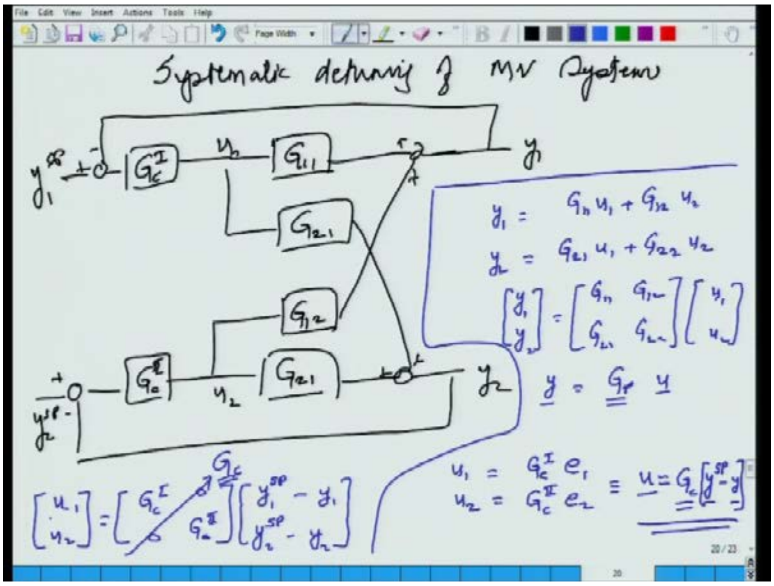
Plantwide Control of Chemical Processes
Prof. Nitin Kaistha
Department of Chemical Engineering
Indian Institute of Technology, Kanpur

Lecture - 8
Multivariable Systems

(Refer Slide Time: 00:20)



Multivariable system, detuning multivariable, systematic detuning of multivariable systems, decentralized system; that means, what it means. Well, so let us say I got a 2 by 2 system. Let us take an example. And man, we have drawn this so many times. I am tired drawing it again and again and again and again.

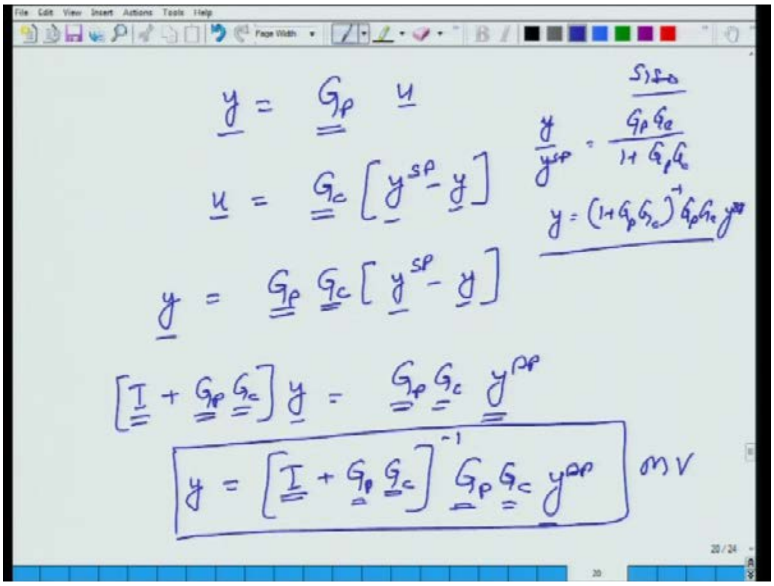
So, if I have a multivariable system, and I am trying to control this guy using a controller, this is my decentralized G c 1. And I am trying to control this guy using another controller, this is G c 2 and of course, effect on 1 of 1, this is u 1, this is y 1, this is u 2, this is y 2, effect on 2 of 1, effect on 1 of 2, effect on 2 of 1. And this is y 1 set point, and this is y 2 set point, of course, plus minus negative feedback plus minus. Let say this is plus plus plus plus plus.

If I have this kind of a system, then let us just do a mathematical; let us just develop the equations of the relationship between y 1 s p and y 1 y 2 and y 2 s p and y 1 y 2. If I well how do we do it, we let us do it here. Let us use this place, this space to do it. Then, what I have is, y 1 is equal to G 1 1 u 1 plus G 1 2 u 2 and y 2 is equal to G 2 1 u 1 plus G 2 2 u 2. In matrix form, I can write y 1 y 2 is equal to G 1 1 G 1 2 G 2 1 G 2 2, u 1 u 2. This is a representation of the same equation, where here I am saying y vector is equal to G p process transfer function times u vector matrix of process transform function.

Now, that we have done this, what is u equal to? You will find that, u 1 is actually equal to G c 1 times e 1. That is actually decentralized control in that input 1 is moved based only on e 1, you do not care about what is happening to error 2, and u 2 is moved based only on error 2 independent of what is happening to error 1, which is basically in matrix form saying u u 1 u 2 is equal to G c 1 G c 2 0 0 e 1 e 2, and e 1 e 2 is actually y 1 set point minus y 1, and this is actually y 2 set point minus y 2.

So, if I look at this equation, let me call this the controller matrix G c and that is I repeat again. The fact that the off diagonal terms in my controller matrix are 0 implies I am doing decentralized control. So, now, what I am saying is that, this is equivalent to u vector being equal to G c matrix, which is a decentralized matrix times y set point minus y, you know this vector. So, well I substitute for u what I have derived here, and if I substitute what I have derived here may be we should do in the next page, insert new page.

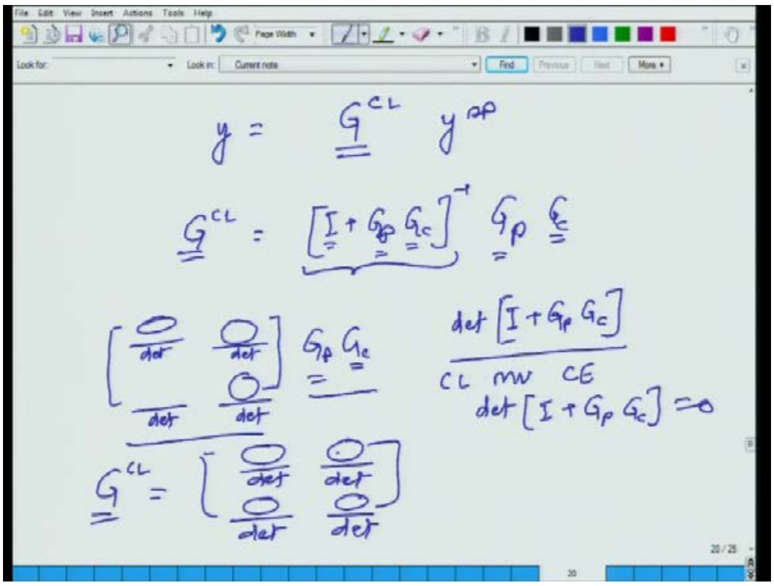
(Refer Slide Time: 05:59)



So, what I had was y is equal to G p matrix times u, and what I had was, u is equal to G c matrix times y set point minus y vectors. When I substitute for u, what I will get is, y is equal to G p G c, and please note here that the order of matrix multiplication is important because matrix multiplication is not commutative, times y set point minus y. And, if I take the y term to the left hand side, what I get is identity matrix plus G p G c matrices, this is also a matrix, times y is equal to G p G c matrix times y set point. And therefore, what I will get is, y is equal to identity matrix plus G p G c inverse times G p G c, of course matrix multiplications times y.

Remember for SISO systems what we had do you see the analogy for SISO systems what we had was, y over y set point was actually G p G c, the individual transfer functions divided by 1 plus G p G c and so, what you actually have is, y is equal to 1 plus G p G c inverse times G p G c times y set point. Do you see the analogy? This is for a SISO system, and this is equation for a multivariable system. But it is the analogy with the SISO system is pretty clear.

(Refer Slide Time: 08:16)



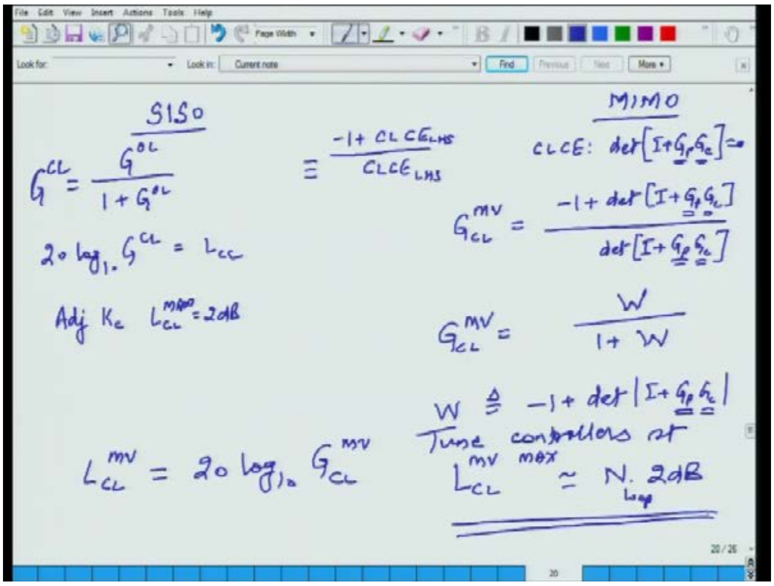
Now, when I am inverting this matrix, so now let me say, my multivariable system is, y is equal to sum matrix G close sum matrix of transfer functions times y set point, and this matrix of transfer functions G close loop is actually equal to I plus G p G c inverse times G p is what I get. Now let us look at this matrix which I am inverting. G p G c would. So, let us say I have a 2 by 2 system. If I have a 2 by 2 system, when let us say I have got I plus G p G c. So, let us say 2 by 2 systems for a 2 by 2 systems, I plus G p G c would be what? The inverse matrix for the 2 by 2 system, that we have would be a 2 by 2 inverse matrix, and this inverse matrix will have 4 terms; 1 2 3 4. And then of course, there will be this inverse matrix times G p G c.

This inverse matrix will have, as the denominator, in the denominator it will have a term which is common to all the. When you invert matrix, what do you do? You calculate the cofactors and divided by determinant of the matrix right. So, what I have is, determinant of this matrix I plus G p G c. This determinant is there in the denominator of all the terms. So, there will be something in the numerator and then, there will be the determinant you will be dividing by the determinant. It will be some numerator. You can do the algebra.

But the point is that, the determinant will come in the denominator. And when you execute the multiplication of this matrix with this, the determinate will remain in the denominator. So, the ultimate matrix that you will get which has 4 terms, will still have the determinant. So, G close loop will have the determinant in the denominator of all the 4 terms which relate y, the response of y to a change in y set point.

So, therefore, this guy the determinant, this determinant occurs in all these transfer functions. In the first transfer function, in you know in all the four transfer functions relating y 1 y 2 to y 1 set point y 2 set point. Therefore, this determinant actually determines the stability of the closed loop multivariable system. Because when I change the controller tuning, the determinant changes, therefore, the denominator changes. And therefore, the roots of the denominator change and as the roots and this denominator, this determinate is common to all the four transfer functions 1 2 3 4 right. So, it is. So, the closed loop multivariable characteristic equation is; determinant of I plus G p G c is equal to 0.

(Refer Slide Time: 12:28)



Now, let us talk about multivariable tuning. How do we do SISO tuning? What is the basis of SISO tuning? SISO tuning is what? You take G open loop divided by 1 plus G open loop. Define this as G closed loop. And then what you do is, 20 log on base 10 of G closed loop. you define this as L C L or L close loop and then what you basically do is, adjust K c such that L C L max is equal to 2 d B. That is what we do right?

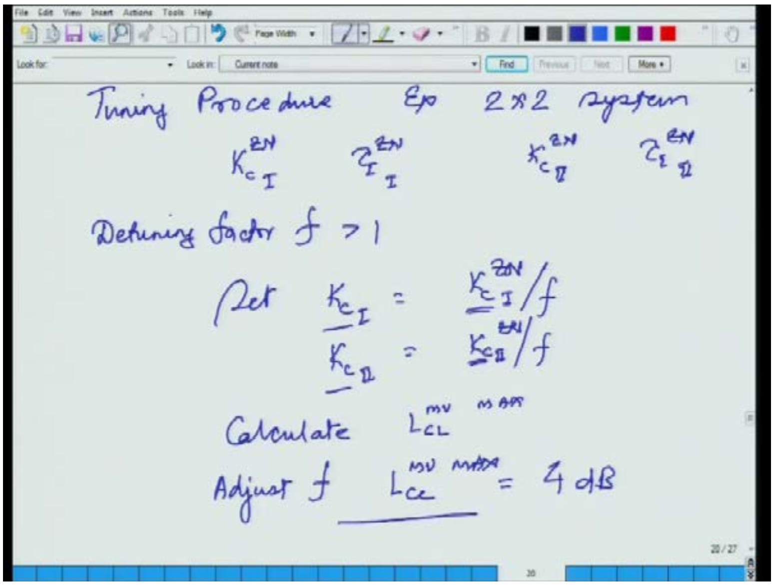
Now, if I take a MIMO system, before I go to the MIMO system, before I go to the multivariable system, please note by analogy, this G C L is nothing but closed loop characteristic equation minus 1 plus closed loop characteristic equation. See the closed loop characteristics equation is 1 plus G open loop is equal to 0 0 minus 1 plus closed loop characteristic equation divided by closed loop characteristic equation. Or L H S, left hand side of a closed loop characteristic equation. You can see that, the log modulus whose hump I am trying to make sure does not is 2 decibels. That log modulus is coming from closed loop characteristic equation, left hand side of the closed loop characteristic equation minus 1, that is the numerator and the denominator is the left hand side of the closed loop characteristic equation.

Now, I by analogy I define, for a MIMO or a multivariable system, I have closed loop characteristic equation is what? Basically determinant of I plus G p G c. This is the closed loop characteristic equation right, equal to 0. Therefore, by analogy I define for a multivariable system G closed loop for my multivariable system purely by analogy as;

minus 1 plus determinant of I plus G p G c divided by determinant of I plus G p G c. If the numerator I called, I call the numerator as W, then this is actually where W by definition is minus 1 plus determinant of I plus G p G c.

So, what shall we do now? This is my closed loop multivariable analog of G C L SISO G C L. Therefore; L C L multivariable by analogy is 20 log on base 10 of G C L multivariable. And now, what I must do is choose my tuning parameters for the decentralized controllers in a manner such that, L C L multivariable max is approximately equal to number of loops times 2 d B because each loop, so, number of loops. So, if it is a 2 by 2 system I will tune my multivariable system. So, that I get an L C L m B multivariable max of 4 d B. If it is 3 loops 6 d B, 4 loops 8 d B. Now, let us come to tune such that, tune controllers, such that this criteria is satisfied. This criteria right here. That is what I want to do. Insert a new slide.

(Refer Slide Time: 17:59)



So, here is the systematic procedure, procedure tuning procedure. Let us say I have got a 2 by example tuning procedure, and as an example I take a 2 by 2 system. So, I have got 2 controllers which we have been looking at. Obtain K c for example, Zeigler Nichols for the first controller; I reset time tau I Zeigler Nichols for the first controller. You also obtain the gain Zeigler Nichols for the second controller, and the tau I the reset time Zeigler Nichols for the second controller. This I can obtain using SISO techniques. These are the individual tuning parameters.

Now the question is; I want to detune both the loops equally. So, what do I do? I choose a detuning factor, detuning factor f and what Liben recommends is; you can detune either you can keep the tau I, the same and detune only the controller gains, and that is what I like to do. Other people detune everything.

So, if you an f is greater than 1 that is the detuning factor. So, what you then do is, adjust f just a second. What you then do is, detune K c, the controller gain, you know set K c 1 is equal to K c Z N. Whatever you had calculated earlier, divided by f and K c 2 is equal to K c 2 Z N divided by f. So, what I am doing is; I am detuning both the tuning parameters, both the tuning, this guy and this guy from their respective individual Zeigler Nichols tuning parameters by a factor of f.

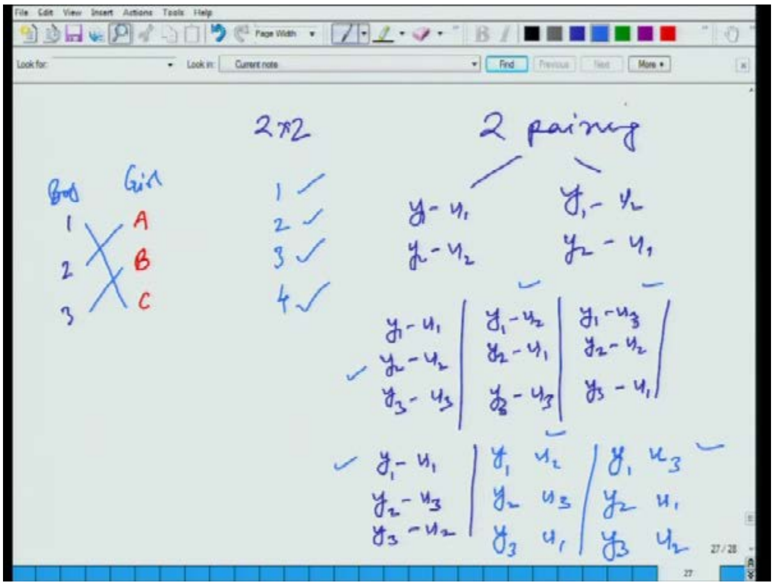
So, once I have detuned, then I can calculate, because now I know my tuning parameters. I calculate L C L multivariable max. Adjust f, adjust the detuning factor such that, L C L multivariable max is equal to for a 2 by 2 system, it is actually 4 d B. So, you keep adjusting, the you adjust the controller gains, detune the controller gains by a factor f such that, you get a maximum closed loop multivariable log modulus of 4 d B for a 2 by 2 system, 6 d B for a 3 by 3 system and so on, so forth. So, this is the multivariable decentralized systematic procedure for detuning individual controllers in a interacting multivariable system. It is to derive this, that we went through all the trouble and all the theories etcetera, etcetera, etcetera.

So, that is that now last, but not least, I want to until now, well until now what we have done is; now if I have a 2 by 2 system, I am assuming u 1 is controlling y 1, u 2 is controlling y 2. A more fundamental question is, should u 1 control y 1 or should u 1 control y 2, and u 2 control y 1. This is the paring question. What should I pair? u 1 y 1 u 1, y 2 u 2 or y 1 u 2, y 2 u 1. This is the paring question. Until now, I have being assuming the paring is fixed.

But now, the question is in a multivariable system. You also have a choice to choose the paring. How should I pair? So, there are interaction matrixes that are used quite commonly to figure out which paring gives us favorable interaction. You see, if you remember multivariable systems, the off diagonal terms G 2 1 and G 1 2, and when you are trying to control both y 1 and y 2, they introduce an additional feedback path. And now, if by adjusting my paring, the G 1 1 and G 1 2, the off diagonal terms can give too much interaction, and for a paring that has not been, that has been chosen appropriately, that interaction can be minimized or mitigated.

So, paring can be used in a manner to make sure that the off diagonal interaction terms are as good as possible so to speak. At the very least, we need techniques to figure out where the off diagonal terms the interaction is so bad, that particular paring gives such a bad interaction, that you know, it really makes control difficult. So, how do we or what matrix do we use to basically reject bad parings, bad parings that lead to interactions terms that are very, you know that are while, what should I say, I am I am struggling for a word, may be not good. So, we want to characterize interaction, characterize the severity of multivariable interaction, and based on that those matrixes, we should be able to figure out these parings the interaction is bad, these parings the interaction is not so bad. So, these paring are worth considering, those parings are worth throwing out the window.

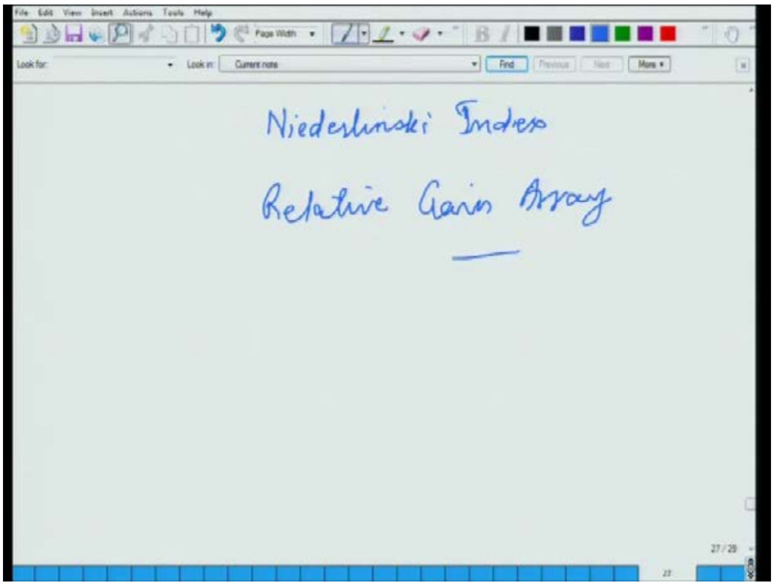
(Refer Slide Time: 24:52)



So, for example, if you have a 2 by 2 system, you got 2 possible parings. What are the 2 parings? y 1 u 1, y 2 u 2 and the other paring is y 2 sorry y 1 u 2, y 2 u 1. Let us say I have got a 3 by 3 system. Now how I have got how many parings? I think six pairing, let us see. So, y 1 u 1, y 2 u 2, y 3 u 3 that is one and next one can be I can flip this guy's I can flip the first 2 guys. So, what I will have is y 1 u 2, y 2 you know y 2 u 1 and then I keep other guy fixed y 3 u 3. That is number 2. Then I could have y 1 u 3, y 2 u 2. Then I could have y 3 u 1.

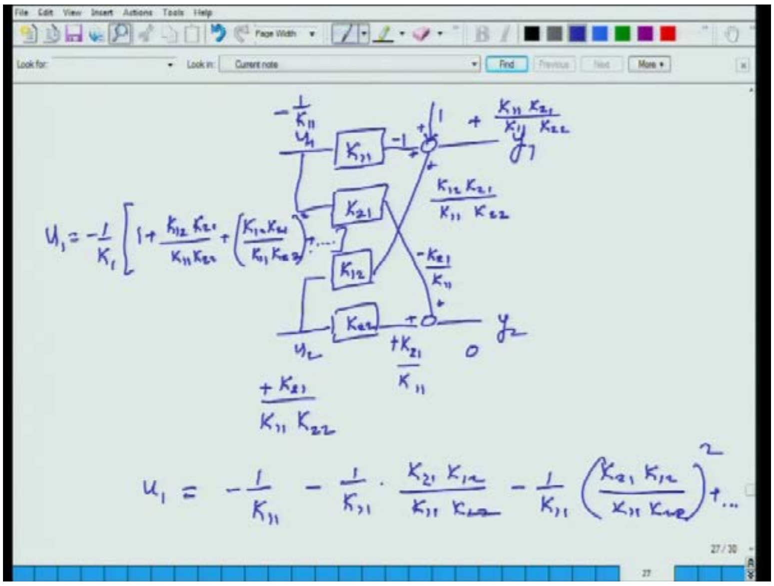
Then I could also have, you know y 1. Oh man this is getting confusing. Then I also could have, let me flip this two. y 1 u 1 y 2 u 3 y 3 u2. So, 1 2 3 4. y 1 u 2 y 2 u 3 y 3 u 1, y 1 u 3 y 2 u 1 y 3. So, these are the 6 possible combinations, parings, couple parings that are possible, input output parings, boy girl or input output parings that are possible. Which one to implement? That is, the question which one to implement? Should I implement this, should I implement this, should I implement this, should I implement this, should I implement this, should I implement this? Which one gives me favorable multivariable interaction, where the off diagonal terms are not as severe, where the interaction multivariable interaction is not as severe? If the multivariable interaction is not as severe, the need for detuning will not be as severe and I will get better control. Right it is like matching compatibility in couples anyway.

(Refer Slide Time: 28:05)



So, that is the question that we are trying to address. And for that, there are two indices, two matrixes that are very commonly used Niederlinski index, Niederlinski index and the other one is relative gain array. And what I will do now and finish of is the Niederlinski index. So, the relative gain array. Niederlinski index is best understood, if I use a 2 by 2 example. And these are both steady state matrix extendible to dynamics where you put in transfer functions. But let us just look at the steady state gain. So, let us say I have got a I have got a system, and I am looking at only the steady state relationship. So, if I make a change in u 1, there is a transient in y 1, but ultimately y 1 ends up changing by so much, the steady state change in y 1. I make change in u 1 what is the steady state change in y 2. I make a change in u 2, what is the steady state change in y 2, what is steady state change is y 1.

(Refer Slide Time: 29:07)



So, that steady state change is essentially the gain given by the gain. Let us say I have got K 1 1, K 2 1, K 1 2. So, let us say I am paring 1 with 1, 2 with 2. So, 1 is paired with 1, this goes like this, and this goes like this. What I am doing is, I am paring y 1 with u 1, and I am paring y 2 with u 2. And let us say I do not have a control system; but let us say I am an operator. If I am an operator, and I am told; look boss you have to maintain y 1 and y 2 had set points.

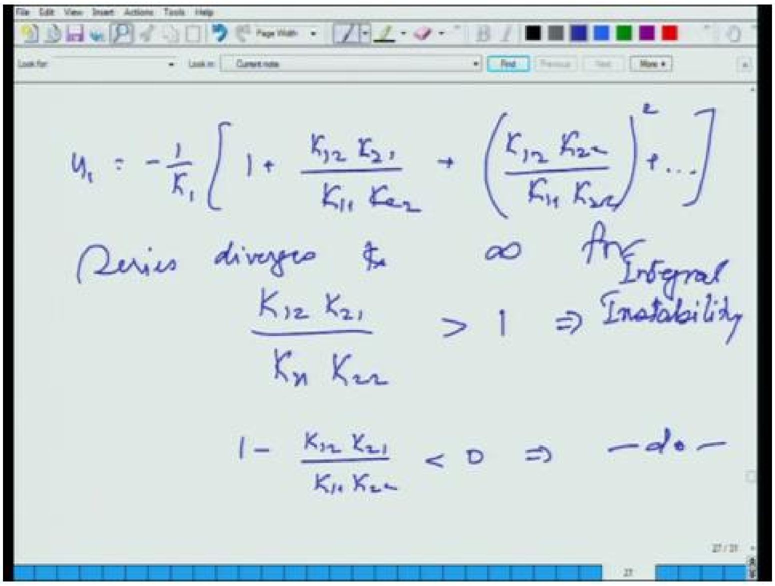
As an operator, what do I do? I look at y 1, and I see well it is not at set point; I may keep adjusting u 1 until it goes to set point. Once you y 1 has gone to set point, then I look at y 2, then I see, well y 2 is not at set point, then I keep adjusting u 2 until y 2 is set point. But because I make a change in u 2, the interaction term, the cross term causes y 1 to deviate from set point. So, I keep adjusting u 1 until y 1 is brought to set point. Once I have done that, because u 1 has changed, I keep y 2 would have deviated. So, I keep adjusting u 2 to bring y 2. I keep doing it again and again and again and again, hopefully if I have done it sufficient number of times, y 1 and y 2 will get to close enough if not exactly to their set point. So, that is the basic thinking that I will apply, to derive Niederlinski index. So, let us say everything is at set point. So, set point is zero. So, y 1 is at 0, y 2 is at 0, u 1 is at 0, u 2 is at 0. Everything is at set point as is at steady state; I am where I want to be.

Now, let us say a disturbance comes in, and let say the magnitude of this disturbance is 1 and everything is getting added up. Because the disturbance came in, now y 1 has gone to 1. If y 1 has gone to 1, what should I do to u 1, so that, this signal becomes minus 1. Then, 1 minus 1 will bring this to 0. Yes or no? So, to repeat, I was at steady state a disturbance came, caused y 1 to deviate. I as an operator sees that y 1 has deviated. So, then what I do is, I make u 1 minus 1 by K 1 1. If I have made u 1 minus 1 by K 1 1, and I wait long enough for the transients to pan out, what I will find is, u 1 has been made minus 1 1 by K 1 1, and therefore, this signal is minus 1 and now minus 1 plus 1 will bring y 1 back to 0. However, when I have changed u 1, this signal becomes K 2 1 by K 1 1 negative sign. And therefore, y 2 goes to minus K 2 1 by K 1 1. In order to, then I say y 2 has deviated y 1 is at set point y 2 has deviated. What should I do to u 2, in order to bring y 2 to set point?

Well if I change u 2 2 minus K 2 1 by K 1 1 K 2 2, then this signal after multiplication by K 2 2 will become minus K 2 1 by K 1 1 plus. So, this will become plus. And once this has become plus, these 2 terms cancel out. Because, these two terms cancel out, y 2 goes back to 0. Even as y 2 has gone back to 0, now u 2 has changed. And because u 2 has changed, this term will become K 1 2 K 2 1 divided by K 1 1 K 2 2. And therefore, now y 1 will go to plus K 11 K 2 1 divided by K 1 1 K 2 2. So, now, I have brought y 2 to set point, but y 1 has deviated. In order to bring y 1, I will again make a change. That will change will again affect y 2. And then once y 2 is affected, I will again make a change u 2, and that change will again make a change to y 1.

So, you see if I keep doing this again and again and again, after one circle, what I find is that. If I look at u 1, after disturbance came in, the first value, that one gave it was minus 1 by K 1 1. And then, y 1 deviated after I have done adjustments to u 2 by K 1 1 K 2 1 K 1 1 K 2 2. In order to bring this to 0, I will have to make a change of minus 1 by K 1 1 times whatever was the deviation K 1 1 K 2 1 K 1 1 K 2 2. And then, if I do it again, what I will get is, K 1 2 K 2 2 whole square plus and so on, so forth. So, basically what I am saying is, u 1 would change by an infinite series if I keep it doing again and again and again, it will change by 1 K 1 2 K 2 1 divided by K 1 1 K 2 2 plus K 1 2 K 2 1 divided by K 1 1 K 2 2 whole square plus and so on, so forth. This will be the infinite series.

(Refer Slide Time: 36:44)

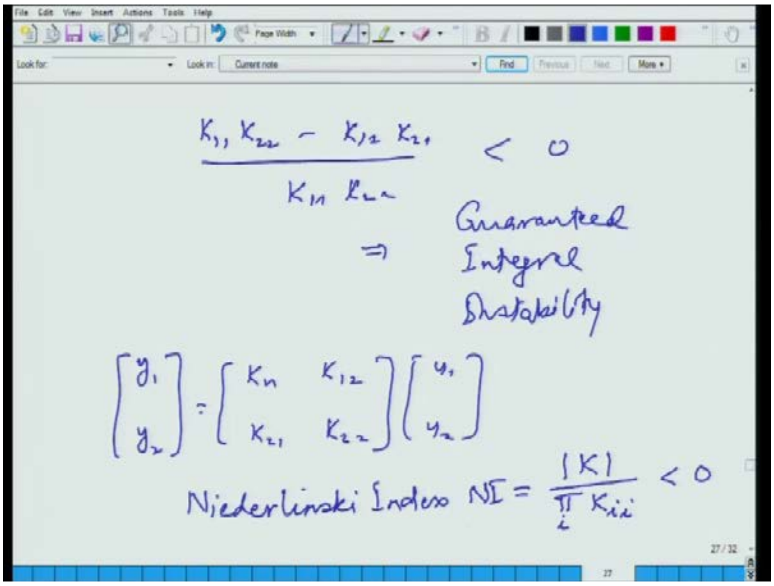


u 1 has to change by this guy. 1 plus K 1 2 K 2 1 divided by K 1 1 K 2 2 plus and K 1 2 K 2 2 divided by K 1 1 K 2 2 whole square plus and so on, so forth and infinite series. This infinite series will converge to a finite value if and only if, K 1 2 K 2 1 series converges, for converges to a finite value. Converges to a finite value, for this multiplication factor being less than 1.

On the other hand, if this multiplication factor is greater than 1, series diverges to infinity for this guy. Yes or no? Therefore, if this condition is satisfied, guaranteed the interaction is so bad, that the, if I am trying to control both y 1 and y 2 at set point, if I am trying to maintain both y 1 and y 2 at set point, guaranteed, the inputs will blow up, the input will blow up to infinity. u 1 will blow up to infinity. u 2 will blow to infinity. Plus infinity or minus infinity, the change would be, the magnitude of the change in the inputs required to maintain both y 1 and y 2 is actually blows up to infinity.

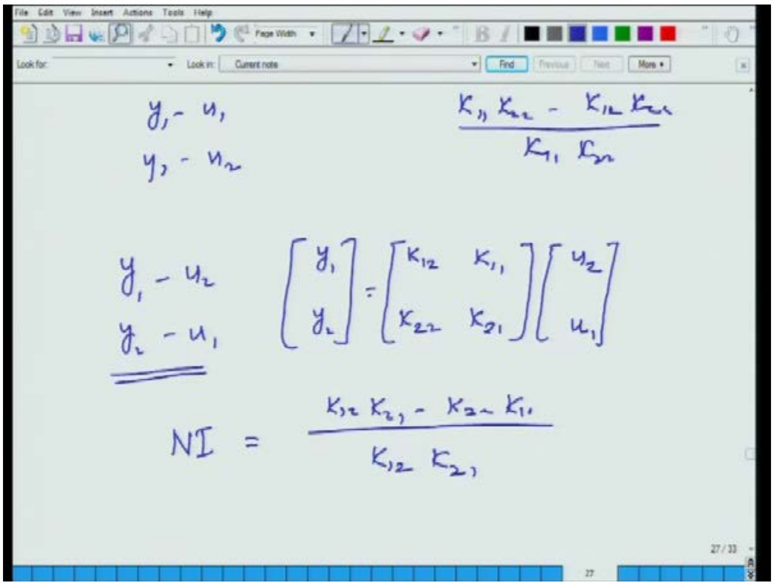
So, if this is true, then you have instability. In fact, it is called integral instability. And why do we call it integral instability? Integral action causes 0 offset. So, if you are using the p I controller in order to maintain things at set point, to get 0 offset, then integral action is in there. And if integral action is in there, and you are trying to maintain the process output at set point, and if this condition is satisfied, you get instability, more specifically integral instability. Integral indicating, that we are trying to maintain both y 1 and y 2 at set point. So, that means, is if 1 minus K 1 2 K 2 1 divided by K 1 1 K 2 2 is less than 0, this implies integral instability.

(Refer Slide Time: 39:59)



And what that means is, if I solve this further K 1 1 K 2 2 minus K 1 2 K 2 1 divided by K 1 1 K 2 2 is less than 0, implies guaranteed integral instability. Now, if you look at the input output relationship, what I had was, I was trying to do y 1 y 2 and y 1 is being controlled using u 1. So, this is actually K 1 1 K 1 2 K 2 1 K 2 2 into u 1 u 2. So, if you look at this matrix, what is it actually the determinant of the gain matrix, the steady gain, the steady state open loop gain matrix, divided by the diagonal terms. So, we define Niederlinski index as equal to determinant of the gain matrix for the pairing that you are recommending, or that you are testing it for, for the pairing being tested divided by the product of diagonal terms. This is called the Niederlinski index. And if this Niederlinski index is less than 0, then you have guaranteed integral instability, what that means, is that the interaction terms are so bad. That if you try and control y 1 at set point and y 2 at set point, if you try to control the output at set point, guaranteed things will blow up.

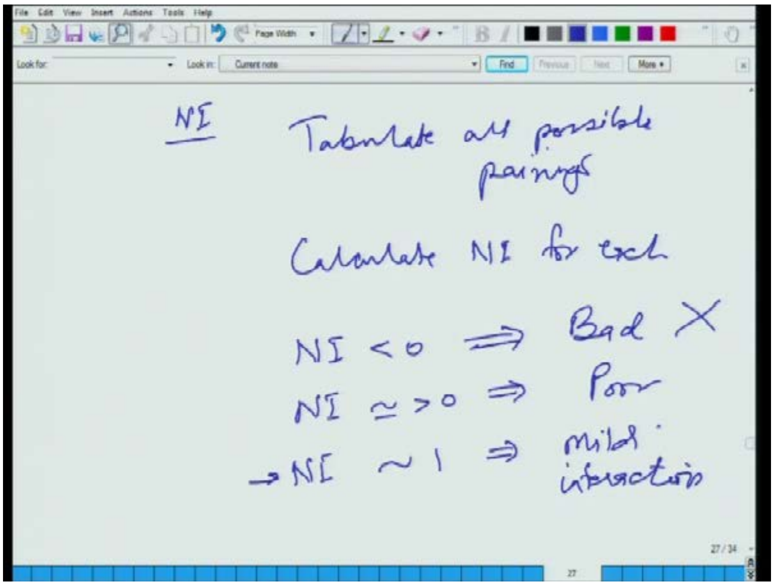
(Refer Slide Time: 42:25)



Continuing with this 2 by 2 system, let us say right now I got the Niederlinski index for the pairing y 1 u 1 y 2 u 2. And then, Niederlinski index was K 1 1 K 2 2 minus K 1 2 K 2 1 divided by K 1 1 K 2 2. Let us say I do a different pairing. Let say, I say, y 1 is paired with u 2 and y 2 is paired with u 1. This is the other paring. In this case, what you will have is, you still write the same thing. However, now what I am doing is, y 1 is being controlled using u 2. So, what I will do is y 1 is being paired with u 2, and u 1 is controlling, is trying to maintain y 1. So, in this case what I will get is, this guy would be K on 1 of 2 K on 1 of 1 K on 2 of 2 and K on 2 of 1. Do you see that the gain matrix earlier for y 1 1 y 1 u 1 y 2 u 2 pairing was? Do you see that the columns have been shifted? I have just interchanged the columns.

So, interchanging of the columns of the gain matrix corresponds to flipping the pairing. And for this guy, the Niederlinski, for this paring, the Niederlinski index would be determinant and the determinant is actually K 1 2 times K 2 1 minus K 2 2 by K 1 1 divided by K 1 1 K 2 2. Now, divided by diagonal terms, so K 1 2 K 2 1. Do you see that the sign of the determinant has changed? And therefore, usually for 2 by 2 systems, one of the pairings would give you a positive Niederlinski index; the other pairing is likely to give you a negative Niederlinski index. And what that means is, one of the pairings is integrally is, can be integrally stable, the other pairing is guaranteed to integrally unstable.

(Refer Slide Time: 44:58)



So, what am I trying to say, what I am trying to say is; if you have a pairing question and you want to use Niederlinski index to figure out which parings give you very severe interaction, then what you do is; tabulate all possible pairings, calculate Niederlinski index for them, for each N I less than 0 these are bad pairings; N I Niederlinski index close to 0, but greater than 0, approximately equal to but greater than 0. These are, this cannot be, these are poor pairings, and Niederlinski index close to 1, and these imply mild interaction. These are the parings, which where the Niederlinski index is close to 1; that means, if you look at the Niederlinski index, and if the Niederlinski index is exactly 1. What that means is, numerator is equal to denominator. Then you will get cancelation and you will get 1. What that means is, if numerator is equal to denominator, denominator is K 1 1 K 2 2, numerator is K 1 1 K 2 2 minus the product of the off diagonal terms.

What that means is, the product of the off diagonal terms is 0; that means, at least 1 of the interaction terms, either K 1 2 or K 2 1 is 0. At least 1 or both. What that means is, at least from the steady state perspective, there is, you know the additional feedback path introduced due to multivariable action. Multivariable interaction is broken because; one of the off diagonal terms is 0. So, Niederlinski index close to one those pairings are worthy of further consideration. Niederlinski index pairings that gives Niederlinski index less than 0 are guaranteed to be unstable. So, those pairings you do not need to think about at all. Now, one thing that I want to point out is, Niederlinski index is a, you know necessary, but not sufficient condition for stability. In the sense, that if Niederlinski index is less than 0, you get guaranteed instability.

If Niederlinski index for a paring is greater than 0, well there is no guarantees. And I mean this you know, I mean if you have a single input single output system, you can always screw up the tuning. So, that the closed loop system is unstable. So, Niederlinski index is essentially a necessary, but not sufficient condition for integral stability, while integral stability. So, what we are saying is that, if Niederlinski index is 0, guaranteed instability, if it greater than 0, well you may still get instability depending on how you tuned your controllers. So, that is that, I think it is a good time to end.

Thank you very much.