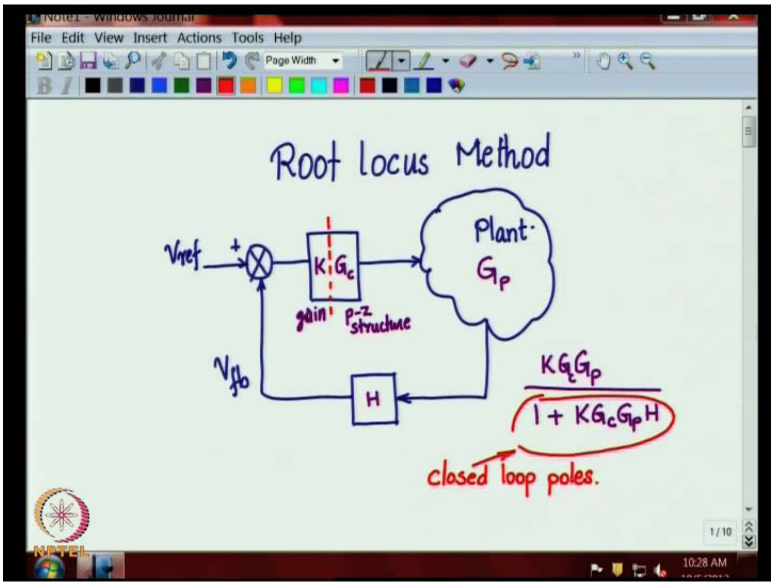
Switch Mode Power Conversion
Prof. L. Umanand
Department of Electronics System Engineering
Indian Institute of Science, Bangalore

Lecture - 33
Controller Design - II

Good day to all of you. Today we shall continue the design of the controller for dc dc converters. In the last class we tried the design with the trial and error approach; also called the Ziegler Nichols method; and we saw how we go about tuning the PID parameters p i parameters to match a performance specification. In this class today, we shall discuss a more formal approach called the root locus technique. What has been popular is both the root locus technique and the method by the bode plots; bode plots have also been very, very popular in the design of the controllers.

However, we shall be adapting only the root locus technique; the reason being that in many converters you have zeros on the right half of the s plane like the boost converter for example, which has zero on the right half of the S plane in the small signal model. Because of that the bode plot will not work, the bode plot will work for non-minimum phase systems only therefore, we shall take the root locus approach, which is a more general procedure whether it be for non-minimum phase systems or any other system as long as you have linear time invariant systems.

(Refer Slide Time: 02:18)

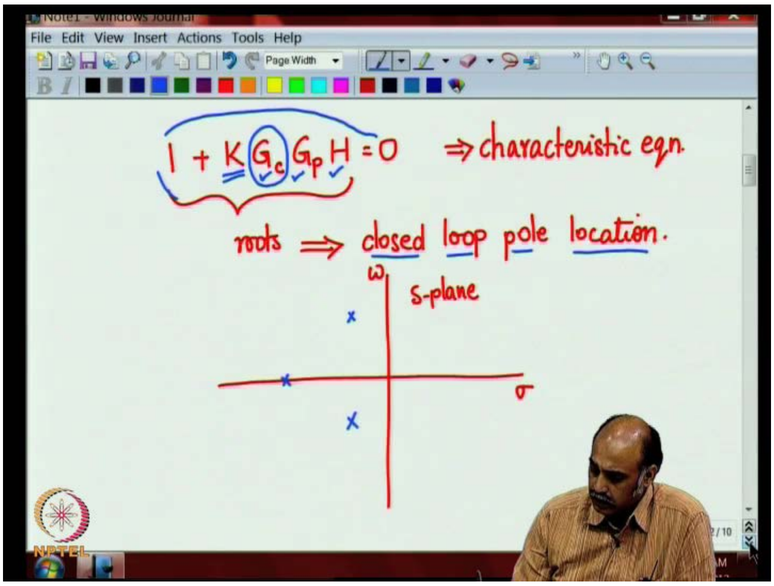


So, that is why we are going to focus on the root locus method. So, more generic approach, it has not been so popular before the advent of the computers, because it is very computation intensive and then doing hand calculation is of very problematic situation. Therefore, the bode plot approach was more preferred. However, after the advent of the computers, the computation and all other bull work was shifted over to the computers and finding the root loci was not a very difficult task. And therefore, root locus static gaining popularity and today it is a very popular method to design controllers.

Now if you take a general control block diagram as I am indicating here. Let us take, let me take a general controller, which has a reference a feedback, a controller and a plant feedback with feedback sensing signal conditioning and this fashion; let say this is our general control system block diagram. Now let us say the controller has a transfer function, which is split into two parts. Now, I will split the controller into two parts; one part is just simply with gain and I will call it as k; the other part using the structure of the controller, whole zero structure of the controller and let me call that as G c; this is the gain part and this is the pole zero structure. Likewise the plant has a transfer function and ?let us call that one as N p G p.

The feedback portion has a transfer function, and let us call that one as H. Now, the closed loop transfer function is given by K G c G p by 1 plus K G c G p H. So, this is the closed loop transfer function. And after our controller design, what we are interested is in the location of roots of this; this basically gives the poles of the closed loop transfer function. The roots of this will give the location of the closed loop poles.

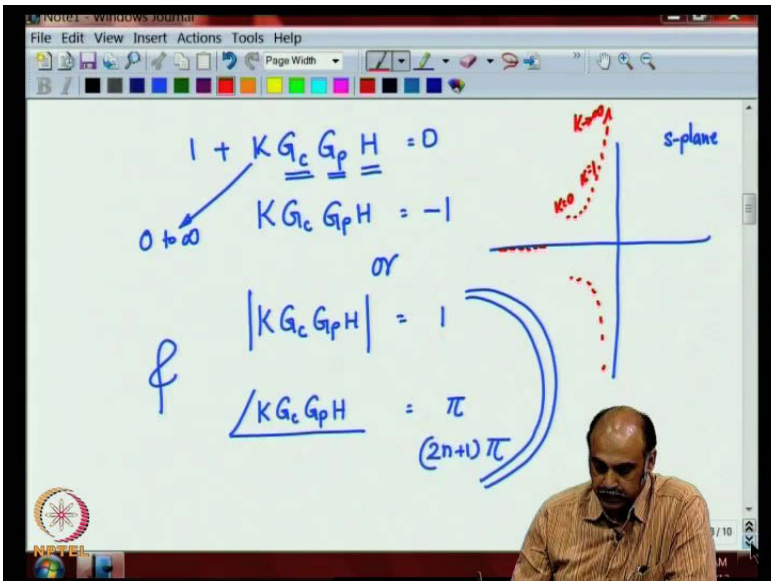
(Refer Slide Time: 06:58)



Therefore we say that 1 plus K G c, plant transfer function H is characteristics equation. The roots of this will give you the closed loop pole location. So, the idea here is that let us say that you are given the flexibility to choose the closed loop pole location. This is the S plane, we choose the closed loop pole locations on the left of S plane; stable according to a particular performance criteria, look at the transfer step response and then match it to the performance requirements.

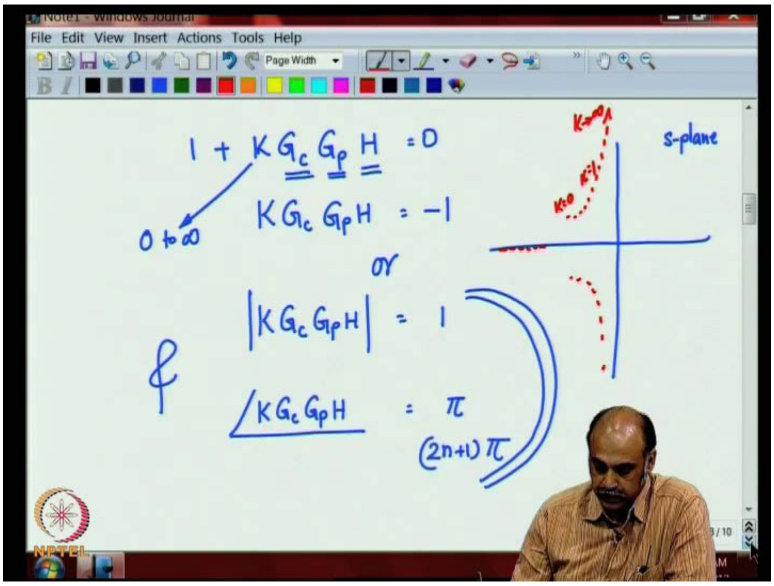
So, the closed loop pole location is decided by the designer. And then that is used to calculate the unknowns in this equation; the unknown in this equation is this; this is known, the structure of the controller is known, we are, the designer is giving the structure location of poles and zeros. The model of the plant is known, the model of the feedback portion of this activity is known, the only unknown would be k; and if we know the roots which is the closed loop pole location, which let us say the designer proposes, and then says let me place the closed loop poles at these, these, these points; and then use those to calculate and find out K, (( )) that is the principle you, it is basically like working back, you decide this should be the final ultimate closed loop pole location. And then find out what is the value of gain K, which will meet the spec; if it does not meet your performance specification, change this controller structure, repeat the process and then try to find K. So, that is the principle.

(Refer Slide Time: 10:18)



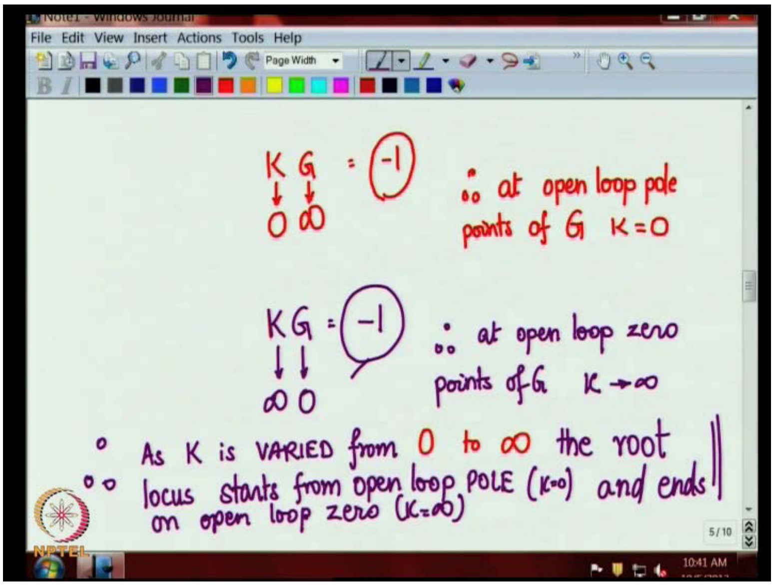
Now, if you look at this 1 plus K G c controller plant H equal to 0, what are the condition that satisfies this equation, which will give the roots K G c G p H should be equal to minus 1 or the amplitude magnitude of K G c G p H should be equal to 1 and the angle K G c G p H should be pi 180 degrees or in the more general terms, we could have 2 n plus 1 pi. Now, these two constraints, if they are matched, then all those points in the S plane for a given G p, for a given H, for a given G c, varying K from, varying K from 0 to infinity. If we plot all the points that satisfy these two constraints, then we probably we will get sequence of points so on, something like that (( )) many, many such point. Now the locus of all these points is called the root locus; this may be at K is equal to 0, K is equal to 1 so on upto K tending to infinity per each. So, that is basically the principle.

(Refer Slide Time: 13:00)



Now, let us look at, let us look at just a simple closed loop system KG by 1 plus KG, where H is equal to 1, G c is equal to 1; let me take such condition 1 plus KG is equal to 0. Now this has or rewriting KG is equal to minus 1, K into numerator polynomial by denominator polynomial; these are all functions in S is equal to minus 1. Now, look at the numerator polynomial of G of G, this is 0s of G, these are poles of G. So, at values of S, at values of S, which are tending to the poles of G, you will find that this portion tends to 0 and the whole G, G tends to an infinite value. If G tend to an infinite value, K should tend to 0.

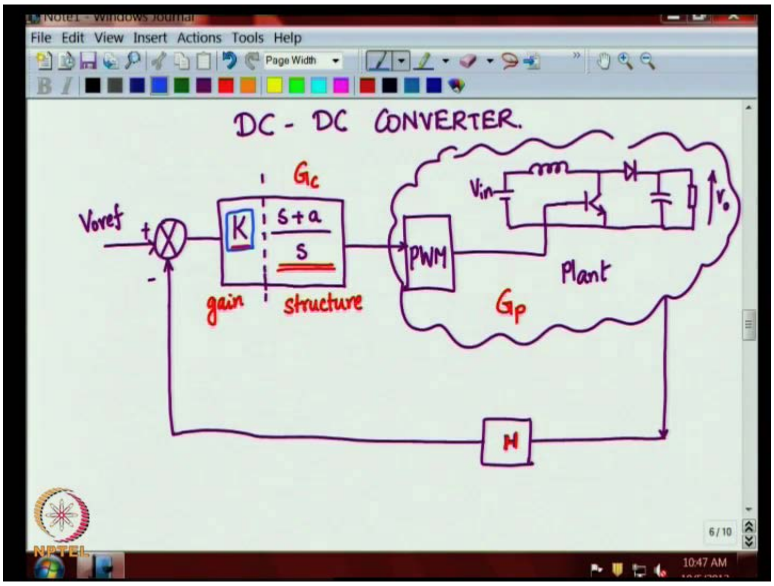
(Refer Slide Time: 16:23)



So, if I write this as I will put it in the next page K G minus 1; if G tends to infinity, then K should tend to 0 to satisfy this condition. So, at open loop pole points at open loop pole points of G K is 0. Now to the same KG equals minus 1; at p, at p open loop 0s, so as at values of S that are matching with the 0s, G becomes 0, G becomes 0. So, K has to go towards infinity to satisfy these conditions; therefore at open loop zero points of G, K is an infinite value. So, therefore, we can say a general statement valid for root locus as K is varied from 0 to infinity. The root locus starts from open loop pole K equal to 0 and ends on open loop 0, K infinite. So, this is a general statement that we can make for root locus method.

Anyway, you need not worry, this is concept that if you keep in mind when you are trying to adjust the placed poles, the computer does the job of calculating all the roots solving these equations, specifically this equation, and walking to you the root loci of the system. So, if there, if it is a third order system, if the third order system, then you will have three root loci; if it is a second order system, you will have two root loci; first order system one root loci. So, for every order there is a pole, open loop pole, and then if you have an open loop pole, then a root loci starts from the open loop pole, and then ends at open loop zero. If the 0 is finite, 0 is not there, then it will end at infinity, it will asymptotically tend to an infinite a point at infinity. Thinking that equivalently saying that zero is at infinity.

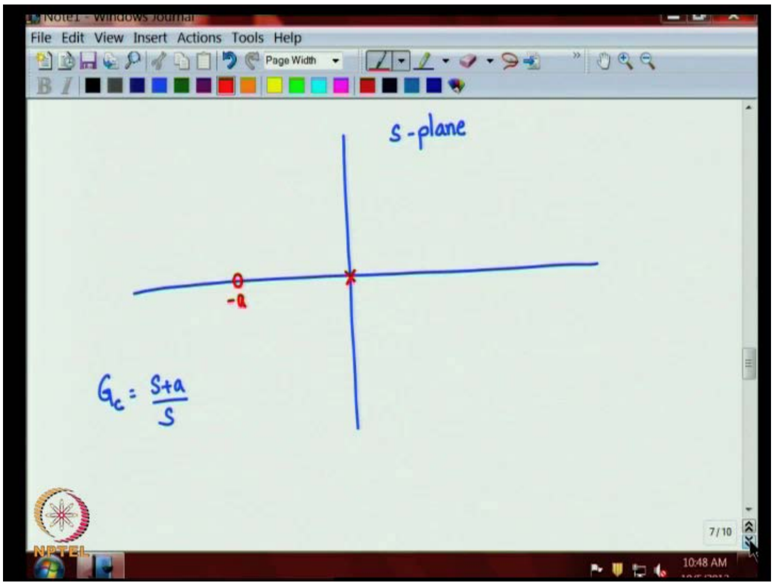
(Refer Slide Time: 21:26)



So, now, coming back to our problem, which is the dc dc converter, dc dc converter, let us have a reference, v naught reference, a comparator goes through a pi controller. And this goes let us say to a boost converter, this time we will take a boost converter, yesterday we took a buck converter probably to keep let us say you have a boost converter like this is v naught v in. So, let us have a pulse width modulator, which will drive this switch. This is your plant or we could include the pwm also into the plant. Now if you could probably do that, include the pwm also into the plant; now this controller, pi controller can be written as you have an integrator S, you have a 0 to account for the proportional part, and then you have one consolidated k. So, this is the split for the pi controller, and the output of it passes through H. So, this is how the system looks like, and importantly want to bring your attention to this pi controller transfer function.

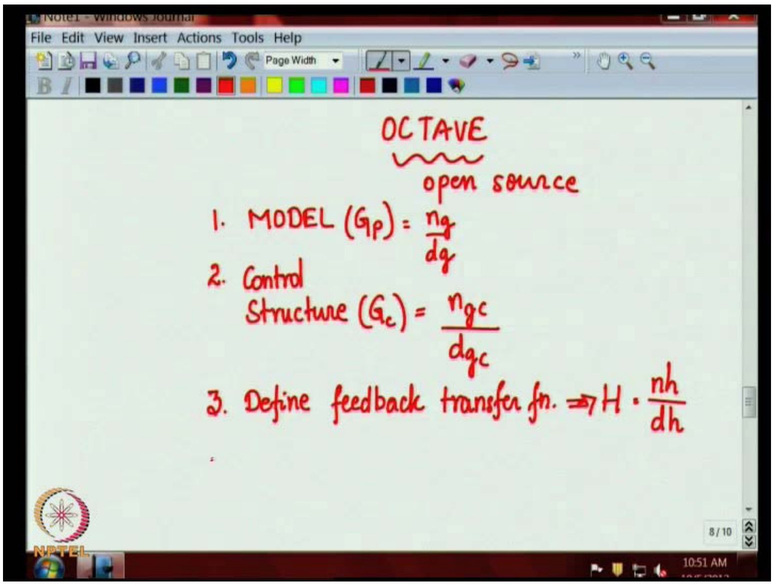
So, you have the i portion and the p portion, you have a consolidated gain, this is positioned by the designer is the control structure, how many zeros and poles and where they are located; and this is the gain. So, this would be our G c, this would be G p and you have the H. Now, let us apply the root locus method to design what we have to design this. So, the moment we decide that it is a pi structure, you have 1 0 and 1 pole to place on the S plane.

(Refer Slide Time: 26:14)



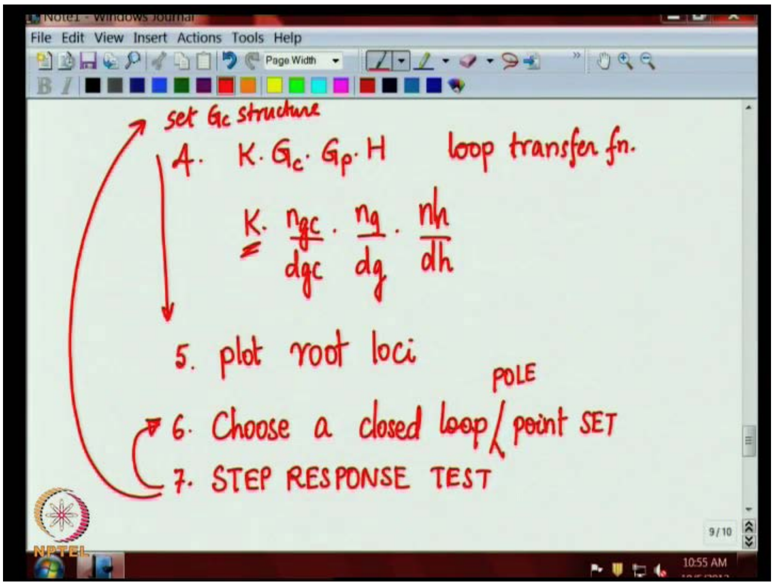
So, if you take the S plane the controller G c is K, K is separated out S plus a by S. At x is equal to 0, you have a pole and at x equal to minus a you have a 0. So, this is the positions, positioning of opponent 0 for the pi controller and K is the variable parameter that is varied for the whole locus.

(Refer Slide Time: 27:38)



So, the steps is we will go to the computer, we will open an m file, a text editor, you can either perform this iterative action by running a program either in Matlab or octave. Octave is open source, compatible with Matlab language; so I will be using here octave for running the scripts. So, the scripts should be partitioned in the following manner. First we need the model, after we had the model of the plant that is G p, which will give you a numerator polynomial and denominator polynomial. Decide on the control structure, which will give you the numerator polynomial numerator polynomial of the controller and the denominator polynomial of the controller, and then define feedback transfer function of the sensor, which will give you H a numerator polynomial of the H and denominator polynomial of the H any two split it that way.

(Refer Slide Time: 29:59)

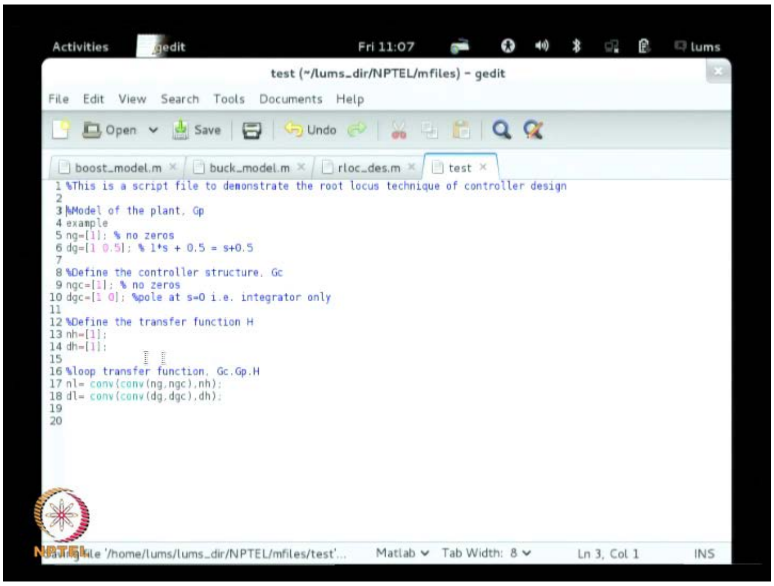


After that, we try to obtain K G c G p H, this is the loop transfer function. So, loop transfer function is nothing but K into n g c by d g c in terms of polynomials n g by d g into n h by d h. So, numerator and denominator of polynomials of G c and numerator and denominator of polynomials of G p and numerator and denominator of polynomials of H. Then using K has a parameter, varying K from 0 to infinity, we try to obtain root the locus.

So, the root locus, then we try to a plot, the root locus or loci of the above transfer function, then we choose a closed loop point, choosed a closed loop point set. Why I means set is that if it is a first order system, we are choosing one closed point; if it is a second order system, you need to have a two closed loop pole points, closed loop sorry pole points or if you choose on any one locus of the root loci or the corresponding K, the other points are automatically chosen on all the other root loci.

Then do a step response test, check the step response, whether it is satisfactory or not; if it is satisfactory, then you can stop here and that could be the value of K; if it is not satisfactory, again you can go and choose this (( )). Even after many iteration, you do not get a satisfactory this one; you can go back here to the point 3 to the set G c structure. And then follow the procedure again. So, this way, you try to obtain the optimal, not optimal reasonably good specify performance matching, and that value of K which you have chosen would be the one that you will plug in into the simulation.

(Refer Slide Time: 34:27)

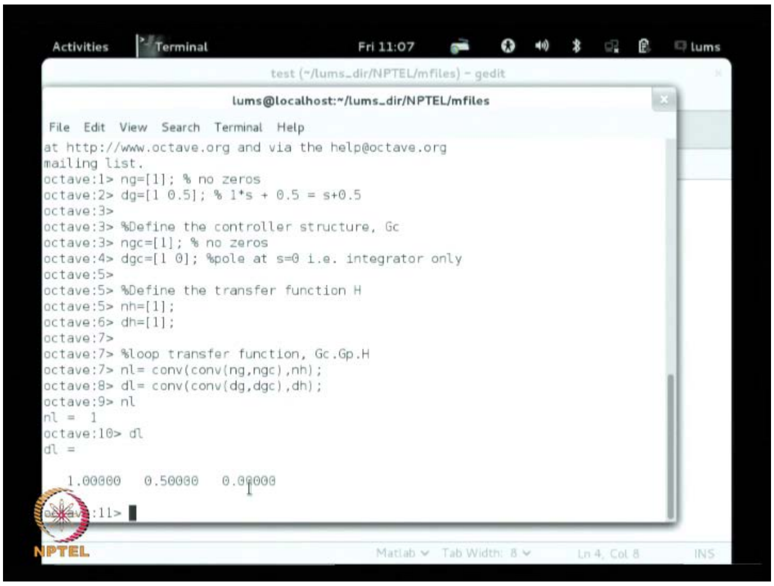


So, this step we will try to follow in Matlab or octave and try to see how it works. Now let me go switch over to the computer, which will do the simulation. So, I have here a blank screen, it is a text editor; and let me go through the process of writing the script. So, this can be a m file and we shall seen this is a script file to demonstrate the root locus technique of controller design. Now the model, model of the plant; now take an example model, this is not the example model of a Dc-Dc converter, but let us say some arbitrary example.

Now, let us defining equals 1; this means that we are having the numerator polynomial, we have only S to the power of 0, no 0s. Now dg - the denominator polynomial let us say if we design as 1 0.5, then it implies the denominator polynomial having S 1 into S plus 0.5 into the S to the power of 0 that is S plus 0.5. So, each coefficient rise to the power of so this is 0.5 S to the power of 0 1 into the S to the power of 1 and so on, you can add anymore.

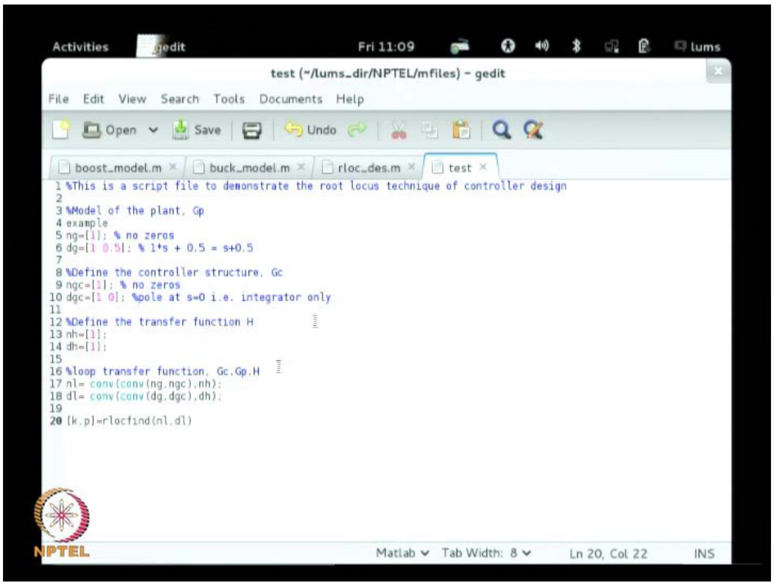
Let us take such a simple plan first order plan and then see how we go about doing. So, after that we define the controller structure. So n G c, numerator polynomial of the controller, let us say is 1, again no 0, and denominator polynomial is 1 0, which means you have a pole at S equal to 0; that is integrator only. Then define the transfer function H is nh, right now we will keep it simple 1 and 1, no poles and no zeros, just one; let say the loop transfer function, which is G c into G p into H. So, this is G p, this is G c. So, the numerator polynomial, the loop transfer function is the convolution of, convolution of n G c and nh; the denominator polynomial of the loop transfer function is convolution multiplication of the polynomials, straight forward polynomial multiplication and we get, fine. So, this is the if would like to see, what you have got till this point this copied, let us first save this document, we save it in some place, I will call it as test.

(Refer Slide Time: 40:18)



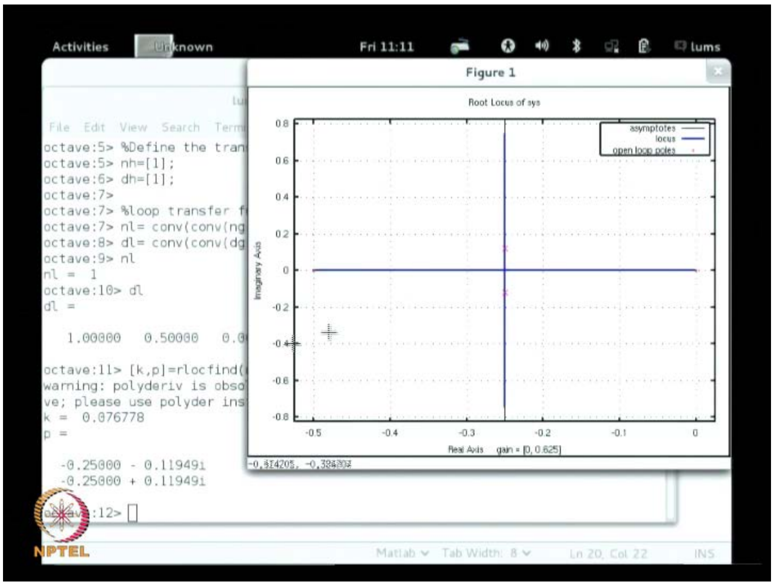
You can copy this word to octave and paste that in octave to see what you get. So, this is execute the nl is 1, dl is… So, this is s to the power of 0, s to the power of 1, s to the power of 2. So, the closed loop transfer function is a second order one order being contributed by ng dg, another order contributed by G c, H does not contributed by any order because no poles and zeros. So, you have a second order system.

(Refer Slide Time: 41:26)



Now, let us look at the closed loop let us look at the root locus. So, let us say you have keep j K and p equals function r locfind the… So, if you give r locfind the loop transfer function numerator polynomial and the denominator polynomial may plot the root locus and allow give to click up point on that. So let us say I will save that, and then I will copy and then go back to octave.

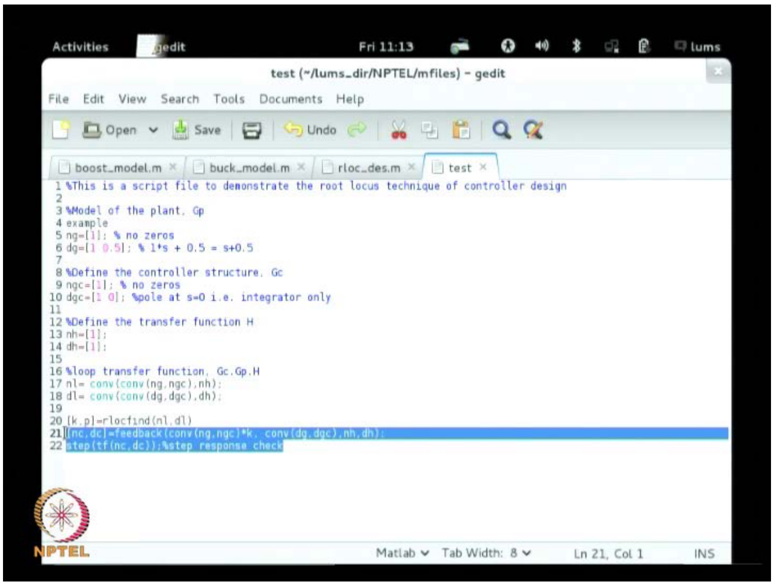
(Refer Slide Time: 42:16)



And let us say we paste that. So, it will present to you the root locus plot. So, there are, this is the real axis of the S plane, and this omega axis are the imaginary axis of the S plane. The blue lines are the root loci. There is one pole here, let me see if I can maximize this S; observe the one pole here, this the pole that has been contributed by the designer placing the controller at S equal to 0, there is a placed an integrator used an integrator structure. So, there is 1, S is equal to 0 open loop pole G c G p. There is another pole here at 0.5, S is equal to minus 0.5, because of the plant.

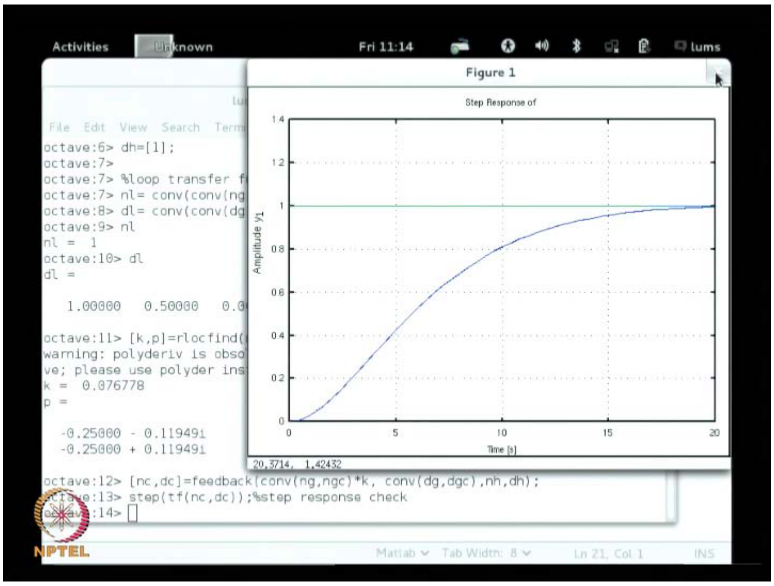
Now, these two are the root loci coming in here. Now you could choose, you could choose a point on this; it will indicate these are the points chosen, the closed loop point chosen. And the value of K at these points is 0.076778, and the closed loop pole locations are these two. So, that it is what you could indicate. And you could use this value of K let us say, and get the closed loop system with K, which is K G c G p H by 1 plus K G c G p.

(Refer Slide Time: 45:03)



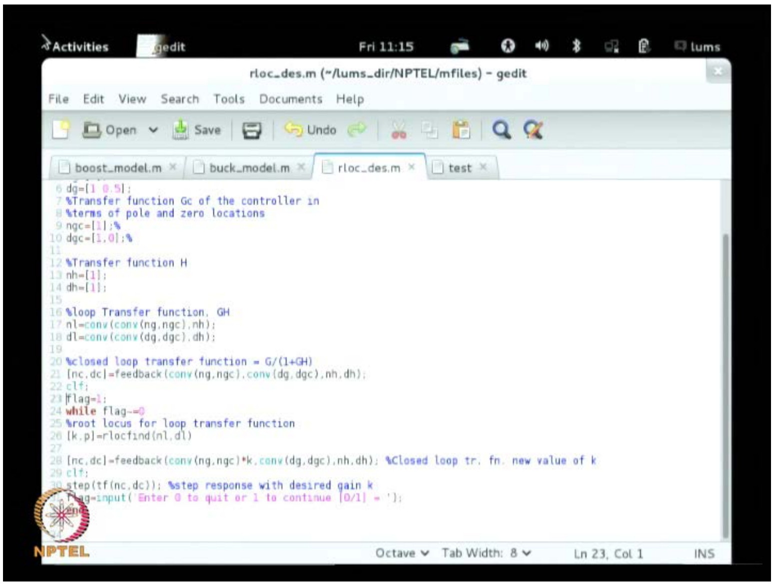
Now further to get the closed loop system we so let us say the closed loop numerator polynomial denominator polynomial is given by feedback the into k. So, K numerator into ng nc; that is gc numerator polynomial of gc and gp, then multiplication of the denominator polynomials, then specify the feedback polynomial. So, that would give you the closed loop transfer function. After you get the closed loop transfer function, you could do a step response check using step convert it into a system nc dc which is the… So, this is a step response check. So, if I take these two copy and go to octave, an octave let me paste them.

(Refer Slide Time: 46:42)



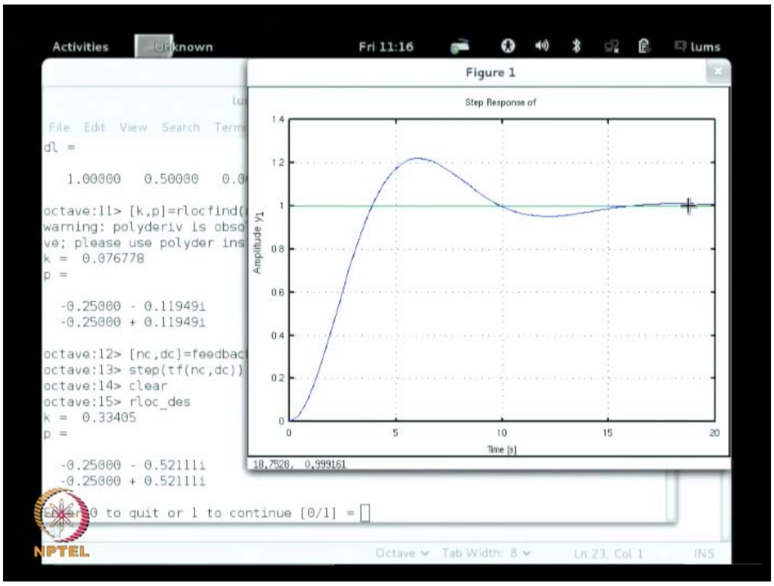
So, this will make the closed loop transfer function, then the step response of the closed loop transfer function. When you execute it, you will get the figure, which is the step response, you have with respect to time and the amplitude for a unit step, this is the response. Now this can be made iterative. So, let us make it iterative, such that you can keep repeating.

(Refer Slide Time: 47:34)



So what we do? We try to put into a loop. So, this is same function, what I, the portion of root locus, then the portion of one with a feedback and the step calculating step response can be put within a while loop. And you can ask query 0 to quit or 1 to continue. So, this would tell us tell me for now remove some of these things. So, I have saved into another file called r locus design.

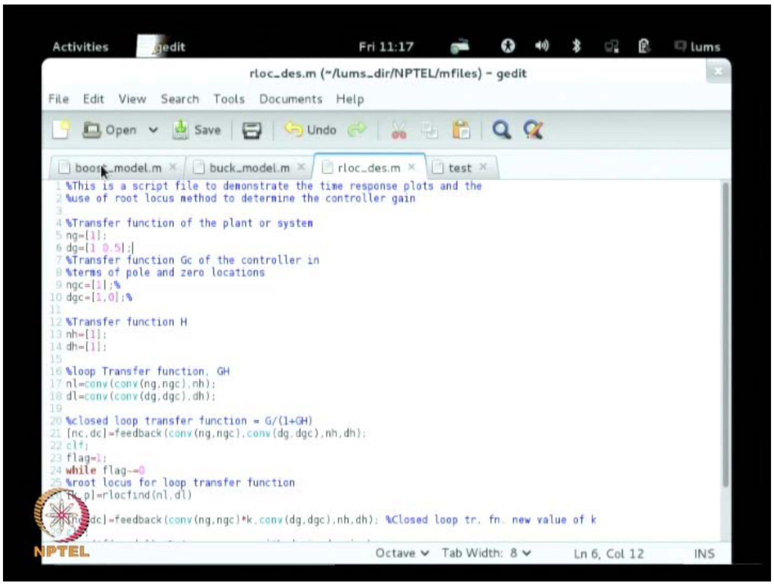
(Refer Slide Time: 48:38)



Now, they are let me called r locus design. So, this is the first call; it will tell you that this is the root locus, let me choose point; then after the choice of the point, the step closed loop function is calculated, the step response is plotted using a step response of this nature. Now you could take let us continue, you will presented again with the root locus.

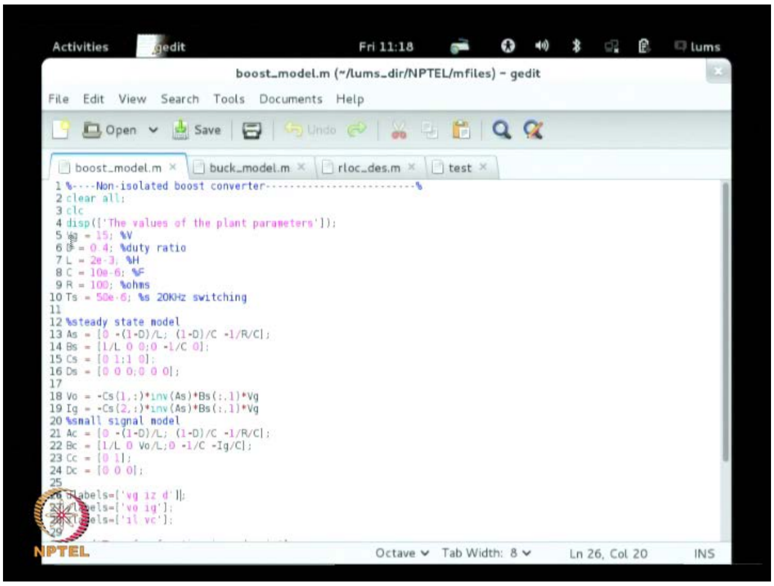
And let say we choose some other closed loop point, we choose some other closed loop point and you will see that the K value is lesser, and see K value in the previous iteration of the point is 0.334. Now in the present iteration, it is 0.07 much lesser and the over shoots have been reduced. So, on you can keep iterating choosing different closed loop points still you get the response according to your performance specification. So, that is the, then you can put zero and come out.

(Refer Slide Time: 50:28)



Now, the thing is that let us try it on dc dc converter; for that, we need to modify a bit this root locus design point; this should become the model of the converter, model of the converter that you would want to use. Now let us say we want use the model of the boost converter for now. So, I have prepared here the model of the boost converter in this file here.

(Refer Slide Time: 50:54)

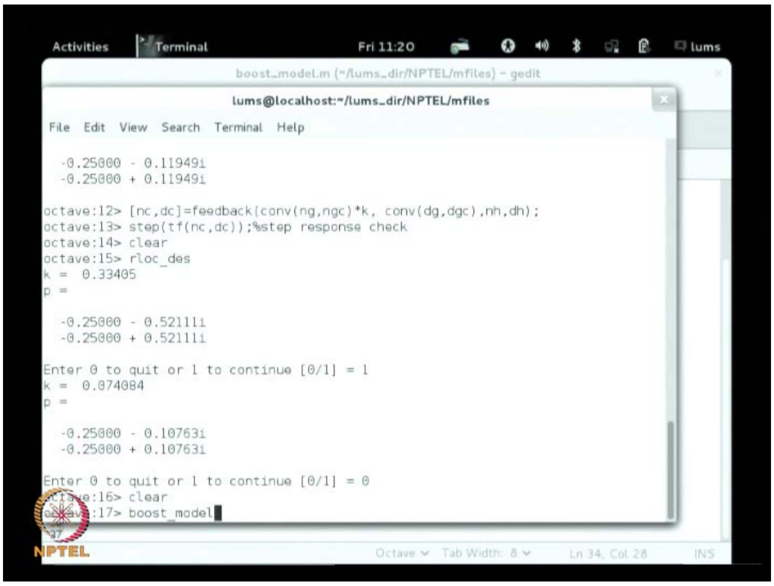


Let me open this script file. So the boost converter, these are the parameters vg 15 volts duty cycle of 0.4 inductance, 2 milli henry, capacitance 10 micro farad, load resistance of

100 ohms and switching period of 50 micro seconds or 20 kilo hertz switching. The steady state model we have gone through that we have developed that before. The A matrix said 2 by 2, you have be a B matrix, the B matrix is 2 by 3 meaning there are three possible control inputs, which is it could be the input v n itself are iz the load or d the duty cycle. We are of course, bothered about control with respect to D. And we have the steady state model, and then the small signal model ac. Now the small signal model is the one that will be using for control.

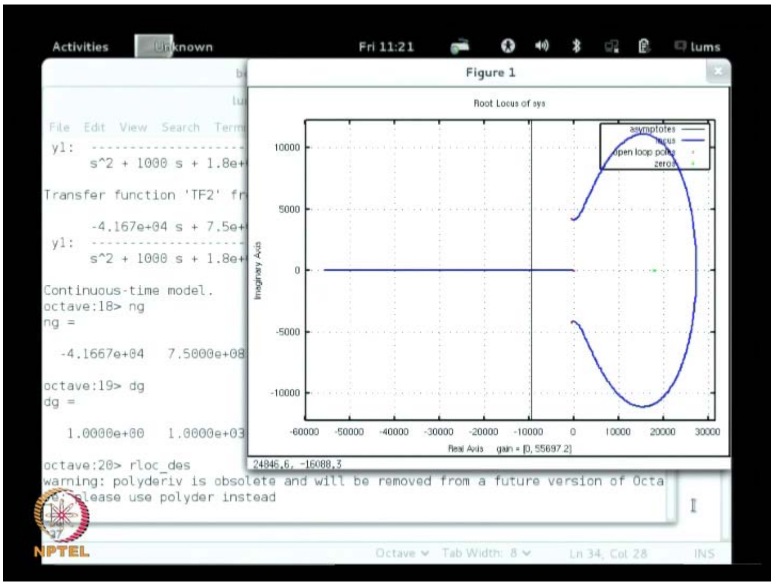
And after substituting this values it will execute, and then you convert the state phase model to the transfer function model; this is zero pole, zero pole display and this is the transfer function model, let us just take, so this last statement says ss 2 tf state phase to transfer function of Ac, and B matrix we are taking only the that column which represents the duty cycle input column then the C matrix and the D matrix this will give you the ng and dg. Now this ng and dg you will use it in the root locus design. So we could do let us disable this, so that ng dg disabled here and we execute this boost model.

(Refer Slide Time: 53:27)



And from the workspace, we will clear all let us boot boost model.

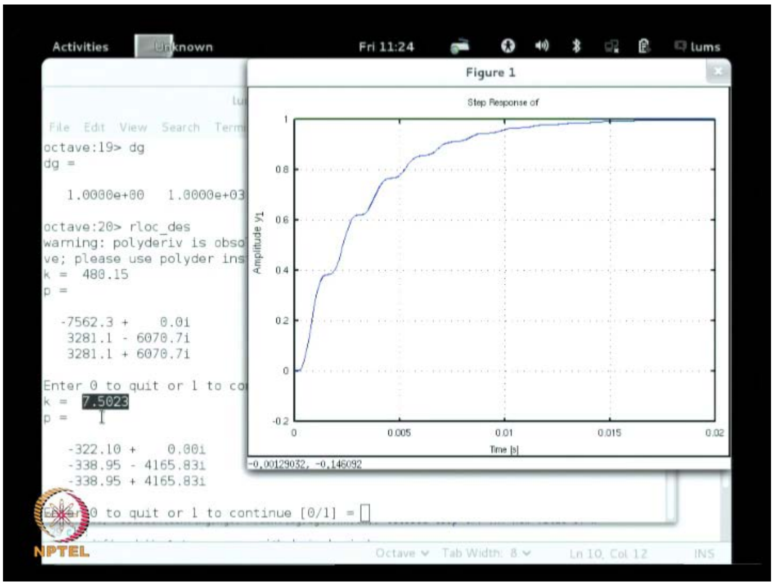
(Refer Slide Time: 53:35)



You will execute that. So, you will see the steady state outputs, the transfer function, this is the transfer function of the boost converters small signal with respect to, with respect to vg input with respect to the load the second one and then with respect to duty cycle. And it gives you the ng and dg, the numerator polynomial ng and the denominator polynomial dg of the transfer function with respect to the duty cycle input; that is s. So that is what you have errors ng and dg. Now you could run the root locus design script such that it takes this ng and dg, and applies the integrator just only a plane integrator, because in the root locus we just kept the plane integrator, if you remember.

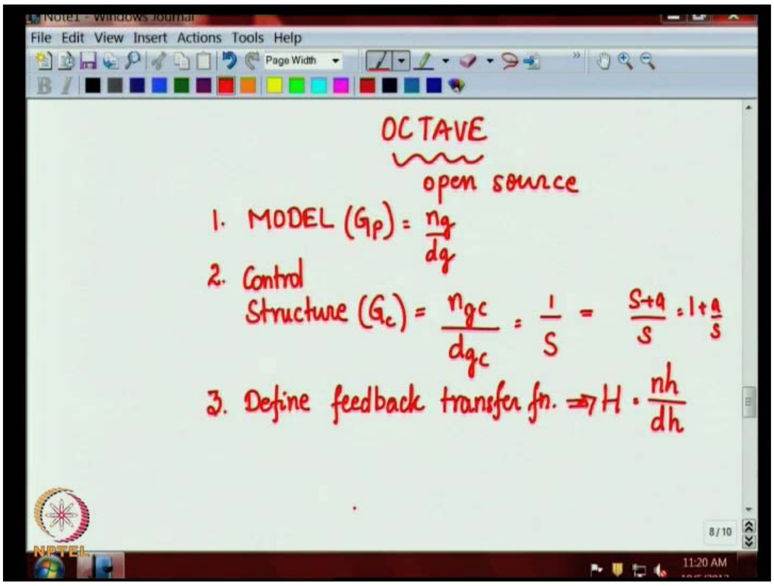
Remember that in the, we have right now the control structure is 1 by s, which is an integrator of course, you can make the control structure p pi pid later on. But the concept is to propose that, and then we get a root locus plot generated by the rloc find 2. And we need to specify closed loop location, observe that this is the 0, this is the right of the s plane; if I click anywhere on the right of the s plane, which means you are trying to choose a closed loop pole on the right of the s plane, you will get step response, which is unstable zero and then starts growing.

(Refer Slide Time: 55:58)



So, you have to now as this is pretty well compressed, let me zoom it towards this point here, and you see this is the 0, and I have to choose in the left of the S plane somewhere here. Let me now make the choice at around this, automatically other three are chosen, and then you have a step response which is something like this, for this value of K. These are the closer point locations. For this value of K, which is used as the closed loop as the controller gain, and then K into G c into G p divided by 1 plus K into G c into G p into H that closed loop transfer function has a step response like this. So, you can keep iterating and try to get better and better step responses, and then design the value K; once the value of K is designed, then your controller is defined; everything in the controller gets defined; you can then add, now you can then add as home work to this.

(Refer Slide Time: 57:17)



You can add, we had right now given 1 by S, you can now make it S plus a by S, which means now you have put proportional, proportional plus integrator, which is 1 plus a by S, is a proportional plus integrator, add these and then try it out. And then choose the gains accordingly, then plugged at into the simulation. So, we will just try a simulation of the boost converter in the next class, and then go on to the next topic.

Thank you for now.