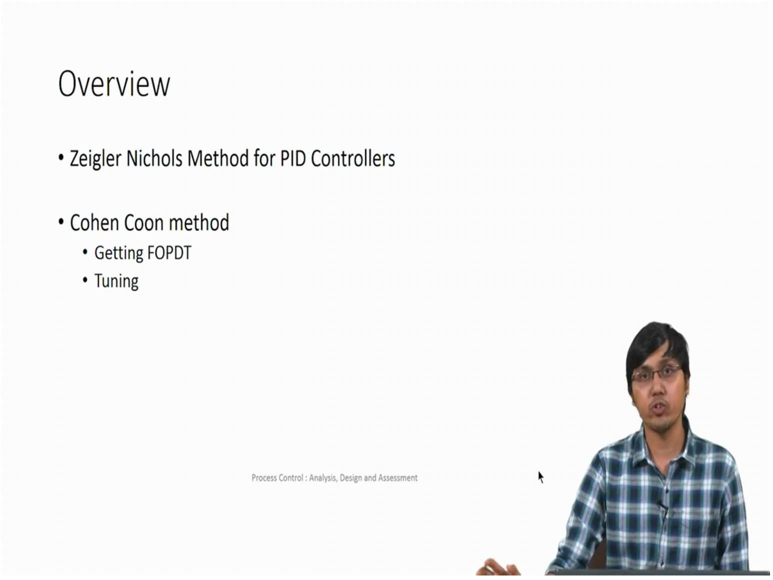
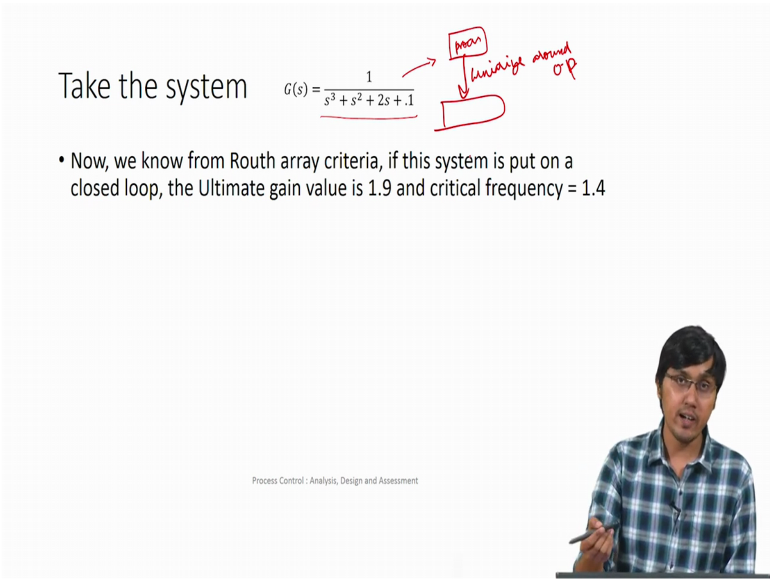
Process Control-Design, Analysis and Assessment

Doctoral Research Scholar Parameswaran S
Department of Chemical Engineering,
Indian Institute of Technology, Madras

PID Tuning

(Refer Slide Time: 0:22)





In this topic we are going to talk about PID tuning. So, we are basically going to look through Ziegler Nicole's method and Cohen Coon method and we will just give an idea of all these things and you can try out with all these formula. There are a lot of formula for PID tuning like there is a, there is a lot of ways in which you can tune the PID controllers. So, these are the one of the most commonly used methods, so and you can also try using these things and you can refer the literature like there are a lot of tools for PID controllers, between the PID controllers you can use any methods (())(0:51), etc.

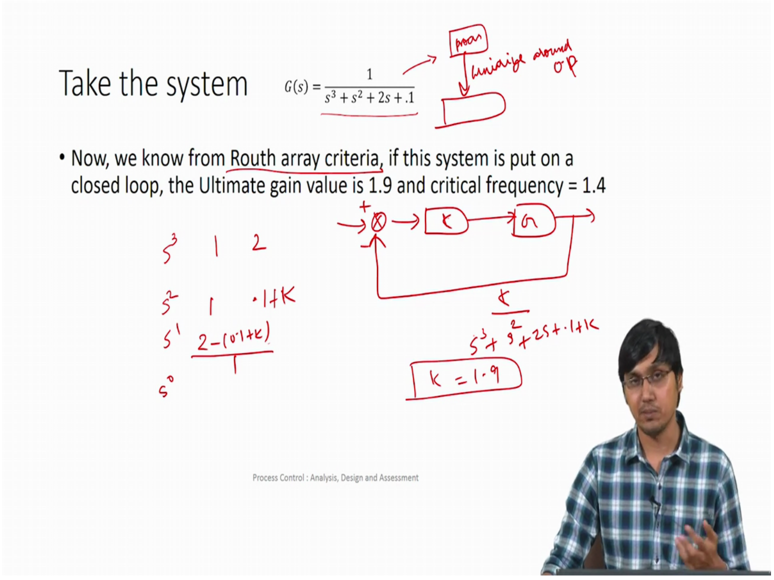
So, basically you will get an idea of what these 2 methods are. So, again we take the system to be some process, which is going to have a transfer function 1 by S Cube + S square + 2S +0.1. So, again, remember when we say this is a transfer function, this is the transfer function of the system or the process, so these are the process, so we get a linear model of it around that operating point.

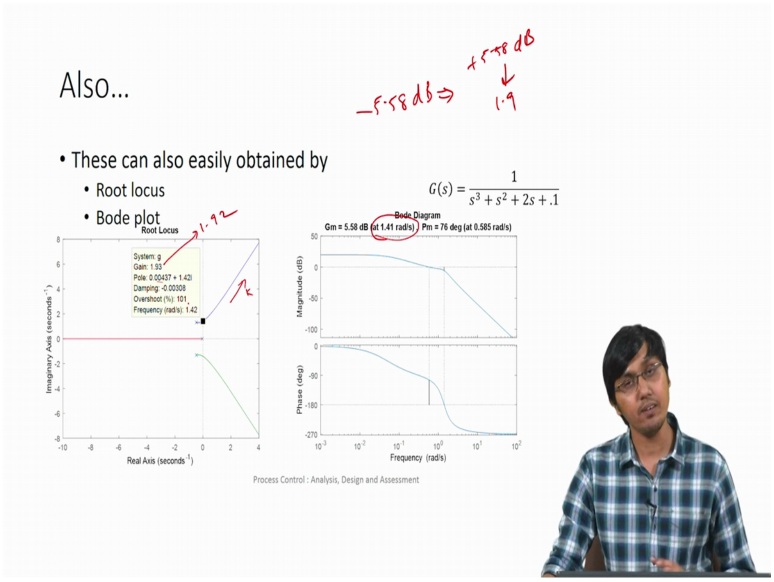
Why I am telling this is because like, let us say I have to maintain the temperature of the boiler to be like 200 degrees Celsius, I am just telling some numbers, random numbers, so this, if you take a boiler to be a non-linear equation, like if you will take it to be a system which is non-linear, then basically this transfer function will vary when when you if you will limit the boiler at around 200 degrees Celsius or 50 degrees Celsius or 40 degrees Celsius.

So, it is important that this transfer function is like around which point you will get this transfer function. So, what we normally do is, we will build the system around that point at which we tend to operate, at which we decide to operate, so that is what we say if the operating point. So, I want to maintain the temperature of the boiler at 200 degrees Celsius, so what I do, I will tune the PID controller like around the operating point. So, I will linearise the model around the operating point's, and I will get a model of it and then I will try to use that model to find the PID kinds.

So there are different methodologies when you have a very non-linear model but this is the idea you have to have. Take basically a linear system around the point at which we decide to operate it. So, we use of basic PID controller structure. So and this is the system which we already seen and we found like, okay, we found that ultimate gain value to be 1.9. The ultimate gain value is the gain at which the system will start going into the oscillations basically. It is a gain at which the poles of the closedloop poles of the imaginary axis, it can lie on the root locus.

(Refer Slide Time: 3:39)





And it is also the gain margin of this, so everything is related, right, so that is what we saw in the previous thing. And critical frequency is approximately 1.4. So, this is what we have seen before. So, now from, you can use all kind of criteria, we have already solved it but you can solve it, like you can put S cube, S square, S 1 and then 0 and then you can put 1, 2, 1 and 0.1 here. And basically we have to remember, we have to use, we have to use Routh array criteria for a closedloop transfer function.

So, basically when we have a transfer function like this is the setpoint and this is the summer block and this is the gain, transfer function and then you feed it back, so basically what you have is a closed loop transfer function in nothing but K by this is G of S, so basically S cube + S square + 2S + 0.1+ K, we already saw this in the previous lecture, this one but still again we are writing it down. So, this becomes 1 by KP. And we also saw that when this will become 0, when you have liked 2-0.1+ K by 1, so that will become equal to 0.

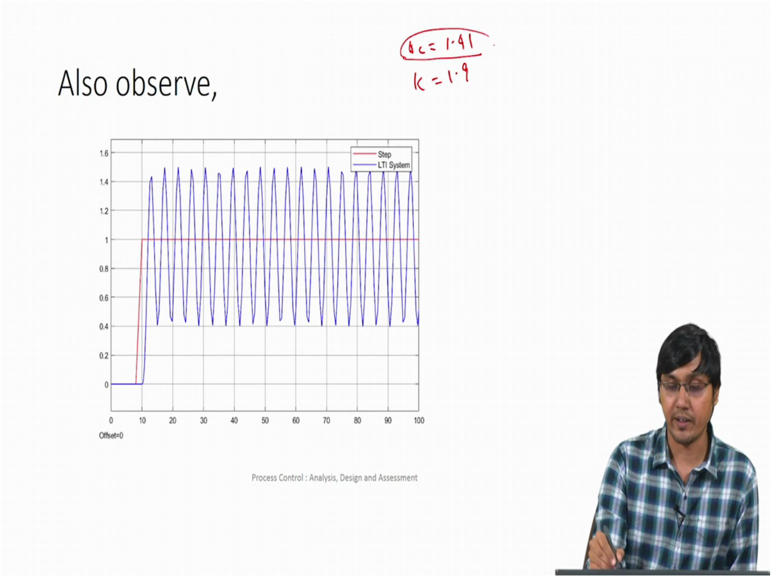
And this we say like if this is (())(4:50) so basically we can take like K equal to, from this if we quit this to 0, we get K equal to 1.9. So, this is the ultimate gain value from Routh array criteria. So, we already saw this in detail in the previous lecture, so you can go back and see but this is the simple thing to find what is the ultimate gain value. So, again, also like taking the root locus or taking the Bode plot, whatever technique you use, it has 2 come to the same thing, right.

So basically enough for the same system if you drop the root locus, root locus is drawn using the open loop transfer function, okay. So, as we varies again, it crosses the imaginary axis and if you see what is the gain at the imaginary axis at this point, then basically it is like 1.93. It is not exactly the imaginary axis, it is somewhat little towards the right of the plane but you can imagine that it is less than, again it is increasing in the gain.

So, to on the imaginary axis, somewhat close to 1.92 or something because that is what, this poll is not on the imaginary axis, so this will tilt towards the right of this plane. So, if you decrease the plane, so that you will get on the imaginary axis. And we can see here from Bode plot again like at -1 degree, what is the gain, the gain is nothing but -5.58 DB, which is nothing but which is saying that to make this system unstable, we have to add additional 5.5 DB to the gain, that is what the gain margin says.

Which is nothing but if you convert it to multiplicative gain, it is nothing but 1.9. So, everything gives there is the same value. And if you see here, the Bode plot also gives the frequency, the frequency is 1.41 and here also if you can see, the frequency is 1.41, so approximately it is 1.4. So, basically like all these methodologies should be giving the same values of the critical frequency and the same value of the ultimate gain. So, by doing different methods, you should not get different values, so that, keep in mind that one.

(Refer Slide Time: 6:54)



So, now again, like what is this Omega C equal to 1.41, what does that apply? It implies that when the gain is not equal to 1.9, this term is going to oscillate at the frequency of 1.41. So, that is what we saw before also. So, when we increase the gain by 1.9, when you multiply the gain by 1.9, that is when this term will go into oscillations. And at what frequency it will translate, it will translate at the frequency of, angular frequency of 1.41. So, why , if you ask the question why, because that is when we can see the wall crossing through and you can say this comes around, that is the kind of wave that goes in phase.

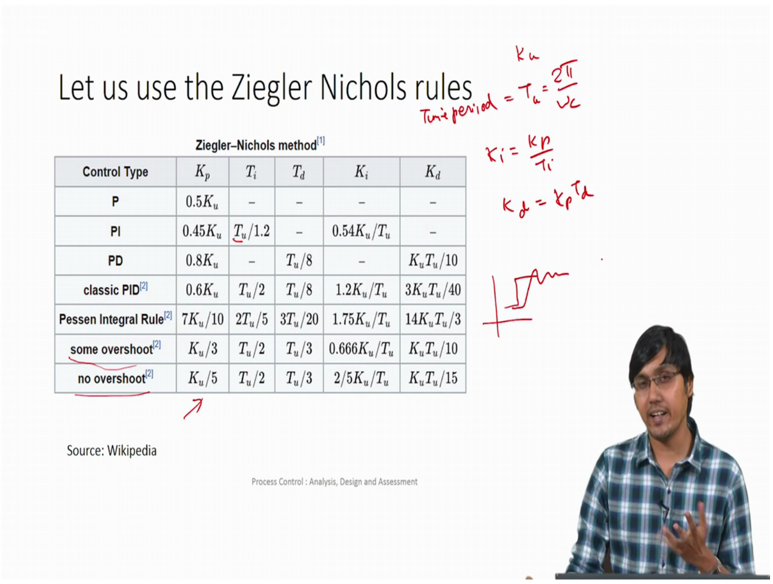
At that frequency goes, it comes around the loop in phase and it has the gain of equal to1, so it keeps oscillating at the frequency of this particular OmegaC at which the phase is in, the sine wave is in phase around the loop, so that is what we saw. So, this critical frequency is nothing but the, we will tell the frequency at which the system will the system will oscillate when the gain is increased to 1.9. When the gain is equal to the critical gain, that is when the system oscillation will oscillate at the critical frequency.

So, this is in, OmegaC is nothing but the angular frequency, so we will convert it into normal frequency. So, we know that 2 pie F nothing but OmegaC, so F is nothing but OmegaC by 2 pie. And frequency nothing but 1 by time period, so one but I figured equal to 1 OmegaC by 2 pie, which means time equal to 2 pie by OmegaC. So, if you can calculate the time, time comes around at around, I think around 4.48, you can substitute the value of 1.4 and see all these things. It comes around 4.48 which we can see here.

Basically we can see, we can pick this waveform, we can pick this period. So, the sine wave starts here, it completes one oscillation here and then it completes one oscillation here. So, basically within 10 seconds basically it completes 2 times and it has some more gap. So, basically ya, 4.48 should be working, right. So, that is what we can see, approximately 4.48, if this is 4.48. It is like it is like you can zoom it and see, basically becomes equal to 4.48, that is what we can see here. So, yeah, so basically everything, even if you plot and see, you will be able to find the value of the critical frequency.

Even by plotting you can find you can find the time period of 1 sine wave here and then with this time period you can find omega critical frequency by just backstab shooting. You remove the time period and you find OmegaC and that should be around 1.41. Just remember this, all these methods should give the same value, that is what we are seeing.

(Refer Slide Time: 9:54)



And then Ziegler Nicols method, that the table, so from Wikipedia we have taken this table. There are a lot of tables, so basically what you do, once you find the ultimate gain, that is KU, and once you find the, this is nothing but the period of oscillations, TU is nothing but the time period, which is nothing but TU is nothing but, we have derived in the previous slide, what is it, 2 pie by OmegaC, it is 2 pie by OmegaC. S

o, we have calculated the Tu value. So, once you know the value of KU and TU, KU is nothing but the gain at which the system will go, once you got KU and TU, you can substitute these values and then that find the value of KP, TI, Td and etc. That is what you can substitute. So, basically you can write Ki is nothing but KP by Ti. And KD is nothing but KP into TD. So, that is what you can substitute and senior, it will be like, you will be getting this K I and Khedi value.

So, in lookup table, you can have it in the paper and just look, find this value KU and TU and just substitute this into that and then you can see that. So, there are a lot of rules, like this is not the only rule that we have, a lot of rules that we have, so basically that is what. And one more thing that we can say here is like there are some kind of some other table, which says like, some overshoot, no one should. So, you can substitute, once you find KU and TU, you can just play around with this and see whether this gives no overshoot.

So why it is overshoot basically, so whenever, I want to give a step change in my, I want to go, I want to change the set point, check the variable from 50 degrees Celsius to 60 degrees Celsius and I want to give a step change. Then the control variable actually goes around and then comes like this. So, what temperature it will reach, more than the steady-state temperature is what is called the overshoot. So, one thing, (())(12:01) overshoot, why it is important is suppose you have a tank, okay and then you want to change the level from 50 percentage to 80 percentage of the tank level.

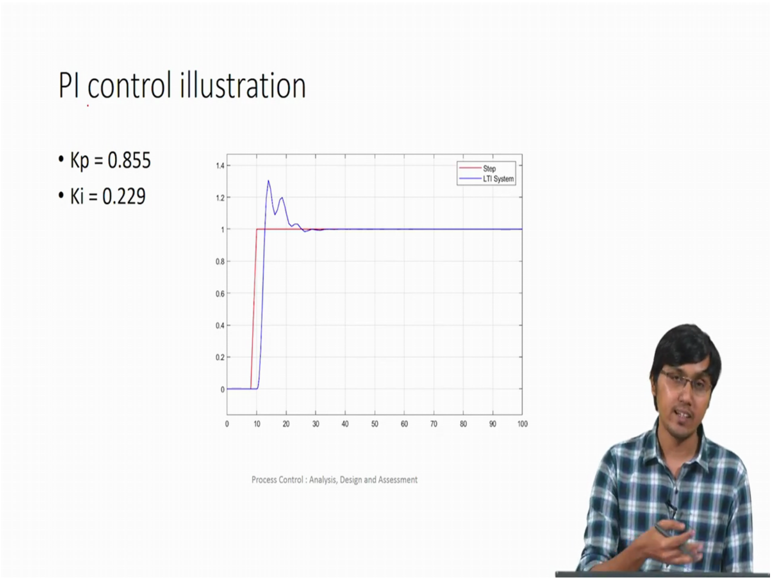
Now, you are interested to ask like if it has an overshoot, whether the tank will, like whether the water will flow out of the tank or not. So, basically if you have a tank something like I will just use this. So, if you have a tank, it is having like 50 percent, so you want to give a step change from 50 percent to 80 percent, we will take that. So, you want to see whether the tank, whether the level will go outside of the tank level or not. If it goes out, what happens, if it goes to 100 percent, then what happens, the water will flow out of the tank, right which is not desirable.

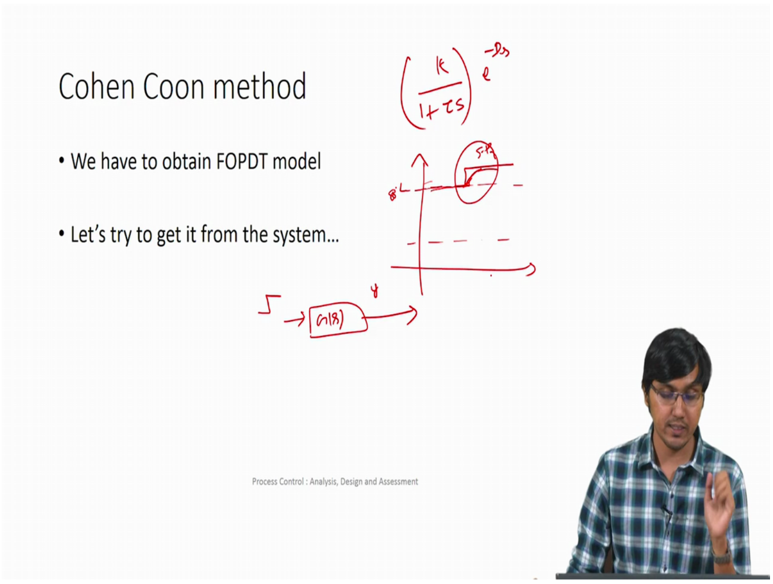
So, overshoot is kind of, you can think like overshoot why it is important in such sense. And so, this, when you do not want the overshoot basically you can use this particular formula. You can check this, similar tenancy what happens. All that you have to calculate is KU and TU and you can use different methods for KU and TU, UK News Routh array, you can use root locus, you can use Bode plot, everything gives the same value and you find this value and then you substitute, you use this, any of the form +, whatever it is here.

This is not the only site, there are a lot of rules available, you can go to, you can do the literature survey and then you can find it out and then you today and see what happens. And this classic PID is another thing that is like, mostly people, some, why it is like interesting is because what it says is like I will have a decay ratio of one fourth. So, if you have something like this and then settling like this, what it basically says is if you see this amplitude and this amplitude, you call it as A and you call it as B, then what it says is B will be nothing but, it will be like one 4th of the A, that is what it says.

So, 1 by 4 decay, everytime it decays, it is like one 4th value, 4 times in decreases basically. So, A will be 4 times bigger than B, that is what it says. So, this is another way of tuning. So, all these things are 4 different, all these things are like some form + for one objective. So, there are a lot of ways you can get this PID controller, so you can display around with this and you can write for yourself.

(Refer Slide Time: 14:19)





For illustration we have just taken a simple PI controller. Basically we have used this particular formula and this is what the output we have got and then this is the response we got. So, this is just for your illustration, you can just try with yourself with something and then you can verify how each things behave. So, all this is for getting an intuition of what these things do. So, when we have, we did a lecture on what the effect the PID gains have on the system.

So now we use different tuning methodologies and say okay, this is better for this thing and for this application, this seems better and this application this seems better, etc. And everything has its own advantages and disadvantages. So there are a lot of rules available, you can just so that you can just see. So, now, next method we are going to see is like Cohen Coon method. So, why this method is like Interesting is this is the next widely used method Cohen Coon is the next widely used method that you are going to have.

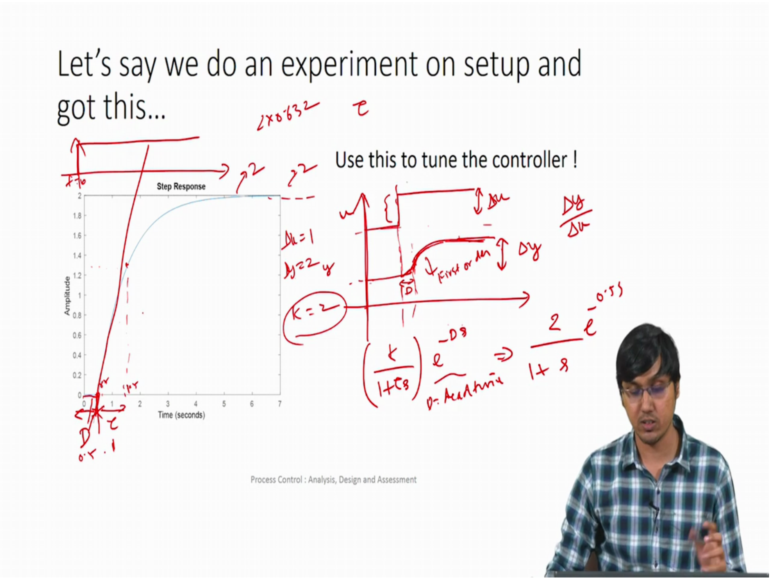
So, here is what we are going to do is we are going to get a FOPDT model. So, a FOPDT model is nothing but a first-order + that time the process. So, any system you can represent this something like K by 1+ tao S into e power - ds, so that is something like you can represent any system. So, let us say you have, you have been given a plant and let us say you are having say boiler effect. So, let us say I am giving you a system, I am giving you a system, I ask you to tune the PID controller for it. So what is it you are going to do?

So you, we may not actually from 1st principles start deriving all the things, like e equations and energy balance equations, energy conservation, etc. and finally we get, that is one way of doing it. The another thing is I have a system, have it now, can I do something quickly to tune the PID controller? And because like (())(16:21) deriving from 1st principles is again using approximations and then again that is not a very great methodology, because there also you will say some approximation.

So why do not take, just use a system I have and then I play with it, so that is what I can do. So, what I am going to do is basically I am, I wanted to maintain the setpoint to be like around 80 degrees Celsius. So, what I am going to do is I am going to give a step change, like close to 80 degrees Celsius. Okay, so what I am basically doing is I am using G of S, I am giving a step change , I am giving a step change in the inflow like the manipulated variable. So, I am giving a step change to the mall reflected variable of G of S and then see how the output changes.

For example if it is if it is boiler or something, what I am going to do, I am going to just put more fuel to it, so that I am, I am giving a step change in the fuel and then going to see how it is going to, how the temperature is going to vary. So, this is going to vary something like this. So, and now with this, can I get the model? So, my objective is to do this experiment to get a model, basically to get some coefficients and then use these coefficients and model to get some PID values KP, Ki, KD values. So, that is what is happening here, so this is the Cohen Coon method.

(Refer Slide Time: 17:47)



So we will try to see for the same system, for a system like this what is the, let us say like this is what I have got the response. So, basically what I have done, I have given a step change in the (())(17:53) variable at time T equal to 0. And this is what I got as a response. So, basically this is how my output is going to vary. So, what are the values I am going to get from this and this is how I am going to use these values for getting the controllers parameters is what I want to see here.

So, now here, if you can see here, basically I will just read out the things here, so that it will become easier for explaining. So, I have given step change in any manipulated variable, so this is what I am giving this thing as the manipulated variable. And now what happens, my, this is U, okay and what happens, my output is going to change like this, okay. So it is going to go like this, so it is something we can see. And 1st thing what I am going to say is, see this actually looks like it starts after sometime, right. But maybe actually it started here but it was not going so fast.

So this is one thing I will seek. And then I can visualize this as something like it is a first-order response, I can visualize this as a first-order response, she, I will say this is a first-order response's, which is starting after certain time. So, this I will call as dead time. So, I can consider this as a first-order response with the dead time, so that is what I can think like this. So it actually this is not a first-order dead time, basically it has started increasing here itself but it is slow enough and after that after a while it starts behaving (())(19:25) but for everything I have the approximation, here also I am just approximating things.

And then what I am going to do is I am going to just take this as a dead time and then I have the first-order system. So, it is basically like, the first-order system's response is something like this and am just shifting it, I am dealing it by time, by a dead time, therefore it is like, I am visualising this like this and then I am going to find what is the first-order that time process model. So, basically what it means, I am having a first-order model, which is actually having a delay of D seconds.

So, this is what it tells, this time is about, this is the dead time basically. D is the dead time. So, now what is the way I can look at. I can, what I will basically do is I will draw a slope basically, maximum, in this line what is the maximum slope I can draw? Basically I can draw something like this, so we are distant, I will draw something like this slope and that point at which it intersects the x-axis, that distance, dead time, what time it intersects the x-axis, that time is called dead time.

So, I can do this as a T. And then what I can do, from 1st, I is that it is a first-order, what I can do is I can tell like at 63.2 percentage, right, 0.623 of the total value, here it is 2 focus, at some value around 2 into 0.632, which is like 1.2 or something. So we have somewhere around, this is what is the time that is there, that is the time constant, right. So, that time constant I measure after the dead time. So, after the dead time, whatever is the measurement, that is what I am going to do, after the dead time, what is the time constant, that is what I am going to say here.

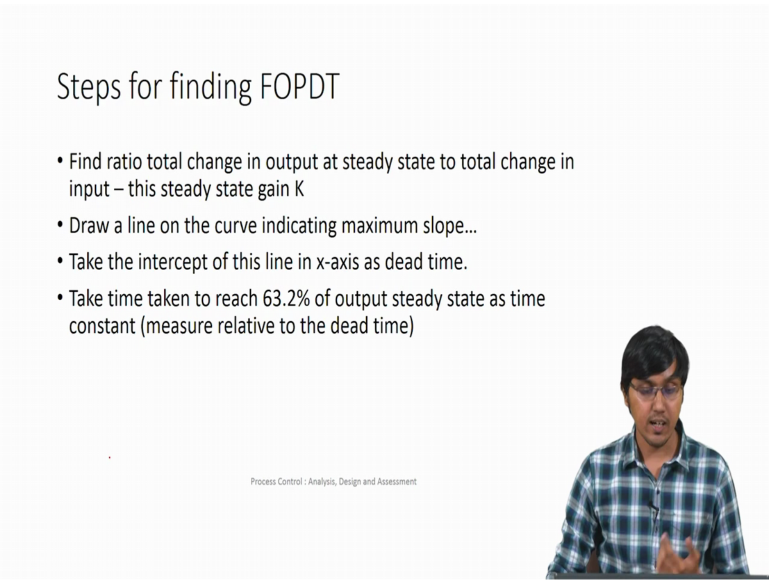
What is the time it reaches after a dead time, I estimate for the daytime, what is the time that it takes to go to 0.632 of the total threshold value. So, threshold value is 2 here, so this is nothing but one point to something. So, this we have seen from the first-order response thing. So, the time the time constant is nothing but the time taken for which 63.2 percent of the threshold value. So, that is what we observe from the first-order system's thing and that is what we are applying here and then we can find tao like this. And what is again, again is nothing but for step change in the input, so this changes U is 1, increased by 1.

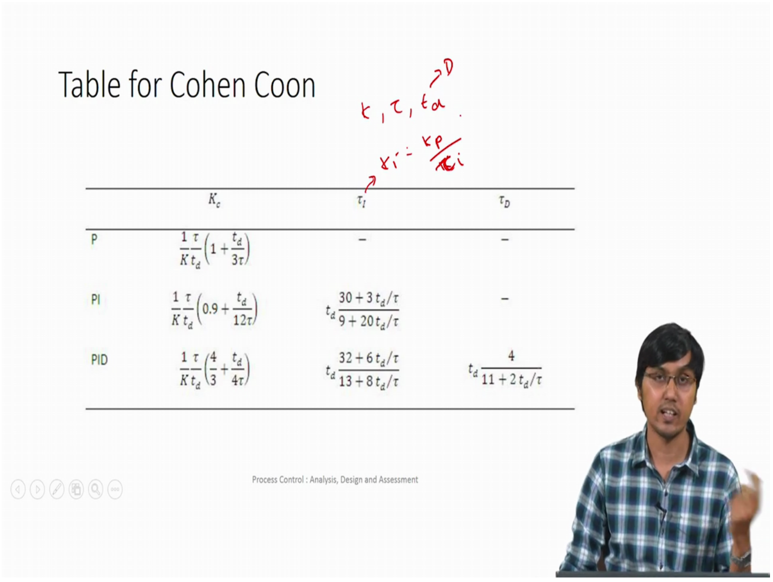
What is the final change I get here. So let us call it as Delta Y and let us call this as Delta U forces, if I give a Delta U change, how much change Delta Y has happened in the steady-state condition. The steady-state condition is like this here. So, if you give a step change, if it becomes 2, so basically it means for a 1, for Delta U it is equal to 1, for a unit step change, I am getting Delta Y is equal to 2. So, again is nothing but nimble 2. So, I get the gain equal to 2 and at 10 equal to around 0.5 and this Tao looks like around 1, right.

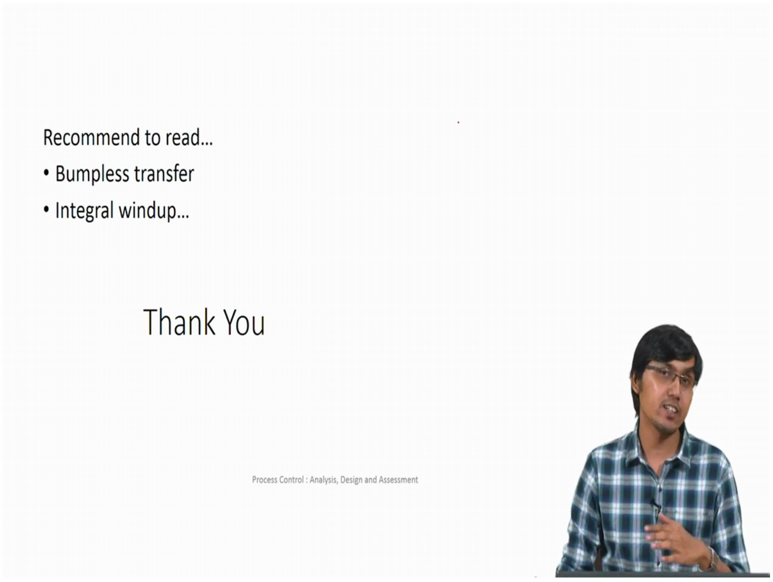
So my, so this is like a 0.5 to 1.5. So, this is like Tao's nimble one. So, basically this system, this particular system is nothing but approximately 2 by 1+ S e power -0.5 S forces, this is how we get the (())(22:36) value of that time, time constant and gain. So, again is the Delta Y by Delta U input. Like how much fuel I added, that is the input, that again is the fuel, that is Delta U, that is the input. So, Delta U is a change in the input given and Delta Y is how, at steady-state how the output change has happened, so that is the gain, Delta Y by Delta U is again.

And then delta, then D is nothing but if you draw a slope, where it intersects, that is what is the dead time. After a time it starts corresponding, that is what basically it means. And then Tao is nothing but after this dead time to reach 63.2 percent of the final value, final Y value, that is what is the time it takes (())(23:15). So, this is how we can find the FOPDT of a system. And that is what basically is given here.

(Refer Slide Time: 23:25)







You can go through this. And then, and once we add the value of K t, Tao and Td, Td is nothing but the dead time. We can use this table to find the values of KP, Ti and PID KP, Ti and Kd. So, that is what we can do it here. And this is nothing but, so Ki is nothing but, Ki is nothing but KP by Tao i. We have already seen how to convert from time constant of the gains of internal derivative, when we did it for Ziegler Nicols.

So, that is what, so you can you can try this Cohen Coon and then you can see how it works, how it performs. So, it is just some work, some play, play with the values and take some system transfer function, plot the graph and then take the FOPDT and then see what happens, etc. And then if you are still, if you are interested, there are a lot of, lot more concepts in PID controller, even in industries. Like when we use, if you go to the actual plant, like there are a lot more practical constraints in implementing the ready controller.

And some, there is something called a bump less transfer. Why it is called bump less, what is a bump less transfer and what is the integral windup. So we have an integration, if you have a P I Controller, it has integration in it. And what happens when this integration, it goes to a very high value and, you can just read through all these things. These are all like some interesting things, you can just read through, which we are not covering but just some, it is all available in the Internet, you can just go through this about what is the bump less transfer and what is integral windup.

So, basically you can play with all these things and you can try to implement something using Matlab, using this bump less transfer an integral windup and see what happens. So which may be some exercise for you. It is an interesting thing to do, thank you.