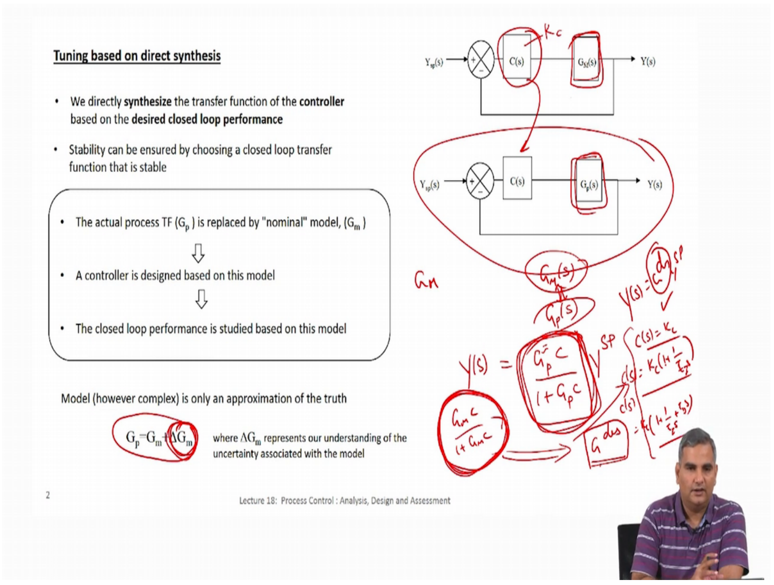
Process Control - Design, Analysis and Assessment

Professor Raghunathan Rengaswamy
Department of Chemical Engineering
Indian Institute of Technology Madras
Lecture No 18
Controller design and tuning - Part 2

So we have come to (())(0:15) of the course, we are at the last lecture for the first part which is looking at closed loop control systems and looking at 3 different types of controller P, PI, PID and what do those mean, how to tune the controllers and so on. In the last lecture we talked about tuning based on 2 distinct ideas one is what are called as stability based tuning where you look at how far you can push the controller in terms of the best performance and getting it to the edge of stability in stability and then the tuning rules basically back off from that point so the Ziegler-Nichols tuning table if you look at the tuning let us say for a P controller you would have noticed that the gain would be kept at 0.5 times KCU.

What basically that means is that if I keep the gain and KCU the system becomes unstable but I get the best performance but to hedge for the fact that the model might not be correct or I could have disturbances and so on, I pull back the value of KCU by 50 percent that is the reason why I actually keep the gain as 0.5 times KCU, so that is what I mean by backing off from the most aggressive value that you could have for the control gain and then using the same Ziegler-Nichols we saw that you could tune the PIA controller and depending on the KCU and PCU a gain and the time constant or the integral time of the PIA controller is chosen based on the table and if you noticed for all of those there will be some multiplication factors for this KCU and PCU and so on those you can basically interpret as how much you are trying to back off from the most aggressive P controller tuning, so that was the stability based tuning. I also said that the other way to do the same thing is what is called performance based tuning and that is what we are going to see here in this lecture.

(Refer Slide Time: 2:33)



Tuning based on direct synthesis

- We directly **synthesize** the transfer function of the **controller** based on the **desired closed loop performance**
- Stability can be ensured by choosing a closed loop transfer function that is stable

- The actual process TF (G_p) is replaced by "nominal" model, (G_m)

⇩

- A controller is designed based on this model

⇩

- The closed loop performance is studied based on this model

Model (however complex) is only an approximation of the truth

$G_p = G_m + \Delta G_m$ where ΔG_m represents our understanding of the uncertainty associated with the model

2

Lecture 18: Process Control : Analysis, Design and Assessment

However ultimately after all of this is done also said that there are tables that you look at and then use those tables for fixing values for your controller, so we know that let us say if I have a closed loop system let us look at the closed loop system here I have just drawn this as 2 loops so that we understand what we are doing and this will be important when you go forward. Some more advanced concept in control.

So typically what we will do is we will have a process as I said before we will model it linearize it, get a model okay now I am going to call that as G the transfer function for the model and for the first time I am going to kind of distinguish this from GPS which is the process transfer function okay the process transfer function in some sense you can think of this as the true process whatever that might be and whatever we use as a model whether it is first principal and linearise or first principal so third linearisation or just testing and then building a model based on data.

All of those are basically trying to approximate this process but nonetheless they are not exactly process, so you might think of them as G model transfer function which is supposed to be a faithful representation of the process, so you might want to think of this model as some GM plus Delta GM in a very simple conception idea that there is a model that I have and anything that is left over between the model and the process let me call that as Delta GM, so this is basically in some sense the uncertainty associated with our modelling right.

So this is not captured, this is unknown and hence there is always going to be difference between GP and GM and then we have to kind of account for it and there are very mathematical and very theoretical ways of accounting for it which is the field of robust process control and so on which we are not going to talk about here but nonetheless it is worthwhile to start thinking about controller in a slightly more sophisticated fashion at this point because the second portion of this course is going to need this differentiation in terms of models and process to happen because we are going to get in a closed loop diagram both transfer function for the model and the process.

Till now we have never made this differentiation because what we assume is this model is process, right so we base our control based on a model but since model is processed while we analyse how the controller will do is work and once implemented we again you the same transfer function because we assume process and models are the same, however for the first time what we are going to say is whatever we are trying to do in terms of tuning is essentially based on a model, so we put a model transfer function in a closed loop and then we are going to say if this model transfer function has to be controlled very well what is a controller?

Okay and then in reality what will happen is, this controller is actually implemented in a real closed loop and that real closed loop is going to work with the real process which I am representing by GPS. Now if it turns out that this and this are exactly the same, so whatever is the performance that you expect to get out of this exercise you will get here if this and this are the same, however if this is different from this and that difference is the Delta GM then this controller will still do reasonably well if this Delta GM is small but the performance that you will get from here will be different from the performance will get in the real actual plant implementation.

So this is reasonably simple idea but still sometimes we kind of miss this and then say okay process and model and we keep using them interchangeably, so it is a good idea to remember this. So ultimately what we are going to do is the following, so we are going to say look we are going to design a controller based on a model and then we are going to implement the controller and then if you want to realistically study effect of what this controller will be on process then you have to actually truly implement this and see what happens.

Simulation wise if you want to actually design a controller based on a model and then see how it will actually do in a real situation what you could do is you could assume this GP is you know GM by Delta GM and then you can take different forms for Delta GM and then keep the controller same and then studied this loop quite well, so that you can answer questions like even if the process is different from the model by this Delta GM and I can use different Delta GM, how well would this controller which was designed based on the model still do okay, so that kind of study that you can easily do, so that is something to keep in mind.

Now at this point anyway even when Ziegler-Nichols we assume the model is a process right, so again we are going to look at the model transfer function and then see how I can tune the controller. Now in the Ziegler-Nichols tuning and stability based tuning what we did was we chose a form for this KC and then we said at different values of KC what happens to the closed loop, when is it stable? When is it unstable? And so on and then we found the limit of stability in terms of ultimate gain and then we found ultimate period and then we were able to tune the PID controller, however ultimately after all this tuning if you say what is the performance of the controller, so the performance of the controller is how well does this y follow y set points, so this is something we have seen which is GP the true process GPC divided by 1 plus GPC y set point.

So the true behaviour right, once the controller is given if I want to see then I have to look at this transfer function, how close this transfer function is to one basically gives you how well the controller is going to do because if it is one exactly ys is equal to y set point, so that means output is exactly following the set point and so on but in reality it is never going to be 1 but it is going to be some other transfer function, so ultimately you have to remember that the performance of the real implementation of the real plant performance is actually dictated by this and we never know exactly what the GP is, we have a surrogate for that GP which is GM.

So typically what we do is we tune a controller based on GM and all the analysis of how well the controller will do will be based on this where we again assume GP is GM because we do not know what GP is and here is where you could do something like GM plus Delta, GM and so on to see different cases but ultimately the proof is really when this is implemented in a control system and then you see what happens with the actual process. Nonetheless without doing that if really we want to say how well the controllers is going to work one transfer function we can directly look at this GMC by 1 plus GMC assuming GP equal to 0.

Now in Ziegler-Nichols and other stability based tuning we are not looking at this transfer function ever. Remember that we are looking at this to construct the denominator polynomial but we are not directly specifying a performance and then constructing the controller rather what we are saying is okay if I have a controller form when will this system transfer function be stable, so we are only looking at this stability aspect of this transfer function and then we are saying okay when it is stable KC is this much let me put it back and so on a right.

Instead another way to think about the same thing is this is what is ultimately deciding the performance in the closed loop, so instead of doing all this complicates things where you choose control structure and then back off and so on why not I specify what my performance sponsor function is. I directly say I desire a transfer function of this form. Now this is what I will get from a controlled design, this is what I desire.

If I equate both of this and then compute a controller then once I put that controller back in I will get my desired performance okay so instead of doing the roundabout way it seems where I am going to choose a controller and find the limits of stability and then come back and back off and then say okay but heuristic and I can see how well that does by putting into this is one way instead of that you might say look I am going to look at something as my desired transfer function and since this this ultimate closed loop when this equals to this desire transfer function then I have got everything I want.

I do not have to really worry about stability here in the sense that I can choose G decide which is stable, so if I choose G decide which is stable the closed loop has to be stable because ultimately I am already saying this y of s is G desired y set point and this is what I am specifying and since I am going to be smart and specify something which is stable, so I do not have to worry about stability all I need to worry about is whether I can achieve this and I actually know functional from for this which is GMC by 1 plus GMC, so as long as I choose a controller which one substituted into this equation will give me my G desire then I am all done okay, so that is the basic idea of direct synthesis.

So I am directly synthesising the controller with a very specific performance transfer function in mind so that is what this race. So this is a very interesting and very nice, very simple idea seems like but this idea came quite late after quite a bit of feedback control had been done and this idea is the one that then leads on to this internal model control and many other very interesting ideas in terms of model predictive control and so on. So this direct synthesis is a key idea, so to summarise since I know the closed loop transfer function form I say look I am not going to do a controller base on stability, what I am going to say is I directly am going to put what the desired transfer function is.

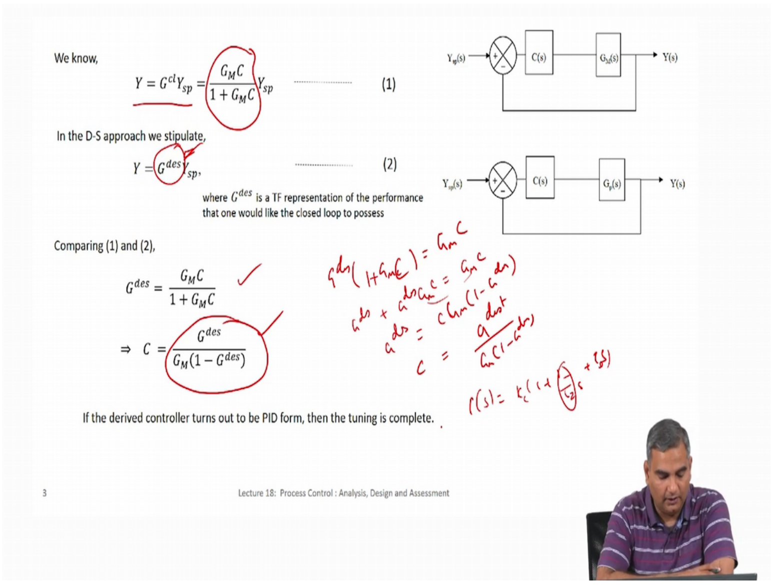
I am going to choose a desired transfer function in such a way that it is stable because that is a smart thing to do and then I am going to simply equate this to this and then compute the controller, so it looks like all advantages of course it is very powerful idea and lots of things about control and performance limitation and the control and so on you can learn from this idea and actually we will see in the second part of the course we are going to use this idea quite heavily. Nonetheless there are certain disadvantages in the sense that you are going to choose G desired and you really do not know what is the limit of your performance at least in Ziegler-Nichols tuning because you are pushing yourself to the limit you know what is the best performance that you can get.

Here there is a notion of what you want which might not really be anywhere close to the best performance that you can get, so that is one general conceptual idea that you have to keep in mind. The other idea that you keep in mind is that when we do this Ziegler-Nichols and so on from whatever experiment or exercise that we do, we actually get the tuning parameters of the PID controller directly. In this case the G desired (())(13:55) and then I computer controller and I am going to get controller structure and remember if it is a PID controller, if it is a P controller, the controller structure is KC.

If it is a PI controller we know the controller structure is KC times 1 tau Is and if it is PID I know it is KC 1 plus 1 by tau Is plus tau Ds right. So these are the structure that are already defined but when I put a G desired and then compute a controller, the controller need not be in any of these structures okay, so they might not even be in a PID structure, so if you trying to tune a PID controller using this and if the controller that comes out itself is not in the PID form then you cannot back out what other tuning parameters okay.

So that is the disadvantage but that is largely overcome by choosing G desire in such a way that the controller turns out to be PID or in cases where it is not turn out to be PID there are other minor modifications you can do such as adding a filter and so on and still getting the form that is a PID form and something else before and so on, so all of that has been worked out but I just want you to get the basic difference between the notion of Ziegler-Nichols tuning on this direct synthesis.

(Refer Slide Time: 15:15)



We know,

$$\underline{Y = G^{cl}Y_{sp}} = \frac{G_M C}{1+G_M C}Y_{sp} \quad \cdots\cdots\cdots \quad (1)$$

In the D-S approach we stipulate,

$$Y = G^{des}Y_{sp}, \quad \cdots\cdots\cdots \quad (2)$$

where G^{des} is a TF representation of the performance that one would like the closed loop to possess

Comparing (1) and (2),

$$G^{des} = \frac{G_M C}{1+G_M C} \quad ✓$$

$$\Rightarrow \quad C = \frac{G^{des}}{G_M(1-G^{des})} \quad ✓$$

$$G^{des}(1+G_mC) = G_mC$$

$$G^{des} + G^{des}G_mC = G_mC$$

$$G^{des} = CG_m(1-G^{des})$$

$$C = \frac{G^{des}}{G_m(1-G^{des})}$$

$$C(s) = K_c\left(1 + \frac{1}{\tau_I s} + \tau_D s\right)$$

If the derived controller turns out to be PID form, then the tuning is complete.

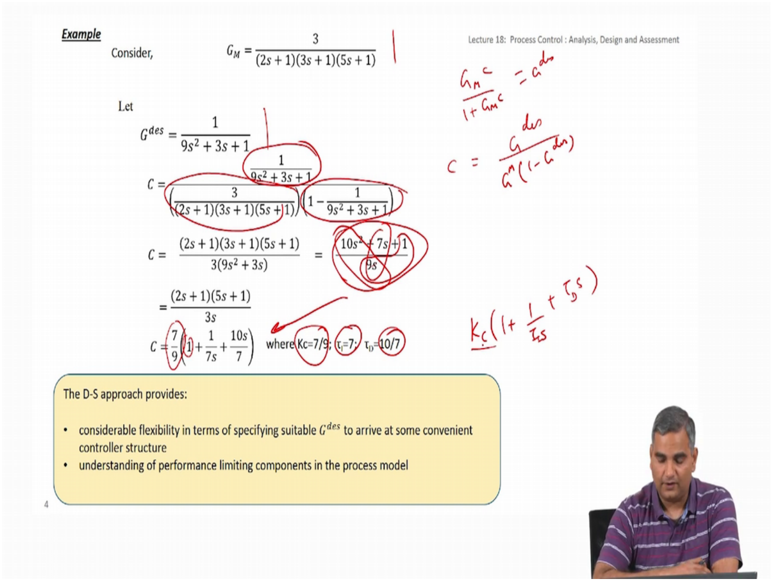
3 Lecture 18: Process Control : Analysis, Design and Assessment

So now that we have said this so let us look at how this controller structure is chosen, so y is this G closed loop y set point is something that we have been talking about quite a bit and we said this closed loop GP C by 1 plus GPC, now we have replaced that GP by GM, so basically this is a closed loop controlled with a model transfer function and as I said before I am going to stipulate whatever is the desired transfer function I have, so G desired what I am looking for, so this is something that I have to specify because this is a performance metric that I am specifying for this controller and once I have this performance metric then if I want to get this performance metric this has to be equal to this, so that is this equation here GMC divided by 1 plus GMC is G desired.

So if you do some very simple manipulations let us do that, so this is G desired times 1 plus GMC, GMC equal to GMC then what we need is we need all the controller term to one side right, so basically what we are going to do is we are going to have G desired plus G desired GMC equal to GMC right, so this is just expanding this so I take this to the other side, I take this ad G desired is C times GM 1 minus G desired is what you will get by taking this to this side and then collecting terms then this will give me C equal to G desired divided by GM times 1 minus G desire which is what I have here, so very simple manipulation that we do here.

Now this is the controller structure and the controller structure depends on whatever I chose as the desired transfer function and whatever is the model right, so remember this has no PID notion yet, so if it turns out that the derive controller is in the PID form then the tuning is complete. Why is the tuning complete? Because if I get SC as some form here which is PID form then I can write it in this KC 1 plus 1 over tau IS plus tau DS and then once I write this these numbers I can simply read of as coefficients of the polynomial. So tuning is already done, so that is a beauty of this, if this turns out to be in the PID form.

(Refer Slide Time: 17:42)



Lecture 18: Process Control : Analysis, Design and Assessment

Example

Consider, $G_M = \frac{3}{(2s+1)(3s+1)(5s+1)}$

Let

$$G^{des} = \frac{1}{9s^2+3s+1}$$

$$C = \frac{\frac{1}{9s^2+3s+1}}{\left(\frac{3}{(2s+1)(3s+1)(5s+1)}\right)\left(1-\frac{1}{9s^2+3s+1}\right)}$$

$$C = \frac{(2s+1)(3s+1)(5s+1)}{3(9s^2+3s)} = \frac{10s^2+7s+1}{9s}$$

$$= \frac{(2s+1)(5s+1)}{3s}$$

$$C = \frac{7}{9}\left(1+\frac{1}{7s}+\frac{10s}{7}\right) \text{ where } K_c=7/9;\ \tau_I=7;\ \tau_D=10/7$$

The D-S approach provides:

- considerable flexibility in terms of specifying suitable G^{des} to arrive at some convenient controller structure
- understanding of performance limiting components in the process model

4

$\frac{G_M C}{1+G_M C} = G^{des}$
$C = \frac{G^{des}}{G^M(1-G^{des})}$
$K_c\left(1+\frac{1}{\tau_I s}+\tau_D s\right)$

So let us take one example to understand how all of this works, so let us consider the transfer function model which is of this form. Now I have to do tuning for this, so I know again that my closed loop is going to be GMC divided by 1 plus GMC, so this is going to be G desired and now I have to choose G desired here, so here in this case let us assume that I chose a G desired of this form. The reason why I choose this is because this will help me cancel certain terms and get in the PID form and this is where the challenge lies, how do you choose a G desired which is better than the open loop while still giving you a PID form whenever you do this computation okay.

Again as I mentioned before when we talk about Ziegler-Nichols tuning and so on this will give you an idea of how you understand these 2 tuning approaches and how do you derive insights and lot more understanding based on this. Nonetheless ultimately when you are actually going to implement this again there are tables for MC tuning, there are tables for stability based tuning different types of tuning such as Cohen coon and so on that you can directly read off okay, so that is not really the problem here, the problem is you could read off but what is the underlying fundamental basis for this is what we are trying to teach okay.

Again so now that I have G desired now I know this C from the last slide I know it takes the form G desired I think divided by GM times 1 minus G desired, so this is a form. So this is G desire this is GM and basis 1 minus G desire. Now if you do simple algebra and then right it in this form you will get this C to be this form, so this looks like little strange form right here, so what we can do here is the following we can write this in a PID form because first term, this term will give you… this by this will give you the integral term. This by this because as soon as we will get cancel we will give the proportional term and this by this because I will have an s here will give me the derivative term, so this is what we can do.

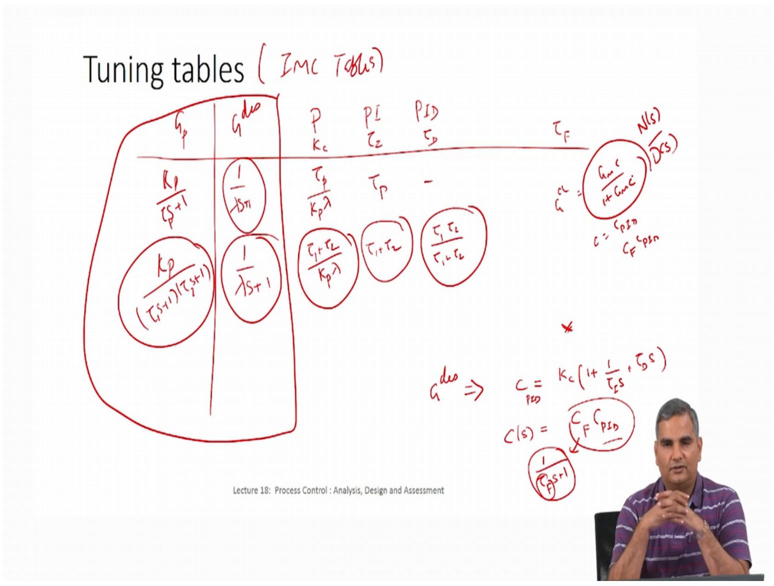
Now when you put all of this together and then likely rewrite it in the PID form you will get, so let us just look at how this works out, so I said this divided by this will give you the KC, so you see 7 by 9 gives you KC then I said this divided by this will give you the I term, however so this looks like one over 9s because since we are taking the (())(20:28) over 9 out this will become 1 over 7, so 7 and 7 gets cancelled you will get 1 over 9 times s which is what this term will be and then you have this term which will give you 10 over 9 times s because again we are taking 7 over 9 you will get 10 over 7 because 10 over 7 times 7 over 9, 7 on 7 will get cancel you will 10 over 9 term, so this is how this pole things work here.

So now when you look at this expression, now you notice that this is of the form KC 1 plus 1 over tau Is plus tau Ds, so that is the form that you have, so if you read off KC 7 by 9 okay 1 over tau I is one over (())(21:12) so tau I is 7 and tau D s is 10 over 7 by s, so tau d is 10 over 7, so this is what I mentioned in the last slide where I said once you are able to choose a G desired in such a way that the controller ultimately comes out in the PID form then it is easy to simply read off these number us and those become the tuning parameters of my PID controller.

So look how easy and beautifully this tunes the controller but the trick is in actually choosing the G desire. The key advantages of this direct synthesis approaches is actually specifying the performance directly yourself, so you know what you are going to get if the process and model are going to be very similar, so that is one good thing and more importantly when we start the next portion of this course we are going to see that we can really understand what limits the controller performance when I have a process right.

So what are the components of the process which I have to take special care of and what are the components of the process which will allow me to get a good control in what are the components of the process which will inhibit or prohibit you know great control and so on that is something you can easily understand from this very simple idea of direct synthesis. This will then form the basis for understanding what are inverse response systems? How do you control this inverse response systems and time delay systems, how do you control them unstable system how do you control them and so on, so all of that you can actually very nicely understand from this notion of direct synthesis.

(Refer Slide Time: 22:58)



Tuning tables (IMC Tables)
G_p
G^{des}
P K_c
PI τ_I
PID τ_D
τ_F
$\frac{K_p}{\tau_p s+1}$
$\frac{1}{\lambda s+1}$
$\frac{\tau_p}{K_p\lambda}$
τ_p
-
$\frac{K_p}{(\tau_1 s+1)(\tau_2 s+1)}$
$\frac{1}{\lambda s+1}$
$\frac{\tau_1+\tau_2}{K_p\lambda}$
$\tau_1+\tau_2$
$\frac{\tau_1\tau_2}{\tau_1+\tau_2}$
$\frac{G_p C}{1+G_p C}$
$N(s)$
$D(s)$
$C = C_{PID}$
$C_F C_{PID}$
$G^{des} \Rightarrow C_{PID} = K_c(1+\frac{1}{\tau_I s}+\tau_D s)$
$C(s) = C_F C_{PID}$
$\frac{1}{\tau_F s+1}$
Lecture 18: Process Control : Analysis, Design and Assessment

As I said before ultimately you will see tables for direct synthesis which will also be called as IMC tables which we will…IMC idea we will talk about later but I will tell you how the tables are generally going to look, so if you are going to look based on this direct synthesis approach there is one thing which is this the process model, so once you have given process model, in Ziegler-Nichols you never saw a process model right, so everything about the process model is somehow encapsulated in ultimate gain and the period because the whole tuning table was going to use only the ultimate gain and the ultimate period and depending on what the process is the ultimate gain and ultimate period will change that is how they bring in the process specifically into the Ziegler-Nichols or the other kinds of tuning tables.

Whereas here we will actually get the process transfer function itself here, so for example let us say this is KP over tau s plus 1 and then because this is based on direct synthesis you also have to give a G desire okay, so let us say the G desire in this case let us say is one over lambda s plus one then you could have the P controller term, the PI term, the PID term and so on. So P controller term will be KC this will be tau I and this will be tau D and in this case I have this then the P term is going to be a function of this KP tau 1 lambda, the lambda will also come in because lambda is the desire transfer function, so this is going to be tau P by KP lambda, so this could be you know one table that you have and this tau I could be tau P itself and something like this here okay so this is one possible tuning table.

Now the way you read this is supposing the process transfer function is this for you KP over tau PS plus 1 and the G desire that you want is one over lambda s plus 1 then you have to tune, if you want to tune a PID controller then you have to have this KCS tau P over KP lambda and the tau Is tau P, so that is what this is. Now if let us say you do a second order transfer function tau ns plus 1 and tau to s plus 1, so if this is your second order transfer function and then let us say your G desire is still the same in this case this could be tau 1 plus tau 2 divided by KP lambda and this could be simply tau 1 plus tau 2 this could be tau 1 tau 2 divided by tau 1 plus tau 2.

So now the way that you interpret this is if your transfer function turns out to be this and you want this desire transfer function then if you are going to tune a PID controller you should have tau one plus tau 2 by KP lambda as a P term tau 1 plus tau 2 as the PI term or I term and this has the D term tau D, tau 1 tau 2 divided by tau 1 plus tau 2. Now other than this you will also notice in this that there is also another thing which I will call as some filter tau f also in some cases will be filled up here, so the idea here is in many cases when you do this G desire you not directly get a PID form, what you will get is based on this G desired tuning if you get a C and if the C of the form KC 1 plus 1 over tau I as the reason why I keep writing this is because I want you to get comfortable with this form so that you do not have to commit this to memory then everything is fine you got this coefficient and you do whatever you want to do however there are cases where.

So if I call this as PID structure the controller that comes out of the direct synthesis after you give a desired transfer function, in some cases might have not only a PID structure because remember this we are computing it as based on the G desired, so there is no reason why this should fall in to the PID structure by choosing G desired we are forcing this to fall into the PID structure, so that is something that you want to keep in mind. In some cases it will still not happen, so it might be let us say of a form of a PID which you can separate out but then there might be something else and many cases this CS might turn out to be of this form tau f s plus 1 okay.

So controller is not only a PID controller in the front there is another transfer function which is called the filter transfer function which is basically another first order process that is appended to the PID controller, so that means this tau f has to be figure out, so there are cases where you will see also tau f but tau f again will be a function of whatever parameters you use in your G desired and whatever are your process parameters, so as long as you look at these 2 columns of these tables and you are able to simply compute PI KC tau I tau D based on the parameter values here and in cases where this is also filled up then you know that your controllers of this form and this tau f also will be computed as a function of this parameters.

Now you do not have to panic that suddenly I have PID controller what happens? Nothing happens because remember the closed loop transfer function is basically for us again GMC by 1 plus GMC whether the C is CPID or whether the C is CF times CPID does not really matter because once I put any controller here then if I want to analyse this I am going to simply do this as a numerator by denominator and I have already shown you many times how you analyse transfer functions of numerator by denominator.

So really this adds only another wrinkle in terms of more terms but conceptually nothing changes at all. So it is important to remember that the fundamentals are basically how you analyse the transfer function and that remains the same once you have this transfer function forms whatever C is given you simply put that in and then you will be able to analyse this okay, so with this I and the first half of the course.

Just a quick recap we started with what is the notion of control? Then we talked about what are the notions of the open loop and closed loop control and we talked about why we need closed loop control. We talked about server control and regulatory control or disturbance rejection control and then we said for doing control you need models. The first looked at physical systems and how we model them.

Then we talked about models as being comprised of conservation equation and constitutive relationships, we took certain examples to show how to write this models. Now the conservation equations can be mask conservation, energy conservation, momentum conservation and so on and the constitutive relationship are thanks that have been studied and derived from data from large number of observations and once you have these model equations we got to what is called as state space form of this model equation where I have states which are differential states as far as this course is concern on the left and side, so I get questions of the form DX1 by DT equal to F of function of states and some inputs and DX2 by DT function of states my inputs.

Then we showed how we can linearize this state space equation to get linear state base questions and then from there we moved on to this notion of single input single output control where we said we are interested in one output and controlling that one output with one input.

At that point I described how you think about these variables as seen there are inputs on manipulated variable, disturbance variable, exogenous variable and so on.

So all the terminology I described and then I said what is a disturbance variable and what is a manipulated variable all of that to some extent depends upon the design that you have a so if you have a particular line where you put in a valve flow in could become a manipulated variable but if you had a valve but you did not manipulated or left it at only one position then it can actually be disturbance variable, so it is important to understand conceptually what this means so that when you look at the design you are able to see where I can manipulate the variable where I have to treat variables as disturbance variables and so on and I also said when you think about straight space models, so you could have multiple stage what you can say look a single input single output control system.

Anyway after that we solve this stretch space models in time domain and then showed the output will be in what is called the convolution form or a convolution integral and then we also took the same state space equations and then multiply this e power minus STN showed you laplace transform and showed you how the differential equation, stretch space models to get converted into algebraic equation and once we have these algebraic equations in laplace domain we showed how the convolution solution in time domain becomes simply a multiplication solution in laplace domain.

So once we have that then we said okay you might get the output in the laplace domain but really ultimately for you to understand all of this you need to get the output in the time domain, so we said the processes you have a time profile for the input you convert that to laplace transform multiplied by a process transfer function get the output and then you do the inverse laplace transform and when you do the inverse laplace transform I said the most important idea this partial fraction idea you can do laplace transform of any inverse of any function which is a form a numerator polynomial by denominator polynomial.

From then on we understood how this inversion is done and there are only couple of rows in the laplace table that we use to do this inversions and once we did this inversions then I said any transfer function which is NSODS form follows the same procedure there is nothing special anymore. Then we asked the question if I have NS over D as transfer function, how do I figure out if transfer function is stable and so on and then we said if the denominator polynomial is in a root resolve form then basically look at the roots of the denominator polynomial which could be real or complex and as long as all the roots or strictly left of plane then we can say the system minus what we called as stable or bounded input bounded output stable which basically means whenever I give a bounded input the output will always be bounded if all the poles are strictly in the left of plane.

So this is as far as the open loop system is concern. Then we said okay that is the first time were we started by saying a control needs moral, now the model has been described, model has been solved, model has been analysed, so how do I do this closed loop control, so I talked about the notion of feeding back the error and the controller taking in the error as input and then giving the manipulated variable as output that way we were able to close the loop and then I showed you in an open loop the inputs are UFS and DFS and Y is the output but once you close the loop UFS is not an input anymore because it is going to be manipulated and then the values for that is going to be described by the controller but the controller is really in some sense driven or manipulated by the set point that we give.

So the inputs in the open loop which are US and DS now gets transformed into Y set point S okay and DS and then in the open loop we have transfer function between YS and US, YS and DS and then you can add them to get the overall effect on Y because of the linearity property the same thing translates to closed loop you have while and you have a term corresponding to YSP set point and the term corresponding to disturbance okay.

Now I said once I show you how to get these constant functions then analysis is straightforward because we are going to use the same NS over DS form and do whatever we did for close look open loop systems the same thing that you going to do for the closed loop systems, so that is what we understood at that point and then we said okay when we talk about this control what is the structure of the controller then I introduce the notion of PID control, P controller is a proportional controller where you are saying my input or the changes to the manipulate variables that I am going to make are going to be directly proportional to the amount of error I have.

In PI I am going to have 2 terms one is manipulated variable values proportional to the error I have and the other term is also going to be cumulative effect I am going to have a term corresponding to an integration of all the errors that I have had till now, so the first one is just based on present, the second term is based on the past and if you extend this idea you could come up with PID controller where the first term is based on the present here, the second term is based on past and the third term is based on future because I am going to take the derivative the rate at which error is changing which will give me an idea of how things are likely to change in the future.

Then PID controller in this time domain I showed how they get translated into the Laplace domain and the general controller structure is KC times 1 plus 1 over tau Is plus tau Ds which is a standard PID structure. Now you put this standard PID structure into your equation Y is some transfer function times YSP plus some transfer function times D then you get the effect of set point and disturbance on Y and to do analysis you do the same things that you have done in open loop, so that is the first part of it and then we saw how to analyse this, what are the advantages and disadvantages of these controllers through simple final value theorem.

I showed that if you have a P controller you will always have an offset, however I also showed for some simple examples how the P controller is quite fast and changes the closed loop time constant reduces the closed loop time constants very efficiently when compared to the open loop time constant, so P controllers are quick acting, easy to implement and the disadvantages off set, the offset can be removed completely by including an integral action is what we showed again through final value theorem.

So the advantages of PI controller or that there is no offset but we also showed that because of the integral action you can introduce oscillations. Now if you want to get rid of offset and minimize effect of oscillations then you go for PID controller, however the disadvantages of PID controllers is because I have a differential of an error if the sensors measuring your output are noisy then your controller will chat a little bit.

Of course one way to avoid that is the smooth measurement but if you smooth the measurement then the effect of future actually get lost a little bit in that process so conceptually PID while (())(38:38) grade in real implementation there are issues with some of these, so unless there is very special need there is a very specific case where uses typical PID is not used that much, PI is the controller that use most and P controllers are used in cases where you want fast control but you would not really worry about the minor off set that you have, so that summarises PID portion of this course.

Then we ask this question okay once we choose whether we going to use P, PI or PID how do we actually find out the values for this KC tau I and tau D. Then I said before we do that because this partial fraction idea is not going to be enough for us to understand this. I introduce the notion of Routh table, so here I said if I have a denominator polynomial which has coefficient which are all constants then you can actually get the root resolve form and then understand stability, however if I have this coefficient also as functions of the controller parameter then we want to know what happens to the stability of the system and without knowing the controller parameter we cannot actually get the roots and so on okay.

So the came up with this Routh stability table where remarkably you can actually make the decisions about whether there are any roots on the right of plane without ever computing the roots okay so that is the beauty of this Routh table and the stable season because now I have which are functions of the controller parameters, so I showed you that if you have a polynomial, if all the coefficients are positive then you go to the next type if you have even a single coefficient which is negative then we know there is at least one root on RHP.

So there itself you can stop and say there is a problem but if all the coefficients are positive then there is hope that there is no roots in the RHP, so further verify it what you do is you construct this table and if the first column of the table has all positive numbers then we guaranteed that there are no roots in the RHP again the beauty is we are not actually computing the roots at any time, however if there are negative numbers depending on the number of sign changes can figure out how many roots are in RHP. The number of roots in RHP is equal to the number of sign changes that you see in the first column okay so this is a Routh table.

Then the next lecture we showed that how this Routh table is useful because we looked at stability as being… The tuning of controllers being done in 2 different ways one way is through stability emotion the other is through direct performance based tuning. In the stability notion we want to see how far we can push the controller get the values and back off to figure out how far we can push the controller we can use Routh table. In the direct synthesis or performance based approach we do not worry about how far you can push the controller instead we say we know the closed loop transfer function, so let me directly gave transfer function for the closed loop transfer function and then back out the controller.

The advantages here is since you are specifying the desire closed loop behaviour you can keep it stable so you do not have to worry about stability. The disadvantages conceptually is you really do not know how much you can purchase out of the system we are just doing something based on your understanding of how much you can get. The other issue of direct synthesis is if you are really looking at only PID controllers because I am computing a transfer function for the controller it need not fall in the PID structure however by choosing G desired properly I showed you that you can get to the PID structure and in cases where you do not get to the PID structure there are some things you can do where you collect the other term and then generate what is called the filter function.

So your control becomes a filter times the standard PID structure and then I said you do not to panic because I have this extra filter because the closed loop is simply GMC by 1 plus GMC and C can be any form really you can do the same kind of analysis that you have been doing. I also mention that these are very important for you to understand fundamentally how all of this operates but if you are practitioner and then you are looking at implementing control then basically there are these tables, there are these tuning tables (())(42:58) tables, tuning tables which are other types of table. So all of these tables will give you the PID parameters in terms of either the process transfer function directly or in case of IMC table you have to have a process transfer function and the G desired form and the parameters in these will be used to compute the tuning constants in the PID control okay so with this the basic controller part of this course is done.

The next part I will start with this direct synthesis and then explain how we are going to look at more complicated dynamics which we have not seen until now and the first dynamics than going to talk about is what is called inverse response systems then I am going to follow it with the term delay system which will introduce difficulties which will force us to move out of this very simple numerator denominator polynomial form it is going to introduce some more complications that needs to be addressed then we will also look at unstable systems and then we will look at truly multi variable systems in the next one fourth of the course. So I will see you back again for that portion of the course, thank you.