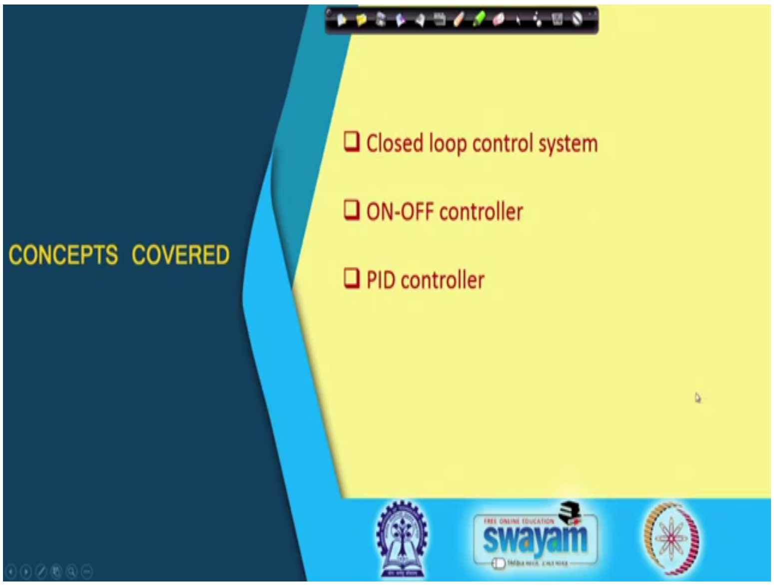
Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 30
Design of Control System

You have seen through the various demonstration experiments that we have discussed so far, that how we can connect various kind of devices with your microcontroller and do something. Like for example, you can turn on or turn off a light, you can glow an LED, you can turn it off depending on so many things. There are so many kind of sensors, you can read some parameters from the outside world and using actuators and output devices, you can output some data to the outside world.

Now in this lecture, we shall be talking about something called Control Systems very briefly. The basic idea about control systems and why it is important in the context of embedded system design ok.

(Refer Slide Time: 01:15)

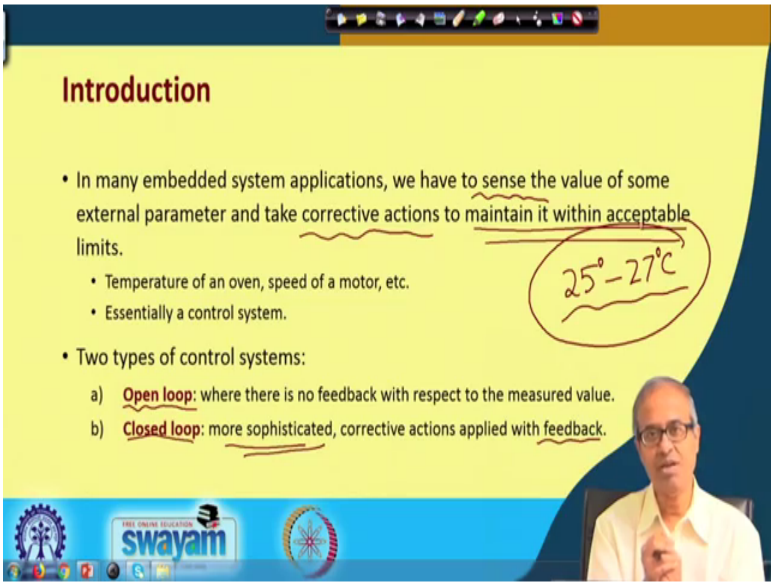


Now in this lecture, we shall be talking particularly about something called closed loop control systems and the different kind of controlling mechanism like ON-OFF control and PID control mechanisms.

Now, let me tell you here suppose I am interfacing a heater with a micro controller, I am turning it on well if I feel hot, I turn it off well if I feel cold again, I turn it on again. But, suppose I want to automate this process, I want to design my system in such a way that whenever the temperature falls below a certain level, let us say 25 degree centigrade, if it falls below I should turn on the heater, if it is above I should turn off.

But, for that mechanism there has to be a feedback in place like not only I should be able to control my heater, I should also be able to read the value of the current temperature. So, I should able to know automatically whether my temperature is lower or higher, that is the notion of a control system or a closed loop control system that comes into the picture.

(Refer Slide Time: 02:41)

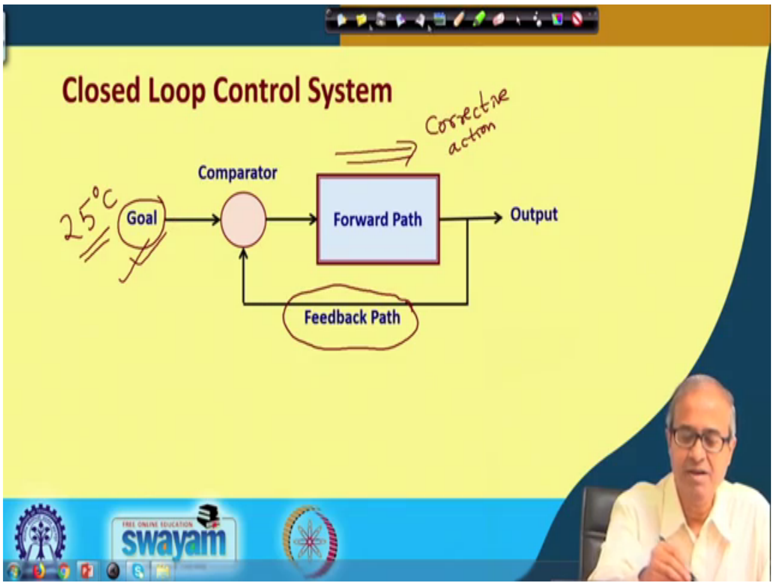


So, what we are saying is that there are many embedded system applications like the one I just now talked about we have to sense the value of some parameter. In the example, I took the parameter was the temperature it can be humidity, it can be anything, it can be pressure, it can be anything in fact ok. Now depending on the value you have sensed, you can take a corrective action. Why? To maintain it within acceptable limits.

Like let us say, for temperature you may say that my acceptable limit is 25 degree to 27 degree Celsius, this is your acceptable comfort zone, if you find that a temperature is falling below it you should turn on the heater. If you find the temperature is going above it well, if there is a air conditioning mechanism, you should be turning on the air conditioning mechanism, this is the notion of sensing and feedback I am talking about ok. So, there can be many applications you can think of temperature of the room or an oven, speed of a motor, there are many application where a motor has to be rotated at a particular speed revolutions per minute RPM. You have to read the RPM value and you have to appropriately control the motor.

So, that RPM remains at level, these are examples of control systems; broadly speaking control systems can be either open loop or closed loop. Well open loop means there is no feedback mechanism that you are only turning on or turning off the heater, but you are not reading the value of the temperature, but closed loop says that not only you are controlling, there is also a feedback mechanism. Naturally, this is more sophisticated because, based on the feedback you can send some corrective signal. Let us say here, to maintain the temperature within this range, you have to send the heater control accordingly ok. This is the basic idea.

(Refer Slide Time: 05:09)

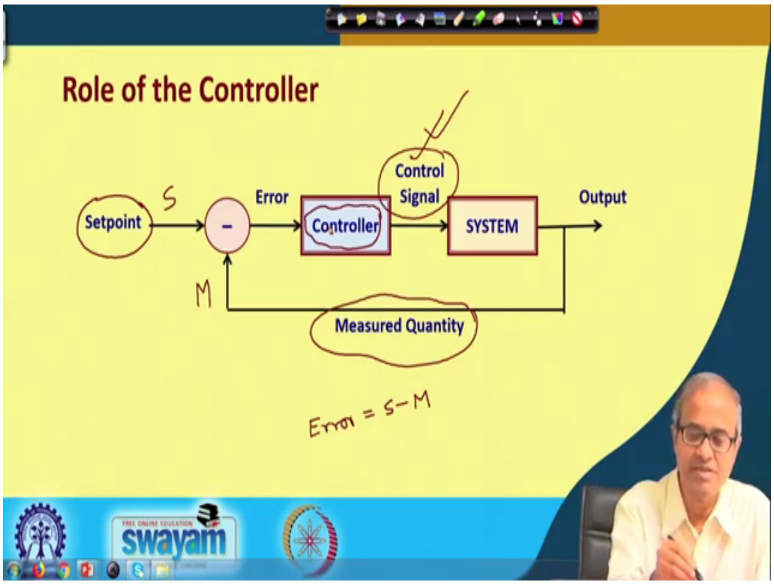


So, let us look at a diagram of a typical closed loop control system. What does this diagram contain? It contains well here, I am not talking about in a specific example of a temperature or pressure anything, it can be any kind of physical parameter. Here we start with a goal, goal means we want to maintain the parameter at some particular level for temperature let us say, I say it is 25 degrees since Celsius this is my goal let us say.

And from the output let us say the room where I am sitting, there will be a feedback path; that means, there will be some sensors, which will be reading the temperature and it will be sending me back some feedback value. There will be a comparator, which will be checking whether this feedback, whatever you are reading is less than or greater than the set goal depending on that you will have to send some kind of a corrective action along the forward path corrective action.

So, that the output value of the parameter is as close as possible to the set goal, this is the overall you can say purpose of having a closed loop control system, let us be little more specific.

(Refer Slide Time: 06:49)

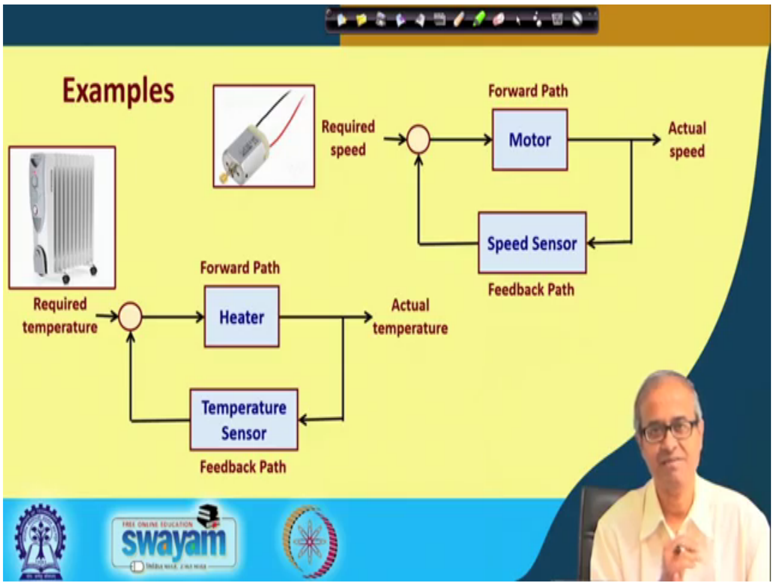


Let us expand the diagram a little bit more and let us talk about the goal here, we are calling a set point same thing we are setting a level, where you want to maintain the value of a physical parameter to and the output through some sensor we are measuring it.

Let us say my set point was S my measured value was let say M, let us say I define my error to be S minus M. So, how much different it is? Whether that equal if they are equal it is fine, I am already I am already quite happy because, my output is just at the set point, but depending on the error whether it is less than or greater than, there will be a controller which will be sitting here, controller will take a decision that how to generate a control signal for my system.

So, my system here is my heater or AC machine or whatever or motor whatever I am trying to control. So, there is a controller my microcontroller will be acting as this controller mechanism in software that will take an intelligent decision, how to generate the control signal so that my output is as close as possible to my set point this is where the role of the controller comes in. So, there can be various alternate designs of controllers we can think off.

(Refer Slide Time: 08:35)

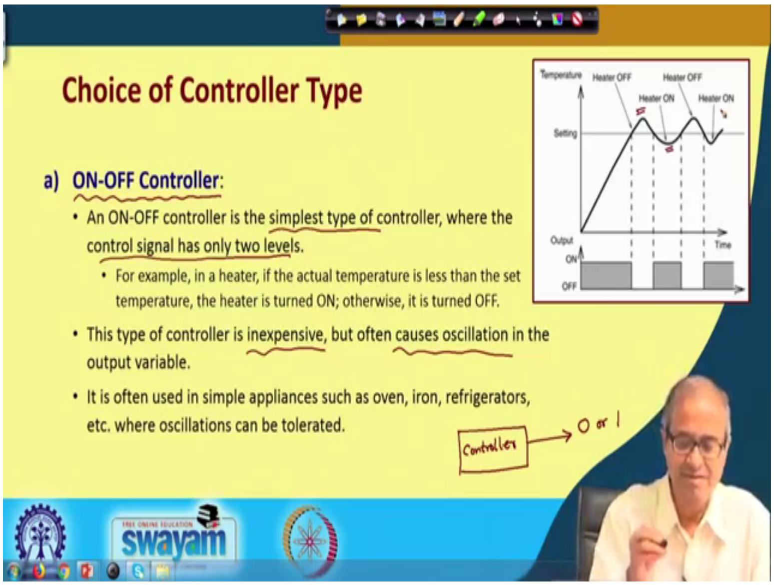


First let us look at some examples, this is a very simple example of a room heater suppose, I have a room heater where I have set a required temperature. Well here I am assuming that I am in a place which is quite cold well, if I will means if you are residing in a place which is hot you cannot appreciate this example, suppose I am staying in a place which is very cold.

So, I need a heater room heater. So, there will be a temper sensor in the feedback path the actual temperature will be sensed the comparator will be finding a difference required temperature minus this and in the forward path the controller will be controlling a heater ok. Take another example suppose, I want to control the speed of a DC motor here we are showing a small DC motor. So, some required speed as specified let us say, I have specified 100 revolutions per minute 100 RPM a slow rotation.

So, there will be a speed sensor in the feedback path, it will be sensing the speed in some way we shall see these experiments later depending on the difference the controller will be activating the power supply of the motor to turn it on and off so that, speed is maintained right.

(Refer Slide Time: 10:03)



Now, talking of the role of the controller, there can be various different types of control mechanisms. Let us start with this simplest kind of controller this is called ON-OFF controller ON-OFF controller means, we either turn on the device turn off the device depending on whether the measured value is less than or greater than the preset well with respect to the heater. You can appreciate it, if the temperature is going down I turn on the heater, if the temperature is going up I turn off the heater right.

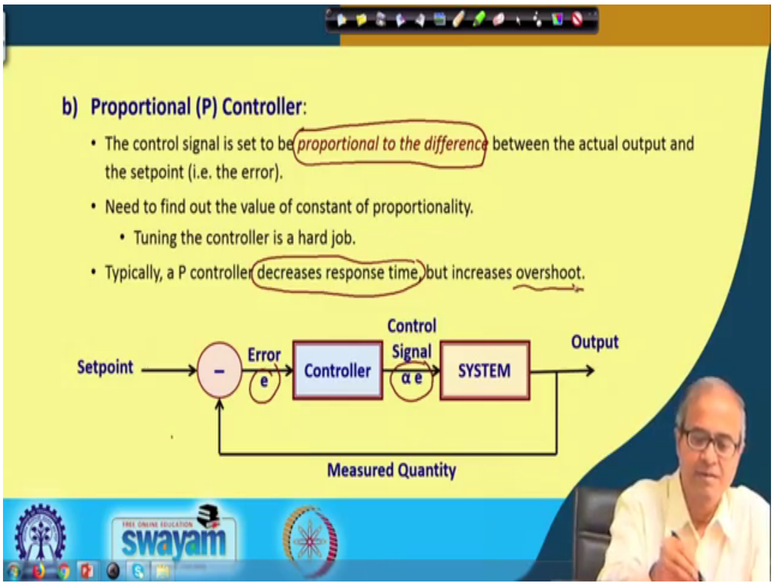
So, as I said this is the simplest type of controller, where the control signal has only 2 levels, if you say that here I have the controller. So, controller will be generating a single output, which is either 0 or 1, 0 means off 1 means on, nothing else. So the heater example, I have already taken this is very simple to implement because, you are having a single digital output generated by the controller, but it can cause oscillation in the output variable. How? You see this diagram on the right. Here, the graph that I showing it is a temperature versus time graph. Suppose in a room, where I am sitting, I am measuring the temperature and I have a set temperature let us say here, I have set it to 27 degree Celsius let us say.

So, when the temperature is less than 27, this is my controller output my controller is on. So, the heater is on. So, the temperature will slowly go up, now as soon as it crosses this set level I turn off the heater. Now you see as soon as it turn off the heater, the temperature suddenly does not start to fall because, heater is already glowing it is heating. So, temperature will still increase for some time then it will stop increasing any further.

So, there will be an overshoot like this here, here and then when you turn off the heater the temperature will again slowly start to go down as soon as it is process, you again turn on the heater again heater will take some time for the coil to become hot. So, this will again go down and again this process will repeat ok. So, you will see there will be a continuous oscillation like this up down, up down, up down this will go on, it is not that it will be very stable at a certain level, it will be going 26, 28, 26, 28 something like this will happen right.

So, this kind of a controller is quite good for this kind appliances oven see just in a heater 1 degree difference does not matter ok. But, there are some applications where you need to control the parameter very closely, there you cannot use this kind of a simple controller.

(Refer Slide Time: 13:19)



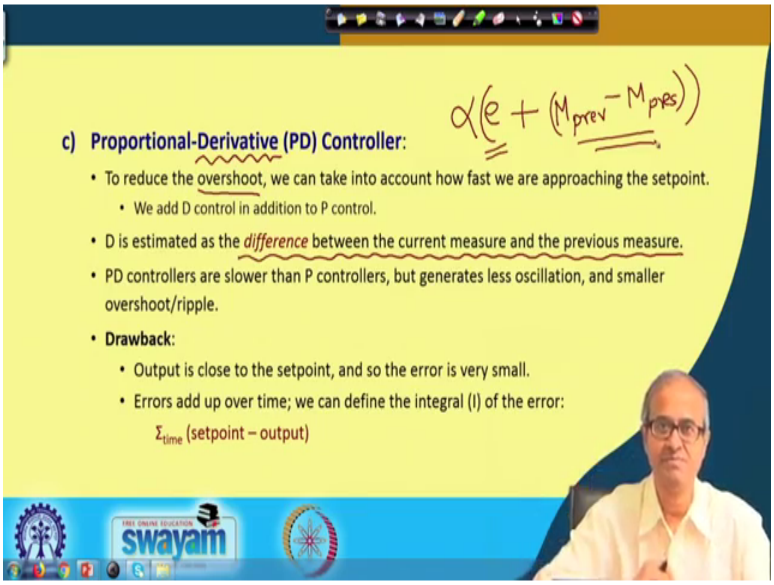
So, you can have something which is slightly more sophisticated. Here you say, I use something called a proportional controller called P controller.

So, what does this do? Here my controller is not generating a digital signal, but is generating an analog signal an analog signal is proportional to my error, proportional to the difference, if I see my room temperature and my set temperature is quite different, I sent a high voltage to the heater. If I see my difference is very small I sent a lower voltage to the heater so that the heater does not suddenly become very hot and there is an overshoot ok, but if the difference is very large I sent a large voltage. So, the heater heats up quickly. So, that it very quickly the level attends the required set value.

So, looking into that same diagram again, if your error is e your control signal will be proportional to e right. So, if e is 0 then you are not sending any corrective signal. So, whatever level was there you leave it, this is how it works right now the advantage, you are getting out of this proportional controller out of or means as compared to the on off controller. Let us say is that here the response time becomes less because, depending on the difference depending on the error the magnitude of the control signal is varied.

So, if the error is large you send a larger corrective signal. So, that the error can be reduced very quickly, but if it is not the other way then you can it will, but you see here response time is decreased, but overshoot still remains it can go up and it can again go down, this kind of overshooting will still be there. Because, see you are still sending a control signal and control signal will change the value of the parameter in one way and you cannot instantaneously shut it off, it will take some time for it up there will be some overshoot.

(Refer Slide Time: 15:53)



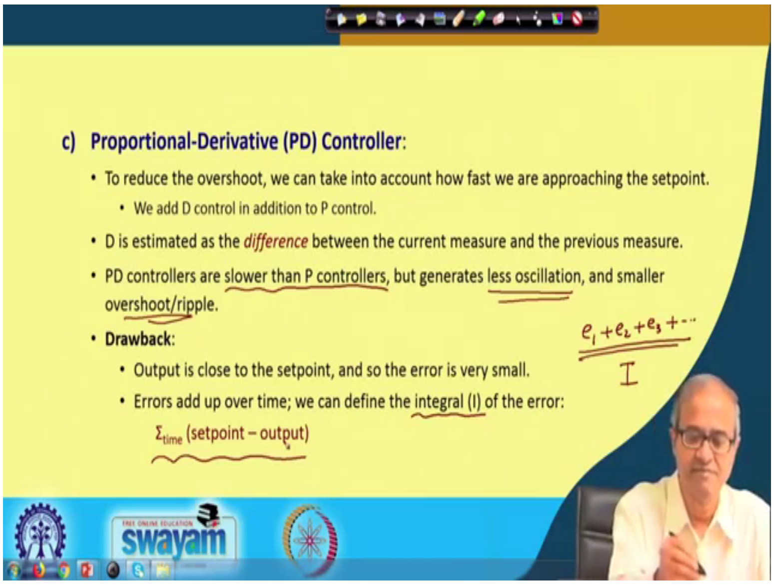
So, what you do in the next step is that well this proportional controller is there, you add another flavor to it, add a derivative concept proportional and derivative controller, because in proportional controller I mentioned that there was the problem of overshoot. So, to reduce the overshoot; so, we see mean overshoot happens when? When that means, you are see in case of proportional controller, you are trying to reach the set point very quickly by applying a large control signal, but as you reduce the response time in that overshoot can happen.

So, in the derivative part you are also taking care of the fact that how fast you are approaching the set point that also you are taking care of derivative means the difference last value of the measured variable minus the present value of the measured variable. So, derivative is nothing, but a measure of the difference, difference between the current measure and the previous measure, you continuously you are sensing the output variable, you saw that last time the temperature was let us say 21 now it is 23. So, it has increased by 2 degrees; that means, increasing very fast. So, you should decrease the rate such that overshoot will not be there.

So here, the control signal that you are generating it will be proportional to the error not only error plus it will also be the proportional to the difference; that means, you are measured value previous minus measured value present, this is your derivative. So, this will be proportional to both of these of course, the proportionality constant may be different for the 2 cases because, you may want to tune that you will be giving more importance to proportional or more importance to derivative ok.

So, PD controller works better in terms of reducing overshoot because as you are approaching the goal faster, you reduce the control signal so that overshoot will be less.

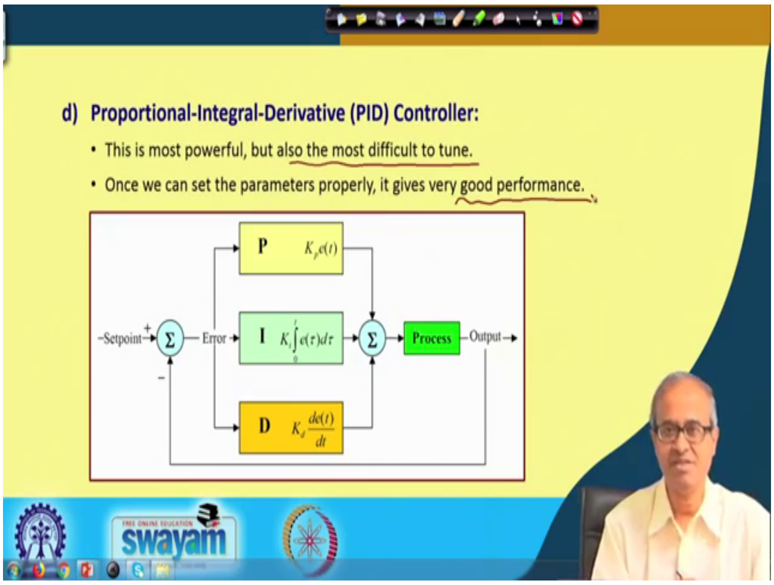
(Refer Slide Time: 18:37)



But one thing is there; these PD controllers are slower than P controller because, you are deliberately slowing the rate of convergence ok. If it is closer you make it slower that is why it is lower, but oscillation will be less and also overshoot is less ok. But you see, one drawback is that that in the PD control the output becomes close to the set point; that means, the error is small, but errors tend to accumulate over a period of time, if we allow for a small amount of error, this errors will go on adding.

So, you also need to consider a third parameter that is the integral, but integral the mathematical concept is summation over a certain period of time, if you just add up the error; that means, set point minus output; that means, error 1 plus error 2 plus error 3, if you just add up the errors this will define the value of the integral. So, you can add a third point, where summation over time this error, this will give you how much error you are accumulating over time. So, you add this third term in your control one will be proportional to the error that is your P one will be proportional to the derivative that is D and here one will be the proportional to the cumulative error that is I the 3 taken together is called PID control proportional integrated derivative.

(Refer Slide Time: 20:21)



This is what this PID controller is all about this is; obviously, most powerful, but now because we have 3 different things to tune what will be the 3 constant of proportionality is for a particular application this is most difficult to tune ok. So, you have the set point you have the measuring mechanism, you have a proportional part in the controller integral part, derivative part, you add all of these 3 up, they can have 3 different constant of proportionalities K p K i and K d and you generate a consolidated control signal for the process, you are trying to control. Now again, I told you that this is most difficult to tune, but once you have tuned it, this system will give the best possible performance among all possibilities.

So, PID controller is good in this sense and in embedded system applications because, you are talking about controlling some appliances and all of these are feedback control systems. So, you may choose to select the appropriate mode of control for some appliances like the refrigerator this on off control is good enough because, you might have heard your refrigerator making a sound in the night, it is suddenly turning on mixed sound suddenly turning off, those are on off control, but you can have PD control or PID control also in some more sophisticated application, where errors in the output cannot be tolerated beyond the certain limit ok.

(Refer Slide Time: 22:05)

So, to summarize based on the different kinds of controller ok. I am not talking about PID taken together individually only proportional, only integral, only derivative, they all serve some purpose in terms of response time this on off has a smallest response time, it is fastest you either send 0 or 1, but for the others proportional is next one integral is next derivative is lowest. In terms of overshoot for on off the overshoot is highest then proportional then integral then derivative, but for total error for integral the error is 0, you are trying to minimize the error because, some of the errors is your integral. And this on off gives very large proportional is very small error derivative, there is very small change in the error over time because, you are always trying to keep it within a certain limit.

But all these things are really not required, you need to understand better about the parameter, you are trying to control in an application and then you can select an appropriate method of control. Now in the experiments that we shall be showing for simplicity mostly, we shall be using on off kind of control because, it is easier to show you demonstrate and also easier to write the program. Because, you see whether you use ON-OFF or PID control depends entirely on your software nothing to do with the hardware the hardware remains the same.

So, with this we come to the end of this lecture, where we have discussed some basic concepts about control systems in particular the feedback control systems and some standard ways of generating the control signal to minimize errors and other desirable properties.

Thank you.