# Sliding Mode Observer Design for Nonlinear Systems Application to Fault Estimation

Welcome back. So far, I have been discussing the observer design for the linear system as well as the linear system with disturbance using sliding mode control. In this lecture, I am going to extend that result for the nonlinear system, and again I am going to use the philosophy of sliding mode control to design the observer. So, for the purpose of discussion, we have to satisfy the following key objectives. So, you can see here that in the presence of disturbance, I obviously need some kind of robust observer to estimate a state from the information of the output and input. So, in the previous class, we have already seen that in the observer design, the assumption everywhere is like this: I know the input, I know the output y(t), and I know the mathematical model of the system.

So in the previous class, I took a mathematical model like linear time invariant with or without disturbance. Okay, now I am going to take a mathematical model that is fully non-linear. Okay, because all physical systems are non-linear in nature, and due to that reason, practically this. This particular topic is important from a practical implementation point of view.

And after that, we are also assuming the observability condition; particularly, this is a non-linear observable design, and for that reason, we are going to assume that the uniform observability concept is satisfied. And under that condition, I am now going to estimate all states from the output and input representations. That is our key objective. So after that, we are also going to show that using sliding mode design, we have to show that finite time convergence occurs with respect to the sliding surface. And after that, obviously I am going to show the disturbance rejection capability, and practical implementation is also very, very important.

So, I am going to implement this idea in the robotic system, and after that, I am also going to discuss some challenges. In this lecture, I am also going to tell you how to apply sliding mode philosophy to actually detect the fault. So, the outcome of this lecture is a robust sliding mode observer design for the non-linear system with finite time convergence to the sliding surface. Obviously, I am not able to estimate in finite time. So, for that, I am going to discuss another class of the observer that is based on higher-order sliding mode control in the subsequent lectures.

And obviously, the sliding mode observer is robust with respect to disturbances. So, this is the description of the system. I hope that you are already familiar with it, because I have already discussed nonlinear observer design. So, basically, this term is going to contribute to the nonlinearity, and I am assuming that this particular term is known. I am also assuming that this term is uniformly Lipschitz with respect to $u$, and under this assumption, I am able to obtain some Lipschitz constant $L_{\phi}$.

Now, another assumption I have is that this represents the unknown function, and this unknown function is bounded by some known function. So, obviously, the exact form of f(y,u,t) is not known to us, but I am assuming that it is bounded by some known form. Another important assumption you can see here is that p is somehow related to the output vector and q to the disturbance. So, I am assuming that the dimension of the output is greater than the dimension of the disturbance and less than the number of states. So, under that assumption, I am now going to proceed toward some more assumptions, and what is an assumption? You can see here that I have two matrices: one matrix is the output matrix, and this is the matrix through which the disturbance is going to enter.

So, I am assuming that the rank of this is equal to the rank of D. This means that this is an unknown input because it is not known to us, and for that reason, it is called an unknown input that is affected by the output. That is the meaning. After that, we have already seen that whenever we are talking about the sliding mode control, we basically have two phases if we are just plotting a two-dimensional plane. Obviously, plotting in a higher-dimensional plane does not give us a clear visualization, and for that reason, I am going to plot in this plane, but this methodology is equally applicable for any higher order.

plane. So, what are we basically doing? We have a phase called the reaching phase, and after that, I have to maintain the sliding phase. In the sliding phase, our system is essentially independent of control by design philosophy; for that reason, during sliding, I do not have any injection term that will control the sliding. Therefore, some assumptions are required, and those assumptions are represented by these three pairs: A, D, and C. And we are assuming that that particular reduced order system is somehow minimum, representing the minimum phase system, because this ensures that our sliding mode becomes stable. So, our sliding phase become stable.

So, for that this assumption is required. I have already told you that one can utilize an observer for actuator fault detection as well. So, at that time, some further assumptions are required. I am going to assume at that time that somehow u is going to appear in terms of something like B, and at that time I am going to assume that the dimension of B and the dimension of D are equal. So, basically, those are the kinds of things I am going to note here, and after that, now in this observer design, particularly the sliding mode observer.

Can design to estimate x; that is our core objective. After that, I have to design an estimate of $\bar{x}$ in the presence of the unknown input; that is our key goal. In the literature, you can see that people are also mentioning the unknown input observer. And obviously, I will achieve robustification through the sliding term. And the sliding term I am going to design is based on the output information only.

So, now I have already told you about the assumption. So, basically, this assumption is related to the minimum phase system. So, how do you calculate invariant 0, because some of you may or may not be familiar with it? So, you have to calculate this matrix that is called the Rosen-Brock matrix, and we have to show that it will not lose rank. So, this condition should be satisfied. So, somehow we are basically assuming that since the order of the system is n and I have only q number of outputs.

So, at least n – q states that are somehow unobservable from the output. That is actually a stable. So, how is that stable? If I assume this, then this particular kind of thing is critical for observer and controller design, and I have taken a very simple matrix. So, this is homework for you; please check it. Now, in order to design a sliding mode controller or observer, I required some kind of transformation, a specific form I required, because I have to inject some kind of discontinuous gain.

So, here if you interpret the system, this is our system, original system here. So, suppose that if φ is not there and I have just ẋ equal to A, D, and C matrix, and if I assume this as some kind of unknown input, you can see that I can visualize this system as an LTI system. So, I am going to apply the transformation for that particular system only, and obviously, based on this transformation, whatever φ term that contains the x and u is also going to transform. So, what this transformation is suggesting is that A matrix is actually converted into this kind of form. Similarly, the D matrix will convert like this.

Several times I have used these kinds of transformations, and now you can see that I get this kind of transformed system. Here, the phi matrix $\varphi_1$ and $\varphi_2$ are also separated due to the application of this transform x, that is x = $T_c$ x. Now, you can see that for the convenience of the observer design, what I am going to do. I am again going to introduce another coordinate transformation, and at this time, this coordinate transformation is actually depending on some kind of gain matrix L. So, under this transformation, now our system is this.

Okay, and this particular system is quite suitable for the observer design. So, what do you have to do? You have to take original system, you have to apply this kind of transformation and after that you have to apply this kind of transformation. So, here during the application of this transformation our objective to design L. So, one of design parameter basically came here for the observer, okay. Now, how do we design an observer again? So, philosophy is exactly like the Luenberger observer, but you can do whatever is known; you should copy it.

So, this term is known to us; I have copied it here. You can see that everywhere I am writing $C^2y^{-1}$due to the transformation because I know that everything will transform $Cy$here, so that will transform, and due to that reason, this term $C^2y^{-1}$comes into the picture. This is what I am assuming: that $\phi$is actually known, but $\phi$ is partially known because it contains the information of $x$, and during observer design, I only know $\hat{x}$, and for that reason, I have made that kind of substitution everywhere. So, in place of $x$, I have applied this kind of transformation. I hope that you have become familiar with this.

Now, what is our main goal? Our goal is to actually inject this discontinuous term, and how we have basically designed this. We have designed this based on the unit vector approach.

So, I have to select $k$ gain. So, I have to select $k$ gain such that I will actually decompose the whole motion into two parts, that is, the reaching part as well as the sliding part.

That is our main goal. So, in order to make sure that everything is stable, particularly regarding stability prospects, I have to be very concerned about the sliding phase. Because during the reaching phase, I have a discontinuous term. So, this continuous term forces the trajectory to land on the sliding surface, but during sliding, I have to be very, very careful. So, I am assuming that the $k$ gain I am going to select is such that this whole matrix becomes symmetric negative definite. Why are these kinds of assumptions required? Because finally, I have to prove stability using the Lyapunov method.

And obviously, I have unknown function and $\nu$ ensures the finite time stability. I have already told you that I have two semesters. So, $\nu$ term is very, very crucial for us that will actually somehow give us the guarantee that if you start anywhere in this particular region, so you can force to the sliding surface. Now, whenever I design the observer, how do I design the gains? So, a strategy is exactly like Luenberger's. What can you do? You can define the errors.

So, this is the actual state $z_1$, and this is the estimated state $\hat{z}_1$; similarly, this is the actual output $y$and the estimated output $\hat{y}$. So, if you look carefully, what we have done is apply two transformations such that I will get some kind of convenient form, and after that, I will actually express the system in terms of the error dynamics. I will take the rate of change since I have to control the error equal to 0, and for that reason, I am going to control its rate of change $\dot{e}$. So, philosophy is exactly the same and after you can verify that we will get these kind of equations. So, whenever you are going to design an observer for a nonlinear system, a lot of calculations are involved.

So, what is my suggestion? Please try to take a pen and paper, and after that, try to repeat all the calculations. And then what can you do? You can design the discontinuous injection. Now, I have already told you that the gain matrix $K$ we are going to select is such that we will get something that is negative definite. Now, what am I going to do? I am also incorporating one more matrix $A$, and now we are trying to select an expression in such a way that this whole expression becomes negative definite. One more important point here: you can see that $\phi(T^{-1}z, u)$, and after that, I have this term.

So, what I am assuming is that if these terms are not present, in the absence of this term, it is somehow inherent to this term. So, if you see the error dynamics carefully, then I have somehow an LTI system with disturbance. It means that I have a simple system $u$again, and here I have $z$, and $z$ is nothing but what will be expressed in terms of error, and I have some kind of bounded perturbation that is actually reflected due to the Lipschitz constant. And due to that reason, if our nominal system is stable without any kind of Lipschitz perturbation, then obviously it is also stable if some kind of Lipschitz perturbation comes into the picture. That is just the consequence of the bounded input–bounded output stability of the LTI system.

And due to that reason, this condition basically comes into the picture. So, in the nominal linear system, we are actually designing $K$ in such a way that $K$or $L$ are the design factors, because $C_2$ is coming from the system dynamics, and $A_{11}$ is coming from the system dynamics. So, just two parameters I have, $K$and $L$, we are going to select them such that this whole matrix is stable. What is the meaning of stable? It means that all eigenvalues of this matrix should lie in the left half-plane. What do we have to show? We have already seen that our dynamics is represented in the form of $e_1$and $e_y$.

So, the direct injection term associated with $e_y$ is the injection term. So, what do we have to do now? We have to prove two things. I have to prove that $\nu$is, and I am going to select such that $e_y = 0$ an infinite number of times. Once $e_y = 0$ infinitely, at that time this dynamics is actually stable. So, both proof is required, and obviously, I am going to proceed using the Lyapunov method.

So, now, error dynamics are decoupled due to the choice of $L$ and $K$. Here, why am I saying this is decoupled? You can easily see that there is no term of $e_y$ actually involved in the first dynamic. So, independently you can able to do the analysis. That is the beauty of transformation. Now, sliding mode ensure infinite time and observer is robust with respect to unknown disturbance.

Here, I have not written $f(t, x, u)$. Now, you can see that it is possible to show that if this linear matrix inequality, if this matrix is $\leq 0$, then I can design the reduced-order dynamics; reduced-order dynamics means once this collapses, at that time it is also stable. So, using Lyapunov, it is possible to show that this big expression comes into the picture, where $P_N$is nothing but $P$, $I$ is the identity matrix, and $L$ is the gain, and $a_n$ is $A_{11}, A_{21}$. Now, again, this system I have already told you about is represented by some kind of linear system with some kind of perturbation, some kind of Lipschitz perturbation. And due to that reason, you can always be able to get some kind of Lyapunov function, which is a quadratic Lyapunov function, where $P$is a positive symmetric definite matrix. What is the meaning of positive symmetric definite? $P = P^T$, and you have to make sure that all eigenvalues of this matrix are considered.

Okay, that is strictly positive. And now what I am going to do is, again, apply the philosophy of Lyapunov. So, somehow our error dynamics is transformed in one dimension. Why is this one dimension? $E_1$is somehow a kind of vector; transpose it if you will, then 1comes into the picture, along with the $P$matrix I have selected. So, this is one dimension. So, what is the way to apply Newton's law? Since $v$is positive, you just have to maintain $\dot{v}$negative; then you will be able to converge towards $v = 0$.

So, that kind of philosophy I am going to use. Now, what am I going to do? I am going to take the $\dot{V}$, and I have already told you that $e_1$dynamics and $e_y$ dynamics are both the decoupled dynamics. So, you can easily perform a separate analysis, but here you have to understand some kind of inequality. So, this inequality is very famous in linear control theory and robust control theory. So, if you have some kind of vector $X^T Y$, then you can always separate them. You can see that here only the $X$term will appear and here only the $Y$term will appear, and after substitution, you will get this kind of expression.

Now, you can see that I can further simplify this kind of expression as well; why? Because this will satisfy the property of the Lyapunov function, I have written it this way. So, by using this, you can easily get. Now, finally, I will get this big matrix, and what is our main goal? I have to put this $\leq 0$, then automatically $\dot{V} < 0$. So, now this also conforms to the exponential decrement of $V$. So, if you select some kind of matrix such that apart from this additional term, if I add this term, I will show that it is $\leq 0$.

So, in place of this, I can substitute $-\alpha P$, and then I can confirm the exponential stability. It means that I will design gains $K$and $L$ such that I will achieve this kind of exponential stability. And in this way, I can make sure that $e_1$is actually tending towards 0 as $t$tends towards infinity. And now there are several toolboxes that are available in MATLAB using which you can confirm or design a gain such that you can prove that this matrix will satisfy the $\leq 0$ criteria. So, what I suggest is that you go to MATLAB, and after that, you can solve this matrix inequality.

This matrix can be easily represented like this; you can easily check this. Now, once you solve the LMI, there is one more important thing to note: since $V = \alpha V(t)$ by designing this matrix, we have already seen. So, this system will represent some kind of scalar system with some kind of inequality. So, I have told you already that whenever you have some kind of differential inequality, you have to convert it into a differential equation, and you will be able to solve it easily. So, with that philosophy one can able to see that solution can looks like this.

So, now, compression system I can able to make and always I can able to show that I can able to bound the solution of $e(t)$ by this $\omega^T$. Why am I doing this? Because this kind of information will be utilized whenever I select the gain of the switching term, the $e$term will also explicitly appear there, because whatever disturbance I have considered is basically a function of $t$, and after that, $y$is also a function of $u$. So now, due to that reason, whatever $k$I am going to select, so I am telling that error can actually, I do not know where error is, and due to that reason, I have to increase our gain based on this compression system such that error will always compensate it.

That is our philosophy. And by selecting this particular gain, it is possible to show that if I apply this, then in finite time $u_1 = 0$. So, again, how do we show this? So, you can see that several terms are associated with this particular $u$.

. So, I have provided the interpretation. So, this represents the system dynamics; this is the

Lipschitz nonlinearity and the unknown disturbance. It means that whenever you are selecting $k$, $k$ contains several subterms. So, why are several subterms required? So, this will give you, this slide basically gives you the interpretation of each and every term. So, why do I have to keep this here? Because in this way, you can clearly understand why gain should be a function of several terms.

So please go through it. Now, $k$, if I select like this, then it is possible to show that I can maintain $e_y = 0$. Since this system is running inside a computer, you can also select a very large value of $K$, because I am not going to apply anything through the actuator. So, this is beauty. So, in observer, you can able to apply any kind of larger gain. Now, how do we prove this? I am going to take the Lyapunov function.

So simple a Lyapunov function. Here I am maintaining a positive definite matrix as an entity. I will calculate the $\dot{V}$. One more important thing I want to tell you is that in sliding mode control, you might have seen that people are proving that it is $\leq -\eta \mid s \mid$, and somehow this is directly implying that $\dot{V} \leq \alpha V^{\beta}$, where $\beta$lies between 0and 1. So, if you set $\beta = \frac{1}{2}$, then you will exactly get this kind of expression by selecting $v = x^T x$. So, exactly in the same way, I have selected this in place of $s$, since now I have to make sure that $e_y = 0$in finite time, for some time equal to $t$, and due to that reason, the sliding surface I have selected is exactly as $e_y$.

After taking the derivative of this and substituting this whole dynamics from here to here and then I will get this term. Now, what I am going to do is, again, give the interpretation of each and every term, how you can manipulate it, because finally what I have to do is show that this is $\sigma e_y^T e_y$, which should be less than some kind of constant $\eta e_y$; that kind of thing I have to show. So, how do you show that? For that, I have to understand each and every term, and then I have to manipulate it. And due to that reason, I have retained the interpretation of each and every term that involves here. And what is the role of this term? This is very, very important because this term I have to dominate over all terms so that it is compensated.

Now, I also know that this function is initially one I have assumed is bound by some known function $\rho(y, u, t)$. So, this function is not exactly known, but I am assuming that it is always bounded. Under this particular assumption, it is possible to show that $\dot{V}$, and obviously this assumption is also required; then I can easily further simplify this by using the notion of Lipschitz constant. After designing something new like this and substituting it, I will get exactly this, and finally, you can see that I will get exactly the same expression.

In this way, I can force the error to equal 0 in finite time. Now what I am going to do is apply this kind of methodology to some practical system. And here our goal is a little bit different. So, I am going to design a sliding mode observer, but the purpose of the observer is to detect any kind of fault that may enter through the input. So, I am able to estimate that fault and what kind of fault is basically entering that I want to quantify. So, how do we do that? You can see that this is actually a mathematical model of a single link flexible joint.

So, how can one make a flexible joint? You can put in one spring. So, this is the non-linear equation of the spring here, and here at every joint I am going to put this, and due to that reason, the overall system becomes highly non-linear in nature. Now, all other constants are defined such that $\theta_1$ and $\omega_1$ represent the motor position and velocity. Similarly, everything is defined in this class of somehow fully actuated systems. Now, here I have used this kind of matrix for a simulation study.

And the operating region I have to fix because this is a non-linear system. So, I always have to fix the operating region. So, I fixed the operating region. And after that, I am only assuming that whatever fault is going to enter the system will enter through the control channel. So, that is the kind of assumption I have. So, one can be able to represent this whole system in this particular form because I have, when I designed the observer, started with this form.

So, I am just trying to show you why this form and that form are matching, and here this $f(t)$ is nothing but fault; the fault is through the input channel, that kind of assumption I have. I have applied this kind of transformation, and after that, the $A_{11}$, $A_{12}$matrix comes like this; the $T$matrix will come like this. And after that, the $C_2$ matrix will come like this, and finally, you can also see that the non-linear term has been transformed. So, in order to design, observer transformation is required, and I have selected $\alpha$, $L$, $\phi$, everything based on the LMI, and then I have designed this gain $K$ because of $K$. So, actually, if you see this lecture, this lecture is concept-wise very simple, but calculation-wise you have to be very, very careful because there are a lot of transformations, a lot of gains, and LMI you have to solve.

So, every time you have to be very, very conscious. Now, once we are in sliding mode, then $e_y = 0$ and its rate of change is also equal to 0. I am assuming that this is actually a way in which one can realize the solution of a differential equation with a discontinuous right-hand side. I will discuss what the concept of equivalent control is in the subsequent lecture; I have discussed it several times. That philosophy of Utkin is that once you maintain sliding, the average value of that should also equal 0.

So, it means that $e_y$is perfectly 0, but the average of this is also equal to 0. So, if you calculate the average, then some kind of control comes into the picture, and regarding how to calculate the average, you can see this is essentially error dynamics. So, you have to set $\dot{e}_y = 0$, and at that time, in that particular way, you will be able to understand what kind of control is basically acting on the system during the sliding. Practically, to calculate this $v_{\text{eq}}$, you can pass through the low-pass filter; easily, you can see what kind of $v_{\text{eq}}$is actually acting here. Now, since I assume that I am already able to estimate $z$and $\hat{z}$, this term will disappear for that reason. And now, since I know $D_2$, which is from the system matrix, I can easily know what kind of fault has basically entered the system.

So, in this particular way, this is nothing but the pseudo inverse, because it might be possible that $D_2$ may or may not be a square matrix, and due to that reason, I have to calculate some kind of left inverse that is also called the pseudo inverse. So, I know this: I know the $v_{\mathrm{eq}}$from passing through the low-pass filter. So, I can estimate the fault. So, in this way, I can utilize sliding mode control for the fault, and this is a very excellent application of sliding mode control. You can see that I have actually injected this kind of fault, and after that, we have reconstructed this kind of fault.

Actually, the MATLAB code of this contains 4 to 5 slides. So, in order to reduce the number of slides, I have not provided the MATLAB code here, but during the tutorial, I am going to provide you with the MATLAB code. So, this kind of fault I am assuming is inside the system. So, one can be able to reconstruct that kind of fault, and these are the error dynamics of the error. You can see that everything is converging toward 0.

So, now it is time to conclude this lecture. So, what have we done? We define the robust sliding mode observer for the nonlinear system. Obviously, a lot of transformation, at least two transformations, are involved, and after that, the LMI condition also comes into the picture to show the stability during the sliding and converging to the sliding surface. So, both should be stable. So, for that, I have taken the help of the LMI plus Lyapunov stability.

LMI conditions reflect due to the Lyapunov stability criteria. I have validated this system on a flexible joint robot, and I have also performed fault detection; we have also demonstrated robustness through the disturbance rejection property of the observer, because you can see that everything will converge to 0in the presence of disturbance as well. So, with this remark, I am going to end this lecture. Thank you very much.