Arbitrary Order Robust Exact Differentiation

Welcome back. In the previous class, I talked about the differentiator design based on higher-order sliding mode control. And what we have seen is that we have proposed a differentiator in such a way that its structure is finally converted like a super twisting algorithm. And once I have a super twisting algorithm, then by designing two gains, it is possible to show that I can get the signal as well as its derivative in finite time. Another important contribution we have seen in the previous class is that in the presence of bonded Lebesgue measurable noise, the accuracy of the differentiator based on super twisting is extremely high. So, one of the great contributions or one of the best contributions of the higher order sliding mode control is the differentiator design.

In this class, I am going to generalize the calculation of the differentiator up to rth order. So, how do I construct a differentiator such that I can obtain the derivative up to the rth order using higher order sliding mode control, and that differentiator should be robust and exact? So, for the purpose of discussion, obviously, I am going to talk about the arbitrary order robust exact differentiation process. And obviously, why are we moving towards the higher sliding mode-based differentiator? Because traditional differentiators or numerical differentiators are highly sensitive with respect to high-frequency noise. And most of the higher-order sliding mode control, actually finite-time convergence gives us a very beautiful property because if we are able to show that in the absence of noise, if our system is finite-time stable, it is possible to show that if the controller is homogeneous, then in the presence of noise also, I can be able to achieve better asymptotic stability.

And due to that reason, we are trying to design finite-time convergence control in the absence of disturbance, and obviously, that leads to asymptotic stability in the practical situation. That is the main goal. So, in order to achieve that finite-time convergence, I need the exact information of σ, $\dot{\sigma}$ up to $\sigma^{(r-1)}$, and the $(r-1)$-th derivative $\sigma^{(r-1)}$we have seen, so that I can create a homogeneous controller. So, now what kind of uncertainty and disturbance is allowed in the previous class, particularly if I have information about σ? So, in order to generate $\dot{\sigma}$, we have seen that even if σcontains some kind of noise, suppose that noise is $\eta(t)$, if this is bounded and Lebesgue measurable, I can be able to get very high accuracy in the presence of this particular noise. So, the same kind of accuracy we are planning to maintain if I go for the higher-order derivative up to $\sigma^{(r-1)}(t)$or even higher.

So, what is the solution? Again, the solution is based on homogeneous differentiation and differentiator structure for arbitrary order, and here we are going to talk about two different classes of the differentiator. One is recursive, and the other is non-recursive. It is possible to show that the non-recursive best differentiator is just a consequence of the high-gain differentiator, where we are going to give different scaling to the correction term. And obviously, in the absence of the noise, our main goal will be to get the exact differentiation. That is our main goal.

And obviously, when I get the differential process done exactly in the absence of the noise, in the presence of the noise, we will always get very high accuracy. So, let us try to understand one of the needs of the differentiator. So, I have some kind of system, and what we are basically assuming is that I have a single input, single output system. Obviously, one can also extend this for multi-input, multi-output systems. And here, suppose that I only know the measurement of output and that I do not know the exact model of this system; I only know the order of the system.

So, for that reason, this is called a black box system, also referred to as a black box system. So, how can we stabilize this black box system in finite time? With just the help of the output information. So, for that, I need an $r-1$th order differentiator. If I am assuming that the relative degree of this particular problem is r, what is the meaning of relative degree? You can take σas an output and try to do successive derivatives. And if control explicitly appears at $\sigma^{(r)}$, then that is called the rth order derivative.

If you design control based on discontinuous control action and finally Filippov regularization comes into the picture, it is also called higher order sliding mode control. So, in order to implement or realize higher-order sliding mode control, an arbitrary order differentiation process is basically one of the requirements. and this will preserve several properties, exactness means in presence of uncertainty, also performance of the overall system that is exactly same like the disturbance free system. And obviously, I can maintain finite time stability and asymptotic properties; this means that in the presence of uncertainty, it is possible to show that I can always converge in the vicinity of the equilibrium point. In finite time, I am going to remain there.

And I can always decrease the size of the ball by designing the proper gain. So, what is the key benefit? I can able to develop robust exact differentiator, enable output feedback control. So, that is nothing but a kind of dynamic output feedback control for the black box system. And most of the system actually falls in this particular class. Again, I am going to assume that whatever signal $f(t)$ is going to contain some kind of bounded and Lebesgue measurable noise.

In the previous class, we have already understood the meaning of bounded. Lebesgue measurable means any noise that is integrable in any finite interval. In this way, I can maintain consistency of existence and uniqueness. And also, we are able to maintain the stability of observer- or differentiator-based control. So, now we are talking about some kind of Lipschitz constant such that we are assuming f_{0k}.

So, the $(k-1)$th derivative, suppose I want to calculate it, then that is bounded by L.
If I want to calculate the kth derivative, then obviously I have to increase the assumption. The assumption is the same as the super twisting algorithm, with one more condition: that this is bounded by $k+1$derivatives, which are bounded by some kind of constant L, and this estimation should be exact in the absence of noise.

Obviously, in the presence of noise, we will get some kind of accuracy with respect to some bound of the noise; we are going to look into this particular problem again.
So, this is called a recursive differentiator.
Why a recursive differentiator? You can easily understand.

So, let us try to see this algorithm from the lower two blocks from the back. If you look carefully here, $\dot{z}_{k-1}$, that structure is exactly like the super twisting structure. What we have seen is that if I substitute this signal v_{k-2}, then I will get the derivative of $v_{k-2}$with respect to z_k. That we have already seen. If I have to just take the first derivative, what you can do is substitute $k = 2$.

If you substitute $k = 2$, then you can easily understand that I will calculate the **first derivative**, that is
$z_1 = \dot{z}_0$.
Now, suppose you have to calculate the **second derivative**; in this way, I have to increase the structure such that
$z_2 = \dot{z}_1 = \ddot{z}_0$.
And finally, your original signal will come here, you can see, that is
$z_0 = f(t)$.
So, if you see each structure, obviously each one looks like the **super-twisting structure**, that is
$\dot{z}_i = -\lambda_i \mid z_i - v_{i-1} \mid^{1/2} \text{sign}\,(z_i - v_{i-1}) + v_i,$
with
$\dot{v}_i = -\lambda_i^0 \text{sign}\,(z_i - v_{i-1}).$
And here is how to maintain this.

You can see that the power of this is actually different. So, in order to maintain homogeneity, I have already told you that I need two properties, namely contractivity as well as homogeneity. So, somehow with contractivity and homogeneity, I am able to guarantee finite-time stability. And how do you maintain that? So, the last term should be weighted homogeneous with degree 0. So, I have to scale $z_k$and $v_k$in such a way that whatever things appear here have relative degree 0, and in this way $z_k$and d_k, where $d_k = dt$, have homogeneity degree 1, and in this way it means time is homogeneous with respect to degree 1; and finally, I am able to create some kind of vector field or some kind of differential inclusion, which has an overall degree of homogeneity equal to -1, that is our main goal.

So, in order to do that, what do we have to do? We have to adjust the weight systematically. $L$is again some kind of design parameter, and that design parameter we have already seen is nothing but the Lipschitz constant of the signal. The higher derivative of the Lipschitz constant of that particular signal that I am going to substitute here. So, we are actually going to see the weight tuning whenever we calculate the second derivative, third derivative, and fourth derivative once I take the example. Now, another thing is why this is recursive? You can see that once I get z_0, which means $f(t)$, and z_0 is here.

So, in this process, you can see that the second term, which is $z_1 = v_1(t)$, z_1, is now going to be replaced here. I am going to slowly increase, and finally, I will come here, and due to that reason, this structure is recursive. It is possible to show that another structure is non-recursive, where I will use the correction term that is nothing but $z_0 - f(t)$, which is also possible. Now, how do we show the convergence property? We have maintained the weight, recursive weight, in such a way that the overall degree of homogeneity of this differential equation is -1. Now, it is possible to show that I can substitute in the noise-free case; if there is no noise, then this signal $f(t)$ is exactly equal to z_0, which is equal to f_0, and z_i that I am going to construct by the higher derivative of this.

So, what do we basically have to do? We now have to tune the parameters λ_0, λ_1, and $\lambda_k$that are associated here, as you can see. So, that parameter we have to design such that I can prove the finite time convergence of this particular algorithm in the absence of noise, because if we are able to prove the convergence in the absence of noise, obviously in the presence of noise with the help of the accuracy of homogeneous differential inclusion, I can proceed. So, the main theorem: For any $\lambda > 0$, please come here again; you can see here what the meaning of $\lambda_0$is; this is the gain. So, somehow finally f_0, $\dot{f}_0$, $\ddot{f}_0$, ..., $f_0^{(k+1)}$will appear. So, $f_0^{(k+1)}$is going to appear here.

So now, if I am going to actually select some kind of gain that is $\lambda_0 \geq 1$, it is possible to show that there exists an infinite positive sequence, and that sequence is nothing but the sequence of $\lambda_1, \lambda_2, \lambda_k$, up to λ_k, if I want to do k-th order differentiation. And what am I going to do? I am going to normalize the error. And what is the meaning of an error? So, you can see that $z_0 \to f_0$; then $z_0 \to f_0$. Similarly, $z_0 \to f_0$ in the absence of uncertainty. If our algorithm will work, so, for that what I have to do, I am going to normalize the error.

So, the first error is defined by z0 minus f0(t) by L, and it is possible to show that by the construction of this weight, we have maintained a degree of homogeneity minus 1, and obviously, finite time convergence comes into the picture due to the contractivity. So, in this way, I can easily able to prove the convergence of this algorithm. So, for our purposes, what do we have to do? We have to set the weight k1 in such a way that the overall algorithm has a degree of homogeneity equal to minus 1. And after that, we are trying to apply the dilation property or detectability property of the set, and finally, that will imply the contractivity. It means that in a finite time, I am going to converge towards the smaller set.

So, after normalizing the error, it is possible to show that I have some kind of recursive structure such that the last one, if you see carefully, looks like the super twisting algorithm. And it is not difficult to show that once this is a super twisting algorithm, if our gain is properly designed, then what happens? eta k minus 1 is equal to zero. eta k minus 1 dot is also equal to 0, and eta k is nothing but the derivative of the kth signal. Similarly, if this is equal to 0, then sequentially it is possible to show that at the same time everywhere eta 0 is nothing but zi equal to f0; after that, eta1 is going to converge to eta 0 dot, and in this way, the convergence of the whole algorithm basically comes into the picture. Now, most of the time, the order of the system is 2, 3, 4, up to 5, with a maximum of 5, or if it goes beyond that, then you can calculate the weight.

So, one of the sets of gains it is possible to show; this gain is calculated by Professor Levant, and they have also provided an alternative option. If you apply this kind of gain, then you can calculate up to the fifth derivative. So, suppose that if you have to calculate the first derivative, you can just tune these two gains. Similarly, you can able to proceed for any any higher order. Another structure that is called a non-recursive structure can be seen everywhere; I am going to utilize the same information z0 minus f(t).

So, this is exactly motivated by the Luenberger observer, where both a and b equal 0, or you can see the high-gain observer. Why a high-gain observer? Because whatever L, L is the gain matrix of the simple differentiator, it should be very, very large. So, now that you have z0 minus f(t) everywhere. So, now this scaling factor is going to change because now z0 and f(t) must maintain homogeneity in some specific way. Here in recursive everywhere, initially eta, then eta1, something like that, that is going to change; you can just see it in the actual coordinate frame.

So, here z0, z1, and zk minus 1, and due to that reason, you can see that the homogeneity weight is different, but once you are going to keep everywhere z0 minus f(t), then you have to set this gain in some specific way. So, please be careful whenever you define the non-recursive structure. So, we have actually applied both kinds of structure in several practical problems. And what is our conclusion? It is better to utilize a recursive structure. So, now we are trying to see the performance of this particular differentiator in the presence of the noise.

So, suppose that I have some kind of Lebesgue measurable noise; this noise is obviously integrable and has some kind of magnitude that is less than or equal to epsilon. Now, what can you do? If you calculate the ith derivative, then it is possible to show that the accuracy is given like this. So, how does accuracy basically come into the picture? So, that is basically due to the relative degree. So, in the case of the second order, we have already seen that the first derivative, if I have z0 f0 here, then what happens is what kind of sequence will come here, which is 1 and k equal to 0. So, that is accuracy of this particular differential process for the first order derivative is given by big O epsilon.

First order, then the square root of the big O. Similarly, you will be able to proceed whenever you are going to calculate the k plus 1th derivative. Obviously, this is one of the best possible accuracies, whatever. So, in literature, if you see several differentiators actually existing, it is possible to show that sliding mode differentiator. That is based on higher-order sliding mode control, basically, and is recursive or non-recursive in nature.

So, their accuracy is the best possible so far, regardless of what differentiator is actually developed in the literature. And obviously, the fundamental limit for differentiation under noise is also discussed by Komlow-Groove. So, that is going to be satisfied by this particular algorithm. So, obviously, whatever differentiator is proposed by Professor Levant that is actually optimal, in the presence of the noise, we will lose finite time stability, but we are very close to the origin. Now, this is the recursive form and this is the non-recursive form.

For initialization, you can take z_0 equal to f_0. And in this way, you can reduce the time of convergence of the differentiator because this differentiator structure is running inside a computer. So, one more important thing is that you can also select a very high gain. So, any gain you can select whenever you are developing some kind of structure. So, now I am going to give you some sets of the gain that are proposed by Professor Levant.

So, obviously, once you get lambda k, you will easily be able to get the next lambda. So, here you can see that for k equal to 1, if you are just interested in the first derivative, then this set of gains is okay. If you want the second derivative, then you have to select this kind of gain. In this way, up to the fifth derivative, the explicit gain condition is given by Professor Levant.

And this combination is the best combination. So, he has calculated based on the methodology of homogeneity and some optimality. And after that, he has actually given this kind of gain condition. And this is nothing but a fifth-order differentiator. So, you can easily implement this kind of differentiator using MATLAB.

So, you can either use Simulink or write the code. So, this is homework for you; what can you do? You can take one sinusoidal signal; you can actually write the program for this particular differentiator, and after that, you can implement it on your computer. So, what have I done? Just for application purposes, I have taken one f(t); f(t) is a sinusoidal and cosine signal such that it is possible to show here that the Lipschitz constant for this is equal to 1. After that, I am now going to develop the fifth order derivative of this, and it is possible to show its robustness. So, obviously, the differential process is robust, and that is also exact, and the tuning of the higher order differentiator is very easy because explicit gain is actually provided by Professor Levant, and using that, I can implement any r.

or higher-order sliding mode homogeneous controller. So, a homogeneous structure is already provided by Professor Levant. Now, they have also give one of the way using that you can able to calculate the derivative. So, this is not difficult to actually utilize these kinds of concepts to design stabilization problems or output regularization problems for any other system. And obviously, the advantage is a differentiator; the differentiation process, as well as the control design process, is robust, exact, and recursive in nature. So, both the controller structure and the differentiator structure are actually recursive in nature.

Okay, so now I am going to solve some kind of output regularization problem, and output feedback is given like this. So, what is our assumption? Our assumption is just z0 information I have. So, we will construct some kind of differentiator that will work inside the computer, and after that, I will tune this gain L, which is the supremum of C phi Km. So, that depends on the parameters of the system. So, if you take some kind of system like x dot(t) equal to a(t, x) and then b(t, x) into u, what can you do? You can take z0 equal to x(t), and after that, you can calculate the higher-order derivative and suppose that at the rth derivative, control will explicitly appear; at that time, you can design C and Km, and based on that, you can just tune the gain of the last one, and after that, everything will work fine automatically.

So, what is this theorem basically telling? Suppose that if you have to design some kind of discontinuous control, this continuous control you can develop based on a pseudo structure that is a pseudo sliding mode structure or quasi sliding mode structure, and you can couple this with the differentiator design; after that, it is possible to show that the output and all its higher derivatives are equal to 0. The zero-order dynamics means that once you put the output and their higher-order derivatives into the picture, that is equal to 0. If that is stable, then overall system is also stable, even if we will do feedback linearization case. So now, a sketch of proof is very, very easy.

So, si is zi minus sigma i. We are defined like this. And after that, what am I going to do? I am going to form a differential inclusion, where the degree of homogeneity is minus 1. and weight I have adjusted such that I am able to maintain the contractivity property. Then overall, the structure means the system is based on the differentiator as well as the recursive feedback, like pseudo sliding mode control, pseudo structure-based sliding mode control, or quasi sliding mode control. So, overall system will give us the finite time convergence of the output variable. So, now the proof is very simple; xi can be defined like this, and after that you can take the rth order derivative such that now the control, discontinuous control, explicitly appears.

So, in order to implement this, I need the information of what kind of information is required x0, x1, xr-1. So, I can easily construct that kind of information from this particular differentiator structure. So, here I am going to solve a tracking problem. Suppose I have to track z1 equal to sigma i, and I am assuming that the higher derivative of the reference signal is given; then I can easily develop this kind of differentiator. Obviously, I have set the homogeneity weight such that the overall structure is homogeneous, and the degree of homogeneity is equal to minus 1.

We have maintained the contractivity property such that what happens is that the whole differential inclusion collapses to some kind of ball where all variables s0, s1, and s minus r, si, are nothing but the error between zi and sigma i, which is equal to 0. And obviously, if there is no disturbance, then we can show that the overall system is globally uniformly finite time stable, because I am assuming that I can scale gain everywhere in the space, because all controllers are homogeneous. So, with the help of homogeneity and contractivity, and obviously by designing the degree, one can ensure global finite-time convergence with just information about the output. Some kind of sampling interval. Suppose that whenever we are implementing continuous control, we are obviously implementing it in some kind of numerical way, and at that time, if the sampling interval is tau, it is possible to show the accuracy of the output.

You can see that it is tau to the power of r. So, if you take a 0.01 sampling interval and if r is third-order sliding mode control, you can see the accuracy. It means that practically sigma is equal to 0, sigma dot is equal to 0, and all higher order derivatives are also 0. If you have some kind of noise in the measurement, then at that time you will be able to see this kind of accuracy. So, both accuracy is actually due to the consequence of the homogeneity and we for the second order case in previous lecture, we have already seen how basically this relation comes into picture. Just you can scale the homogeneous differential equation, and after that, force the homogeneous differential equation to maintain its structure, and in this way, you can be able to explicitly get the bond of epsilon.

Now, what is a summary? The summary is that we are going to combine r-th order sliding mode control with r-th order differentiators, and after that, I am going to design some kind of finite-time convergence controller for a black box system. So, we are assuming that mathematical modeling is not accurately represented. Another beauty is that obviously everything we are implementing with the help of computers, and we have some kind of sensor that has noise. So, it is possible to show that in the case of the presence of noise as well as discrete sampling, I can maintain asymptotic accuracy. Most of the time, other control systems which are not homogeneous cannot maintain asymptotic accuracy.

At that time, we were somehow able to talk about the ultimate bond or uniform bound. This will give us the means to basically create some kind of dynamic output feedback-based controller if the model is not exactly known. So, if the model is known and you have output, then you can design some kind of observer. But if the model is a black box, then you have only one choice. What can you do? You try to get some kind of help from a robust exact differentiator.

So, here you can see that I have to design a homogeneous sliding mode controller. And suppose that phi contains some kind of noise which is actually bound by the measurable noise. So, here actually whatever gain of this controller should dominate that noise, and due to that reason, we are assuming that it might be possible for the noise to also be time-varying and state-dependent; still, I am able to design the control. And how do you actually adjust the convergence? It means that if I want faster convergence, I have to scale the gain. This is the scaling factor of the gain, where lambda is greater than 1, as you can see.

So, if you are going to scale the gain, obviously, the time of convergence is very, very fast. At that time, each coordinate system is scaled with some kind of lambda, lambda squared, and lambda r. So, somehow we have ellipsoidal kinds of things. So, I have a higher-order ellipsoid in nature.

And if you tune the gain, everything will collapse in finite time. And in this way, I can also be able to select the gain, quasi-controller gain. Quasi-controller, we have already seen that we have two different structures. One structure is the pseudo structure, a pseudo first-hour sliding mode type structure. Another one is a quasi-continuous structure. And obviously, we have discussed that a quasi-continuous structure is somehow more realistic if you want to reduce the chattering.

So, this is for the first order, this is for the second order. So, what is the differentiator going to do? They are going to calculate the value of sigma dot. Similarly, if you have a third-order system, then the differentiator is going to give you these two derivatives. You can measure this derivative from the output directly. So, what have we done? We have taken one practical model that is the car steering control problem. And suppose that you want to maintain your output on this particular trajectory, and I assume that this trajectory is sufficiently differentiable.

So, the relative degree of the system is 3, which means that I am assuming that g(x) is at least 3 times differentiable and bounded. And what is the meaning of relative degree 3? It means that I started with y, and then I calculated y dot. So, basically if you calculate y dot, then somehow theta dot comes into the picture, and after that v by l tan theta will come into the picture; after that, if you take the third derivative, then control explicitly comes into the picture. So, due to that design, the relative degree is 3; please check it, and after that, if the relative degree is 3, then I can easily design control like this. What is the meaning of this, and what is the meaning of z0? z0 is nothing but y minus g(x), which is exactly equal to the sigma, and what is z1? z1 is nothing but sigma dot.

In this way, I am able to proceed, and here you can see that if I contain just gain alpha equal to 1. So, I have designed quasi-homogeneous control. And this is the structure of the differentiator, and using this particular structure, it is possible to show that if I select gain, you can select any higher gain, and this algorithm is basically running inside the computer. So, in finite time sigma equal to z0, after somehow z0 that is equal to z1 and sigma dot equal to v0. In this way, if I measure z1, z2, and z3, I can easily calculate the value of the derivative.

And in this way, I can easily implement the control, and due to that controller, whatever I have written, I have written in the language of the differentiator. Then only the differentiator and controller will actually work together, and they are going to maintain sigma equal to 0, sigma dot equal to 0, and sigma double dot equal to 0. And once sigma equals 0, I can maintain y equal to g(x). Some kind of finite time t greater than or equal to T, and after that, I am going to maintain it. So, what have we done here? We have to track this particular trajectory, and we are assuming that we are going to implement the same kind of algorithm through a computer, where a step size is 10 to the power of minus 4 seconds.

So, it is quite close to the continuous time implementation, and after that, it is possible to show that the third-order sliding mode control algorithm comes into the picture again. Control is obviously discontinuous. And finally, the trajectory tracking and control input for the car is given like this, and then it is time to conclude this lecture. So, I have developed the concept of an arbitrary order differentiator. I have also discussed how to achieve finite-time convergence of the black box system if I only have output information.

How to develop control in the language such that the computer where the differentiator is running and our original plants will communicate, and after that, they will stabilize the control and obviously achieve accuracy in the presence of the noise that is given like this. Now, what is the implication? So, whenever you want high-precision output feedback control, particularly in robotics applications, when defense or medical applications come into the picture, it is possible to show that you can easily apply the higher sliding mode control based on just the output information. So, with this remark, I will end this lecture. Thank you very much.