Process Control - Design, Analysis and Assessment

Professor Dr. Raghunathan Rengaswamy
Department of Chemical Engineering
Indian Institute of Technology, Madras

Lecture 27

Controllers for Unstable Systems

Let us continue with our 27th lecture, till now we have talked about difficult to control dynamics in the second part of this course we spend considerable time on looking at inverse response systems and how one would control inverse response systems and we followed that up with control of time delay systems, talked about the difficulties in getting the controller in terms of getting a realizable controller and then I showed how the direct synthesis approach could be used to design the controller and while we implement the controller we also see that you have to use the model during the implementation of the controller also and I said as we go forward you will see more and more of the use of model and directly in controller implementation.

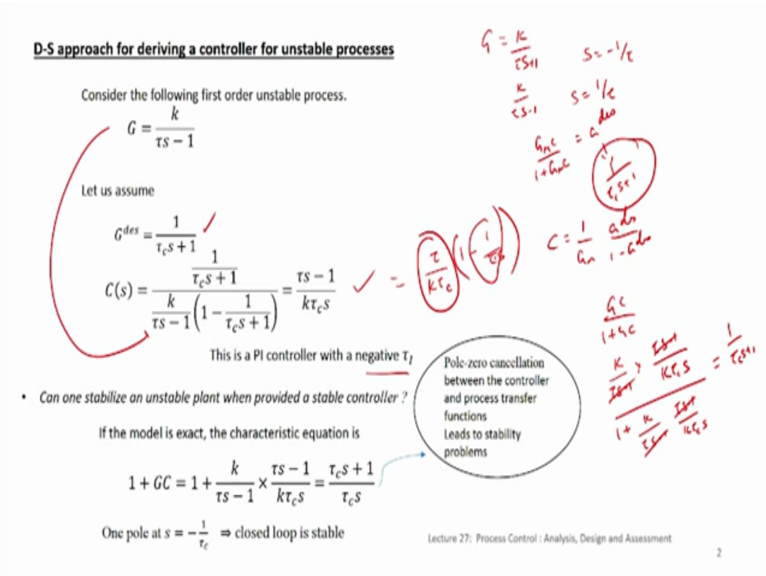
Nonetheless we saw how we could get reasonable performance with closed loop control when you have inverse response and time delay systems of course the time delay systems also introduces other complications in terms of stability analysis and I showed you a new way of doing stability analysis for time delay systems. Now we come to the kind of the last part of this difficult dynamics that we are going to address in this course and that is control for unstable systems, till now if you notice we talked about processes which are basically stable and the instability could arise when we actually put the process in a closed loop and try to get more and more performance out of the process.

So we have looked at systems which are inherently stable and not complicated from a stability viewpoint but we always looked at by closing the loop is there something that can go wrong and can the closed loop be unstable while the open loop or the process itself is stable, so in other words till now we have looked at how much performance gain I can get from a stable process before I push the closed loop itself unstable ok, so that is the kind of question that we have a so.

In other words we have always designed controllers for performance while making sure they do not go unstable however if the system itself it is unstable that is a process transfer function itself has unstable poles then before we even talk about and the performance of the system the real first question that we need to answer is whether I can actually make this unstable process go into a stable closed loop ok, so that is a question that we have never answered till now so this is the first time we are going to start answering the question.

So the main question is if the process is unstable can you still come up with controllers which will make the closed loop stable.

(Refer Slide Time: 03:11)



So we will start again to explain this idea with a very simple first order unstable process, so I have here a transfer function k over tau s minus 1 typically we have looked at G is k or tau s plus 1 in which case s equal to minus 1 over tau it is a stable pole however if I have k over tau s minus 1 then s equal to 1 over tau becomes a unstable pole because this is a pole in the RHP.

Now let us see what happens if I do not take this into account at all and then say I am going to have a G desired which is basically a stable G desired remember the closed loop G mc by 1 plus G mc the closed loop transfer function is what we calling as G desired. So now what I am going to do is I am going to see if I actually choose a G desired which is this that basically means I am saying there exists a controller I have assumed that I there exists a controller which can satisfy this relationship and once I satisfy this relationship I know though the model itself is unstable because this is a closed loop transfer function the, closed-loop is stable because I have chosen a stable transfer function here.

So the assumption is there exists a controller and of course we have solved this and then we have said the controller is basically 1 over G m times G desired by 1 minus G desired so let us see what type of controller we get, so if I substitute this 1 over G m here and G desired divided by 1 minus G desired I get this form here and you will quickly notice that this is going to be you can write this as tau over k tau c 1 minus 1 over tau yes so this you can write this like this and now you will notice that this is your PI controller with a gain of tau over k tau c and you are going to have a time constant which is going to be minus 1 over tau , so not time constant so sorry the integral parameter in the PI controller is going to be minus 1 or tau so this is going to have a negative tau i.

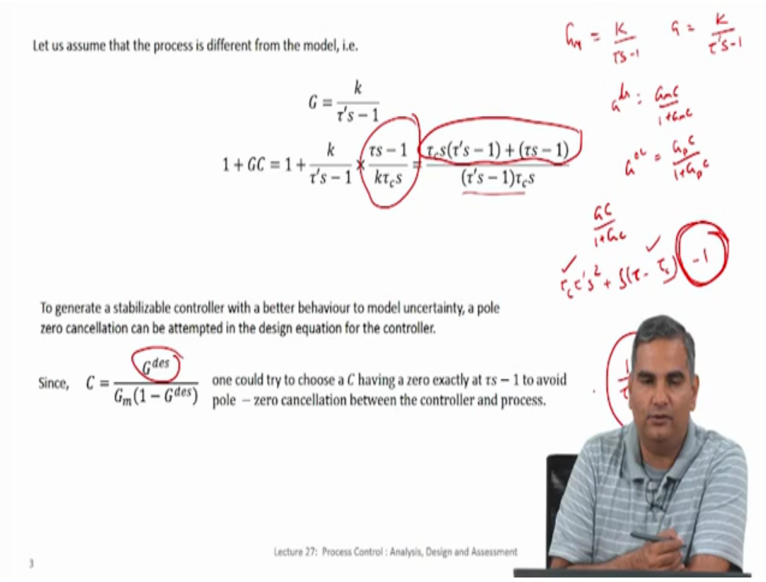
So now I look at this controller and then say well this is a PI controller which is implementable and I have an unstable process and I have got the G desired which basically is stable, so it seems like I can always get a controller that will stabilize this first order unstable process and give me a closed loop response which is stable, ok.

So then we can ask this question as to what is the problem with this right and when we look at this controller again the only thing that seems different is that now I have a tau i for a PI controller which is negative other than that there does not seem to be anything that is wrong ok, the thing that you also want to notice here is when we do this closed loop one thing I just want to point out when we do this closed loop which is G c by 1 plus G c, now G is k over tau s minus 1 see we have computed as tau as minus 1 divided by k tau c s divided by 1 plus G is again k over tau s minus 1 I have tau s minus 1 over k tau c s.

So when I do this closed loop computation of course this is going to turn out to be 1 over tau c s plus 1 because this is how we have actually designed the controller but nonetheless I just want you to notice something what we are going to do is we are going to do this pole zero cancellations in the closed loop transfer function computation ok, so this is something if you notice we have never done before in the closed loop computation we did do pole zero cancellation in the controller computation before when we talked about inverse response systems and so on but we have never done a pole zero cancellation during the closed loop transfer function computation so that is another difference.

So the two differences that I want you to notice are one is the first time we are getting a negative tau i for the i part of the PI controller and then we are also doing a pole zero cancellation in the closed loop which closed loop transfer function computation which we have not done before so we want to see because of this are there any issues that come about when we do this controller design for unstable processes.

(Refer Slide Time: 07:29)



Now you will see something interesting supposing let us assume for a moment that we designed a controller which was based on G m equal to k over tau s minus 1 let us assume that the actual process ok G is slightly different from this the k is the same but let us say I have a time constant which is very slightly different or slightly different from tau which is tau prime and then remember when we talked about this before I said you can design a controller based on G m and that is your notion of the process and you can actually say ok the closed loop behaviour will be G desired if G is G m but to really see the impact of a process which is slightly different from the model that you have chosen what you should do is? You should re compute G c by 1 plus G c using the actual model that you think is going to be operating.

So basically what we are saying is while G desired is G mc by 1 plus G mc to see the real effect of plant model mismatch when we want to compute the true closed loop transfer function you should kind of put G process c by 1 plus G process c and when this G process is slightly different from G m we want to see what happens of course when G p is G m G cm will be G desired, so this is something that we have talked about several times before.

So when we do that I take this G is k over tau as tau prime s minus 1 and let us just compute 1 plus G c and see what happens so when I do 1 plus G c I have 1 plus k or tau i tau prime s minus 1 the controller still remains to be the same because the controller was built based on the model which is tau s minus 1 divided by k tau c s ok, now notice in the previous case because tau prime and tau were the same we cancelled this in the closed loop transfer function we got our pole zero cancellation but now we have assumed that this tau prime is slightly different from tau.

So we cannot do this cancellation so I will get this tau c as tau prime as minus 1 plus tau s minus 1 and divided by this and then you will have another G c but remember the stability of the system really depends on the zeros of this 1 plus G c which is what we saw again in the last lecture on Nyquist stability, so we look at this polynomial and then say the poles of G c by 1 plus G c are going to be the roots of this polynomial which is the zeros of the numerator of 1 plus G c.

So we will expand this and see what happens so when I expand that then I get tau c tau prime s square plus s into tau minus tau c minus 1 is what I get, now this is a positive number this will also be a positive number because tau and tau c are in my hand because tau is the model tau, so as long as I choose a tau c which is smaller than tau which is what anyway I would want because the closed loop has to be faster.

So the close loop time constant has to be smaller than tau so this will also be positive but whatever I do I cannot get rid of this negative number so this will be negative and if you go back to the Routh table that we talked about remember the first step when you do this polynomial is once you make the leading coefficient of the polynomial positive then the first level of check we do so that there is no pole in RHP is to check for if every coefficient is positive, now this is positive this is positive but this is negative and I can do nothing about this so this will have roots in the RHP, so this is something that you cannot get rid of.

Now so what that basically says is that even if you have a small difference between tau and tau prime ok this cancellation does not work anymore and because this cancellation does not work anymore I land up with unstable poles in the closed loop transfer function, so even if there is a minor change between the true process and the model that is used then the controller becomes unstable, so this is not at all a robust controller so this controller will work and will work well only when the process and model are exactly the same which is very unlikely to occur.

So this is not a controller that can be implemented to stabilize the behaviour and we notice that this really happens because we have this pole zero cancellation not happening in the controller equation but actually this pole zero cancellation in the nominal case where the process and model are the same is happening in the closed loop transfer function computation, so as a general rule pole zero cancellations in in the closed loop transfer function computations can lead to robustness problem so that is something that that we should be vary off and avoid it as much as possible.

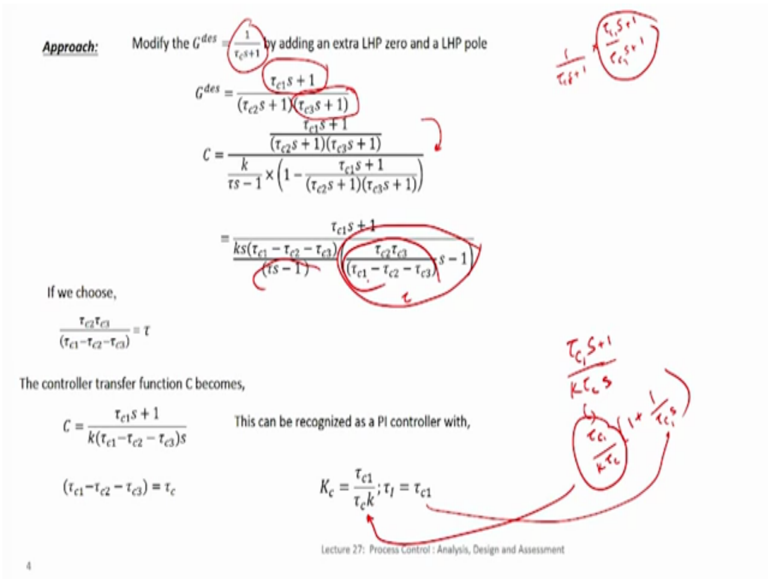
So now that we have this we are back to the question of how do I solve this problem right because it looks like if I use this direct synthesis approach I get a G desired and I get a G desired which is basically going to tell me to keep a zero where the unstable pole for the model is and then that will essentially come into a pole zero cancellation situation in the closed loop, ok.

So one idea is to see whether we can avoid this pole zero cancellation and make the controller robust also and again this is the beauty of this whole idea, so basically what it says is ok there are two places where I can do the pole zero cancellation I do not want to do this in the closed loop transfer computation so maybe I should do this pole zero cancellation in the controller computation right but if I want to do this pole zero cancellation in the controller computation if I give the standard G desired which is 1 over tau c s plus 1 then there is no possibility of a pole zero cancellation in the controller computation.

So basically what that does is it forces me to do to choose a G desired which is not my preferred 1 over tau c s plus tau c s plus 1 but something else so basically I am constrained to change my G desired because of the unstable pole, so this is where if you have an unstable process again that limits your performance just like how inverse response limited your performance just like how time delay limited your performance unstable processes also limit your performance because you are not free to choose any G desired you want because if you choose any G desired you want then that will leave a pole zero cancellation in the closed loop computation which is as we showed here is a problem in terms of the stability of the closed loop.

So we will see whether we can modify the G desired, so that we induce a pole zero cancellation in the controller computation so that is the basic idea.

(Refer Slide Time: 14:58)



So the standard technique to do this would be instead of saying I will have just 1 over tau c s plus 1 what I am going to do is I am going to add another tau c s plus 1 term here another tau c s plus 1 type of term here then I have a new G desired which is tau c s tau c1 s plus 1 divided by tau c2 s plus 1 times tau c3 s plus 1, so the motivation is so I have this one over let us say tau c s plus 1 term here and I want to go away from this so that I can get a pole zero cancellation at the same time I do not want to go very far away from this also because this is a G desired that I like the most.

So it is like this so I am going to multiply and divide this by a similar term which is tau c1 tau c in this case if you assume this is tau c2 just so that i show you the same thing there so supposing this was your most desired situation now you are not able to get this because the pole zero cancellation occurs in the closed loop, so I want to have something very close to this but at the same time where I have the possibility of inducing this pole zero cancellation in the controller computation, so what I am going to do is I am going to add one term here and one term tau c3 s plus 1.

Now if you notice if I keep tau c1 tau c3 very close to each other than this and this will be very close to each other and basically you go away from this tau c plus s plus 1 form but by not by much and I can control how much I go away from this form by choosing this tau c1 and tau c3, so if they are chosen close to each other than this numerator and denominator are very close to each other.

So in terms of going away from a desired transfer function you are not going very far away however now I will show you by doing this modification how we can actually induce a pole zero cancellation in the controller transfer function. So we start with this G desired which is tau c1 s plus 1 divided by tau c2 s plus 1 times tau c3 s plus 1 and then simply go through the algebra, so this step is just putting this back in and in this step what we are doing is we are taking this to this side and then simplifying it and then when you further simplify it you will get this form right here so this is a form that that I want you to look at.

So basically you can do this computation at home this is very simple algebra so once you do this algebra this controller is tau c1 s plus 1 divided by k s tau c1 minus tau c2 minus tau c3 divided by tau s minus 1 and inside the bracket I have tau c2 tau c3 divided by tau c1 minus tau c2 minus tau c3 s minus 1, now you notice that my interest is in introducing a pole zero cancellation in the controller transfer function.

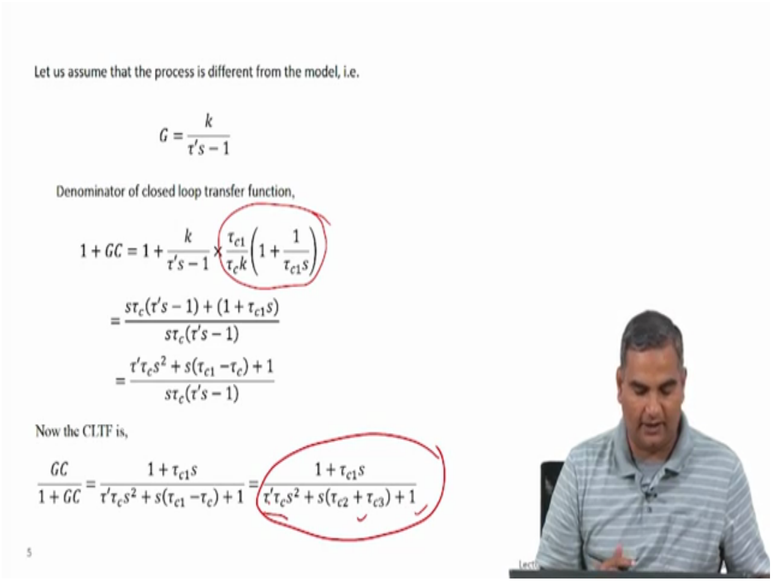
So to do that somehow I have to cancel this term and this term and now notice this tau this k over tau s minus 1 is a model, so the tau is what I have chosen as one of the model parameters to represent the process tau c1, tau c2, tau c3 are all choices that I made for my G desired, so I can choose them to be whatever I want so if I choose this to be equal to tau then I left tau s minus 1 and in the numerator i left tau s minus 1 so I can do this tau s minus 1 tau s minus 1 cancellation.

So notice how I introduce a pole zero cancellation in the controller computation so if I set tau c2, tau c3 divided by tau c1 minus tau c2 minus tau c3 is tau then I can cancel this tau s minus 1 and this tau s minus 1 and what I will be left with for the controller will be tau c1 s plus 1 divided by k times tau c1 minus tau c2 minus tau c3 yes ok now just for convenience if I just write this as tau c1 minus tau c2 minus tau c3 is tau c, a new definition of a variable then the controller will become tau c1 s divided by k plus 1 divided by k tau c s ok and this you will again quickly recognize this is the PI controller and the PI controller so this again this is the same trick that we have been using for a while now so this can be written as tau c1 divided by k tau c into 1 plus 1 by tau c1 s ok, so this tau c1 by k tau c is the gain of the (P) a PI controller and this is the integral time constant of the PI controller.

So what happens now is remarkably when we temper our expectations for G desired by going away from what we would like to have which is this form of 1 over tau c s plus 1 by adding a pole and a 0 to the G desired and then as I said before you can control how far away from this G desired you are going by manipulating this tau c1 and tau c3 values but basically the introduce this and then once we introduce this and put this is put this into the controller computation then we get another choice where I can set this tau c2 tau c3 divided by tau c1 minus tau c2 minus tau c3 is tau which induces a pole zero cancellation in the controller computation which leads to a PI controller for an unstable system.

So this is a very interesting idea now right, so see the difference between this and the previous case in the previous case there is no pole zero cancellation in the controller computation.

(Refer Slide Time: 20:48)



Now that we have done this now for all this effort that we have taken in doing this we want to know whether we get any purchase out of this what that basically means is that because of doing this as my controller become robust is a question that we are going to ask and to answer that question we take a process which is different from the model that we have used which is k or tau prime s minus 1 the same thing that we did in the previous analysis that I showed you.

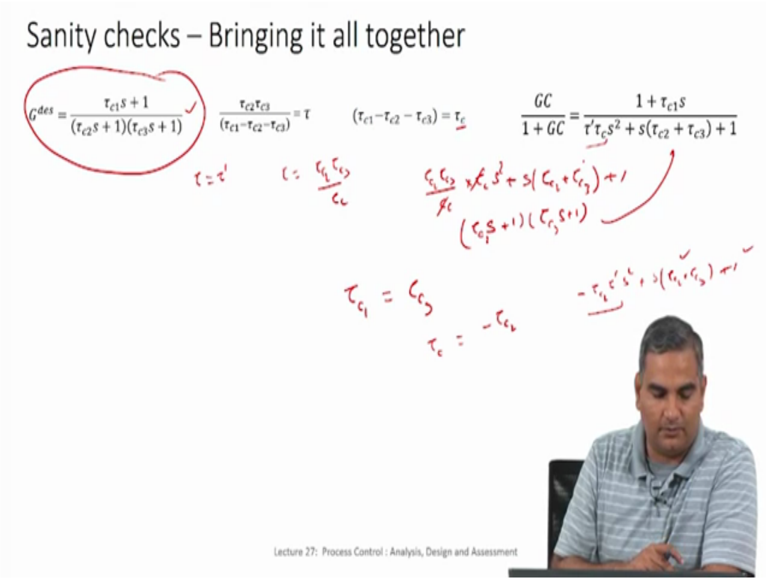
Now I do 1 plus GC I have 1 plus k or tau prime s minus 1 and now this becomes a controller this controller in the previous case had a 0 in the right half plane which was tau s minus 1 because I have cancelled this out this controller is a simple PI controller that I have here and now if you do the manipulation and do this you will get a closed loop transfer function which is of this form which is given here after all the manipulation is done, so you can follow through these equations and then basically derive this closed loop transfer function.

Now when you derive of this closed loop transfer function you notice now the denominator is where I have this as a positive number because tau prime is a true process time constant and that basically is a positive number the way we have defined this tau c is my choice and I choose that to be positive tau c1 tau c2 also my choice, so this is also positive this is also a positive number and you can show that this is going to lead to a stable closed loop, ok.

So by introducing the pole zero cancellation in the controller computation even though the process is different from the model that is used for the calculation of the controller we show that the closed loop will still be stable which is very different from what we had before and also look at how beautiful it is in terms of final result actually being a PI controller even for an unstable process the only thing is the PI controller is designed in such a way that the G desired if the process and model are the same is not going to be a very simple one over tau c s plus 1 but on top of it there is a zero and pole that have been added, so in some sense we have gotten a robust controller by giving up on the performance that that we expect.

Now let us do some kind of sanity checks on all of this, so if the kind of things that I am going to do is that if this tau prime is actually equal to tau then I am going to see whether I am I will just to make sure that I have not made any mistakes here I just want to show you how we can do some of the sanity checks so you would expect if tau prime is equal to tau then you would expect to get the G desired that we actually use to compute the c that is one thing that we can look at and then we can also look at whether I can do some tricks to still get whatever I want and I will show you why all of that will create problems and so on in the next slide, ok.

(Refer Slide Time: 24:08)



So if you look at this the G desired is the following with the definition tau is tau c2 tau c3 divided by tau c1 minus tau c2 minus tau c3 and this is tau c and G C by 1 plus GC assuming the process is slightly different from the model is this equation, now just for curiosity if you put that there is no plan model mismatch and then you say tau is tau prime then you will see that tau is tau c2 tau c3 ok, so you know tau is tau c2 tau c3 by tau c because this is the definition here.

So if tau prime is tau then if you take the denominator of this tau prime is tau so tau prime is going to be tau c2 tau c3 by tau c times tau c s square plus s into tau c2 plus tau c3 plus 1 so this and this will get cancelled now you can write this as tau c2 s plus 1 times tau c3 s plus 1 so which is what will go here so which will what be your G desired, so if this tau prime is exactly tau then we get the G desired that we started working with right.

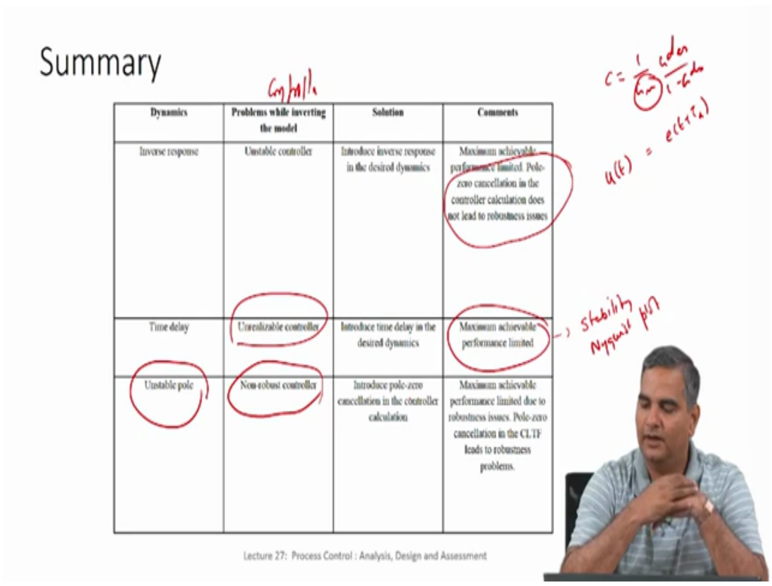
Now if tau prime is different from tau then you actually do not get this G desired you get something slightly different from this G desired so everything seems to work out here nicely now if you said oh well now that I have this and I seem to have a stable controller maybe now I go back and think about this a little and then say well if I want my G desired to be of the form 1 over tau c s plus 1 maybe what I can actually do is.

Now that I have done this what will happen if I actually set tau c1 equal to tau c3 ok so if that happens I will cancel this and this and then it looks like everything is great right because I get my transfer function form which is of the form 1 over tau c2 s plus 1 which is the best possible that I have been wanting and still the closed loop is all ok and so on. So you will quickly notice if you do this tau c1 is tau c3 then you will notice that tau c will become minus tau c2 ok and once you have tau c is minus tau c2 then you will have a negative coefficient here which will make your system not robust if tau prime is not equal to tau, so if tau prime is not equal to tau you have to retain this term as the same term.

Now if I said tau c1 equal to tau c3 to make sure that I get only a form of 1 over tau c s plus 1 then from this equation tau c1 and c3 will get cancelled now tau c will become minus tau c2 then you will have minus tau c2 tau prime square plus s into tau c2 plus tau c3 plus 1 this could be a positive number this could be a positive number but I basically have a problem here because I get a negative number here which will again lead to unstable behaviour.

So this beautiful illustrates that while you have this G desired as long as these two are not the same terms the system will be stable even when you have process model mismatch but the minute you make these two terms the same if I had a process model mismatch immediately I will get a negative here ok, so even a slight process model mismatch will lead to problems ok so it seems to make sense in terms of all of these ideas coming together in this example of controller design for unstable systems, ok.

(Refer Slide Time: 28:01)



So at this point we can kind of summarize the three different things that we have learnt in controlling difficult to control systems, so we started with inverse response and I described inverse response as systems where the initial direction of change is different from the final value and I gave you examples and showed you why these are difficult systems to control and the problem while inverting the model which is where this is a problem while inverting the model is basically related to the controller because the controller is 1 over G m G desired by 1 minus G desired right.

So the inversion comes in the controller computation so if I have inverse dynamics then I have unstable controller so the solution to this we saw was to introduce inverse response in the desired G dynamics itself that is include inverse response term for G desired then what happens is I have a pole zero cancellation in the controller computation and the comment is we are forced to introduce this inverse response in the desired dynamics.

So basically the maximum achievable performance that we can get out of these systems become limited the other comment is the pole zero cancellation in the controller calculation does not lead to robustness issues we showed that even if the zero of the model (trans) processed transfer function is quite different from the model we still do not see robustness problems in the particular example that I showed you.

So this is as far as inverse response is concerned then we went to time delay and if you use whatever G desired you want with the time delay system then I showed you that you will get an unrealizable controller which basically meant that if you wanted to find the out the u t from the controller then it would require error at t plus tau d that is future errors that it would require which is not feasible so you will end up with a unrealizable controller.

So the solution is to temper our expectations for G desired by actually introducing time delay in the G desired dynamics in which case again the maximum achieved performance is limited because you have to really have things happen with a delay and it just says there is no way of getting around it. The other comment here is that the stability study requires newer ideas and that's where we came up with this Nyquist plot for understanding the stability of the systems and which involves quite a bit of complex mathematics but basically we finally reduced it to simply looking at some plots to understand what happens.

The last difficult dynamics we talked about was unstable systems or unstable poles we talked about system with one unstable pole you can expand this to have more unstable poles and how you describe those and so on and while you inverted this it did not seem like a problem it just looked like you got a negative tau I for a PI controller but other than that just the first loop did not seem to have any problems but I showed in a closed loop if the process is even slightly different from the model then you get controllers which are not at all robust it means if the process is slightly different from the model you get an unstable controller.

So learning from here where we said this pole zero cancellation in controller does not lead to robustness problems whereas here we saw this robustness problem came because I am doing pole zero (compute) cancellation in the closed loop transfer function computation so learning from here we said let us move the pole zero cancellation from the closed loop computation to the controller computation.

So we introduced pole zero cancellation in the controller computation and this basically means that I cannot use any G desired this is where the performance limitation comes I have to temper my G desired with more terms which I did and then we saw that we got a PI controller and in the close loop even with process model mismatch we saw that we can get robust controller which was not possible without this.

So interestingly all of these are nice inter related ideas borrow of one idea from one system to another and so on and it is also very nice way of thinking about control and if you do not have very difficult dynamics in your process most of controller design is very simple and controllers operate quite well and only and of course you can ask for any performance you want and to a large extent you will be able to achieve that performance and the controller design and thinking about controllers become complicated only when you have complicating dynamics.

And in terms of complicating dynamics we showed you three different types of dynamics and if you were not doing this using the direct synthesis approach you just let us say doing the stability based tuning it would be very difficult to understand where the performance limitation comes from for this difficult dynamics right you will still do some find the gain at which the system becomes unstable and then you will back off and so on in that way of doing you have no notion of how much you are backing off and how much you are backing off because of difficult dynamics and so on.

So those notions all get clubbed into this backing off idea however when you do this direct synthesis approach for controller design and start understanding this then if you say the my gold standard is G desired and if there is a process which allows me to choose G desired any which way I want then I would say that process is not limiting me in any sense it is not the actual G desired itself but the choice of G desired should be completely up to me in which case the process does not limit me from choosing any G desired.

So I would say there is no performance loss however if I have to add certain things to my G desired because of the dynamics then we say that I have some performance limitations that are introduced because of those difficult dynamics in the inverse response case I am forced to add an inverse response in my G desired in the time delay case I am forced to add a time delay in my G desired and in the unstable process case I am forced to add extra poles and zeros in my G desired.

So there are certain constraints that are placed on G desired and I have to adhere to those constraints and that is where the performance limitation comes from, so hopefully this simple idea of direct synthesis leading to these kinds of insights for difficult dynamics is clear and interesting to people taking this course, thank you and I will see you in the next lecture.