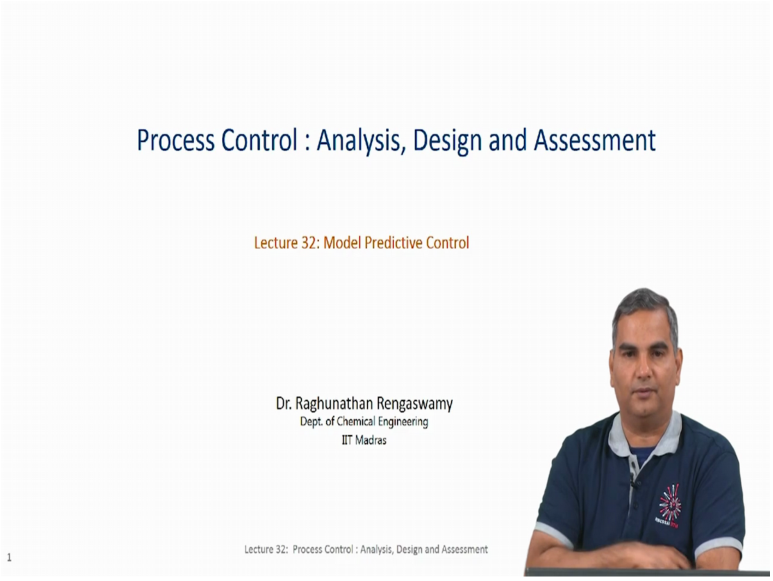
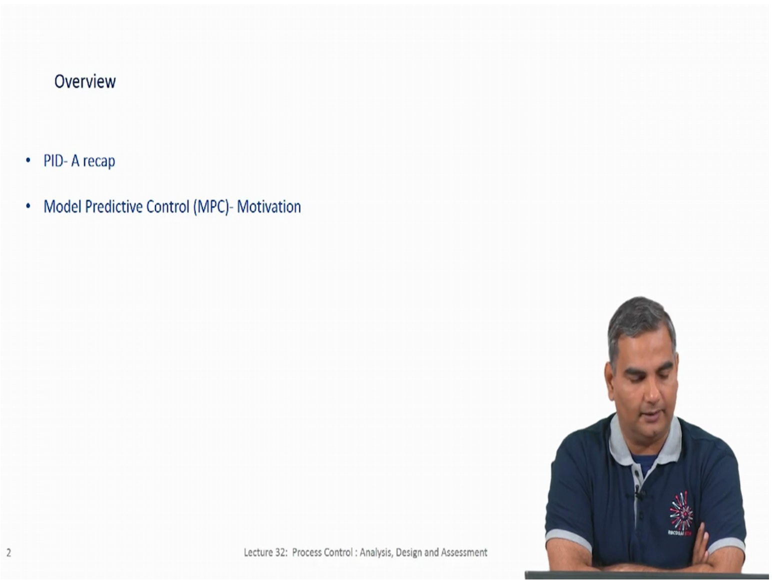
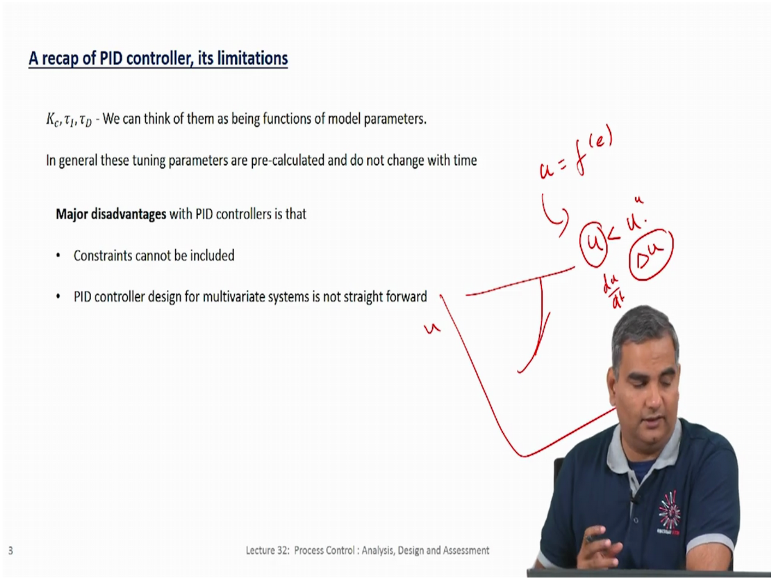
Process Control-Design, Analysis And Assessment
Professor Raghunathan Rengaswamy
Department Of Chemical Engineering, IIT Madras
Model Predictive Control-Part-1

(Refer Slide Time: 0:38)







This is the 1st lecture that I am going to give in a series of lectures on model predictive control. As I mentioned in the last lecture, many industrial processes are nowadays being controlled using model predictive control. And in this lecture I am going to talk about what a model predictive control is and the basic thematic ideas and how it is different from PID control and so on. So, to do that, what we will do is we will give a quick recap of whatever we have learned in PID control. And then basically I will motivate this idea of model predictive controllers.

So we noticed that that PID control is basically a very simple structure for controller, where we look at the error between Y and YZ point and then we decide how much we should more actuate that or how much the control variable or the control input should be. If you are just going to use the error itself for making this decision, then you are going to say it is going to be KC times error. Now, that is basically saying, I see this error I am going to make this change. And if you are also going to look at past errors, then you are going to do an integral Edt.

And if you are going to also look at future, then you are going to do dE by dt and the parameters that multiply the integral term and the differential terms, 1 over tau I and tau d. Now, clearly depending on what process is being controlled, these parameters are going to take different values. So, in some sense you can think of these parameters themselves as being functions of model parameters. You might actually say, look PID is also using the model in some form to come up with these parameters KC, tau I and tau d. Nonetheless, we have been discussing this for a while in this course.

these tuning parameters are precalculated and they do not change with time generally. there also gain scheduled PID controllers and so on. But in general, as far as we are concerned in this course, these do not change. And while these parameters are actually computed as different numbers for different models, they still only encapsulate some notion of the model. those are not directly model parameters themselves. So that is something that we have to keep in mind.

So we saw that as we wanted to control more difficult systems, then we had to go away a little bit from the PID framework, most notably for the time delay control where we have to really use the model already in the controller implementation. So in some sense, while most standard loops, in fact there are statistics which says about 90 to 95 percent of all control loops are PID loops. So the standard basic control loops are still PID loops, because they are very very effective, very simple to understand and very simple to tune and work with and so on.

But there are problems which require little more sophistication and that is where we start talking about model predictive controllers and other types of controllers. And we also notice, actually for processes with difficult dynamics, such as let us say inverse response process, we saw that we still came up with a PID controller only. Right. And if we take the example of the time delay system, we still came up with a PID structure only, except that the error was also including another term, which was calculated from a model.

But still, once this error was modified by using some of the model computations also, on that modified and we still had a PID controller working. So in some size PID controller is a very powerful idea that that is really is… In fact for unstable systems also if you remember, we came up with a PID controller. Right, so basic loops are still PID controllers, if there are difficult dynamics, then there are PID controllers, there are things that you do to modify the PID controller, to make it work better. Same with cascade control, cascade control again you will have 2 controllers, one P and maybe outside PI controller.

And feedforward controller is basically again going to be a PI or PID controller with some input from the model for the disturbance, okay. So in that sense even the difficult dynamics, you can think of as modifying the PID controllers with more structure, more model information and so on. However if you really want to address this as a multivariable controller problem, then either you have to do some kind of decoupling, which can become complicated after a while because the more decoupling you do based on these models, then they want us to worry about how certain these models are, what happens to the uncertainties and so on.

So in general one could assume that PID controller design for multivariable systems is not straightforward, so you have to really be careful about how you design these controllers. And the other rig disadvantage of PID controllers is that we cannot include constraints in the control, okay. this is the 1st time I am using this word constraint, so I want to explain what this means. So, when we do these computations with PID controllers, we compute U as some function of error and whatever value we have for U, we assume, that the manipulated variable can take.

So we never think about what is the maximum value U can take and this is because after a while, though we try to connect the physical system with control and so on, after a while we start getting very accustomed to looking at these variables and then say things work all right. But really I think about flow control, you have let say input flow of water, now based on the pipe dimensions and based on your control valves, the going to be various constraints. Based on the pipe dimensions and the pressure and the pipe can withstand and so on, there is definitely going to be a maximum flow that you can get.

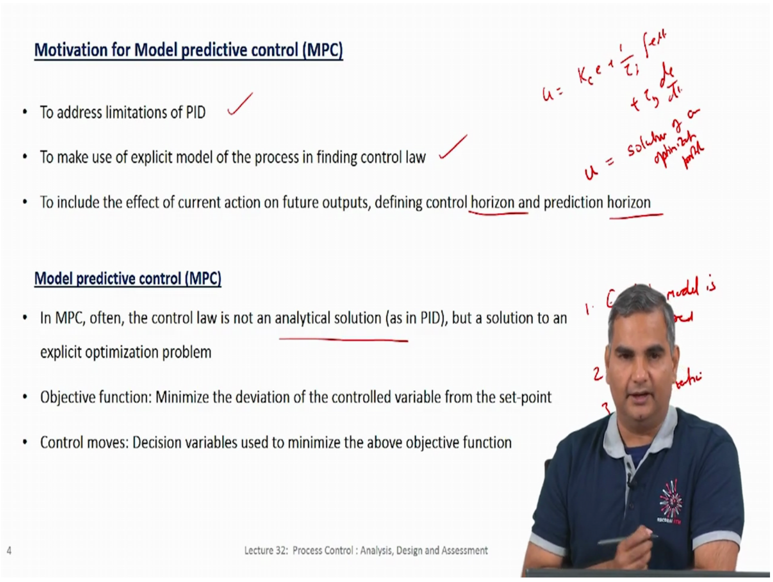
So basically the U itself is going to have some upper bound value, okay. So if it is flow, so you cannot have more than this flow in a pipe, this is something that we intuitively understand. But this is never factored into the PID controller. One way to factor this is to say if the PID controller computes an U which is much greater than the maximum value this U can take, basically just keep it maximum value open, right, and then do not do anything. Now this is what is called clipping in PID control, however this can create all kinds of problems which we will see later.

So it is actually better to explicitly include this constraint in the controller design, rather than saying let me compute controller value for U and then basically decide whether that value can be taken or we need to clip that value and so on. So that is something that could be handled better, more explicitly. So this is 1, the other thing is when we talk about controllers, when we talk about let us say I have U profile like this, so basically it is not only the upper limit on U, okay, that is of importance, it is also rate of change of U. So what does rate of change of U mean?

Supposing lets you think about a standard pipe that you use, so you kind of open and close the tap, right. Now if I asked you to get the tap to be completely closed and completely open, the next second, right, nothing about this. Supposing that you had to really open the tap, you might not be able to do that in a second, right. So the rate at which you can change the U is also limited, okay. So a completely close by, if you want to open the tap to fully open, the if you are physically doing it, you know depending on how many rotations you need to do and so on, it might take you 5 seconds, 10 seconds or whatever it is.

Whereas the PID controller in its computation is not taking this into account at all. What is just saying that U is this and then the U profile is this, that basically says dU by dt can get to values that are not feasible in real time. So that is also a constant on dU by dt or I am going to call it delta U because they are going to move onto discrete formulations quickly. So this will also be constraint, so within a short period of time, I can only change U by this much, right. that constant is never included in PID design.

(Refer Slide Time: 9:39)



So, that most of controller, you trains for the manipulated inputs and how you include them in a controller is a very important question. And of course PID controller was never designed originally to take this into account. So the idea of model predictive control is 1st to address the limitations of PID. the basic limitations of PID are constants cannot be used and number 2 is that when multivariable processes are there, the deriving this PID controllers become more complicated. So model predictive control you can think of as being thought of to address these limitations.

Now model predictive control is also going to explicitly use a model of the processor in finding the controller. Now, as I mentioned before in a typical PID controller, though we can think of this KC, tau d and tau I as being derived from the model parameters, they are not model parameters themselves, they are not the models themselves. So, explicitly immortal is not being used in finding a controller. Whereas in model predictive control, we will use the model of the processed directly in finding the controller.

And then in model predictive control, we are going to start introducing the notion of horizon. Okay, there is going to be controller and its horizon, prediction horizon. We have never talked about this till now, I just will allude to this here and this is going to be a key difference between PID and model predictive control. So in simple layman terms to explain this now, if I am looking at output for a PID control at a particular time, I am looking at the difference between that output and the setpoint at that time, I call that as error.

And all the decisions I make are based on the current error, right if it is K. Or in other cases, it might be based on my past errors, it is based on, if you do dE by dt, it is based on the rate but you can think of if you do a backward difference, you can think of the change in the output between the last time and the current time. So in other words, everything is based on whatever errors that we have seen and the current output and how far away from the setpoint it is. Whereas we might actually think about not only the error now but what is as likely to be in the future if I make a control more currently, right.

So that notion of horizon into the future, that we can think about from the control viewpoint. So this is something that we always do in our lives. Basically no decision is taken only based on all that has happened before and what you have currently. We always take decisions based on you know what we think is optimal thing for us to achieve. So you know you might have heard this in most interviews, they will ask you very want to be 5 years down the line or one might ask what skills do I want to get you know in the next 3 or 4 years.

So all of this have a notion of horizon into future and it is not only, okay, I want to prove myself today, right, so if you want to learn let us say musical instruments, you cannot say I want to learn it today, you know where will I be with this instrument 3 years down the line, for years down the line and so on. So that is the kind of future outputs in terms of performance matrix that we want to decide, right. And how do I include that in a controller design, that can be done using model predictive controller, using this notion of horizon, which we will see. Okay.

So, now another very interesting thing is that. Since we kind of restrict ourselves to looking at the error currently and the past errors and maybe just one error, next error or one before, when we do the derivative terms, basically the control action that we are planning for is only at the current time, right. So we only get a current control move U, okay. Now the error is based on past and dE by dt I need but the control move is only based on U. But again pink about how we will look at this in real life.

Supposing you said, I told you this interview question, right, so, they say where will you be five-year term the line. If you said fiver down the line I will be so-and-so, then in your mind there is always this question of if I have to be so-and-so in 5 years, then I have to do these things in the next year, this thing in the year after that and so on. So while you set yourself targets for a certain number of years and say this is what I want to achieve, the effort that is needed to achieve that is also being planned at that current time itself.

It is not as if, if I have to be so-and-so 5 years today I do this. that is not a decision you take, right. So you take okay, I have to do this, you have to learn this, and so on. So you are also deciding about what you need to do at that current point and little bit into the future so that you can be so-and-so. Clearly you will not plan for what you should do 5 years after your time because your goal itself is a five-year goal. So maybe you will say, look, if I have to be this in 5 years, at least the next 2 or 3 years I have to do this, this and this.

So you are already planning for the next 2 to 3 years. So that again brings in a notion of horizon. So, the control move your planning, not only the current control move but you are also planning into the future. And based on these control moves into the future, what is the output that you are going to get and that is also into the future. As all of that is a function of horizon. So, model predictive control enhances the framework of PID by looking at constraints as we talked about in the last slide, looking at using the model explicitly in finding the controller and then also this notion of horizon which is what we will talk about and then I will show you mathematically how this is done in model predictive control.

Now model predictive control basically solves an optimisation problematic every time. So this optimisation problem is solved and the solution to the optimisation problem itself becomes the computation of control moves or U. So, in that sense it is not analytical solution like PID, red, PID is basically you will say K, KC Randall plus one word tau I integral Et plus tau dE by dt. So this is an analytical expression, whereas U in the model predictive control is a solution of an optimisation problem, okay.

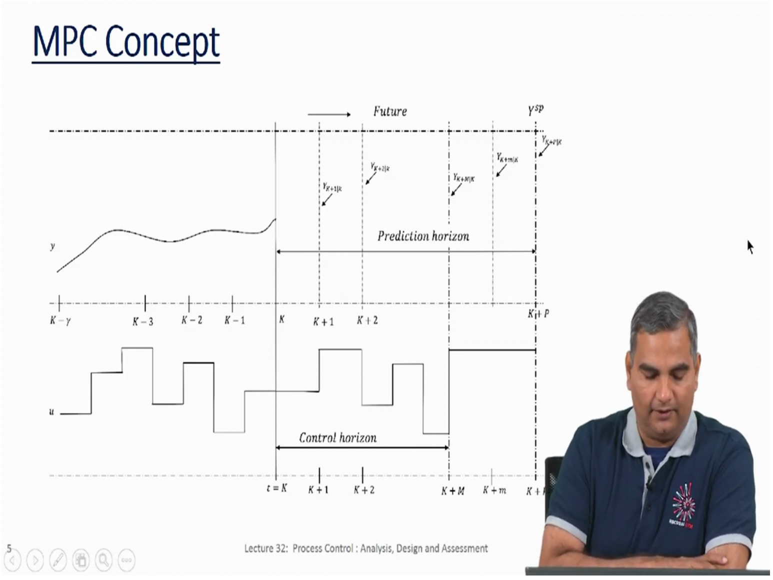
Now in the linear case, people have shown that you can actually show that even in an optimisation solution you can get some analytical forms and gain of that analytical form changes over a period of time and so on. So those are more advanced concepts but as far as we are concerned in this undergraduate control class, you just need to know that it is a solution to an explicit optimisation problem and not analytical expression like this which is used in PID.

the minute we talk about the solution to optimisation problem, we will describe this in little bit more detail in the next lectures, but when we talk about an optimisation problem, then admittedly we have to talk about what is called an objective function, we have to talk about the constraints and we have to talk about the decision variables that are used in the optimisation formulation. So we will describe these in detail as we go forward. But the key difference that I just wanted to point out is that the U itself is computed as a solution to the cupboard to an optimisation problem.

So if I want to summarise this line, then there are 3 main features of MPC which are different from PID. the 1st feature is that an explicit model is used at all times, okay. And now we are started with PID which is analytical expression, then we looked at as we kept learning more and more, we started using more advanced concepts, we saw how the model starts creeping in within the PID framework. Now MPC is completely abandoning the PID framework and directly using the model in control, so that is one big difference.

the 2nd difference is the notion of horizon which is I am not going to make plan for a control move only now but I am also well to make plans for control moves into the future, that is something that we talked about. And the 3rd important difference is that this is going to be an optimisation problem that is solved, which is the solution to which we control move. And it is not analytical expression like what you see in a PID control.

(Refer Slide Time: 18:40)



So we will look at the concept of MPC in more detail here. Now there is another change that we are going to introduce before we can formulate the MPC concept. Which is basically till now we have looked at all transfer functions and all variables as continuous variables, Whereas we are going to start looking at this notion of discrete variables when we talk about MPC control, I will explain that in good detail as we go forward.

So now that I have kind of explained the differences between MPC and PID, I will come back to each of these ideas in more detail and show you how we can formulate MPC control for 1st single input single output systems and then show you how you can extend that very easily to multivariable systems. So we will pick up from here in the next lecture, thank you.