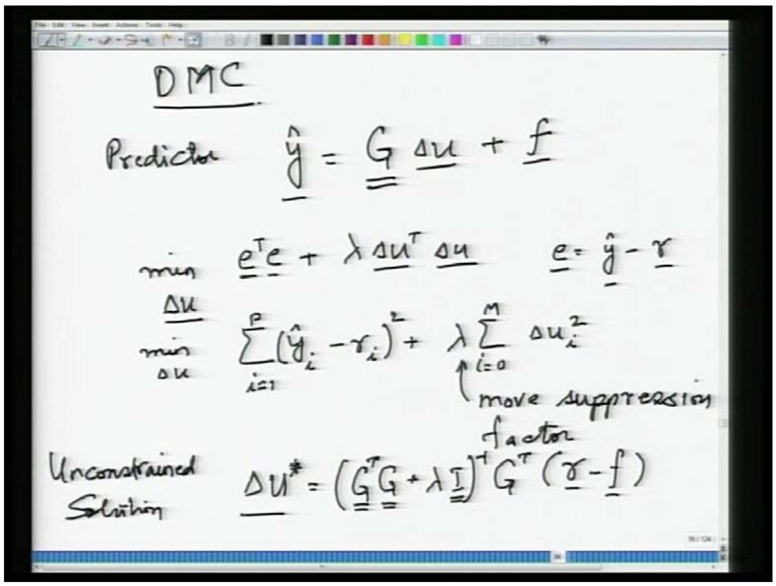
Plantwide Control of Chemical Processes
Prof. Nitin Kaistha
Department of Chemical Engineering
Indian Institute of Technology, Kanpur

Lecture - 10
Model Based Control

(Refer Slide Time: 00:26)



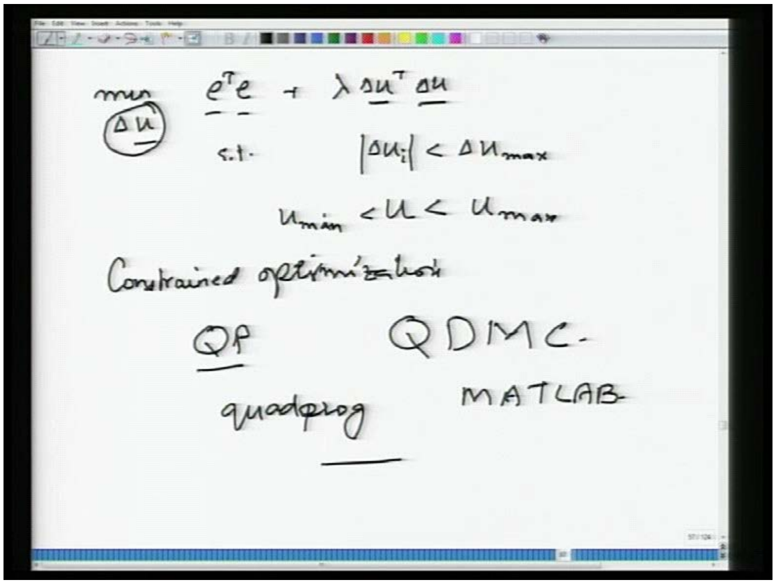
So, welcome to this next lecture. We have been talking about dynamics matrix control. And for the unconstrained problem, what we saw was that my predictor is predictor is y hat predicted the vector for SISO system is equal to the dynamic matrix G times, the set of control moves that are to be taken over the control horizon plus the vector of free response; free response corresponds to no control action, if all the delta uses 0, how would the output move.

So, this was my n step predictor; and if I am trying to minimize over the unknown future control moves e transpose e plus, where these are vectors lambda times delta u transpose times delta u, which corresponds to close set point tracking plus lambda times control effort. If I am trying to minimize this and just to understand this better this e transpose e, actually e corresponds to e is equal to y predicted minus the reference trajectory. So, what we have in essence, or in effect is that we are trying to minimize, if you want to look at it in its full form, what we are trying to minimize is this y transpose i being the time index, so i goes from next sampling instant to p sampling instant a head minus reference trajectory i whole square plus lambda times summation i is equal to 0 to M delta u i square.

So, this is my objective function and I am trying to figure out, what should my delta u be such that this objective function is minimized by the way notice from the objective function that, the more the lambda the more penalty I am imposing on delta use; that means, if lambda is increased control moves are getting penalized.

So, this is called the move suppression factor, move suppression factor, M is the control horizon, T is the prediction horizon and I have my predictor up, there we saw that the unconstrained solution for this is the unconstrained solution is delta u star, star indicating optimum is equal to G transpose G plus lambda I inverse times G transpose times yeah, this was the solution that we looked at last time of course, there are always constrains.

(Refer Slide Time: 04:41)



And those constrains are two of the most common constrains in any loop would be that I want to minimize over delta u the vector e transpose e plus lambda times delta u transpose delta u, and subject to the most common constrains delta u modulus the maximum change in any of the elements has to be less than delta u max for example, in one time instant you can only move the valve, let us say by 2 percent or 5 percent you cannot change the valve position in one time instant or in one sampling instant by more than 5 percent you cannot expect the valve to go from 0 percent to hundred percent.

So, there will be this constrain and of course, there is always the constrain that the actual signal to the valve, which is u should be between u max and u min and u max for typically correspond to a fully open valve u min would correspond to a fully closed valve alright.

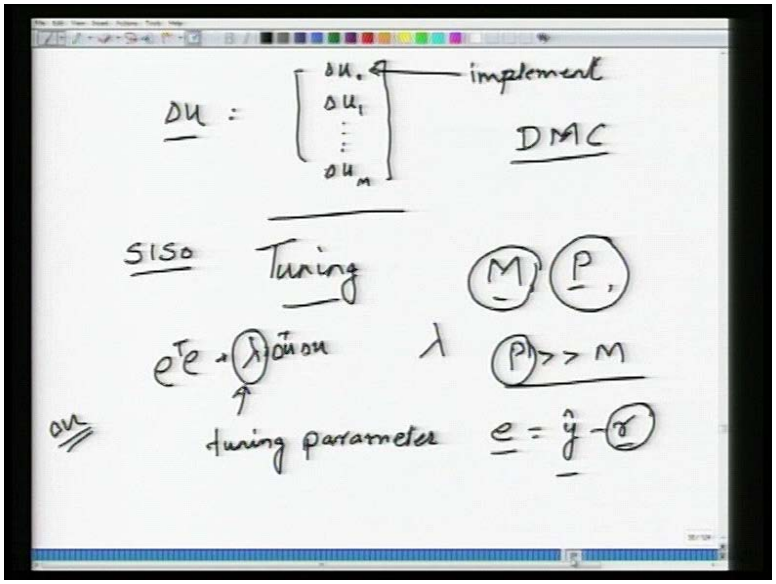
So, for solving this problem now this problem is a constrained quadratic optimization problem, and it is a constrained optimization and there are what are known as quadratic programs, that are that are pretty efficient at solving these types of problems, where the objective function is quadratic in nature.

And you know the constrained have a certain have a certain properties, and this is referred to as QDMC quadratic dynamics matrix control, and one of the most I mean that I know of that that I use quite often is quadprog in matlab, this command or this function is this subroutine is does solve quadratic programming problem.

So, quadprog and mat lab is typically involved there is a lot of theory that goes into and I do not think, we go into it here, point to be noted is what I am trying to do is figure out the best control moves which are here, so that what should my delta u be my future set up control moves be, so that my objective function is minimized, and my objective function is trying to minimize the deviation from the reference trajectory, and the control effort.

So, we want as close you want y to go as close as to the reference trajectory without too much control effort that is what I am trying to do through this minimization problem, and there are standard algorithms that do that for you.

(Refer Slide Time: 08:00)



Now I have calculated after I do whether it is the quadratic program or whether it is an unconstrained DMC I have calculated my delta u, and delta u is actually the small step that I should take right now, the small step that I should take in the next time instant and so on, till delta u M, I am at the current time instant this one gets implemented, so this control moves gets implemented, and then I wait for the sampling instant at the next sampling instant I get my new measurement, and then I repeat this whole process all over again.

So, I am here I do my optimization calculate what the future set up moves should be of those future set the current one gets implemented, and then at the next time instant, when I get here, I repeat the whole process all over again the this is how DMC or most mpc techniques I mean that that techniques may differ in how in the kind of predictor that they have, but essentially, they are common in the sense that you got the quadratic objective, and you are trying to minimize that objective by figuring out what you should do to the control input over the control horizon future control horizon.

So, this is how deep DMC in particular is done or is implemented, what a some of the things that we have missed which are worth pointing out right now, we have looked at right now SISO, DMC mile generalize to MIMO system in a little bit tuning parameter, how do tune just like you have got a PI controller or the PID controller.

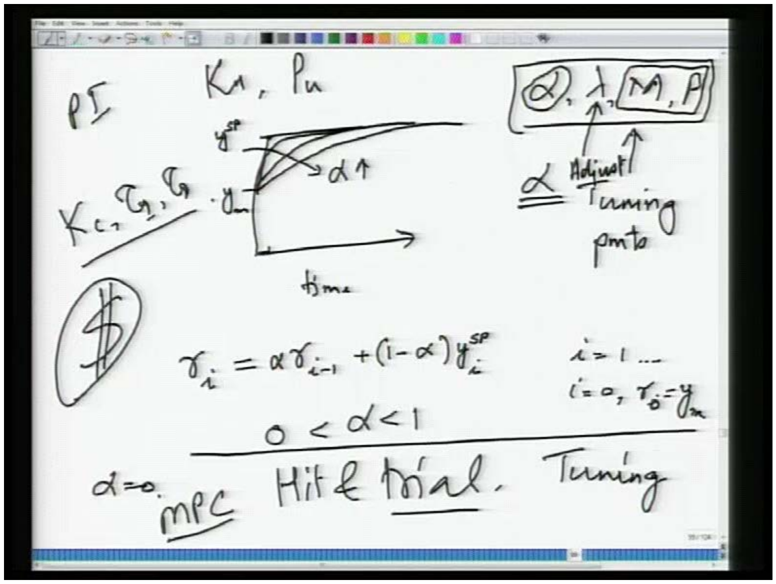
Where you have k c tow y and tow d, tuning, what are the parameters in your hands that you can adjust to get the kind of control performance that you want to do that or you can think, what should be my control horizon M, what should be my prediction horizon P, I can also think of that moves suppression factor lambda, this moves suppression factor actually if you, if you try and minimize e transpose e, where lambda equal to 0.

What you will find is you get very aggressive delta use delta use are very aggressive large changes in delta use, and when you start making large changes in your control input, that essentially drives the feedback loop or the control system towards instability. So, to stabilize your feedback system that is why we added this lambda time’s delta u transpose sorry delta u term, this lambda is a tuning parameter, and it is called moves suppression factor.

I told you just a little while ago it’s a tuning parameter and this something that your control, what should I say control system designer adjust to get the kind of close look performance that you want, if you choose 2 smaller value for the control horizon, then the control can the controller can only make, so many control moves to ensure set points tracking or close set point tracking or close reference trajectory tracking, that being the case if this is small air controller becomes aggressive, it will it will try and make all the moves that are necessary, so that in the future the error in the set in the in the predicted output is as small as possible.

So, small m's will lead to an aggressive controller, large m's would lead to a sluggish controller similarly, for the prediction horizon same thing, note that p would typically be much larger than M, how should you predict over while at least one time constant, you know if the response takes thirty minutes to line out step response, you should be making a prediction for at least 10 15 minutes, and M would typically we say half or one-third, may be even one-fifth of or one-fifth of p the problem well, there is another tuning parameter that I should talk about see I we did not talk about that reference trajectory yet, you see I when I wrote error I defined error as y predicted minus, the reference trajectory how do you get the reference trajectory, this is something that I will discuss right now.

(Refer Slide Time: 13:18)



Let us say I am here, this is where my y is and my set point is here, and of course, this is time axis, so current is here, y measured y set point is here, that is where I want to be well then, the reference trajectory is typically is defined as an exponential rise to the set point from wherever you are, this exponential rise can be fast that is a fast exponential, or it can be sluggish, and one of the ways of doing, it is to say that r i is equal to r i minus 1 times alpha plus 1 minus alpha times I would say it should be well y set point I which would typically be constant y set point i, and we will say that i goes from i is equal to 1 to large and at I equal to 0, r i is equal to y r 0 is equal to y measured notice ,and an alpha is typically between 1 and 0.

So, if I say alpha is equal to 1, if I say alpha is equal to 0 alpha is equal to 0, what I have is r one will be equal to 0 times y 0 plus 1 times y I set point; that means, at the next time instant I am here at the next time sampling instant, my reference trajectory would be my reference would be with this, so that is like a step increase to the set point, that is my reference trajectory as I keep increasing alpha from 0 to 1 I get more and more damping.

So, as alpha increases I get slower and slower rise to the set point. So, you can imagine that again this alpha is a parameter in the hands of the control system designer, if you choose small alpha I want a fast rise to the set point what that would do is make my control system aggressive. So, and the larger alpha I use the more sluggish the control system becomes alright.

So, alpha lambda m and p are the tuning parameters that are in the hands of a control system designer to tune your model predictive controller to get the kind of response that is desired to get the kind of control performance that is desired tuning parameter.

What you would have is typically designers would choose reasonable values for these fix then to reasonable values also take to be a reasonable values or reasonable exponential rise, alpha will have to be chosen appropriately for that, and then adjust this adjust the lambda to get, whatever it is whatever is the type of close look control performance that you want.

The problem I would not call it a problem is just, even though model predictive controllers have existed for so many years, how do you choose reasonable values of alpha lambda M and p, you see if you have a PI controller things are very simple get the ultimate game, get the ultimate period, and then you got your tuning table or tuning table etc,, and that tuning table you can implement, and that tells you what are your reasonable values for k c tow and tow d.

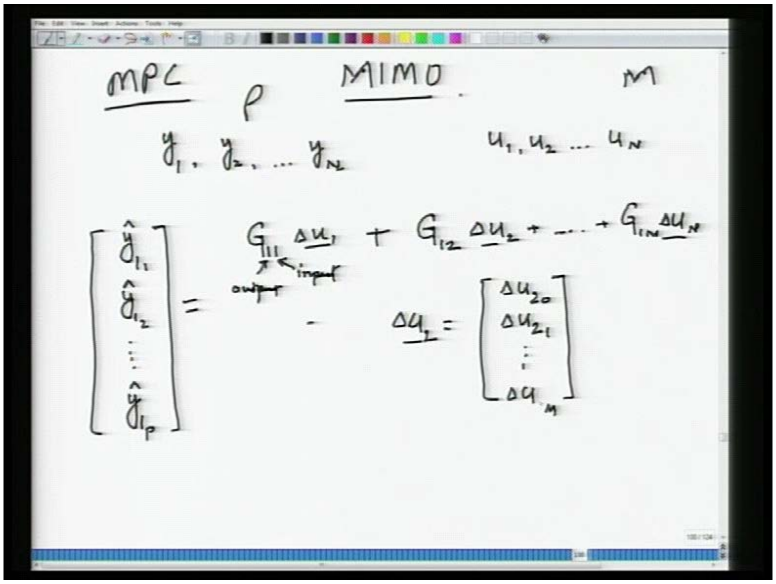
So, there are standard tuning methods for PID controllers, unfortunately fortunately model predictive controllers, there are yet not sys systematic standard procedures that can be applied to get alpha lambda m and p, it is more hit and trial, you keep on adjusting lambda until you get the kind of control performance that you want.

So, tuning is essential in hit and trial, and hat is one of the major disadvantages, it is one of the major disadvantages of essentially all mpc techniques, all model based control techniques and therefore, because the tuning is not trivial, you do not have a standard procedure, just blindly applied and there you are it requires effort and engineers being lazy, engineers by definition somebody you know remarked engineers by definition or lazy people, there has to be significant justification to justify that.

If I am applying model predictive control, sure it will take me effort to built that built, and tune that controller, but that effort is worth it because it brings about such, and such a significant improvement in control performance and that significant improvement in control performance is actually translating to for example, extra profit and when I say extra profit that is not just a few dollar, you know substantially extra profit.

So, this trade off is always there model predictive control takes efforts, takes time to design what is the benefit that it brings, and when is that benefit good enough to justify putting in that much effort that trade off is always there, and one needs to be aware of it hopefully as we go through this course, you will you will get to see where mpc makes sense, and where a PI or PID controller would do just fine, but that is for later.

(Refer Slide Time: 20:25)



One of the advantages of mpc is that it can be readily extended to MIMO systems, multiple input multiple output for the time being let us, just consider square system, so take a square system then what you have is you got output 1 output 2 and let us just say output N, these are outputs that need to be controlled, and in order to control these input outputs you got u 1 u 2 and u N these control inputs.

Now, if I want to predict for the for the sake of convenience, let us just assume that we are predicting y 1 to y over the next p time instance, and u 1 to u N or going to be moved or the control horizon for each one of them is again m, this is just for the sake of convenience it is not necessary.

So, prediction horizon is p for all the outputs control horizon is m for all the inputs, just for the sake of convenience let us say you are doing it like this, then if I want to predict y 1 meaning y

1 predicted at time 1 from now, y 1 predicted at time 2 instance from now and, so on y 1 at time p from now, hat indicating prediction, if I want to do this what I will have is I will have the dynamic matrix G 1 1 times delta u 1, the effect on output 1 of input 1.

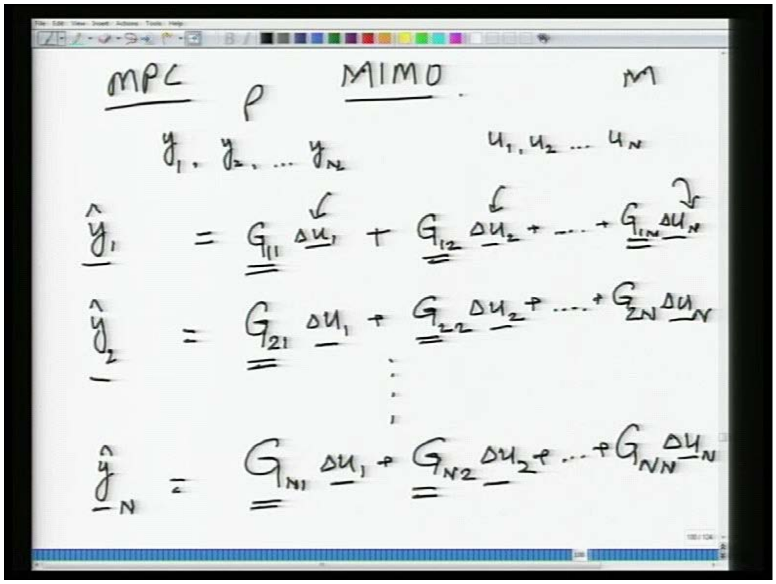
So, there will dynamic matrix corresponding to the 1 1 pairing, that is G 1 1 plus, so you see if I if I change output 2 or input 2 that also affects y 1, that effect comes through the dynamic matrix number 2, which is G effect on 1 of input 2 times delta u to plus effect on 1 of input n times delta u N.

And by the way what is G 1 1? G 1 1 is equal to I get a small unit step change to delta u 1 to input 1, so this is unit and in response to this I record to what happens to output 1, so this is u 1 and this is output 1, and this change in this record of output the transit response of output 1, this gives me co efficient G for and these step coefficients are what do you know we have, so discussed this earlier right, what happens, what is G 1 2? Well I give a small step to input 2 records what happens to output 1, this coefficient going G to 1 right and, so on.

So forth what is delta u 1? Delta u 1 the vector is and, so on change in input 1 M instance from now right, similarly what is delta u 2 well, delta u 2 will be just have to rub this off delta u 2 change in input 2 right now, change in input 2 1 instance from now and so on, so forth chain in input 2 M instance from now and so on, so you can define delta u 1 delta u 2 delta u prediction of how output 1 would respond over the prediction of how output 1, would respond over the prediction horizon what is disguise to changes in u 1 u 2 and u n to changes in the input.

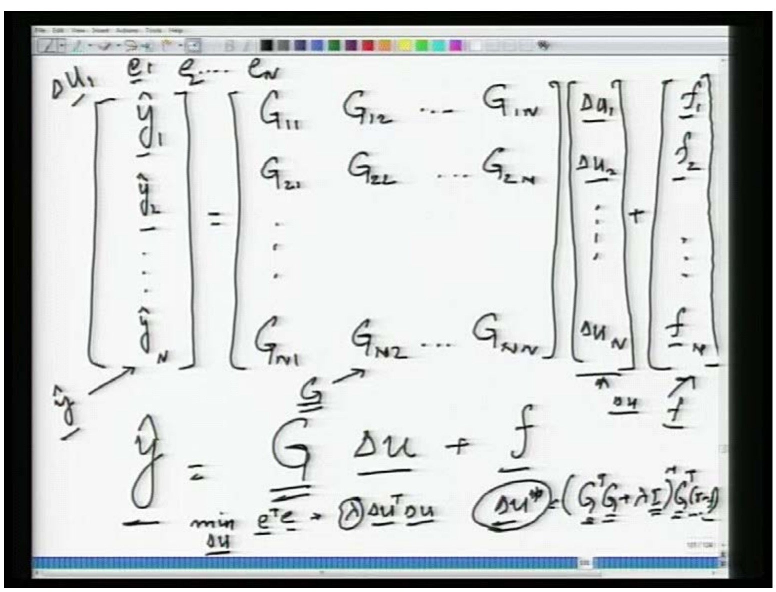
So, what I have I mean if I if I right this in matrix form, and I simplify this what I have then is y 1 predicted over my prediction horizon comes from here.

(Refer Slide Time: 26:09)



Similarly, I will have y 2 predicted over the prediction horizon would be dynamic matrix G on 2 of 1 delta u 1 plus G on 2 by the way, these are all matrix just to clarify that on 2 of input 2 delta u 2 plus G on 2 of input N delta u N, and so on so forth I can do this for all the outputs and what I get is 1 N of one delta u 1 plus on N of 2 delta u 2 plus on N of input N delta u N, and you can see if I put all of this in matrix form.

(Refer Slide Time: 27:31)



What I can get is y 1 predicted y 2 predicted and, so on y 1 predicted is equal to G 1 1 G 1 2 G 1 N G 2 1 G 2 2 G 2 N by the way I made a mistake is minor one, but this is the prediction because of control action plus of course, there will be the free response I forgot the free response of output 1 plus the free response of output 2, and free response of output N.

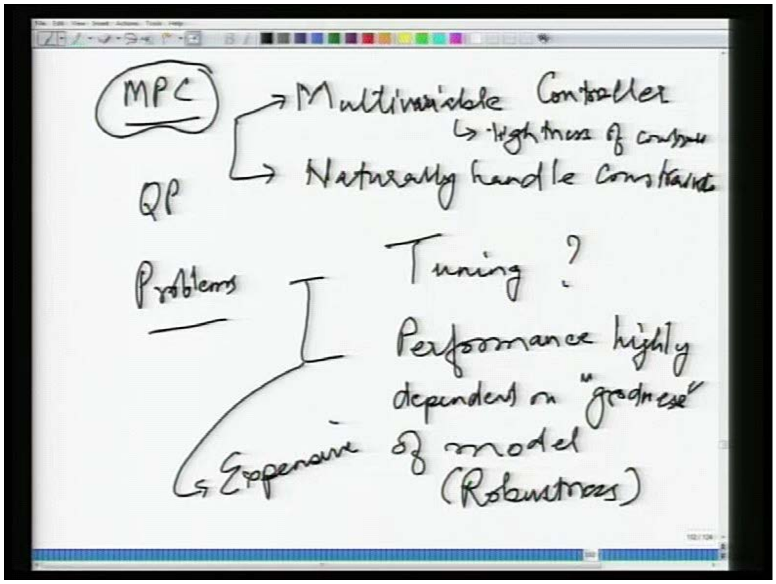
If I do nothing how would the output 1 2 and N change over time. So, there is there is also the free response component. So, I think I forgot that blah blah blah G N 1 G N 2 G N N times delta u 1 delta u 2 delta u N plus f 1 free response in 2 free response in for this multi variable system I again find, if I call this vector y hat is equal to the G, this whole big matrix is the multi variable dynamic matrix, now big G times delta u, where this is what is been called delta u plus, this is where it is been called f if I call this f if I call this delta u, and if I call this G the matrix, and if I call this y the prediction notice, I had a very similar equation for a SISO system.

So, what we are saying now is whether it's a its a SISO system or a MIMO system multiple input multiple output, my predictor is the same is of the same form and therefore, if I am looking at unconstrained optimization minimize over delta u e transpose e, where I have got reference to output 1 and output 2 output N plus lambda times plus lambda times, what can I do not know lambda times delta u transpose by the way here, I can have well do not worry about it, I could have different lambdas for different control inputs, there is also the issue of scaling the y, so that they have they get relatively the same the same weight age.

But, if you do not worry about those scaling equations the point is that the form of the equation is the same, and therefore, the minimum solution will also correspond to the same form unconstrained solution, would be delta u star will be the same thing go transpose G plus lambda I inverse times G transpose times r minus f well.

So, the formalism remains the same whether, it is a single input single output system, or a multi variable system, I did all this for a square system, where u had N inputs, and N outputs option you know matrix methods can also handle non square systems, the formalism would still remain the same and I would like to go on, it because then that becomes the course and process control it is really not essential.

(Refer Slide Time: 32:55)



So, the point is that advantages of MPC DMC if one of them, it is truly a multi variable technique notice, that when I am inverting this and since this matrix, has got all the components in it meaning, this matrix has got all the components in it G 1 1 G 1 2 G 2 2 all the you know.

So, this matrix takes care of effect of u 1 y 1 and u 2 and y 1 u 2 and y 2 and so on, so forth and I am inverting this matrix, the control moves that I get what I mean to say is the delta u that I will get it. So, delta u 1 what I do to control input 1 depends not only, but e 1 is error in variable 1, but it also depends on other error into error in N and so on so forth.

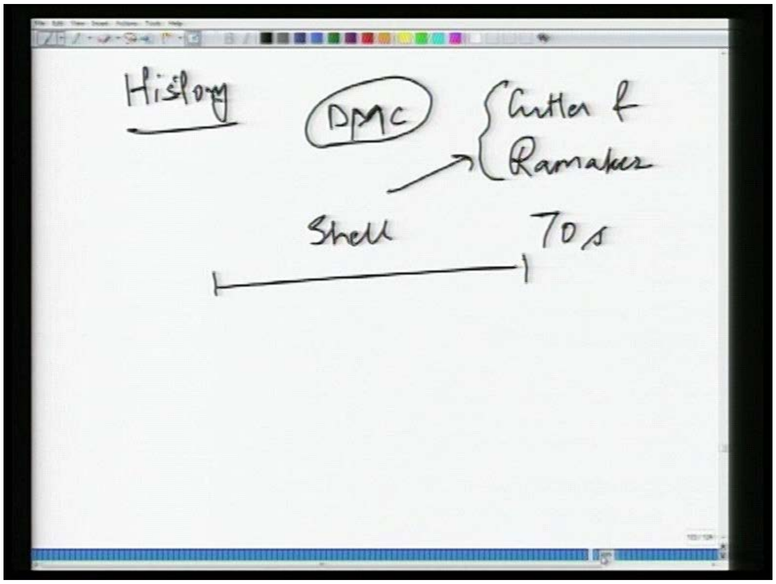
So this what would I say the calculation of the optimal control moves which comes to either matrix inversion or quadratic programs it is naturally multi variable, that means the change in input 1 depends not only the error input 1, but also error input 2 sorry, the change input 1 depends not only error in output 1 but also error in output 2, output 3, output N, so what how I move input 1 depends on all the errors, how I change output 2 input 2 depends on again all the errors.

So in that sense multi variable formulation of DMC or MPC, it is truly a multi variable controller because, you take you know the control moves take care or take into account all the interactions that can occur, or that are there in the multi variable system, it is a truly multi variable constrain and you see when you are doing Q p, you can naturally handle all types of constrains, you know you can incorporate the constrains of your physical system naturally handle constrains.

Problems tuning there are no standard procedures, you just have to do hit and trial whatever works is good. How do you do tuning, how do you tune a multi variable controller is a is more an art there is no systematic procedure for it.

Tuning is a problem, performance degradation performance highly dependent on goodness of model I will just call it goodness of model, as your process drifts your model, which was really good is more longer as good, and you will find just like a in the smith predictor performance will start to degrade, and close look system may actually go towards instability robustness is an issue.

(Refer Slide Time: 37:54)



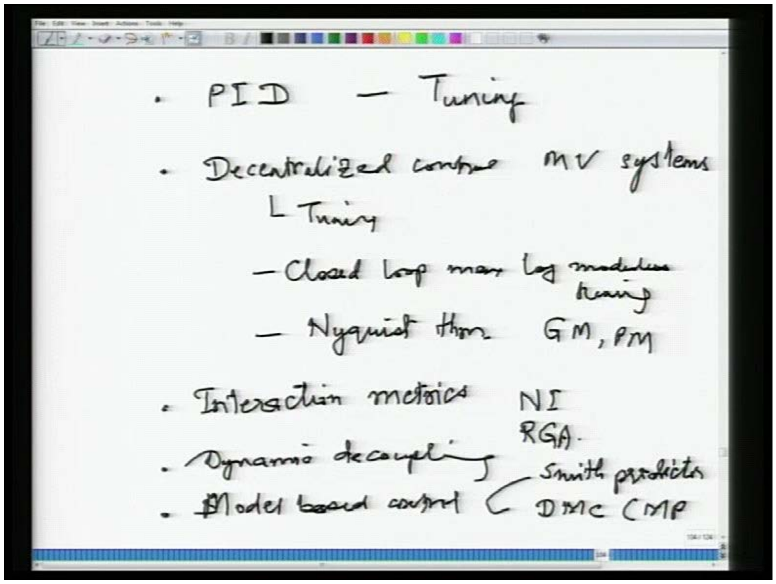
So, this is actually called you know robustness, even if the process drifts will your controller give reasonable controller performance, that is the quick that is the question and of course, it is expensive, these are some of the problems, So like I said before since it is a multi variable truly multi variable controller in multi variable applications in particular tightness of control will be really good compared to you know decentralized PI type of controllers, you can naturally handle all type of control constrains like, I said before these advantages the economic benefit that tight control, and handling of constrains bring in these most far out way, the effort that will take to tune the controller etc, and then and only then is it justified to apply MPC as a as a short corollary oh no not corollary just as a as a note on history alright.

So, what I was saying was just as a note of history or on about the history of MPC, DMC was actually formulated originally by cutler and ramakar cutler, and ramakar if I am not wrong and these were not academicians, they were actually working with shell in one of the refineries in shell and this happened in the late seventies.

So, what I would like to point out is that this technique, the whole the whole technique of model predictive control has its origins in the process industry and not in academics you knows mathematicians, and control academician's and. so on so forth.

Of course, once it was proposed academicians took to it like, bees take to honey and lot of theory has come into being because of academic interest, the point is well the point is this is essentially a technique that was that came out of the industry, and so we should not think that industry may nothing as happens, we just operate and procedure lots of good thing happens in industry also lots of you know powerful research happens in industry also this is an example of that I can give you many more examples, but to controls this is relevant I think I am done; however, let me because we have got some time, let me summarize what we have done over the past I do not know six seven eight lectures.

(Refer Slide Time: 39:37)

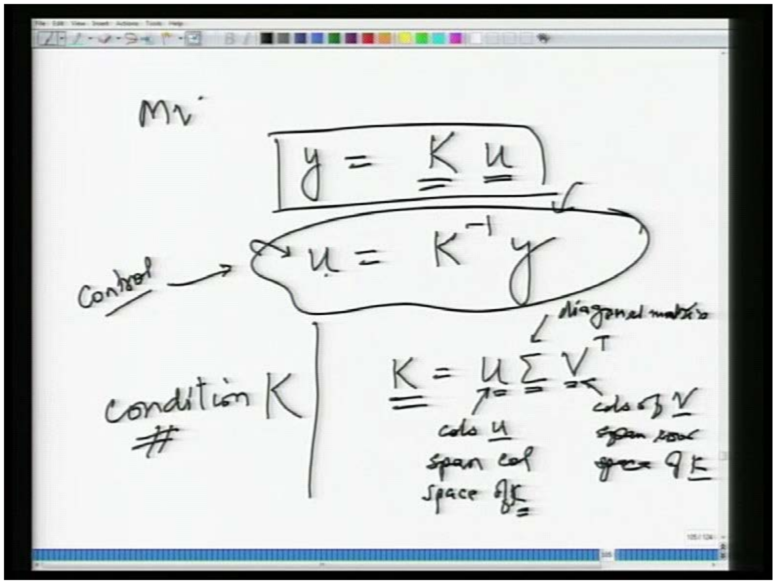


We have looked at PID controllers, we have looked at how to tune them, we have looked at decentralized control for multi variable systems for multi variable system, and in order to understand how to tune it, we went through closed loop maximum log modulus tuning log modulus tuning in order to understand, how the hell this came about we went through Nyquist therom right half plain pole this 0 and, so and so forth.

Nyquist theorem we also discussed in that process in margin phase margin for multi variable systems, we also discussed interaction metrics not matrix, but metrics interaction metrics and these were Netherlansly index relative Ganaway.

We also looked at decupling, dynamic decupling and finally we have looked then we looked at smith model based control techniques, model based forgive my handwriting and then we look at smith predictor for processes were difficult open look dynamics, and we have just finished looking at dynamics metrics control, which is actually one of the techniques for MPC, the techniques differ approaches the same it is just, the algorithmic details that are different from one approach to the next to the other, all this we have looked at maybe I should in this context I should also point out when you are looking at interaction, let us say you have got a multi variable system.

(Refer Slide Time: 42:17)

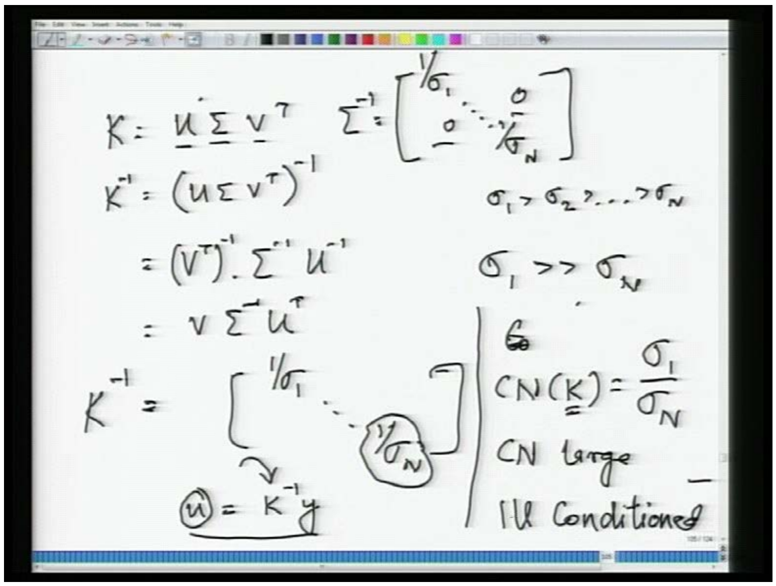


That multi variable system will have y output is equal to gain metrics times input right, well just to whenever, you see a metrics that is one of the things, that that I recommend to all regardless of your application, whether it is a control application in control, what we are trying to do is given y set point, what should I do to you what you are actually trying to do is what should you be, so that y is wherever I wanted it to be what should I do to input that is the problem, that I am trying to solve through that feedback loop alright, because you are trying to invert, your open look model this is your open look model right, if I make a change in you through the gain metrics I get whatever is the change in y in output, the inverse problem is I have my y some place, I want them to go to 0, what should I to you.

So, in controls we are always trying to invert the process model, that is just by way of this is just very straight forward way of looking, since you are trying to invert one of the first thing that you need to do is look at the condition number, condition number of k, what do you mean by condition number of k, well I have to explain, it let me tell well let us see this goes into Eigen vector and Eigen directions Eigen value.

So to simplify it let us just take the singular value decomposition of k, there is always a singular value composition of any e metrics, whether it is non square or fat skinny does not really matter for the time being let us just consider, it to be a square metrics because k inverse will exist only for square metrics metricsis, there is always what is called a singular value decomposition, and in singular value decomposition what you get it is u and v columns of u span, which space output space column space, column space of k, columns of v span of v metrics, span rows space of k this metrics is a diagonal metrics.

(Refer Slide Time: 45:32)



And since it is a diagonal the property, that is this comes from singular value decomposition SVD is that this singular values are in decreasing order well; that means, this sigma 1 greater than sigma 2 is greater than blah blah blah is greater than sigma N.

Given this also what you find is u transpose u is equal to the identity metrics; that means, all the columns u are of magnitude 1, and perpendicular to each other to each other, similarly v transpose v is equal to I, now when I am calculating k inverse, you see because inverse of u is u transpose inverse of v is v transpose similarly, inverse of k trans v transpose is v.

So, when I am transpose. So, k is equal to u sigma v transpose k inverse will be u sigma v transpose inverse, and that would be v transpose inverse times sigma inverse times u inverse, and v transpose inverse is v, sigma inverse is of course, sigma inverse, where u sigma inverse would be right, if sigma is that sigma inverse would be this times u transpose, singular value gives me this decomposition, notice this would be v times 1 by sigma 1, so on 1 by sigma n times u transpose right.

What I wanted to say was when you are inverting the metrics this actually shows it quite clearly; you are dividing by or multiplying by 1 by sigma 1 by sigma 1, and so on. So forth, these are called the singular value, sigma's are called the singular values without going into too much theory point that I was trying to make was even for if you take a decupling system, where is everything is you know diagonal and let us say these for the sake of understanding, you know these are these are identity metric sis, let just say then what you find is k inverse is this now if sigma 1 is much greater then sigma N, when I am trying to invert y is equal to k inverse oh sorry u is equal to k inverse y, that is what I am trying to do in control right, then what I have is y has certain deviations I want to bring it back to 0.

What should I do to you, so therefore, I am trying to solve this equation and when I am trying to solve this equation, if this singular value if the smallest singular value, which is sigma N is very small, this number would be very large right, and if this number is very large some of the use may actually blow up right.

So, the point that I was trying to like this is the condition number is defined as condition number or let us call it; condition number of metrics k is equal to largest singular value divided by smallest singular value.

Now, if this number condition number is large, that implies there are large singular values, and there are some very small singular values and then when I am trying to invert the metrics, what that would cause is some of the use will turn out to be very large numbers. So, my output has deviated only a little bit in order to bring it back, I will have to change some of my inputs by very large amounts, such a system is called an ill conditioned system, beware of such ill conditioned systems, because it would be hard to control, such ill conditioned system.

So, when we are looking at interaction metrics, and relative Gainery, and this and that thing I think I forgot, so what you should do the first thing, that you should do is look at the gain metrics calculate, it is condition number, if this condition number is large beware, it is very likely that your controller will not be able to perform too well, and the problem is inherent in the system the system itself is ill conditioned may you are better off not controlling a some of the things, that you are trying to control.