Sliding Mode Control and Applications
Dr. Shyam Kamal
Department of Electrical Engineering
IIT(BHU) Varanasi
Week - 11
Lecture-55

Welcome back. In the previous class, I was talking about the difference equation with a minimabased algorithm for discrete-time systems. And we have seen how to extend the twisting algorithm and the super twisting algorithm back to a stepping-based design for the discrete time system. One of the important aspects is how to design output-based feedback. Because most of the time, if you see the practical scenario, then in order to measure the states of the system, we need the sensors. And suppose that if the order of the system is n , then n number of sensors cannot be fused with the system because the complexity of the system will increase.

And due to that reason, people try to explore the output-based feedback. In this lecture, I am basically going to explore how to design output feedback-based sliding mode control for the discretetime system. And we are going to follow the approach that was actually coined by Professor Bijnan Bandyopadhyay and his group on how to design multirate output-based feedback. Professor Janardhanan has also contributed to several works that are related to multirate output feedback.

So, the purpose of the discussion is basically multi-rate output feedback-based discrete sliding mode control. Here it is possible to show that only we are able to talk about the quasi sliding mode control. Quasi sliding in the absence of disturbance, we can also only talk about the quasi sliding mode. So, now what is our control objective? So, if I have some kind of discrete time system and if I have a fewer number of measurements of the state, then how can I achieve robust tracking or robust stabilization in the case of bounded disturbance? And obviously, we are going to talk about the implementation of output-based feedback. We are assuming that full state measurement is not available, and how to maintain some kind of quasi-sliding mode if some kind of bend is given to us.

It means that in a discrete-time system, if I am in discrete space, I will create a manifold, and after that we are trying to remain bounded after finite time; if I come here, then our trajectory will remain bounded, and finally, what happens is that asymptotically or somehow we become Lyapunov stable in the vicinity of the equilibrium point as k tends towards infinity. And we are also trying to optimize the control computations. And for that, we are going to use the first sample system. So, I have some kind of system, a discrete time system or a continuous time system, where on both sides we are going to add some kind of analog to digital and digital to analog conversion circuit. So, basically, here I have u_k and here I have y.k.

Now, based on these two combinations, y and k, I am going to design some kind of uk, such that I can be able to achieve some kind of desired objective. So, this kind of sliding mode control is called multirate sliding mode control, and since the output is only involved for that reason, we refer to this as a multirate output feedback. And whatever reaching law, because whenever we are talking about sliding mode control, we have two phases: one phase is reaching and another is sliding. So, we are also trying to discuss the multirate reaching law. which is responsible for some part of disturbance compensation.

Obviously, we cannot talk about the sensitivity of disturbance in the case of a discrete-time system because control is only actually available at discrete instants. And we are also going to talk about the lifted system representation. We will see what the meaning of the lifted system is in subsequent slides and, obviously, output feedback-based stabilization. So, most of the systems in nature are represented using the continuous time system. Obviously, we have several discrete time systems that are related to biology due to measurement.

There are several discrete time systems, but whatever implementable means controlled that actually comes into the picture. So, now most continuous time systems are basically controlled by computers or some kind of digital hardware, and for that reason, the implementation of discrete time algorithms has a very high demand, particularly in industry. So, what happens then? Our system is continuous, but whatever algorithm we design is somehow discrete in nature, and for that reason, some degree of stability is required. So, we have to show that even if I do a discrete implementation, then the degree of stability for the continuous-time system is preserved. And obviously, the reason this kind of thing comes into the picture is that our system is continuous and the implementation algorithm is actually based on the computer, and most computers actually follow the algorithm of the discrete principle.

And obviously, all computer control systems create a discrete time closed loop system, and due to that reason, we have to prefer discrete time sliding mode control over continuous time sliding mode control. So, let us try to see some kind of control options. So, one of the beautiful control options is called state feedback. So, that is for both classes of problems: continuous time and discrete time, and the assumption is that all state variables are measurable and one can utilize all state information to design control. It means that if I have a system, like suppose that if I take a very simple system,

$$\dot{x}(t) = Ax(t) + Bu, y = Cx$$

where: $x \in \mathbb{R}^{n\times 1}$: state vector, $u \in \mathbb{R}^{m\times 1}$: control input, $y \in \mathbb{R}^{l\times 1}$: output, $A \in \mathbb{R}^{n\times n}$: system matrix, $B \in \mathbb{R}^{n\times m}$: input matrix and $C \in \mathbb{R}^{l\times n}$: output matrix.

Then I can design feedback based on the information of all states, where k is nothing but some kind of gain matrix.

And it is possible to show you are able to place the eigenvalue at any point in state space if A and B are controllable. A similar kind of thing one can design for the discrete time case, but in reality, all state information is not available, and for that reason, people start exploring static output feedback. What is the meaning of static output feedback? Suppose that this is a continuous time system, and at that time I have some kind of output that is represented by C into xt. In the case of the discrete, our system, particularly if I am talking about the LTI system, a linear time-invariant system, is actually defined like this: at that time, y k is nothing but C x_k. Using this static measurement, we are trying to design the controls.

At that time, it was possible to show that stability was somehow compromised. In several places, if your system order is very, very large, then you cannot guarantee the stability. So, another option is dynamic output feedback. So, basically, dynamic output feedback is trying to estimate the remaining state if the system is observable from the output, and we are assuming that the input is available to us. So, based on that, we will reconstruct the remaining state, and then again we are designing a state feedback kind of controller, but we have to always run one kind of algorithm, and their order is exactly the same as in most cases.

Obviously, we have reduced the order observer concept, but most of the time the order of the observer is exactly the same as the order of the plant, and due to that, unnecessary higher order dynamics come into the picture. So, now another method that is called multi-rate output feedback. So, what is the basic idea? We are going to sample the input and output at different rates, and then we are going to design the control. So, we have two different approaches here because if you see some kind of system, two things are basically accessible here: this is the control side. So, I have UK, and here I have nothing but the output side.

So, this is y k , and the state is nothing but lying inside; this is the x k . So, I have some direct interaction with the output and input, and due to that reason, one can be able to play with the output and input in different ways to implement the control. So, the first option is called periodic output feedback. So, what is the basic idea of periodic output feedback? Control input is sampled faster than the output. And, another approach is that here we are doing the reverse thing: the system output is sampled faster than the input.

So, we are trying to understand both approaches, and after that, based on the second approach, I am going to design the sliding mode control in the presence of disturbance. So, suppose that I have some kind of continuous plant. So, here output is continuously available. I am assuming that A and B are controllable and A and C are observable. So, both condition, it means that system is completely controllable and observable.

And what am I going to do? Now, I am going to sample the output at some kind of tau seconds per interval. But the control I am going to apply is by tau by N. So, how do I do that? So, for that first I have to represent this system in discrete time. So, this is the discrete version of a continuous time system. One can easily convert this continuous-time system into a discrete-time system by using Euler discretization.

Now, here you can see that I have some kind of time interval from 0 to tau. So, at this time, I have some, suppose I have some output y0. So, what am I going to do? Now, I am going to sample this interval, tau by tau, by N. I am defining that as delta. So, each time delta I am going to update the control.

So, how do I update the control? Since y 0 remains fixed between these two time intervals, I am going to change the gain. So, you can see that in each time interval, I am going to change the gain. Okay, the same set of gains is now going to be utilized whenever the next instant comes into the picture, which is 2 tau. So, now tau to 2 tau our output is y tau means whatever output here that is actually updated here. Again, the output is y 2 tau here, and I am going to repeat the same set of gain.

Okay, so in this way, I am actually sampling the output at a lower rate and updating the control at a very fast rate, and due to that reason, first periodic output feedback comes into the picture, and you can easily see that here, x delta delta is nothing but where we are going to apply the first control. So, that is given by $\phi(x_0)$ and multiplied by $\tau k C(x_0)$. So, suppose that we are designing control based on the output. So, here y is equal to C times 0 , and after that same kind of control, I am going to update. So, you can see here that Cx_0 comes into the picture and k_0 is the gain.

So, finally, I have this kind of structure. Whenever we talk about 2 delta, it is because after delta, I have to update the control. So, at that time, what happens now is that a state is updated by x_Δ, and now x_Δ again I am going to substitute like this, and then I will be able to get it like this. So, here gain is updated by k_1. So, this control input comes into picture $k_1 y_0$, and y_0 is nothing but Cx_0.

So, in a similar way, I can proceed, and it is possible to show that finally, I will get this, and in matrix representation form, one can easily write it like this. So, at ϕx_τ you can see here that in this diagram now after that what I am going to do y_τ I am going to update after tau instant only. So, between the time when y_τ comes into the picture, x_τ is given like this. And here, τk is nothing but this matrix; this whole matrix I have defined as τ or $\gamma \cdot \gamma$, you can, this is nothing but capital τ or γ, whatever you want.

Now, with periodic output feedback y_0 and a different set of gains I am using, the same kind of things as I have explained. So, I have explained the same kind of things pictorially here. So, between 0 and tau, I am keeping the information of y_0 while applying control. You can see that we are updating the control like this:

$$\begin{aligned} u(0) &= k_0 y(0), 0 \leq t < \Delta, \\ u(\Delta) &= k_1 y(0), \Delta \leq t < 2\Delta, \\ &\vdots \\ u(\tau - \Delta) &= k_{N-1} y(0), \tau - \Delta \leq t < \tau, \\ u(\tau) &= k_0 y(\tau), \tau \leq t < \tau + \Delta, \\ u(\tau + \Delta) &= k_1 y(\tau), \tau + \Delta \leq t < \tau + 2\Delta, \\ &\vdots \\ u(2\tau - \Delta) &= k_{N-1} y(\tau), 2\tau - \Delta \leq t < 2\tau. \end{aligned}$$

So, now whenever $x_{2\tau}$ comes into the picture, this kind of structure comes into the picture again, and the closed loop tau system can now be represented like this.

And now I have to talk about output injection. By seeing this, I am going to design some kind of G. And how is this possible? If ϕ_τ and C both pairs are observable, then only I can design output-based feedback. And here τk, because we have already seen the expression of that is τk that comes into the picture, so that should be G. Now, there are several approaches, so one of the approaches that is LMI-based approach, so based on LMI you can be able to design this gain, the set of gains.

So, here basically, if you see carefully, then I have to design k_0 up to k_{n-1}; the rest of the terms are coming from the system. So, based on this LMI you can design the gain. So, this is the theory for periodic output feedback. Now, the next theory is fast output feedback, and several old literature sources are actually related to fast output feedback. You can see that the literature dates back to 1957.

So, these papers are extremely good. And now, this multirate output feedback technique is showing, as you can see in this particular diagram, what I am going to do; whatever control I am going to apply is the linear combination of the outputs here L_0, L_1, and L_2, where L_1 and L_2 are the gains of the output. So, first, mathematically, you can see, and again, I will come back to the diagram. So, suppose this is the discrete-time system and this is the output. Now, output samples at every delta second and input applies at the tau second. In the case of periodic output feedback, what was our observation? We are sampling, and we are applying the control at each delta interval, but whatever output we are sampling is tau.

However, here we have exactly the reverse case; we are going to sample output delta, and after that, we are updating control after tau. Okay, so you can see here that somehow, once I sample output at delta, and suppose that at that time our initial condition is x_0, then y_0 is nothing but Cx_0, because of this expression. Now, what is y_Δ ? Once the first sample comes into the picture, that is basically given by C , and I have delta. So, here is ϕx_0, and if you substitute it into this particular system, then you are going to get this. And similarly, I am going to get this at 2 tau.

And similarly, I will get up to $\tau - \Delta$. And then what I am going to do is stack all the output together, because by stacking all the output together, I have to design the control, and the first control u_0 I am going to implement in the system. So, if you see, then this kind of matrix representation comes into the picture, and at that time, I have defined this whole output as y_τ. And, this is the initial condition.

Obviously, how to calculate the initial condition is one of the challenging problems, because I do not know the initial condition of the system.

If you know the initial condition and the system, then one can tell that using the principles of differential equations, I can get all states, and for that reason, x_0 is not available. So, that is the kind of assumption we have. And y tau, 2 tau again I have exactly the same kind of structure; just x 0 is changed by tau and u tau, similar kinds of things y 3 tau that come into the picture. So, finally, if you see the first output sampling system, this is our original system and output in the case of the first sample; I am able to represent here 2 tau, then tau comes into the picture, tau then 0 comes into the picture, and due to that reason, this kind of dynamics comes into the picture, y k plus 1. So, that is just like some kind of dynamics that is actually involved due to some kind of output, faster output sampling, okay.

Now what I am going to do is first design a control like this, and obviously I am stating that whatever control a structure like this has, somehow this will map to this. So, mapping this, I am going to design L0, L1, Ln minus 1, because I have to show that the whole closed-loop system is stable. That is one of the important jobs we have to show. So, how should we proceed? xk plus 1 is phi, and after that I am going to substitute this kind of control here.

So, here also, I have this kind of system. So, from this, one can see now that I can substitute x_k here, provided that this particular term, whatever it is, should be because I am going to represent it in terms of $y\_k + 1$. So, x k plus 1, whatever I have here. So, x k plus I am going to substitute from here. So, due to that reason this should be invertible. Then only I can do this, and after that, from here I can see that this is y_k plus 1 equal to x_k plus 1 .

So, somehow y_k is also x_k, and this is nothing but some kind of constant parameter that is given by this. Now, C is invertible if there are no poles at the origin. So, this kind of thing we have to maintain and f is always in our domain. So, we can ensure that there is no pole at the origin. The delta system is observable because we are creating output-based feedback, and if n is greater than the observability index of delta, then one can easily show that the invertibility condition of this particular matrix holds, because gain design is ultimately related to the invertibility of this particular matrix, and the feedback system can be represented like this.

Why the feedback system? Because finally, I can design control only based on the information of L0, L1, L minus n , and what L0, L1, and L minus n are. So, one can obtain that kind of gain by doing the first output sample, which we have shown: L1, L0, L1, and L n. So now our main aim is to design that gain, so based on this particular expression I can always be able to design that kind of gain, and the only restriction is how to actually obtain the initial condition. So, now we have to give emphasis on the initial condition and how one can select the initial condition. So, the guideline is that you can select any initial condition initially, and finally, due to a bad initial condition, some error will propagate, and we are going to show that this propagation of error is not going to harm the stability of our overall system.

So, how do we prove that? So, again xk 0 into xk means xk equal to xk , and here since I have to apply the u k , but instead of that, I have actually utilized some kind of false information, and due to that reason, some mismatch comes into the picture, and that we have represented like this. So, basically, x k plus 1 can now be represented like this. And I know the x_k plus 1, and after that I know the delta_u k plus 1 ; one can obtain it from here, and if we substitute from the system dynamics. So, you can see here that this kind of equation is actually coming from here if you see u_k.

So, u_k plus 1 is nothing but L_0 into y_k plus 1 . So, from that, I have one expression, and this is the dynamical equation and error dynamics, which is given like this. So, now we have to show that error dynamics will also be stable. Because if error dynamics is stable, then initially we can initialize any kind of control that is the guidelines, and it is possible to show that this is stable because whatever structure is here, this should be stable, and this should be stable because we have a block diagonal form, and due to that reason, it is possible to design L such that I can maintain this. In literature, it is possible to show that L1 can easily design based on that, and there are some kinds of constraints we can always put in place because whenever you are going to implement control, the actuator always comes into the picture, and again, the LMI kind of condition that we have copied from the reference is whatever we have suggested. So, this is the theory for the multirate output feedback and what are the guidelines that you are going to calculate from the past N outputs.

And that will help you adjust the noise amplification, initial control correction, and closeness of the desired state feedback. It means that whatever output feedback we are going to implement is the same as the state feedback. Why? Because whatever gain I am going to select, you can see that I am going to select based on the state feedback only, and for that reason, multirate output feedback is very close to the state feedback, and we know that the performance of state feedback is always better; therefore, the performance of multirate output feedback is also better. Another algorithm you can apply for the same system is the multirate output feedback-based representation of the system we have already seen, and after that, one can compute based on this expression. So, you can multiply by C 0 transpose, and after that, you can actually calculate the state.

You can take the increment again, and after that, you can represent x_k. So, you can see that the previous state and previous control are involved here in order to compute x_k, and for that reason, the current measurement of y_k and the previous measurement of u_k-1 means that one can calculate x_k, and based on that, one can design the state feedback control. So, this is also one of the ways such that by using previous input and current output, you can design the state feedback control. So, this is the methodology using that you can able to design the state feedback control based on the multirate output.

Now, what I am going to do is talk about sliding mode control. So, since disturbance comes into the picture and due to that reason, we can only talk about the quasi sliding mode control. So, obviously, in quasi sliding mode control, in the presence of disturbance or absence of disturbance, we are assuming that our trajectory will always remain in bend. It might be possible that crossing and re-crossing occur, or that crossing and re-crossing do not occur; that does not matter. Whatever disturbance I am going to consider that is a mached disturbance means that it is only going to enter through the control channel, just like in the continuous case. We are assuming that here before k less than 0 , when this system is started, at that time there is no disturbance.

And after that, we assume that this system is controllable and observable because we are going to design the multirate output-based feedback. So, we have selected the same kind of sliding surface, a classical sliding surface, where c transpose is the design parameter, and d k is actually a bounded disturbance. So, d tilde k is actually bounded. So, c transpose d tilde k is also bounded. And it is possible to show that this disturbance, if I know the lower bound and upper bound of it, since d tilde is bounded and c is known, we can calculate if the original disturbance d tilde is bounded.

So, we can calculate the mean and spread of the disturbance because, using the mean and spread, one can design some kind of law that will actually provide better performance in the case of the disturbance. And we have already seen several laws that actually contain the information of either the mean or spread. So, most of the time we are actually incorporating the information of the mean into the reaching law so that I can guarantee that in the presence of the disturbance we can realize the quasi sliding mode control. So, this is one of the laws that is Bartoszewicz's law; whenever I was talking about discrete sliding mode control, we discussed this at that time. So, d_k is the original disturbance and d_0 is nothing but the mean, and after that, s_d_k, which is the design term, and what we have seen is that if delta_d, delta_d is nothing but the actual spread of the disturbance.

So, if x 0 is greater than the spread two times the spread of the disturbance and if these two conditions are satisfied, then one can always show that this kind of terminology comes into the picture. So, somehow this will be an adaptive kind of reaching law, and you can see that if x 0 is less than 2 delta, at that time there is no correction term like this. So, due to this term, one can converge smoothly, and these two terms are actually responsible for the disturbance rejection. Similarly, if x 0 is greater than 2 delta, at that time this kind of algorithm comes into the picture. So, it is possible to show that an infinite number of steps actually is equal to 0 .

And now, based on this particular reaching law, you can take this reaching law and design based on these two logics, and after that, I am able to take s equal to c transpose xk . And after that x k plus 1, I will be able to take ct × k plus 1 , and × k plus 1, I am going to substitute from the dynamics, and s k plus 1, I am going to substitute from the right-hand side of this particular Bartoszewicz reaching law, and it is possible to show that this kind of law comes into the picture. Here, you can see that in order to implement this law, I need this information because this is what we have already designed by the logic; d0 is nothing but the mean of the disturbance. So, everything is actually known, but I now have to show how to measure x k . For that, we again have to represent the system using multirate output feedback, but here the difficulty is that uncertainty comes into the picture.

So, because the original system is uncertain and due to that reason, I first have to represent the multi-rate output feedback representation of the disturbed system. So, for that reason, I have to revisit the theory again. And what I am going to do is develop first the multi-rate output version of the disturb system, and this is one that can give like this. Similarly, whenever we are talking about the multirate outputs, this kind of thing comes into the picture, as we have already seen. And after that, what I have to do is represent the lifted system, and in multirate output feedback, we have already seen the meaning of c and d .

And now, just one extra term comes into the picture. Now, this extra term c d can be defined like this: c 0 and d 0 , which we have already copied from the multirate output feedback, represent the original system. What we have seen is that it is possible to show that from the y_k I can compute the x_k, and then I can replace x_k here in order to design the control action. This is the output equation, and from the output equation, I am assuming that it is observable. If n is greater than the observability index, it means that from the output information and control information, I can calculate the state. Here, actually, Ly, Lu, and Ld are the terms that come into the picture, and in this way, xk can be calculated.

Here you can see that. what basically physical interpretation of xk , that is yk, uk minus 1 and after that, that is the previous disturbance information. Now, what we are going to do, since this feedback is not implementable, is that I will design based on x k because this information is not available to us, and due to that reason, we now have to define a new variable, and again, the new variable we are going to show is bounded because C is bounded, phi is bounded, and L d is also bounded. So, this has some kind of bond, and due to that reason, I can assume that e k is bonded. Now, this is nothing but the spread and the mean.

And so, this is the mean, and this is the spread. Now, again we are going to design the Bartoszewicz reaching law, but here one modification comes into the picture due to disturbance, and here this modification is e k minus 1 minus e 0 . So, that is the contribution of Professor Bijnan Bandyopadhyay's group that modifies the Bartoszewicz reaching law, and after that, it is possible to show that the bend of the uncertainty means quasi-sliding mode band is actually increased by delta e, because obviously, we are going to design control based on the output feedback. So, performance is not exactly like state feedback. Some kind of modification, somewhere we have to pay.

So, we are going to pay in terms of the quasi-sliding mode band. And obviously, this is the algorithm. An algorithm is like that: s d k equals 0 for k greater than k star. And now you can see that whatever control I am going to propose, So, that is actually given like this in this way.

Now, again we know everything: d, e, e, and e. So, all terms are known to us. Now, x k. So, here again we have to drive the control. So, how do you drive the control? We are going to start with this, and finally, we will end up with this expression. Now, we have to do control calculation. So, in order to calculate the control, we have to replace this xk in terms of the y k .

So, I have done that kind of replacement, and obviously, at k equal to 0 , if you do this kind of replacement, then whatever this term is will cancel out, and finally, whatever term remains inside the control is actually a known term. So, this is the control, but initial condition. So, again, the initial condition, I am going to provide it like this. It means that you can give any initial condition. Since, by design, it is possible to show that gain is such that whatever initial condition is finally stabilized.

So, we have taken a second-order system and calculated all parameters based on the LMI, and after that, one can easily see that by designing this kind of control, one can remain in the vicinity of the sliding surface, and all states, which is the system output, will exactly converge to 0 . So now it is time to conclude this lecture. What we have seen in this module, we have basically discussed the discrete sliding mode control. We started with some notions of stability; after that, we talked about classical discrete sliding mode control, then the difference equation with minima-based discrete time sliding mode control, where we discussed higher sliding mode control. Finally, we came to the output-based discrete time sliding mode control, and we discussed two approaches.

So, based on the second approach that is output feedback based approach and using the modification of Bartoszewicz law, we have designed the discrete sliding mode control or quasi sliding mode control. One can actually extend this result again with the help of a discrete difference equation with minima. So, with this particular conclusion, I am going to end this lecture. Thank you very much.