**Course Name: INTELLIGENT FEEDBACK AND CONTROL**

**Professor Name: Leena Vachhani**

**Department Name: Multi-disciplinary primarily for Mechanical, Electrical, Aerospace and Chemical engineering streams**

**Institute Name: Indian Institute of Technology Bombay**

**Week - 03**

**Lecture - 13**

Hi, in this video we will look into overview of control techniques. So far, we have seen that where we will apply proportional, proportional integral and proportional derivative and PID controls, different kinds of structures and so on. At the same time, we have seen feedforward technique and where we can benefit from feedforward technique as compared to the feedback technique. So in this case, we have six industry standards that one can follow. These are cascade control, mid-range and split-range control, control by adding some nonlinear elements, ratio control, selector control, and feedforward control.

In this lecture series, I have covered feedforward control in a bit detail, but rest of the five types of the control structure, one can look into and look into what are the different benefits and limitations of each and every method. By looking into the nonlinear structures, one can add into the control loop actuators and sensor compensators. Instead of completely working on the nonlinear structure of the system, one can compensate and make the system behave like a linear system. Similarly, one can look forward for adding limiters in order to take care of the actuator saturation limits. For example, you are working with motors.

It can work within certain range of input values. And if your controller is generating commands to be given which are beyond this range, one can look forward that, okay, these limiters take care of the range in which the control input is to be given. Similarly, one can see that when we have to apply the ratio control, it is at ratio of certain outputs to be controlled. When the objective is the inputs to be given in a particular ratio, then one is introducing nothing but a nonlinear element into it. So purposely we are adding certain nonlinear elements in order to take care of the objectives of the control that is expected by an application.

One can also look forward for controllers with logic. For example, we say that if this condition is satisfied, then to apply this particular controller, if otherwise do something else and so on. So when such kind of logic is considered, one can go forward for a fuzzy logic way or a switch systems way to make sure that things are in place. So in one way, we can say that, okay, these nonlinear elements that we are adding into the control system in order to simplify the controller task. Again, we will be looking into certain cases where the controller is having the PID structure itself.

But in order to enable this controller to have a PID structure, what else is needed? And this is where you might have to add certain more nonlinear structures into the control loop to satisfy that the controller is a simplified one. Our objective is a simplified controller designs in order for the implementations to be very simplified and one should be able to implement it even in the embedded systems and work with very nice conditions, working with different environment conditions and objectives are being satisfied there. In this regard, we will understand what is the difference between when to apply the gain scheduling method and the adaptation methods shortly. So in order to understand that, let's see what kind of system characteristics exist for any kind of system.

So we have been talking about LTI systems, which is linear time invariant systems, in order to see that how the controller is theoretically being analyzed and the benefits of it can be given in terms of the guarantees and so on and so forth. So the study of the LTI system has helped us in understanding the behavior of the controllers, so to say that. But in practice, none of the systems is actually an LTI system, but the behavior can be mimicked as linear time invariant. So as long as the system is more or less LTI, then a small non-linearities or some changes can be considered as disturbances, and that's the reason controller is working. But there is a possibility that the system is primarily behaving like a nonlinear system.

Then can I still use PID systems and so on is what we will discuss it in terms of the methods that exist in order to enable the controller to have the PID control kind of a structure. Similarly, if there is a large scale system, then can I use PID? If it's a multi-input, multi-output system, can I use PID? So let's try to understand the difference between these LTI systems, linear, non-linear, large-scale, and MIMO systems, and accordingly we can simplify our designs as the controller is simplified. So LTI system, as I mentioned, these systems have properties that do not change over time, and their response is linear.

It means we will be able to apply the principles of superposition theory. This we have said in the beginning itself, and we studied in last lectures about the LTI systems itself. For example, we have the circuits like RLC systems. We represent the system or sometimes give analogy in terms of the spring mass damper system, similar to the vibration control example. Linear filters or heat exchangers are other kind of LTI systems.

Now comes to the nonlinear system. These systems, the output is not directly proportional to the input. This will come as a phenomena. This may be because there are harmonics, there are bifurcations, or the system itself is very chaotic. So this relationship between output and input cannot be said in a linear way.

And that's where we say it is some function of the input and states, but which cannot be represented as linear systems. More importantly, the principle of superposition theorem cannot be applied here. So these systems are either we can consider in terms of electronics, diode or transistor circuits. These are large deflection pendulum system, chemical reactions or, in fact, the robotic systems. There comes large scale systems.

And large scale systems means these systems are described by a large number of states. Large numbers of states means we are talking in terms of thousands, ten thousands and so on and so forth. Now, in order to control these systems, can I still use the PID controllers? We'll look into that and under what conditions can I use the PID and what conditions we cannot. We should not attempt at all.

So, such large-scale systems, some examples are electrical grids, railway networks, power plants, financial market, and many other things. There comes MIMO systems, multiple input, multiple output. So you see that what we are describing it as the difference between large scale and MIMO system. Large scale, we are saying that it's a possibility that there's only single input and single output, but the number of states for the system is very large. Whereas when we say multiple input, multiple output systems, the number of inputs are definitely more than one.

Number of outputs are more than one, either or. All right. So this will allow complex interactions between the multiple inputs and outputs that you have considered. And it may increase the capacity of the system. It represents this increased system capacity more or less.

So these set systems are wireless networks, process control blocks, automation systems, flight control systems where multiple flights characterizations need to be done. So the input from different flights are being taken care in order to give certain outputs and so on and so forth in order to solve the scheduling and routing problems and so on. Let's see linear versus nonlinear systems and what could be the ways to simplify the system or understand the system dynamics in order to, again, design the controllers in a PID way. So there are two ways to look at, okay? So if my process dynamics is varying, means we are talking about nonlinear systems, process dynamics is constant, then we can use a controller with constant parameters.

So this more or less is talking about linear systems. And this linear systems I can have, I design my proportional gain or integral gain or the derivative gain for a particular controller configuration, then those gains remain constant over the period of time. Unless there is a significant model change happens, these controller gains are constant. Now, what happens if the process dynamics is varying? So, again, what we can consider as two cases when the controller itself has varying parameters, okay?

We will have to design the controller with varying parameters, means that if I am designing with proportional integral control, then proportional gain and integral gain itself is varying with time or some state value Now, if these variations are unpredictable variations, then we will use the adaptive control. Means we will more or less change these gains on the go. Means when the system is running, that time we will make the decisions on changing the values of the gains. But if these are predictable variations, then we will use the gain scheduling.

And that comes from the fact that these predictable variations can happen if I am able to define the process dynamics, even though it is nonlinear, that relationship to certain extent, that particular model is not changing with time. Okay, then what are the ways to do this gain scheduling way? Let's try understanding the gain scheduling method now. This particular, as we said, the process with characteristics that are, though it is nonlinear, but it changes with time or changes with operating conditions. Then we will have the method called autotuning.

And this autotuning is more or less being described in terms of a table. So I have a controller and a process. The controller gains are coming, are read through this particular table. Now this table is more or less like a lookup table, such that I have the input to the table is either the current state or some other combination of the condition of that particular process. And that particular condition is given by some combination of the states.

And that combination needs to be identified, of course, in order to put those values into this table. So the controller gain as we see in this block is getting changed depending upon what is the state of the processor. This means I can consider this as an operating range as well. So in terms of the practical way of implementing it, if I have particular operating range, okay then this particular then controller gain set 1 is to be used if I am changing the operating range then controller gain set 2 should be used, and something similar.

So this particular operating range so this table this is one way of looking at it this table has the input as the range values operating range values and the output is your controller gains. And these are KP, KI, and KD So range one, you use some values. Range one means let's say we are working with DC motors. So one to two volts, you consider these values, two volts to three volts, you consider these values and so on.

DC motor though is a bad example because more or less it's a linear system. And so this KP, KI and KD values do turn out to be all constant values. All right, but to understand it that way. All right, so what is the operating range that we are talking about? Now, input to this table.

Now, this is defined by scheduling variables, more or less. What is the schedule that we are going to consider in order to change this control? That schedule may be dependent on time, it may be dependent on the state combinations, as we are talking about, or input combinations. This particular scheduling variable which is like the input to this table is to be identified and this is what the trickiest part of gain scheduling. It should be selected such that it correlates well with the process dynamics.

It can be measured signal, it can be control signal or an external signal as well. Its range is quantified into number of discrete operating conditions. So I gave an example of operating range of DC voltages, which were only dependent on the input values. But at the same time, it can be some values of some conditions of measured signal, means the output signal, control signal, or an external signal, some other system that is controlling, that is affecting the process. But these signals that we are talking about or a combination of these signals needs to be quantified into number of discrete operating conditions, discrete steps, so that I can give it as an input to the table and I can consider this forming a lookup table.

Now, the controller parameter are then determined by the auto tuning methods, automatic tuning of this. One can also use some learning methods later on, we can think about doing that. Controller parameters are stored in this table for all operating conditions. Now, these operating conditions that we are saying, first is we discretize it. These operating conditions are then identified with the help of measured signal, control signal or external signal or combination of these.

All right. Given this particular idea about the gain scheduling, we have one method which is very promising method for gain scheduling. At the same time, it does some kind of control using PID of nonlinear systems. Let's try understanding this. This is called iterative feedback tuning.

And this is a method in terms of getting the PID control values with the help of optimization methods. And at the same time, this is an online method for adjusting controller parameters. In the previous example, we were considering a lookup table, whereas here it is doing the adjustment online. And let's see how it tries to do it. It's underlying principle is computing the gradient of controller errors with respect to the controller parameters.

So what are those errors and finding out the gradient of these so as to reach to the local minima and so on and so forth. So we consider forming this as an optimization problem where we consider that there is an objective function This objective function is some function of output y of t and u of t over the period 0 to time, some period, time period t. Here we will consider that the controller is PID control, and that's how we will try to address this, finding the minima of this particular objective function. Let's see how we find it out.

At the same time, we know that this is a nonlinear system, but we will consider that a particular operating range, the system is behaving like a LTI system for short duration of time or short interval of states operating range. This system is behaving as an LTI system. All right, so when we try to solve this particular objective function, we'll try to find the loss function with respect to the controller parameters, KP, KI and KD. So given this particular output Y, which is nothing but a negative feedback system, the output Y can be presented in the form of inputs YSP, disturbance D, and the noise N. This is a straightforward transfer function way of getting the output return in terms of various three different inputs.

Here, our error is y minus ysp. This is what we have been doing it from the beginning, just to repeat it, repeat the variables over here. We have the controller C here, the system P, this is my process block, and this is my controller block. I'm considering negative feedback, so this is a subtractor, which is giving me an error signal, and the input is YSP. This is YSP, output is Y, I consider getting the noise at the output.

I consider disturbance acting at the input to this process plant, and this is my error E. So this equation turns out to be nothing but a simple transfer function way of representing output with respect to three different inputs, YSP, disturbance D, and the noise N. What we will do now is in order to get the gradients, we'll find dou Y by dou C, which is nothing but the partial derivative of Y with respect to the controller gains C. C is, you remember, these are three inputs, three gains we have to consider, KP, KI and KD, and that's why we say partial derivatives in terms of E. Similarly, we'll find dou U by dou C in terms of E, all right?

Now, the major issue to solve is that error E is known, but the process P is unknown. When we are trying to find these values, dou Y by dou C and dou U by dou C, and especially when we write it in terms of E, what we have is P is unknown. I can measure error, but my process P is unknown because I have the nonlinear characteristics of the system. All right. And that's the reason we are going by the optimization way.

And that's the reason we will come up with an algorithm which is iterative. OK. So here is one empirical method, experimental method to find to overcome this particular issue. What we will do here, we'll do first experiment, which will store output value Y1 for a control error signal E1, all right? So if I have E1 and the output Y1 is available, then I compute

And at the same time, what we will do in the second experiment is I will select set point as E1. As the previous case, I was having control signal E1, I record Y1 for some YSP. But now we will consider set point as E1 and store the corresponding E2 and P2 values. What is the advantage that we get? We are able to write this Y2 for the second experiment in terms of E1, D2 and N2.

In the second experiment, my disturbance and noises could be different from the first experiment. But at the same time, my set point value is E1. Now, this E1 is nothing but dou Y1 by dou C. All right. So, that is the reason. So, this particular E1 by P is given by dou Y1 by dou C.

And that is the reason I am able to write it in this way. At the same time, E2 can be represented again in terms of E1. Because my YSP is E1 here and D2 by D, while E1 is coming from my first experiment dou U1 by dou C. So this is what 1 by PC, 1 by 1 plus PC times E1 gives you dou U by dou C. We are talking about getting this particular value of dou y by dou c and dou u by dou c in order to get the objective function values and the gradient of the objective function in order to calculate the gradients of this optimization function.

So what happens is now, as soon as I write in terms of the experimental values y2 and e2, which we have recorded from the experiment, now we are able to calculate this gradient dou y1 dou c and dou u1 by dou c in the following way. If my disturbance, if I consider the negligible disturbance and noises, I can consider dou y1 dou c approximately as c times y2 and dou u1 by dou c approximately same as e2. Now, if I look at the previous equations, what we have is if I have to calculate mathematically this dou y by dou c, I needed the process variable P. which now experimentally I am able to calculate with the help of these two experiment, this dou y by dou c is is equal to c times y2 because y2 is measured quantity c is my kp, ki, kd gain for a particular instance I will be able to get c value, this dou y and dou c value and now see how the iterations happen what we will do now This particular IFT algorithm iterative feedback tuning is we will consider this fixed duration experiment.

We will consider storing Y1 and corresponding E1. The next step was second experiment. With the same duration, we will put this previous error value E1 as the set point E1 and store Y2 and E2. With this, we are able to calculate dou Y by dou C, dou U by dou C, as we have seen this. We are able to find UK and YK, which is for a given set KP, KI, KD, this is what my input values U and output values Y are.

And so I will be able to calculate dou J by dou K, which is in terms of if I start writing in terms of dou Y by dou C, dou U by dou C terms. Once I have this particular gradient value available, then I will modify the controller parameter using this controller and repeat this experiment. Keep doing it till the gradient is sufficiently small. The gradient is sufficiently small means I have reached to the local minima of the objective function, means for a given output controller values, what is the minimum error that can be achieved by adjusting the gain values Kp, Ki and Kd, which is given by this particular C vector - KP, KI and KD. And this way, this iterative feedback tuning has allowed us to do the changes in KP, KI, and KD, depending on the values of the output value Y and at the same time, what is going on in the process dynamics.

Here, we did not identify the process and tune the controller values. It went into doing this experiment multiple times and setting up the controller values such that the error goes to zero. This way of doing the iterative feedback tuning is applicable to nonlinear systems and how often you have to do it on the go depends upon how much is the nonlinearity in the system or how much the conditions in the operating range changes, how frequently the operating range changes is what we can consider for applying this algorithm repeatedly. Since this is an online experiment being done, one has to make sure that, okay, first attempt is to tune the KPK and KD and then getting the performance output of the controller. The reference for the, especially for the IFT is available in this book by Marlin.

One can refer for more information on IFT. Thank you.