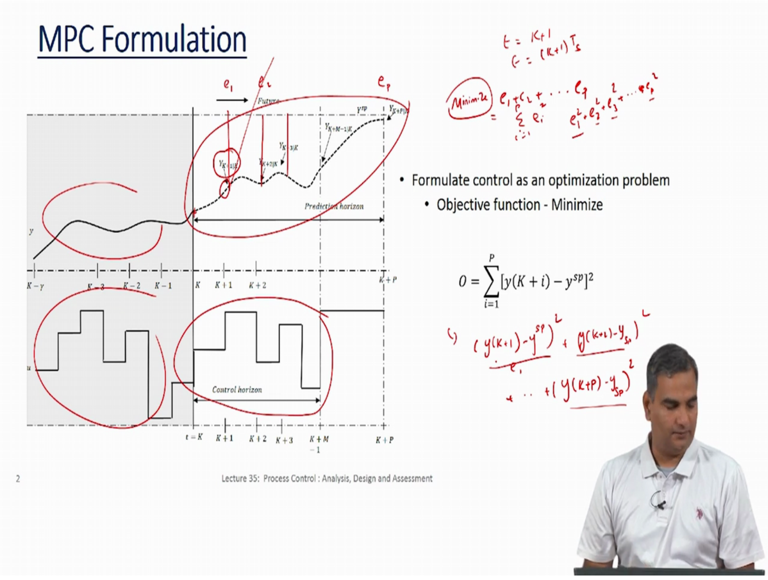
Process Control-Design, Analysis and Assessment
Professor Raghunathan Rengaswamy
Department of Chemical Engineering
Indian Institute of Technology, Madras
Model Predictive Control-Mathematical Formulation-Part-2

Let us continue with our mini series of lectures on model predictive control. In the 1st lecture on model predictive control I talked about the general philosophy of model predictive control and used a picture to explain the basic ideas in model predictive control. And I said there are 3 important ideas, one is the use of the model directly in Controller computations. And the second thing I said was, we introduced this notion of horizon. Instead of just saying that I want my error to go to 0 right away, I look at the error over a period of time into the future.

And then try and see what I can do currently to make sure that the errors in the future are minimised. So when we talk about errors in the future, then it also gives us an opportunity to think about the control moves that I will make in the future. I am not restricted to just thinking about the control move in the current time, so there is a horizon and which I am going to decide what Control moved to make and horizon in which I am going to decide how the errors to behave. So that is the basic idea of model predictive control.

And the horizon and which and thinking about how the error should behave is called the production horizon and the horizon in which I am thinking about what control moves to make is called the control horizon. Now, the final aspect of model predictive control is actually putting all of this together as an optimisation formulation so that the model predictive controller problem can be solved. And when we put all of this in an optimisation formulation, then the solution to the optimisation formulation gives me the control inputs. The control inputs are not written as analytical expressions but they are a solution to an optimisation problem.

(Refer Slide Time: 2:11)



So we were working with this picture in the last class, this is a notion of the control horizon, so this is where I am going to make these moves. And I am going to look at the prediction horizon, which is generally larger than the control horizon. So that I can kind of think about and change the behaviour of the error in a prediction horizon. And the key aspects of this figure, the other key aspects of this figure are that this is the past, things that have already happened and this is into the future and we also said that we are moving away from the continuous framework and we are looking at all of this in a discrete framework. So we have to keep that in mind.

Now, when we were talking about optimisation problems in the last class, I said there are 3 major components in an optimisation formulation number 1 is the objective function, number 2 are the constraints in the formulation, number 3 are what other decision variables. So, if we understand each of these carefully, then we understand the optimisation formulation, okay. Now the solution to this optimisation problem in case where there were no constraints, I showed you in the last class and if it is a univariate optimisation problem, we saw that for a function to take a minimum value, the 1st derivative should be 0 and the 2nd derivative should be positive.

And in a multivariate problem, for a function to take a minimum value, the 1st partial derivative with respect to all the variables should be 0. And the 2nd partial derivative which is written in terms of a matrix called the Hessian matrix should be positive definite if the function takes a minimum at those values, okay. So that is the solution to the optimisation problem in an unconstrained case. In a constraint case, it becomes a lot more complicated and typical optimisation course will talk about that. However, as far as this course is concerned, what we are going to do is we are going to simply formulate the problem and then use optimisation solvers that are available with all software programs that come now, such as Matlab, Sci lab and anything else that you might think of.

So, basically we are going to use the optimisation program to actually solve this problem, we are not going to do this analytically, except to kind of understand how this works and which is what I showed you in the last lecture. So, if you were to now formulate the MPC control problem, essentially we have to take these notions of prediction horizon, controlled horizon and then basically say how does this translate to the 3 components in an optimisation formulation, the 1st component being the objective, the 2nd component being the constraints and the 3rd component being the decision variables.

Now let us start with an objective. So, what I am going to do is I am going to literally say in words what we want in terms of an objective and then I will show you how that translates into an objective function mathematically. So, if I were at this point right here, I said I already have a measurement at Y of K. And so basically what I can do in terms of manipulating or modifying or shaping the error in the future has to start only with time K +1 and remember time K +1 when I write K +1, I have already said the true time is K +1 Times Ts, we are only doing it at discrete time intervals.

So, what I can do is I can think about modifying the error and since I have taken a particular prediction horizon, I can think about what defying all the errors in this prediction horizon. So, if you think about P time steps into the future, then Y K +1 given K, which is a measurement which I do not have yet, so that is something you have to remember because we are doing all of the sitting at time T equal to K. So I am predicting into the future, what the measurements will be. So, basically I need to somehow figure out how I am going to do that for now let us assume that I have some predicted what the measurements will be in the future.

Then I can basically say the error at each time is the reference. So, if I take this difference, that is they are between YK +1 and Y setpoint at K +2 this is the error, K + releases the error and so on, so, I will get these errors. So, you might call in some sense error at 1, which is K +1, error at 2 and the error at the peak future time instants. So, let us say I have these errors and from a controlled viewpoint my interest is in minimising this error. So, somehow I have to collectively minimise all of these errors E1 to Ep.

So, when I try to minimise the 1st thing you can think of objective function is to say let the minimise E1to E2 all the way up to EP, okay, so this basically I can write is i equal to1 to p Ei, right, so this something that I can do. But you will notice right away that there is a problem with this. The problem is for example, if you have something like this, so I have this point and then let us say I go here and then come down here, then I could have positive and negative Es which will make this becomes smaller and smaller, you can even take it to 0.

However the problem will be that each of the errors themselves will not be 0. So, if you are minimising, the other thing is the idea is to take the errors above setpoint in which case all the errors will be negative. So, from a minimisation viewpoint, it is a good thing to keep getting smaller and smaller values. So, in which case the E 1s can become very negative and you will still be minimising this. But your objective is not that, your objective is to ensure that E 1 to P all as close to 0 as possible, so just adding them up will create all kinds of problems.

You cannot really use this as a minimisation objective. So the minimisation objective that people usually use is to say, let me minimise not Sigma Ei but Ei square. So, what this means is instead of minimising E1 + E2 + E3 and so on up to EP. I want to minimise usually E1 square + E2 square + E3 square all the way up to Ep square. So, now what happens is, when we try to minimise, that means the lowest value this can ever take is already known, which is 0 because this is sum of square. And that 0 you will get only if each of these terms are 0.

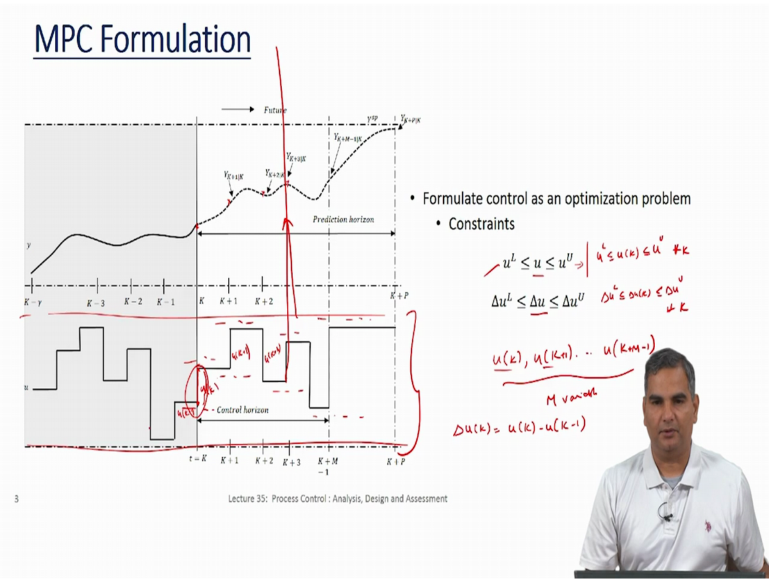
So since no term has the ability to become negative, all of them are positive, the lowest value is 0 and the lowest value is achieved whenever E1, E2, E3 all the way up to EP are 0. So, even if you have one negative, that squared will be a large positive value. Okay, so basically the idea is a set of minimising Sigma Ei, you minimise Sigma Ei square and if you minimise Sigma Ei square, then you are guaranteed, the optimisation objective keeps going down, collectively all the E 1 to Eps are going out, it is not possible for one to become very negative and so on, okay. That is the key idea.

So, if I want to collectively minimise all of these errors, E1 to EP, basically I do sum of square and minimises so that the least value I get. So, this is called the least square problem formulation. So, in this case since we are already at K, the 1st error Ei is actually the error between the K +1th instance in the future and Y setpoint. So, if you look at this, what, if I expand this, what this will give me is Y of K +1 - Y setpoint square, which will be the error 1, the 1st error term + Y of K +2 - Y setpoint whole square and so on.

And the last term will be K + p - Y setpoint whole square. This will be the P terms in the objective function. Now, if this objective function is minimised, then each of these errors are minimised. That means whatever I have predicted in the future in terms of what the measurement of the output value will be, those are actually pretty close to the setpoint because each one of these terms is going to go smaller and smaller. When these terms go smaller and smaller, that means the difference goes smaller and smaller, that means the output is very close to the setpoint.

So, from a control viewpoint, the object of minimising the errors collectively is taken care of if I look at an objective function like that. So, that is the 1st component of the optimisation problem which is basically the objective function. Now, when we look at this, we say okay, this objective function is something that we have defined already, now how do we define the other 2 components of the optimisation formulation.

(Refer Slide Time: 11:51)



So, once the objective function is defined, the top portion of this picture we are kind of done with, now we are going to focus on the bottom part of the picture and the constraints are generally going to be on the puts, there could be constants on the output also, those are more advanced formulation is, which we are not discussing here. So, let us look at the bottom portion of this and this is where the manipulated variables are being shown. So, the idea is from a model predictive control viewpoint, what we said was we want to make these control moves in such a way that the errors are minimised.

So, we have anyways posed the minimisation of errors as an objective function, so that is taken care of. Now what does this move plan is something that we have to see. So, the moves that I am going to make, so remember we are still talking about the single input single output control, so there is only one U. But since there are many control hope that I am going to make, that is I am going to make this formulation identify the values that I should keep for this input variable at different times, I am going to think of each of them really as a variable, so I might say the variables that I have, which I will come back in the next slide.

UK, UK +1 all the way up to UK + M -1. So, conceptually what we are saying is, I have already got the measurement, after I got the measurement, I have checked whether the measurement is close to setpoint or not and so on. So, that is the different computation. After that from a controlled viewpoint, what I need to do is I need to make the 1st control move at time T equal to K and clearly because always control moves take some time to affect the output, this is already there, so it is not going to get affected. So, from an output viewpoint, every output after this move is made is going to be affected.

But since we are looking at discrete formulation, we are looking at outputs at specific time K +1, K +2, K +3 and so on. So because of this input move, Y at K +1, K +2, K +3 and so on will be affected. And what we are saying is unlike the standard PID controller where we will look at this move in a continuous domain, what we are going to do, since we are anyways talking about the horizon, over which we are looking at minimising the errors in the output, will also look at the horizon on which we are going to plan our manipulated variable moves.

So this is UK value and which is as yet unknown, so we do not know what this value is. Only when we know this value, we will be able to predict, that is how we are going to connect these values to the predictions through a model with we will see later. As of now, if you just think about this as variables, so UK is available for which I need to find the value, UK +1 is a variable for which I need to find the value, UK +2 is available for which I need to find the value and so on, all the way up to U of K + M -1.

So, if you take a look at this, there are these M variable for which we need to find the value. And based on these values, the output will change. So, the idea of optimisation is how do you choose these values so that these outputs are as close to Y setpoint as possible, okay. So, that basically means if I make some decisions for this, how will it affect the output is something which I should bring it, that is why the notion of model, then, which we will see as we go along.

But as far as we are concerned, these are the variables and the constraints are written in terms of the decision variables or functions of decision variables as I talked view before. So, you might say if I have a control input, such as a flow through a pipe, we already discussed this, there might be a minimum value and a maximum value it will have. Minimum value if it is low, it is going to be 0 or the maximum value is going to be, when I keep the eyes completely open, what is the flow and so on, okay.

So every value I take care, U has to be between U lower and U up. I have thoroughly written this for the input U, so since each of these variables represent the same U at different times, so this constraint should be obeyed by everyone of these, okay. Though I have UK, UK +1, these are the same inputs but at different times. If the general input itself is constrained between lower and upper, then the input at every time instant should also be constrained between the lower and upper.

So, basically if I want to generalise this and write this, so this is on the U input itself, if I want to generalise this and that this as caskets at every instant in which I make a change to U, so I will write this as U lower lead than equal to, U of K less than equal to U upper and this is the mathematical notation for all K. So, irrespective of what the K is because this U of K is still representing the same input but at different times, okay. So, that is the constraint you have. So, this is written tenderly but you have to write this for U at every time instants, so those are constraints on U.

And similarly when we make a change, so this basically talks to some constant, generally on the U itself, what values it can take, where it should be, so that is one type of constraint. The other type of constraint is at a particular time, how much can I remove this letter you buy. So, if I am here, okay, if I move this year, so the difference is this point was UK -1, this point is U of K. So, if I define Delta U at K as UK - UK -1, so this is where I was at, once I get UK, I compare, so the delta difference is UK - UK -1.

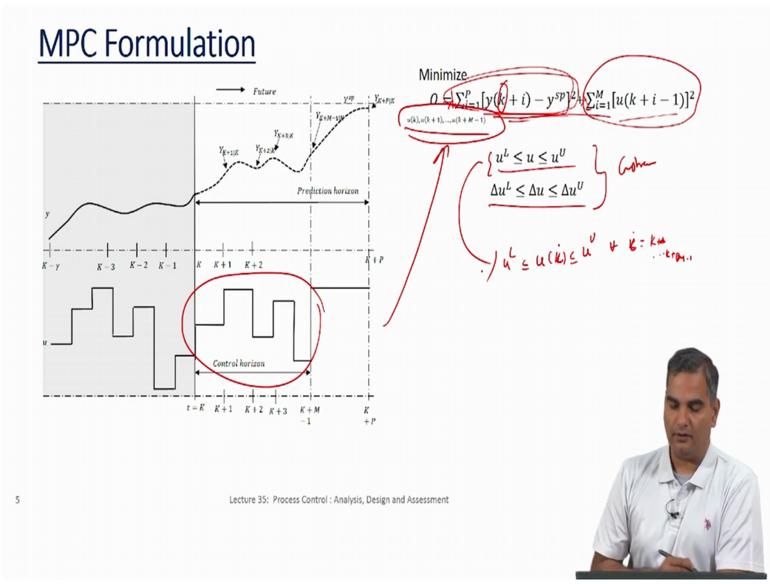
Then what typically happens in control systems is there will be a constraint of how fast I can move by control work, right. So, in a very short time in which I am going to take my manipulated variable to a new value what is a way that I can do this. So, I can directly give a constraint on Delta U. Now, again I have written this as for the input itself, so the delta U change basically tells you how much you can change at a particular sampling time and much like how I got here, here also I have to have Delta U lower less than equal to Delta UK.

And delta UK is always defined as the difference between what value I chose at K and what value it was that K -1. So, if it is UK +1, it will be the difference between, if it is Delta UK +1, it will be the difference between UK +1 and UK and so on. So, much like how I said this is for the actual controlled input and this has to be applied at all UKs, this Delta U is also for the same controlled input but at different times I am taking these deltas. So, this is also applicable for all Ks at different time instances of making the control move, then I will have this Delta U Upper for all K, very similar to what I had here.

So, these are the kind of constraint that we look at when we formulate the model productive control problem. Now, this constraint, I spoke about this, now this constraint says that whenever I make a move, what is the limits that I can kind of achieve, right. So, some limit that I have. So, while there is an overall limit in terms of the actual value itself, that is also a limit, this limit should be applied on Delta U everytime I make a delta U. So, in other words, if I am here, I cannot make a delta U like this, there is a limit on this.

So in this case, clearly the way we have drawn this, the delta U will at least be something like this because all of these values should be within that limit, okay. So, this Delta U limit will be fixed internal or you could make a change over a period of time. So, if this is a fixed limit, then you can say this is the maximum delta U we can take or you could, you could have things where the changes from sampling time to sampling time, if your application really calls for it. But in general this is actually a physical hardware constraint. So you will say this is the maximum delta U I can take.

(Refer Slide Time: 21:24)



Now we have talked about 2 of the components of the optimisation formulation, one is the objective function and the other one are the constraints and the constraints are of 2 types, one is on the range of U itself by the other one is on the range of Delta U. So, once we have these 2, the 3rd component is actually the decision variables and the decision variables are available for which I have to find values and in this case I have to make a decision about the control move plan, what are the control variable values that I am going to have.

So this is the same controlled input but at different times, again just to reiterate. So, have to make decisions about what value this controlled input will take at time K at K +1, all the way up to M -1, these are the M variables that I talked about. Okay, so now you see how the MPC formulation has become an objective, optimisation formulation. From this picture if you understand these equations, then you have basically understood the main concept of MPC. So, the objective function tells you that you are trying to minimise error.

In some cases when you want to generalise this, you might say I do not want to just minimise the other but I also want to minimise the control effort that I take, which is I do not want to take very large control moves U because that takes money and effort. It might be that the objective function is not only the error squared that I have, this is a generalisation of the formulation. But also this U square, what this U square tells me is if I take very large control moves, this will be large, which will be bad for minimising the objective function.

So I want to keep the move sizes as small as possible. So, in some sense what it says is how close can I get to my setpoint without expending too much effort, which is typically what we do in all of the problems that we solve. So, we have a goal, we want to get to that goal at the same time many times we want to see what is the kind of minimum effort that is possible to get to this goal. Not simply because maybe we are lazy, it is because once you minimise the effort toward this goal, you have more time to pursue other goals that you have, that you want to pursue.

Very similar idea in control, so this is your pursuit of your goals which is going to the setpoint at this is the effort that you need to put into to get into that setpoint. So, in some sense you want to minimise both of these, that is you want to minimise how far away from the goal you are and you also want to minimise the kind of effort that you are going to put in. So, this is the objective function. Now the beauty of model predictive control formulation is that, you could just do this part of the objective function, or you could add both of these, or you can wait for these objective functions using weighting factors which will tell you what is the relative importance of this.

If lambda 1 is much greater than lambda 2 and then they importance is I want to be very close to my goal, irrespective of the amount of effort that it might require. But if you say lambda 1 and lambda to our kind of stealing them, so that these terms are equal to each other in magnitude, then the question is, well, I want to get to my goal but I want to have a balanced life, so I do not want to do crazy effort, I want to also kind of balance between how fast I achieve my goal and the amount of effort that I need to put in.

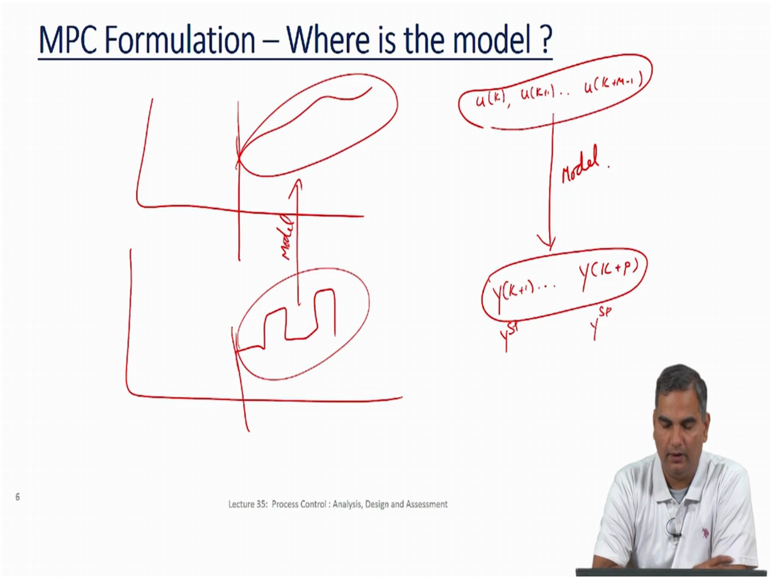
So, once you have these things, then you can start thinking about these objective functions more conceptually and then you can formulate objective functions that you can solve. Ultimately if you are simply going to solve this as an optimisation problem, you could even come up with any kind of nonlinear objective and then say let us solve this optimisation problem. So, what you are doing in model predictive control is you are moving all the efforts into solving this problem to an optimiser and as long as optimiser keeps getting better and better.

So you have better nonlinear optimiser, you have better constrained optimisers and so on, then you could actually formulate any problem you want and the formulation is very intuitive, you know why you are writing each objective function term and the formulation becomes intuitive. And the solution, you do not never worry about because it is an algorithm that is out there which will solve an optimisation solution. So, that is the key idea here. So, this is objective, we already talked about this constants and these are generically written for you.

So this should be converted to UK at every K because the UK also represents the same U, similarly delta UK also represent the same input, so this should be applied at every time instants, as this would be applied at every time instant. So, this is a constant part of the objective. From a control viewpoint, what this Physically says is that the input moves that you are planning should all obey physical constraints, such that this objective is minimised and I have these as the decision variables UK, UK +1 and so on.

So, just to be sure that you make explicit, so when I expand this, I will have U lower less than equal to UK less than equal to U Upper for K. So, I can even say, since I am using K here, so let me use i here, Ui for all i K +1 to K + P, okay. So this is how this is expanded. Similarly you can also expand this and this K in this light talks about the time instant at which we are sitting and then this is all prediction into the future. And these are control moves into the future, UK, UK +1, UK + and -1 and so on, okay.

(Refer Slide Time: 27:26)



So, this is going to be K to K + M -1 because this is talking about the control moves. So, this is how I combine this and put together formulation for model predictive control. Now, in the last slide we talked about the formulation, we talked about the notion of horizon and how all of this are combined into a mathematical formulation. So, the next question that you might ask is okay, where is the model in all of this? So, you have written Y, you have written U, you have written an objective function, and you have written constraints and so on but where is the model in this?

So, as I said before, remember when we talk about this, when I am sitting at time T equal to K, I have this output and I have these inputs I am going to make. Now I do not know the values of this and I have not made decisions yet how about these also. Okay, however the formulation is going to relate the decisions I make to what happens here, right. So, if I want to make decisions here, then I should somehow figure out what is the effect of the decisions on the output, so that I can actually say this will be effective and I want to compare that effect, whatever I get with the setpoint.

So that is the basic idea of optimisation. So, in that sense what I need really is somehow I have to connect what moves I planned to make in the future to what will be the impact and this is where the model comes, okay. Now in many cases we have looked that models which are continuous models but in this case what we are interested in is, the move plan that I am thinking about are UK, UK +1, all the way up to UK + M -1. I want to know how these are going to affect YK +1 all the way up to YK + P.

So, these are future outputs that I am expecting. So, the move plan that I have when I actually implemented this move plan, what will be the output, right. Now, this output I am going to check how close they are to setpoint and I am going to make changes to this so that these outputs are as close to setpoint as possible. And this is where model comes into picture. So, we will explain how model comes into this to complete the whole MPC formulation in the next lecture, thank you.