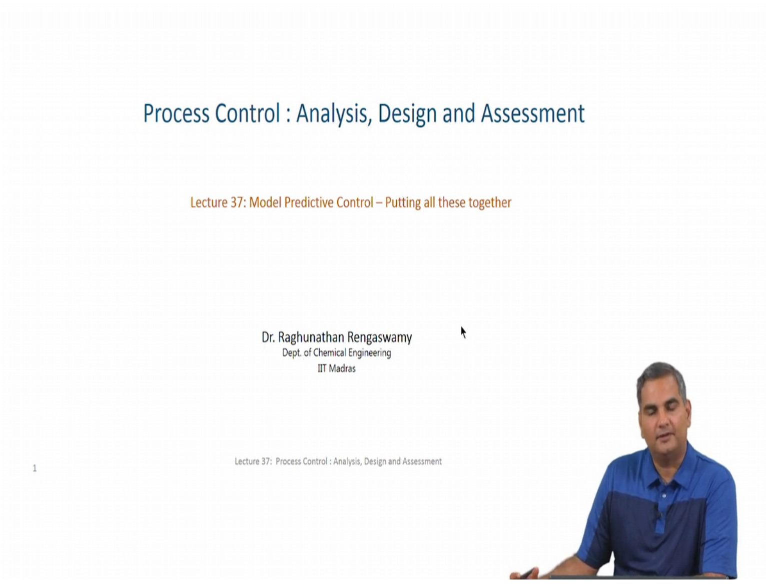
Process Control - Design, Analysis and Assessment
Professor Raghunathan Rengaswamy
Department of Chemical Engineering
Indian Institute of Technology, Madras
Model Predictive Control – Putting all these together

(Refer Slide Time: 0:17)

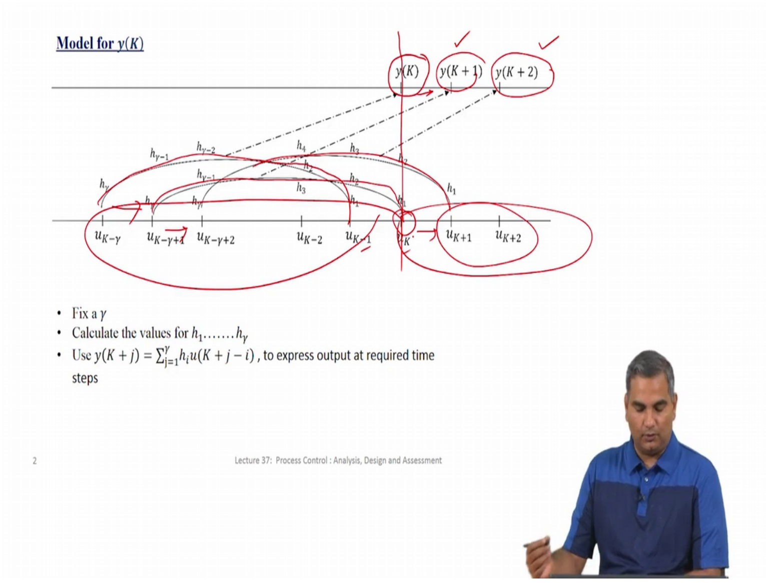


I have talk about how an optimisation algorithm has an objective function, constraints and decision-variables and I have related all of these to be controlled view point and then I also talked about how we get a discrete model of the process, so that we can together this model predictive control formulation. So in this lecture I will quickly combine all of these together to show you how you optimisation, formulation is put together and the beauty of this model predictive control formulation is that since we have an optimisation formulation and not an analytical expression for the control law, any optimiser can be used to solve this problem.

So if you find more efficient optimiser you can simply use this formulation with that optimiser, so the accent in terms of the model predictive control material is more on the concepts of formulation and how do we formulate this problem rather than the solution which is standard optimisation solution but in fact this makes it very nice because we can come up with various more answers and enhancements in the formulation and while we are doing this we do not have to really worry about how we are going to solve this right.

So ultimately once you put all of this together if you have an efficient optimiser that can solve the classic problem that you are interested in solving from a model predictive control viewpoint. We simply give it to the solver to solve it, so in that sense, in some sense the solution to the formulation is kind of (())(1:55) to some extent from the formulation that makes it easier for us to (())(2:01) this formulation better.

(Refer Slide Time: 2:07)

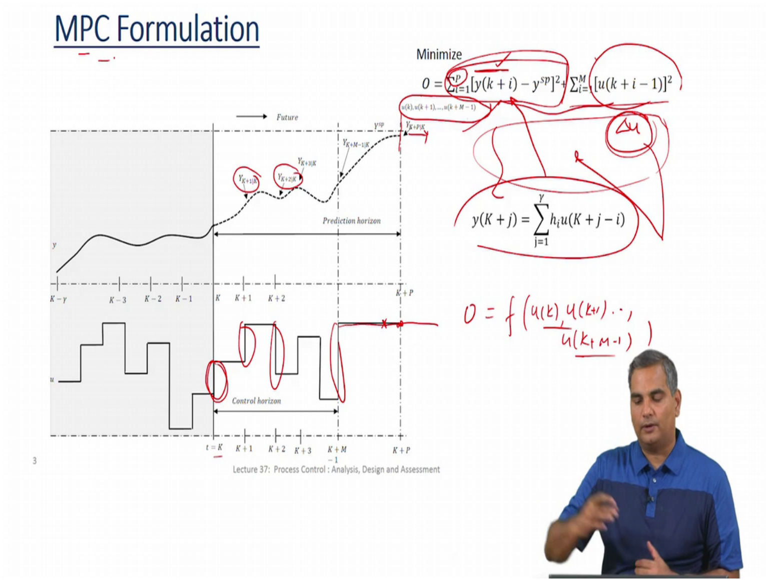


So I am going to recap some of the things that we have seen as part of this module, so here is one picture which talks about how this model is used, so for example if you have an output at the K times then that is working with this envelope here which basically says to predict yK you need to know uK minus 1 uK all the way up to uK minus gamma. Now if you want to predict yK plus 1 what you do is you simply move this one step ahead and also this envelope is moved one step ahead, so if I want to predict yK plus 1 you look at this envelope here.

So I need to know uK, uK minus 1, uK minus 2 all the way up to uK minus gamma plus 1 because we have slided this up one more. Similarly if I want to look at yK plus 2 I look at this envelope here, so I slide this one, slide this one, so I need values for uK minus gamma plus 2 all the way up to uK plus 1 for me to be able to predict yK plus 2. So this goes on, so if I am at times K I still can write these expressions for yK plus 1, yK plus 2 in terms of inputs that have already occurred which are here and input that I am going to make now and future inputs, so this is a key idea which we have talked about quite a bit.

So in other words if I am interested in controlling my system and looking at how I am going to take my output values close to set points in the future starting from yK plus 1, yK plus 2 and so on. Basically what it says is I can write these as functions of the current manipulated variable value that I should keep and future manipulated variable value I should keep, so I should write these as a function of these manipulative variable values which is what we have already talked about quite a bit in this series.

(Refer Slide Time: 4:07)



So now we have everything to put together this optimisation formulation. As a recap we are at the time p equal k and whatever control move that we are going to make at K, K plus 1, K plus 2 and so on. These will effect outputs in the future so they will start effecting the output at K plus 1 given K, K plus 2 given K.

I already talked about given K meaning we are at time T equal K okay so these outputs of 2 yK plus p are going to be affected by the choices that I make and the way I make my choices is that I am going to make a choice for uK, uK plus 1, uK plus 2 all the way up to uK plus M minus 1 that means I am making M choices here and then what I am saying is after I make this choice here then I am going to say I am not going to make any more choices, I am going to let the input be at the same value okay.

So when we formulate this all we need to find is value for uK, uK plus 1, uK plus 2 all the way up to uK plus M minus 1, those M values if I find then I know that UK plus M is the same as UK plus M minus 1 and so on till the end okay till the prediction horizon and we also said that it does not really make sense to go beyond prediction horizon because the change that I make here will only affect an output after this but our prediction horizon itself stops here and if you assume the system or (())(5:39) then up to K plus P minus 1 you could have kept making these choices but we use this as a tuning parameter and then say though I could make choices up to K plus P minus 1 to affect all the way up to K plus P of the output I am going to stop at K plus m minus 1.

Just as a matter of convenience and I am going to use this M itself as a tuning parameter in my MPC formulation. So that is what we decide to do here, so if you notice then we think about putting all of this together as an optimisation formulation we see that we can define an objective function here which says in all future time instance whatever is the y that has to be very close to y set points, so I am going to say yK plus I minus y set points square which allows me to have this yK plus I from I 1 to P be close to the y set point and not only that we also say the same time I do not want to make too much control moves, so I am going to kind of trade off both of this, so this is the objective that I come up with. I could have simply come up with just this first term also it does not matter.

This is a more generalised formulation and of course we can make this even more general by adding terms for Delta u and so on, so that will not only say that I do not want to make very big control values but at each instant I do not also want to make very big changes, so if I want to have constraints that I want to impose on this then I could either put them in constraints here or I could add them in the objective function and since I am minimising this all this delta u changes will also be minimised okay so there are 2 options but traditionally people can put that in the objective function and then weight these various objective function terms and then correspondingly tune how aggressive I want my controller to be and so on so typically these 2 terms are definitely used.

You could also use the 3rd term delta u and on top of it while you use it in the objective function you can also put some constraints on u and delta u as a part of constraints also it does not matter. Remember the most important thing is once we have converted this into an optimisation formulation then you can pretty much do any of these things you want, we can add any number of constraints and so on as long as you are able to pose problems are solvable and meaningful.

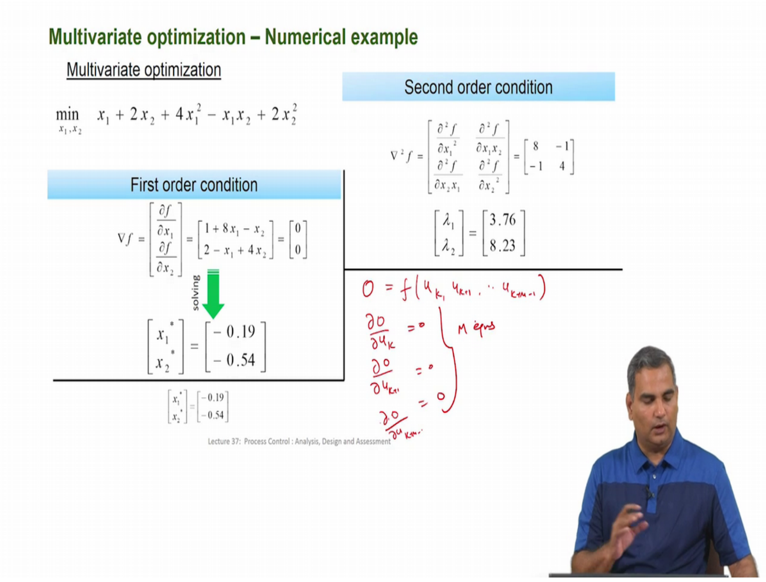
Now when you look at this, so we talked about the objective function, we talked about conscience may be you u delta u and so on and the 3rd aspect of the optimisation formulation are the decision variables and in this case clearly I have to make a decision about uK, I have to make a decision about uK plus 1, k plus 2, k plus m minus 1, so in a single variable case I have this m decision variable that I have to choose, however if you look at this formulation it looks as if these do not participate here at all other than here it does not participate here but remember these terms or these decision variables participate here through the model.

So that is the key idea because depending upon how I change my manipulated variable values the output is going to change, so yK plus 1 will depend on uK, yK plus 2 will depend on uK and uK plus 1, yK plus 3 will depend upon uK, uK plus 1, uK plus 2 and so on right. So that dependence on the moves that I am going to make in the future and the current move then I am going to make is captured by the model, so if I put this model into this term here then you will see the decision variables not only come in the 2nd term, the decision variable also come in the 1st term, so this whole objective function you can think of as some function of uK, uK plus 1 all the way up to uK plus m minus 1.

So the reason why I am writing it this way is I am showing to you that we have now got it into an optimisation form where it is a function of these m decision variables so I used to give it to an optimiser and the optimiser will give me the values for this m decision variables, so the model comes in this here and that is where the part of the model predictive control is relevant. The predictive part is here, so how many outputs in to the future that I am going to predict and that is the predictive part here in model protective control and the control part is actually I am trying to find out what values I should keep for this uK, uK plus 1 all the way up to uK plus and minus 1 right, so these are the control input.

So once I identified these values then I can actually control my system that is where the control part of this model predictive control comes into picture. So you see how this is different from your standard PID control in the sense that there is a prediction horizon, it is not just immediate concern I have. I have a concern about optimising over a prediction horizon and I am explicitly using model for doing this control.

(Refer Slide Time: 11:28)

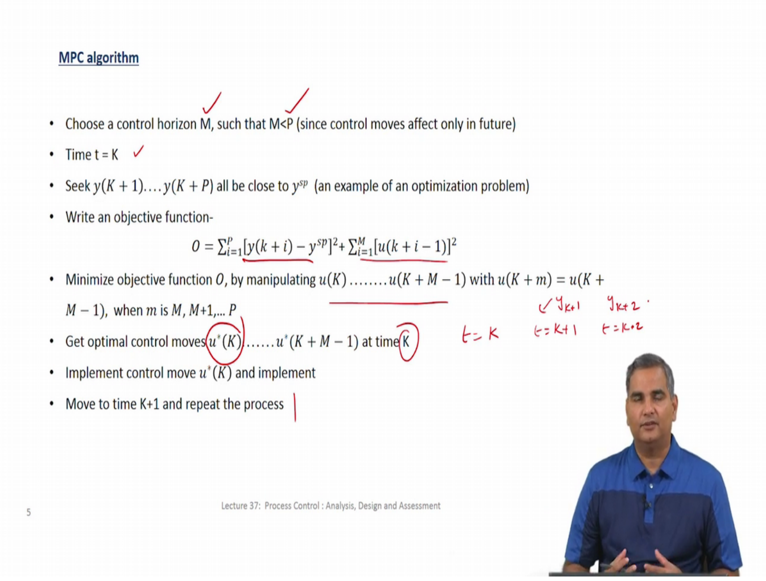


I showed this before in one of the lecture is just to kind of bring that back here, so that we understand all of this comes together in model predictive control. Remember I showed you that if you have a multiple variable and optimisation there is this objective functions there are 2 decision variables you can write this del f del x 1 equal to 0 and get 2 equations to solve this. Similarly in the model predictive control formulation based on what he saw in the last slide I said the objective function can be written as some function of uK plus 1 all the way up to uK plus n minus 1.

So this function has both the model and the other terms and all that which we saw. Now if I want to optimise press then I have to del objective by del uK 0, del objective by del uK plus 1 is 0 and so on and I will get del objective by del uK plus n minus 1 equal to 0, so I will get my m equations in this M variables and then if I solve for this I will get my M values and then if you formulate it right.

It happen that the second order congestion will be automatically satisfied that basically means if I keep these values for this control variables that would essentially optimise my objective function, it will actually minimise the objective function which means that I am going to be able to follow the set point trajectory as closely as the system would allow me to do while not expending very high control effort, so I have the two-term and I can based on the application emphasise one term deemphasise the other term and so on, so that is how all of this works together.

(Refer Slide Time: 13:19)



So in terms of an algorithm what happens is we start and chose a controller as an M, we choose a prediction horizon P and M has to be less than P which we have talked about quite a bit. Let us assume that we are at time t equal to K, so what we are looking for these variables to be optimise and these output values to be very close to set points, so which is reflected by this part of objective function and at the same time we want to ensure that the control moves or not very dramatic. Now we said this objective function is a function of uK, uK plus 1 all the way up to uK plus n minus 1.

So you optimises this is where you simply give it to a solver, so we are talking about a formulation here and the actual solution is really sent to an optimisation solver which will solve for this and then it will give you the optimal control moves u star K up to u star K plus m minus 1. Notice that all of this controller moves you are going to get at time K equal to K, so at time K you really need only one decision to be made which is I have to implement a control move here.

So what you will do is you will implement this okay and then wait and then you will go to t equal to K plus 1 and you will repeat this process and so on and you will keep controlling your process, so that is the way MPCL gather marks. Now 1 question that you could ask is if I am going to implement only one controller move, why did I compute all of these control moves at time K, so I will answer this question through 2 important points that I want to make.

The 1st one is forget why we compute all of this at p equal to k but let us see what makes sense in terms of implementation, right so if am sitting at time t equal to K and I have what is called this move plan which is I have u star for k all the way up to u star k plus m minus 1. One option is to not look at this process till you T equal to k plus 1 and then simply implement this move plan and keep implementing this move plan and then once this move plan is exhausted again to a batch of move plans and then keep doing that, so that is one option. The disadvantage with that option is the following, when we make this decision for this move plan we are sitting at time t equal to k, so all the information that I have till now with me has been collected till time t equal to k and I made my move plan right.

Now at t equal to K plus 1 I have got more information right because I have got another output value yK plus 1 right and t equal to K plus 2 I got even more information I have yK plus 2. Now if I stick to my original move plan then all this information that is coming in the future I am simply going to ignore I am not going to take advantage of this, so it is something like this supposing you are trying to study for an exam and thing that the exam is very hard and you decide that you are going to put a lot of effort and you are going to solve some samples papers and then see how well you will do in your exam. Let us say you start studying today and you plan for a week and so after one day of studying you take a sample paper and do this right.

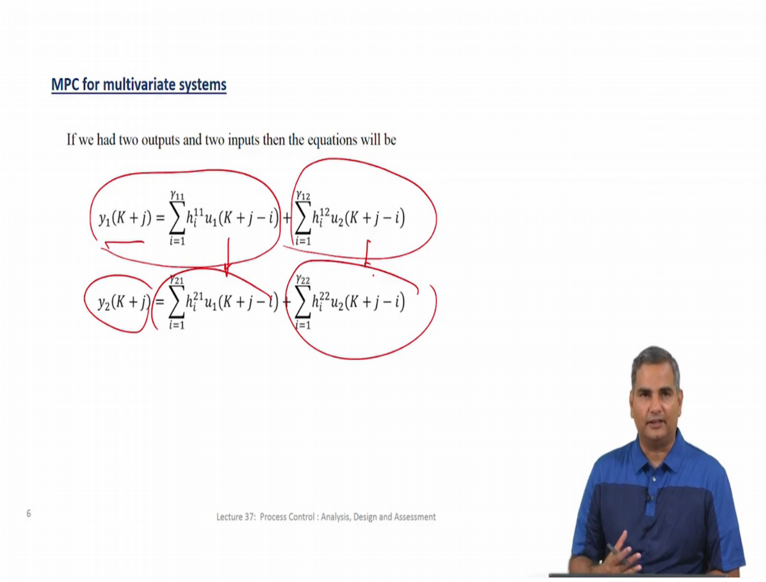
Now if you find a sample paper very easy right you cannot stick to your original plan right you will calibrate based on whatever happened one day later your study pattern because there are other exams that you want to do well so may be (())(17:12) of portion time depending on where you are doing well, where you not doing well and so on. Similarly if you did very poorly in the sample paper then you know the effort that you have plan for this is not enough, so you will redo this, so in that sense it makes more sense to basically keep evaluate what is happening and not stick to your original ideas and as an when there is new information that comes in, in real life what we will do is we will recalibrate our plans and thinking based on that new information right.

If I say I am going to play for the next one week cricket and it starts raining clearly you cannot do it, so we have to re-evaluate the plan, maybe I cannot play cricket I have to do something else and so on right. So that is an important idea for having a move plan right and only implementing the 1st move. Now let us flip the question and say if I have only one plan move that I am going to implement, why do I find it all of this values is another question you could ask, so here you can show this theoretically but I am going to appeal to little bit of intuition to explain this to you.

This M you chose can be used as a tuning parameter to make the controller more or less aggressive okay. So if you have more moves in your bag you can make the controller a little less aggressive than if you have a single move that you are making every time you think you are going to make every time, so the idea is the following supposing you are in a room you are at one edge of the room let us say you want to reach the other end of the room and if I told you, you can jump 10 times and reach the other end of the room what you will do is you will take measure jumps and go to the other end of the room.

However if I said okay you have only one move to jump to the other end of the room, what you are going to do is you are going to really try and jump very hard, so it is going to be very aggressive about your jumping right. So that is the same thing that happens in controllers, so if I say I have only one move that I am going to calculate but there are several predictions in the future I have to make them close to set point then your controller is going to be quite aggressive, so that aggressiveness in the controller could be traded off by choosing this very long and make your controller less aggressive very short and will make your controller very aggressive and so on.

(Refer Slide Time: 19:51)

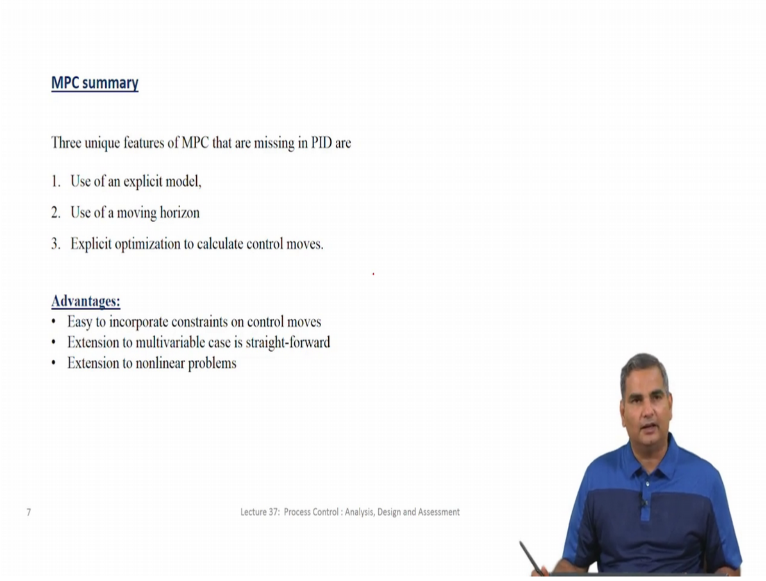


Now one last thing which is something that I wanted to talk about, the great thing about MPC is it does not matter whether it is a single variable problem or multiple variable problem it varies in the extends to multi variant problems, so here I am showing a case where if you have 2 outputs and 2 inputs what will happen is we were only looking at this portion for single input. Now when you have multiple inputs and multiple outputs for each output you will also add on the effect of the other input and you will also add the other outputs and the effect of this input here and this input here.

Now notice how this is structured right all of this simply ads on because we are assuming the underlying process to be linear, so it is very simple extension to multi variant cases. Now in multi variant case what will happen is everything else will remain the same except now if I have single variable and control horizon of M I have M decision variables. If I have 2 inputs and if I have prediction horizon of M then I will have 2 M decision variables to choose values for. M values for u1 from the current move till K plus m minus 1 and M values to choose for you to from the current k to k plus M minus 1.

So basically everything else remains the same except that the number of decision variables that we are going to use increases by corresponding to the number of input that you are going to use, so once that is done then everything else is done in the same way, so this is one of the major attractions of MPC because when you go from SISO to MIMO control you do not need much more sophistication the basic ideas simply translate.

(Refer Slide Time: 21:43)



So in terms of a summary the unique features of MPC that help us are use of an explicit model, use for moving horizon and explicit optimisation to calculate control moves this is something we are taught quite a bit about his of the advantages are easy to incorporate constraints I talk to you about, extension to multi variable case is straightforward which I showed you in the last slide and extension to nonlinear problems is also straightforward because remember at the beginning of this lecture I said we can make the formulation however complicated we want right we can take a model which is nonlinear.

So from formulation viewpoint nothing is going to change except that instead of a linear model I am going to say there is an nonlinear model, so we kind of move all the burden of solving to solvers and you have lots of people working on solving different types of optimisation problems, complicated optimisation problems, convex, non-convex problems and so on.

So you can basically take advantage of all the advancements that are taking place in the optimisation area and then use that in solving MPC problems, so in that sense I do not have to generate new theory if I want to solve non-linear problem whereas traditional approaches whenever you go to a nonlinear problem you are forced to develop a new theory and then the new theory will depend on the type of nonlinearity you are looking at and so on and there will be a lot of restrictions in terms of what kind of nonlinearities can be addressed, how do you address them so you will required different tools for different types of nonlinear problems and so on.

So we kind of get rid of all of this, when I say get rid of we kind of push everything to the optimiser and from a formulation viewpoint we have a clean idea of what we are trying to do, so with that this portion on model predictive control is complete and will also have a tutorial on how to implement MPC controller on a simple example. Thank you.