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 - 02

Lecture - 10

Hi, so in this lecture we will look into feedforward design. And also we'll see what is the difference between feedforward and feedback design? What is the advantages and disadvantages that we get between the feedforward and feedback design? Let's start with what is the idea behind the feedforward design. So here we will look into from the perspective of setpoint weighting.

We have already seen that the PID control structure looks like this, where the first term conveys the proportional term. The next term conveys the integral term and the last one conveys the derivative term. In this equation, we always consider this E of T as YSP minus Y, which is the error between the output of the plant versus the input that you are deciding and which is given by YSP. So, now we will modify this approach by instead of considering E of t as fixed what we will consider some weights given to the input signal or the set point that we are considering. And those weights can be again given differently for proportional integral and derivative terms.

And so that is why we will consider here given E p is some b ysp minus y where b is between 0 and 1. So we are weighing giving lesser weight to the set point value as compared to the output value that we are getting. And similarly, for the derivative error that we want to consider is some weighted sum of ysp and y. So, ysp is being weighted by c, which is where c is between 0 and 1. Now, when we are giving to the integral term, we would like to consider ysp minus y only.

We will not give any weight over here to YSP and why? The idea is very simple. The integral error term or the integral control term is responsible for giving zero steady state error. So if I am giving some weighted sum here, any other weight over here, then my output is going to be matched with that weighted YSP term which is not desired. So this in order to get the zero steady state error we will always consider the error given to the integral term as the true error ysp minus y all right

So now when we consider this kind of setpoint weightings given to the derivative term and the proportional term we can split up the control design in terms of feedback control and the feedforward control So, what it appears here is that when we have given these weighted errors to the derivative and proportional term, the feedback control part turns out to be same as PID form. While the F of S term turns out to be in the feedforward channel term which is coming out as YSP and it looks like something like this which is nothing but a second order term. All right. So now, what we have is this particular C of S term, what we designed as PID, which appeared automatically, F of S is turning out to be a second order system block.

What appears here is the C of S term, and this C of S is exactly like a PID term where my errors are the true errors, Ysp minus Y, and F of S is turning out to be similar to this. So with this structure, the C of S term, which was our PID control term, is giving me the same response to load disturbance in the measurement noises irrespective of what we select as b and c. Now, if b and c are not equal to 1, it means we are giving some weighted values to the proportional and derivative term. In that case, what benefit we get here is a response to the set point changes improves like anything. All right.

Let's see what we mean by the set point changes. For example, I consider a step input given here. If my b equal to one means my proportional term is exactly same. And then what happens is I will get some kind of an overshoot. And before reaching to the steady state value of or the final value of one.

Now, if I give a little lesser weight to the proportional term, which is b equal to 0.5 or b equal to 0, in fact, then initially what happens is I will be able to overcome the overshoot and I can reach the final value faster here without overshoot. Now, without overshoot is one benefit that we are getting. At the same time, when I am giving the less weightage, I have the opportunity to increase the gain of the feedback term C of S, proportional gain term over there. So that is where we get an advantage in disturbance rejection indirectly. What we have here is that if a disturbance appears at some point of time or the noise appears at some point of time, then the disturbance would be equal to one term turns out to be giving the rejection with a significant change in the output response.

But since I have the opportunity to increase the gain term of the C of S term, we can have a very nice disturbance rejection turning out to be here. Similarly, for the noise rejection capability, we can see that there is not any effect, no change you will observe with any changes into the b variables. All right. So what happens here when the set point changes very significantly? We got the smallest overshoot for b equal to zero and overshoot increases with as we move on as we increase the b.

And as we mentioned here, smaller value of B reduce the overshoot of this and allows increasing the gain of the controller to improve the disturbance rejection. All right. Let's see when we consider both the weights b and c with c equal to zero. what happens is you are giving the derivative error term as minus y alone. So, whenever there is a set point change, very much a significant set point change at the input side, then the derivative kick can be avoided because here the output is going to immediately change because of the error term has changed significantly.

But if I am considering c equal to 0, what comes out is that I can avoid the derivative kick. A significant change in the output can be avoided or the significant change due to the derivative term can be avoided if I am considering c equal to 0. b equal to 1 if integral action is implemented with positive feedback around a lag. So this is what one such implementation of the integral term that we have already seen it is that integral action can be managed with the help of a first order term here. That's when we can select b equal to 1 and so on.

So in the industrial terminology, when my b equal to zero and b equal to zero means P and D term has got both of them have got the set point, have got the error term modified. And that's why integral since integral term is not modified it is called I dash PD controller. If b equal to 1 and c equal to 0 means b term is corresponding to my proportional error term, which is again not modified, but only the derivative term is modified. In that case, the controller is called PI dash D, it's only the terminology-wise that comes up. Now, let's see what we have in place.

We have a feedforward term F of S, we have a feedback term, feedback controller C of S for the plant P of S. And of course, we have the sensor measurements being done at Y, which is giving the feedback, which could be filtered in order to avoid the noises. This is typical block diagram and one has to understand that when is what important term here. We have seen that while we are designing this particular entire control system, we will like to have good transient response to the set point changes. And that's where this two degrees of feedback controller is important.

And these two degrees are split into F of S and C of S. So this C of S term is important, it is capable of rejecting load disturbance and measurement noise, while F of S term is giving us a good transient response to the set point. But if the system with error feedback only, if we want to consider satisfying all demands with the same mechanism with only C of S, then it turns out to be a one degree of freedom system. It is basically this F of S which is not turning out to be in the feedback path is why it is called the feedforward term. These terminologies are important to understand and feedforward design is further elaborated here in order to understand the capabilities of feedforward term.

Let's see. We started with this feedforward term, feedforward design by considering that, okay, my error terms for the proportional and derivative terms are having the set point weights, which can be then later on said that when we split into F of S and C of S, which are the feedforward controller and the feedback controller, feedback controller turns out to be same as the PID controller. whereas F of S has the structure something like this. So we have seen that it separates the design problem into two parts now. If I have to come up with some robustness and good disturbance rejection ratio, those design criterias have to be satisfied with the design of PID control, which is by changing the gain term k, integral time constant ti, and the derivative time constant td.

Whereas, when I want to consider a good response to the set point, and what is that if I'm characterizing that response, then I will have to modify or tune the parameters b and c. So first, we can see that, I mean, one can look forward for a stepwise design here by first designing C of S, fixing T I and T D, and after that, look forward for designing F of S as a second step for the good response to set point. Fair enough. Further, we'll look into the systematic design of F of S, some different way of looking at it. So, F of S now you can see in this particular block diagram is split into two blocks, MY of S and MU of S. Each of them is having some dynamical characteristics.

It's not simple static gain is what we are considering here. It's a dynamical, each MY of S and MU of S is a dynamical process governed by a transfer function. All right. So if we consider this kind of block or we consider that the feedforward signals are coming from UFF and this YMFF, let's understand what is the working when the set point changes. So what happens when the set point changes because of this MU of S term, U of F is also changing, and this is responsible for changing the y immediately or the output very immediately by changing the value u, but at the same time since some things have been changed the set point is not the true set point going into going into this summer block which used to be the clear cut the feed forward signal that is getting modified here

now output Y chases YM instead of YSP, all right? So what should it chase YM as compared to YSP when this particular signal UFF is giving the immediate changes in the output Y? So we can understand that MU and MY of S are doing something in sync, and that is something we have to now design. All right. So this gives us a formal way of understanding when the set point changes, UFF changes through MU of S, which gives the desired output Y without looking into the feedback or MY of S.

So immediate part that we said is getting changed because of this particular feedforward control input UFF. Now, as soon as this happens, now we have the desired output modified as YM, which is passing through, which is getting modified with the help of MY. So, under ideal conditions now, Y is equal to YM and E is equal to 0. As a result, now our UFP or the feedback signal is also changing. All right.

So when we looked into the roles of this MU and MY of S, MY of S is responsible for giving desired set point response, whereas MU of S generates the signal UFF, which gives the desired output when the YSP changes, the change in the YSP happens. All right. So now looking into the design of it, what turns out is a transfer function between G, Y, YSP means the transfer function between the output Y and input YSP, the entire transfer function can be returned in this way. So here we have a term. Once we modify this, it turns out that this particular transfer function between output y and input y sp is m y plus p m u minus m y by 1 plus p c.

Now, this particular transfer function will behave like M y which is our designed transfer function if this particular PM either this numerator is almost 0 or the denominator is very large. The numerator of this term is going to be almost zero when p mu minus my is small or the denominator of this term is going to be very large when pc or the loop gain pc is very large as simple as that all right So under the ideal conditions if my feedforward is MU equals P inverse MY, then this transfer function going to be same as MY. So what is the advantage we are getting? So, okay, there's only one point here that this P is representing the transfer function of this G of S, all right?

So please make that point clear. Now, P inverse is the process P is the transfer function of the process and it should be invertible in order to design such kind of a feedforward. But this particular feedforward design is telling us that, okay, I can make sure that my entire control system looks like this MY and this MY is nothing but user-defined transfer function. So irrespective of this, whatever is my transfer function of this P, which is getting canceled out to certain extent over here, my behavior between Y and Ysp is governed by the transfer function MY.

But this is always possible or not. It depends upon if I know the process completely or the model of the processes is completely known in terms of the transfer function P. That's why and at the same time, this transfer function P should be invertible in order to design MU and MY of your own choices. So this is where the catch here is and the design criteria depends upon what is the accuracy level at which you know the model of the system. All right.

So now we have, if we compare feedback versus feedforward, as we mentioned in the beginning, we have, these are two different approaches. Especially the design-wise, this feedforward works on matching two transfer functions because we are dependent on knowing the model of the process plant very, very accurately given by the transfer function P. Whereas, if you look at feedback, feedback controller or the PID controller that sits into the PID loop is dependent on what is the error. So its design is based on making the error small by dividing it by a large number, which is the 1 plus PC form. So as a consequence, feedback is more sensitive than feedback because feedback is dependent on the model parameters completely.

sorry, feed forward is dependent on the process parameters completely, whereas feedback can have some variations possible here. With the feedback, there is a risk of instability because we are putting it into the negative feedback. There is a chance that our stability regions are compromised, but there is no such risk with the feedforward part because it's appearing in the forward channel itself. Therefore, we can consider that feedback and feedforward are complementary. When we combine both of them, there's a high chance that even the tight control objectives could be achieved.

And we already know that what feedback is giving the control objectives getting satisfied and feedforward. There are set point changes kind of control objectives getting satisfied with the help of feedforward. All right. In certain cases, the inverse of the process may be a little difficult to even if the system is invertible, then what happens is the MU or MY should not result in the unstable system transfer functions. One such example of the system P of S is FOTD system, which is 1 by 1 plus ST, E power of minus SL.

As soon as I invert it, what I will get is E power of plus SL. which is a non-causal system and designing such a transfer function MU is going to be not possible. So then how do I design this? The one way is to design and approximate it by considering that P of S is given by one by one plus ST, e power one minus, we can consider the first order SL by two. Representation of a power of minus sL, so this delay term can be represented this way but now even then when I get this P inverse what happens is this numerator gives us a pole on the right-hand side

So in such cases what we can do is we can split this this particular transfer function with a transfer function which can result in giving us the poles on the left hand side when it is inverted and the other part is this 1 minus SL part which is resulting in the right hand side poles when I am inverting it. So this part I am not going to touch but I will just do this. So I have P1 inverse which can be then said as 1 plus ST times 1 plus SL by 2. Whereas I will keep P2 same as 1 minus SL by 2. So I'm going to design the feedforward corresponding to this particular P1 system which is given by some partition of the actual process transfer function.

Something similar with RHP zeros, I can split this in terms of P1, which is given by S minus 1 and P2, which is given by S plus 2. So I will just design the feedforward corresponding to this which will result me P2 inverse equals S plus 2 because if I will consider this as well I am going to get the resultant inverse transfer function as having the right hand side poles which is an unstable system. Alright so then I can consider these two ways. Either I will do the partition part of it or I can consider having the time delays same as P of S so that it is kind of managing the system well by even considering when the delay is same as e power of, it gets canceled out. So if I consider P inverse in this case as one plus ST e power of SL, then I should choose MYS having some transfer function T e power of minus SL.

So this E power of minus SL and E power of SL cancels out and then my MU will be just one plus ST times T. This T is T of S, some T of S. So this way, some such ways we can design in order to compensate for the terms which are responsible for the unstability or non-causal terms. But the important point here to understand here is that when we consider them matching terms, one has to have an exact matching happening here. So one has to know the delay term very accurately, then only you will be able to do the cancellations done and get rid of the unstable part or the non-causal part. All right, now as a summary for the set point weighting, what we have done is we have simpler PID controllers.

We can avoid it using by complete system with two degrees of freedom. We have instead of one simple PID controller, we have one PID controller and the feed forward term here. Then my control objectives are likely to get satisfied and the desired response can be adjusted by the set point weights. Now, in order to determine the set point weights, we consider the transfer function as this part, which is the feedforward terms times the T of S, which is the feedback control system part. We can choose the set point weights b and c as the largest gain of this particular transfer function or whatever is the operating range of frequencies, you consider the maximum of this and choose one that is close to 1.

This way we will get a set point response without overshoot for most of the systems. And I request you to verify this particular claim. So once this b and c are set, one can look forward for varying the proportional gains and integral and derivative gains to certain extent. Earlier we said that design C of S first and then design this, which is the typical way of looking at it. So once we have the design already, we already have satisfied the disturbance rejection criteria by designing proportional gain, integral gain and derivative gain.

One can find this particular transfer function and this transfer function, wherever it is finding it as maximum, choose that particular b and c values. All right. Now, there's another very good point about the feedforward is for disturbance rejection also it has been used at times. But in that case, this particular disturbance that is appearing in the middle of the, so in this particular block, what we consider is that P of S, which is the process block, the disturbance is appearing somewhere in between. So that's the reason it is partitioned as P1 and P2.

So more or less, we have the P of S given by P1 of S, P2 of S. But for the sake of disturbance that is appearing at the summation plot here, in between the process is given by this summer block. Now, in order to reject this disturbance with feedforward, we need to design this GFF. So what we have is whatever disturbance term turning out here, it gets added before even it is applied to the process. So this UFF will be responsible for rejecting or nullifying the effect of the disturbance is what this feedforward term is turning out to be helping us out.

If you notice here, this C of S is already doing some kind of a disturbance rejection as a part of the PID control benefits that we have. But here, if there is a very large disturbance and we further want to attenuate it, then we are resorting to this particular feedforward way. It is not from the YSP side. There was one input which is YSP, Now we said, okay, now since there is a disturbance, which is another input, which is undesired input, can we nullify the effect of it?

Same way is what we are attempting here. So what we have is this now, we would like to get this transfer function between the output Y and the input disturbance T, which is given by Y of S by D of S. All right. So this GYD turns out to be equal to P2 times 1 minus P1 GFF or times this S. This S is nothing but 1 by 1 plus PC, which is actually the term of the sensitivity, which is also called the sensitivity function.

Nevertheless, what we are expecting here that the transfer function between Y and D should be almost equal to 0 so as to reject the disturbance. Which means what we want is the transfer function GFF should be equal to P1 inverse, which is nothing but the transfer function between what is P1 now, if we go back and see what is P1, P1 is nothing but this particular intermediate signal minus and the input to this is the control input U. All right, so what we have a question in mind, when is the feedforward most effective? Whether P1 should be equal to P and P2 equal to 1? If that is the case, then my disturbance is P1 equal to, disturbance is actually appearing at the output Y.

Because in that case, what I have is P1 equals P and P2 equals 1. The second case we had is P1 equals 1 and P2 equals P. so in this case my entire process transfer function is appearing here, which means my disturbance is appearing at the output whereas in this case my disturbance is appearing at the input of the transfer function, so when is this most effective? It is like if I have this particular part, which is the output of it, I have to design GFF equals P1 inverse. In this case, I have to design GFF is equal to identity.

It means the disturbance is completely being, so GFF is equal to 1 means what? My disturbance is applied as the negative of this. Now, if that is the case, then I should have the estimate of the disturbance completely available. Then this is going to be workable solution. Since the disturbance estimation is not available, the process transfer function with the help of when this particular disturbance is affecting at the output with a proper design of GFF, I can reject it very clearly with the help of the feedforward way of designing the disturbance rejection.

So this is what the major advantages that we have been looking at from the feedforward design. We have seen feedforward design for the set point changes. Towards the end, we saw the feedforward design for the disturbance rejection. One can look forward for using these methods effectively depending upon the control objective. Thank you.