Robust Finite Time Continuous Super Twisting Controller Design for MIMO Systems

So, welcome back. In the previous class, I was talking about the parameter estimation problem and what we have seen is that using a super-twisting-like algorithm, it is possible to show that one can estimate an unknown parameter, and once the unknown state is estimated, the unknown parameter can also be estimated using some kind of algorithm. We have developed the nature of the algorithm, its convergence results, and after that, we also discussed one of the applications. In this lecture, we are going to extend the super-twisting algorithm for the multi-input multi-output case. So, one of the very active areas of research is multi-agent systems. So far, I have not taken examples of multi-agent systems, but in this lecture, I am going to talk about multi-agent systems and how one can implement sliding mode control for them.

So, in this lecture, I am first going to develop the multi-input, multi-output variant of the super twisting algorithm, and I am also going to maintain the continuity of the control. Why? Because most of the time we have multiple robots, and that is nothing but some kind of electromechanical system, it is possible to show that continuous control is more suitable for those classes of systems. Obviously, we have several other ways to implement a multi-agent system. Suppose that we have multiple converters and I want to control some kind of current and voltage.

Similarly, if I have multiple energy generation sources and we want to control the voltage, we have several different problems that actually occur whenever we are talking about multi-agent systems. So, let us see how to formulate multi-agent problems and, after that, how to apply sliding mode control for that particular class of problems. So, the purpose of the discussion. One of the main goals is the formation control. What is the meaning of formation control? Suppose that if I have multiple agents, then it is possible that one of those agents might be a leader and all the others are followers.

So, most of the time we have to create some kind of geometrical structure or some kind of geometrical shape with respect to the leader. So, that class of problems is called formation control. Problem or formation control problem because in order to achieve formation, I am going to design the control. The meaning of multi-agent is that I have multiple physical or some kind of virtual systems. So, suppose that we are talking about computer programs; it might be possible that I have several subprograms running together to satisfy some kind of common objective.

So, due to that reason, multi-agent systems can be virtual as well as some kind of physical, like robots, energy systems, or several other applications. Here I am going to specify the dynamics of this multi-agent system by double integrator dynamics. It means that I am talking about whatever system I am going to consider that is actually defined using some kind of second-order chain of integrator form; this means

$$\dot{x}_1 = x_2 \text{ and } \dot{x}_2 = u.$$

Obviously, every system has some kind of disturbance. So, I am also going to assume that there exists some kind of disturbance, but I am assuming that the disturbance is going to enter through the control channel only.

But one can be able to generalize this concept for any order chain of integrators. So, depending on the problem, obviously, the complexity of the problem becomes very high because the overall system has now become a multi-input, multi-output system. After that, another class of problems is called the consensus problem. It means that somehow we are trying to converge to some kind of average of all states, or sometimes a specific goal is given, and all agents are going to converge to that goal, which can be represented in terms of position or velocity. So, both the problem I am going to discuss here and that class of problems we are giving a special name to, which is called the zero formation control problem, because finally, all agents are going to meet at some specific point.

So, now during the solution of this problem, I am going to ensure that whatever the formation control or consensus control, or whatever target I decide, I will finish within finite time; due to that reason, we need some kind of robust finite time control algorithm. Since I have some kind of time-varying disturbance, whatever feedback I define is known to be smooth in nature. And how can this be achieved? It is possible to show that by using classical sliding mode control. Since you have second-order dynamics and here multi-input, multi-output systems. So, what can you do? You can define the sliding surface, and after that, you can apply the philosophy of multi-input, multi-output classical sliding mode control based on unit vector control, and then you can also achieve the same objective, but you cannot guarantee finite time stability if you have a second-order integrator chain of dynamics.

But suppose that if every agent is represented by just a single integrator, then at that time using first-order sliding mode control you can guarantee. So, in this class, I am not going to discuss when I have first-order dynamics, and then you can apply simple sliding mode control, or second-order dynamics, and you can design sliding mode control. Our main objective is to apply higher-order sliding mode control theory to multi-input, multi-output classes of systems, and for that reason, I am going to first extend the super-twisting control in such a way that it is applicable to the multi-input, multi-output system. And obviously, finite time convergence is guaranteed, and one can also prevent the chattering because whatever control I am going to propose is continuous in nature, and obviously, robustness comes into the picture. So, whenever we talk about the class of problems that is called the multi-agent problem.

So, this is a fusion of two different classes of problems. So, mathematically, in order to represent this problem, I need the concepts of graph theory as well as control systems. So, what is the role of the control system? The control system is somehow ineffective whenever all agents are apart from each other. So, they are actually going to push all agents so that they will come in for some kind of specific objective. So, how can one basically decide that kind of objective of formation, which is given by graph theory? And once we achieve that formation, graph theory will maintain that kind of formation with some algebraic or geometrical concept.

I am going to discuss some parts of graph theory here. If you are not aware of graph theory, you can just go through any standard mathematics book; you will be able to understand easily. So, what I am basically going to do, whenever I talk about a graph, is that the graph contains two parts: one part is called the node and the other part is called the edge. So, suppose that I have three agents. Agent means this is one node, this is another node, this is the third node, and this is a connection.

So, this is called an edge. So, I have generalized this theory for n nodes and obviously, in this particular lecture, I am going to limit the number of agents because based on those agents, I have to formulate the Laplacian matrix and everything. What is our assumption? It might be possible; it is possible to show that all agents are not directly connected. And due to that reason, it might be possible that I do not assume I have an edge over every agent. It might be possible that this agent and this agent are not connected, but obviously, this agent is connected through this agent.

And due to that reason, this symbol comes into the picture. So, now one more important point: it might be possible that whatever communication occurs between the two agents is not bidirectional. Now, in order to maintain the formation, one of the requirements is there, and mathematically that requirement is called the requirement of a spanning tree. What is the meaning of a spanning tree? So, we are assuming that somehow all nodes are connected in some way. A mathematical definition is that a directed graph G has a directed spanning tree if and only if there exists at least one root node that has a direct path to all other nodes.

So, this kind of thing we are going to maintain in order to solve the formation control problem, and it is equivalent that globally one can reach any node. Now, there is another algebraic concept called the Laplacian matrix. So, during the graph-theoretical formulation, this matrix automatically comes into the picture due to the communication between the agent, and it is possible to show that this matrix L has an eigenvalue that is always greater than or equal to 0, and this is very, very important because, using the 0 eigenvalue, we are going to prove two things: consensus as well as formation. And obviously, all non-zero eigenvalues should lie in the open right half-plane, which we have to ensure, okay. And if there is exactly one zero eigenvalue, this implies G has a directed spanning tree.

And I have already assumed that in our case, we have a directed spanning tree, and for that reason, we are assuming that we have exactly one eigenvalue and that the rank of the Laplacian matrix is n, which means that L has a simple 0 eigenvalue. So, the proof of this particular theorem is given in this book. So please check. And after that, why are these two things required? So, we are talking about the spanning tree. So, physically, a spanning tree ensures the connectivity of the network and provides the feasibility of consensus and formation control.

And obviously, the Laplacian structure talks about the convergence and stability of collective behavior. So, our goal is to somehow come into a specific formation, and after that, we have to maintain that formation. So, using this Laplacian, it is possible to show that one can maintain the formation and how to achieve that formation; that is the role of the control system. So, in this particular lecture, I am going to assume that I have four followers. So, that is represented by 1, 2, 3, 4, and 0 is going to be represented by the dynamics of the leader.

And whatever topology I have considered that is a spanning tree, it is always possible to show that in some way every agent is connected to this particular leader, and due to that reason, one can show that this satisfies the property of a spanning tree. Now, I am assuming that whatever dynamics of the leader are given by this, and here the assumption is u_0 is not known to us. It means that the control one can actually apply to a leader is not known to us, but obviously, I am assuming

I am assuming that here, agent dynamics means containing some part of the uncertainty, and obviously, this uncertainty is bounded. In this particular formulation, it is possible to show that only this assumption is enough, but several times it is possible to show that only relative position information is available. At that time, it is possible to show one is able to design a super twisting kind of observer. Obviously, whatever super twisting observer we design, that super twisting observer can be inspired by multi-input, multi-output. case, and that is actually given by one of the active researchers, Edward.

So, if you go through the Edward paper, then you will be able to see how to generalize multi-input, multi-output super twisting. Obviously, that multi-input, multi-output super twisting here is not directly applicable if you have second-order dynamics because we know that super twisting is only applicable for relative degree one systems, and due to that reason, we should generalize this super twisting for somehow second-order or second-order kind of multi-input, multi-output systems. However, as an observer, you can apply whatever super twisting algorithm is proposed by Edward. So, if you are assuming that you have position information, then this assumption is also required. But if you know you have both relative position information, what is the meaning of relative information? So, here suppose that if I assume that 1 is going to communicate with 2, then I have some kind of relative position information between 1 and 2.

4 is directly connected with 0; it means that I have relative position information between the leader and the follower. Similarly, I have information about relative velocity. Now, what I am going to do, since I am going to ensure that after a finite time, all the followers are going to maintain some kind of specific formation. It means that some kind of geometrical structure or some kind of geometrical distance is going to be maintained with respect to the leader dynamics. So, this will represent whenever I will put 0, it means that that is the

position of leader.

So, I am going to maintain a specific distance, and this $\Delta$is some kind of matrix. So, I have to provide what kind of relative distance I want to maintain with respect to the leader. And after that, I assume that, velocity-wise, both are equal; one is nothing but some kind of unity matrix. So, you can assume this as a unity matrix, not an identity matrix. I am discussing the effect of follower dynamics, and I am assuming that u_0, which is nothing but the control to the leader, is also bounded, and its derivative is also bounded; I have to achieve this kind of convergence in finite time.

So, what is the key inside? So, full positional knowledge is not required; only relative knowledge is required. It means that there is no need to know the position of this, the position of this, or the position of this; just relative information is okay for the development of the algorithm. What is the sufficient condition for a directed path from the leader to each follower that is not required? Some kind of way a leader can connect to the follower is that this leader is not directly connected to 2, but if there exists some kind of path. So, the leader can first check the information with 4; 4 can communicate with 1, and 1 can communicate with 2. So, direct information of 0 and 2 is not available, but still I can able to achieve the formation.

So, the sufficient condition is a direct path from the leader to the follower. We need it, and if that is the case, that is called the spanning tree. So, that we have already assumed. One of the very important things that comes into the picture during the formulation of the problem, a multi-agent problem, is called the adjacency matrix. So, how do you define an adjacency matrix? So, if $a_{ij} = 1$, it means that the element of the adjacency matrix equals 1if agent j receives information from agent i.

And $b_i = a_{i0} = \begin{cases} 1, & \text{if agent } i \text{ receives direct information from the leader,} \\ 0, & \text{otherwise.} \end{cases}$

So, in this way, I can form some kind of adjacency matrix. In the next slide, I am going to show you how to form the adjacency matrix. Now, what am I going to do? Based on relevant information, I am going to define the relative position error. So, here $b_i$is nothing but the term where leader relative error means x_{1i} and here x_{10}.

So, with respect to the leader, I am now going to calculate the actual relative position between the other followers, and similarly, this will represent, if you see carefully, the distance between two followers because several agents we have seen in this figure are not directly connected to the leader. So, at that time, again error dynamics come into the picture because finally, what is our goal? Our goal is to push all agents towards the specific formation, and due to that reason, some exchange of information is required. $\Delta_{1i}$is nothing but a kind of offset formation, an offset term that we have to maintain; that is our goal, to maintain this much of the offset with respect to the leader. And after that, we can define the relative velocity. So, whenever we are defining relative velocity, we are obviously going to take the derivative of this, and automatically when we take the derivative, if this Δmeans the relative offset is constant, then that becomes 0, and for that reason, there is no offset term that is going to appear in the velocity.

Now, I have already told you that one of the very important things is the adjacency matrix. So, the adjacency matrix is actually defined by this particular algorithm. So, we have already seen that

$$a_{ij} = \begin{cases} 1, & \text{if agent } j \text{ receives information from agent } i, \\ 0, & \text{otherwise.} \end{cases}$$

So, based on that, we have basically created this kind of matrix. So, agent 0will get information from itself, and due to that reason, it becomes 0.

So, due to that reason, in this row, you can see 0everywhere. And after that, I now have to formulate this matrix D, because with the help of this matrix and the adjacency matrix, one can formulate the Laplacian matrix. The Laplacian matrix is very, very important to maintain stability as well as formation. So, how can one basically be able to substitute the diagonals of $d_0, d_1, \dots, d_n$? For that, what we have to do is see here that

$$d_i = \sum_{j=0}^{n} a_{ij} ,$$

where $i = 0,1, \dots, n$. So, if I substitute $i = 0$, then a_{0j}comes into the picture, and after that, I have to vary $j$from 0to n, and then I have to take the summation. So, in this way, I can calculate the $d_i$of each element, and after that, we can define the Laplacian matrix.

And after that, similarly, I can define the adjacency matrix just for the follower, that is, the subgraph. A subgraph is nothing but a subset of the graph. So, in this way, I can easily formulate the Laplacian. And

$$b_i = \begin{cases} 1, & \text{if agent } i \text{ is connected to the leader,} \\ 0, & \text{otherwise.} \end{cases}$$

In this way, basically, I am going to define the matrix $\bar{B}$, and it is possible to show that it is nothing but

$$\bar{B} = \text{diag}\,(a_{10}, a_{20}, \dots, a_{n0}).$$

So, one can construct the full Laplacian matrix based on this, because now I have information about everything. So, all the information I am going to keep inside this matrix, and then I can easily formulate the Laplacian matrix. Now, you can see here that I have defined the formation errors. So, now I am going to stack all the formation errors inside some vector that I have defined as Φ_1, and the velocity I am going to actually stack into Φ_2. So, what is our objective? I have to maintain

$$\Phi_1 = 0 \text{ and } \Phi_2 = 0.$$

So, once $\Phi_1 = 0$and $\Phi_2 = 0$, it is possible to show that all agents are going to maintain some kind of specific formation with respect to the leader that I will establish last. So, for our first goal, using the design of some kind of continuous control, I have to maintain

$$\Phi_1 = 0 \text{ and } \Phi_2 = 0.$$

So, in the next subsequent slides, I am going to first prove how $\Phi_1 = 0$and $\Phi_2 = 0$. This is not a very easy task. So, what am I going to do? First, I am going to represent everything in a multi-input, multi-output, second-order chain of integrators.

Why multi-input, multi-output? Because each agent has some kind of control input. Similarly, the leader also has some kind of control input. So, if you substitute everything, it is possible to show that this kind of structure automatically comes into the picture. Where L bar, which I have already defined, is nothing but the Laplacian of follower dynamics. So, you can easily check that this kind of dynamics comes into the picture.

Now, the Laplacian matrix for the follower graph I have taken here is just to ensure that the theory I am going to develop is valid. So, this kind of spanning tree I have taken, but you are free to select any kind of spanning tree. So, for that, I am going to actually define the adjacency matrix as well as the Dmatrix, and then I am going to define the Laplacian. So, those kinds of things basically I am going to do here, and in this way, numerically you will be able to check how one can formulate the Laplacian matrix and $\bar{B}$. Now, regarding the control objective, we have already seen that I have to design $U$and $u_i$in such a way that this will ensure infinite-time control, and I have to maintain continuity while assuming that whatever leader control exists is bounded.

So, basically, in a control structure, if you look closely, it is possible to show that whatever Laplacian matrix of a subgraph of the follower is invertible; it is not difficult to prove, by construction. So, basically, this is invertible. Now, I have two parts: U_0, which is the nominal part, and U_D, which is the disturbance rejection part. So, if you see any super twisting algorithm, first-order super twisting algorithm, if you see carefully.

So, if you take this kind of structure, just this kind of structure, it is possible to show that in the absence of any disturbance, $x_1 \to 0$as $t \to T$for some finite T. So, now suppose that I have a system like

$$\dot{x}_1 = u + d;$$

now how can I compensate for this disturbance d? Using this control, it is not possible to completely compensate for this disturbance because, at $x_1 = 0$, the control term equals 0, since I have proposed $u$like this. So, now, our aim is to add some extra terms. So, how do you add an extra term? So, due to that reason, in super twisting, what Professor Levant has done in the construction of super twisting is the following. Actually, they know that this controller is finite-time stable. So, based on the homogeneity, they have added one more term such that

$$\dot{x}_2 = -k_2 \text{sgn}\,(x_1).$$

So, this term is basically responsible for the compensation of these kinds of disturbances. So, the exact same principle I am going to use to generalize the super twisting algorithm. So, this super twisting algorithm contains two terms. The first term is exactly the same as you can see in this kind of structure. So, obviously, I have a second-order integrator, and for that reason, I cannot write $|\ \Phi_1\ |$as Φ_1.

So, I have to assign some specific weight. So, I have given a weight equal to $\frac{2}{3}$. It means that the homogeneity degree of $\Phi_1$is λ^3. And here, the homogeneity or degree of $\Phi_2$is basically nothing but I have to maintain λ^2, and in this way, I can make the whole controller homogeneous in nature. So, this is actually based on why I have selected this kind of weight, which is the philosophy of that. And after that discontinuous term, you can see that I have constructed it like this, and this $\dot{p}$I have defined as U_0, and what is U_0? U_0I have proposed like this. What this theorem is telling us is that if you have this kind of dynamics, I have coupled everything with the help of graph theory, and after that, we are going to show that if $\rho$is represented by this.

So, I am considering this whole term as a disturbance because $u_0$and $\rho$are unknown to me. So, it is possible to show that if you design this set of gains, then

$$\Phi_1 = 0 \text{ and } \Phi_2 = 0$$

in finite time. Proving this is not so easy. You have to develop a Lyapunov function, and it is possible to show that the development of some Lyapunov function $V$is possible.

and that is a function of some set W_0. I will define the meaning of this set, and it is possible to show that there exist two positive symmetric definite matrices, and here $\lambda_{\max}$and $\lambda_{\min}$are the eigenvalues of these positive symmetric definite matrices. Then, within this time, I can converge to the equilibrium point. So, what have I done? I have substituted the control, and after substituting the control, it is exactly the same here because I have uncertainty. So, I am going to shift the uncertainty on the higher-order discontinuous term, and finally, this is converted into a differential inclusion.

So, ξI have defined here, and $\dot{\xi}$ I have maintained like this. So, all uncertainty is going to transfer here. So, the differential inclusion formulation involves higher-order derivatives. So, using the Filippov regularization, I am able to talk about the existence and uniqueness of the solution. Now, if you see the Lyapunov function, you can see here that this Lyapunov function is continuous but not differentiable because I am going to use the modulus of S. So, the modulus $| S |$is not differentiable at $S = 0$, and for that reason, whenever I construct $\dot{V}$, it means that our analysis is not valid on this set.

Now, what we have to show, since our derivative analysis is not valid on this set, is that on this set the trajectory is never going to stay. So, if the trajectory is never going to stay, it means that this case automatically comes into the picture again. So, in this way, one can easily prove, whenever they are constructing some kind of Lyapunov function that is not differentiable. So, obviously, $V(\mathrm{W})$is not on this particular set, where $s = 0$and $\omega = 0$, which is also not the equilibrium point, because you can see carefully that here whatever dynamics I have considered.

So, $\dot{\xi}$is going to contain some kind of uncertainty. So, once $S = 0$, the equilibrium point is not equal to 0. So, since this is not an equilibrium point and due to that reason, it is possible to show that even if $S = 0$, the trajectory is not going to be maintained at $\zeta = 0$. So, now, if you take the derivative, this kind of job you have to do by yourself, just for clarity; for the clarity of calculation, I have written everything, but this kind of calculation you have to do by pen and paper; then only can you easily understand each and every step. And after that, I have to apply the Cauchy–Schwarz inequality. So, please do each and every step one by one, and after that, you can define one vector like this, and using this vector, you can now see that a nice matrix comes into the picture.

So, these two matrices come into the picture. So, now our goal is very simple because this term is always positive. So, I just have to show that both this matrix $P$and $\bar{P}$are positive symmetric definite. So, by doing that, it is possible to get the condition on the gain. And, due to that reason, if this gain condition is satisfied, then obviously, I can maintain

$$\Phi_1 = 0 \text{ and } \Phi_2 = 0$$

in finite time. If you see the gain condition carefully, then δ_2 only comes into the picture.

It means that the derivative $\dot{\rho}$ is the only one involved; the bound of ρ is not involved whenever you are going to decide the control action. And, in this way, what have I done finally? I have utilized the Rayleigh inequality, and using the Rayleigh inequality, I have finally converted the whole equation like this. Now, I have two systems. I have already told you that actually, whenever the initial condition is on $\bar{S}$, $\bar{S}$is not differentiable. So, I have to show that on this particular set, the trajectory is not going to diminish there.

And it is possible to show this by the construction of this differential inclusion. So, $\bar{S}$I have removed. So, in this way, I can talk about the existence and uniqueness of the solution. How do you define the field $f(S)$? $f(S)$ you can be able to define, where uncertainty is also there.

And due to that reason, I cannot maintain this particular set. And due to that reason, it is possible to show that, since this is not the equilibrium point, the trajectory is finally going to leave this set. So, only one possibility is when $\xi = 0$; then only we can be able to preserve the structure. Now, the solution property is obviously that when $\omega = 0$, then at some time $t = 0$, one is able to extend this solution. It is possible to show this using a kind of inequality called the Bihari inequality. So, using that particular inequality, one can show that whenever I have this kind of differential inequality, then it is finite-time stable. So, I am not going to state the Bihari inequality here, but if you are interested, then please go and explore it.

Now, one very important theorem plays a role here, since I have some kind of Lyapunov function that is continuous but not differentiable. So, now, if I just show that $\dot{V}$holds almost everywhere and $V(W)$is continuously decreasing, then it is possible to show that the largest invariant set is $(0, 0)$. So, the same kind of things one can establish using this particular inequality, and then one can calculate the finite time. So, those who have interest in multi-agent formulation or multi-input, multi-output super twisting can visit a paper that is published in *ISA Transactions*, and the title of the paper is *"Robust Continuous Finite-Time Controller for Multi-Input, Multi-Output Systems."* Now, once $\omega = 0$after a finite time, then I have only one choice: that the whole dynamics is represented by this, because this continuous part $\dot{\Phi} = U_0$, once $S = 0$.

So, now, the whole dynamics is governed by somehow this control action. So, again I have to prove that $\Phi_1 = 0$and $\Phi_2 = 0$. So, I have constructed this control such that homogeneity is maintained. If you check carefully, homogeneity is always maintained. So, if you construct Vlike this and after that, if you calculate, then it is possible to show that this is positive semidefinite. But the degree of homogeneity is actually negative, and this is positive semidefinite, with $\Phi = 0$.

Actually, $V = 0$, but $\Phi = 0$and $\Phi_1 \neq 0$; that is not going to be maintained, because if you substitute $\Phi = 0$, then finally, the trajectory will leave that set. And in this way, using LaSalle's invariance principle, it is possible to show that if $\Phi_2 = 0$, then I have only one possibility, which is $\Phi_1 = 0$. So, how are people basically proving $\Phi_2 = 0$from here, and after that Φ_1? Now, you can substitute $\Phi_2 = 0$, and by maintaining this, you can see that $\Phi_1 = 0$; that is the choice. So, in this way, you can apply LaSalle's invariance principle. And finally, using LaSalle's invariance principle, you are able to establish that $\Phi_1$and $\Phi_2$are both asymptotically stable.

And after that, by the property of homogeneity, you can see that the degree of homogeneity of the whole differential inclusion, a multi-input, multi-output differential inclusion, is -1.

And due to that reason, due to negative homogeneity and asymptotic stability, I can talk about finite-time stability. And due to that reason, $\Phi = 0$in finite time. So, that kind of conclusion. Once $\Phi_1 = 0$and $\Phi_2 = 0$, then it is possible to show that the error dynamics will converge like this, okay.

Now, once I have this dynamics, I have to prove that it somehow converges to x_{10}. So, how do I prove that? Since I know that whatever L matrix has a construction like this, one of the eigenvalues is exactly 1. So, the corresponding eigenvector is **1**, and after that I can easily incorporate that term here, and in this way I can show this. Similarly, I am able to show that velocity is going to converge to this point. You can easily do this, and whenever formation comes into the picture, $\bar{L} + \bar{B}$is invertible, which means this matrix is invertible.

So, you can easily prove the formation as well. Formation with respect to some kind of constant formation with respect to the leader you can achieve. And in this way, I am able to get complete formation, which means all position vectors are going to converge here, velocities will converge here, and I am able to maintain the directed spanning tree. What I have done during the simulation is that I have taken 4 agents, and after that, $x_1 = 0$ and $x_2 = 0$. Initially, I am assuming there is no control over the leader; this means that the leader is actually moving with whatever position they have and will maintain it. And after that, I am going to maintain a 0-distance formation, and for that formation, I have taken 4 agents, and due to that reason, all are in different positions.

I have taken this set of gains, and these are the initial conditions, and these are the set of disturbances I assumed, and after that, you can see that for a distance of 0, the formation occurs. So, all agents are going to actually, since the control to the leader is 0due to that reason, and if the initial condition equals 0, what the leader is going to do or the initial velocity they have is that they are going to maintain their initial velocity of 20, and due to that reason, all followers are going to converge to 20. So, this is nothing but the consensus problem, and it is also called the zero-distance formation problem. So, this is the error estimation, and this is the control.

So, if you see control carefully, then control is continuous. And, after that formation as well, we have maintained respect to 20, because I am giving you equal to 0. So, by Newton's first law, means their position is not going to change. In this way, I can maintain the space formation. This is an error in the estimation.

And finally, control, easily you can see that control is again continuous. So, what is the conclusion? First, we have understood how to formulate the multi-agent problem, and after that, we have designed some kind of continuous finite time controller, so in this way, you can be able to understand how to design continuous super twisting and higher order super twisting for multi-input multi-output systems, because so far, whatever. Higher sliding mode control I have discussed; most of them are only applicable for single input, single output systems. But using this methodology, you can also be able to apply higher sliding mode control for multi-input, multi-output systems. And there are several systems that are actually represented by multi-input, multi-output form, and due to that reason, I have included this particular lecture in this particular course. So, with this remark, I will end this lecture. Thank you very much.