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 - 04

Lecture -20

Hi, in this video we will look into data driven way of PID control. As we are moving on to learn more about the data driven PID control, since it is data driven, it is going to be compute heavy. At the same time, this will also be giving benefits of the PID control, which has simplicity. The control methodology that we are selecting here is PID always because of the simple control structure. Because of it has some meaning of the control parameters.

At the same time, operators can be trained easily for the tuning, and that is the main reason we have been selecting PID control throughout our practical control methods in this lecture series. For the data-driven PID case, we would like to consider PID controller design for highly nonlinear systems. Now, one is we are using it for highly nonlinear system. What we studied so far into its origin of the PID control was mainly on the linear systems, LTI systems, and so on and so forth. At the same time, we tried simplifying it with the first order system, second order system, considering the dominant pole analysis and so on.

And at the same time, since its PID control has its own benefits, people have applied it in highly nonlinear cases as well. Now when it comes to the highly nonlinear case, one is one can derive its model. It's a nonlinear model, fair enough. Even though when we consider it's a highly nonlinear model, there's a chance that we have missed out some particular variable and whatnot. So in that case, the modeling or that particular variable relationship is not known.

In that case, getting the model for that particular nonlinear system is complex. One is complex and perhaps not available. So in that case, we resort to the learning methods. But at the same time, what is available for such systems is if I am applying this input, this output is available. Corresponding to this input, this particular output is available.

So this input output data for this input and this output, this input output association is available to me or correspondence is available to me. But when I am learning through this particular data, sequence of data that is like for particular series of some sequence of inputs, this is what the sequence of outputs I am getting. At the same time, it is not correct input-output relationship. We are talking about dynamical systems. So, there comes the way out for designing the controller with the help of the learning methods.

At the same time, the amount of data requirement is going to be very heavy and the compute requirement for working with this data is going to be heavy. So, the concern is always going to be the learning cost, how quickly I am able to do it. And how quickly and how accurately I'm able to do it. The structure of the PID is going to remain as PID control because we can implement this PID control very easily. Now comes the concept.

Now let's try looking into concept on how I am going to design the PID tuning rule for nonlinear system cases. So, what we have we already have here is the historical data as I mentioned, input sequences available and corresponding output sequences available. At the same time input 1 is correspond is giving you output 1, input 2 is giving you output 2 and so and so forth. Now, this is just not input which is the control input given and the output. Because the system is dynamical, the sequence of input is affecting the output and that is where we will have to redefine what is input to this data, what is this input data we are talking about and what is the output data that we are talking about.

But as a concept, the memory or the database will have its input and output. Mind it, we are not talking about system input and system output here. It is the database input and the database output. And that needs to be, that's basically creating the database for us. Now, we cannot keep all the cases possible.

So that's the reason there's going to be some data or similar data. So we'll have to define what is the similarity that we're talking about. And based on that, we'll be giving the input to the local controller, which is our PID, means we'll be giving the proportional gain, integral gain and the derivative gain. So somewhere we are talking about we'll be giving query to the database, this query will say, okay, my query is near to these two data entries. And this particular data entries are like a bunch of data, which is similar data.

And this is from the similar data we are assessing what should be the PID tuning rule, PID tunes, and that is given as an output. And for this particular nonlinear system, the query and the output of the database which we want to use it for the PID control, the gains, that is going to be highly nonlinear, correct? So now what we have to look into is understanding these concepts, this is one way of doing it from many different ways, some of the ways that we'll cover it in the next lectures. So this one way, in this one way, what we consider here is that this data that is the information phi t minus one, which is available for previous time instant. And this particular query, you can see that this has timestamp, right?

So this query is timestamped because I'm dealing with dynamical systems. Now, dynamical systems, because these are governed by differential equations or the discrete time equations and so on. Of course, we'll have to work with discrete time equations here because this is the digital world that we are talking about. So the database is queried by this information vector at t minus one. As we said, we will have the similar neighbors identified, and then that gives the local controller parameters corresponding to the query at t minus one.

All right, so let's look into how we are going to form the information vector and the output vectors. So, let us consider our normal way of looking at the non-linear system input u of t and output y of t. So, this y of t is say some function of this information vector. Of course, this information vector we will have to figure out how we are defining it. If I consider this y of t as a discrete time equation, because this is what is the relationship that the system model gives us.

Of course, that model is unknown to me. So then information, then this right hand side turns out to be some linear combination of y of t minus 1, t minus n y means n y previous from I'm calculating the time output value of the system at time t then I am considering ny such samples previous samples of y in this particular sequence y of t minus 1 to y of t minus ny. Similarly it will have the nu previous input samples. So, this sequence u of t minus 1 to u of t minus nu is being considered. So, this information vector at t minus 1 considers ny previous samples of output and nu previous input samples, all right.

This we are considering it as a variable in this particular case, but for a particular system this ny and nu is going to be constant. Okay. Now, when I design a controller using PID, I consider the error input E of t equals Y of t minus R of t. Now, in this particular lecture, we are considering the controller is PID. So, U of t is of this form, which is Kp proportional gain times E of t, integral gain with an accumulator, derivative gain with delta u of t

Now this delta is a difference operator which is given by in the z domain is given by 1 minus z inverse frequency domain Okay so when we apply this difference operation on various on both these sides then I can consider this delta 1 minus z inverse if i consider this as the u of t minus u of t minus 1 then I can write in terms of k i e of t because my accumulator is going to be given by so this delta u of t is actually u of t minus u of t minus 1 and this term is coming by when i consider u of t it is k i summation e of t minus and when i consider u of t minus 1 it gives k i summation e from t minus 1. So, this is the previous sample we consider. So, basically t equal to 0, e of k I should have considered. So, k from 0 to t this will be if I am considering e of k here then these samples are ki summation e of k, k equal to 0 to t minus 1, that's the reason when I take subtraction I am only going to get left with k i e power of t e of t here, k p delta y t because now my term here, this u of t minus u of t minus 1 is nothing but e of t minus e of t minus 1.

So, my considering that I have already reached this steady state, e of t minus e of t minus 1 is r of y of t minus r of t minus y of t minus 1 minus r of t minus 1. Now, considering that I have already reached a steady state r of t and r of t minus 1 is equal, that is why I am left with y of t minus y of t minus 1, which is delta y of t, which is this term. I can do something similar for the derivative part here. Now, this, we cannot fix Kp, Ki, and Kd here because the system is nonlinear. If it is linear, then we would be able to do it, but since the system behavior is nonlinear, I will not be able to consider fixing it.

My operating point could be something else, somewhere, even if I'm changing the operating point by some small point. Even if I linearize it, linearized solution will is going to give me different transfer functions and so on, that's the reason I will have to consider time varying pid gains and this particular vector big kt bold kt is now kp of t ki of t and kd of t And then we are going to consider u of t as some function of this phi prime. This is another information vector output vector that we are considering here rather. So this phi prime of t is the k vector the input at time t, output at time t to t minus 2 because I am getting this delta u of t with the help of two previous output variants and the previous u of t minus 1 that is being used in e of t minus 1.

So, basically I will be able to use this particular equation to calculate u of t, which is equal to u of t minus 1 plus all this k i e of t and so on, k p term and then k d term, all right. So, in order to calculate this u of t, what is needed is the proportional gain K of T, U of T minus 1 and three samples of output variance and output variable in R of T to calculate this E of T. All right. OK. So, what I can write now is y of t plus 1 which is the system response at t plus 1 is some function of phi tilde. Now, this phi tilde is y of t to ny previous samples plus the current t sample.

The proportional integral and derivative gains k of t. The current command r of t and previous nu samples of input. So, this particular output now typically for a non-linear function and PID control way we will consider this ny to be greater than or equal to 3 and nu to be greater than or equal to 2. I can always consider increasing this particular information vector dimension, but 3 or close to 3 should be good enough, but more than it should be more than 3 at least. So, now I will start by saying that when I have when the output response y of t is to be calculated it is dependent on the this phi tilde this information vector phi tilde t which is a sum function of it.

I do not know that particular h mapping here h function or h map here. Now, phi tilde as we said is going to be some previous samples of y r of t and u of t to ut minus 1 because I will be able to calculate this particular the gain vectors as a function of phi bar. Now, what is this phi bar? This is phi bar is again can be calculated is nothing but the information vector comprising of NY samples of previous samples of Y, the previous K of T minus 1, the previous value of the proportional gain R of T and the previous U samples. Since future sample is not available to me, when I have to calculate these entries to be done into the database, this particular sample value is not known.

And how will I calculate the gain k of t that requires y of t plus 1 now? OK, so what we did in this case, let's let's look into it once more that the system output at t plus one is a function of some information vector, which is a function of k of t as well. So, k of t in order to calculate k of t which is my proportional gain. So, basically you have this particular phi tilde right. Now, this phi tilde depends upon your output response y of t plus 1 depends upon certain values of this.

But these values we are trying to control with the help of PID control. So, that is the reason we wrote this particular PID gain vector in terms of this new vector which is comprising of the future vector, future value of the output and the previous values of this. All right. But that is what the system response is going to be. But this particular Yt plus 1 is not available to me.

When I am looking into giving a particular PID gain values, I do not know what the output is going to be. But since I am looking forward that the PID gain that I am designing is going to give the steady state output. So, I can consider that the output Y of T plus 1 is equal to R of T plus 1. which is to certain extent since it is a reference signal is available and that I can feed it into the database entry now. All right.

So now my job here is to consider getting generate these initial database. Then once the initial database is ready, right, what we can do is we can query the database. Now, when I'm querying the database, it is not necessary that this particular information, that particular query and that information vector matches. So since I have a database, I have multiple queries into multiple information vectors being kept here. Now, when I'm querying it, it is not necessary that it will match with this particular information vector.

This particular query is somewhere sitting close to this particular information vector, but not with others. So then I will choose the select certain neighbors of this particular query that I had. Now, this query is close to a particular set of neighbors. And first is how many neighbors to consider? How do I calculate this distance?

This needs to be defined. And for various different ways, you can define the distance. Various different ways you can select the neighbors based on which your method is going to be different and what way you want to tune the PID will evolve. Once this particular, we have selected the neighbors. Now, we need to, with the help of these individual neighbor information vectors are giving me a set of PIDs, right?

PID values. So, this will be say KP1, KI1, KD1. Say this particular information says KP2, KP, KI2 and KD2. So this vector and this vector and that too, this particular one is just not the PIDs, right? It could be the output vector could be different.

So one needs to match the output vector and then say that, okay, I have so many values which are near to me. Then how do I calculate my value? So my value as in where it is being queried. And these are the neighbor outputs, the output values of the neighbors. so how do I calculate what is the best pid gains for me for the query that I made.

So now I have this new query and the corresponding so in step so step three is computing this particular part based on the selected neighbors Once the PID values of this particular query is available, I am going to add this particular query because now I have the query as an input vector is available and the output vector is also available because now I have calculated based on the neighbor. Next time, there's a possibility that I'm getting the same query. So then instead of doing all this step one, step two, step three, I can directly get this particular information vector as the information vector entry into the database and the output vector can be figured out. OK, so then I will have to keep populating the database entries with this help because I started with an initial database of certain, say, 20, 30 values.

But I need to keep adding the new values to the database. So this step four does this for me. There's a new query here. New values have been created. I will add it to the database so that the next time I don't have to do this first three steps.

But at the same time, if this query is very close to certain information vectors, then I should not consider adding to the database. So I need to keep removing the redundant data here. So same data I should not keep adding. Very close by values, close by information vector values should not be entered. So, there should be a way to remove these redundant data so that my database is not growing exponentially with each sample value and to certain extent.

We should be having distinct information vector entries into the database. Now, how do I define that distinct? Who is distinct from the query? Who is from the existing information vector? There comes the method to find out.

And that defines the step five. So in the next video, we will look into one way of defining, describing these five steps and complete this loop, this understanding of the PID parameter tuning using data driven way. Thank you.