Course Name: INTELLIGENT FEEDBACK AND CONTROL

Professor Name: Leena Vachhani

Department Name: Multi-disciplinary primarily for Mechanical, Electrical, Aerospace and Chemical engineering streams

Institute Name: Indian Institute of Technology Bombay

Week - 03

Lecture - 15

Hi. So, in our in my last video, we saw the MIMO structure, the transfer function matrix form of the MIMO structure, where we considered these M inputs and N outputs connected through the different transfer functions. And since it is LTI system, we said it is the output YI can be written as a linear function, linear combination of their inputs. At the same time, we figured out an important variable pairing problem in order to consider the MVSISO solutions. And this variable pairing problem is to say clearly as which output YI should be controlled with the help of the manipulated variable Uj.

In order to find this particular pairing let us consider two input two output system and its open loop relationship as we have said in the matrix form instead of matrix form now I am saying I am just mentioned this in terms of the linear combination form. So, y1 is said in terms of u1 and u2, similarly output y2 is said in terms of u1 and u2. And its block diagram can be seen this way, where your output Y1 is being affected by U1 and U2. G11, U1 is plus G12, U2. So observe this particular structure.

So far, we were looking into single input and single output system. If Y1 is U1, then it's only G11 that transfer function that we have been talking about. But since now, if I want to consider as a multiple variable SISO system, then even if I'm considering Y1 being controlled with the help of U1, there is an interactive term G12 or G21 for Y2 playing certain role over here. All right. So now let us consider the control loop problems.

Let us consider pairing between Y1 with U1 and Y2 with U2. So, what happens is if I consider a control loop between Y1 and U1, the controller GC1 is to be designed.

Similarly, between Y2 to U2, controller transfer function GC2 is playing a role here. Now, let us study the effect of U1 over Y1. Okay, so under the condition when the loop 2 which is between Y2 and U2 is open and when it is closed.

Let us consider the case when loop 2 is open. If loop 2 is open then Y1 is given by G11 U1 because loop 2 is closed, so Y2 is being governed by directly by U2. If loop 2 is open, then Y2 is directly governed by U2 and therefore, there is no interactive term that is going to play for Y1. So, the transfer function between Y1 and U1 is going to be directly Y1 is equal to G11 times U1. All right.

And Y2 is going to be definitely G21 by U1. But since it is open, so I am not going to take account of this. I am interested in the transfer function between Y1 and U1 as of now. All right. So, then if I have the loop 2 is closed now.

In that condition, my loop 2 is closed. So, now this particular loop is going to create something in getting into Y1 here. So, in that case. What is the transfer function between Y1 and U1 is what we have to figure out because this particular loop is getting closed. So, you can see that this particular adder part, this particular signal is having some loop being closed over here.

When the loop was open, this particular signal was getting affected only with the change in the U2. But now when the loop is closed, then changes into the U1 is also getting affected because this particular loop is closed, and that is approaching Y1. And this particular transfer function we can figure out because this is a linear system again. So, considering that other input which is your U2 input is not there, then the transfer function can be written in the form of this between Y1 and U1 which turns out to be having this term. which is as we have seen it, it turns out to be that it is dependent on this GC2, which is the controller transfer function of control loop 2, right?

So, you see that the control loop 2, as soon as it gets closed is affecting the output y1 through this transfer function. which is primarily driven by G12 and G21, which are the two interactive transfer functions over here. All right. Fair enough. So, what we have here is we can call these particular effective gain G11 effective which is changing the original

G11 transfer function by this particular transfer function after designing the controller for the control loop 2.

Okay. So, now controller GC1 must use this particular transfer function G11 for the controller design. Alright, if that is the case then the presence of this g12 and g21 need to be noticed here all right and at the same time one has to make sure that irrespective of the loop 2 is closed or loop 2 is open, the controller GC1 should be something that I design it and is able to do the control objectives between Y1 and U1. So, it turns out that our aim here is to have a method to determine the relationship between U1 and Y1 with loop 2 closed without knowing GC2 because now this becomes a chicken and egg problem. We are designing the controllers GC1 and GC2.

When I am designing GC1, GC2 is playing a role. Similarly, in a vice versa, when I am designing the GC2, GC1 will play a role. So, now without knowing GC2, how do I consider this particular relationship Y1 and U1 to be known so that I am able to design the GC1 and GC2 independent of each other. Let's understand this relationship at a limiting case, which is the steady state right now. What we have at steady state?

At steady state, S is tending to zero, which is the frequency term turns to be zero. Now, assuming that the integral action is there for the controller GC2, there is a background behind considering the integral action for the controller gain GC2, because I am considering the steady state now. Now, at steady state, since our controllers are proportional integral and derivative terms, now even if I have the proportional term in the controller, the integral term is the most effective or is the most responsible term in the steady state. Since I am doing the steady state analysis, here I am assuming that I am considering only the integral action for the controller GC2. And therefore, if when S tends to 0, I can consider GC2 tends to infinity.

And now, if that is the case, then the G11 effective at S is equal to 0 turns out to be this. And as soon as I consider GC2 is turning to be the gain of the controller of control loop 2 is very large, then in that case, I can write G11 effective as G11 minus G12, G21 upon G22. So, this way I would be able to design GC1 irrespective of GC2. If I consider this particular gain term, then I will be able to design the gains for the controller 1. As a different perspective, I can consider this steady state relationship as just simply the gains of Y1.

For Y1, I can write K11U1 plus K12U2 and Y2SK21U1 plus K22U2. So these, the controller job is to design this K11, K12, K21 and K22. Just keep a note of this as of now. All right, so let's come up with a particular method called relative gain array, which is using these effective gains and so on in order to find the variable pairing problem. This was rather developed by Ed Bristol, a control engineer from Foxboro, and he developed this as a heuristic technique.

It predicts the interaction between control loops when multiple SISO loops are used. This is what we want because only if these interactions are minimal, then only I'll be able to use multiple SISO method. He designed this particular relative gain lambda ij between input uj means I am saying uj and the output i which is yi and is defined by the lambda ij which is gain between input j and output i with all other loops open. Similarly, it is a ratio between these two gains and the second gain, the denominator gain is between the same input j and output i with all other loops closed. So, if this particular gain value is changing significantly when the other loops are open or closed, then this is not a good pairing option.

But if these gain values are almost same, then there is a chance of considering this variable pair between output Yi and input Uj. All right. How do I calculate this now? heuristically, this ratio is giving me a very nice idea that, okay, these two gains should be almost same, and that is why lambda ij should be almost equal to 1, then the variable pair yi and uj is good to consider. To calculate this particular gain term, we say that okay, when I am considering again this two input two output system, the output y1 and input u1, the gain between the two is given by the partial derivative dou y1 by dou u1 when u2 is constant.

Because by this particular control loop 2 is now open. If the loop 2 is open means u2 is constant. So there I get G11 of zero, which is K11. When, you remember, we have K11 given by, is said as Y1, Y2, K11, K12, K21 at steady state. U1 and U2, all right?

So this is how the K11 is coming up here. Similarly, when I have to consider control loop, all the control loops to be closed, then the output, the gain between output y1 and u1 should be considered when y2 is constant. So this y2 is constant gives you g11 effective of zero. And this is what we had considered. If I consider it by the method by g11 effective way of finding, which is going to give you g11, we found this value as g11 minus something, all right?

But in general, when I have to calculate, if it is multiple input and multiple output system, then it is easier to find this with this particular formula, which is lambda ij given by dou y1 by dou uj, when all the ui's are 0. And of course, i is not equal to j, all the other ui's are 0, except uj. And this particular gain, when all the loops are closed, are found when yj is equal to 0. Means the output, all the other outputs are 0 except yi. All right.

So now here comes the matrix of values, lambda ij's, which is like, for example, it's a two input, two output case. So I will get lambda11, lambda12, lambda21, lambda22, which is this defines the variable pairing between Y1, U1, and this one is Y1 with U2. This is Y2 with U1, and this defines Y2 with U2. All right. So this is what my lambda matrix is about.

What is the significance of this RGA? We have this instead of single element, now I am considering this as a matrix. And what should be the value of these RGA elements? To certain extent, we said in order to have the variable pairing, a nice variable pairing means that particular lambda ij value is almost equal to 1. If that particular value is close to 1, then the gain when the other loops are open or closed are almost same, and that's the value we are looking forward.

So, for a two-input, two-output system, as we said, the elements can be returned in the matrix form, lambda11, lambda12, lambda21, and lambda22. Let's see what are the properties of the lambdaij Because we are pairing between the y1 is either controlled by u1 or u2. So, there the row elements, the sum of the row elements is going to be 1. Similarly, sum of the column elements are going to be 1.

All right. So, each row will sum to 1 and on each column will sum to 1. So in case of the two input two output system I can write my RGA matrix like this by identifying only one element lambda11. Because now if lambda11 is given, this value becomes 1 minus lambda11. This particular column should also be 1.

So, this value becomes 1 minus lambda11. Therefore, this element lambda22 is equal to lambda11. All right. So, our job for two input two output system reduces it to finding just one element of the RGA matrix which is lambda11 or any other any one element of the RGA matrix. Coming back to the juice blending problem, which we discussed in the previous video, where we had two flow streams.

U1 is the 40% juice stream. U2 is the water stream. And Y2 is the flow control. Y1 is my composition control. Let's consider that Z1 is the volume fraction of juice in stream 1, which was 40% in our case.

And Z is the volume fraction of juice in the blend stream Y1, which is 30% that is desired. At the same time, I already know that the flow rate is F1 plus F2, which is given by Y2. And now we should be able to frame up our dynamics here in terms of saying that, OK, since no juice concentration is there in the water flow, the composition of F times Z, which is the composition at the output side is equal to nothing but the composition at F1 and Z1 given by only F1, Z1 which is coming from the U1 side of it. So if this is what is the problem, and we have this F1 flow is given by 3 GPM, and F2 is given by 1 GPM, your problem is to consider Z1 as, we already said it is 40% juice value, juice concentration, which is Z1 is 0.4 mole fraction juice. All right.

So we will be able to frame our problem in terms of getting the transfer functions values G11, G12 and G21, G12, G21 and G22. Frame this up in terms of the values given here and then find out what is lambda11. Try this exercise and you will understand that as soon as you have this lambda one one, lambda one one approximately in this particular. I'm giving this the answer to this lambda one one turns out to be almost near to zero. Very small value point two or point three that comes up, whereas lambda one two or lambda two one turns out to be near to one value.

It means it shows that lambda12 and lambda 21 approximately near to 1 value shows that the variable pairing should be with output y1. We should have control with u2. Similarly, y2 should be paired with u1 and y1 should be paired with u2. If that is what the variable pairing we consider, we would be able to design an MVSISO system. All right, let's understand this.

Let's take a quick review of this relative gain array that is introduced in this video. What we have is RGA matrix that contains the individual relative gains as elements. And these elements are given by lambda IJ. This lambda IJ is calculated as the ratio of gains. Ratio of gains when the loop is open, when the other loops are open and when the other loops are closed.

All right. So if this is what is the M input and N output system, then the sum of each column is 1 in the RGA matrix form. So, it says that lambda ij summation i equals 1 to n, lambda ij is equal to 1. Similarly, sum of each row is 1. At the same time, if I am getting lambda ij equal to negative, it means it is a failure or an unacceptable term that we have.

Negative value of lambdaIJ means what? I have a gain. So it's a ratio of two gains. If the numerator gain is positive, the denominator gain is going to be negative. So changing the sign means what?

It was earlier negative feedback system if I'm designing. The other case, when the loop gets closed, the system becomes positive feedback and it becomes unstable. So negative lambda IEJ is a no-no case. completely reject pair. It gives you an indication that this particular pair option of YI with UJ should not be tried at all.

That's a very good answer that we are getting, by the way. At the same time, if the lambda IJ value is very large, particularly if lambda IJ is very large as compared to the value 1, then we should also not be considering such pairing options because there is a large change in the gain and we should not consider that. Now, let us consider this particular case. If I have G of 0, means the steady state gain is given by something like this, you see very nice 1 terms are turning out, but the rest of the 2 terms are also close to 1. As we see, the corresponding RGA turns out to be giving you 400, 400 values.

Since this is based on the property of the RGA, the sum of rows should be equal to one and the sum of columns should be equal to one, we say the anti-diagonal terms turns out to be minus 399. We said the negative values, it's completely no-no case. So we are ending up in choosing lambda11 and lambda22, means variable pairing option as Y1 with U1 and Y2 with U2 with such large gain. What happens? This particular case, let's understand this with just by simply adding a 5% model error.

So now if I'm adding this model error by 1%, you notice that this particular term G21 of 0 is having 5% error. Instead of 0.95, let's say it has 1. So now the corresponding RGA turns out to be this. This negative term as I as we say is a no-no case, it's a complete no case you saw that if you would have considered a variable pairing with here as y1 with u1 as y1 with u1 and y2 with q2 y2 with u2 if this kind of pairing is considered As soon as I have a very small model error here, what we are getting a negative RGA element here.

As soon as negative RGA element is a complete chance of going into instability. So, the large values of RGA element is also signifying that the system is no longer robust enough. So, let us not try considering the variable pairing, which gives you a very large RGA element. All right. OK, so we in conclusion, if it's a large value or a negative value, it turns out it is sensitive to the model uncertainty.

And we should say that, OK, pairing is not possible. One should use a single loop, which is multiple output with multiple input. Don't try to consider multiple loops or don't try decomposing it into multiple systems here. Large value of relative gain indicates effect of model uncertainty on pairings. If at all you have gone with variable pairing, if at all you have gone with multiple variable SISO ways by introducing multiple control loops, it will result into model uncertainty.

All right, let's consider pairing with large systems. So far we have considered giving you two input, two output in order to get some kind of idea that these RGA ways are a good way of looking at it. Let's take certain examples with large systems. Of course, this itself is still a three input, three output case. So if that is what is the RGA matrix given to you, then the question is which input to be paired with which output.

It turns out we'll have to consider logical reasoning. The logical reasoning means I would be definitely rejecting these pairs, which are kind of negative values. The values which are near to 0 is also rejected. So, this is not the solution we should consider for. So, let us consider now each row-wise.

Now, as soon as I have rejected these two, I am left with only one pair in this particular row, which is my Y3. This is my lambda32, which means the Y3 should be paired with U2. This gets frozen. So, as soon as I say this now I cannot consider the input U2. So, I will have to consider this particular lambda element which is saying that Y1 should be paired with U1 and now I am left with only one choice with Y2 which is Y2 should be paired with U3.

So, such kind of logical reasoning typically gives you a single option. If there are multiple options, then there are other methods to narrow down to a single option. All right. If I have the if I have multiple input and multiple outputs, it may be possible that I'm looking forward for lambda or the RGA matrix to be square matrix. So what we have is, for example, the gain matrix, which is not square, but I want the RGA matrix to be square, then I can consider RGA being formed with the help of this GN element-wise multiplication with G and GN inverse transpose.

For the non-square RGAs, I can consider Penrose inverse here. So for M input and N output, and when M is greater than N, I can consider left Penrose inverse, which is like RGA's input scaling dependent in this case, and other way around. So there are other methods for non-squared RGA matrices. In this case, our objective is to eliminate M minus N inputs. Since I have M greater than N case, M is the number of inputs and N is the number of outputs.

So number of inputs is greater than number of outputs. So, I would like to eliminate m minus n inputs and then work with just the exactly n inputs to be paired with n outputs. So, criteria here in order to reject these or eliminate these m minus n inputs is to consider that the jth input is considered effective if the sum of jth column is large and vice versa. We'll look into this particular procedure now. What we'll do here is we'll consider finding the gain matrix.

We'll calculate the non-square RGA. We'll calculate the sum of each column of the RGA. Now, we will eliminate the inputs that correspond to the smallest sum of the columns. So then we'll just keep those n inputs which are giving the larger sum of the columns values.

And those, after eliminating this, then again one can calculate RGA, which turns out to be n input, n output RGA matrix and look forward for this.

So these heuristic methods to a certain extent have been developed, but at the same time, they are very, very powerful methods because they come from the idea that, okay, I'm dividing, I'm decomposing the system in terms of multiple SISO inputs. At the same time, when I'm doing that, that time, whether the other loops are open or closed, I should not get the very significant change into that particular system performance. Let's take an example here. We'll not look into the system behavior or the working of the system, but let's see what are the inputs and outputs and can we do, and given a particular RGA, will we be able to figure out what input should be paired with what output. So, in this particular polyethylene reactor case, the control variables are reactor temperature T and the ethylene concentration C2, whereas our manipulated variables are superficial velocity of the feed phi, catalyst feed grade Qc and the feed temperature Tf.

Okay, so I said, just take it blindly that this is three input, two output case. And let's see if I can do something in order to understand the RGA part and finding out the variable pairing with the help of RGA, this case. So steady state gain of this particular process is again, you have three input and two output case. So this gives you two cross three matrix form. These are steady state gains, by the way.

And when I compute the non-squared RGA matrix, it turns out that we get this kind of RGA matrix. Again, 2 cross 3. And when I take the sum here, I see that this particular sum is very small. And this suggests that TF should be eliminated. So because TF is perhaps not making much of the control over the rest of the two, the control variable C2 and T over here.

If that is the case, then our problem is corresponding lambdaIJs are still high. And so I cannot consider these variables. So it suggests that C2 should be paired with phi and T should be paired with QC. But these gains are still on the higher side. So should I consider this kind of variable pairing or not?

Let's analyze it in a slightly different way as well. I can consider combination of two inputs with two outputs. So, I will have if I am considering two inputs. So, for example, I am eliminating TF here. So, what I will get is two input two output system QC,phi with C2,T.

Then let us see what the RGA comes out. If I reject element QC, then I have a subsystem S2, which will have two inputs, phi and TF, for the same outputs. And the corresponding RGA is this. If I consider rejecting or eliminating phi here, then I have QC and DF as two inputs and I get the RGA for the corresponding subsystem S3. It's not subsystem, the option S3 as 23 and 23 options.

Now, you see that the difference between if I consider option S1, then the RGA element value is 28, whereas for option S3, I have the RGA value turning out to be 23. I am perhaps not going to consider option S2 because the RGA value is extremely high. So, eliminating QC is not an option at all. All right. So, but it is still, it is least for S3.

So, in the previous case, I was getting S1 as large. Even for S3, it turns out that it is giving the large RGA matrix. So, do we have other methods that can indicate interactions here? Yes, this is another method called condition number, which is developed by Cao and Bliss in 1996, but still being used. So what this says is that the condition number is based on eigenvalues of the gain matrix.

Now, this eigenvalues, if I arrange it in largest to smallest, then condition number is the ratio of the eigenvalue, the largest to smallest eigenvalue. So it also, one can look forward, dividing it into a set of square matrices and getting these condition numbers. For this, we will compute SVD for each set and then we'll find the smallest CN is going to be the most effective one for pairing options. Smaller condition number indicates well-conditioned system. It means the pairing is possible.

All right. So now what we will consider that if I have a square matrix G given by and I have done the SVD decomposition, which gives you U sigma V transpose V transpose and sigma turns out to be my diagonal matrix, which is giving me the eigenvalues from sigma 1 to sigma R. So I know that R is the rank of G transpose G. If the large value of condition number, it says G is ill-conditioned. So, if that is what is the case, I can always consider for getting for non-singular matrix, how do I get it and so on and so forth.

And if condition number is large, CN is large if both G and G inverses have the large inverse, large elements. And that's what, that's the reason we are calling with large condition values, G is ill-conditioned. We had this particular example, and example had three options of systems S1, S2, S3, when we consider elimination of one input value, phi, tf, or c. So in that case, we have the sigma one, sigma two turning out to be these values, and these condition numbers are turning out to be these. Now, this condition number of 123 corresponding to this particular option S1 signifies that it is well conditioned, and we should consider the option S1, which was also one particular case of non-square RGA was also giving the same answer.

The caveat here is that one can always consider when we are dealing with condition numbers, Sorry, this slide. Here, With the idea of a smallest condition number coming up for the option S1, which gives me more heuristic answer saying that, okay, I should use perhaps the option S1 because even in the RGA case, also in this particular case, what we had the RGA values 28 turning out to be in option S1, and 23 in case of option S3, which are not significantly different from each other. So, I still had the option S1 and option S3 both to consider in this case.

But with condition number, now I am certain that I will be going with option S1 because this is giving me the least condition number and lesser changes into the gain will be affecting it. Let's look into one more case. If my gain matrix, which is at steady state, is given by one, zero, 10, and one, for this particular case, the RGA matrix is very nice, one zeros and zero one, shows that the variable pairing that is coming up from the RGA solution is that Y1 should be paired with U1 and Y2 should be paired with U2. But in this case, the condition number turns out to be 101, which is still large. And therefore, what should we consider here?

Here, turns out that if I change K12 by just simply 10% increase, which is 0.1, it gives you some sigma value, which will change this particular RGA matrix significantly. And you can assess what the condition number is helping you to understand in terms of the robustness of the variable pairing. So it turns out that as soon as I consider this as 0.1, this particular gain matrix becomes singular. This corresponds to the systems which are difficult to control because as soon as your gain is 0.1, these are almost giving you the singular matrix, so condition number has no significance now because there's no sigma largest and sigma small here in this case. This is losing the rank here.

So in such cases, the variable pairing is not a good idea and that's why one should resort to the single loop cases, single loop case here. So, the idea here is to understand that RGA along with CN is giving you some of the other answer and this should satisfy you to attend for variable pairing and see the simplified control solutions. So our procedure for getting the condition number is we will arrange the singular values in order of largest to smallest and we'll find the CN. If CN is more than 10, then one or more inputs should be deleted is what we can consider. Let's take this as an example.

Again, this is gain matrix given to you for which the diagonal matrix, when we do the SVD, singular value decomposition turns out to be giving you the largest sigma value is 1.618, whereas smallest value is 0.0097. In this case, now with this particular decomposed diagonal matrix, condition number turns out to be 166.5, which is much larger than 10. Let's take the same example of the gain and see what the RGA matrix gives us. Now, RGA matrix turns out to be giving you certain negative values. If that is what is the case, then as we did the logical reasoning the other time, we will not consider these values.

If we are not considering these values, the negative values, again, here this value is almost equal to 0, so we will reject. We will consider 2.2165, which is Y3 paired with U1. So that's why it is marked with red. Now, if I'm considering this, then I'm left with a choice on for the second row as choosing 0.5407, 0.5407, which is Y2 with U3. And therefore, I'm left with only one choice, Y1 with U2.

Right. If that is the case, then let's consider arranging this particular if if that is what is the case. OK, so this three value, two value and point five value is still OK when the RGA matrix were considered. They are still not almost equal to 1, but I should have selected 1.22 in a real good state. But since I wanted to satisfy all the output variables, I had to choose the variables, the RGA elements, which are giving me at least the positive element values.

Let's consider pairing based on both condition number and RGA. Condition number suggests two variables can be controlled because I had very large. You see that this is this is what is the small. If I consider this as smallest, the condition number turns out to be very large. But if I consider only two values which are close to each other.

So it is saying that, OK, if I consider these two. Only two inputs and two input system, two input, two output system, perhaps I have a better answer. So, let us make two cross two subsystems. If two cross two subsystems is considered, there are three possibilities because we have with two output variables. We have y1, y2, y1, y3 and y2, y3, three options.

So, with respect to y1, y2, again we have three choices u1 u2 u1 u3 or u2 u3, if that is what is the case then what are the condition numbers with respect to these combinations coming up and what are the RGA element corresponding RGA elements that comes up because this is two cross two system again I can create a matrix out of it for example it is 39 so this is 39 this element is definitely 38 this element is 38 and 39 because I know for 2 cross 2 only one element is enough to find the entire matrix. Okay so with condition number and RGA, let's see what is the consideration that we can consider If I have to consider the lowest value of the condition number, I will consider this value or this value. Right. Which is giving me pairing options Y1, Y3 with U1, U2 and or Y2, Y3 with U1, U2.

OK. All right. So corresponding RGA values are almost same. So fair enough. So two options I can explore for getting the multivariable SISO system.

All right. So what we have considered. OK. All right, so in this particular video, we considered getting the MIMO system decomposed into multiple variable SISO systems with the help of RGA and CN method. So RGA and CN can be considered as tools to find out what is the best variable pairing possible.

At the same time, they are also telling that what is not to be explored. That is more important. So RGA and CN definitely tells you that those variable pairing options should not be explored at all. But whatever is coming closer to the desired values will definitely give you answers of satisfying control objectives with the help of simplified controllers, with the help of the multiple control loops and simplified PID control structures. Thank you.