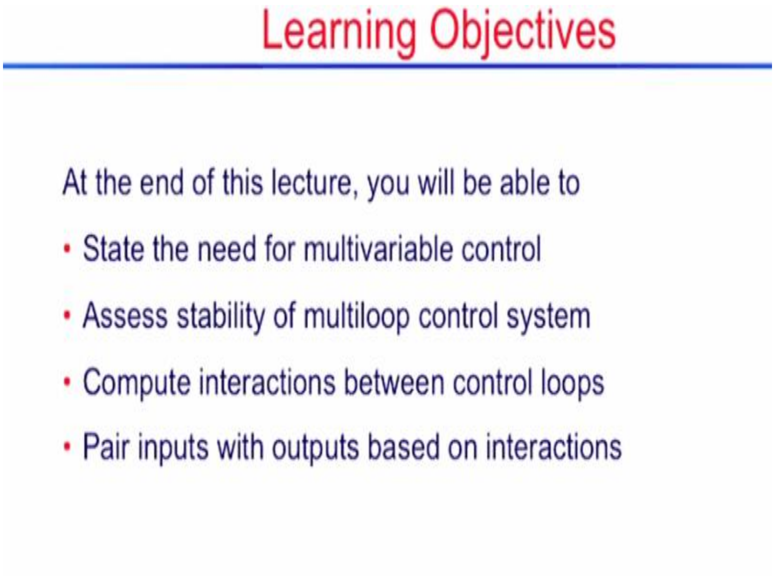
Chemical Process Control
Prof. Sujit. S. Jogwar
Department of Chemical Engineering
Indian Institute of Technology– Bombay

Lecture - 43
Introduction to Multivariable Control

Hello, students welcome to the last week of this course. In this week, we will be focusing on two topics which are of importance in terms of process control. In the first half of this week, we will dedicate our time to what is known as multivariable control and the later half we will spend on batch process control because controlling a batch process involves a lot of additional things than what we have been looking so far which was very easily applicable to continuous processes. So let us get started with multivariable control.

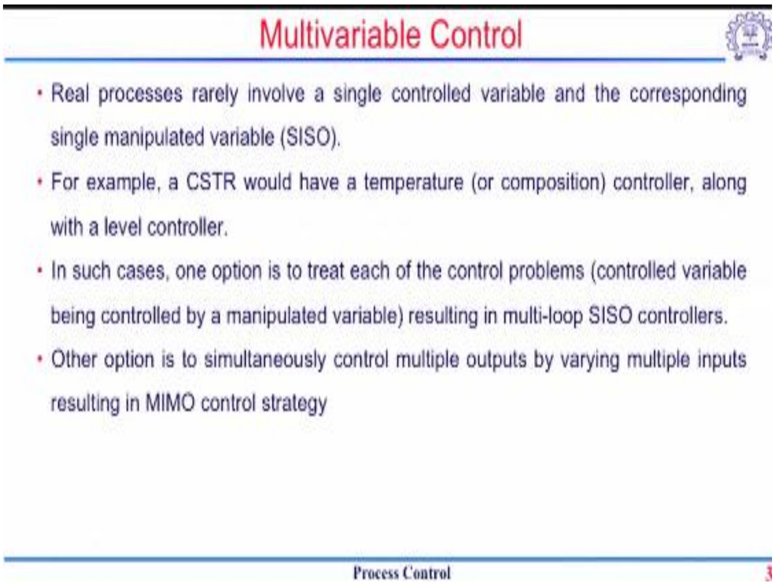
(Refer to Slide Time: 01:04)



Here are the objectives for this particular portion of this week. First I want you to learn why it is important, to look at the need for multivariable control. Then if you have a system which has multiple variables, then you should be able to assess the stability of that particular system. If you have such a system how do you compute whether those control loops interact with each other or not? How do you quantify interactions in such a system?.

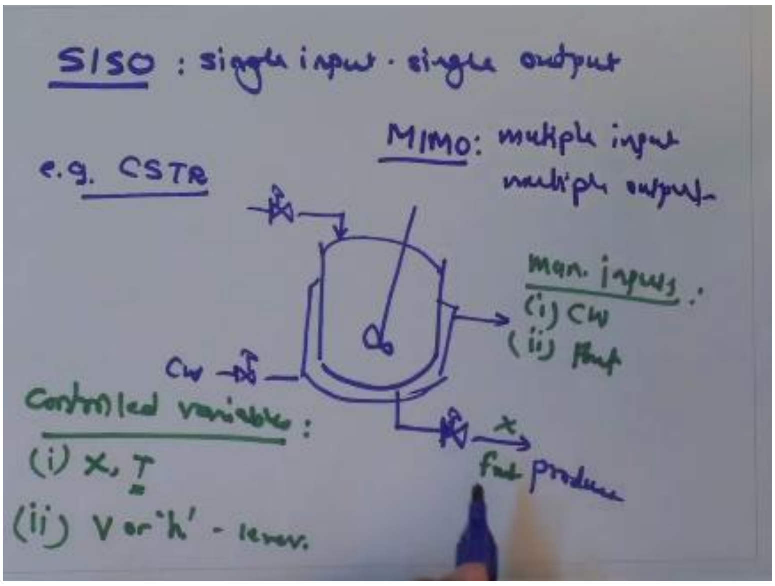
And lastly, how do you pair any specific input with a corresponding control output. So we will see all those things as we move along in this particular week.

(Refer to Slide Time: 01:54)



So to motivate this, let me first start with a comment that so far we looked at control algorithm when we had a single controlled variable and a single manipulated input. Now let us keep aside some of those traditional advanced controllers where we had cases where we had more controlled variables and few manipulated inputs or the other way round. But when it comes to the final PID control or the final control law, it always dealt with one controlled variable which has its set value and one manipulated input which will be used to reach that particular steady state value or setpoint value.

(Refer to Slide Time: 02:46)



So all those were known as SISO strategies which means single input single output, but when it comes to a real process you will rarely have a single variable which needs to be maintained or controlled at its value, and you will rarely have a single manipulated input dedicated to it.

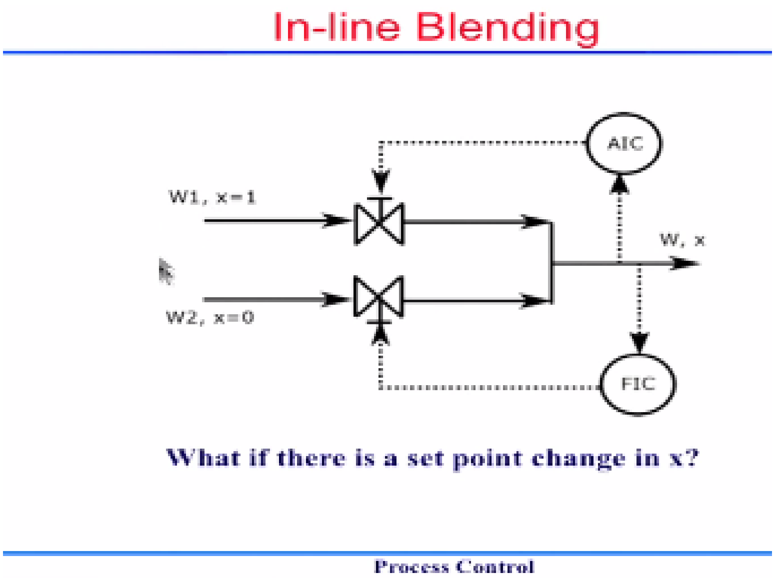
Let us take an example of a CSTR. So let us say this is a CSTR and you have a certain feed which comes in, you have a jacket around it. Depending on what sort of reaction it is carrying out exothermic or endothermic you would also have a product valve. I have put this CSTR to achieve a certain extent of reaction. So the objective would be either composition at the outlet or the temperature inside the reactor.

This will be an inferential control strategy if I want to use temperature to infer the composition so that temperature is one controlled variable. Additionally, to make sure this reactor does not overflow or does not drain down you would also have volume or level control. So you will also have a level of control into this particular process, and then in terms of manipulated inputs, you can have the coolant or the hot fluid flow rate.

If it is a coolant, then you will be putting in cooling water here, and you will come out here. Your manipulated inputs can be your cooling water flow and the outlet flow, so this is F_{out}. You can see that even for a simple system like a CSTR you will have multiple controlled variables and you will have multiple manipulated inputs and what you can realize is that if I change some of these things, let us say if I change volume or level inside the reactor, F_{out} which is the outlet flow rate of this reactor would change and when it changes it is ultimately going to affect the purity or temperature inside the reactor. When you have these 2 controllers they will also have some sort of interactions within them.

For example this F_{out} which is a manipulated input affects both the level as well as the composition or temperature inside a reactor. So that is why in real process what you really end up having a MIMO which is multiple input, multiple output system. So let us take another example here.

(Refer Slide Time: 06:13)

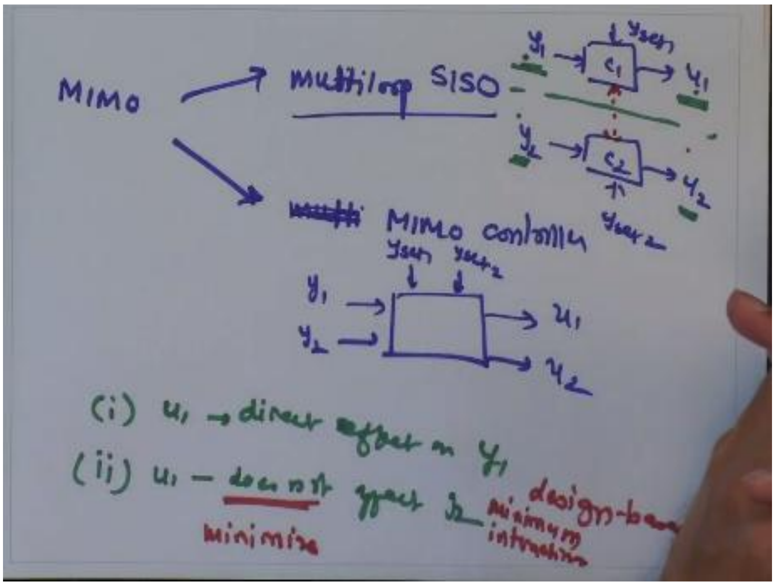


So here what I am showing is an in-line blending. You have 2 pure streams, with some component A and some component B and they are mixed together to achieve a certain final blend quality which is known as a final composition and a certain flow rate of that particular stream. There are 2 controlled variables. One is I want to control the final composition and I also want to control the final flow rate.

And the 2 variables which you have here are the flow rates of these 2 streams. So you can see that this is a MIMO system, there are multiple inputs and multiple outputs. In order to control these types of system there are 2 broad categories of control structure which you can implement. What I am showing you is that I have dedicated one manipulated input to control one particular output.

For example, the way it is shown here I am controlling the composition, this AIC is a composition controller. So composition is controlled by using this stream W1 and the flow rate is controlled by using another stream which is W2. So I am dedicating one manipulated input to one controlled variable when you do this, this is a known as a multiloop SISO strategy.

(Refer Slide Time: 07:34)



So if I want to do MIMO control, there are 2 ways. One is I can dedicate one input to each output and I can have a control loop around it. So you will have multiple such control loops depending on how many control objectives you have and that is a known as a multiloop SISO strategy or I can have a single controller which sort of has 2 controlled objectives like maintaining W as well as and simultaneously it will change W1 and W2. So that is known as a MIMO controller. Real multi input or multi output controller which will be something like this. So it will take in both the inputs, both the controlled variable, it will take in both the set values and then accordingly will give you the 2 inputs.

In terms of a SISO strategy what you have is, you will have one controller c1, which is dedicated to y1 and you have a second controller c2 dedicated to u2. So that is the fundamental difference between multi-loop SISO strategy and the fully MIMO control strategy. So here I am showing you what happens if you have a multiloop SISO control strategy here.

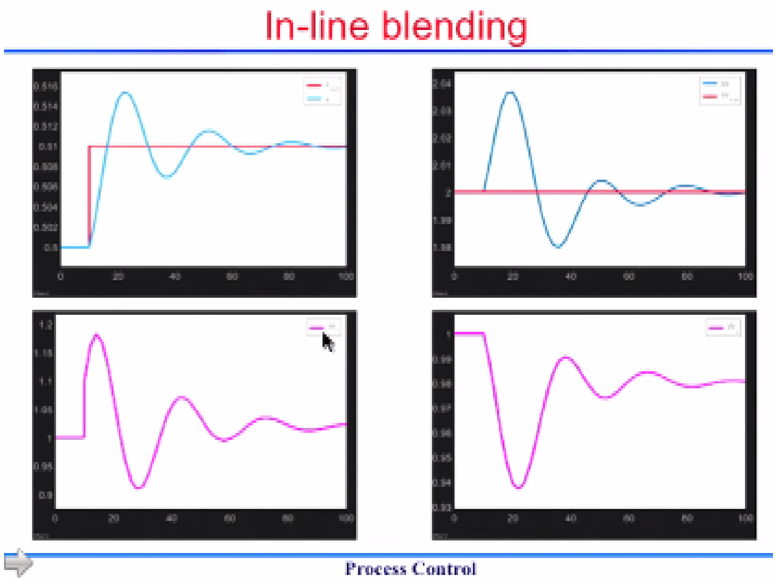
Let us say that this process is at steady state right now and the customer demands that there is some sort of a setpoint change which has to be done. So currently you are operating at let say 0.5 fraction and then you want to change it to 0.51, because some specification has changed downstream. Because of that this AIC controller will get triggered because now you want to increase the purity of this stream.

You would want to increase the amount of W1 which goes into this particular blend. So this AIC will increase W1, but W2 was kept as a steady state value. Now the sum of W1 and W2 goes beyond W. So this second controller which was a flow controller will try to cut down W2. As it reduces W2, W will be maintained.

But as you have put in less amount of W2 and more of W1 your x will again increase. So what you will see is that these 2 controllers will try to catch each other. So every time one controller satisfies its objective the other controller gets disturbed and eventually they will have to find what is known as the sweet spot where the addition of W1 and W2 would still be equal to W which is a set value, but the ratio of these two will ensure that x is also maintained.

So let me show you what happens or how these responses look like if I simulate this particular system.

(Refer Slide Time: 11:02)



So here are the simulations results, the top 2 curves show the 2 controlled variables and the bottom 2 show the 2 manipulated variables. So you can see that I had made a small setpoint change in terms of x, so because of that your W1 increased. As W1 increased your W also increase so the other controller started reducing W2 and the first controller increases W1, but while doing that you can see that some significant oscillations are introduced into the system.

What I am showing you is a final optimized controller strategy. If I use or change the controller parameters it is also possible that these 2 controlled variables do not converge to the setpoint value, but the oscillations start increasing, there are growing oscillations for certain values of control parameters. Instability occurs when you have such kind of interacting SISO control loops.

So when you want to do such kind of multi-loop SISO control strategy it is easy to implement, but very difficult to design. First and foremost, you have to dedicate one input to one output. That way I have shown here if I have 2 controlled variables and 2 manipulated inputs, I want to decide first that u1 will be decided based on y1 and u2 will be decided based on y2.

So when you want to do that there are certain factors which you want to incorporate. One is that u1 should have direct effect on y1, because if I am dedicating a particular input to control a particular controlled variable there should be a direct effect so that this particular controller will be fast and at the same time what you would want is u1to not affect y2 or any other outputs.

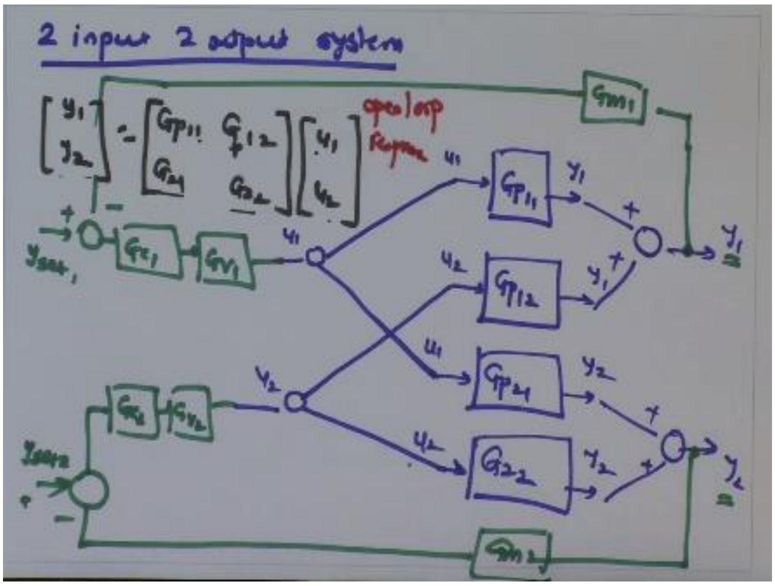
Now this is very idealistic assumption and if that was the case then these strategy would work perfectly. But in reality what happens is even you can see from the previous example of in-line blending, when you change u1 it not only affects y1, but it also y2. So what happens is, when you find a variable which has a direct affect on y1 you also want to minimize its effect on the other output.

So rather than having 'does not' you will try to 'minimize' effect on other variables. So that is sort of known as design based on minimum interactions. So you want to minimize interactions between 2 controllers, so ideally you would want that, when your controller one is acting it should not affect controller 2, but as long as this u1 affects y2 this controller will also get affected.

So when you want to decide these input-output pairs, you want to ensure that the interactions which are happening between these controllers are a minimum. So for that first of all we need to find out how do we quantify these interactions. So there are some ways in terms of quantifying these interactions. So we will try to see how do we go about it.

First we look at a simple 2 inputs 2 outputs MIMO system and we will try to extend the analysis which we have done earlier for a SISO strategy to this multiloop SISO strategy.

(Refer Slide Time: 15:07)



So we will consider a two inputs, two outputs system. Earlier when we had a single input and single output we had a single process transfer function between input and output. Now you have two inputs and two outputs, so you will realize that you require four process transfer functions. So this is Gp_{11}, Gp_{12}, $Gp_{21,}$ and Gp_{22}. So Gp_{11} represents the interaction of u1 on y1, Gp_{12} represents the effect of u2 on y1.

Gp_{21} represents the input of u1 on y2 and GP_{22} represents the effect of u2 on y2 okay. So now here what we have assumed based on this if I want to write the equations what I will get is,

$$\begin{bmatrix} y1 \\ y2 \end{bmatrix} = \begin{bmatrix} \mathrm{Gp11} & \mathrm{Gp12} \\ \mathrm{Gp21} & \mathrm{Gp21} \end{bmatrix} \begin{bmatrix} u1 \\ u2 \end{bmatrix}$$

So your y1=Gp_{11} * u1 + Gp_{12} * u2 because both inputs affect the output and y2= Gp21 * u1+Gp_{22} * u2.

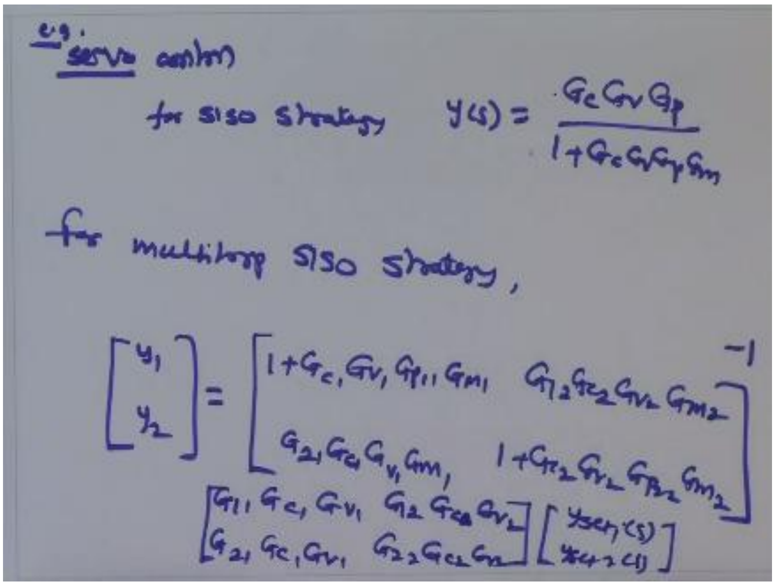
So in this figure if I want to find out what is my final y1 it will be the summation of these two contributions similarly y2 will be the summation of these two contributions. So that will be my open loop representation of this particular system. Now let us see how do we incorporate control into this. So I have said that in terms of control I am going to dedicate y1 to control u1 and I am going to use y2 to manipulate u2.

So let me also represent here u1 and u2, so this u2 is going to go here and here and this u1 is going to go here and here. So all I need to find out is u1 and u2. So let us say you have a controller, so there will be a measurement of y1, and then that will be compared with the set value of y_{set1}. Then you will have a first controller Gc1, you will also have a valve Gv1, and then it will give you u1.

Similarly, you will have a measurement of the second variable; you will have the corresponding y_{set2}, you will have Gc2, Gv2 and then y2. So you can see that your final multiloop SISO strategy will look like this where you have dedicated one input to one controller. This controller deals with y1- u1 and this other controller deals with y2- u2 and you can also see that there are some lines which are going across these two controllers.

That is why it is going to result in some sort of interactions. So for this particular system how do we get the closed loop response.

(Refer to Slide Time: 19:47)



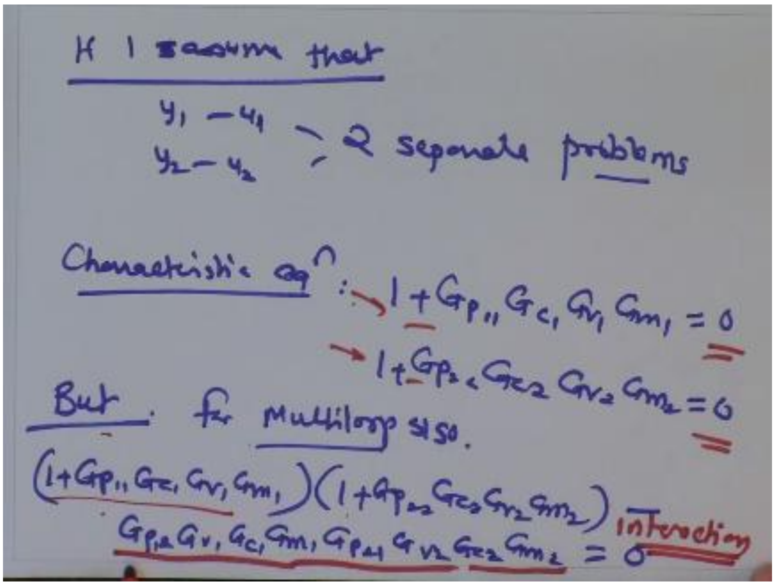
So again we will limit ourselves to only Servo control for example. So for a SISO strategy, what we had was if I want to see how my output in the closed loop changes it was $\frac{\text{Gc Gv Gp}}{1+\text{Gc Gv Gp Gm}}$. For multiloop SISO strategy you can simply extend it in the same way.

So what you get is

$$\begin{bmatrix} y1 \\ y2 \end{bmatrix} = \begin{bmatrix} 1 + \text{Gc1 Gv1 Gp11 Gm1} & \text{G12 Gc2 Gv2 Gm2} \\ \text{G21 Gc1 Gv1 Gm1} & 1 + \text{Gc2 Gv2 Gp22 Gm2} \end{bmatrix}^{-1} \begin{bmatrix} \text{G11 Gc1 Gv1} & \text{G12 Gc2 Gv2} \\ \text{G21 Gc1 Gv1} & \text{G22 Gc2 Gv2} \end{bmatrix} \begin{bmatrix} y_{set}1(s) \\ y_{set}2(s) \end{bmatrix}$$

So you can see that is simply an extended version of the same formula. And now let us see the interesting thing.

(Refer to Slide Time: 22:19)



So when you had two single loops, if I assume that y1-u1 and y2-u2 were two separate problems, for stability of this system you would want to solve the characteristic equation which will give me the stability of that system. So for the first thing, it will be 1+ Gp1 Gc1 Gv1 Gm1=0, and for the second controller, it will be 1+Gp22 Gc2, Gv2 Gm2 =0. So if these were two separate problems, then that is what you would get as a characteristics equation.

For multi-loop SISO strategy there is two more transfer function. The characteristic equation is given by the determinant of this first matrix. When you expand this in terms of the cofactor expansion you will realize that, in the denominator, it will be the determinant of this matrix, so that is the characteristic equation for this system.

So characteristic equation is given by the determinant of that particular matrix, and it comes out to be,

$$(1+G_{p11} G_{c1} G_{v1} G_{m1})*(1+G_{p22} G_{c2} G_{v2} G_{m2}) - G_{p12} G_{v1} G_{c1} G_{m1} * G_{p21} G_{v2} G_{c2} G_{m2} = 0$$

So what you can realize is that this particular characteristic equation, its solution is different than considering these two solutions independently.

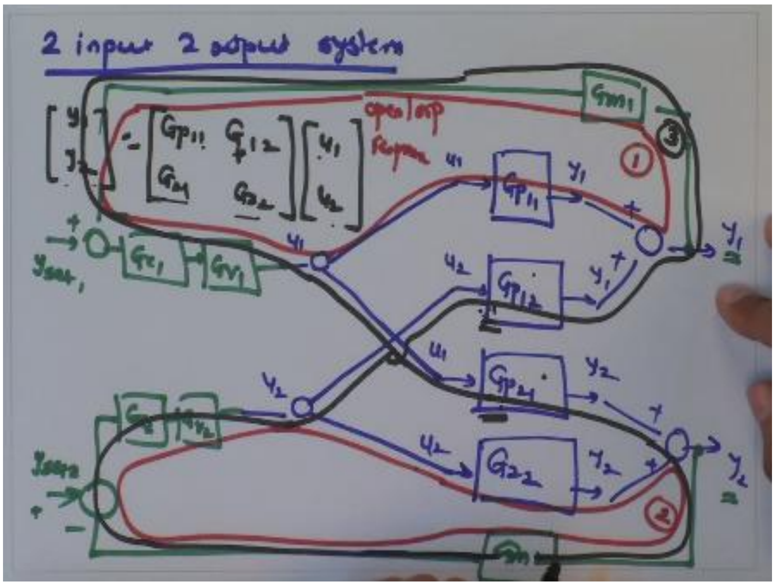
So if I assume that these two controllers are operating independently that is wrong because if I design these controllers by assuming these 2 controllers to be separate what I end up having is a characteristic equation which satisfies these, but the real characteristic equation also has this additional term. So this is known as the interaction term. So this term is going to cause the interactions into this system.

If this term is 0 then automatically the characteristic equation comes out to be this multiplied by this equal to 0. So automatically it tells me that the two characteristics equations are the same as assuming that the original system can be thought of as two independent loops. So you can see that this particular extra term is going to cause the characteristic equation to deviate from the fundamental characteristics equations of 2 separate control loops.

And that sort of quantifies the interaction in this system. So where does this term come from so let me show it to you through the same diagram which we had seen earlier? So if you recollect what we had talked about earlier where we generally get this characteristic equation from is from the closed loop behavior that one comes because you have negative feedback. So every time there is a closed loop you will get one term of that corresponding closed- loop characteristic equation.

If I ask you a question about how many loops are there in this particular figure that will give you an answer about the interactions.

(Refer to Slide Time: 26:39)



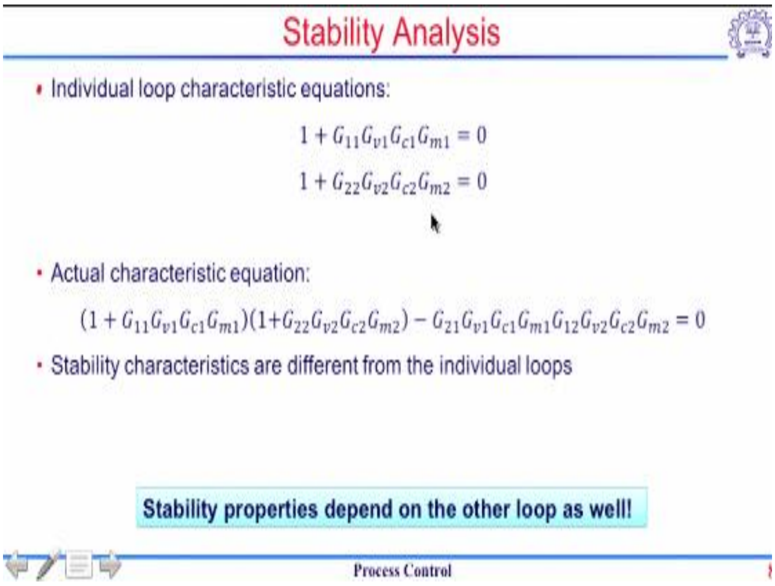
So let us say I want to find out the loops one is an obvious loop let us say I start with this u1 and I move this way. You can find that this is loop number 1. Similarly, I can start with u2, and I can find my loop 2. So if these were the only two loops in the system, then the characteristic equation would have been the characteristic equation of this loop multiplied by this=0.

And that was the case I told you about when we did not have that interaction term. So where does this interaction come from? For that, you have to look at the third loop so let me just trace it down for you. So let us say I can start with u1, so I move it this way then I go to y2 and then from here I have to go here, and you will see that I will finally close the loop this way. So there is an additional loop which is a third loop which causes that interaction term.

And this loop goes through the transfer function like Gp21 and this Gp12. So as long as you have this transfer function known as cross transfer functions, if any one of those transfer function is non-zero, your two control loops will interact. Now note that this loop will not be formed if either of them was equal to zero. So you do not necessarily need that both these cross transfer functions should be there.

Even though one of them is zero, then you would not have this third loop, and then your interaction in the system would go to 0. So this is the third loop which gives you the interaction between the two control loops, and it will be quantified by that big term which I had in that closed loop representation.

(Refer to Slide Time: 28:52)



So what take-home message I want to give you from this is that when you have this multiloop SISO controllers for a system which has multiple inputs and multiple outputs. These are easier to implement, but what happens is the stability of this system depends on the other loop. So if I want to design a controller for this u1 and y1 pair I also have to look at how the other controller is tuned.

So that interaction decides the final stability of the system and as I said for the previous example of in-line blending what happened was, if I change one of the control loop parameters the same stable system can also become unstable in terms of both the controllers. So the controller properties of one controller affect the performance of the other controller.

So that is why it is important to make sure that ,the interactions between the loops are minimized. So we will focus on that after the break. So thank you.