Process Control – Design, Analysis and Assessment
Professor Raghunathan Rengaswamy
Indian Institute of Technology Madras
Department of Chemical Engineering
Multivariable Control – Part 2

We will continue out lectures on multivariable control so in the last lecture we looked at truly multivariable controller design which included two single input single output controller designs and design of a decoupler matrix which made the whole overall design multivariable and at the end of the lecture I had mentioned that in all the things that we have seen we have always kind of target a particular input to control a particular output, so if I have a real multivariable system where I have several inputs and outputs in everything that we have seen till now we have looked at pairing these inputs and outputs so one of the question that are not answered till, now is are there approaches to rationally think about how we can do this paring is something that we have not talked about till now.

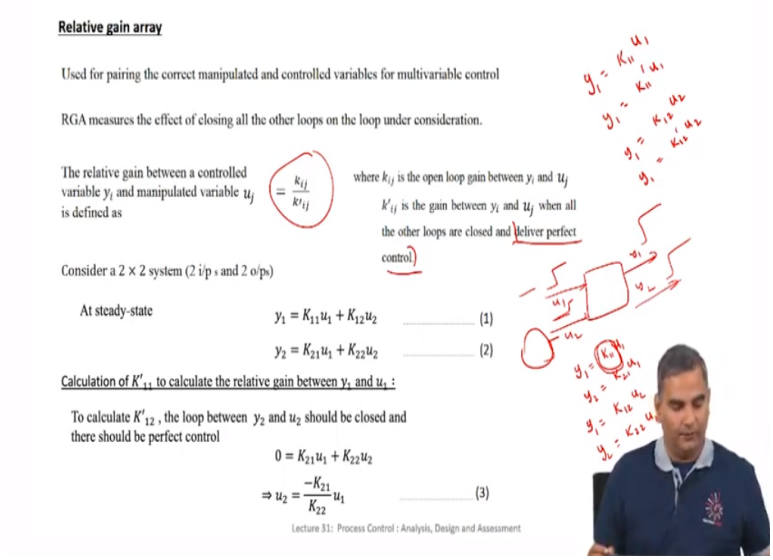
So in this lecture I am going to introduce this idea of relative gain array which is used to do this paring so you could basically kind of do this pairing and then think about all the other techniques that we have discussed and described in detail till now in the last lecture on decoupling we looked at the transfer function between these various inputs and outputs and then directly you saw stands transfer function in deriving this decoupling matrices, so that is the reason why we call this dynamic decoupling whereas in relative gain array basically this is kind of first step in some sense before you actually do all the controller design because you have to do the paring if you are going to come up with let us say fifteen twenty hundred single input single output loops that you are going to design and implement.

So in that sense the less information you use to get a first cut idea of how you should do the pairing the better so in terms of the model that is used for relative gain array basically we are going to look at only the static gain between inputs and outputs to do this really to gain array and I will explain all of this presently, so relative gain array as we mentioned is used for paring the manipulated and controlled variable, so the basic idea is the following so let us again start with a two by two case and then see how this works supposing I have let us say two inputs and two outputs, now I can define again between each of this inputs and outputs.

So for example if I give a step input here and this changes and obviously this will also change and I am not changing this, so this is maintained at zero then basically I can relate again to each of these so Y1 will be K one-one U1 and Y2 will be K two one U1, so basically what we are saying is when I just change Y1 U2 still at zero, so the effect on Y1 and Y2 can be attributed to U1 and that is basically characterized by again, so Y1 is K one-one U1and Y2 is K two one U2 similarly if I kept okay so here U2 is remaining constant right, now if I step U2 up but keep 1 the same then the change in Y1 I will say K one two U2 and the change in Y2 I will say it is K two-two U2.

So there are these four static gains that you get, in this case what we are going to do is basically we are going to look at how we pair this U and Y and what is a best pair that we can come up with and so on by looking at this steady state gains and I have for this two by two case I have already defined these for steady state gains, however this experiment that we talked about is in open loop and only one input is changing at the particular time now what we really what to do is the following, so we want to find what we are going to call as a relative gain and what is that relative gain.

(Refer Slide Time: 02:29)



So a relative gain between a control variable Y and a manipulated variable UJ is defined as KJ by K prime IJ and we will do this definite KIJ is the open loop gain between YI and you J which is what we saw here so we talk about this open loop gains there are four open loop gains between

Y1 and U1, Y2 and U2 Y1 and U2, Y2 and U2, so there are these four gains, now what we want to do is now if I change U1 I know this is the gain right but this is an open loop supposing I were to close the other loop that is I already have a controller in place between Y2 and U2 now if I change U1 what will happen to Y1 is an interesting question ask right would the same gain be there or will the gain be different.

Now since we basically do all our design remember control designs based on this open loop transfer function we are really looking at how much you will change Y an then designing a controller but in a multivariable case whenever U1 changes it does not mean that the U2 is not changing at all so the problem is when I have a particular gain or a particular process model which is an open loop transfer function between U1 and Y1 and I design a controller it is not reflecting the reality in a closed loop where both U1 and U2 are changing, so what I want to do is I want to find out the effect of closing the other loop on say U1 and Y1 loop.

So basically what that means is I am going to define something which is the gain between YI and UJ when all the other loops are closed and deliver perfect controls, so we will understand what this means, so this is an important word so basically all we are saying is so typically in open loop I would have had this K one-one U1, now in closed loop I am going to have something else which I am going to call as K one-one prime U1 and this becomes a prime of the effect of the other inputs you can also kind of think about this, now that we have been doing this for long enough.

So we should start kind of making connection and start thinking about these in slightly more sophisticated ways remember in the decoupler when you had U1 decouple transfer function with Y1 and U2 decouple transfer function with Y2 the model if it were just U1 it would have been G one-one but the minute we made it U1 decouple it was G one-one minus other terms the, the other terms come from the interaction of the other loop, in some sense the model itself change when you look at things in open and closed loop that was a dynamic version of this this is a very simple static version of this where we are saying if I had kept both loop open and then simply gave a step in new one what would be the gain I will get which is K one-one.

Now I want to contract that with what will happen when I closed the other loop in which what I am saying how I have to define what closing the other loop means so I am going to say closing

the other loop means that the other loop is doing perfect control whatever it needs to so it is doing if the other loop is doing perfect control how much of that perfection is of going to affect my current loop is the question that we are asking and that change as is reflected in the gain is this K one-one prime you want, so this is a basic idea and the relative gain for this will be K one-one by K one-one prime.

Now the other relative gain supposing you were to take the gain between one and two then Y1 is K one two U2 and similarly once now if I close the U1 loop the gain I will get will be K one two prime U2 and so on so you can do this for all possible combinations which will be four in this case Y1 U1, Y1 U2, Y2 U1, Y2 U2, so you will get this four modified gains, now the relative gain is basically the ratio of this gain in open loop and the gain that will result if the other loops are closed and perfect control.

So that is the basic idea and the ideal situation is when I simply have relative gain as one because that means both gains are same in which case whatever I see in open loop I will get the same performance even when I close all the other loops and make them perfect control, so if the other loop are not going to have any effect, so that will be the ideal situation but in reality that is not going to happen, now you will notice that when we do this computation it will be very-very similar to your dynamic decoupling so I would really encourage you to go and look back at that lecture and then see how some of these ideas are quite similar.

So now let us look at this equation here so at steady state we are going to derive this relative gain array for a two by two case very simple and then can easily extend that to three by three and so on, so at steady state if I were basically in open loop then and I change both let us say U1 and U2, so these are the kind of equation that you will get and then we describe what perfect control for the other loop means and then we show you how to derive this and relative gain array, so I have let us say change in U1 change in U2.

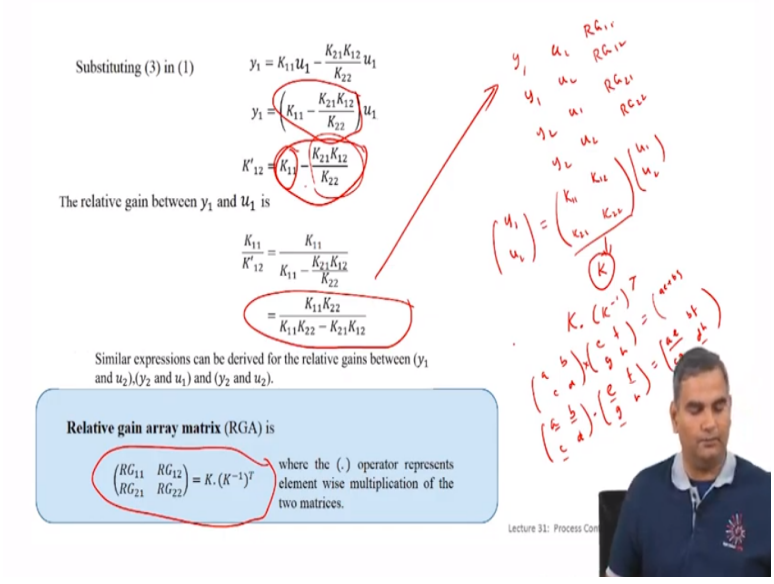
So I can write Y1 this is basically now static models that we writing, this is you can think of this as ultimate effect of U1 and U2 changes on Y1 and Y2 that is reason why we are using only the gain and as I mentioned before the reason why we are using inly the gain is because we want to use as little and is little information as possible right at the beginning, so that we can do pairing

and once we do pairing whatever decisions we are made we will live with those and then correct for the other things through the many technique that we have talked to till no.

So ultimately if U1 and U2 have taken certain values then I am going to have Y1 is K one-one U1 plus K one two U2 and Y2 is K two one U1 plus K two-two U2, now what I want to do is that I want to let us say compute the relative gain array or the gain between Y1 and let us say U1 so first what I am going to do since I am looking at Y1 and U1 the other loop becomes Y2 and U2 and if the other loop has perfect control we are talking about from a disturbance rejection viewpoint here we are not talking about changing set point and so on so when we talk about perfect control for the other loop that means Y2 would have been initially having a value zero and perfect control means it should stay at zero.

So what we are going to do is we are going to set the second equation to zero, so that is how this notion of the other loop delivering perfect control come into picture so since we are delivering perfect control Y2 is not changing from it is value at all right, so in which case I will get zero is K two one U1 is K two-two U2 from this you can get U2 is minus K two one by K two-two U1, so this is the equation you get now what you do is you now substitute this equation back into this for U2.

(Refer Slide Time: 12:44)



So you substitute this equation for U2 then you will see that you can do some simple manipulation, now you see this here, so which gives you K prime one one one one one which is

basically the relative gain when the other loop is closed and perfect and just like dynamic decoupling you will notice that initially it would have just been K one-one if you are just looking at you one but because we want to bring in the effect of the other loop through perfect control of the other loop there is another additional term that comes out so basically what this says is if the other loop was closed you will see this relationship between Y1 and U1.

So I can think of this as the gain between 1 and U1 when the other loop is closed and perfect so the relative gain between Y1 and U1 is K one-one divided by K prime one-one, so which is basically K one-one divided by this right here and if you just multiply this K two-to it up you will get this expression here so these are simple algebraic manipulations that you can quite easily follow through very-very simple and again like I said before since you have been taking this course for, so many lecture it is time to kind of go and consolidate all your understanding through the connection between these various concepts you will see how this is very-very close to the dynamic decoupling idea except that we are working with only statics gain.

Now this was done only between Y1 and U1 this relative gain you can do a relative gain between Y1 and U2 Y two one U1 and Y2 and U2 I will leave this as an exercise for you to basically look through these various transfer function and do these computation for the other things so that easy for you to understand this, now this so basically there will be a relative gain one between Y1 and U1 there will be a relative gain, let us say so we can call this as one-one because it is been one and one so you will have a relative gain one two between Y1 and U2 you will have a relative gain two one between Y2 and U1 and relative gain two-two between Y2 and U2 now the formula here is the following.

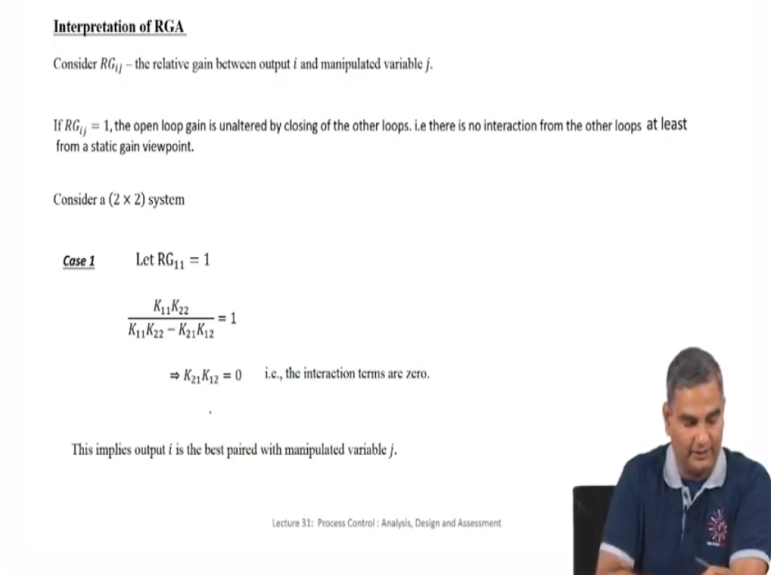
So you need a little bit of matrix algebra to understand this so let me explain this so you can actually go ahead and do this for all the four and actually check this result we will leave that as an exercise for you basically if I want to get the relative gains, so what I is I start with the standard gains which is K one-one K one two K two one K two-two, so if I have Y1 and Y2 the standard gain matrix K will be K one-one K one two K two one K two-two multiplied by U1 and U2, so if you call this as a K matrix then the relative gain matrix which is RG one-one RG one two RG two one RG two-two is K dot, so I will explain what this dot means K inverse transpose right.

Now once I have a K matrix K inverse is easy to compute and the transpose of that is also easy to compute so the only thing I have to explain is what is this dot K inverse transpose means this is not your standard matrix multiplication, now if I have two matrices let us say A B C D and E F G H, now if I do a standard multiplication between these two then the first element here will be AE plus BG and so on, so it is what you have learned from high school and so on, now there I called this dot operation which is an element with multiplication, now what that is, is very simple if it is A B C D dot E F G H then what you do is you simply multiply element by element.

So the first element of this operation will be AE this will be BF this will be CG and this will be DH, so this is the element by element multiplication so basically it is different from this standard matrix multiplication, so if you want to compute the relative gain array all you need to do is get the gain matrix do an inverse of that take a transpose of that and then do a K dot this matrix, where this dot represents element by element, so if it is ABCDEFGH the first element is A times E this element is B times F this element is C times G and this element is D time H, so that if you will get this the nice thing about this is this kind of generalizes this to any size.

So if it is three by three it is same K dot K inverse transpose expect that the K matrix will be a three by three matrix similarly if it is four by four will be a four by four matrix and so on, in fact it will be worthwhile for you to actually do this computation for two by two to see these results match, now that we know how to compute this relative gain array can we interpret the relative gain array.

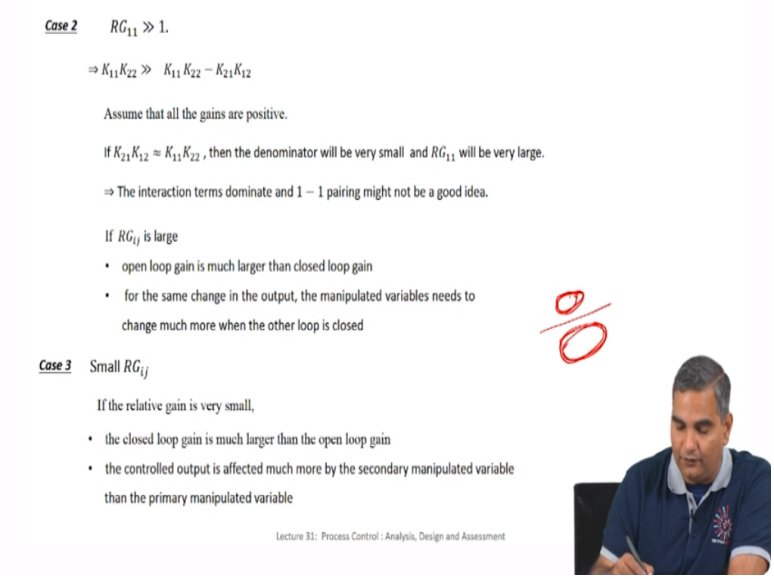
(Refer Slide Time: 17:59)



Now as I said before relative gain array is one is the best situation because what it says is relating when remember is the ratio between open-loop gain divided by gin which take into account the effect of the other loops on the open loop gain now if that gain also turns out to be the same as the open-loop gain then basically what it says is we really do not have to worry about closing the other loops when we design a controller for this pair because closing the other loops does not have any effect on the overall gain anyway so you do not have to worry about it right so that is the best case if you set that relative gain one-one term to zero and see what is the effect of this then basically says the interaction term should go to zero.

So there should be no interaction only then you will get perfect one if there is slight interaction you will not get one but just as a limiting case that you understand that relative gain one is the best and that will come out if you have zero interaction.

(Refer Slide Time: 19:09)



Now if the relative gain is much-much greater than one, so then what does this mean which basically means the denominator which I have here the denominator should be quite small that basically says that the interaction terms dominate, so that basically says that if I had a open loop gain and the denominator term becomes very small, so I assumes that when I make a certain amount of change in new one I will get a certain amount of change in Y1 but that will happen in the open loop but once I close the loop since the denominator has the gain once other loops are closed it says that since the denominator is zero the actual effect U1 will have on Y1 will be much less than what I thought of before because once I close the other loops those effects come and reduce my overall forward effect is what it says.

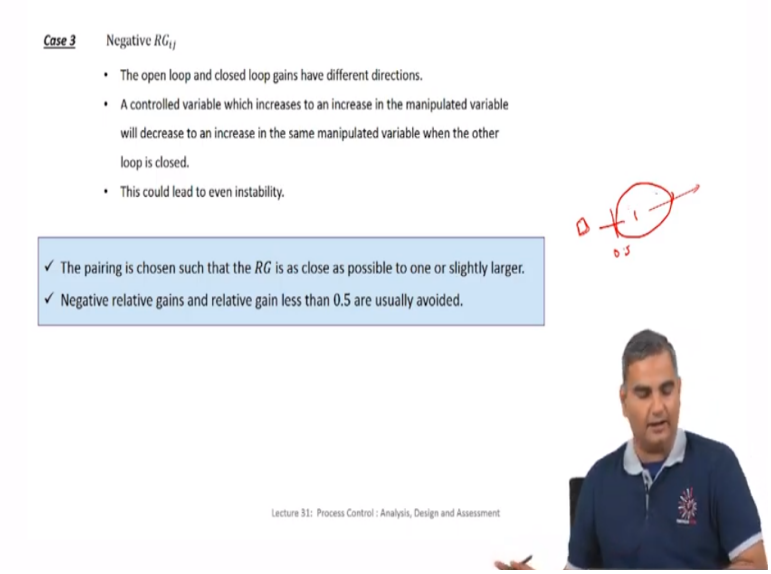
So in other words basically if you design a controller based on this pairing using open loop transfer function then you are going to be very optimistic in the sense that oh if I make this change in ne one there is going to be this amount of change in Y1 which will be true in the open loop case and you would have designed you controller based on that, but once the interaction effect kick in when all the other loops are closed then what is going to happen is whatever change you expect for a change in U1 while you design the controller is not going in the multivariable control sense.

So in once you closed all the other loops the gain is going to be very low, so if you choose this it is not going to be good idea because you cannot do a good design based on the open loop now if

the relative gain is very small that means the opposite, so I have the open loop gain on the top and then when I close the other loop the gain becomes much larger than this that is the reason why that ratio becomes small, so what this says is something interesting it says if you pair this input to the output you get a particular gain but when I closed the other loops, so that means U2 and U3 and U4 are all changing and those changes are inducing much bigger changes in Y1 that is the reason why this ratio becomes small.

So this gain which is because of change in U1 is much less than the change in Y1 that occurs because of closing U2 U3 and so on that means U2 U3 and other things seem to have bigger effect on Y1 then the U1 that you have chosen for this, so that basically means maybe these other secondary manipulated variable which you did not choose for this are better for this because they seem, so induce much bigger changes in Y1 that hat we have chosen, so again very small relative gain is not a good idea to have.

(Refer Slide Time: 22:05)

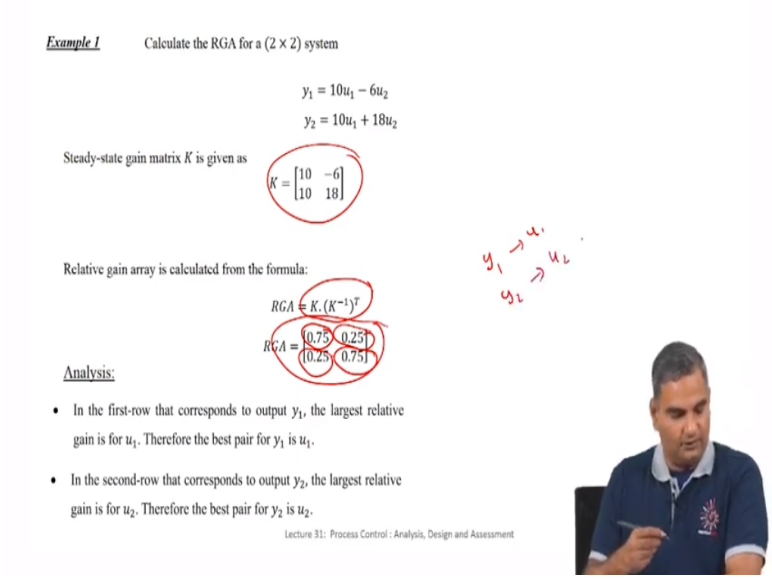


Now the negative relative gain if you have that is a real problem because what it says in open loop if you think, let us say when I change U1 Y1 increase but actually when you close all the other loops it say that if you increase Y1 U1 will actually decrease, so imagine all the pandemonium or the confusion that will delay that will result if you actually tune assuming that increase in U1 will increase by one whereas when all other loops are closed it actually decreases by one, so this can create all kinds of problem leading to instability.

So the key idea is that one is the best right negative you really do not want very small is not good because basically that says the other loops have a lot if not more effect then you our control variable than the manipulated variable you are chosen, so this is I not good very large is also not good we said because the interaction effects are, so much that they are actually going to bring down this gain quite a bit, so the best is to go with something around one or larger smaller is a bigger problem.

So anything below point for you avoid and then you try to choose things which are close to one oneness idea might not get that or slightly larger than one is also okay, so this is the general rule of thumb for relative gain array right.

(Refer Slide Time: 23:39)

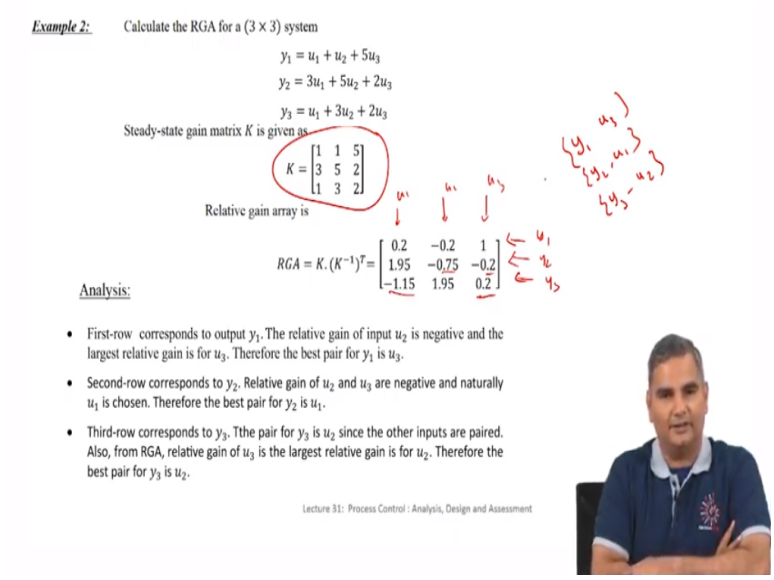


So now let us take a very simple example to do this supposing you had, let us say the gains like this Y1 is ten U1 minus six U2 Y2 is ten U1 plus eighteen U2 that means the steady state gain matrix is this a relative gain array is K dot K inverse transpose, so if you take this K you can do this computation by hand or by software that you use for analyzing control system and then you will get a relative gain array of this.

Now let us try and interpret what kind of paring we should do, so the relative gain between Y1 and U1 is 0.75 and Y1 U2 is 0.25 Y2 and U1 is 0.25 and Y2 and U2 is 0.75, so in this case remember we said we want to avoid everything less than 0.5 and we want to be as close to one slightly lower or slightly larger I okay, so in this case it is very clear cut, so we know that Y1

should be pair with and Y2 should be paired with U2 so in this case it is very-very simple the pairing rule for this example, now let us take one more example which is a three by three system to let you understand this better.

(Refer Slide Time: 24:52)



So three by three again I have gain matrix which is this, now you do the same relative gain computation here then you get this matrix here, so this is rows for Y1 this is for Y2 and this is for Y3 and this is for U1 this is for U2 U3, so if you notice when I look at Y1, now and the one that is the best is U3 which is as close to one as possible so I will pair Y1 and U3 is one pair, so basically means when I design my controller see, so controllers I will use U3 as a manipulated variable for Y1 and clearly we when you look at Y1 to we want to avoid negative.

So basically Y2 will be paired with U1 and then when you look at this here you will notice we want our negative you want to avoid less than 0.5, so the best choice in this case is Y3 will be paired with you two so this is how you look at the relative gain array to do the pairing and once you do this pairing then you go ahead and then do your C, so control loop design for three single input single output loops where U3 is paired with Y1 U2 is paired with sorry U1 Y2 are paired and U2 Y3 are, now once you do the pairing you can directly go ahead and then do dingle input single output control designer you could use dynamic decoupler with these choices and so on so you can do all kind of things once you have chosen a basic control structure that you are going to use.

So with this we have completed the traditional multivariable or traditional advanced control ideas that we use in the next lecture we are going to start really looking at optimization based controller which is the most modern controller that is used in fact I would go as much as saying that almost most of the multivariable are more advanced control implementation in industry our model predictive controller and this notion of model predictive control and model predictive controllers is going to become more and more popular because solving online optimization problems is becoming more and more efficient and we can include constraints and all kind of advantages are there when we use model predictive controllers.

Now model predictive controllers by themselves are a very large subject area however I think it is very important even at an undergrad level to start understanding what model predictive controller are so to be able to teach model predictive controller what I am going to do is I am going to basically talk about the concept and then show a formulation for a single input single output control system how you would formulate a model predictive controller for that and then I will point out how the multivariable model predictive controller design id trivially extended from a single input single output multivariable single input single output model predictive controller framework just so that we understand this as I said before there is a lot of literature and lots of exciting work that is happened and that is happening in the space of model predictive controllers.

So these lectures the four or five lectures that I will give on model predictive control largely introductory but I hope that we give you the tools and understanding you need to start reading and the more advanced version model prediction controller nonetheless I think it is a good idea to introduce this idea or this concept at the undergrad control level, so I will see you next class with series of lectures on model control predictive controller thank you.