2;
80 -
81 - [y_s,t_s]=step(G_cl,(t_sim-t_step));
82 - v_ac=delta_Vref*y_s;
83 -
84 - boost_converter_simulation;
85 -
86 - clc;
  
```

And we are applying a load step transient of you know a reference transient of 0.2 volt ok. So, let us first run it and we will use.

(Refer Slide Time: 17:47)

```

1 - clc; close all; clear;
2 -
3 - %% Parameters
4 - boost_parameter; Vin=3.3; Vref=5; R=1;
5 - D=(Vref-Vin)/Vref;
6 - Io_min=0.5; R_max=Vref/Io_min;
7 - Io_max=10; R_min=Vref/Io_max;
8 -
9 - f_sw=1/T; w_sw=2*pi*f_sw;
10 - z_c=sqrt(L/C); w_o_ideal=(1-D)/sqrt(L*C);
11 - Q_ideal=(1-D)*R/z_c;
12 - w_rhp=(R*((1-D)^2))/L;
13 - f_rhp=w_rhp/(2*pi*1e3);
14 -
15 - %% Control-to-output TF Gvd
16 - num_c=(Vin*((1-D)^2))*(-1/w_rhp);
17 - den_c=(1/(w_o_ideal^2))/((Q_ideal*w_o_ideal));
18 - Gvd=tf(num_c,den_c);
19 -
20 - %% Open-loop Output Impedance
21 - num_o=(1/((1-D)^2))*L;
22 - den_o=(1/(w_o_ideal^2))/((Q_ideal*w_o_ideal));
23 - Z_o=tf(num_o,den_o);
  
```

Select BW fraction of f_rhp 1/5

p = 0.2000

Let us run it. So, it will ask for f rhp and we know for voltage mode we generally go for one fifth of the rhp.

(Refer Slide Time: 17:54)

The screenshot shows a MATLAB script for a boost converter simulation. The code includes parameter definitions, transfer function calculations for the control-to-output TF Gvd and open-loop output impedance, and a warning in the Command Window about algebraic loops.

```

1 clc; close all; clear;
2
3 %% Parameters
4 boost_parameter; Vin=3.3; Vref=5; R=1;
5 D=(Vref-Vin)/Vref;
6 Io_min=0.5; R_max=Vref/Io_min;
7 Io_max=10; R_min=Vref/Io_max;
8
9 f_sw=1/T; w_sw=2*pi*f_sw;
10 z_c=sqrt(L/C); w_o_ideal=(1-D)/sqrt(L*C);
11 Q_ideal=(1-D)*R/z_c;
12 w_rhp=(R*((1-D)^2))/L;
13 f_rhp=w_rhp/(2*pi*1e3);
14
15 %% Control-to-output TF Gvd
16 num_c=(Vin*((1-D)^2))*1/w_rhp 1];
17 den_c=1/(w_o_ideal^2) 1/(Q_ideal*w_o_ideal 1];
18 Gvd=tf(num_c,den_c);
19
20 %% Open-loop Output Impedance
21 num_o=1/(1-D)^2 1/L 0];
22 den_o=1/(w_o_ideal^2) 1/(Q_ideal*w_o_ideal 1];
23 Z_o=tf(num_o,den_o);
24

```

Command Window Output:

```

Warning: Discontinuities detected within algebraic loop(s), may have trouble solving
> In boost_converter_simulation (line 14)
In Boost_SSM_VMC_PID (line 84)

```

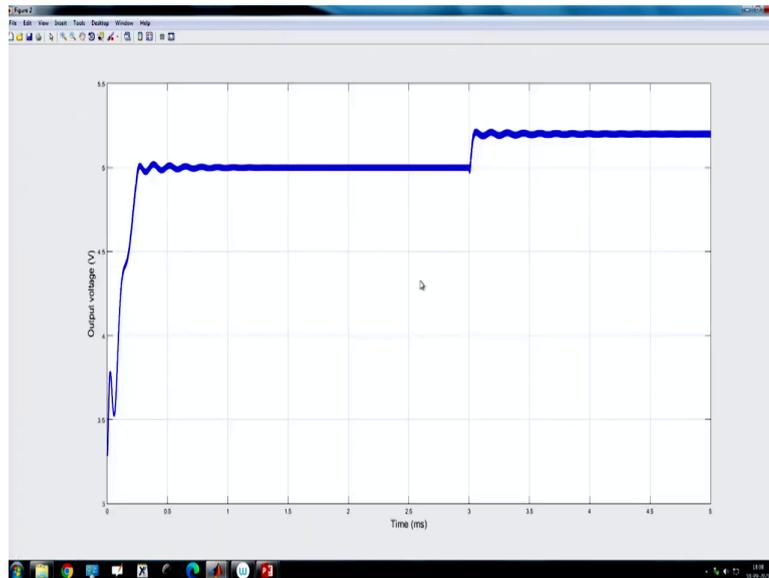
(Refer Slide Time: 17:55)

The screenshot shows the same MATLAB script as above, but with a Bode Diagram plot overlaid on the code editor. The plot shows the magnitude and phase of the transfer function Gvd. The magnitude plot shows a low-pass characteristic with a roll-off at high frequencies. The phase plot shows a phase shift from 0 to -180 degrees.

The Bode Diagram plot shows Magnitude (dB) on the y-axis (ranging from -40 to 40) and Frequency (rad/s) on the x-axis (logarithmic scale from 10⁰ to 10⁵). The phase plot shows Phase (deg) on the y-axis (ranging from -180 to 0) and Frequency (rad/s) on the x-axis (logarithmic scale from 10⁰ to 10⁵).

So, let us do that.

(Refer Slide Time: 17:57)



So, this is the one fifth of the rhp. So, this is a blue color.

(Refer Slide Time: 18:05)

```
1  
2- figure(1)  
3- plot(t_scale,i,'LineWidth', 2); hold on; grid on;  
4- xlabel('Time (ms)', 'FontSize', 15);  
5- ylabel('Inductor current (A)', 'FontSize', 15);  
6  
7- figure(2)  
8- plot(t_scale,V_o,'LineWidth', 2); hold on; grid on;  
9- xlabel('Time (ms)', 'FontSize', 15);  
10- ylabel('Output voltage (V)', 'FontSize', 15);  
11
```

Now, we are using the same red color for current mode control. So, we will go to current mode control we will hold it.

(Refer Slide Time: 18:13)

```
19 % Open-loop Output Impedance
20
21 num_o=(R/2)*[1/w_esr 1];
22 den_o=[1/w_p 1];
23 Z_o=tf(num_o,den_o);
24
25
26 %% Small-signal design of Type-II Compensator
27
28 p=input('Select BW fraction of f_rhp ');
29 theta_rad=atan((2*p)/(1-(p^2)));
30 theta=rad2deg(theta_rad);
31 PM=90-theta; w_c=p*w_rhp;
32 theta=(90-PM); theta_rad=deg2rad(theta);
33 K_c=(p*w_rhp)/(K_g); w_cp=w_rhp; k_n=0;
34 num_con=K_c*den_c;
35 den_con=[1/w_cp 1 0];
36 Ge=tf(num_con,den_con);
37
38 %% Large-signal tuning of PI controller
39
40 % K_cp=1; k_n=Vref/Vin;
41 % K_vp=15.5;
42 % k_n=100000;
```

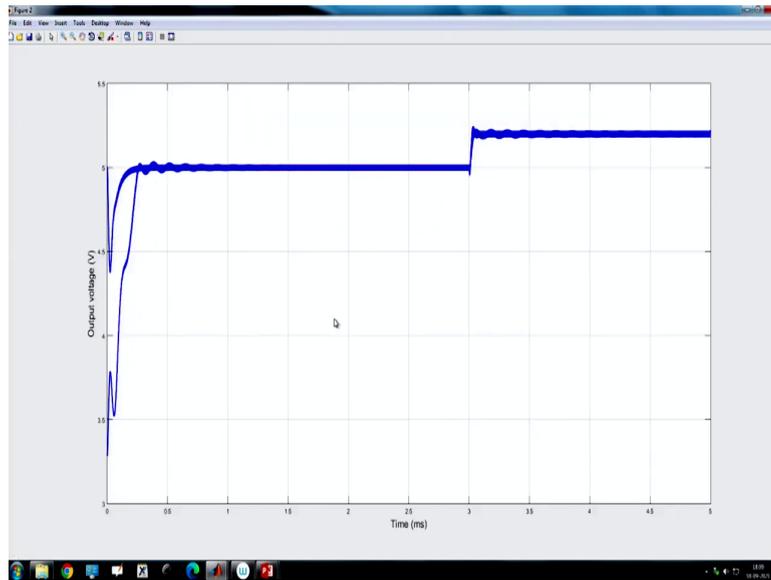
Now, we will go for traditional current mode control where we can go up to one third of the rhp zero. So, one third of the rhp zero we can go right.

(Refer Slide Time: 18:24)

```
Warning: Discontinuities detected within algebraic loop(s), may have trouble solving
> In boost_converter_simulation (line 1)
In Boost_1st_SSM_CMC design (line 36)
```

One third of the rhp zero we can go.

(Refer Slide Time: 18:26)



And let us check ok. So, I think we should change the color.

(Refer Slide Time: 18:35)

```
1 figure(1)
2 plot(t_scale,i_L,'LineWidth',2); hold on; grid on;
3 xlabel('Time (ms)',FontSize,15);
4 ylabel('Inductor current (A)',FontSize,15);
5
6
7 figure(2)
8 plot(t_scale,v_o,'LineWidth',2); hold on; grid on;
9 xlabel('Time (ms)',FontSize,15);
10 ylabel('Output voltage (V)',FontSize,15);
11
```

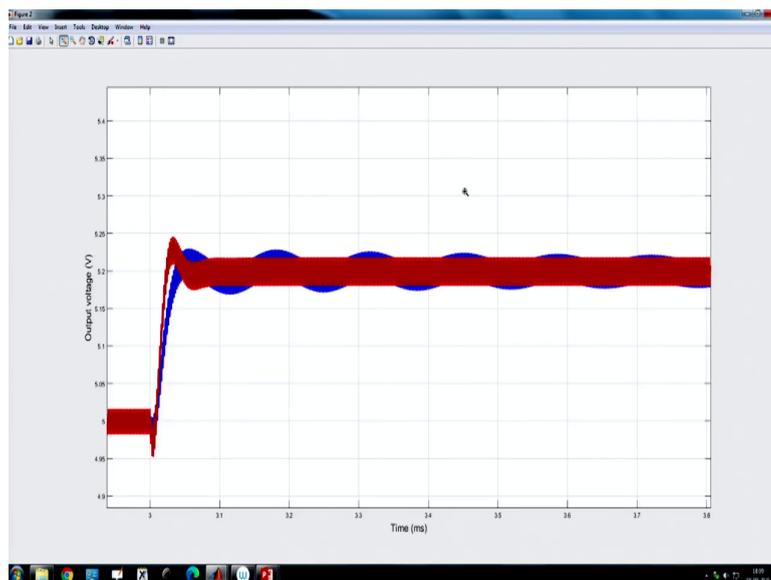
```
>> figure(1)
>> plot(t_scale,i_L,'LineWidth',2); ho
>> xlabel('Time (ms)',FontSize,15);
>> ylabel('Inductor current (A)',FontSi
>>
>> figure(2)
>> plot(t_scale,v_o,'LineWidth',2); hc
>> xlabel('Time (ms)',FontSize,15);
>> ylabel('Output voltage (V)',FontSize
>> ft >>
```

So, we should change the color one third of the rhp zero. Just hold on. We are redrawing this we are ok.

(Refer Slide Time: 18:43)

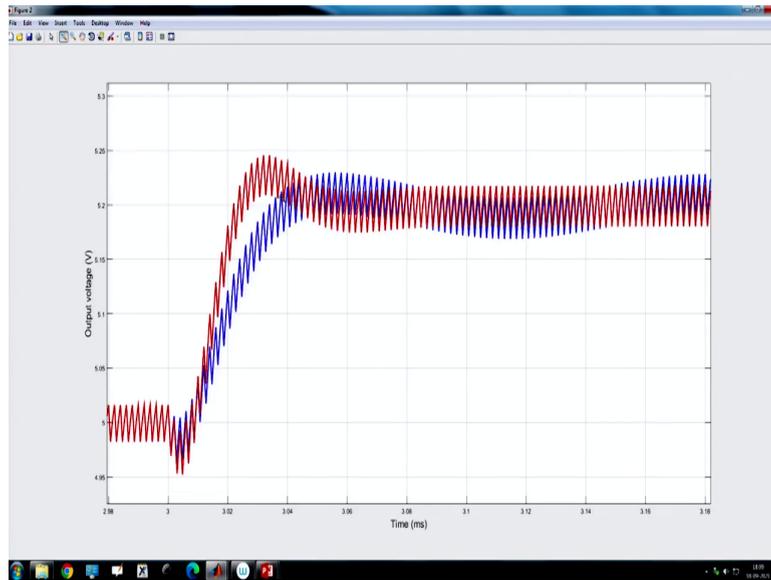


(Refer Slide Time: 18:47)



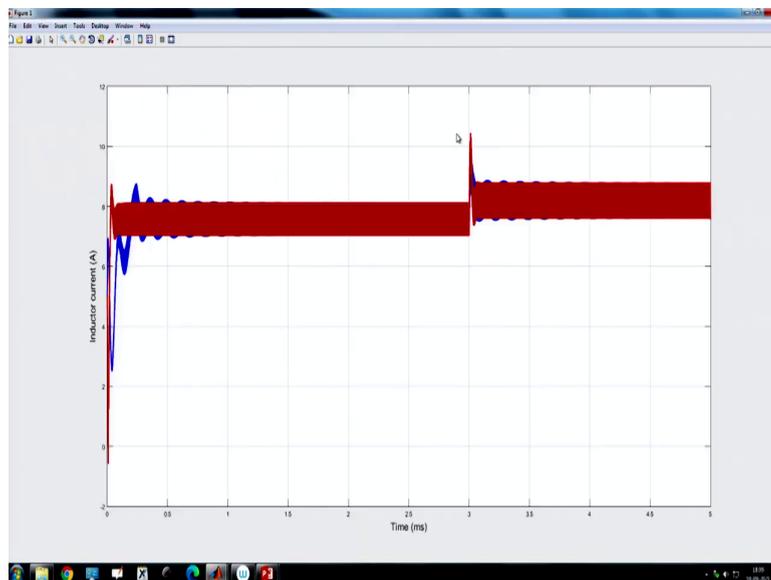
So, we have drawn one for voltage mode control.

(Refer Slide Time: 18:49)



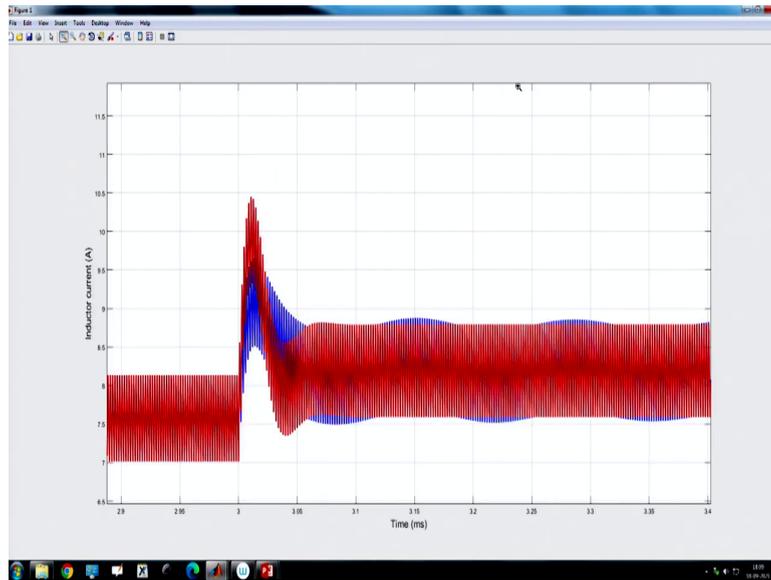
And the other for current mode control.

(Refer Slide Time: 18:51)



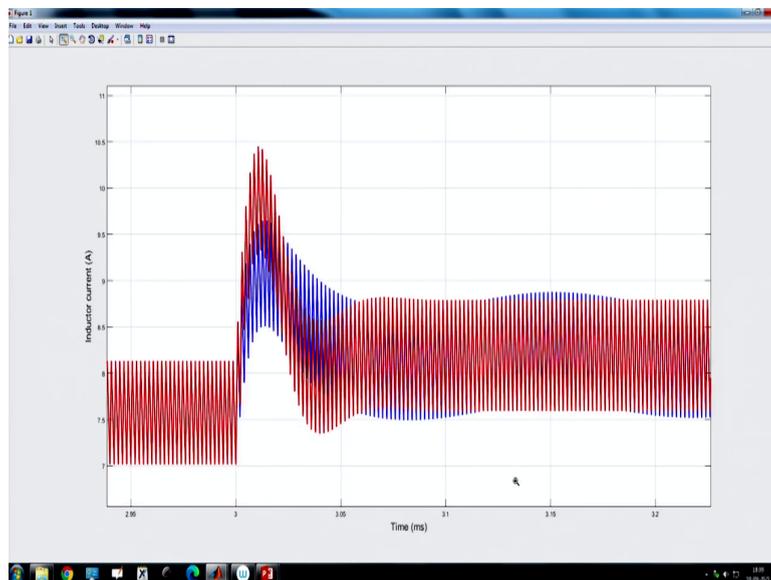
And these two are their inductor current.

(Refer Slide Time: 18:55)

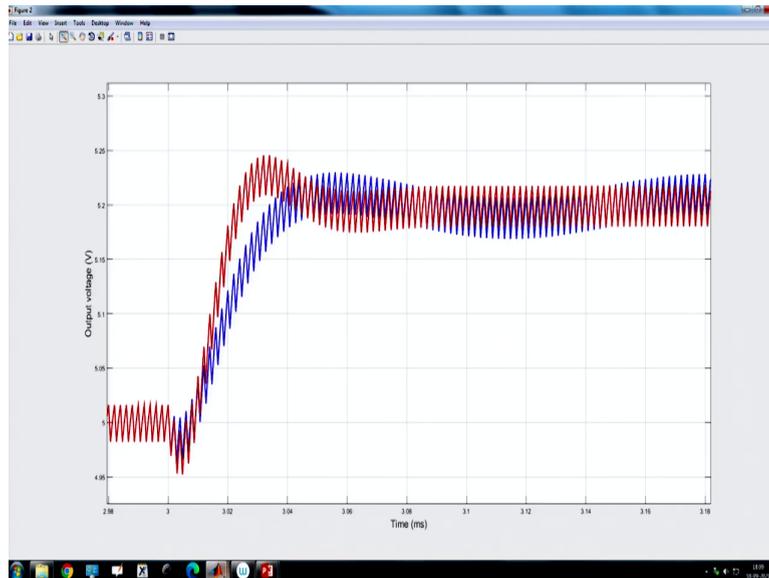


For voltage mode, we took one fifth of the rhp zero.

(Refer Slide Time: 18:57)



(Refer Slide Time: 18:59)



For current mode, we took one third of the rhp zero and it shows a non-minimum phase behaviour. Now, we are going for large-signal based tuning.

(Refer Slide Time: 19:10)

```
%% Small-signal design of Type-II Compensator
% p= input(Select BW fraction of f_ohp);
% theta_rad=atan(2*p)/(1-(p^2));
% theta_rad=rad2deg(theta_rad);
% PM=90-theta; w_cp=p*w_ohp;
% theta=(90-PM); theta_rad=deg2rad(theta);
% K_cp=(p*w_ohp)/(K_ol); w_cp=w_ohp; k_n=0;
% num_con=K_cp*den_con;
% den_con=[1 w_cp 1 0];
% Gc=tf(num_con,den_con);

%% Large-signal tuning of PI controller
%
% K_cp=1; k_n=Vref/Vin;
% K_vp=15.5;
% K_vi=40000;
% num_con=[K_vp K_vi];
% den_con=[1 0];
% Gc=tf(num_con,den_con);
```

Where we will disable this small-signal tuning comment and we are going for large-signal tuning.

(Refer Slide Time: 19:14)

```

31 % PM=90-theta; w_c=p*w_rhp;
32 % theta=(90-PM); theta_rad=deg2rad(theta);
33 % K_c=(p*w_rhp)/(K_g); w_cp=w_rhp; k_n=0;
34 % num_con=K_c*den_c;
35 % den_con=[1w_cp 10];
36 % Ge=tf(num_con,den_con);
37
38 %% Large-signal tuning of PI controller
39 %
40 K_cp=1; k_n=Vref/Vin;
41 K_vp=15.5;
42 K_vi=40000;
43
44 num_con=[K_vp K_vi];
45 den_con=[1 0];
46 Ge=tf(num_con,den_con);
47
48
49 %% Transient parameters and transient response
50 t_sim=5e-3; t_step=3e-3;
51
52 L_max=15; L_min=-10;
53
54 Delta_Ia=0; Delta_Vin=0; Delta_Vref=0;

```

And as I told for large-signal tuning, the voltage can be derived using the expression.

(Refer Slide Time: 19:22)

PID Control Tuning in VMC Boost Converter
Lec. 36

- An ideal boost converter

$$G_{vd}(s) = \frac{V_{in}}{(1-D)^2} \times \frac{\left(1 - \frac{s}{w_{rhp}}\right)}{\left(1 + \frac{s}{Qw_0} + \frac{s^2}{w_0^2}\right)}$$

where,

$$w_0 = \frac{(1-D)}{\sqrt{LC}} \quad w_{rhp} = \frac{R(1-D)^2}{L} \quad Q = \frac{R(1-D)}{z_c} \quad z_c = \sqrt{\frac{L}{C}}$$

So, if you go for that. So, let us first derive in voltage mode control. We know how to you know how to derive the gains and we have discussed in lecture number I think 36, we have discussed in detail different parameter of the open loop plan.

(Refer Slide Time: 19:40)

Gain Scheduling – PID Controller Tuning in VMC Boost Converter

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

$$k_1 = \frac{1}{Q w_0}, \quad k_2 = \frac{1}{w_0^2}, \quad \tau_D = \frac{1}{w_{rhp}} \quad K_i = \frac{w_c (1-D)^2}{F_m V_{in}}$$

where,

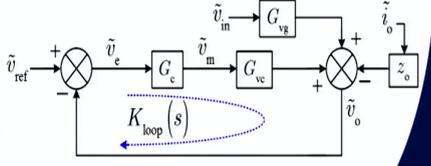
$$w_0 = \frac{(1-D)}{\sqrt{LC}} \quad w_{rhp} = \frac{R(1-D)^2}{L} \quad Q = \frac{R(1-D)}{z_c} \quad z_c = \sqrt{\frac{L}{C}}$$


Then for the PID controller design what will be the optimal PID gain, k 1, k 2 these are here k 1, k 2, then tau d is a band limited derivative we are considering, integral gain this will require the crossover frequency and from here we are getting all the controller gain that we have discussed.

(Refer Slide Time: 19:59)

Type-II Compensator Design of a CMC Boost Converter

$$K_{loop}(s) = \frac{k_g \left(1 + \frac{s}{\omega_{esr}} \right) \left(1 - \frac{s}{\omega_{rhp}} \right)}{\left(1 + \frac{s}{\omega_p} \right)} \times G_c$$

$$G_c = \frac{k_c \left(1 + \frac{s}{\omega_{cz}} \right)}{s \left(1 + \frac{s}{\omega_{cp}} \right)}$$



And we know that how to current mode control we have also discussed you know; that means, you know there is esr zero, rhp zero, and we can go to type 2 compensator design.

(Refer Slide Time: 20:11)

Type-II Compensator Design of a CMC Boost Converter

$$G_c = \frac{k_c \left(1 + \frac{s}{w_{cz}}\right)}{s \left(1 + \frac{s}{w_{cp}}\right)}$$

$$w_{cz} = w_p, w_{cp} = w_{rhp}$$

$$PM_{desired} = 90^\circ - \tan^{-1} \left(\frac{2w_n'}{1 - w_n'^2} \right); w_n' = \frac{w_c}{w_{rhp}} \approx \frac{1}{3}$$

$$k_c = \frac{2w_c}{R(1-D)}$$

And the design parameter we have also discussed how to set. That means we have to take omega c by omega rhp and that we are taking like a one third; that means we are taking 1 by 3 and this will compute the gain that we have derived.

(Refer Slide Time: 20:29)

Large-Signal PI Controller Tuning Parameters for a Boost Converter

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

During step-up transient

$$k_p \approx \frac{C}{L} \times \frac{v_{in}}{i_o} \times \left(1 + \sqrt{\frac{\Delta i_o}{2Di_o}}\right)^{-1}$$

During step-down transient

$$k_p \approx \frac{C}{L} \times \frac{v_{in}}{i_o}$$

$$k_n = \frac{v_{ref}}{v_{in}}$$

For large-signal tuning, we know the expression and as I told the expression is somewhat you know is not very complex somewhat. So, we have computed this value. This is for load transient we can also do for reference transient. We have computed this value and those

values are plugged in into this MATLAB code and we are comparing now we are using a different trace.

(Refer Slide Time: 20:54)

```

1 figure(1)
2 plot(t_scale,i_L,'LineWidth',2); hold on; grid on;
3 xlabel('Time (ms)', 'FontSize', 15);
4 ylabel('Inductor current (A)', 'FontSize', 15);
5
6
7 figure(2)
8 plot(t_scale,V_o,'LineWidth',2); hold on; grid on;
9 xlabel('Time (ms)', 'FontSize', 15);
10 ylabel('Output voltage (V)', 'FontSize', 15);
11

```

```

>> figure(1)
plot(t_scale,i_L,'LineWidth',2); ho
xlabel('Time (ms)', 'FontSize', 15);
ylabel('Inductor current (A)', FontSi

figure(2)
plot(t_scale,V_o,'LineWidth',2); ho
xlabel('Time (ms)', 'FontSize', 15);
ylabel('Output voltage (V)', FontSize
fx >>

```

So, we are using green color trace; green color trace.

(Refer Slide Time: 21:01)

```

31 %% PM=90-theta; w_cp=w_rhp;
32 %% theta=90-PM; theta_rad=deg2rad(theta);
33 %% K_c=(p*w_rhp)/(K_g); w_cp=w_rhp; k_n=0;
34 %% num_con=K_c*den_c;
35 %% den_con=[1 w_cp 10];
36 %% Gc=tf(num_con,den_con);
37
38 %% Large-signal tuning of PI controller
39 %%
40 K_cp=1; k_n=Vref/Vin;
41 K_vp=15.5;
42 K_vi=40000;
43
44 num_con=[K_vp K_vi];
45 den_con=[1 0];
46 Gc=tf(num_con,den_con);
47
48
49 %% Transient parameters and transient response
50 t_sim=5e-3; t_step=3e-3;
51
52 I_max=15; I_min=-10;
53
54 Delta_I_o=0; Delta_V_o=0; Delta_V_ref=0.2;

```

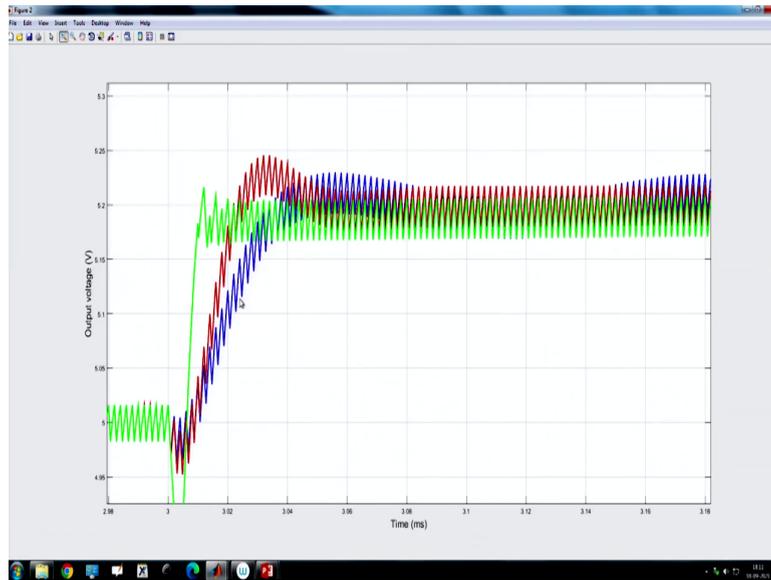
```

Dialog to 'None'
> In boost_converter_simulation (lin
In Boost_1st_SSM_CMC design (lin
Found algebraic loop containing:
'boost_converter_CMC/load'
'boost_converter_CMC/Sun' (algeb
'boost_converter_CMC/PWM CMC
'boost_converter_CMC/PWM CMC
'boost_converter_CMC/PWM CMC
'boost_converter_CMC/PWM CMC
'boost_converter_CMC/Boost Conv
'boost_converter_CMC/Boost Conv
'boost_converter_CMC/Boost Conv
'boost_converter_CMC/Boost Conv
Warning: Discontinuities
detected within algebraic
loop(s), may have trouble
solving
> In boost_converter_simulation (lin
In Boost_1st_SSM_CMC design (lin
fx

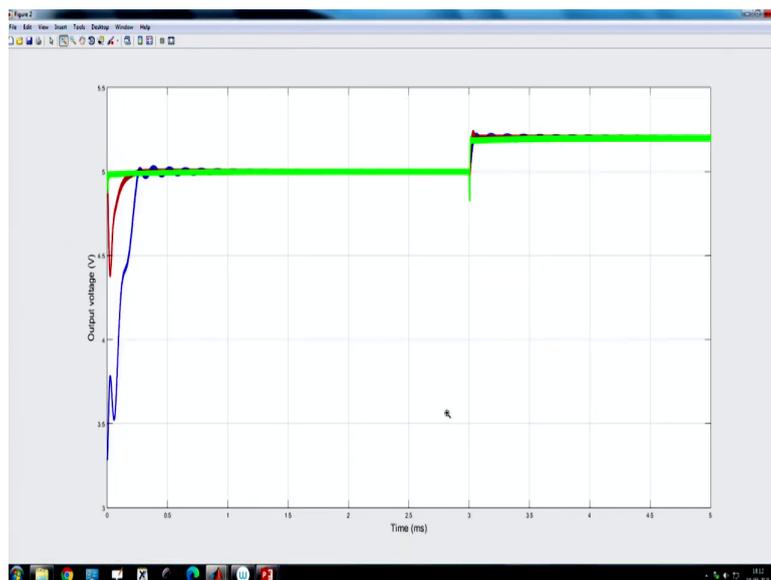
```

And we are using now our large-signal based tuning.

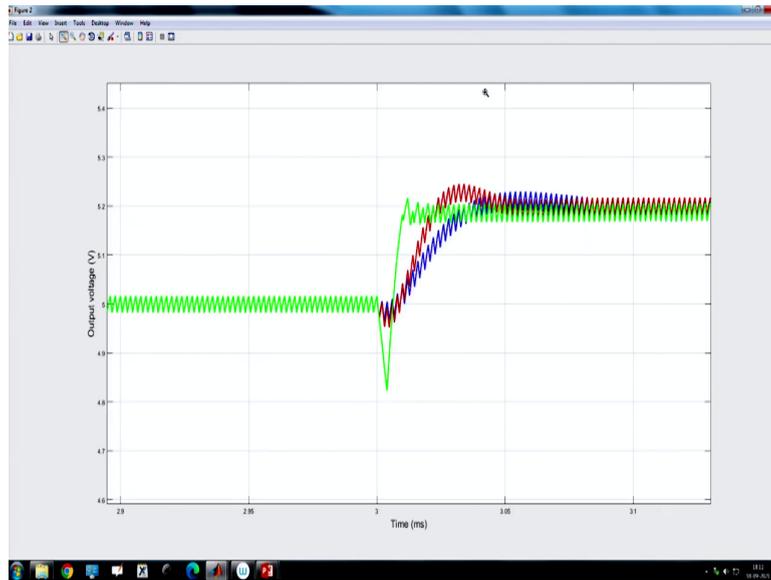
(Refer Slide Time: 21:03)



(Refer Slide Time: 21:06)

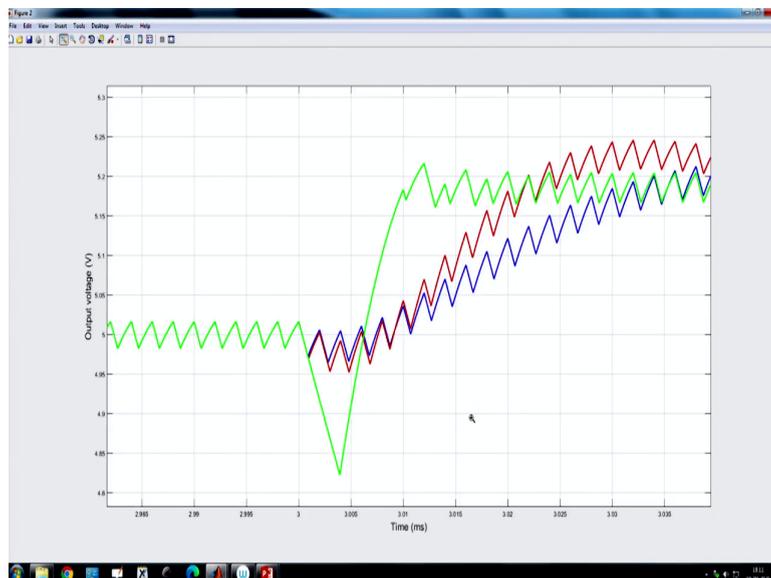


(Refer Slide Time: 21:07)



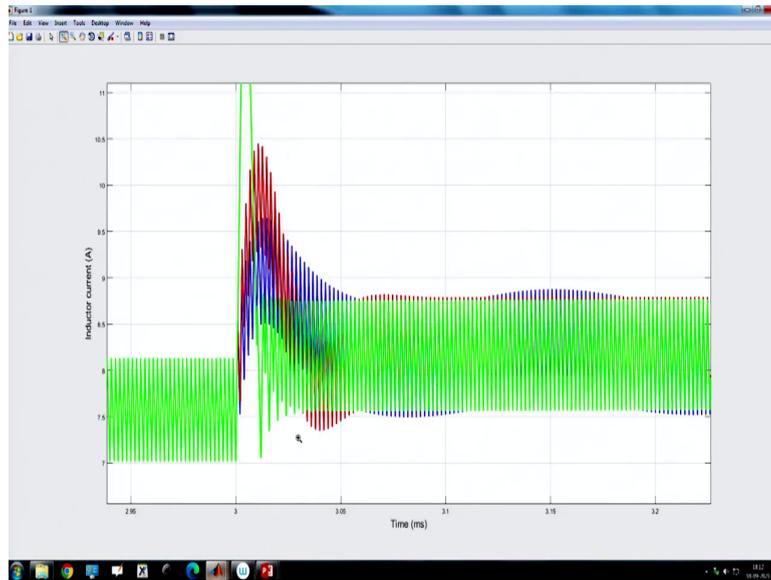
So, in large-signal based tuning you will find.

(Refer Slide Time: 21:09)



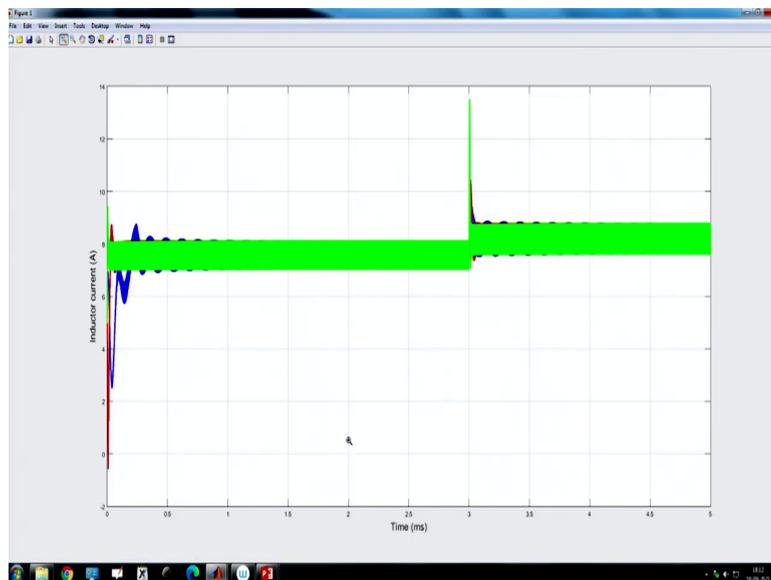
That it is reaching in one cycle, but there is a problem: the response is very fast, but unlike in buck converter where the time optimal control also achieve minimum voltage undershoot, but here the voltage undershoot drastically increases because your on time is large.

(Refer Slide Time: 21:27)

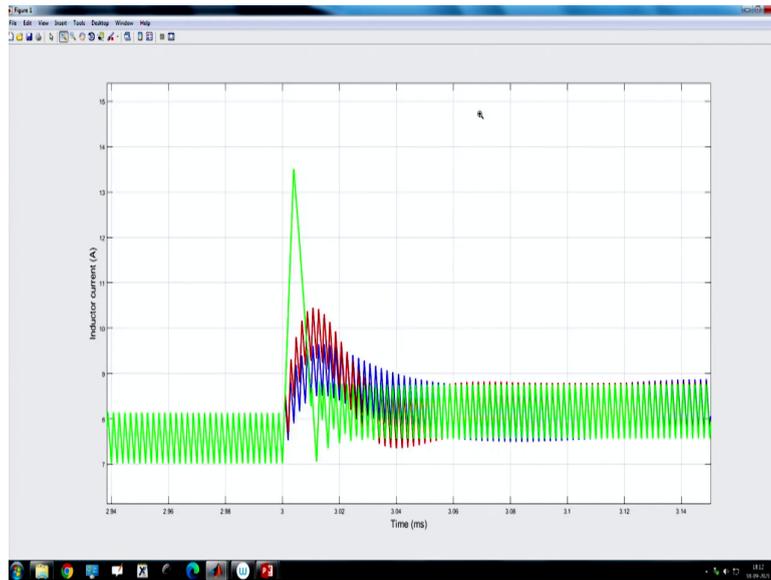


And if you go to the current waveform, you will find.

(Refer Slide Time: 21:29)

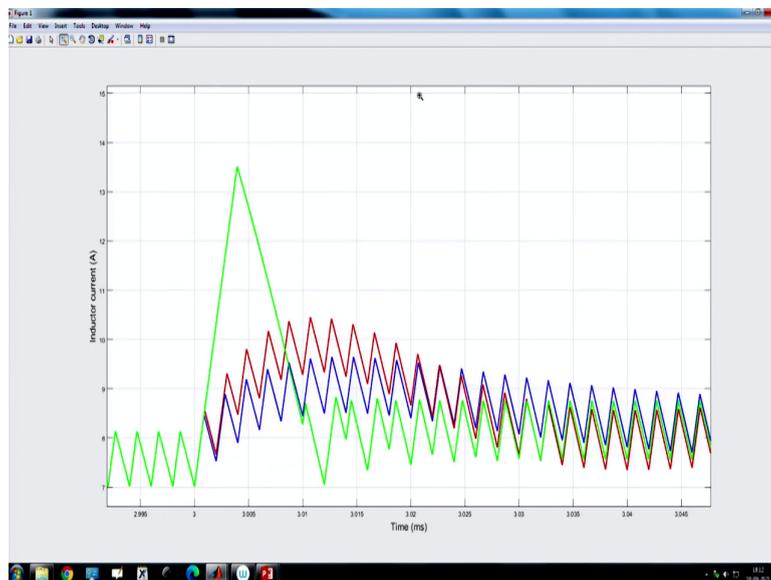


(Refer Slide Time: 21:30)



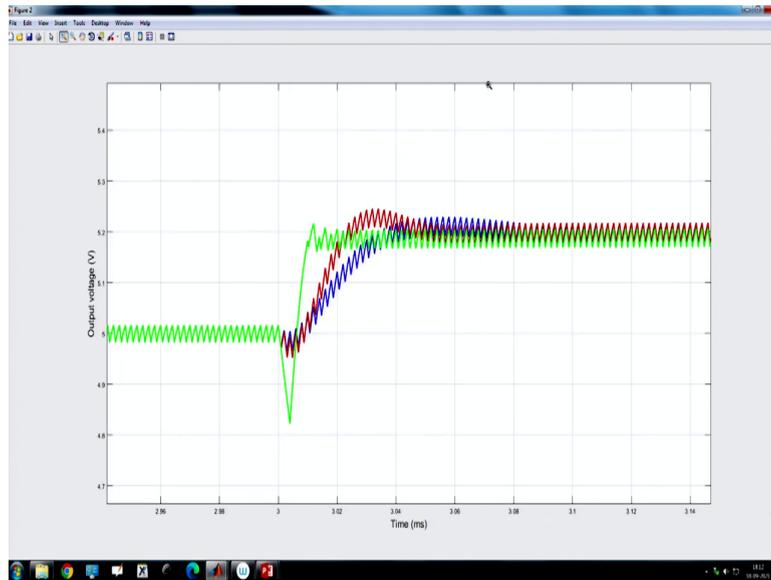
You have a high on state on time.

(Refer Slide Time: 21:32)



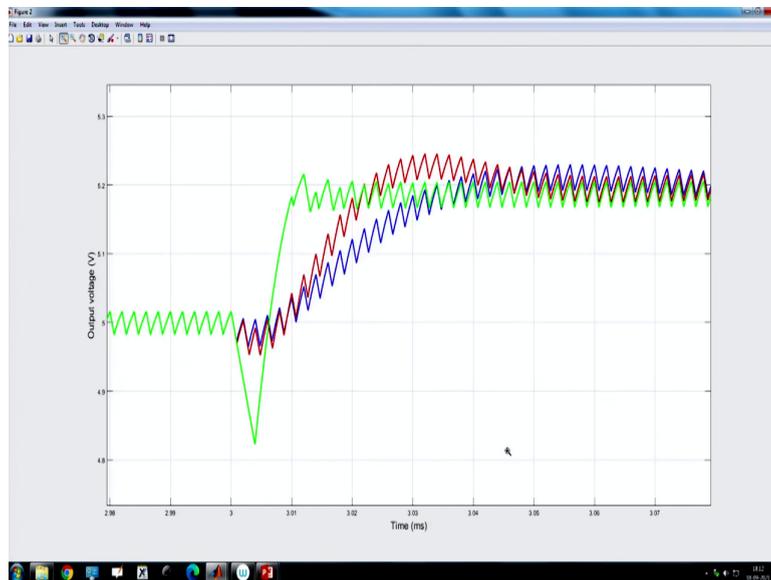
Because the current is increasing and as a result of a longer on time, your voltage will simply fall.

(Refer Slide Time: 21:46)



So, while it achieve very fast transient response, you know it is in reaching in five switching cycles.

(Refer Slide Time: 21:48)



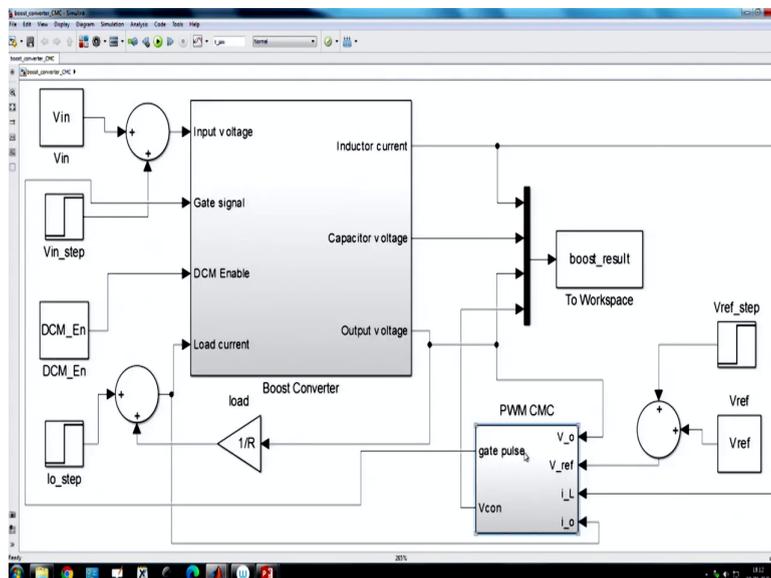
Current mode is faster than voltage mode and current mode, but it comes with this price.

(Refer Slide Time: 21:52)

```
31 % PM=90-theta; w_c=p*w_rhp;
32 % theta=(90-PM); theta_rad=deg2rad(theta);
33 % K_c=(p*w_rhp)/(K_g); w_cp=w_rhp; k_n=0;
34 % num_con=K_c*den_c;
35 % den_con=[1/w_cp 1 0];
36 % Ge=tf(num_con,den_con);
37
38 %% Large-signal tuning of PI controller
39 %
40 K_cp=1; k_n=Vref/Vin;
41 K_vp=15.5;
42 K_vi=40000;
43
44 num_con=[K_vp K_vi];
45 den_con=[1 0];
46 Ge=tf(num_con,den_con);
47
48
49 %% Transient parameters and transient response
50 L_sim=5e-3; L_step=3e-3;
51
52 L_max=15; L_min=-10;
53
54 Delta_Ic=0; Delta_Vin=0; Delta_Vref=0;
```

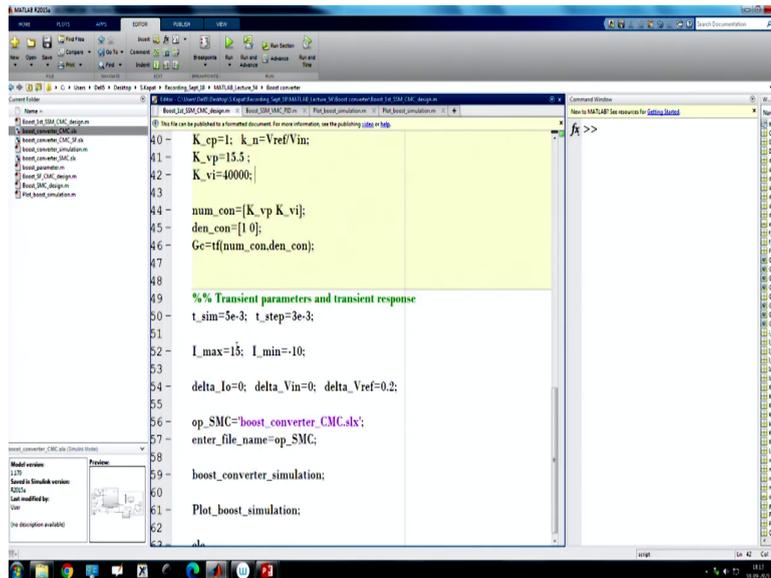
But, we can always set a current difference.

(Refer Slide Time: 21:56)



So, if you go to current mode control, you know if you go to our current mode control.

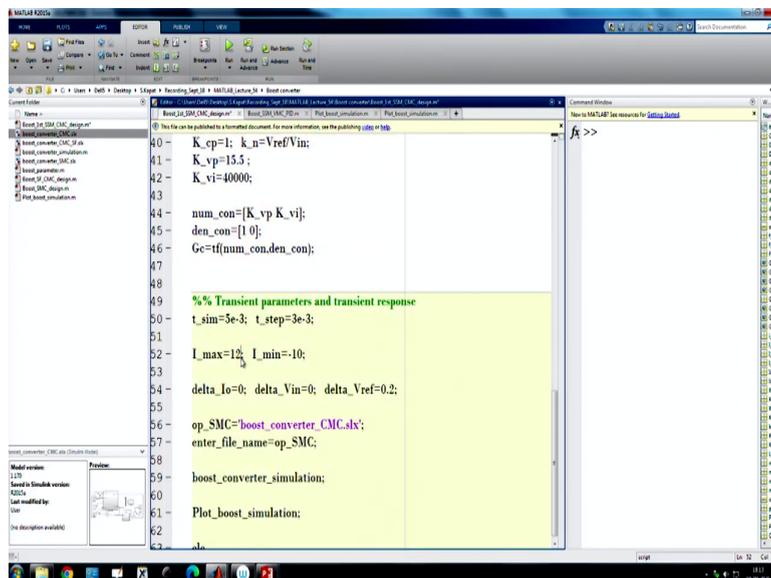
(Refer Slide Time: 22:20)



```
40 - K_cp=1; k_n=Vref/Vin;
41 - K_vp=15.5;
42 - K_vi=40000;
43
44 - num_con=[K_vp K_vi];
45 - den_con=[1 0];
46 - Ge=tf(num_con,den_con);
47
48
49 %% Transient parameters and transient response
50 - t_sim=5e-3; t_step=3e-3;
51
52 - I_max=15; I_min=-10;
53
54 - delta_Io=0; delta_Vin=0; delta_Vref=0.2;
55
56 - op_SMC='boost_converter_CMC.slx';
57 - enter_file_name='op_SMC';
58
59 - boost_converter_simulation;
60
61 - Plot_boost_simulation;
```

So, let me check I max. So, here it is 15 ampere. Let us say we make it 12 ampere.

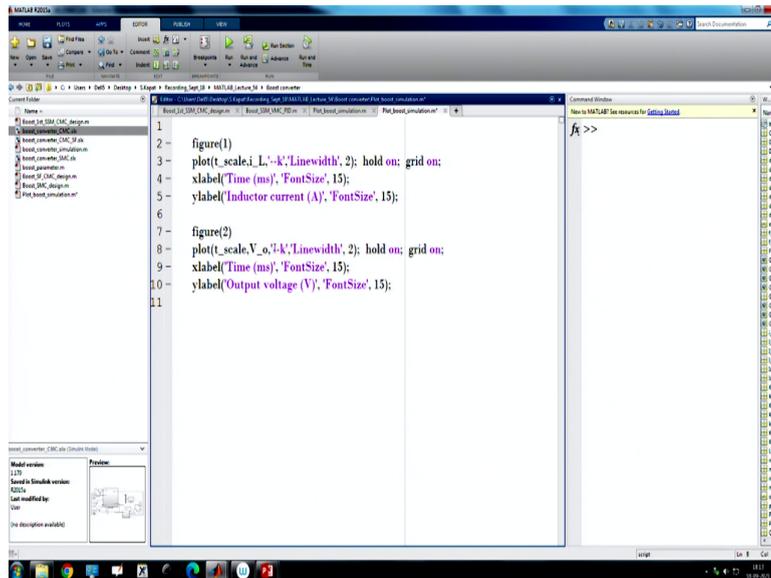
(Refer Slide Time: 22:22)



```
40 - K_cp=1; k_n=Vref/Vin;
41 - K_vp=15.5;
42 - K_vi=40000;
43
44 - num_con=[K_vp K_vi];
45 - den_con=[1 0];
46 - Ge=tf(num_con,den_con);
47
48
49 %% Transient parameters and transient response
50 - t_sim=5e-3; t_step=3e-3;
51
52 - I_max=12; I_min=-10;
53
54 - delta_Io=0; delta_Vin=0; delta_Vref=0.2;
55
56 - op_SMC='boost_converter_CMC.slx';
57 - enter_file_name='op_SMC';
58
59 - boost_converter_simulation;
60
61 - Plot_boost_simulation;
```

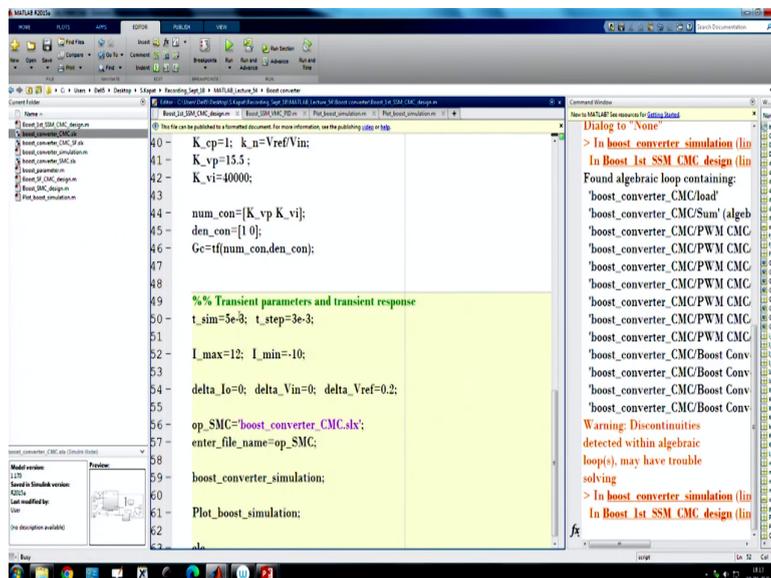
So, we are setting a current limit. We rerun it; we want to rerun it and now we are using black color.

(Refer Slide Time: 22:30)



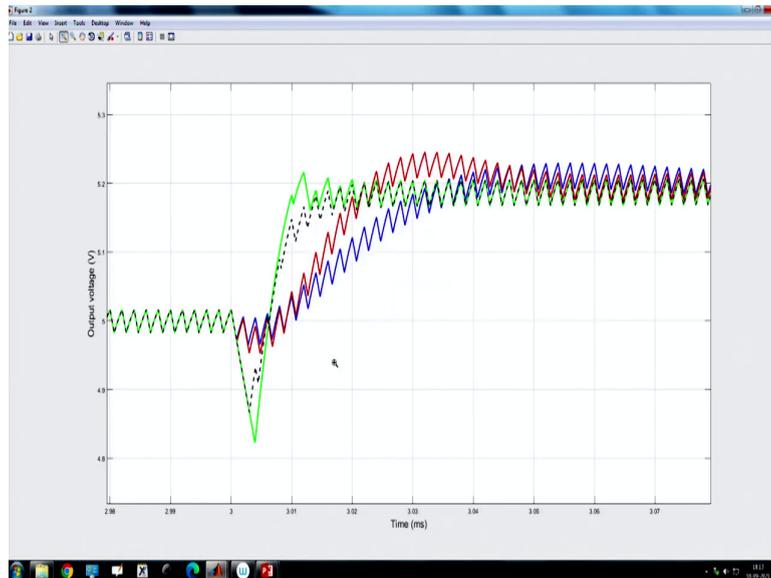
Let us say black dotted color dash color. So, if we want to set a current limit and run it.

(Refer Slide Time: 22:39)



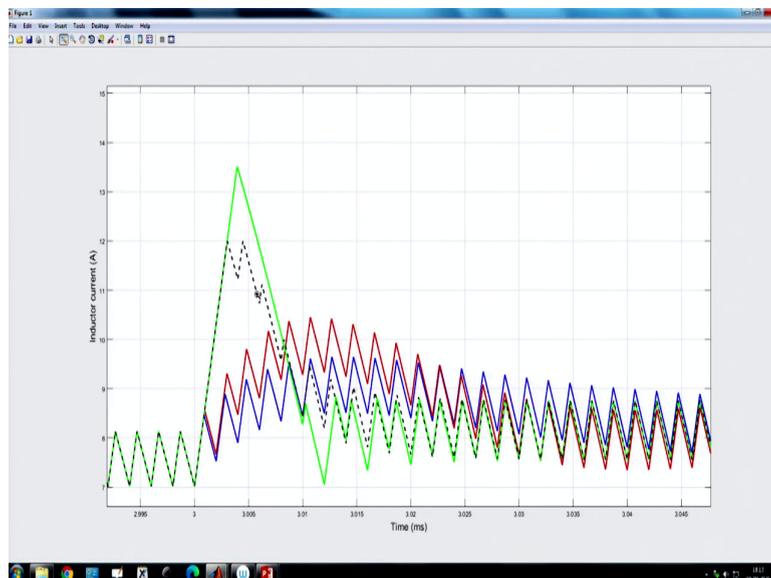
And let us see what happens.

(Refer Slide Time: 22:42)



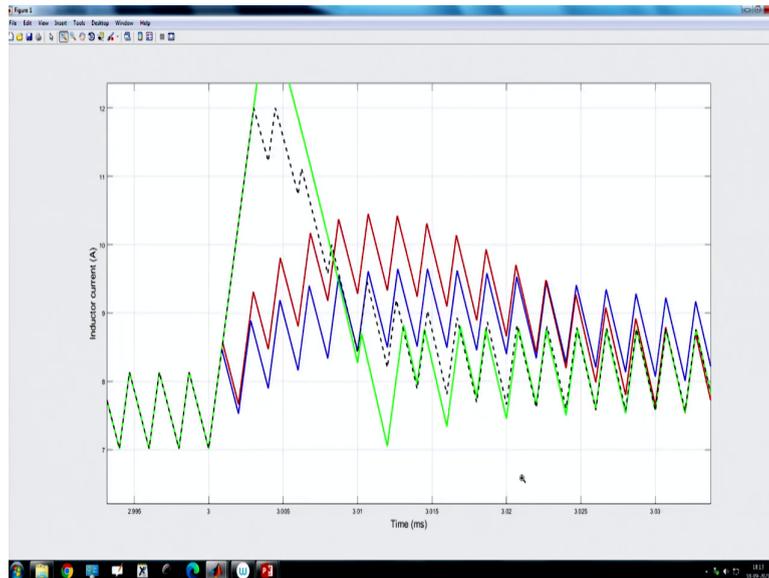
Because now we are putting a current limit.

(Refer Slide Time: 22:45)



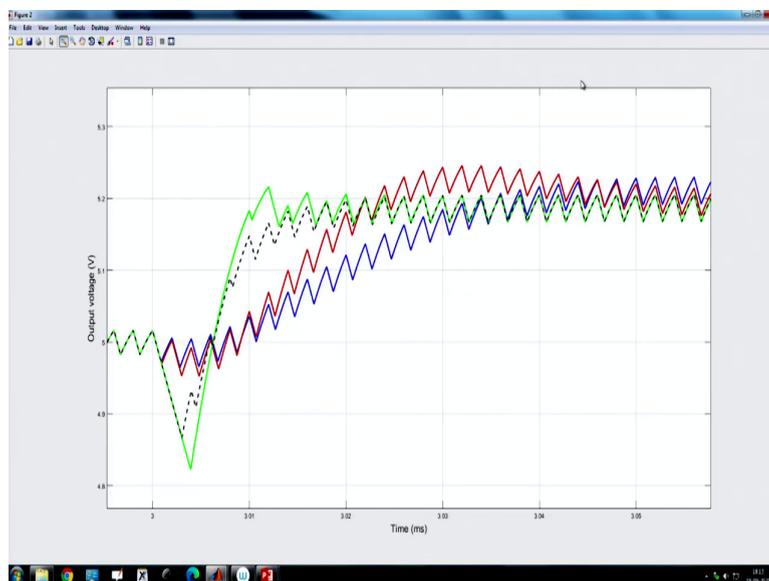
So, once you set a current limit, you see.

(Refer Slide Time: 22:48)



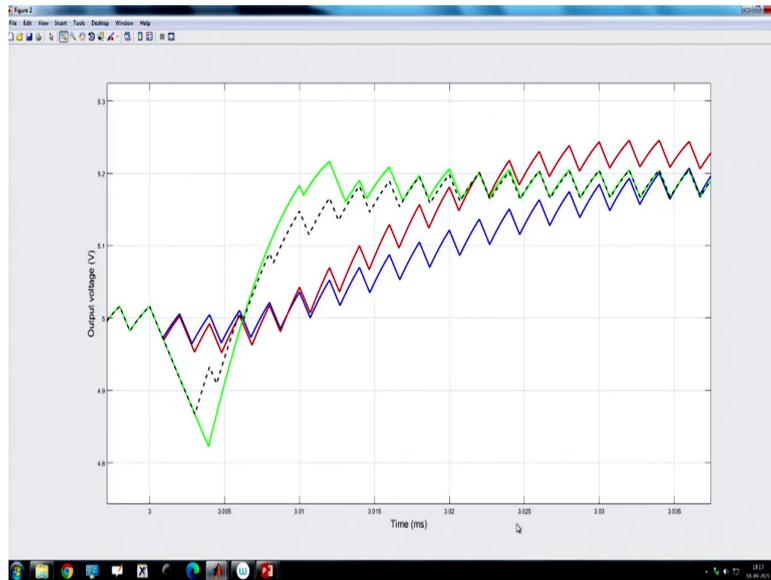
The inductor current is reduced to 12 ampere because we can set according to our current rating.

(Refer Slide Time: 22:56)



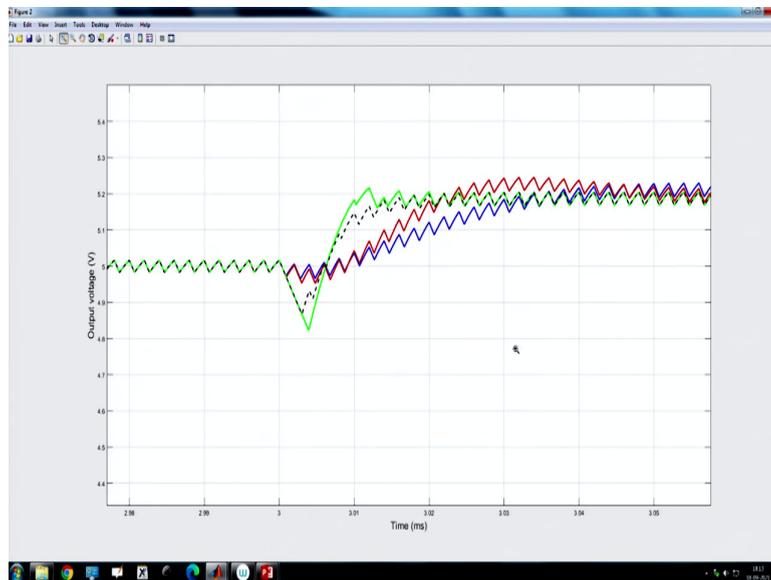
And the response is still much faster than both these control techniques.

(Refer Slide Time: 23:00)



Because it is taking a few more cycles. Earlier it was just taking 5 cycle, but now it is taking you know 10 cycle.

(Refer Slide Time: 23:07)

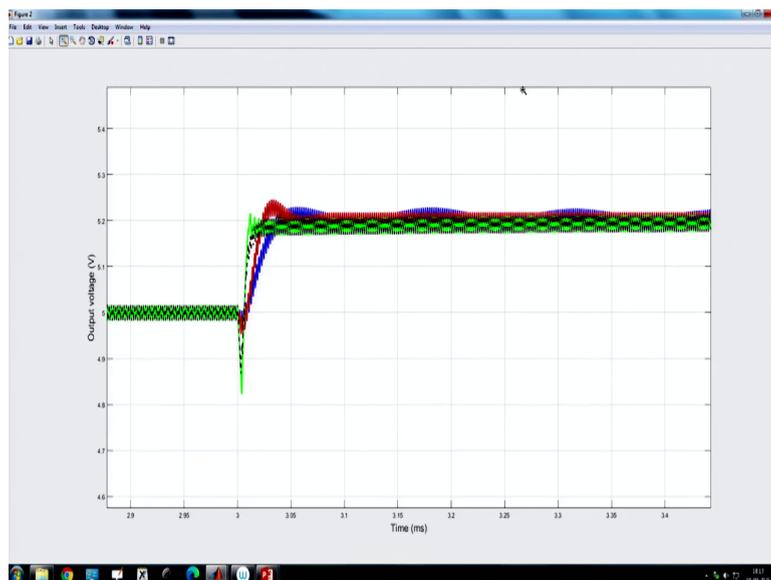


But, it is faster compared to our you know traditional current mode and voltage mode control.

(Refer Slide Time: 23:10)

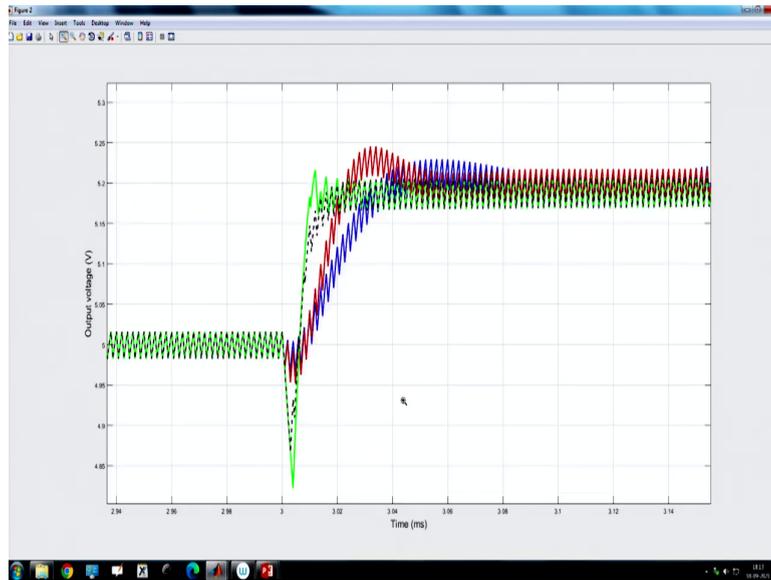


(Refer Slide Time: 23:12)



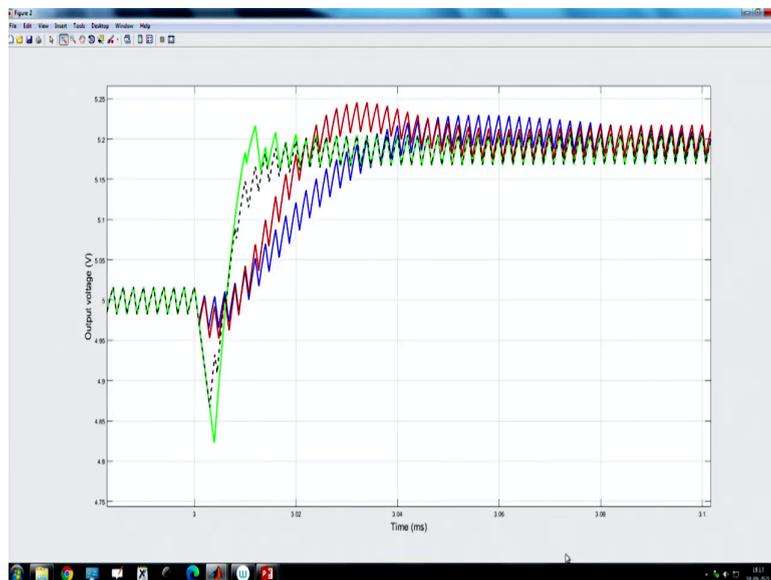
Because you can see.

(Refer Slide Time: 23:13)



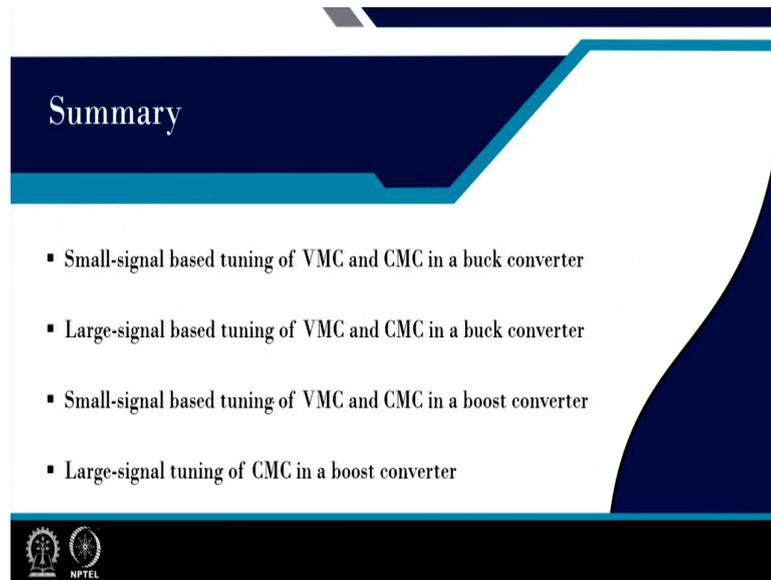
So, this blue one black one is reaching much faster, right?

(Refer Slide Time: 23:16)



So; that means it is much better than and will show you in the subsequent lecture with the experimental result we have; that means, with optimal tuning using linear control, then with optimal tuning sorry. We have compared the optimal tuning, which is a linear controller tuning. Then we also set our current limit and we want to show that all these responses are much faster than what we can achieve using small-signal based tuning.

(Refer Slide Time: 26:44)



Summary

- Small-signal based tuning of VMC and CMC in a buck converter
- Large-signal based tuning of VMC and CMC in a buck converter
- Small-signal based tuning of VMC and CMC in a boost converter
- Large-signal tuning of CMC in a boost converter

NPTEL

So, in summary, we have discussed small-signal based tuning of voltage mode and current mode control in a buck converter. We have discussed large-signal based tuning of voltage and current mode control of buck converter. We have discussed the large-signal and small-signal; we have discussed small-signal based tuning of voltage mode and current mode control in a boost converter.

And we have compared the large-signal based tuning of current mode control in a boost converter with and without current limit and we saw the large-signal based tuning can significantly improve the transient performance of DC-DC converter compared to small-signal model. So, with this I want to finish it here.

Thank you very much.