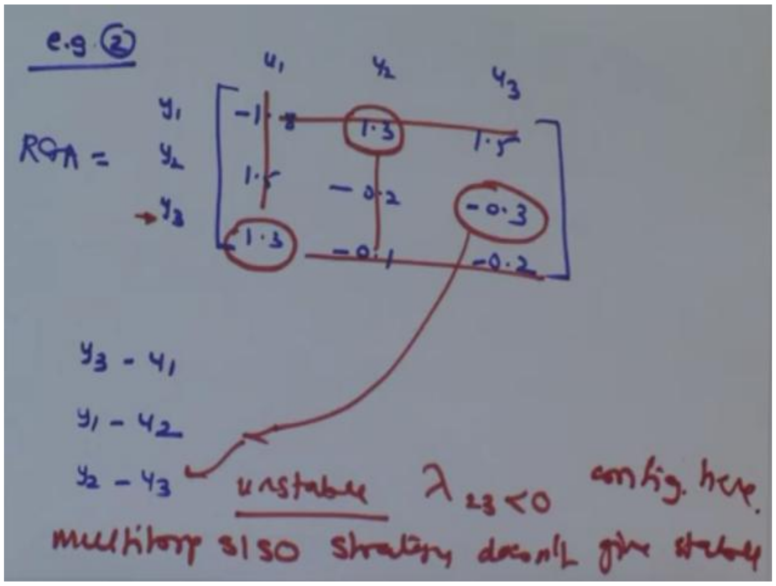
Chemical Process Control

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

Lecture - 45
Tuning of Multi-Loop SISO Controller

Welcome back. So we have been looking at how do we pair inputs and outputs by using relative gain values, and we saw one of the examples, wherein we use these relative gain values to select a pairing with minimum interaction or excluding the pairings which do not make sense and let us continue that with another example.

(Refer to Slide Time: 01:00)


e.g ②
RGA =
$\begin{array}{c|ccc} & u_1 & u_2 & u_3 \\ y_1 & -1.8 & 1.3 & 1.5 \\ y_2 & 1.5 & -0.2 & -0.3 \\ y_3 & 1.3 & -0.1 & -0.2 \end{array}$
$y_3 - u_1$
$y_1 - u_2$
$y_2 - u_3$
unstable $\lambda_{23} < 0$ config. here.
multiloop SISO strategy doesn't give stability

So let us take another example of a 3×3 system. We have y1, y2 and y3 as the controlled variables and available inputs are u1, u2 and u3 and the corresponding relative gain array are given by this. So let us now again use the same principle and try to find out, how do we pair. So you can note that we can start with y3 and the corresponding value which is closer to 1 is 1.3, the other 2 values are negative. So you can do a pairing between y3 and u1 and correspondingly you will get another 2×2 system as this.

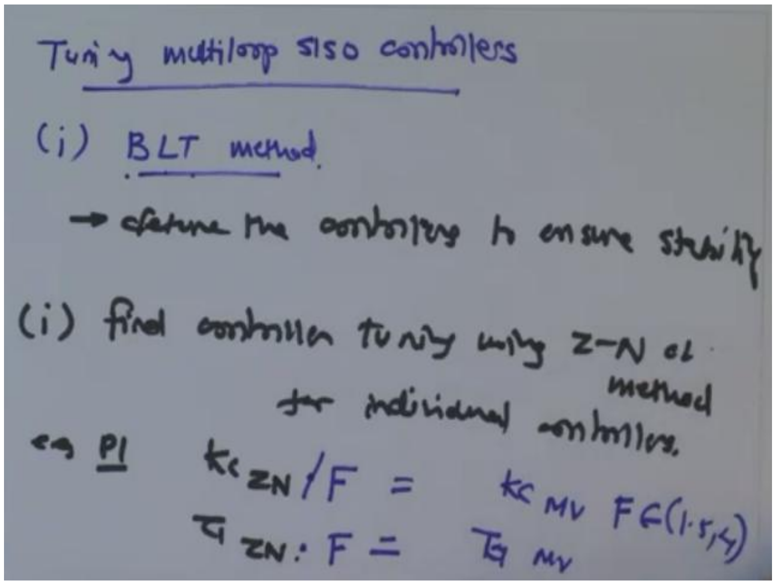
The next pairing which you can do is y1 and u2, out of these four entries, this one (1.3) is closer to 1. So accordingly u2 and y1 are paired. Now what you end up with is y2 and u3, and this is unstable because the $\lambda_{2,3}$ is negative. What this example suggests is that it is not always possible to use relative gain values and find out a multi-loop SISO configuration.

Here what you are seeing is irrespective of whatever you do, you will not be able to get three pairs for which you have all λ's to be positive. So the conclusion here you can draw is multi-loop SISO strategy does not give stable configuration here. So what do we do in that case, we go with a full-blown multi variable controller, as I had said earlier. Those are little difficult to design, but here there is no choice but to go for a full multi variable controller.

So that is why I included this example to tell you a point that it is not always possible that for any MIMO system you would be able to get a configuration of multi-loop SISO systems. Let us now see we have now done the pairing between inputs and outputs by using Niederlinski Index and then followed by the relative gain array. How do we go about really tuning these controllers? We have seen during the controller tuning lectures.

We saw that there are different methods like Heuristic methods or Direct synthesis methods by which you can tune these controllers. But those were all limited to a single input-single output or SISO control strategy. Now when you have no interactions in the system, then automatically you can say that those same controller tunings can be carried forward for these MIMO systems as well; however, if there are interactions, then the stability properties of the entire system are much different than individual SISO controllers, because of these severe the interactions, more are the challenges in terms of ensuring stability.

(Refer to Slide Time: 05:13)



Tuning multiloop SISO controllers

(i) BLT method.

→ detune the controllers to ensure stability

(i) find controller tuning using Z-N CL method for individual controllers.

eg PI $k_{c\,ZN}/F = k_{c\,MV}$ $F \in (1.5, 4)$

$\tau_{I\,ZN} \cdot F = \tau_{I\,MV}$

So one of the simpler methods of tuning these multivariable controllers is known as a BLT tuning method. It stands for Biggest Log Modulus tuning method. We will not go into the detail of this method, but I will explain what is the philosophy behind this method. The basic principle of this method is that whenever you have interactions in a system, then whatever controller tuning you can find by considering these controllers to be independent, you cannot work with those tunings. You have to back off from those tunings in terms of reducing the gain or increasing the time constant so that whatever stability margin which you had used for your individual controllers you would want to have a figure of safety on top of those tuning. So your controllers will be detuned, or they will be reduced in terms of their aggressiveness whenever you have interactions in the system. That detuning factor is given by F.

So what they say is, you detune the controllers to ensure stability and the method is in step 1. You find controller tuning using Ziegler Nichols closed-loop method for individual controllers and while doing so, make use of the relative gain values. So if the relative gain for that particular input-output pair is positive, then while conducting this Ziegler Nichols closed-loop method, you will be closing the corresponding input-output loop. But the other loops would be kept open that would be the case when $\lambda_{i,j}$ is positive, and if the corresponding $\lambda_{i,j}$ is <1, then again you will close the corresponding loop for which you are finding, so you will close that loop as well as all the other loops while conducting the Ziegler-Nichols testing.

Let us say for example if you are using a PI controller; you will get $Kc_{\text{Ziegler Nichols}}$ and a corresponding $\tau_{\text{I, Ziegler-Nichols}}$ by using the formulas. For a multivariable system, the $Kc_{\text{multivariable}}$ and $\tau_{\text{I multivariable}}$, are obtained by dividing this gain $Kc_{\text{Ziegler Nichols}}$ by safety factor F, and you will multiply the $\tau_{\text{I, Ziegler-Nichols}}$ by safety factor F.

F stands for a safety factor or a detuning factor. Whatever gain you get, you would reduce the gain by this factor and whatever time constant you obtain, you will increase it by F and the value used is between 1.5 and 4. So if the interactions are less, then you will use a value of 1.5, and if the interactions are severe, you will use a value close to 4. There are books on Biggest Log Modulus tuning method, which explain the way to find out the value of F. We will not go into the details but if you want to use Heuristics, then what you can do is, you can select the value of F between 1.5 and 4 depending on the value of the corresponding relative gain. If that gain is closer to 1, you will use F value close to 1.5.

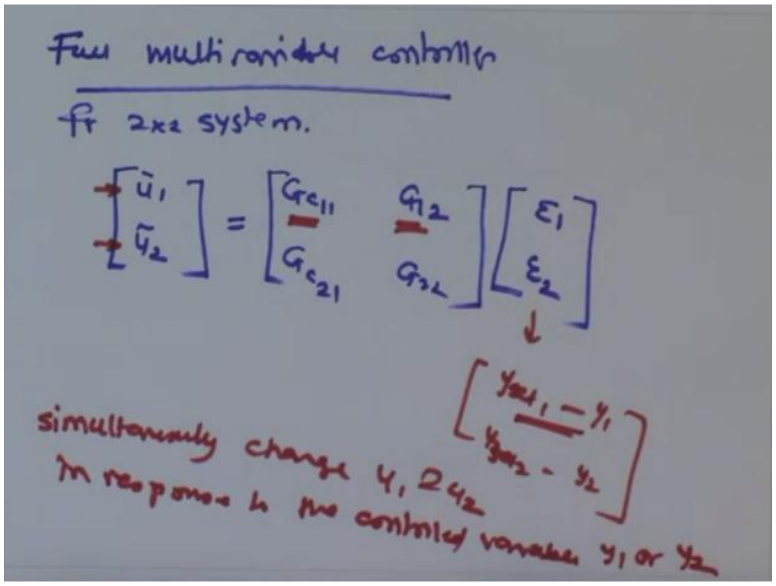
If the relative gain is deviating from 1, then you will use the detuning factor close to 4. So that is how these individual controllers within the multi-loop SISO architecture would be tuned, and so far we have looked at how do we go about selecting this input-output pairs by using relative gain array method, as well as Niederlinski index and both these methods, are steady-state method.

Their advantage is that you require minimum process information to decide or to conclude about these pairings. However, as these are only steady-state methods, the main limitation is that the analysis is valid only for steady state and by ignoring the dynamics a lot of times the pairings which you come up with may not have favorable dynamics. What I mean by that is the controller pairings which you are choosing may be very slow dynamically.

And therefore the interactions can become significant and then the corresponding performance would be poor. That is why if we look at recent literature, there will be a lot of work done on incorporating dynamics while finding out these tunings. So I will try to close this session by talking about the example, which we considered at the beginning of this particular video, which was the case where multi-loop SISO control strategy was not possible.

And in that case what we go with is a full-blown controller, so a real multivariable controller.

(Refer to Slide Time: 11:27)



Full multivariable controller

for 2x2 system.

$$\begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{bmatrix} = \begin{bmatrix} G_{c_{11}} & G_{12} \\ G_{c_{21}} & G_{22} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix}$$

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} \downarrow \begin{bmatrix} y_{set_1} - y_1 \\ y_{set_2} - y_2 \end{bmatrix}$$

simultaneously change u_1 & u_2 in response to the controlled variable y_1 or y_2

So here you do not need to pair any input with the output. Here, for a 2×2 system, you can compute the two manipulated inputs by using four transfer functions. So you will have G_{C11}, G_{C12}, G_{C21}, and G_{C22}, which will take error from output 1 and output 2. So this is $y_{set1} - y_1$ and $y_{set2} - y_2$. So what it essentially tells me is that if I have an error or I have to control this y_1 at set point 1, it is not only going to change u_1 alone but u_2 as well.

So simultaneously change u_1 and u_2 in response to the controlled variable y_1 or y_2. So here you are taking simultaneous decisions and that sort of automatically incorporates if there are any interactions, then the controllers would behave accordingly. The only limitation of this part is that now instead of 2 controllers which earlier we would have used in case of multi-loop SISO controllers, we will be using four now.

Let us say, if you have a PID controller you would have six parameters to get Kc, τ_I, τ_D for controller 1 and Kc, τ_I, τ_D for controller 2. Now the number of parameters which you have to get have almost doubled. So now you have to get 4 of these controller transfer functions, and if all of those are PID, then you will have 12 such parameters to be tuned, and this is just for example for 2 × 2 system.

As you go on for 3×3 or 4×4, you will realize the number of those controller transfer function would increase substantially, and therefore the tuning becomes a very difficult problem. That is why a lot of times when you talk about industrial implementation at least at the regulatory control level, you will see that as far as possible, you would try to go with a multi-loop SISO control strategy.

Only when it is not possible to get such kind of a pairing, then you would have a multivariable controller something like this, but again this will be done only for a subsystem of the problem where you get a non-feasible pairing. So even if your original system is 10×10, if 8 of those input-output pairings can be assigned, then only for the remaining 2×2 system, you will use a full-blown much variable controller.

So that kind of simplifies the problem and as we had seen in the previous week about these advanced controllers like a DMC or MPC. Those high-level controllers can incorporate or can compensate for these interactions, and they would be able to make the corresponding moves. They will give the setpoint changes such that those interactions are accounted for. So that is how in a real system of multivariable control.

You will try to have a multi-loop SISO controller at a regulatory level and followed by an advanced controller which would take care of those interactions and make the necessary moves. So we will stop here, and in the next part of this week, we will focus on how do we go about controlling a batch process. Thank you.