Sliding Mode Control and Applications

Dr. Shyam Kamal

Department of Electrical Engineering

IIT(BHU) Varanasi

Week - 12

Lecture-59

So, welcome back. In the previous lectures, I talked about the arbitrary time stability. In this lecture, I am going to talk about a new phenomenon that is actually somewhat dependent on the delay. So, you are all already well aware that during implementation, whenever a delay comes into the picture, it is a general perception that delay is harmful. But it is possible to show that sometimes, if you introduce some kind of artificial delay with some kind of guarantee, then your overall system is actually stable, and that is called artificial delayed output feedback, and a lot of work can be seen in this direction that has been done by Emilia Fridman. So, Emilia Friedman is a sister of Professor Leonid Fridman, and Professor Leonid Fridman is working on relay while she is working on delay.

So, you have already seen a lot of contributions in sliding mode control that are actually given by Professor Leonid Fridman. So, now in this lecture, what I am going to do is introduce some kind of artificial delay inside the twisting algorithm such that, without the information of the rate of change of x_1 , I can stabilize x_1 and x_2 at least asymptotically. And at the same time, I am going to show you that using that particular methodology, one can generate continuous control for a first-order system. And parallelly one can able to reject some kind of matched and unmatched uncertainty whenever we have second order system.

Obviously, at that time control become discontinuous. So, the main intention of this particular lecture is to actually talk about output feedback based on artificially delayed output. Okay, so what we are basically going to do inside the control is introduce some kind of delay. So, I am going to take exactly the same structure as twisting, and after modifying twisting, we are going to show that it is possible to achieve robustness and, using output only, you can control the plant. So, the main objective of designing output feedback is that output feedback is more reliable whenever we are talking about control design because fewer sensors are required.

And whenever we are making some kind of dynamic output-based feedback, then obviously the order of the system is increased. So, what is our main intention? To explore the output feedback twisting algorithm for the uncertain system with relative degree 2. You can also apply the same algorithm for relative degree 1. And our approach relies on the output and artificially delayed output. And obviously, we will achieve practical asymptotic stability.

Most of the time, I have already told you that in the presence of the disturbance, we will get some kind of accuracy with respect to our desired trajectory or desired point. And due to that reason, in practical implementation, even if you get practical asymptotic convergence, it is good for the practical implementation. And obviously, our main aim is to reduce the full state measurement. And obviously, whenever you are going to introduce some kind of delay, you have to prove stability. And stability proof is not easier because your finite dimensional system is somehow converted into an infinite dimensional system due to the delay.

So, we are also going to show that there exists some kind of continuous, weighted homogeneous, strict Lyapunov function, and this Lyapunov function is actually inspired by the work of Professor Morino. One more important point I am going to highlight is here. You have already seen the twisting algorithm, which is actually their algorithmic structure that looks like x 1 dot equals x 2 and x 2 dot equals k 1 signum x 1 and minus k 2 signum x 2 . So, this is the structure, and obviously, here I have some kind of bounded disturbance, and what are the guidelines I have to make k 1 greater than k 2 plus the value of the disturbance d 0 . So, I am assuming that this disturbance is actually bounded almost everywhere by d 0 .

So, in this particular combination, it is possible to show that x 1 and x 2 both equal 0 an infinite number of times. But what is a restriction? I have to apply the information from x1 and x2. If you see several animals, particularly bats, inside the forest, what are they doing? So, whenever they are actually moving inside the dense forest, they are somehow emitting some kind of sound, and they are always calculating something while moving easily. So, it means that they are always somehow emitting one sound, and whenever they receive a reflection, they are doing some kind of calculation. So, somehow that calculation is not based on the exact disturbance; some kind of approximate distance is being used to calculate this.

So, basically, the main motivation of this particular algorithm is inspired by that only. So, first I will come to that algorithm, then you are easily able to understand. That you can see here, suppose I have a first-order system, then I have actually proposed some kind of algorithm that is the integral of the sign of the output, and here is some kind of term you can take, and I will tell you how to design this term. So, that is the delayed version of this. Explicitly, if you see here, then things become clearer that I have information on x1, and after that, we have some kind of delayed function here.

And using their sign information, it is possible to show that you can control your position as well as your velocity, just like the bat. So, in order to actually understand the formation, stability and application of this algorithm, some background is required and due to that reason, I am going to first clarify the notion. So, R plus I am going to represent as a positive real number; the signum function I am defining almost everywhere like this:

$$\mathrm{sign}(x) := \begin{cases} 1 & x > 0 \\ [-1,1] & x = 0 \\ -1 & x < 0 \end{cases}$$

It means that I am talking about a flip-off solution. And obviously, here we are talking about Banach space.

So, this space is a complete space, and where Banach space of continuous functions means that we are assuming that if we are talking about the Cauchy sequence, then all Cauchy sequences in this space are also convergent. So, without knowing the limit, if you take any two sequences, then they are going to converge to some point, which is the limiting point. And in this way, we are defining ϕ, and this is nothing but the norm of ϕ. A homogeneous norm, I am assuming that whatever function I am going to construct here, in order to formulate the delayed twisting algorithm, will satisfy this kind of property. So, this is called a homogeneous norm.

Homogeneous Norms:

$$\|\phi\|_r = \left(\sum_{i=1}^{n} \|\phi_i\|^{\frac{\rho}{r_i}}\right)^{\frac{1}{\rho}}, \rho \geq \max_{1\leq i\leq n} r_i$$

where $r_i > 0$ for all $i = 1, \dots, n$.
And this is nothing but a ball:

$$B_\rho^\tau = \left\{\phi \in C_{[-\tau,0]} : \|\phi\|_r \leq \rho\right\}$$

So, it is possible to show that the trajectory is finally going to converge to this particular ball. So, now since this is an infinite-dimensional system, we first have to extend the concept of weighted homogeneity, and we are going to take the guidelines of the paper developed in 2017 by Professor Efimov's group, which actually extended the concept of weighted homogeneity for the delayed system. So, suppose that I have some kind of function that maps from Rn to R; then we know how to define the homogeneity. And here, this is nothing but dilation.

Weighted Homogeneity (I): - Dilation matrix: $\Lambda_r(\lambda) = \text{diag}\{\lambda^{r_i}\}_{i=1}^n$ ** Definition 1**: $f: \mathbb{R}^n \to \mathbb{R}$ is homogeneous of degree m if:

$$f(\Lambda_r(\lambda)x) = \lambda^m f(x)$$

** Definition 2**: $f: C_{[-\tau,0]} \to \mathbb{R}$ is r-homogeneous of degree m if:

$$f(\Lambda_r(\lambda)\phi) = \lambda^m f(\phi)$$

Weighted Homogeneity (II): - **Definition 3**: $f: C_{[-\tau,0]} \to \mathbb{R}^n$ is r homogeneous of degree m if:

$$f(\Lambda_r(\lambda)\phi) = \lambda^m \Lambda_r(\lambda) f(\phi)$$

We have already seen this during the development of higher order sliding mode control. Now, here in place of Rn, we have some kind of space that is generated by a continuous function, and at that time we are going to define weighted homogeneity or r-homogeneous degree m like this. And similarly, weighted homogeneity occurs whenever we are going to assign different weights to the vector field that is given like this. So, these are the concepts we are going to utilize during the development of the algorithm. Stability is also very important.

So, we are going to take the help of some kind of theorem developed by Professor S. P. Bhatt in 2005 in collaboration with Professor Bernstein, and it is possible to show that this function V2x is bounded by this homogeneous function, and after that, f is locally bounded and r homogeneous, and that is given like this; and after that, they will satisfy this kind of property. So, this is not simply a Lipschitz property; it is actually a little bit different from the Lipschitz property. Now, I have already told you that if you have a first-order system.

System Description: Consider the perturbed system:

$$\dot{x}_1 = u + d(t), x_1 \in \mathbb{R}, d: \mathbb{R}_+ \rightarrow \mathbb{R}$$

where $|\dot{d}(t)| \leq d_0$ (bounded disturbance derivative)
Proposed Controller:

$$u := -k_1 \int_0^t \text{sign}(x_1(s))ds - k_2 \int_0^t \text{sign}\big(\alpha(x_1(s), x_1(s-\tau))\big)ds$$

Initial condition: $x(s) = \phi(s), s \in [-\tau, 0]$.

Closed-Loop System:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -k_1\text{sign}(x_1(t)) - k_2\text{sign}\big(\alpha(x_1(t); x_1(t-\tau))\big) + \dot{d}(t)$$

where $\alpha(\cdot)$ maintains weighted homogeneity.
In this lecture we have taken

$$\alpha(x_1(t), x_1(t-\tau)) \text{ as } |x_1|^{0.5}\text{sign}(x_1(t)) - |x_1(t-\tau)|^{0.5}\text{sign}(x_1(t-\tau)).$$

So, what is the main motivation of higher order sliding mode control? We have seen that when I introduced the sliding mode control, we actually have to minimize the chattering. So, suppose that I have some kind of surface like $\dot{x} = u + dt$. So, if you design $u = -k \times \text{sign}(x)$, and if k is greater than the maximum value of disturbance. We know that $s = 0$ an infinite number of times, but chattering comes into the picture and that is proportional to the gain of this. So, later Professor Levant suggested that you can increase the relative degree of the system, and after that, you can design $\dot{u}$; $\dot{u}$ is nothing but what looks like the super twisting algorithm, and its form is $k_1\text{sign}(x_1) - k_2\text{sign}(\dot{s})$.

And in this way, whenever you are going to apply u, you have to take the integral. So, here in order to implement this algorithm, you need s and $\dot{s}$. So, a differentiator is required. After that, one criticism comes into the picture. Since you already know $\dot{s}$ and u control, it means that you are somehow aware of the disturbance.

And if you know the disturbance, then any controller, any PI controller, or any PID controller is able to compensate for this. And due to that reason, the justification of this algorithm is not given to create continuous control. And here you can see what we have done. We are saying that okay, if I have just information about the s , I will create a delayed version of that information, meaning whatever first information I am going to save. And after that, I am going to utilize this information; obviously, this is not like some kind of information where some quantity and their delayed version are present, and a homogeneous combination lies here such that I can maintain overall homogeneity of the system.

Okay, and then if I apply it, I can create continuous control here, and at the same time, I can reject the disturbance. Okay, so properties like first sliding mode control come into the picture asymptotically, x1 equals 0 , but the control is continuous. And, if you write in the form of a chain of integrators, then due to the integral option, this is obviously represented by a second-order system. Now, one of the choices in this lecture is the choice I have taken, but you are free to select any kind of choice; you just have to maintain the homogeneity. And why is homogeneity required? Same as the classical, whenever I am talking about the classical twisting algorithm, then at that time $u1 = k1\text{sign}(s1) - k2\text{sign}(x2)$.

So, again, write in the form of the algorithm, and then let us check whatever I have suggested during the development of higher sliding mode control. What I told at that time was that x 1 can be scaled by λ^2, and x 2 can be scaled by λ^2, like this. So, λ is a kind of positive number. I have given weight 2 , and here I am going to give weight 1 . So, with respect to this weight, it is possible to show that here 2 minus and 1 due to dt come into the picture.

So, overall homogeneity is -1 . So, the same kind of things I am trying to preserve here. And, due to that reason, I have given weight half since the signum function is here. So, finally, this becomes weighted homogeneous of weight $\lambda = 0$. Now, whenever we are trying to prove this. So, proof of this is not so easy because here you can see that any generalized α can work.

Just guidelines are that α is maintained weighted homogeneity, and here I have some kind of functional, meaning infinite dimensional functions, which means I have a function space involved here. It means that I have an infinitedimensional space. So, whatever extended notion of homogeneity that we have to utilize here. So, now, in order to prove the stability, actually at that time, Professor Morino has proposed some kind of Lyapunov function. So, it is possible to show that if you take this kind of coordinate transformation, then type of Lyapunov function.

Coordinate Transformation: Introduce time-varying state transformation:

$$z_1(t) := \frac{x_1(t)}{L(t)}, z_2(t) := \frac{x_2(t)}{L(t)}, L(t) > 0 \, \forall t \geq 0$$

Transformed System Dynamics:

$$\dot{z}_1 = -\left(\frac{\dot{L}}{L}\right) z_1 + z_2$$

$$\dot{z}_2 = -\left(\frac{\dot{L}}{L}\right) z_2 - \frac{k_1}{L} \operatorname{sign}(z_1) - \frac{k_2}{L} \operatorname{sign}\big(\alpha(L(t)z_1(t), L(t-\tau)z_1(t-\tau))\big) + \frac{\dot{d}}{L}$$

Obviously, the analysis is different, but it also works in this particular problem. So, z 1, I have done this kind of transformation, and here this is a time-varying transformation. You can see that, because L t depends on the time and for that reason, I have to maintain L inverse as well. L inverse here. So, L t should be positive, and L inverse should also exist and not be 0 at any time that I have to maintain.

Topological Equivalence Requirements: For stability preservation:

$$|L(t)| \leq p_1 \forall t \geq 0, |1/L(t)| \leq p_2 \forall t \geq 0, |\dot{L}(t)| \leq p_3 \forall t \geq 0$$

Example: Logistic function $L(t) = \frac{L_0}{2 - \exp(-t_0 t)}$.
So, in that particular language, in the coordinate of z , one can write the transform system like this, where I am assuming that it is necessary to preserve stability. So, this is somehow a topological equivalence. It means that initially, I have some coordinate frame z 1 . Now, I am talking about some kind of coordinate frame x 1 , and Lt is involved there. So, both are topologically equivalent, and 1 by Lt is also less than p2, and we are also assuming that their derivatives exist, which means L one can be selected.

So, one of the choices of L one can see that this will satisfy all conditions whatever I have written here, and it is possible to show now that if you select k_1 greater than or equal to k_2. So, if you see the classical super twisting algorithm, then its condition is k_1 and $k2$ and whatever maximum bound of the disturbance. So, using that condition, it is possible to show that x_1 and x_2 both equal 0 , infinite 10 . Obviously, whenever you introduce a delay, the gain condition is going to change, and that becomes a little more complex.

Complete gain conditions:

$$k_1 \geq k_2, (k_1 - k_2) \geq L\left(\pi_1 - \frac{2^{3/2}}{3}\pi_2\right), (k_1 + k_2) \leq L(t)\left(\frac{2^{3/2}}{3}\pi_2 + \pi_1\right)$$

but it is an era of computers. So, obviously, you can easily calculate the gain. So, if you select the gain like this, π_1 and π_2, that is some kind of variable I have defined. So, if you select like this, it is possible to show that the trajectories of z 1 and z 2 , as well as x 1 and x 2 , are going to actually become practically asymptotically stable with respect to this ball. And the size of this ball can be adjusted based on the size of the τ.

And this is the explicit gain condition. And now this algorithm is valid if rate of change of disturbance is actually bounded. So, there is no condition that disturbance should be bounded, but the rate of change should be bounded if you are going to apply it to a first-order system. Now, if you are going to apply it to the second order system, then we will see in the next few slides what kind of condition basically comes into the picture. Now, the proposed algorithm actually works as a controller for both first-order and second-order systems. So, how it works for the first-order system, we have already seen; if I have a first-order system, by integrating the sign of the output as well as the delayed sign of the output-here, "delayed" means in this particular lecture I am going to select like this, but you are free to select otherwise.

Controller Design: For second-order system: $\dot{x} = u + d(t)$. Controller:

$$u := -k_1 \text{sign}(x(t)) - k_2 \text{sign}(\alpha(x(t), x(t-\tau)))$$

Transformed System: State-space representation

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -k_1 \text{sign}(x_1(t)) - k_2 \text{sign}\big(\alpha(x_1(t), x_1(t-\tau))\big) + d(t)$$

where: $x_1 = x$ and $x_2 = \dot{x}$.
And if I apply this, it will work like the controller for a first-order system. It is also possible to show that as a controller one can be able to imply for the second order system. So, if I have a system in this form $\dot{x}_1 = x_2$ and $\dot{x}_2 = u + dt$, then I will design u; I can select it like this, and in this way, I can show that x 1 and x 2 are both practically asymptotically stable. Now, in this whole process, if you see the second-order system with control, then only the information of x(t) output and its delayed information are required. So, one can also be able to utilize this particular algorithm as an observer.

Obviously, you cannot apply the classical twisting algorithm as an observer for the second-order system. Another benefit comes into the picture; it is possible to show that one can utilize this kind of algorithm for the mitigation of unmatched uncertainty. So, if there is unmatched uncertainty, obviously, you cannot reject any kind of unmatched uncertainty, but if unmatched uncertainty satisfies some kind of condition, I will explain the condition; then it is possible to show that again x 1 and x 1 , and obviously, x 1 dot both equal to 0 asymptotically or practically, that is, 0 . So, at that time, since I have a disturbance here, x 1 is equal to 0 and $\dot{x}_1$ is equal to 0 , and due to that, x 2 is tending towards the disturbance. So, you can also able to calculate the value of disturbance or unmatched fault if that is enter inside the system.

Problem Formulation: Consider the system:

$$\dot{x}_1 = x_2 + d_1(t), \dot{x}_2 = u + d_2(t)$$

where: u : control input, $d_1(t)$: unmatched perturbation (Lipschitz continuous) and $d_2(t)$: matched perturbation (bounded).

State Transformation: Convert to matched disturbance form:

$$\dot{x}_1 = p(t), \dot{p}(t) = u + \dot{d}_1(t) + d_2(t)$$

where $p(t) := x_2(t) + d_1(t)$.

And obviously, only output is required, and it provides the observation property. So, I have taken a second-order system, and after that, I defined $x_1 = x$ and $x_2 = \dot{x}$. So, if you see this structure in the form of the algorithm. Whenever we talk about algorithms, we are somehow discussing some kind of autonomous system.

Obviously, this is a delayed system. So, this particular system has there is no explicit control because control I have already employ here. So, this is a closed loop system. So, their structure is exactly similar like classical twisting algorithm. But here the requirement is different, and due to that reason, we have coined a different name: the artificial delayed output twisting algorithm. So, this work is actually developed under my guidance and one of master student has actually proved this, the gain condition of this particular algorithm.

So, here control signal is discontinuous if you are going to apply for the second order system. Now, one beautiful property that it is possible to show is that if you have unmatched uncertainty, you can see that d1t is actually not a matched uncertainty that is going to enter through the control channel. So, it is possible to show that utilizing this kind of algorithm, you can able to compensate some classes of the unmatched uncertainty also. So, you can utilize the concept of sliding mode control to manage the unmatched uncertainty. Complete rejection of unmatched uncertainty is possible if uncertainty satisfies some kind of condition.

Sliding Surface Design: Define sliding surface:

$$S := x_2 + k_1 \int_0^t \operatorname{sign}(x_1(s))ds + k_2 \int_0^t \operatorname{sign}\big(\alpha(x_1(s), x_1(s-\tau))\big)ds$$

with dynamics:

$$\dot{S} = u + d_2 + k_1 \operatorname{sign}(x_1(t)) + k_2 \operatorname{sign}\big(\alpha(x_1(t), x_1(t-\tau))\big)$$

Controller Design & Stability: **Controller:**

$$u := -k_1 \operatorname{sign}(x_1(t)) - k_2 \operatorname{sign}\big(\alpha(x_1(t), x_1(t-\tau))\big) - K\operatorname{sign}(S)$$

For $K > |d_2|$, we get:

$$\dot{x}_1 = z$$
$$\dot{z} = -k_1 \operatorname{sign}(x_1(t)) - k_2 \operatorname{sign}\big(\alpha(x_1(t), x_1(t-\tau))\big) + \dot{d}_1(t)$$

Asymptotically stable by Theorem.
And what kind of condition is that? Here you can see that I have defined $x_2 + dt = p$, and after that, I am taking the derivative of pt. It means that if this disturbance d1 is differentiable and bounded, almost all time, then I can able to convert this system like this. And then I can be able to design some kind of sliding surface. Again, the sliding surface design is based on the artificially delayed twisting. Because we know that whenever I have $\dot{x}_1 = x_2$, at that time this is $\dot{x}_1 =$ this, so this is control for a first-order system we have already seen.

Okay, so in this way I can select the sliding surface. I will take the first derivative; control will explicitly appear, and then again I can design this kind of delayed output twisting here now. What am I going to do? First, I am going to cancel the known terms. So, this term and this term I am going to cancel, and S belongs to R just now. So, I am able to employ here that discontinuous control k sigma mass. And in this way, I can compensate for matched uncertainty as well as unmatched uncertainty.

And it is possible to show that in that particular way, x 1 and z are both equal to 0 in finite time. So, basically, if you look carefully, I have actually converted this like this, and in this process, x 1 is 0 and $\dot{x}_1$ both equal 0 . Now, you can also be able to utilize this algorithm for the design of the observer. So, how can we utilize this algorithm to design the observation problem? You can take the second order system again, and now you can design the observer. So, whenever you are designing an observer, you are just copying the dynamics.

So, I have copied the dynamics. It is not known. So, for that reason, I have removed it. Now, L2 is nothing but a kind of correction term. Now, I have to design this correction term. And I have to define the error dynamics.

So, I have defined the error dynamics. This is the estimated state and this is the actual state. So, in error dynamics, I have this kind of dynamics comes into picture. Now, I have to design L2. So, if I select L2 based on e1, similar to the delayed twisting by applying the theorem, I will explain the proof of the theorem just after this. It is possible to show that one can maintain both e1 and e2 equal to 0 asymptotically in the presence of this disturbance.

Observer Correction Term: **Choose:**

$$L_2 := -k_1 \operatorname{sign}(e_1(t)) - k_2 \operatorname{sign}\big(\alpha(e_1(t), e_1(t-\tau))\big)$$

- Makes error dynamics similar to delayed Twisting - Asymptotic convergence by Theorem

Van der Pol Example: - **System:**

$$\dot{x}_1 = \epsilon h(x_1) + x_2, \dot{x}_2 = -x_1 + F(x,t)$$

where $h(x_1) = x_1 - x_1^3/3, \|F(x,t)\| \leq \delta.**$ Observer:**

$$\hat{x}_1 := \epsilon h(x_1) + \hat{x}_2, \hat{x}_2 := -x_1 + C.$$

Error Dynamics & Controllers: - **Error dynamics:**

$$\dot{e}_1 = e_2, \dot{e}_2 = C - F(x,t)$$

- **Controller options: - Proposed:**

$$C := -k_1 \operatorname{sign}(e_1(t)) - k_2 \operatorname{sign}\big(\alpha(e_1(t), e_1(t-\tau))\big)$$

- **Hyper-exponential:**

$$C := -(k_3 + k_4)|e_1|^\beta \text{sign}(e_1(t)) + k_4|e_1(t-\tau)|^\beta \text{sign}(e_1(t-\tau))$$

- **Delayed static:**

$$C := -k_5 e_1 + k_6 e_1(t-\tau).$$

And obviously, x_1 converges to x_1 cap, and x_2 converges to x_2 cap. So, in this particular way, you are able to apply the delayed output twisting for three different purposes in order to generate continuous control for first order and to generate discontinuous control for second order. The third important problem is the issue of unmatched uncertainty, which you can reject, and the next problem that you can also apply as an observer design. So, applicability of algorithm is become very hard whenever you are going to apply the delayed output twisting. So, what have we done? We have taken the Van der Pol example, designed an observer, and after that, what have we done? We have done the compression. So, here I am telling the controller option that basically this term is responsible if I design C in such a way that e1 and e2 are both equal to 0 practically asymptotically.

Simulation Parameters: $F(x,t) = \sin(2t), \epsilon = 1 k_1 = 2, k_2 = 1, \tau_1 = 0.01 k_3 = 0.25, k_4 = 0.1, \beta = 0.8, \tau_2 = 0.3 k_5 = 0.35, k_6 = 0.1, \tau_3 = 0.3$ Initial conditions: $x_1(0) = 2, x_2(0) = 1$

So, I have to design this, and I have already stated how to select alpha in our case when I started. In literature, there exists some other variant, and it is possible to show that once $e_1 = 0$, and $e_1(t-\tau) = 0$, then control equals 0 . It means that it is not able to compensate for the uncertainty or disturbance. So, this is not suitable whenever an uncertain system comes into the picture, and you have to design the observer. In literature, artificially delayed static output feedback is also proposed, and it is possible to show that delayed output feedback performance becomes much worse if you consider some kind of uncertainty.

But our algorithm will work in any kind of uncertainty, provided that this uncertainty is bounded. So, those are the kinds of things we are actually going to show using the simulation. So, I have taken this set of gain conditions, this kind of disturbance I have taken, and after that, we have actually shown the observer's performance in all cases. So, this particular if you select see like this, then that is called hyper exponential stability. So, that exists in the literature, and this is called delayed static output feedback.

So, based on delayed information, there are several other algorithm is also existing in the literature. So, if you are interested, then please look into that. and this is the performance of all other algorithm. Now, proof. Proof of stability is a little bit crucial because obviously this is an infinite dimensional system.

So, first we have to show that whatever functional differential inclusion comes into the picture, how one can talk about their existence and uniqueness of the solution. So, again we are going to talk about set-valued mapping of the functional differential inclusion, just like the flip of classification. And we are assuming that if you remove the discontinuity set, So, basically, the union of 0 and that delayed output function that is equal to 0 , and I am going to actually collect all discontinuous sets. So, three kinds of combinations basically come into the picture, and it is possible to show that if there is no delay term.

Functional Differential Inclusion: Rewrite system as

$$\dot{x}_{t,1}(0) = x_{t,2}(0)$$
$$\dot{x}_{t,2}(0) \in -k_1 \text{sign}\big(x_{t,1}(0)\big) - [k_2 - d_0, k_2 + d_0] \text{sign}\left(\alpha\big(x_{t,1}(0), x_{t,1}(-\tau)\big)\right)$$

where: $x_t(s) = x(t+s), -\tau \leq s \leq 0$ and state space: $C_{[-\tau,0]}$. So, suppose that I will talk about the classical twisting algorithm. So, at that time, obviously, a solution will exist in the same way as the flip-off solution, but here again, since the trajectory is not going to stay anywhere in this particular combination, they are finally going to leave. So, the same kind of solution will appear, and we also have a similar property of homogeneous degree, because I have constructed the alpha function in such a way that the degree of homogeneity will be preserved. So, solution exist and unique. So, this whole work is reported in this particular paper on the artificial relayed output twisting algorithm that is published in IEEE Transactions on Circuits and Systems. The same kind of work in a geometrical framework, you will be able to see that is published in 2024 by the Abhishek Kumar Behera group.

So, what have we done? We have taken this Lyapunov function. This Lyapunov function is inspired by Professor Morino's work. And obviously, whenever you are using a Lyapunov function, you have to show that the Lyapunov function is always positive definite. So, how do you show? So, for that, it is possible to select that if you select π_1 and π_2 in a specific way that is given like this, then this Lyapunov function will always be positive. After that second step, you have to take the derivative. So, this kind of step you have to do by yourself, because a lot of calculation is involved in this particular process.

So, just for guidelines, I have actually given this slide, but you have to actually do the derivative; a lot of terms will be involved, and you can actually deploy the concept of homogeneity and finally show under which condition I can maintain $\dot{V}$ less than or equal to 0 . That is the way of proving. And if you do that, then this kind of gain condition comes into the picture, and the final bond you can see looks like this. Finally, after that, you have to select the delayed ones.

So, the selection of delay is again based on the Lyapunov criteria. So, this is the functional differential inclusion that comes into the picture because whenever the trajectory is in the vicinity of the delayed output, it is possible to show that it will satisfy these two norm inequalities. So, if you apply the Lyapunov-Razumikhin approach, then it is possible to show
that this inequality is satisfied, and in that particular way, it is possible to show that there exists some kind of minimum value of τ_0. So, I have given the parameters, and based on those parameters, we have also collected the minimum value of the delay.

So, even if you will give delay up to 1.456 second, then our algorithm will work. So, similarly for a set of other parameters, you have to select a delay based on that, meaning how much delay is allowed such that asymptotic stability will be preserved. So, now it is time to conclude this lecture. So, what have we done? We have proposed a novel twisting-like controller as well as the observer. And what is our main intention? Our main intention is to propose some kind of algorithm that will work on the current output information and its delayed version, and for that reason, we have given it the name "delayed output twisting" because the structure looks exactly like the twisting algorithm.

After that, we have also shown how using a Lyapunov function one can prove stability. And the allowed delay we have also calculated based on the Lyapunov-Razumikhin condition. And after that, we have also demonstrated the efficacy with respect to all existing kinds of delayed output feedback that are present in the literature. And what is the advantage? This only requires the partial information and odd performance over the existing delayed feedback method because this is also capable of rejecting the uncertainty, and this will handle relative degree 1 systems as well as relative degree 2 systems. Another beautiful means contribution of this particular algorithm, this is also able to completely reject some kind of unmatched uncertainty. So, you can also able to apply this kind of algorithm for the fault identification problem unmatched fault whenever that will encounter inside the system. So, with this remark, I am going to end this lecture. Thank you very much.