Sliding Mode Observer Design

Welcome back. In the previous class, I was talking about observer design for linear systems as well as nonlinear systems. And what we observed is that, in the presence of uncertainty, one can design a high-gain feedback-based observer, also called a high-gain observer, but that observer suffers from the peaking effect. Now, in this lecture, I am going to explore some different kinds of observers that are based on the philosophy of sliding mode control.

What is our first question? Why use sliding mode observers? And the motivation is again the same: one of the crucial requirements is robustness. So, if I have some kind of linear or nonlinear system, in the case of a linear time-invariant system, $\dot{x}(t) = Ax(t) + Bu(t)$, and $y(t) = Cx(t)$.

So, for this particular system, if there is no uncertainty, then the Luenberger observer is extremely good. If there is uncertainty, then a high-gain observer is okay, but the picking phenomenon comes into the picture. We have some kind of non-linear known term; then again, I can extend the non-linear observer with the help of the Luenberger kind of construction, and then I can estimate the unmeasured state. But once uncertainty comes into the picture and this is the practical situation, I cannot claim that I know the exact mathematical model of the system.

And what was our observation? That observer design is basically based on the philosophy that I know the mathematical model. So, since the mathematical model is not accurate and for that reason, I need some kind of observer that will actually confirm the robustness in the presence of uncertainty, disturbance, or some kind of model inaccuracy. Whenever I talk about sliding mode control, we also expect insensitivity with respect to uncertainty, disturbances, or inaccuracy. And obviously, I have already told you about that traditional observer. And most of the observers are the extension of the Luenberger observer that fails if we have a non-linear system with some kind of uncertainty.

Another class of problems is the fault detection problem because I have already told you that it is equally useful whenever I have to detect some kind of fault; not only detection, I can also perform the isolation. So, if a fault comes into the picture, I will isolate, I will detect, and after that, I will be able to isolate it from the engineering system. And this kind of application is quite useful in aerospace engineering because it is a very, very critical application. If you see the application scenario of the sliding mode observer, it is useful for robotic systems, power systems, automotive control, aerospace, and nowadays people are also using it for biological systems. So, let us start the construction of the sliding-mode observer.

So, you can see here that, in order to understand the philosophy, I am going to start with a linear system. Initially, I assume that there is no disturbance. And obviously, the second step is to introduce disturbance or uncertainty, and then we are going to analyze the performance of the sliding mode observer again. So, for simplicity of presentation, I am going to start with a linear time-invariant system. I am assuming that $x(t) \in \mathbb{R}^n$, so I have $n$states, $u(t) \in \mathbb{R}^m$, so I have $m$inputs, and $y(t) \in \mathbb{R}^p$.

Therefore, corresponding to n, m, and p, these define the dimensions of the system matrix, the input matrix, and the output matrix, respectively. In order to proceed with the observer design, I need some assumptions. So, one of the crucial assumptions here is that the number of outputs is greater than the number of inputs. After that, I am also assuming that the matrices $B$and Chave full rank. And since I am talking about observer design, while for controller design controllability plays a very important role, observability becomes the key property in this context.

Similarly, for observer design, as we discussed in the previous class, observability is one of the crucial properties. For that reason, before proceeding with the observer design, I have to make sure that the pair $(A, C)$is observable. How can I confirm that? For this, I have to check the matrices C, CA, and then CA^2up to CA^{n-1}, and I have to show that the rank of this observability matrix is equal to n. This verification is relatively easy because I am dealing with an LTI system. However, if we move from an LTI system to an LTV system, that is, a linear time-varying system, then this condition has to be modified accordingly. Now, in order to design an observer, I need the system to be expressed in some specific form. It is possible to show that, if the system is converted into a suitable specific form, then observer design becomes systematic.

What kind of specific form is required? You can see here what our assumption is: that the output belongs to $\mathbb{R}^p$. What does this mean? I have some kind of system. I have the input $u(t)$, the LTI system is sitting here, and I have the output $y(t)$. Now, how many states $x(t)$are required to characterize this system? The idea is that some part of the state vector appears explicitly in the form of the output, and how the output appears is actually driven by this particular matrix. I also know that $y(t) \in \mathbb{R}^p$.

So, what am I going to do? I am now going to apply some kind of transformation. This transformation is of the form $z = T_c x$. So, I am going to design T_c in such a way that I can transform the state $x$so that the $p$outputs explicitly appear in the transformed coordinates. In this way, I can decompose the system into observable and unobservable states. The directly observable states are those that are explicitly associated with the output and, consequently, with the control.

So, I know the output, and I know the control. Therefore, I can guarantee that the lower part of the state is directly observable, and the upper part is not observable. This is the physical interpretation of this particular transformation and how it is constructed. So, what can you do? You can calculate the null space of the matrix C. What is the definition of the null space? Given the matrix C, you can select a vector $v \in \mathbb{R}^n$ such that $Cv = 0$.

So, based on that, one can create this kind of transformation. Once this transformation is ready, you can see that since $y = Cx$ and I have already defined $z = T_c x$, what actually happens is that the inverse transformation $x = T_c^{-1} z$ comes into the picture. And for that reason, if I substitute this transformation into the original system, I will obtain this kind of distributed matrix structure after the transformation. Now, once you apply a transformation like this, it is possible to show that the system matrix is converted into an unobservable subspace and an observable subspace. Why is this an observable subspace? Because it is directly associated with the output. You can see here that the term A_{21} directly affects the output, since the state x_1 contributes to y. And obviously, under this transformation, the input matrix is also partitioned, and it becomes B_1 and B_2.

So, this is the complete system after the transformation. You can see here that $T_c x$is the transformation that I have applied. Our original system, which is given by

$$\dot{x} = Ax + Bu$$

and $y = Cx$, can now be expressed in this transformed form. In this representation, one part corresponds to a state dynamics, another part corresponds to the remaining state dynamics, and this part corresponds to the direct output dynamics. What is the interpretation of this transformed system? I can understand it in the following way: I know $u(t)$, which is nothing but the control input, and I also know $y(t)$, which is the measured output.

So, at least this many states are directly observable, and the remaining states I am going to observe from the first equation. I am going to design the observer. So, how do you design the observer? Since everything is known here, because the transformation is known, I can assume that both $B_1$and B_2 are known. Now, what am I going to do? I am going to add the correction terms. Similar to the Luenberger observer, I am going to copy the dynamics.

So, the same controller that I am using, I am going to utilize for the observer design, and due to that reason, here the exact control input $u(t)$ is going to be used. So, after copying the dynamics, I am going to add the correction term. Here, $L$is the feedback gain matrix. The term that I am going to use has the appropriate dimension and is denoted by ν, which is a discontinuous term. And this discontinuous term $\nu$is now going to be designed based on the sliding mode theory.

So, $\hat{x}_1$and $\hat{y}$are nothing but the estimates of the actual state $x_1$and the output $y(t)$. Now, again, our main goal is that by designing this correction term, I can guarantee that $x_1 \to \hat{x}_1$ and $y \to \hat{y}$. This is our aim. So, for that, I have already told you that this correction term is designed based on the philosophy of sliding mode control. You can see here that $y(t) \in \mathbb{R}^p$.

So, I have p number of outputs here. And in order to design sliding mode control for a multi-input, multi-output system, we already know that there are two different design philosophies. Either you can design the sliding mode observer component-wise, or you can use the concept of a unit vector. So, both approaches I am going to adopt in this particular lecture. So, first, what I am going to do is discuss the component-wise design.

So, $y$has $p$number of components. So, the i-th component of the discontinuous term is defined as

$\nu_i = M_i \operatorname{sgn}(y_i - \hat{y}_i)$,
where M_i is some positive gain and $\operatorname{sgn}(\cdot)$ denotes the signum function. And after that, what is our goal? I have already told you that by designing this discontinuous term, I have to force $y = \hat{y}$. So, I have defined the error $e = y - \hat{y}$, because in order to design the gain, I have to work with the error dynamics. Once the error dynamics are defined, I am going to apply Newton's philosophy, that is, I am going to calculate the rate of change of the error.

So, $\dot{e}_1$is obtained from $\dot{x}_1$and $\dot{\hat{x}}_1$. I substitute the dynamics of $\dot{\hat{x}}_1$from the observer and the dynamics of $\dot{x}_1$from the nominal system after the transformation. If you carry out the substitutions properly and define the error as $e = \hat{x}_1 - x_1$, then you will end up with this particular error dynamics. So, now the system is expressed completely in terms of the error dynamics. Once the system is written in the error coordinates, our objective is clear: I have to show that the error tends towards zero, that is, $e \to 0$, and at the same time the output error $e_y \to 0$ as $t \to \infty$.

It would be even better if I could show that this convergence happens in finite time, that is, as $t \to T < \infty$, the error goes to zero. Most of the time, using classical control theory, this is not possible. However, when I discuss higher-order sliding mode control, I will show you that this kind of finite-time convergence can be achieved.

Now, during the design of the gain and the switching function ν, you can see that there is one more gain M involved. To properly select this gain, I need some further assumptions, because I am only able to directly control this part of the dynamics.

Somehow, this part of the dynamics is directly controllable, and after that, whatever effects arise from here will guarantee that the remaining dynamics are also stable. From the beginning, I am assuming that the pair (A, C) is observable. And due to that reason, the pair (A_{11}, A_{21}) is also observable. This can be shown easily.

If this pair is observable, then I can always design a gain L such that I can place the eigenvalues of the corresponding closed-loop matrix in the left half-plane. This condition is very important. Otherwise, it is possible to show that, once sliding is established, if this condition is not satisfied, then I cannot stabilize the reduced-order dynamics.

In order to further simplify the gain design for the error dynamics, I am going to introduce another transformation. You can see that this transformation is basically based on the gains. Here, I_{n-p} denotes the $(n-p) \times (n-p)$identity matrix.

Now, I am going to utilize this kind of transformation. After applying this transformation and substituting $\tilde{e}_1 = e_1 + Ly$, it is possible to show that the upper error dynamics are converted into this particular form.

You can see that the lower dynamics remain the same, because the first block here is 0, and after that this block is I_p. So now, our transformed system is expressed in terms of $\dot{\tilde{e}}_1$and e_y.

And here, $\tilde{A}_1$, once you carry out all the substitutions, you will be able to easily verify that this term appears naturally.

Now, what am I going to do? I am going to show that the control explicitly appears only with respect to e_y. What was confusing in the previous dynamics? You can see that I had a certain number of control terms; the control was actually based on the error dynamics, and using that, I was trying to stabilize e_y, but that dependence was not very clear earlier and appeared in a coupled manner.

So, due to this transformation, things now become completely clear: the control appears only in the dynamics of e_y. And why am I calling this a control? Because this term is nothing but the correction term, and the correction term effectively acts like the control input in these error dynamics.

So, by designing a correction term that acts like a control for the lowermost subsystem, it is possible to show that $e_y = 0$. So, I have to introduce sliding mode here by designing μ as a switching-type control. Now, if you look at the dynamics of the lower subsystem, you can see that this term is actually influenced by the upper dynamics. And due to that reason, you have to construct some kind of error-dependent bound so that you can introduce the sliding surface properly. Because of this coupling, I am not able to talk about global stability at this stage.

Because whatever bound exists here depends on the error, and the error is directly related to the state. But obviously, by choosing this gain to be very, very large, you can introduce sliding mode control from almost anywhere in the state space. That is something you can always do in principle. So, what we have observed is that, in this particular dynamics, the term $\tilde{e}$ appears explicitly. And due to that reason, whatever gain I design to introduce the sliding mode must depend on the magnitude of the error.

It means that if I start anywhere in the state space, it becomes difficult to guarantee convergence to the equilibrium point directly. And due to that reason, most of the time I assume that all trajectories evolve inside a compact set. Because of this compact set assumption, I am able to select the gain appropriately, enlarge the region of attraction, and achieve sliding mode control using the correction term.

So, I am going to design the gain M. If $\| \eta \| < M$, then I am able to satisfy the reachability condition. What is the meaning of the reachability condition? It means that I can introduce sliding mode in finite time. And since the region is fixed because of this gain selection, in finite time I can enforce $e_y = 0$, and after that, I am able to maintain $e_y = 0$ for all subsequent time.

So, during the maintenance of sliding, it is possible to show that an average kind of control, called the equivalent control, comes into the picture. How do we obtain this equivalent control? Because sliding occurs with respect to the error e_y, and for that reason, on the sliding surface we have $e_y = 0$ and $\dot{e}_y = 0$. So, by substituting these conditions into the error dynamics, you can design or compute the equivalent control. In this way, I can calculate what kind of equivalent control appears when the system is in sliding mode. So, first, I have to show that the reachability condition involving $\eta$is satisfied by the chosen gain M.

So, now you can easily see what I have done. This is the dynamics of the error. I have multiplied the error dynamics by $e_y(t)$, and after that, I have used an inequality of this form. So, once you substitute this inequality into the error dynamics, you obtain a reachability condition that is satisfied. Now, once the reachability condition is satisfied, I can guarantee that $e_y(t) \to 0$ as $t \to \infty$.

And now, I am able to achieve sliding mode in this dynamics with $e_y = 0$, and our domain is also fixed. I have already told you that by designing the gain M, you can enlarge this domain. And once sliding occurs, what happens is that $e_y = 0$, and at that time the dynamics reduce to a reduced-order system.

In this way, you can now see that I am able to place the poles, or equivalently the eigenvalues, of this reduced-order matrix by appropriately designing the gain L. And in this way, I can guarantee that $\tilde{e}_1 \to 0$ and $e_y \to 0$. Since $\tilde{e}_1 \to 0$ also implies $e_1 \to 0$, this confirms that the overall estimation error converges to zero.

In this way, I can obtain the state estimate. So far, I have assumed that there is no uncertainty. Now, I am going to introduce uncertainty as well and examine the performance of the sliding mode observer. So, here you can see that during the construction of the sliding mode observer for an uncertain system, what I am going to do is design a gain matrix G_n. This gain matrix contains the gain L, and it also contains the identity matrix I.

So, I am assuming that the matrix Mcan be expressed in this form. And after substituting this expression, this becomes our new error dynamics. Now, using the same philosophy as before, you can see that I am going to introduce component-wise sliding mode control. At that time, the gain has to be chosen differently. Why? Because now I have some extra terms coming into the picture due to the uncertainty, and these additional terms must be compensated for by appropriately modifying the gain.

So, I have to take care of that particular extra term during the gain design as well, and it is possible to show that $e_y = 0$. Now, since $e_y = 0$, I have already told you that $\dot{e}_y = 0$ also holds. And due to that reason, I am going to substitute these conditions into the dynamics. Because in order to maintain sliding mode control in the presence of disturbances, we have to account for the disturbance explicitly. Here, I also have disturbance.

So, I am going to apply some kind of equivalent control so that the sliding motion is maintained. Automatically, this equivalent control comes into the picture once sliding is established. And due to that reason, everywhere in the reduced-order dynamics, I am going to substitute the control term by its equivalent control.

This occurs automatically. So, there is no need to explicitly design the equivalent control ν_{eq}, because once sliding occurs, this control automatically comes into the picture. Otherwise, it would not be possible to maintain the trajectory along the sliding surface. So, this gives us flexibility in the design and analysis. You can see here that I can compute the equivalent control from this condition, and after that, I substitute it into the dynamics. Then I obtain this form of the reduced-order dynamics.

Now, I will design this gain. You can see that, irrespective of the uncertainty, if the gain is chosen to be sufficiently large, then even in the presence of uncertainty I am able to force $e_1 \to 0$ as $t \to \infty$. And in this way, I can achieve sliding mode observation.

Now, what am I going to do next? I am going to take one practical system. I have already discussed the robot manipulator dynamics, which are given by a fully actuated system. This belongs to the class of fully actuated systems, where I am assuming that some kind of matched uncertainty is introduced through the control channel. Now, what is our goal? If I know the joint position, then I have to estimate the joint velocity.

This term we have already discussed, and $\tau$is the control input. So now, what I am going to do is design an observer. I am going to assume that $x_1 = [q_1 \quad q_2]^\top$ and $x_2 = [\dot{q}_1 \quad \dot{q}_2]^\top$.

The estimated states corresponding to $x_1$and x_2I am going to represent by $\hat{x}_1$ and $\hat{x}_2$. So, the first step I will take is to convert the system into a state-space representation, and then I will design an observer. Whatever part of the dynamics is known, I am going to include it inside the observer. I cannot include the unknown part of the dynamics inside the observer.

So, please be careful here. For design purposes, errors will automatically occur whenever you are designing an observer, and at that time you have to be cautious. Why? Because the observer is running inside a computer. So, if some term in the dynamics is unknown, you cannot include that term explicitly in the observer, since you cannot program something that is unknown.

Now, I am assuming that with this particular structure of matrices, you are able to choose another appropriate set of matrices. And then, after performing the simulation, you can see that I am able to obtain accurate estimates of $q_1$by $\bar{q}_1$ and $q_2$by $\bar{q}_2$.

So, this is just the graph. Now, this is the program. So, I am going to share the file, and if you run your program in MATLAB, you will be able to get the result. Now, what am I going to do? I have told you that in sliding mode control, we have two different philosophies. If you have a multi-input, multi-output system, then you can introduce sliding. in two different ways. So, what is the first way? In the first way, it is possible to show that you are able to do component-wise analysis.

Second is based on unit vector control. So, I am going to discuss how to design a discontinuous observer. Why is that observer discontinuous? Because the sliding variable becomes 0 at once, it is not defined at that time. So, again I am going to take the system because we are actually designing sliding mode control, because in the presence of disturbance, I cannot estimate. For that reason, I have taken a realistic example.

Again, you can see that I am assuming exactly the same thing. Here, I am assuming that P is the output. So, the dimension of the output is greater than the dimension of the disturbance. This kind of assumption assumes that our uncertainty is bounded. then you can able to give guarantee that this system has some kind of unique solution throughout the time interval.

That is very, very important. Now, I am going to do the canonical form representation again. It means that I am going to transform into a convenient form such that I can design an observer, because I have p number of outputs. So, now I am going to convert system in some convenient form, same like the previous. Now, what am I going to do? I am going to design the correction terms. Here, you can see that correction term I have designed based on another philosophy.

Obviously, the structure is exactly like the Luenberger observer. What am I going to do? I am going to copy these two dynamics while removing the unknown dynamics. Why? Because those terms are not known, and due to that reason, I cannot keep them inside the computer. I am assuming that y is known to us, and for that reason, $\hat{y}$, which is generated inside the observer, is also known to us. So, this part is known, and due to that reason, I can write the correction term using $y - \hat{y}$ here.

Okay, the second correction term I am going to design in this form. You can see here that this structure should already be familiar to you from state-feedback design; this is commonly referred to as a unit-vector–based control. More precisely, you can call it output-feedback–based unit vector control, because it is constructed directly from the output error and normalized by its magnitude.

Now, what do we have to do? I have to select the gain value to be sufficiently large so that I can introduce sliding with respect to $\hat{y}$. At the same time, I have to design the matrix P_2 appropriately so that the sliding condition and the stability requirements are satisfied.

Based on some Lyapunov equations, I have to design this, just like in the previous analysis. Now, I am going to keep this gain ρ, which is the gain of the discontinuous term, to be greater than the bound on the uncertainty. I am also going to keep a small positive term so that I can introduce sliding mode control. In this way, I can force the error dynamics to become equal to 0in finite time.

If the error becomes equal to 0in finite time, then obviously I can show that this particular dynamics is stable. So, now the error is equal to 0, which implies that $e_y = 0$, and the equivalent value of its derivative also satisfies $\dot{e}_y = 0$.

So, based on that philosophy, I can again proceed further. This part I have basically taken from a standard book on sliding mode control, which is a very good reference for linear systems. I have to design the matrix P_2, because here you can see that P_2 should be designed based on a Lyapunov equation. Due to that design, I am going to proceed further.

So, you can see the matrices $P_2$and A_{22}. The matrix A_{22} appears because I have designed this part of the dynamics to introduce sliding mode control with respect to the variable $\hat{y}$. Due to that reason, I have formulated a Lyapunov inequality, where Q_2 is a positive symmetric definite matrix. By now, you should be very familiar with the meaning of a positive definite matrix.

So, this matrix is symmetric. Now, what is the meaning of positive definite? You can check the dimension, multiply it by a vector of appropriate dimension, and you have to show that the resulting quadratic form is always greater than 0for all nonzero values of y. After that, I have defined $\bar{Q}$ in this way. So, you have to carefully check this calculation. Now, after that, for the matrix A_{11}, I am assuming that there exists a unique solution for this corresponding Lyapunov equation.

Again, it is possible to show that $Q_1$is positive definite. So, $-Q_1$is negative definite and symmetric. Hence, $A_{11}$is Hurwitz. Then, obviously, the Lyapunov equation admits a unique solution P_1.

So, that kind of result can be shown using classical control theory. And after that, I am going to design the Lyapunov function. Since I have the error variables e_1 and e_y, I take P_1 and P_2 as positive symmetric definite matrices. So, the energy associated with e_1 and the energy associated with e_y are calculated using these matrices, and this energy is a scalar quantity.

So, I am able to add these energy terms. And after that, I calculated the derivative of the Lyapunov function. I have substituted all the corresponding terms from the error dynamics. So, please check this calculation carefully. After that, the discontinuous part is retained as it is, and the uncertainty term $\zeta$is kept according to our assumptions. Now, if you substitute everything properly, then it is possible to show that $\dot{V} \leq 0$.

Now, it is also possible to show that by taking this dynamics, $e_y = 0$in finite time. So, how do you do that? Because I want to introduce sliding mode across this particular dynamics. So, if I am able to show that the overall system is stable and this system is actually $e_y = 0$in infinite time, then our job is done. So, I have already shown by Lyapunov analysis that the overall system is stable.

Now, what do I have to do? I have to show that $e_y = 0$in finite time. So, how do we show that? For that, I have defined a Lyapunov function in terms of e_y. I have calculated its derivative $\dot{V}$. I have substituted each and every term, and again you can see here that if our gain is chosen appropriately. So, every time, what is our observation? The gain of the sliding variable somehow depends on some part of the state and some part of the error. Due to that reason, if I start with a compact set, then every analysis becomes useful for us. In this way, from $\dot{V}_s$, I can show that the system is finite-time stable. And once that is finite-time stable, you can see that our job is done, because $e_y = 0$is finite-time stable. The matrix A_{11}, whatever $A_{11}$is, I am assuming that it satisfies the Lyapunov equation.

So, this dynamics is stable. So, overall dynamics is stable. So, now it is time to conclude this lecture. So, what have we seen? I have started the sliding mode control design from a simple LTI system with no disturbance. When disturbance comes into the picture, I utilize the concept of equivalent control to compensate for that disturbance. And after that, I also talked about how to design sliding mode control based on the unit feedback strategy.

So, what is our key takeaway? Obviously, sliding mode control is highly robust. In the presence of uncertainty, we have also demonstrated that the state can be estimated. I have taken an example of a robot manipulator. Whatever components are known can be recovered in finite time, but the remaining components can only be recovered asymptotically because the $e_y$dynamics tend towards 0 as $t \to t_f$, whereas the $e_1$dynamics tend towards 0 as $t \to \infty$. Therefore, I cannot claim that the estimation occurs entirely in finite time.

And obviously, this is widely applicable to various classes of systems. And I have also give the guarantee with help of the Utkin's theory, Utkin's equivalent theory as well as the theory of Lyapunov. So, what is our main conclusion from this particular lecture? Sliding mode control is the ultimate robust observer solution for the uncertain dynamical system. So, with this remark, I am going to end this class. Thank you very much.