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 - 21

Hi. So, we are talking about data-driven PID control. And we have set out certain steps to develop a data-driven PID controller. These steps included generating the initial database. Then we would be looking into querying this database.

And even though that particular corresponding data entry is not there, we should be able to figure out what should be the nearest values that we should be considering it. So for that, we need to design a method to calculate neighbors and then calculate its PID parameter with the help of these neighbors as a step three. Step four will require PID parameter adjustments and we should be able to add anything new that is adding value to the database here. At the same time, step five should remove the redundant data if that particular query and the information vector that is created is almost equal to already available data. One should be able to reject such entries and should not populate and increase the size of the unnecessary increase the size of the database.

So one thing at a time, let's take step one, which is initial, creating the initial database. One can always look forward for already existing methods like Ziegler-Nichols, CHR method, operators know-hows, or any other PID tuning method. But for this, to know, to apply these methods, which are already known, requires a model of the system. This model, of course, to a certain extent, we should have some idea about what the system is about. Just as a black box, I want to program this controller design and to take it up.

It's going to take a longer time. So to understand about what this particular system is and a bit about what this particular input-output relationship would be, that particular starting point would be just enough to start with some PID tuning. Even though this particular starting model may be very highly inaccurate, then also this step one to step five will make sure that these database entries are getting corrected as we move on, as we have more and more data available. So, then given this particular starting point, we can create initial number of information vectors and the database can be created with it. Now, this initial vector will constitute of jth element of the information vector will consist of this phi bar j and its PID control values, PID gain values.

So, this jth information vector what we had decided earlier was this phi bar j is having the values from previous ny y vectors. This gain vector is already coming out in the form of a separate value. And then this particular reference value at that instance and the control inputs for the previous NU time instance. Now you can see that this phi bar of T has K of T as well as we want the output also to be of the form of K of T. So, this is where we would be able to update for a given values of the previous history or the data points, which constitute the previous Y values, previous the history of Y or the output signal, history of the command signal, the current reference and the previous K of J value.

This is to certain extent would be able to update the new K of J value. So this is K of J which is turning out to be, if I have given this input which constitutes of Y history, then R of T, then U history, history means I have those NU samples, I have those NY samples and this is my current reference and then this is K of J. So, this was calculated previous K of J and this comma K of J, all right. So, once I have given this value, then what is the output value that is coming up? That is something extracted from the previous entry into the database.

Now, given this particular K of j, we would be doing a little bit of tuning adjustment in step 2 or 3 and then the new K of j value is added over here. All right. So this way, this N of zero means the initial number of samples will be will be recorded in database. Now, when I'm giving the starting with this initial value or whatever the database size, which will keep growing with time, means the database size is also growing number in terms of at least it is more than N of zero all the time is what we will ensure. So, this size, because we'll be keep adding the values, we'll be updating the values, information vector in the process from step two to step four.

This particular phi bar i, if I have queried, is an entry in the database, then the corresponding PID gain ki is available. That is that can be fetched out, can be a little bit adjusted and then done the tuning. And that is something we will see. And then but so that for that the gain values of PID gain values are available. But if query is not stored, then what we will do is we will figure out how to get the nearest neighbor or how to get the neighbors of this particular query element.

Now, in order to get the neighbor, I need to define the distance between the two information vectors. Now, this distance I can define with the help of any method. Here, in this case, we have used L1 norm. So the lth element of phi j and phi l. So here what we have is phi vector, phi bar vector, phi bar j, is nothing but my y of t minus 1 to y of t minus ny plus 1, then r of t, then k of t k of t or in fact right so k of t this is at time t that we are talking about then this is u of t minus 1, u of t minus 2, u of t minus n, u plus 1 all right

You can see that this J and T are two different identifiers. This J is the entry into the database. So this is what is your Jth entry. So you will have one, two, two N of T entries. Now we are talking about this Jth entry into the database.

Now, this database, this particular entry consists of this much number of elements. Now, we are talking about lth element, maybe this is my lth element, okay? So, the distance between the two information vectors, which is phi i and phi j, phi bar i and phi bar j, means two elements into this can be described by this particular L1 norm, which is summation from L equals one to n y plus n mu. So because we have these many entries, these many numbers available, right? So this will be phi bar i minus phi bar j.

So for the element and what is the maximum of phi L, phi bar L means what is from all the entries, what is my maximum and what is my minimum for phi L. That is used for normalizing between the elements. All right. So this way, I'm calculating the distance between two information vectors. So what I have is a query phi bar i, and what I have is a phi bar j, which is there in the database.

So I am not looking into the ith entry into the database. I'm considering phi bar i now as a query, which is coming from outside, and this particular query and this jth element means from each and every element of the database is searched and figured out what is the distance between the two between the query and the and each element into the database. given this distance, when we calculate for all such n of i entries now if we will keep doing it for all the entries which is number of entries are n of t and we are not talking about just 10 entries or 20 entries since this is data driven we are talking in terms of megabytes gigabytes we won't be able to search to this much. All right.

So, it means I have to restrict the number of entries for search, which will be the possible neighbors of the database to N of I. Now, for time query, phi bar I is to be considered. So, we will look into getting the k nearest neighbors with smallest distance are selected for step 3. So, we will just restrict out of n of t, we will restrict n of i, right? So, t and i are equivalent now in this case.

So, i-th time, there is a jugaad that we will have to consider in this case. If I keep doing for n of i for this number of bytes entries, gigabyte entries, right? I mean, of course, gigabyte is the data size, but the number of entries are still in terms of 10,000, 5,000 to 10,000 numbers. If I keep searching into that, then my time is in order to calculate the distance itself is going to be too much. So we'll have to figure out some way here to get the almost possible neighbors of this particular query that we have made.

And typically this query is more or less, the new query is going to be close to the previous query. And that's the reason the neighbor search happens almost in the nearer area of the previous entry itself. This is how we can reduce the number of distance computations in this case. Now, once the distance computation of probable neighbors are available, then the idea here is to get the k nearest neighbors. And those will be the k neighbors with the smallest distance with respect to the query here.

So, in this case, we will go to step 3. Now, what we will have is K nearest neighbors of the query phi bar of I. Now, these neighbors are giving you, K neighbors are giving you K, K of I entries. K of J, or in this case, I've considered T, T variable in this case. All right, so K of T.

So with the help of the nearest neighbor, and for example, my query was here, and my neighbors are somewhere, say five neighbors are this much distance away. So I need to figure out a way of giving a weighted sum to this particular query. Now this weight, summation of this particular weight should be equal to 1. This is your weighted sum. So this one is giving you, say this particular entry is giving you K of 1.

So K of 1, remember, is nothing but KP of 1, KI of 1, KD of 1. This is particularly for our discussions over here. This is not equal to, again, terms are being changed, interchanged over here. Let's start with the database. Now, your query at ith instance is phi bar of i, which is looking into the jth entry into the database.

Say jth and some close, some certain neighbors have been detected here. Now, those neighbors are k equal to small k equal to 1, k equal to 2 to k equal to till k equal to k. All right. Now, this 1, 2, 3 to k is what is being referred here as the variable t. So, do not get confused between the indexes that we have come up with here. Now, these neighbors are giving you PID values which were stored into the database and the i-th for the i-th query from the database, we are getting this old value of this as a weighted sum of PID gains.

These weights, summation of weights should be equal to 1 and individual weights should are calculated based on the distance with respect to the information vector phi bar of t. So, this is again we consider the lth entry here. lth element of phi bar i and the distance of the distance metric over here is again L1 norm this distance when we consider for the lth entry is then the particular distance that should have the maximum weight since distance is small We consider one minus small value, which gives you a higher value as a weight here so that the weight of the nearest or the smallest distance gets the highest weight in while calculating the PID gain value for the phi bar i. All right, so this way we got some PID gains from the calculations for the entry which was not existing.

Okay, this was the initial guess. Now what we will do is we will consider adjusting this particular PID parameter and that's what gives you the output of the database finally. So now we have this particular one was calculated. It needs to be updated and stored in the database. So now what should be the way to adjust this PID parameter so that the control error is decreased.

One can consider designing the control objective based on certain features like rise time, steady state error, or in this case, we'll consider damping properties and so on. Okay, so here we will consider rise time sigma to be considered as the feature, the rise time to be minimized as the design objective here. So we'll consider based on the damping properties defined for this particular damping coefficients, which is given by 0.25, again, one minus delta and 0.51 delta. This comes for the binomial model step response For the binomial model, delta is equal to 0 and delta equal to 0 is for the Butterworth model step response.

So, these are since we are dealing with the discrete time systems and these terminologies are coming from the discrete time systems. One can understand this very superficially that my damping coefficients do turn out to be giving some kind of transfer function which we write in the form of 1 plus sigma s and mu times sigma s squared. You see that this particular mu is what we were talking about. And of course, this is again continuous time. So we convert this particular continuous time transfer function to a discrete time transfer function.

And so now my Z transform of the transfer function turns out to be 1 plus T1 Z inverse T2 Z to the power minus 2 means the previous sample and the previous to previous sample that we consider. Now, this coefficients T1 in terms of mu and rho, which rho, which depends upon T1, sampling time Ts and sigma is again coming out from the damping factor zeta and the natural frequency omega n. So more or less things are connected. It's more or less one can consider that if I have to consider the rise time calculations and rise time can be figured out with the help of the coefficients of the transfer function in a in a variable form. And our variable form now comes out to be in the form of mu and rho.

All right. So now the reference output model for for the nice rise time turns out to be y r of i is equal to z inverse t power of t of 1 which is the initial value of the transfer function by this transfer function r i which is nothing but this is how i calculate the output when i have the given input given So this is a simple way of saying that, okay, I considered transfer function as a second order transfer function. I discretized it in the form of getting what should be my rise time parameter coming out. How do I represent my transfer function parameters in the form of the rise time?

Now, my rise time in the previous case was represented in terms of sigma. Now, this sigma now you can see that turns out to be in the form of rho and this rho comes out in the form of coefficients of the transfer function. So, then I can select what rise time I want to consider given the sampling time I can calculate rho and given the rho and mu of the values coming from what is whether it is a Butterworth or the binomial mode I can consider calculating T1 and T2, and then I can calculate what will be my output response calculation wise given this particular R of I. All right. So then I can consider this as a simple exercise of since I need to calculate, I want to minimize the error between the between the reference model and the actual output.

I will consider E of I as YR minus YI, which is my output reference model. And I can consider the error criterion since I want to reduce this particular error, which was dependent on the desired rise time. This is reference model is calculated based on the desired rise time. Now, this desired rise time output looks like should be Yr of I, correct? And the output Y of I is coming out from the system.

This error I want to reduce in the i plus 1th step. So, this becomes my objective criteria given by G of I plus 1 equals 1 by 2 E of I plus 1 square. Now, I want to update the controller gain vector. This controller gain vector can be now updated with the help of the old value. And this use the steepest descent method, which will be eta times dou j by dou k.

How do I calculate dou j by dou k, which is the partial derivative of this objective function with respect to the ki? Now, since k itself is a vector, eta is also a vector in terms of proportional learning rate that you want to consider. For the proportional gain, integral gain, and the derivative gain, your learning rate eta could be different. So this is your gradient with respect to K and how this particular gradient is, whether it is negative or positive is more important for us to update this K value. So K old value, which is coming from the K nearest values, weighted sum of nearest K nearest neighbors is getting updated by the gradient value times this eta.

This eta is a constant, which is learning rate that you can set that you need to tune it. And what I'm saying is that this tuning can be for proportional gain, integral gain and the derivative gain can be different here. All right. Now, the interesting part is about calculating the gradient. Now, this gradient of this objective function that we have selected, which is E of I plus 1 by 2 E of I plus 1 whole square.

This, for example, I'm calculating the partial derivative with respect to KP. The gradient with respect to Kp gives you the element of the Jacobian with respect to the proportional gain. So, this is going to be the objective function. This is since it is derivative and there are the variables are dependent on each other. We will consider first taking the derivative with respect to E

So this is dou j by dou e because j is a function of e, we took dou j by dou e. Then comes dou e by dou y. So if we keep doing this, what we have is dou j of i plus 1 is a function of e. e itself is a function of y, so you take the derivative with respect to y i plus 1. Then y is a function of u, so you take with respect to u.

Then u is a function of p. So, you take the partial derivative with respect to kp. So, that is how this particular entire partial derivative sequence of partial derivatives that you come up with gives you the derivative with respect to kp. Now, individual partial derivatives we need to figure out. So, this dou j by dou e is very straightforward because it is nothing but dou j by dou e at i plus 1. Remember we are calculating for the future values okay, so this dou j of dou e plus 1 is nothing but 1 by 2 2 of e of i plus 1 this is what we considered as the square, and 1 by 2 purposely kept so that 2 and 1 by 2 cancels out so this gives you e of i plus 1

Now comes the fact that I will have to consider calculating the error between the reference model and the current yi with respect to the future value. Now this is what is the trick being considered here and I will give you the reference later on. The trick says that since this is the error between the reference model output and the actual output, we can assume that it is based on the current output and the difference between the current output and the previous output. Since the rise time feature is already being captured into your correction, in the previous correction in this particular correction step, it is important that we have to come up with this future, this derivative part in terms of some output value or some known values.

So if I consider this yi minus yi minus one, which is the difference between the current output and the previous output, which is already available in the form of the information vector, we can assume that this dou E of i plus one is equal with the partial derivative of E I plus 1 with respect to the output at I plus 1 is this particular difference, which we want to minimize anyway. Then comes your the future output value y with respect to the current command value which remains as something like this but this partial derivative of u with respect to kp is just one, that we know that this is output value and the command input is nothing but kp times e of t So that is why with respect to Kp, at least it is equal to 1. And at the same time, that is one can consider as the current E of i. And that is also turning out to be equal to yi minus yi minus 1.

All right. That also is getting captured in terms of yi minus yi. So you can see that E of i is being defined with respect to this. At the same time, in control literature, we have been doing R of I minus Y of I. So to certain extent, which is reference input minus the output value, to certain extent, we are more or less approximating things or approximating it to understand that, okay, where am I getting the ease of applying these theories here?

So what happens is that I am more interested into getting the correct sign for this. So that when I am descending towards the correct value of the k new, which is the most appropriate value for a given rise time that we have fixed up, I should not go away from the values. So I should go towards it and that's the reason the sign of this dou j by dou k is much important than the value. Whether I am going with a very large jump towards the value or is just I am taking a very small step that is coming out from dou j by dou k. And that is the reason we will be able to do some way of understanding that these values are not giving the incorrect sign for dou j by dou k, that is the most important part.

Rest, what is the exact value of dou J by dou K? We know that it is difficult to get because to a certain extent we are breaching the causality nature of the system. We are looking into calculating the future of this when I am calculating the gradients here. So, with this yi minus yi minus 1, we know that we are moving towards the which particular side of the error. Whether we are calculating, my outputs are growing, our outputs are reducing is something that I am bothered about.

And that is where I will consider calculating the gradient, sign of the gradient more accurately, rather accurately as compared to the value of it. OK, so this was one way of doing it. All right. One can come up with various different ways in order to calculate the gradients or defining the objective function more appropriately such that you're able to calculate the gradients. So it's a chicken and egg problem for us.

OK, so in totality, what we are doing here now is we have a system which gives me output and which takes input u. This output vector since we are recording the history also it goes into the database, the database command inputs also go into the database, what else goes into the database is r of t, and the new values of k nu, now this new values are calculated with the help of y r of t All right. So, this new values are updated values that goes into the database. The database provides the neighbors or the exact value of the K if the same query has been attached, gives it to the PID controller and the old values are given to the PID controllers.

So, when you fetch database, that time you are getting this particular PID controller values. You apply the same old values which you calculated from the neighbors. When you apply it, you get the new values of y and that you use it to adjust this PID controller and new values are predicted. So, if we look into this particular flow, things are now getting clearer why we considered i plus 1 and ith values and so on and so forth. All right?

Okay. Okay. So this particular is my update step. And this particular part of the block diagram was not there in the previous case, where previous figure, which we are adding it to make sure that even if my initial guesses from the Ziegler-Nichols or whatnot was wrong, we will still be able to reach out and update the same information vector with the new values of k. Depending upon what was my error between yi and yi-1, all right?

Okay. Now comes the last step, which is how to remove the redundant data. Now, step two and once the step four is finished in sample time TS, this needs to be finished in the sampling time TS. All right. And at the same time, it will be dependent on the very large size of the data set N of I. And at least step two, I spoke about how we can reduce time in getting the nearest neighbors and so on.

And this is what we can say instead of searching k nearest neighbors, what we can do is here is the search can be completed when k information vector with certain distance with a threshold are available. So you have phi bar of i as the entry query. As soon as epsilon one distance away This is what is the epsilon one distance uh ball for example just for the representation in 2d i have made it a circle but in this is n dimensional vector right this is a higher dimensional vector now so this particular in this ball epsilon 1 within this epsilon 1 distance if i get multiple k entries, I just finish it instead of accurately calculating the nearest neighbors which are at the smallest distance away from phi bar i, I can apply this trick, this particular trick to get the k nearest values, k nearest neighbors. Since their individual distances are also available to me, I can apply the same weighted sum approach here.

The second way I can reduce the time by not adding the the information vectors which have the highest high similarity in relationship because those are, you know, creating the data size to increase. And then, of course, this first point itself would also be very tedious to get. So, this can be done by calculating the distance between the lth and again I will consider lth element of k vector and the new ki value that we have received. So you can see that when I'm doing the update step here, K new, this K new is going to the database. Now if that particular information vector is already having that particular K value and which is very nearby, of course one can always update it, but the entire information vector is very close to this particular new value, then we should not, so K old and K new are more or less same, which is coming from the neighbor's calculations and so on, then I will not make any new entry into the database.

So this is where I can avoid by saying that, okay, if this particular K of I is calculated with the help of the K neighbors, and this is the new adjusted value, if they are not very far away values, which are, you know, can be considered as having some threshold value here, If this difference is less than the threshold, then let's not make any new entry into the database. So this way, one can decide its own thresholds to consider new values into the database and not to increase the database size here. That's all for the data-driven method. I have covered one methodology for the data-driven control, which is part of this particular book on PID control in third millennium, lessons learned and new approaches.

This particular video and the previous video material and figures are being taken from this book. One can refer this for more understanding of the contents here. Thank you so much. Thank you.