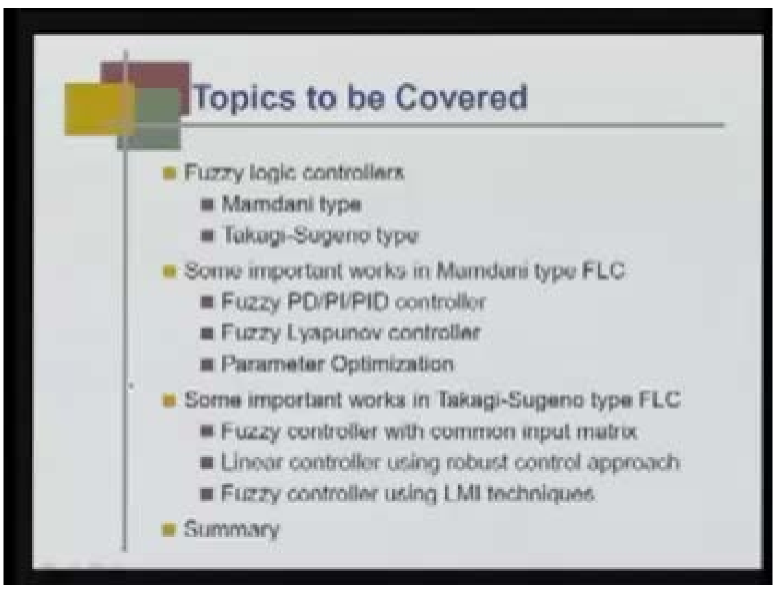
# Intelligent Systems and Control

**Prof. Laxmidhar Behera**
**Department of Electrical Engineering**
**Indian Institute of Technology, Kanpur**

## Module – 4 Lecture – 1
## Fuzzy Control: A Review

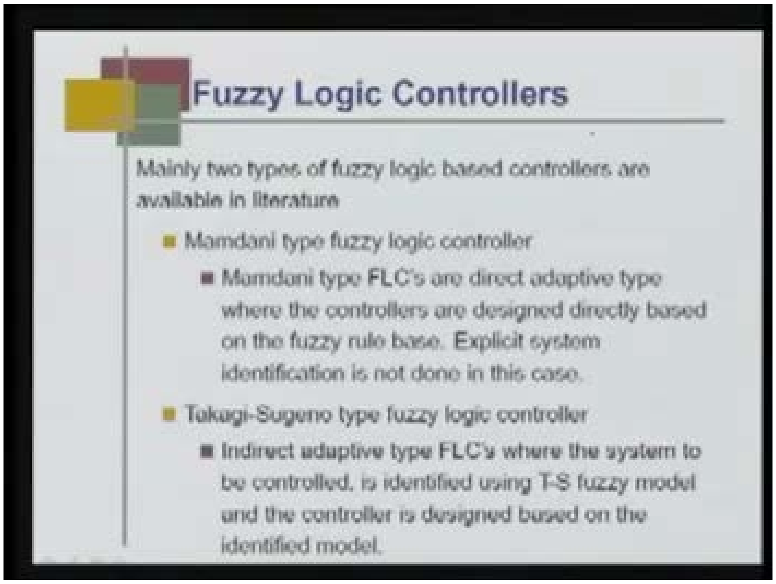
Today, we will be starting on a new subject on intelligent control. The subject is on fuzzy control. This is module 4 and we will be having the first lecture on this module on fuzzy control. Before we go in depth on how to design fuzzy controllers, we will have a review today.

(Refer Slide Time: 00:59)



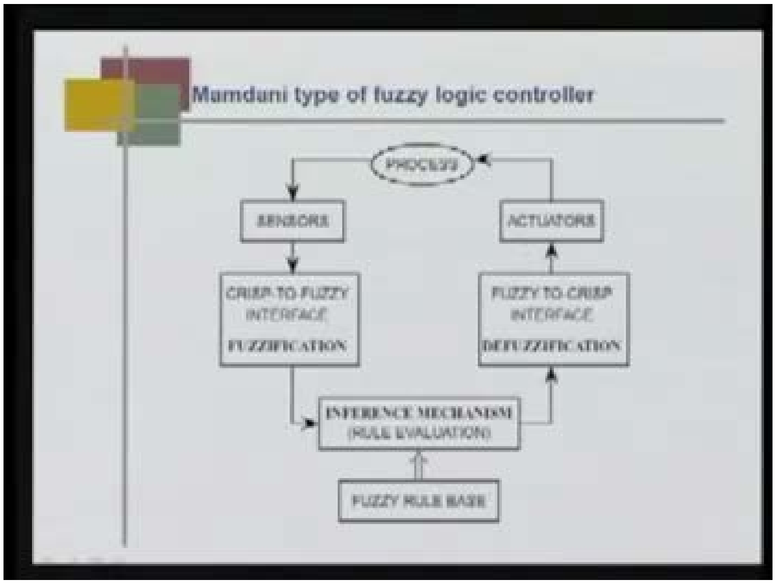

The topics that we will be covering today are fuzzy logic controllers – Mamdani type and Takagi–Sugeno type, some important works in Mamdani type FLC – fuzzy PD/PI/PID controller, fuzzy Lyapunov controller, parameter optimization, and some important works in Takagi–Sugeno type of FLC – fuzzy controller with common input matrix, linear controller using robust control approach and fuzzy controller using LMI techniques.

(Refer Slide Time: 01:33)




Mainly two types of fuzzy logic based controllers are available in literature. The first is the Mamdani type of fuzzy logic controller. The Mamdani type of fuzzy logic controllers are direct adaptive type, where controllers are designed directly based on the fuzzy rule base. Explicit system identification is not done in this case; whereas Takagi–Sugeno type fuzzy logic controllers are normally indirect adaptive type fuzzy logic controllers, but the system to be controlled is identified using T-S fuzzy model and the controller is designed based on the identified model.

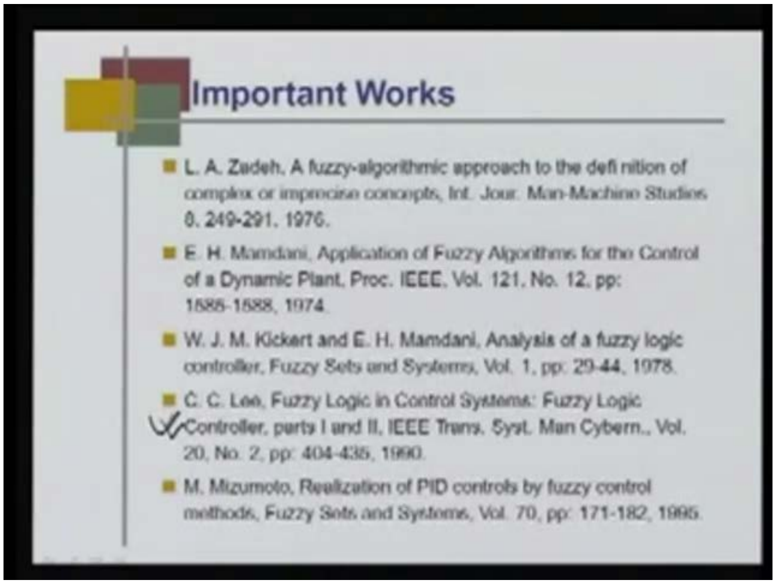
(Refer Slide Time: 02:16)




A Mamdani type of fuzzy logic controller would look like this. You have the process, sensors and crisp-to-fuzzy interface fuzzification. Process sensors means the process output are fed back through sensors. Whatever the feedback is, it is actually a crisp value and so we have a fuzzification model that converts from crisp variable to fuzzy variable. So you have fuzzification. Then, they have a fuzzy rule base. Using the fuzzy linguistic variables that are used in the rule base and the present status of the process in terms of linguistic variable, you have an inference mechanism or rule evaluation, which actually tells us what should be the control action in fuzzy linguistic variable.

That control action is defuzzified to get a crisp control action and is fed to the actuator back to the process. This is a Mamdani type of fuzzy logic controller. Here, the heart of this controller is this fuzzy rule base. Maximum research in Mamdani type of fuzzy logic controllers is regarding fuzzy rule base – how we generate the rule base and how do we optimize the parameters of the rule base. Its controller is simply expressed in terms of a fuzzy rule base.

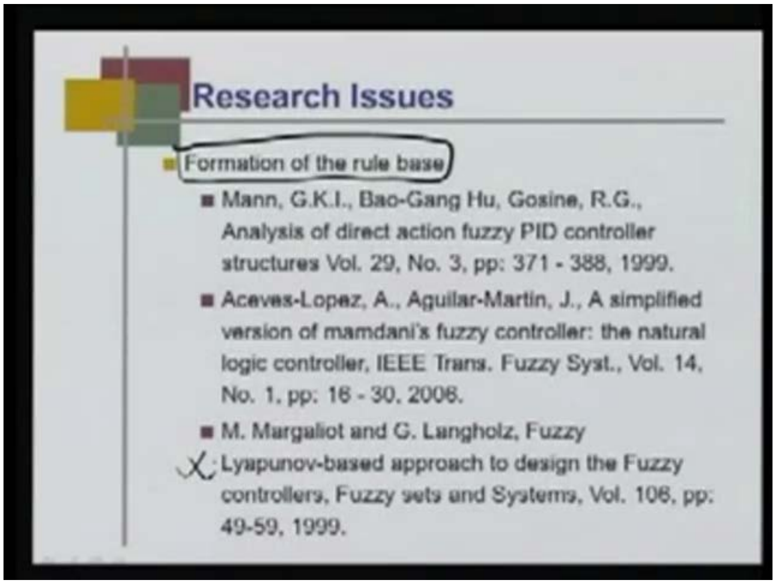
(Refer Slide Time: 04:04)




Some important works are…. As you know, Zedeh is the founder of fuzzy logic concepts – A fuzzy-algorithmic approach to the definition of complex or imprecise concepts, International Journal of Man-Machine Studies in 1976. Mamdani is the pioneer in terms of proposing the fuzzy logic controller and that is why the direct adaptive type of fuzzy logic controllers are Mamdani type – Application of Fuzzy Algorithms for the Control of a Dynamic Plant, (Refer Slide Time: 04:46) IEEE, volume 121, number 12, 1974.

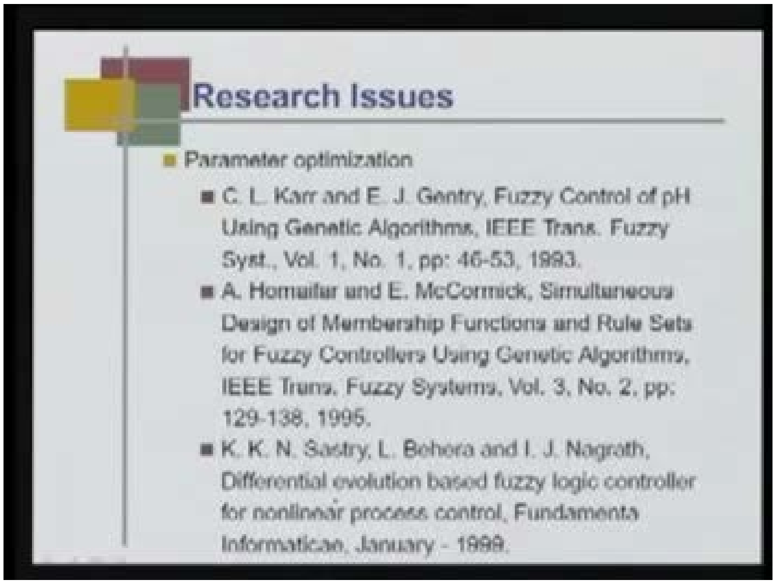
Kickert and Mamdani – Analysis of a fuzzy logic controller, Fuzzy Sets and Systems, volume 1, 1978. CC Lee's Fuzzy Logic in Control Systems is actually a survey paper – Fuzzy Logic Controller, parts I and II, IEEE Transactions on Systems, Man, and Cybernetics, volume 20, number 2. This is in 1990 and I would recommend all of you to study it. Of course, this paper deals basically with Mamdani type of controllers – you will not get anything about T-S fuzzy model in this paper. Then, Mizumoto's Realization of PID controls by fuzzy control methods, Fuzzy Sets and Systems, volume 70, 1995.

(Refer Slide Time: 05:51)



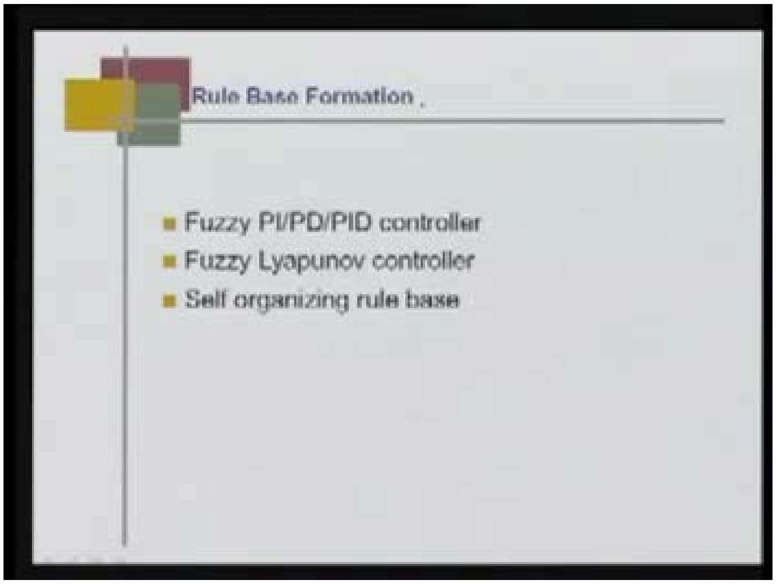

As I said, research issues are always…. Formation of the rule base. How do we form various rules? The papers that would be worth noting or the ones we should go into in detail are Mann, Bao-Gang Hu and Gosine – Analysis of direct action fuzzy PID controller structures. This was published in 1999. Lopez and Martin's A simplified version of Mamdani's fuzzy controller: the natural logic controller, IEEE Transactions on Fuzzy Systems, volume 14, number 1, 2006. Margaliot and Langhoiz's Fuzzy Lyapunov-based approach to design fuzzy controllers is something that we will be focusing on in our future classes – how to design a rule base using Lyapunov-based function or Lyapunov-based approach, Fuzzy Sets and Systems, volume 106 in 1999.

(Refer Slide Time: 07:18)



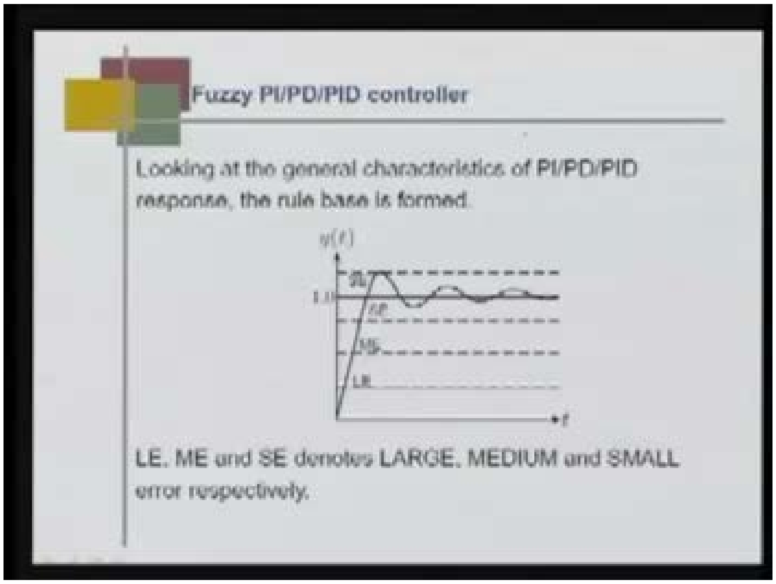

Parameter optimization is another research issue, parameter optimization of fuzzy rule base membership functions – the parameters that are contained in a fuzzy logic controller. Karr and Gentry – Fuzzy Control of pH, Using Genetic Algorithms, IEEE Transactions on Fuzzy Systems, volume 1, number 1, 1993. Homaifar and McCormick's Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controllers Using Genetic Algorithms, IEEE Transactions on Fuzzy Systems, volume 3, number 2, 1995. This is one of our own works – Sastry, Behera and Nagrath's Differential evolution based fuzzy logic controller for nonlinear process control, Fundamenta Informaticae in 1999. Here also, we use another technique called differential evolution to optimize the parameters of the fuzzy logic controller and we have implemented to a pH reactor in real time.

(Refer Slide Time: 08:28)



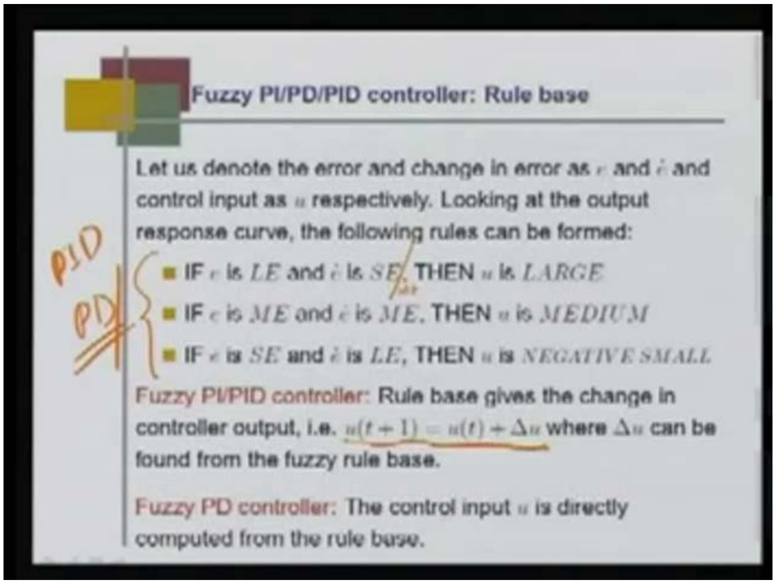

As I said, the rule base formation can be done in three types. One is using the idea of PI/PD/PID controller response – the generic idea that we have of how the normal response of a PI/PD/PID controller would look like. Another analysis is the Fuzzy Lyapunov concept – Fuzzy Lyapunov controller concept. Here, the rule base is formed using stability notion and the self-organizing rule base, where all the parameters of the fuzzy logic controller are generated using the optimization concept.

(Refer Slide Time: 09:26)




Now let us look at the fuzzy PI/PD/PID controller and how it would behave when the normal rule base is formed. This shows the general characteristics of a response of a system, given a step command. The system is at the origin and we give a step command – unity step command 1. Then, we would like our system to behave like this. What we would like to see is whether we can now guess our rules such that our system would follow a behavior of this kind.

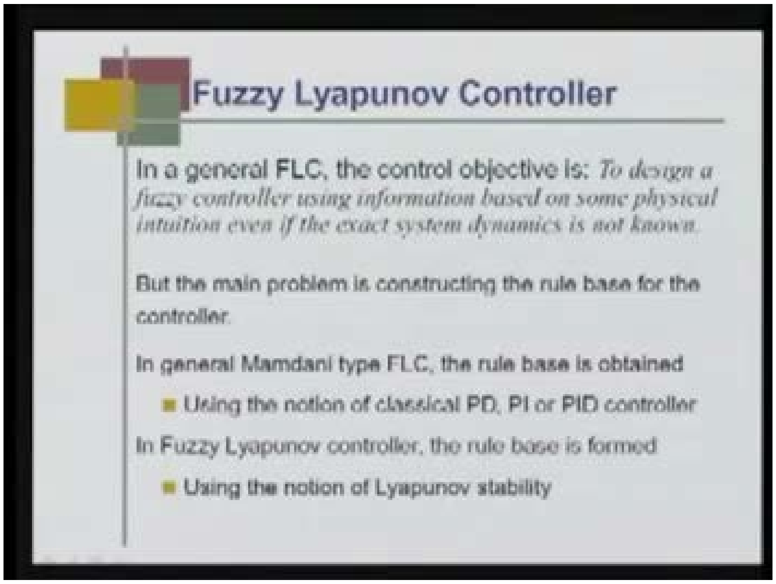
(Refer Slide Time: 10:15)



Let us denote the error, change in error as e and e dot and the control input as u respectively. Looking at the output response curve, the following rules can be formed. If e is large error and e dot the change in error is small - it can be small and it can also be medium; then u is large. If e, the error is medium and the derivative of the error is medium, then control action is medium. This is a PD type of controller. If error is small and change in error is large, then u is negative small. This means my control action should be negative so that the overshoot is not there. Fuzzy PI/PID controller….

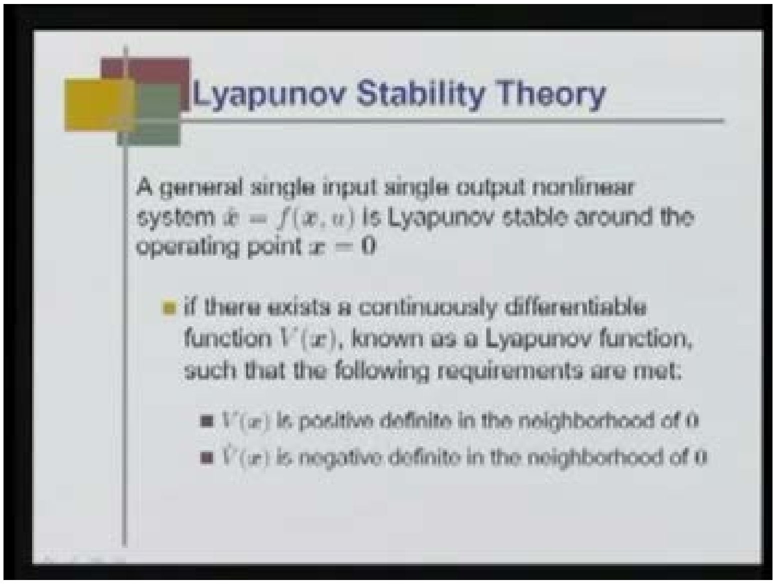
This is how the rules are generated for a PD/PID type of controller. This is how the fuzzy PI/PID controller rules are generated using the normal notion – normal notion of a response of the system to a unity step command. The normal type structure of the controller v in case of a PID controller - I write the control equation as: the present control action is the previous control action plus incremental change in control action and this incremental change in control action is computed from the rule base; whereas, if it is a fuzzy PD controller, the control input u is directly computed from the rule base.

(Refer Slide Time: 12:12)



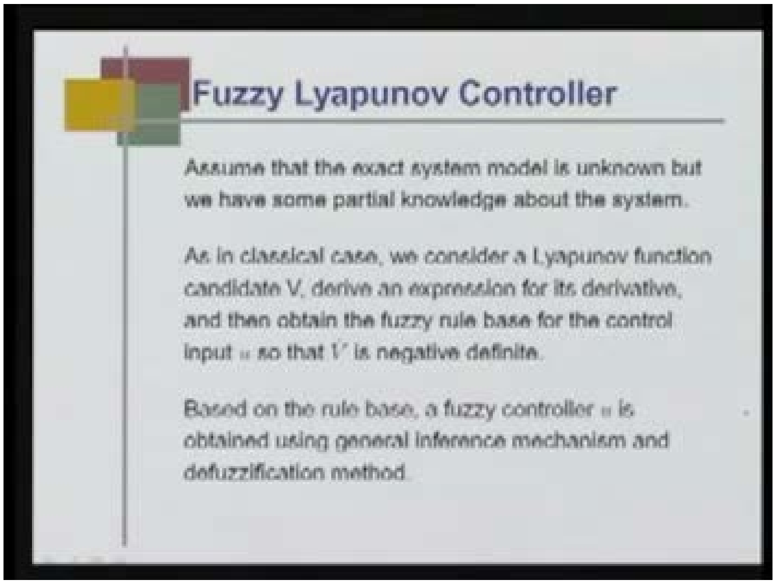

In a general fuzzy logic controller, the control objective is to design a fuzzy controller using information based on some physical intuition even if the exact system dynamics are not known, but the main problem is constructing the rule base for the controller. In general, Mamdani type of fuzzy logic control, the rule base is obtained using the notion of classical PD, PI or PID controller, but in fuzzy Lyapunov controller, the rule base is formed using the notion of Lyapunov stability.

(Refer Slide Time: 12:39)




What is Lyapunov stability? The Lyapunov stability is: A general single input single output nonlinear system x dot f x, u is Lyapunov stable around the operating point x equal to 0, if there exists a continuously differentiable function V x known as a Lyapunov function, such that the following requirements are met: V x is positive definite in the neighborhood of the origin and V dot x, the rate derivative of the Lyapunov function, is negative definite in the neighborhood of the origin.

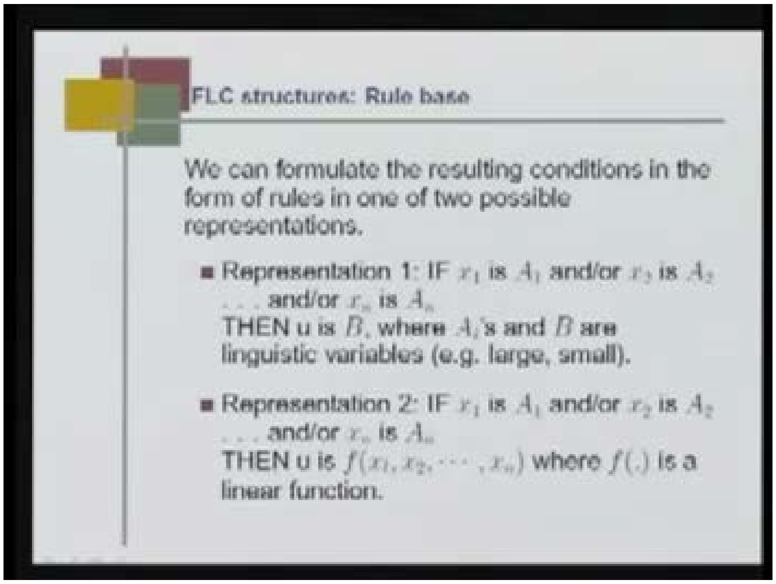
(Refer Slide Time: 13:10)




Fuzzy Lyapunov controller: assume that the exact system model is unknown, but we have some partial knowledge about the system. Then as in classical case, we consider a Lyapunov function candidate V, derive an expression for its derivative, and then obtain the fuzzy rule base for the control input u so that V dot is negative definite. Everything is qualitative; I will just show you how it is.

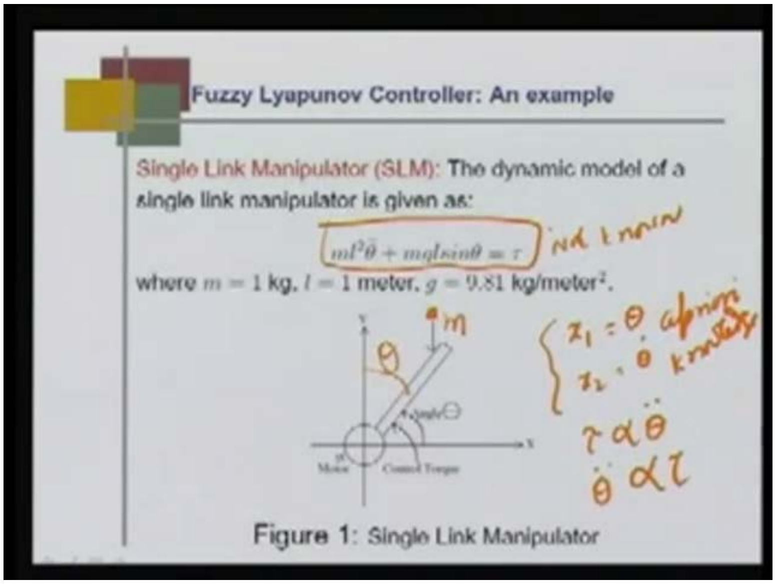
Based on the rule base, a fuzzy controller u is obtained using general inference mechanism and defuzzification method.

(Refer Slide Time: 13:48)



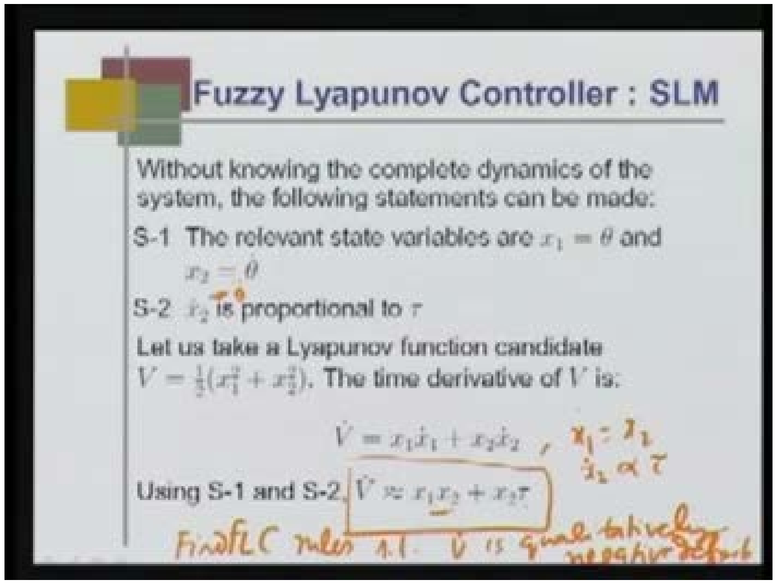

I will just explain to you now. You see that we will take two different structures of the controller for rule base formation. The first is representation 1. My FLC rule looks like this: If $x_1$ is $A_1$ and/or $x_2$ is $A_2$ and so on and/or $x_n$ is $A_n$, then my control action is B, where $A_i$'s and B are linguistic variables, like large and small; whereas in the representation 2, I say if $x_1$ is $A_1$ and/or $x_2$ is $A_2$ and so on and/or $x_n$ is $A_n$, then u is a function of $x_1$, $x_2$ and $x_n$, where f is a linear function. I will just show you. What I am trying to tell you here is that I would like to generate either of these two types of rules. How do I generate this type of rules? How do I define $a_1$ and $a_2$ apriori? This can be done in a very simple manner.

(Refer Slide Time: 15:08)




Let us think of a single link manipulator. This is my motor, this is my link whose mass is given by capital… this is actually not capital but small m (Refer Slide Time: 15:18) and the angle is actually theta and this angle is actually theta. I can write the dynamic equation of this is m l square theta double dot plus m g l sin theta equal to tau. My states are theta and $x_2$ is theta dot – this is my apriori knowledge. Another apriori knowledge is I can say my tau is directly proportional to (Refer Slide Time: 16:04) normal because of the second law of Newtonian mechanics. My acceleration is directly proportional to the applied force or the way you like to know, whatever it is. If I write like this, if I know this kind of knowledge, this much knowledge is sufficient for me to generate the rule base. This is the interesting thing – I do not have to know exact dynamics. I do not require this knowledge: m l square theta double dot plus m g l sin theta equal to tau. I will show you just now.

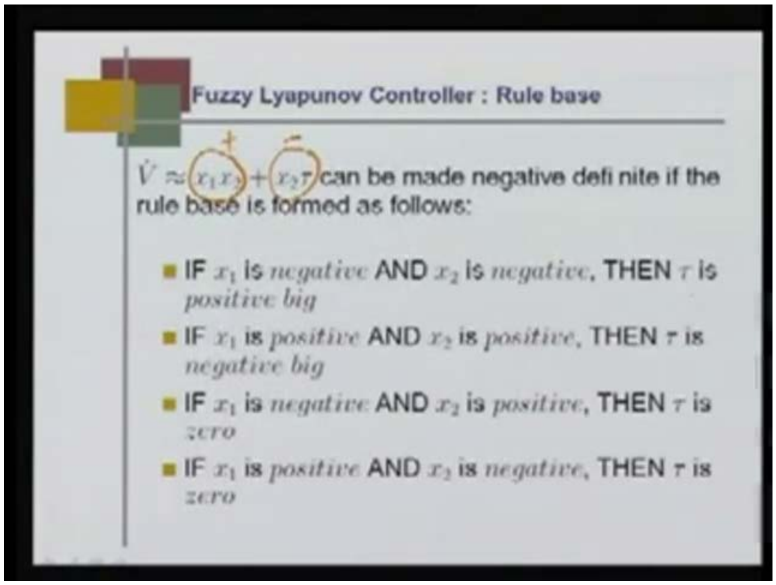
(Refer Slide Time: 16:40)




Without knowing the complete dynamics of the system, the following statements can be made about the single link manipulator. The relevant state variables are $x_1$ is equal to theta, $x_2$ is equal to theta dot, and $x_2$ dot which is actually theta double dot, is proportional to tau. Now let us take a Lyapunov function candidate V is half $x_1$ square plus $x_2$ square. The time derivative of V is V dot is…. This is my Lyapunov function for the system I can always think. Then if I differentiate, V dot is $x_1$ x dot plus $x_2$ $x_2$ dot. So, $x_1$ dot is $x_2$. Here, $x_1$ dot is $x_2$ and I can say $x_2$ dot is proportional to tau; hence, we can write qualitatively V dot, the rate derivative is $x_1$ $x_2$, and $x_2$ and $x_2$ dot is replaced by tau – approximately. This is a qualitative statement and not a quantitative statement because, we do not say that $x_2$ dot is exactly tau – no, it is just proportional to tau. But looking at this expression, I can always say… because there can be some constant here (Refer Slide Time: 18:03) – we do not worry about that.

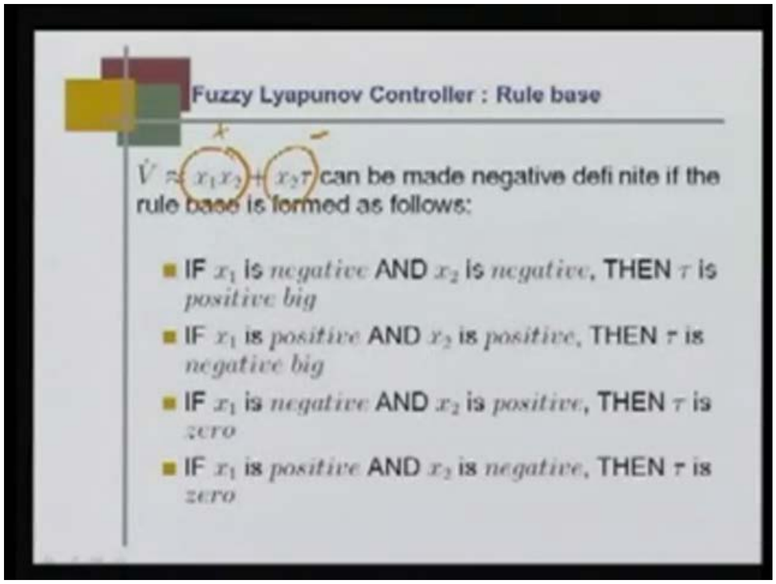
Now, I am interested in defining a rule base and I am trying to do a qualitative analysis – not a quantitative analysis. Had I been doing a quantitative analysis, I would have liked to put some kind of constant here, but since I am doing a qualitative analysis…. What is qualitative analysis? I am trying to develop a fuzzy rule base from this equation, because, the objective is that my V dot – rate derivative of the Lyapunov function – should be negative definite. How do I design a fuzzy controller rule base such that this expression is qualitatively negative definite? That is the objective. What I should write is find FLC rules such that V dot is qualitatively negative definite; this is the objective.

(Refer Slide Time: 19:13)



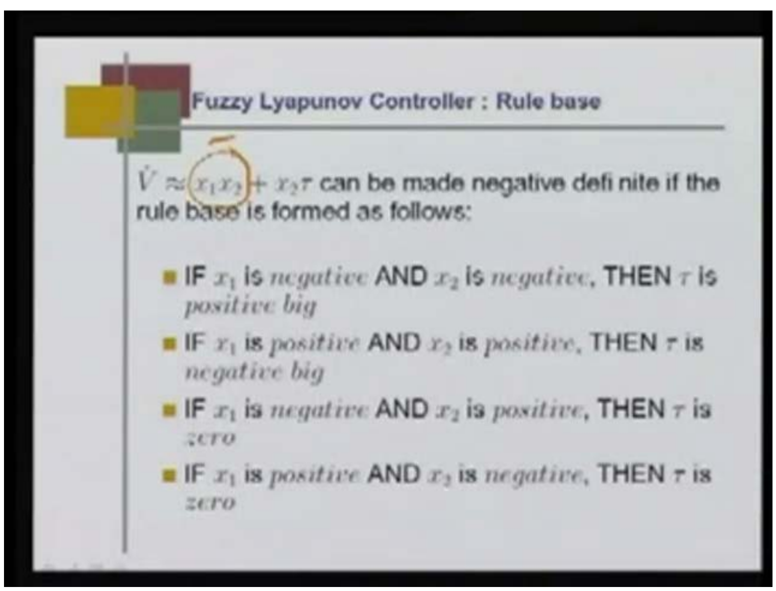

V dot is $x_1$ $x_2$ plus $x_2$ tau. This can be made negative definite if the rule base is formed as follows. If I say $x_1$ is negative and $x_2$ is negative, this quantity is positive (Refer Slide Time: 00:19:27 to 00:19:52 min); if this is positive, this has to be negative, so tau has to be negative because $x_2$ is negative. $x_1$ is negative and $x_2$ is negative, making this quantity positive. Hence, this quantity has to be negative and more than this – qualitatively. To make it negative, since $x_2$ is already negative, tau has to be positive, so tau is positive big. This is one way the first rule is formed.

(Refer Slide Time: 20:05)




The second rule is $x_1$ is positive and $x_2$ is positive. That means this quantity (Refer Slide Time: 20:09) is again positive and this quantity has to be negative now, because, this is positive. To make it negative, $x_2$ is already positive, so tau has to be negative. Then tau is negative big.

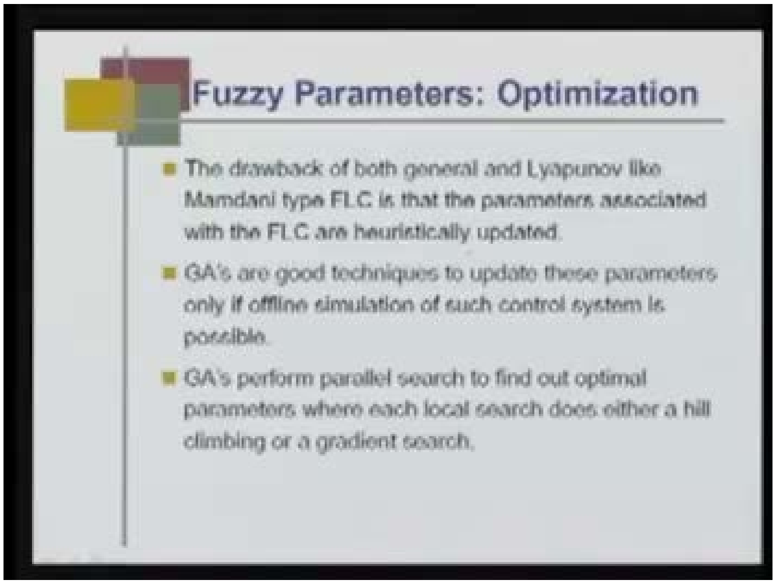
(Refer Slide Time: 20:24)




Similarly, if $x_1$ is negative and $x_2$ is positive, then this quantity is negative, so V dot is required to be negative. Now if my control action tau is 0, then also this is negative. tau is 0 – I can make tau as 0 because this is already negative. Similarly, if $x_1$ is positive and $x_2$ is negative, then this (Refer Slide Time: 00:20:47 min) is already negative. Hence again, tau can be made 0 and my control action is 0.

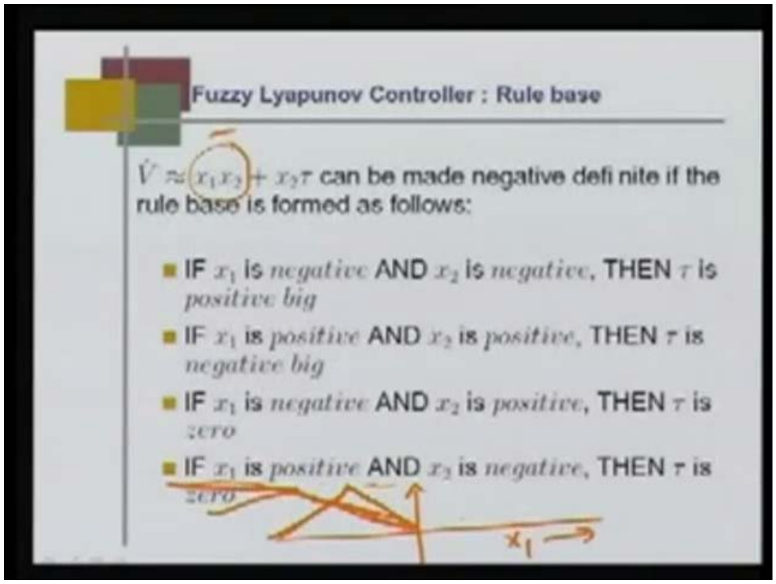
You see how we are generating rules in such a way that the rate derivative of the Lyapunov function V is a negative definite qualitatively. Once I formulate the rules, the next thing I have do is that I have to find out the parameters and I have to optimize the parameters. Parameter optimization is the next thing, but the most important is the rule base formation. This process of rule base formation is very comprehensive and in fact, it is very interesting for fuzzy logic controller. We will be having at least one class on this particular topic – how to generate the rule base using Lyapunov concept.

(Refer Slide Time: 21:42)



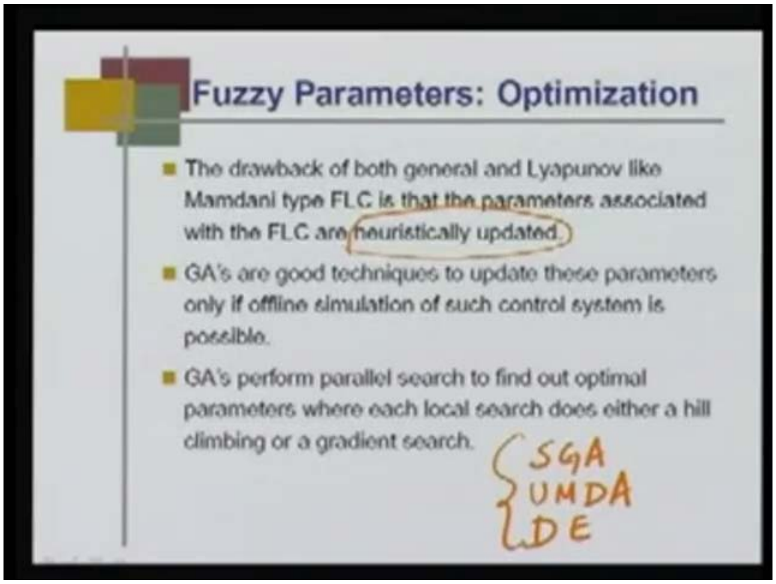

Now, parameter optimization: parameter optimization is that, once I have formulated the rules, how do we fix the parameters or how do we tune the parameters? My rules are….

(Refer Slide Time: 22:07)



For example, here, If $x_1$ is negative…. But when I say $x_1$ is negative, how do I define this $x_1$ is negative? This is my $x_1$(Refer Slide Time: 22:14 to 00:23:05 min) and I have to fuzzify $x_1$. Negative can be like this, the negative also can be like this, this is one type and this is another type. From this side, this is negative. How do I define this negative? Where should I put it? For example, the membership function for negative here is1. In this particular case, this is my membership function and maximum membership function 1 is here – for negative; similarly, for $x_2$ also. How do I (Refer Slide Time: 00:23:10 min) the fuzzification of each variable such that my performance is optimized?

(Refer Slide Time: 23:26)

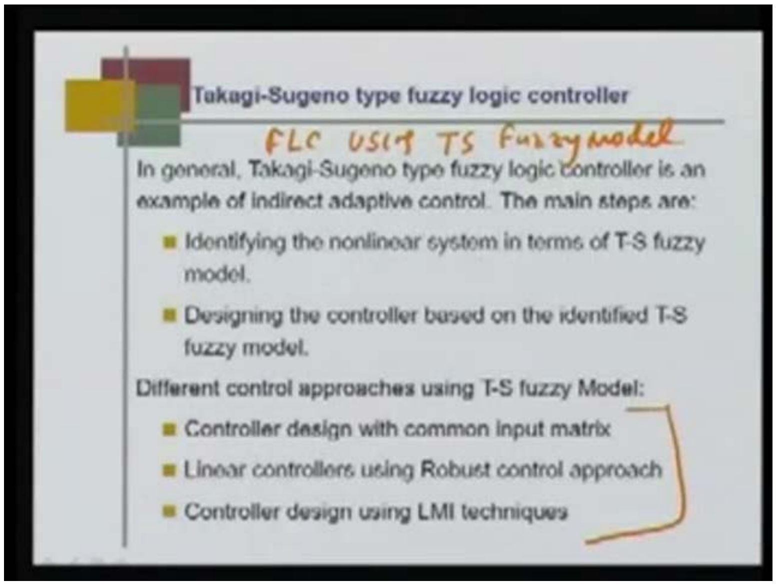



Normally, the first case is heuristically updated. We take the help of heuristics – some kind of a trial and error method. Nowadays, the normal practice has been genetic algorithms or evolutionary computation. What are the GAs? The genetic algorithms perform parallel search to find out optimal parameters, where each local search does either a hill climbing or a gradient search.

In this lecture series, we will show three types of…. One is a simple genetic algorithm, another is univariate marginal distribution algorithm and another is differential evolution. We will be covering all these three algorithms to optimize the parameters of a fuzzy logic controller.

Now, we go to the next part of our topic, which is called Takagi–Sugeno type of fuzzy logic controller.

(Refer Slide Time: 24:44)

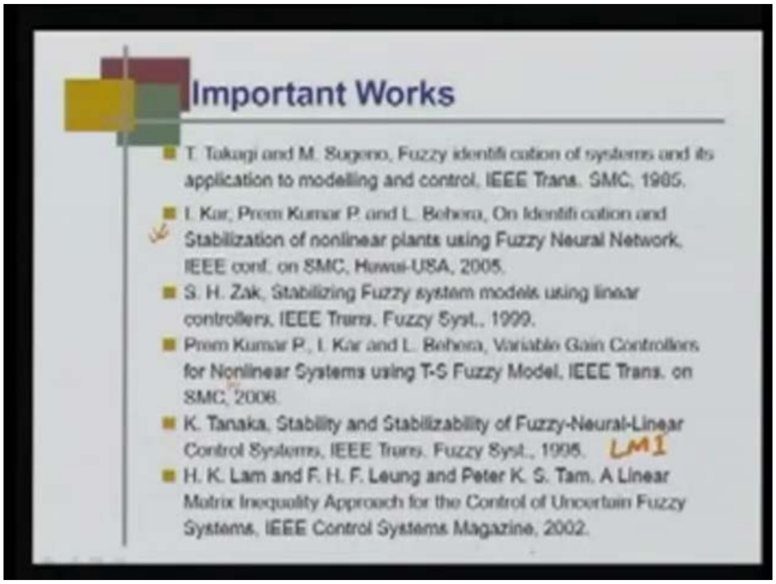



Here, we will outline various research issues that are involved in designing Takagi–Sugeno type. Actually, I would not say that this is Takagi–Sugeno type of fuzzy logic controller. We will say FLC using T-S fuzzy model. What is the meaning of that? We want to design a fuzzy logic controller using T-S fuzzy model, Takagi–Sugeno Fuzzy model. That means any nonlinear system can be represented by or approximated by T-S fuzzy model.

In general, Takagi-Sugeno type of fuzzy logic controller is an example of indirect adaptive control. The main steps are identifying the nonlinear systems in terms of T-S fuzzy model and designing the controller based on the identified T-S fuzzy model. We have a T-S Fuzzy model of the plant. I utilize the T-S Fuzzy model to design my controller. In different control approaches using T-S Fuzzy model, we will be discussing these three techniques. One is controller design with common input matrix, then linear controllers using robust control approach and then controller design using LMI techniques.

All these three approaches are very much prevalent in control research. I will just give you a hint of these different types of control research or what are the different control problems these three types of systems would define. What I am saying is that… what is the meaning of common input matrix and using that, controller design and linear controller using robust control; these are all the different control research issues.

(Refer Slide Time: 27:00)



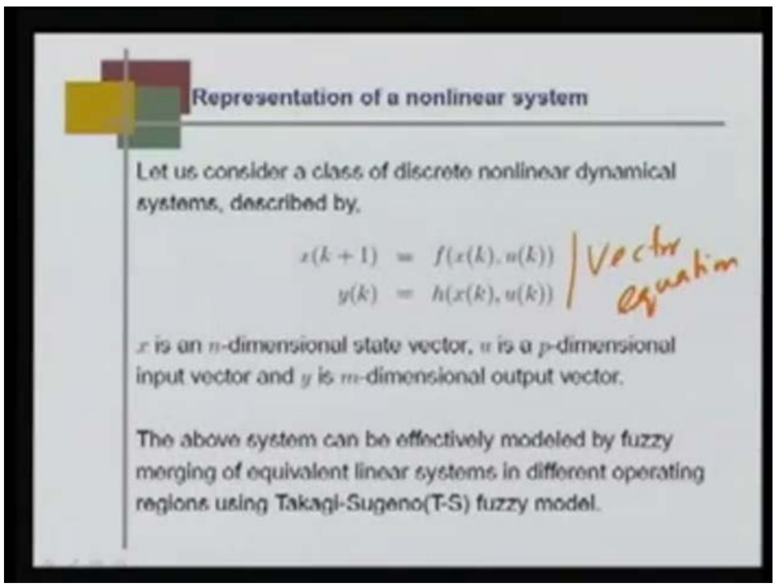

These are the important works in the T-S fuzzy model. First of all, Takagi and Sugeno proposed the T-S fuzzy model. That is in IEEE Transactions on Systems, Man, and Cybernetics way back in 1985 – Fuzzy identification of systems and its application to modeling and control. One of our works is among the other works that are relevant in terms of fuzzy controller using T-S model – On Identification and Stabilization of nonlinear plants using Fuzzy Neural Network, IEEE Conference on Systems, Man, and Cybernetics, 2005.That was the work of Zak. In this work, the second work (Refer Slide Time: 27:54), we are taking about common input matrix.

Zak's Stabilizing fuzzy system models using linear controller appeared in Transactions on Fuzzy Systems in 1999. In this, you see how to design linear controllers using the fuzzy T-S model, but in this case, Zak's approach, the controller parameters, the controller gains are fixed. That means he has a fixed gain controller. We have another paper by Prem, Indhrani and I – Variable Gain Controllers for Nonlinear Systems using T-S Fuzzy model. That is in IEEE Transactions on Systems, Man, and Cybernetics, part b in 2006. In this, we are also designing linear controllers, but the gains are varying – variable gain controller.

Tanaka proposed a notion of Stability and Stabilizability of Fuzzy-Neural-Linear control systems using the linear matrix inequality approach – LMI. That work appeared in IEEE Transactions on Fuzzy Systems, 1995. Lam, Leung and Tam A Linear Matrix Inequality Approach for the Control of Uncertain Fuzzy Systems appeared in IEEE Control Systems Magazine in 2002. You already know that, this of course, is being solved using linear matrix inequality – LMI.

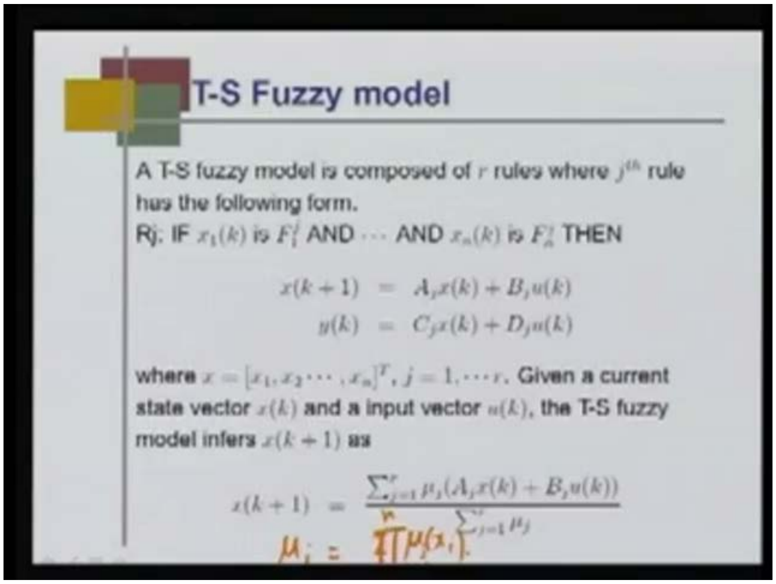
Now, we will go a little deeper into understanding these problems – what are the control problems and how the control problem is formulated for each case.

(Refer Slide Time: 29:55)




Let us consider a class of discrete nonlinear dynamical systems described by x k plus 1 is f x k u k and y k is h x k u k. You see that this is the complete nonlinear plant and these are vector equations. x is an n-dimensional state vector, u is a p-dimensional input vector, and y is an m-dimensional output vector. So, f is a vector and h is a vector. The above system can be effectively modeled by fuzzy merging of equivalent linear systems in different operating regions using the Takagi–Sugeno fuzzy model. What does it mean?

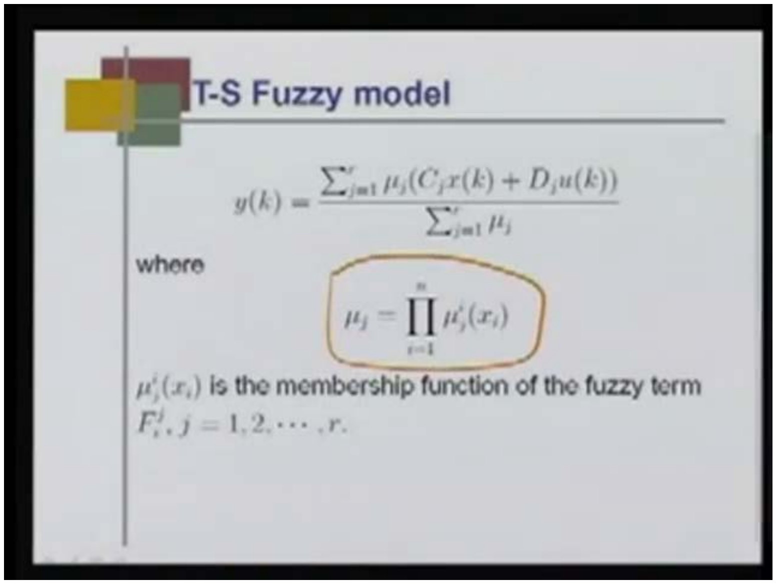
(Refer Slide Time: 30:36)




A T-S fuzzy model is composed of r rules where the j th rule has the following form. What you say is if $x_1$ k is $F_1$ j and so on and $x_n$ k is $F_n$ j, then my x k plus 1 is a linear system $A_j$ x k plus $B_j$ u k and y k is $C_j$ x k plus $D_j$ u k, where x is the system's n states $x_1$ to $x_n$ and the rule here j…. This is the j th rule. So I have r rules. Given a current state vector x k and input vector u k, the T-S fuzzy model infers x k plus 1 as x k plus 1 is sigma j equal to 1 to r $mu_j$ $A_j$ x k plus $B_j$ u k upon sigma j equal to 1 r $mu_j$.

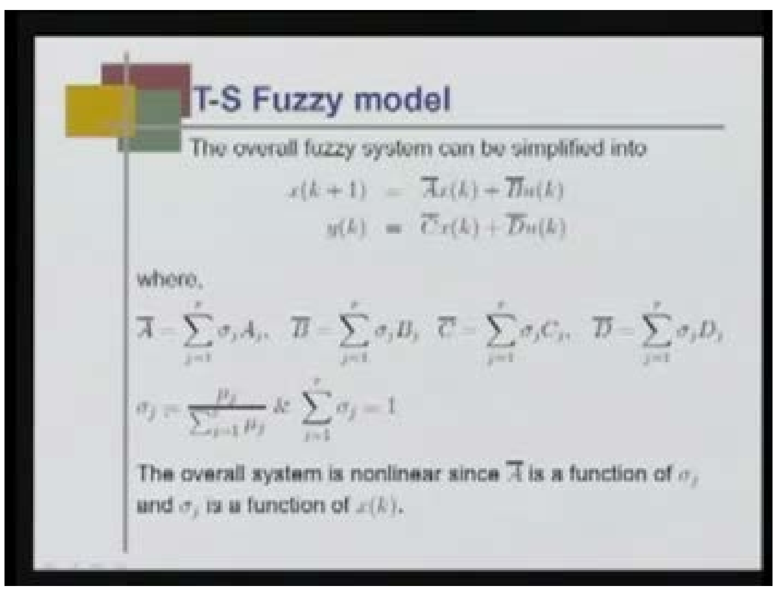
What is this $mu_j$? $mu_j$ is actually sigma $mu_j$ $x_i$ product i equal to 1 to n. $mu_j$ is $mu_j$ $x_1$, $mu_j$ $x_2$ until $mu_j$ because when I say $x_1$ k is $F_1$ j, given the crisp value of $x_1$, I get a specific membership function here. Similarly, given a crisp value of x and k, I get a specific $mu_j$ $x_n$. The $mu_j$ associated with the entire rule is normally computed by the product principle; that is, I multiply each membership function and find out what is $mu_j$.

(Refer Slide Time: 32:38)



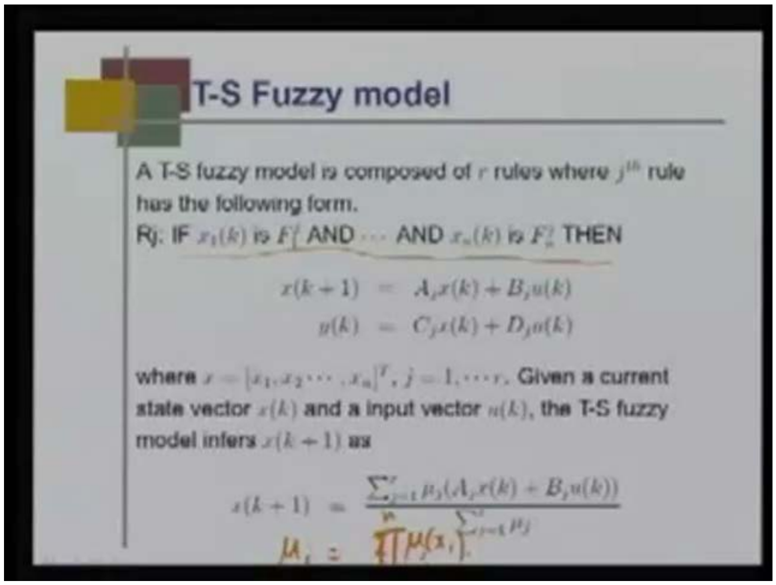

This is our $\text{mu}_j$, the product principle. y k is similarly $\text{mu}_j$ into associated with the rule j and my y is $C_j$ x k plus $D_j$ u k and you sum over j equal to 1 to r, r rules and divided by sum j equal to 1 to r $\text{mu}_j$. So, $\text{mu}_j$ i $x_i$ is the membership function of the fuzzy term $F_i$ j. I explained this.

(Refer Slide Time: 33:14)



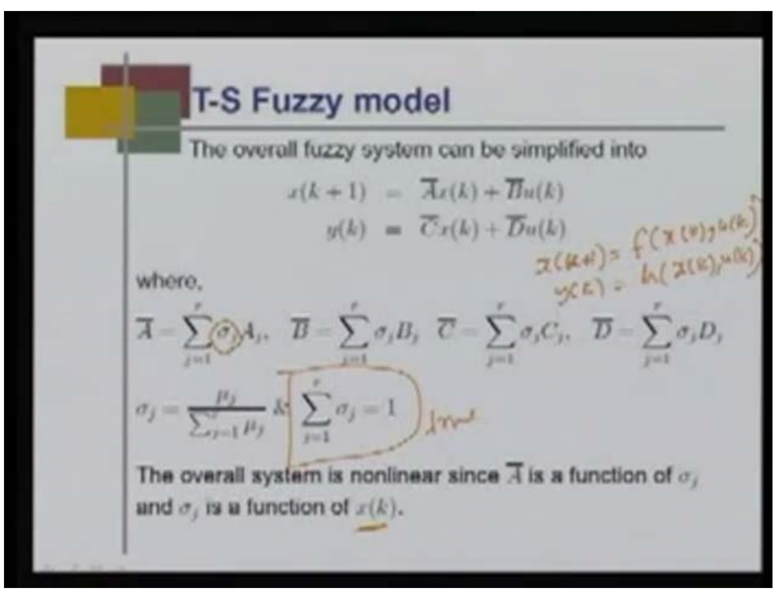
Pay a little attention here because, we will be discussing more and more about this kind of structure; because, we will be going deeper into the control system aspect in this particular model on fuzzy control.

(Refer Slide Time: 33:30)




We had this rule and we have r such rules.

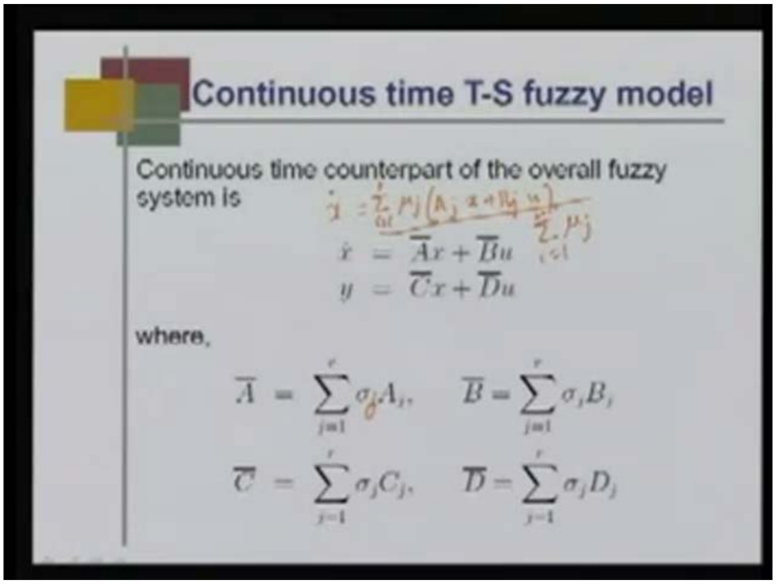
(Refer Slide Time: 33:38)



Given these r rules, now we can write the overall fuzzy system, the fuzzy dynamics that the nonlinear system we described. What was the nonlinear system? The nonlinear system was x k plus 1 is f x k u k and y k is h x k u k. This is my nonlinear system and in the T-S fuzzy model, the nonlinear system would look like this. (Refer Slide Time: 00:34:23 min) You represented the same nonlinear system as x k plus 1 and here instead of this nonlinearity, we represent in a very convenient format, which is A bar x k plus B bar u k. It looks linear. It is not linear, but looks linear: A bar x k plus B bar u k – very convenient notation, and y k is C bar x k plus D bar u k, where A bar is equal to j equal to 1 to r $sigma_j$ $A_j$, B bar is j equal to 1 to r $sigma_j$ $B_j$, C bar is $sigma_j$ $C_j$ and D bar is $sigma_j$ Dj, where $sigma_j$ is a normalized membership function and it is $mu_j$ over $sigma_j$ equal to 1 to r $mu_j$.

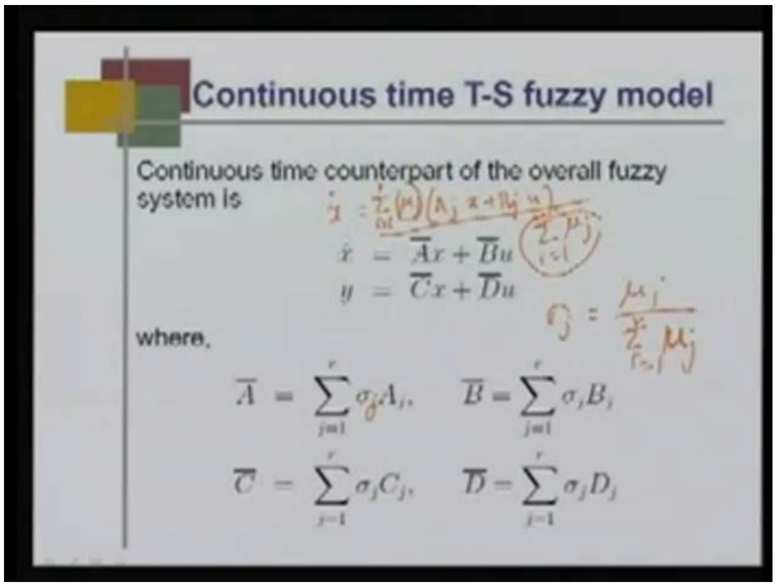
You must know that from j equal to 1 to r, $sigma_j$ is 1. This is always satisfied and always true (Refer Slide Time: 00:35:28min). The overall system looks linear, but it is not linear; this is nonlinear. Why? The overall system is nonlinear since A bar is a function of $sigma_j$ and this $sigma_j$ is a function of x k because, the $sigma_j$ defines the fuzzy membership function of a state variable x k. Hence, the system is nonlinear.

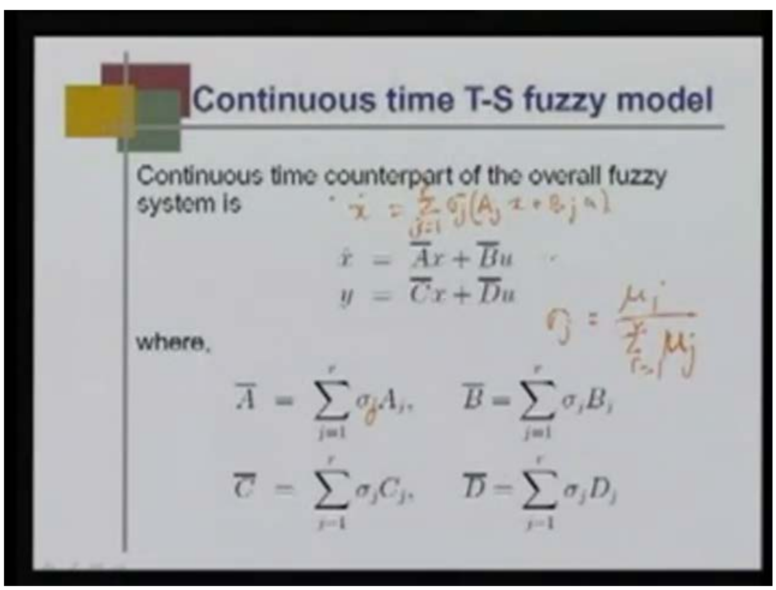
(Refer Slide Time: 36:01)




Continuous time T-S fuzzy model: The continuous time counterpart of the overall fuzzy system is…. Just like we said this is discrete time (Refer Slide Time: 00:36:11min), the continuous time is also similar. x dot is A bar x plus B bar u, y is c bar x plus D bar u, where again A bar is $sigma_j$ $A_j$, j equal to 1 to r and B bar is $sigma_j$ $B_j$, j equal to 1 to r. C bar is $sigma_j$ $C_j$, j equal to 1 to r and D bar is …. You should know how you are finding this. We wrote down x dot is sigma i equal to 1 to r $mu_j$ into $A_j$ x plus $B_j$ u divided by sigma i equal to 1 to r $mu_j$.

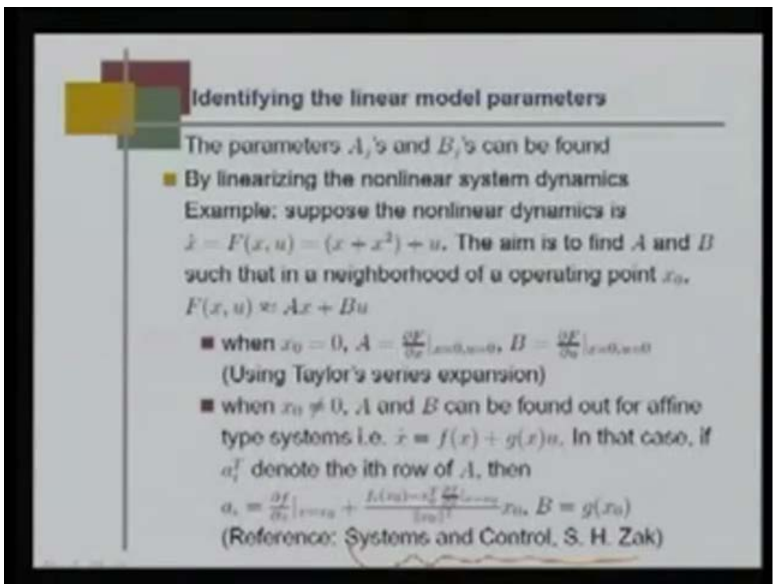
(Refer Slide Time: 37:15)




When I divide this $mu_j$ by this quantity (Refer Slide Time: 00:37:18min) and represent by $sigma_j$, where $sigma_j$ is $mu_j$ by sigma i equal to 1 to r $mu_j$, then this representation becomes x dot is simply sigma j equal to 1 to r $\text{sigma}_j$ $A_j$ x plus $B_j$ u. You can easily see now that if I write in terms of x dot equal to A bar x, so $\text{sigma}_j$ $A_j$ j equal to 1 to r. Similarly, B bar is $\text{sigma}_j$ $B_j$ j equal to 1 to r $\text{sigma}_j$ $B_j$. Be very clear that once I say T-S fuzzy model, then my system dynamics in continuous time looks like x dot equal to A bar x plus B bar u, which looks very similar to linear system but they it is not linear because A bar is a function of $\text{sigma}_j$ and $\text{sigma}_j$ is a function of x and hence, this is nonlinear. Similarly, we also talked about discrete time system (Refer Slide Time: 00:38:48min) which also looks linear, but it is not linear. x k plus 1 is A bar x k plus B bar u k, where A bar and B bar are functions of x k because they are function of $\text{sigma}_j$.

(Refer Slide Time: 39:02)

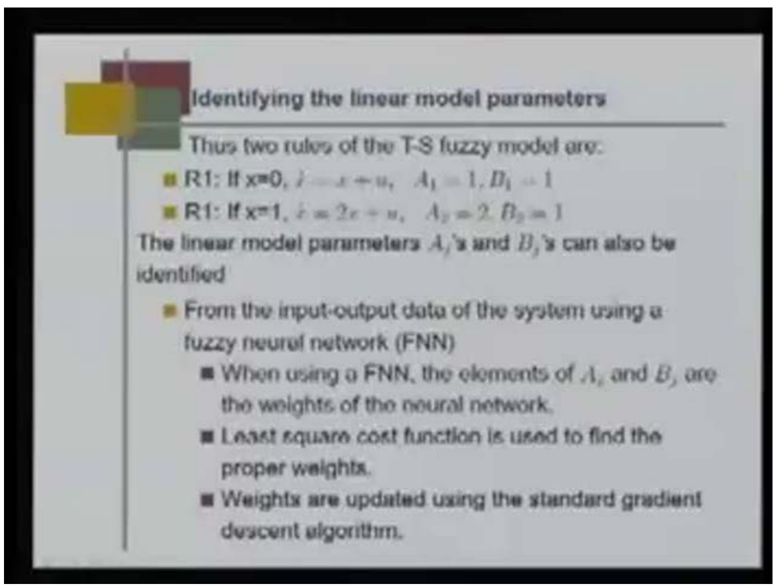



Once we understood what the T-S fuzzy model is, the next step is to derive the T-S fuzzy model. We can derive the T-S fuzzy model by direct system identification or by linearization of an actual nonlinear plant. I will just explain to you, how we linearize a plant.

Given a nonlinear system, you see that x dot is F x, u which is x plus x square plus u and I want to linearize this. We can use Taylor's series expansion but I can only apply that when x is equal to 0 and u is equal to 0. If I am trying to linearize around x is equal to 0 and u is equal to 0, then I can write the expression as this approximation (Refer Slide Time: 00:40:00 min) A x plus B u, where A is dow F upon dow x and B is dow F upon dow u.

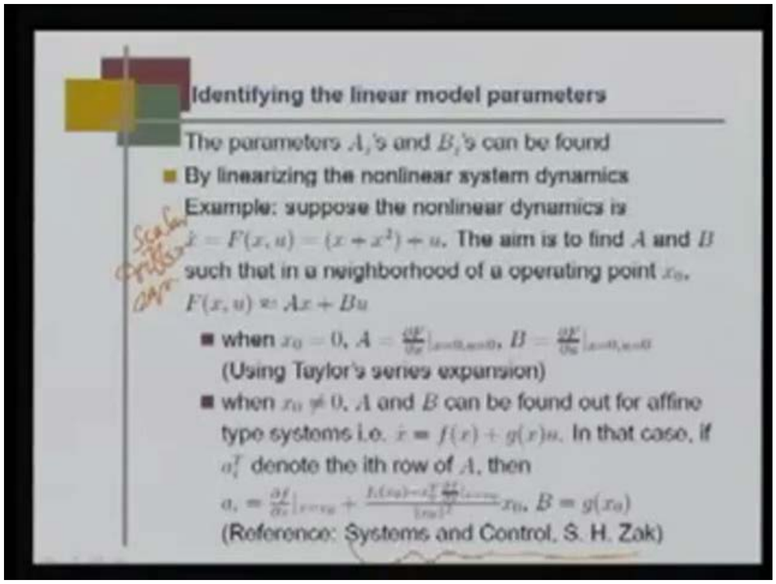
If I am linearizing around x is equal to 0, u is equal to 0, it means origin. But if x is not equal to 0, then you can follow the book - Systems and Control by Zak; there are other books also, where there is a method how to linearize a nonlinear system around some other points that are not the origin. There is a formula here. Given x dot equal to f x plus g x u, we can find out the A matrix. In that case, $A_i$ transpose denotes the i th rule of A where $A_i$ is computed by this formula and B is g $x_0$. There are various ways – you just have to learn how to linearize a nonlinear system around various points. This is not a difficulty. You just try to understand that we can linearize nonlinear systems around various operating points.

(Refer Slide Time: 41:24)




Using two rules of T-S fuzzy model, we can say….

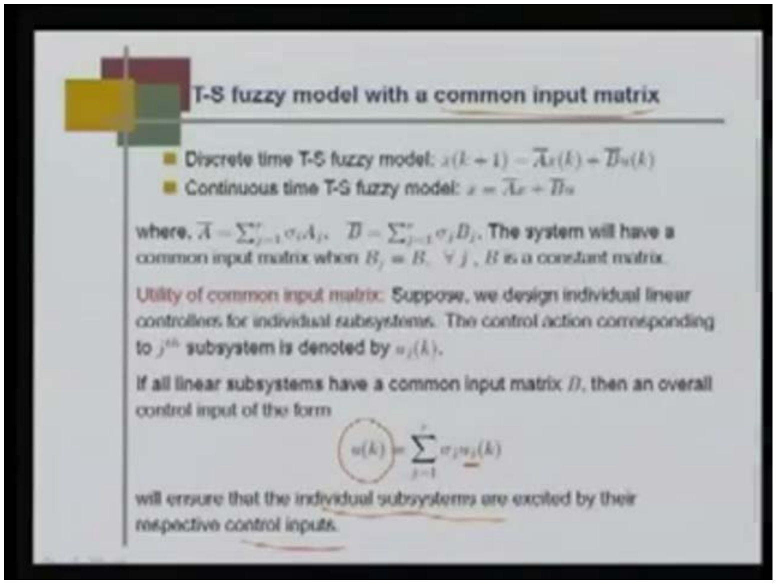
(Refer Slide Time: 41:27)




This is a scalar differential equation – x dot is x plus x square plus u. We can write two rules for this (Refer Slide Time: 00:41:39min). If x is equal to 0, x dot is x plus u and if x is equal to 1, x dot is equal to 2 x plus u, where the corresponding matrices $A_1$ is 1 and $B_1$ is 1, $A_2$ is 2 and $B_2$ is 1. These are all scalar values because, the differential equation is scalar. I just demonstrated this for a scalar differential equation. You can also do it for a vector differential equation. In that case, these will come (Refer Slide Time: 42:12).

This is one approach. The other approach is that we directly use fuzzy neural network. From the input-output data of the system using a fuzzy neural network, we can also estimate these parameters very easily. Using gradient descent algorithm or various kinds of algorithms, you can do that. What we try to do in this case is that we represent a neural network where these rules (Refer Slide Time: 00:42:53min) are encoded in terms of neural network parameters. Then, the neural network parameters are updated using input-output data using the gradient descent rule.

(Refer Slide Time: 43:11)

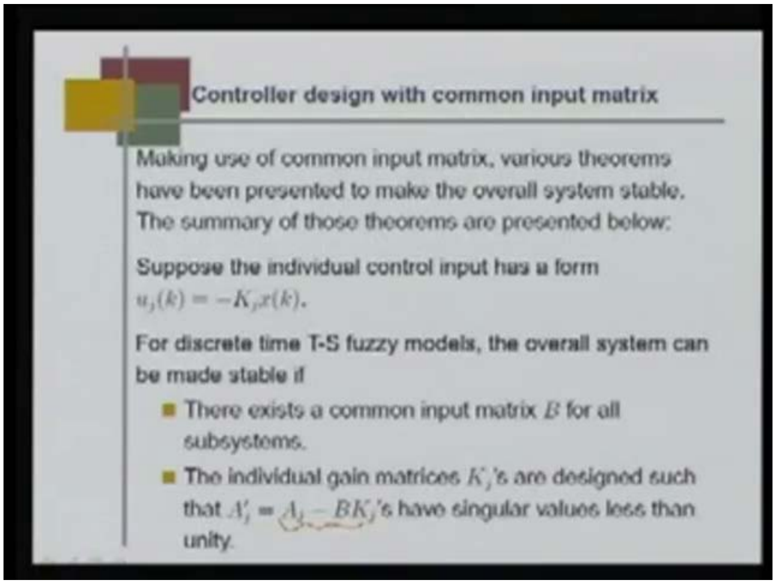



We are now very clear that we can write any nonlinear system using either a discrete time T-S Fuzzy model or a continuous time T-S fuzzy model. This is a discrete time T-S fuzzy model and this is my continuous time T-S fuzzy model, where we have already defined what is A bar and what is B bar. Now I assume that the system will have a common input matrix. When we say common input matrix, it means that for all fuzzy zones, for every rule, the associated control matrix… because for each rule in T-S fuzzy model, we have a linear system dynamics, that is, x dot is $A_i$ x plus $B_i$ u.

If this control matrix $B_i$ is the same for all rules, then this is called common input matrix. This is what I say: $B_j$ is equal to B for all j, B is a constant matrix. In that case, this is called common input matrix. What is the utility of this common input matrix? Suppose we design individual linear controllers for individual subsystems, the control action corresponding to the j th subsystem is denoted by $u_j$ k. What do we do? If I have a common input matrix, I compute what is $u_j$ for individual subsystems such that the individual subsystem is stable.

Once I compute $u_j$, my control action u k, which is a fuzzy blending of all control actions is sigma j $u_j$ k over j equal to 1 to r. This ensures that individual subsystems are excited by their respective control inputs, which is $u_j$ k. That means if I am giving to the actual plant $u_k$, to the actual plant I am actuating the control signal $u_k$, it means apparently that each individual subsystem if they are they are in reality they are being excited by the control action $u_j$. This is a very important notion. This is a theorem that we will prove in one of the coming classes.

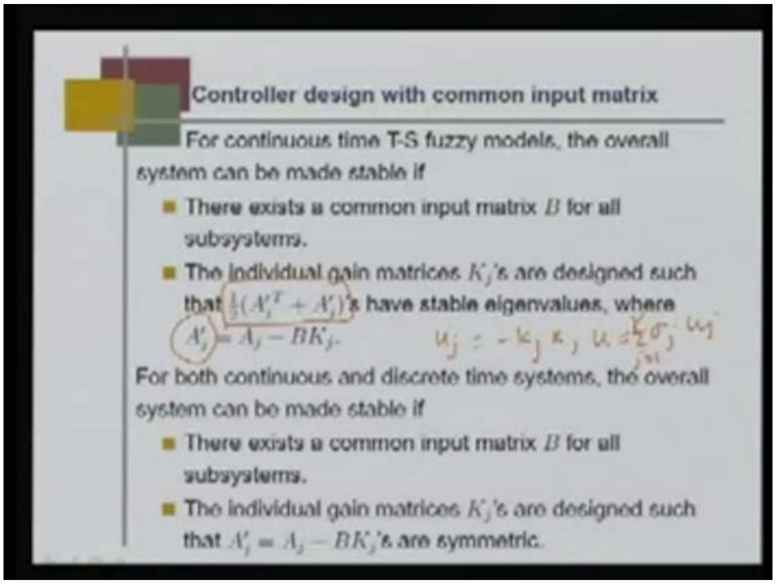
(Refer Slide Time: 45:49)




Making use of common input matrix, various theorems have been presented to make the overall system stable; in fact, we have done extensive work in this area. For example, suppose the individual control input has a form $u_j$ k is minus $K_j$ x k. For discrete time T-S fuzzy model, the overall system can be made stable if there exists a common input matrix B for all subsystems and the individual gain matrices $K_j$'s are designed such that $A_j$ dash is equal to $A_j$ minus B $K_j$'s have singular values less than unity.

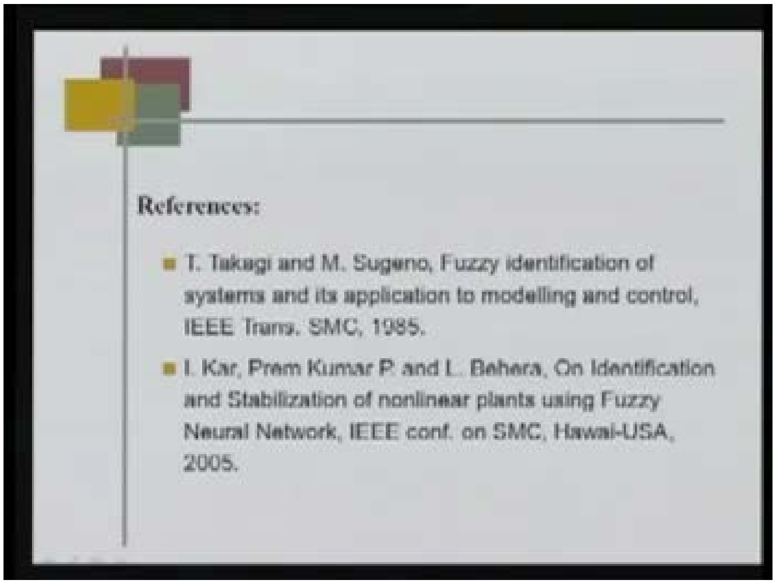
If this is the case, then we can say my T-S fuzzy model, my controller which is this one (Refer Slide Time: 46:31) my controller where $u_j$ k is designed by this formula where $u_j$ k is minus $K_j$ x k and then my system is stable provided $A_j$ minus B $K_j$ have singular values less than unity for each subsystem. How many subsystems do you have? You have r rules and that means you have r subsystems. For r subsystems, each of these quantities has maximum singular values and if less than unity, then the system is stable. The proof and other things will be shown in the following classes.

(Refer Slide Time: 47:16)



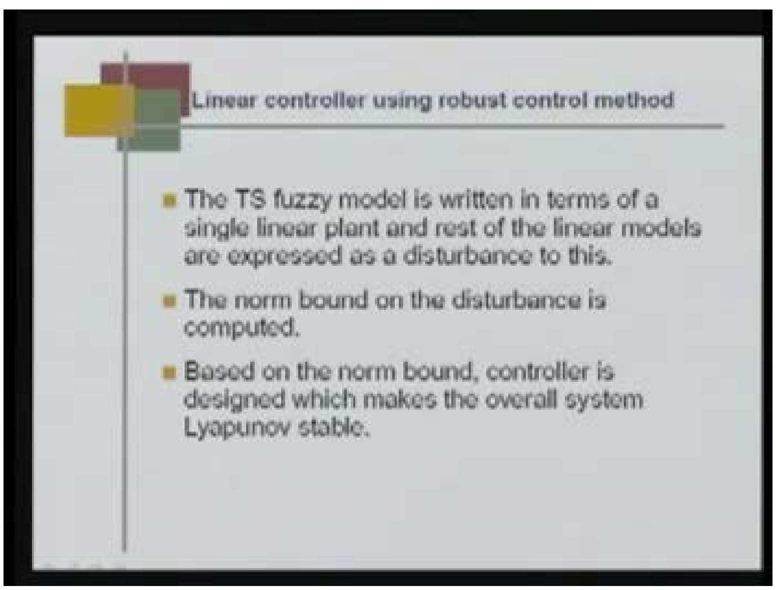

Similarly, another interesting result about this common input matrix is - for continuous time T-S fuzzy model, the overall system can be made stable, if there exists a common input matrix B for all subsystems and the individual gain matrices $K_j$'s are designed such that… This is called Hermitian part of the matrices $A_j$ dash. If this term has stable Eigen values, where $A_j$ dash is $A_j$ minus B $K_j$; that is, if I am designing a controller $u_j$ is minus $K_j$ x and my overall controller is sigma j $u_j$, $sigma_j$ equal to 1 to r; this is my overall controller for a continuous time system; then the system is stable provided the Hermitian part of this $A_j$ dash, which is this one – the Hermitian part (Refer Slide Time: 48:14) has stable Eigen values.

(Refer Slide Time: 48:18)



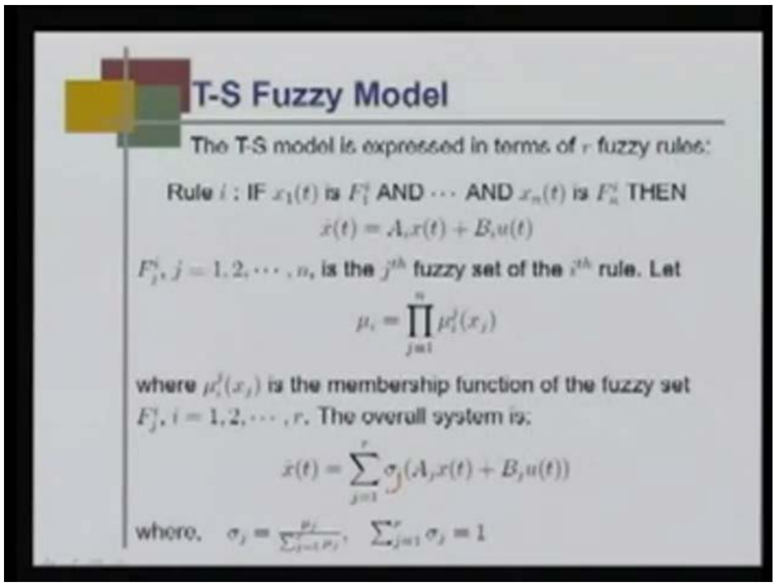

You can look at Takagi and Sugeno's Fuzzy identification of systems and its application to modeling and control and you get the idea of T-S fuzzy model. For the common input matrix, you can get the idea from this paper – On Identification and Stabilization of nonlinear plants using Fuzzy Neural Network, IEEE Conference on Systems, Man, and Cybernetics in 2005.

(Refer Slide Time: 48:45)



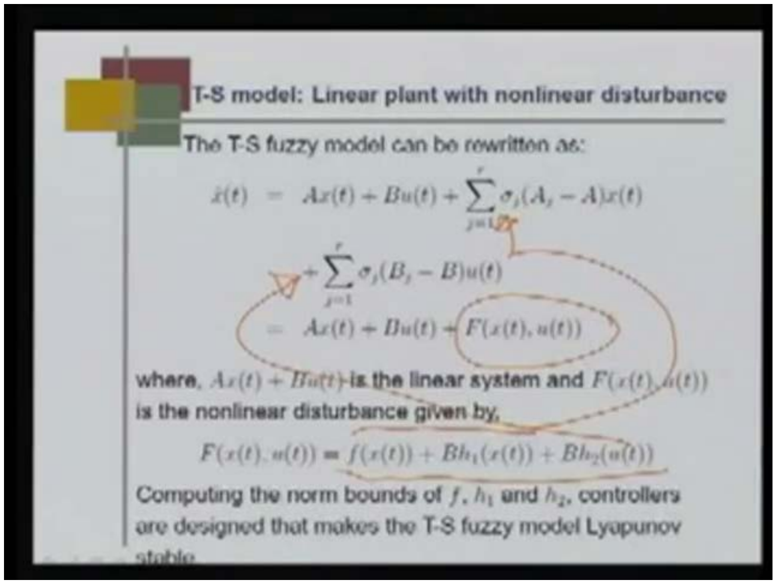
Now, we go to the linear controller using robust control method. In this, the T-S fuzzy model is written in terms of single linear plant and the rest of the linear models are expressed as a disturbance to this. The norm bound on the disturbance is computed. Based on the norm bound, the controller is designed, which makes the overall system Lyapunov stable.

(Refer Slide Time: 49:07)




What does it mean? I have the i th rule as I said earlier. If $x_1$ t is $F_1$ i and so on and $x_n$ t is $f_n$ i, then x dot t is $A_i$ x t plus $B_i$ u t and $mu_i$ is the associated membership function with the i th rule. Then, we saw that the overall fuzzy dynamics x dot t is $sigma_j$ $A_j$ x t plus $B_j$ u t, where sigma j is defined like this.

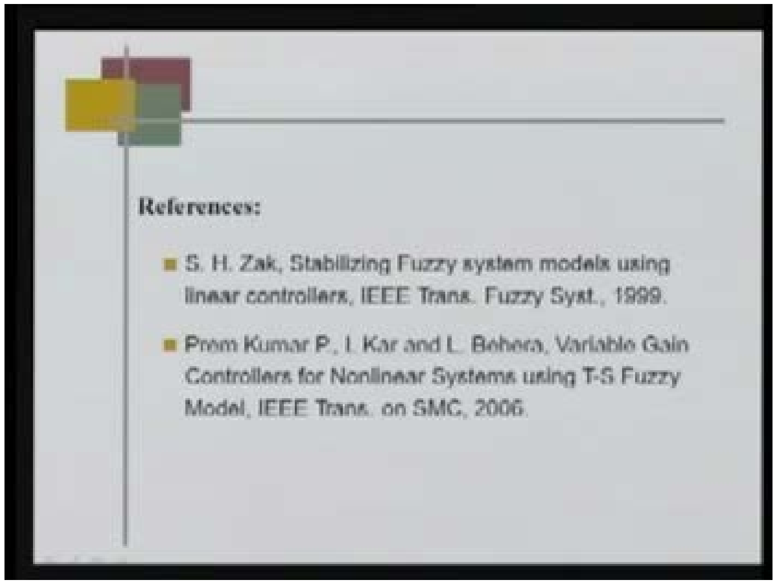
(Refer Slide Time: 49:37)




This overall fuzzy T-S model can be expanded, x dot is as a nominal plant A x t plus B u t and these are we can say disturbance term and this disturbance term (Refer Slide Time: 00:49:53min) can be again further categorized into three categories. This is B $h_2$ u t, B $h_1$ x t and f x t and these two (Refer Slide Time: 00:50:06 to 00:50:15 min) take care of this part and this one takes care of this part and then we define norm bounds on these quantities. By defining the norm bounds, we can design the controller around this nominal plant.

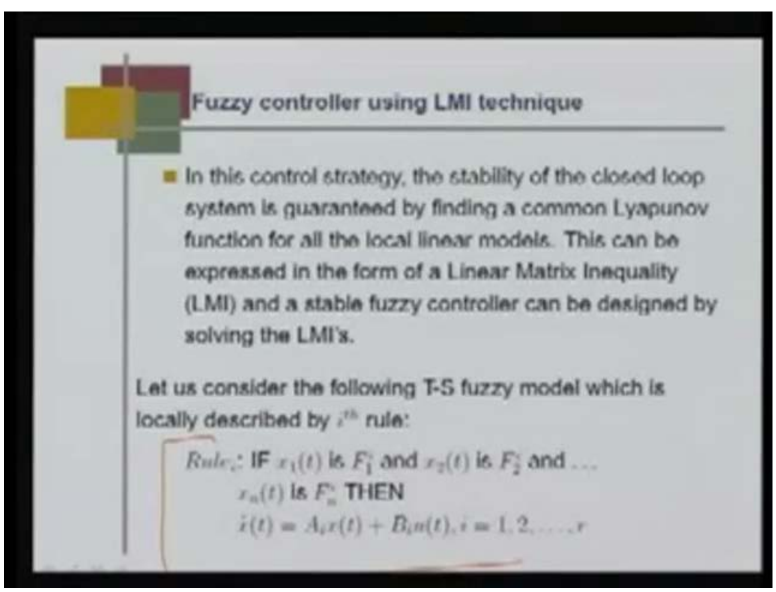
There are various methods to the design controller, which we will not discuss in detail. Computing the norm bounds of f, $h_1$ and $h_2$, controllers are designed that makes the T-S fuzzy model Lyapunov stable. This is the problem formulation. What is the problem? Given a T-S fuzzy model, express this T-S fuzzy model as x dot t around a nominal plant and then disturbance (Refer Slide Time: 00:50:51min). Then, using the robust control theory and Lyapunov stability theorem, we can design the controller.

(Refer Slide Time: 51:01)



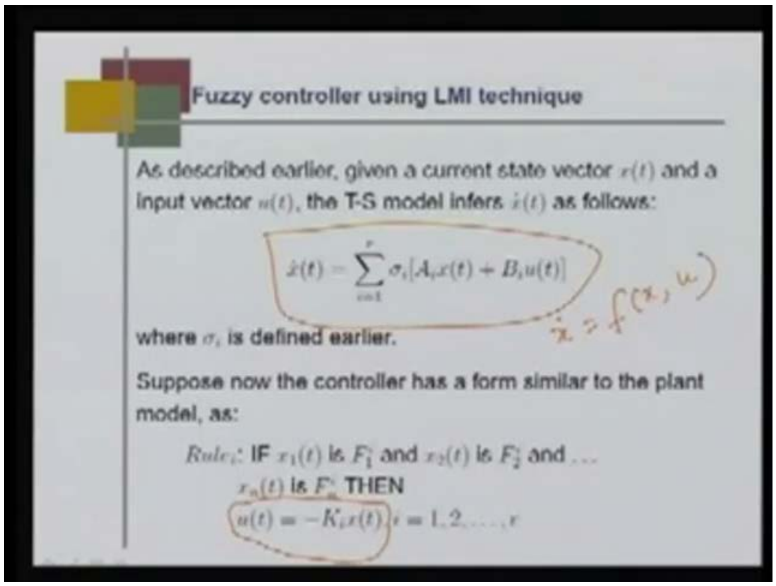

For reference, we have a paper in IEEE Transactions on Systems, Man, and Cybernetics, 2006. Zak also has a paper on this – Stabilizing Fuzzy system models using linear controllers, IEEE Transactions on Fuzzy Systems, 1999. You can refer these papers for more, and of course, we will be discussing this aspect in this class later.

(Refer Slide Time: 51:22)



We talked about common input matrix and we talked about robust control theory – how to design controller using the T-S fuzzy model. Now, we will talk about LMI technique.

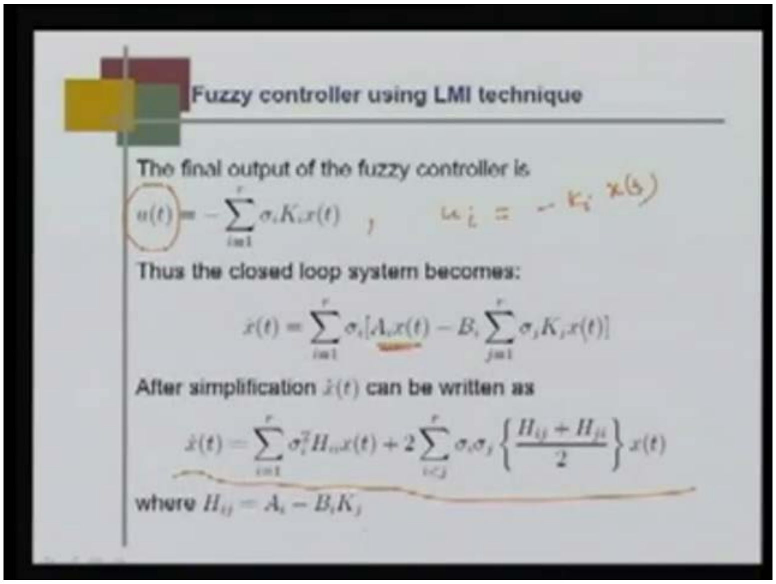
(Refer Slide Time: 51:43)




Again in LMI technique, as we said, the rule is given; given rule i, we have a T-S fuzzy model. This is our T-S fuzzy model. You remember this; because you should learn this model by heart. T-S fuzzy model means x dot t is sigma$_i$ equal to 1 to r sigma$_i$ $A_i$ x t plus $B_i$ u t; you should learn this by heart because, we have to be very clear. When we design a control system, the model should be very clear to us – what it means. This is not a linear model; although it looks linear, this is a nonlinear model. This approximates the system. The nonlinear system x dot is f x and u. This is a nonlinear system. It approximates any nonlinear system.

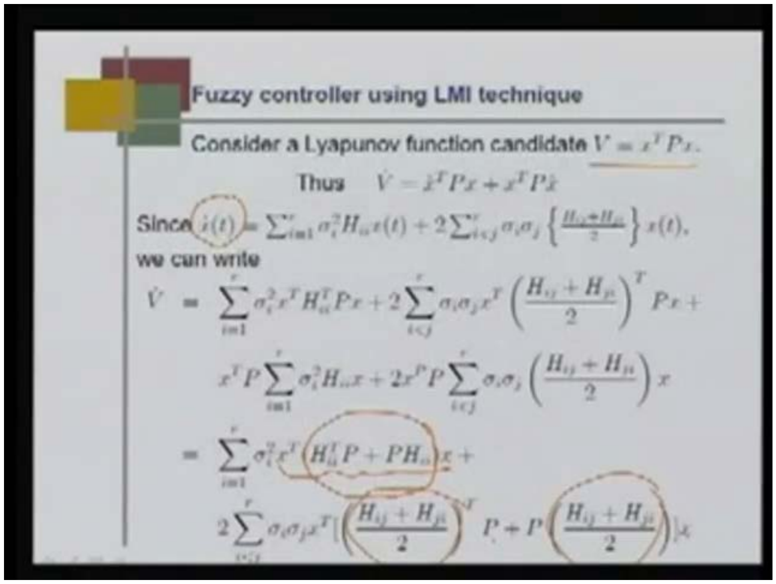
Now, given this T-S fuzzy model, for each rule, for each subsystem, I compute a control action u t which is minus $K_i$ x t.

(Refer Slide Time: 52:43)




If I do that, my overall control action u t is given as minus $sigma_i$ $K_i$ x t, i equal to 1 to r. The individual control action $u_i$ was minus $K_i$ x t for individual and the overall was this one. (Refer Slide Time: 00:53:02 min) Then, the closed loop system is $A_i$ x t plus $B_i$ into u. u is minus this quantity. So minus will come here (Refer Slide Time: 00:53:17 to 00:53:33 min)) and $sigma_i$ $K_i$ x t k, sorry, this is j, so $sigma_j$ $K_j$ x t, j equal to 1 to r. If you put this quantity like this, after simplification, you get this quantity. x dot t is sigmai square $H_{ii}$ where $H_{ii}$ is $A_i$ minus $B_i$ $K_i$ and 2 $sigma_i$ $sigma_j$ $H_{ij}$ plus $H_{ji}$ by 2 x t where $H_{ij}$ is $A_i$ minus $B_i$ $K_j$. We can rewrite this expression in this form.

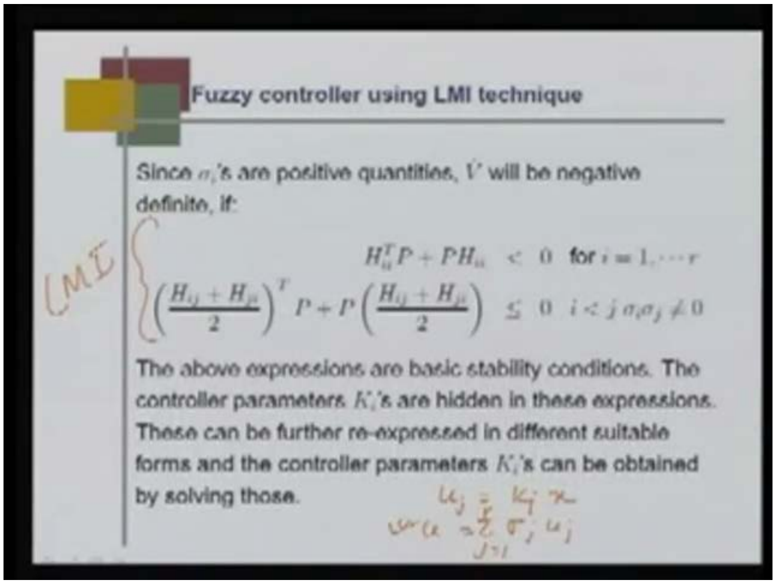
(Refer Slide Time: 54:17)




Once we do that, we take a Lyapunov function because, we have to analyze this particular x dot equal to we can write this term plus this term into x t (Refer Slide Time: 00:54:16min). I want to investigate the stability of this system. The best way to investigate stability of the system, you take a Lyapunov function V equal to x transpose $P_x$. V dot is x dot transpose $P_x$ plus x transpose $P_x$ dot and x dot is given by this expression V. The actual dynamics x dot t is given by this expression (Refer Slide Time: 00:54:40min).

If I replace x dot here, I get V dot finally in this particular format. You see if I put that, I get $\text{sigma}_i$ square x transpose $H_{ii}$ transpose P plus $PH_i$ x. Similarly, here (Refer Slide Time: 00:54:49min) I can write x transpose $H_{ij}$ plus $H_{ji}$ by 2 transpose P, P $H_{ij}$ plus $H_{ji}$ by 2 x. What does it mean? You know that V dot has to be negative definite. That means this quantity has to be negative definite and these quantities also have to be negative definite.

(Refer Slide Time: 55:20)

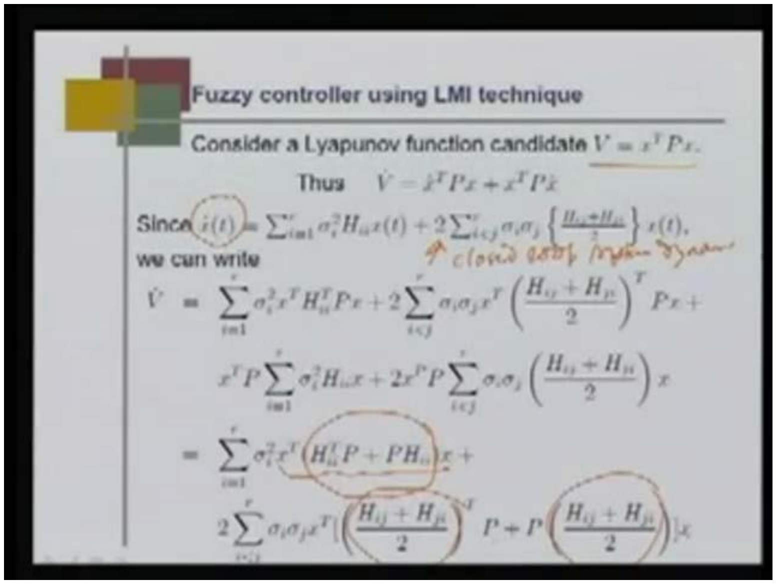



That gives us the condition that $H_{ii}$ transpose P plus P $H_{ii}$ has to be negative definite. Similarly, this $H_{ij}$ plus $H_{ji}$ by 2 transpose P plus P $H_{ij}$ plus $H_{ji}$ by 2 also has to be negative definite. This also can be negative or equal to 0 because, we have already said that this is negative definite. The above expressions are basic stability conditions and these are actually LMI equations – linear matrix inequality. You see that this is a linear matrix inequality equation. The controller parameter $K_i$'s are hidden in these expressions.

These can be further re-expressed in different suitable forms and the controller parameters $K_i$'s can be obtained by solving these expressions. There are various methods – we will not be discussing now. I am just presenting how to stabilize the fuzzy state feedback controller using linear matrix equality.

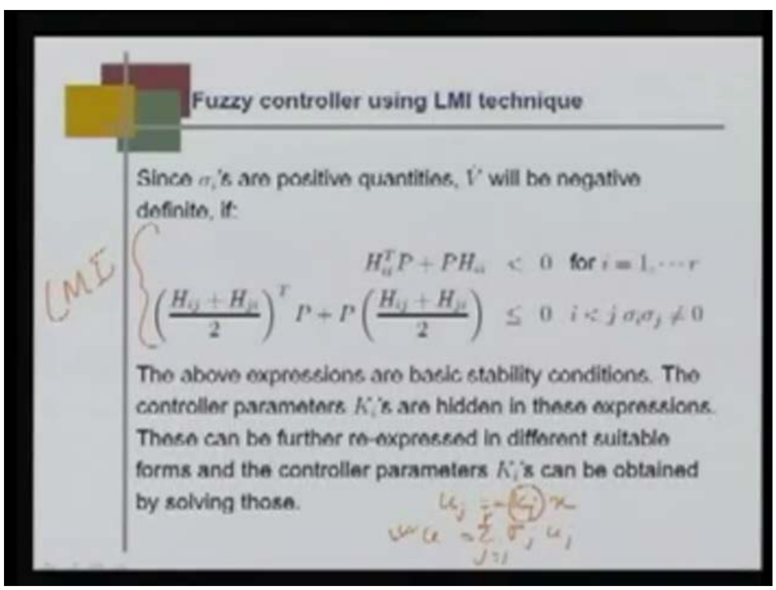
In the beginning, we said the individual control action $u_j$ is $K_j$ x. Then we said u is $\text{sigma}_j$ $u_j$, j equal to 1 to r, where $\text{sigma}_j$ is the normalized membership function associated with rule j. This is my overall control action u.

(Refer Slide Time: 56:57)



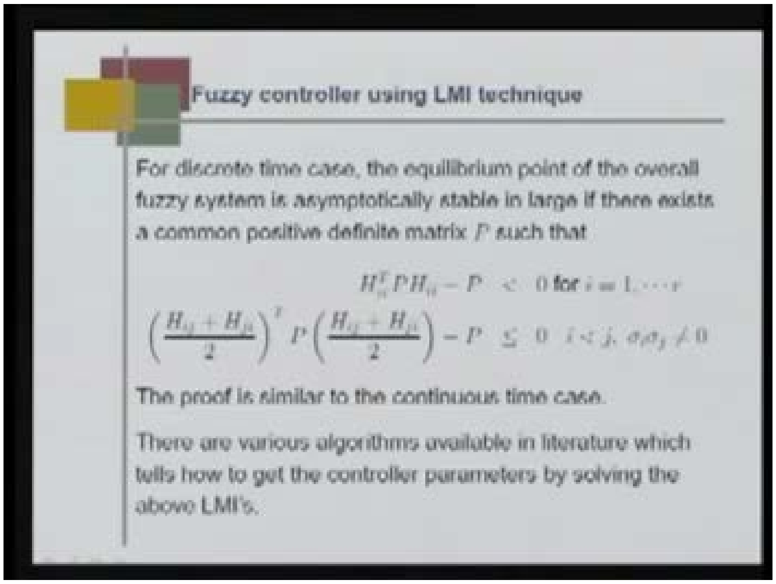

If I give this overall control action, then I said that my overall system dynamics for the closed loop, this is closed loop system dynamics, becomes like this(Refer Slide Time: 00:57:12min)

.(Refer Slide Time: 57:14)



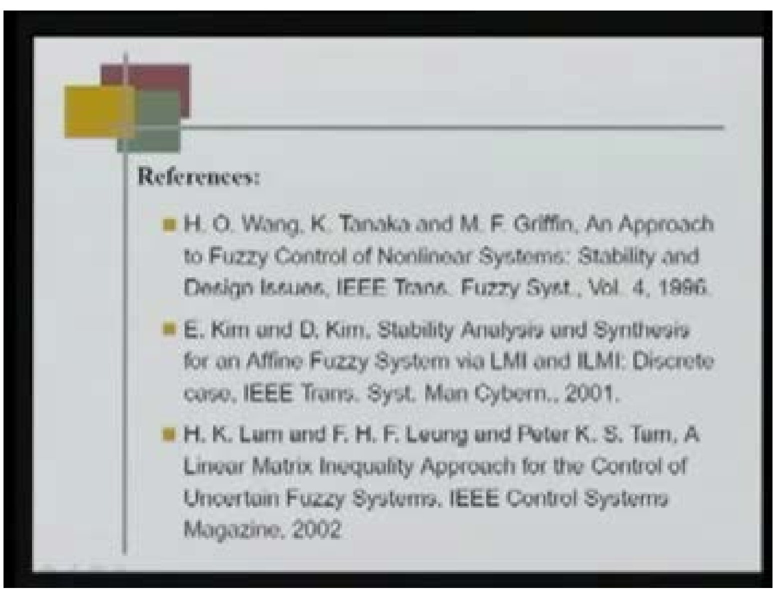

That results in using Lyapunov stability theory to linear matrix inequality equations. If I solve, then I properly find out, finally what should be my $k_j$. This is minus $k_j$.

(Refer Slide Time: 57:30)



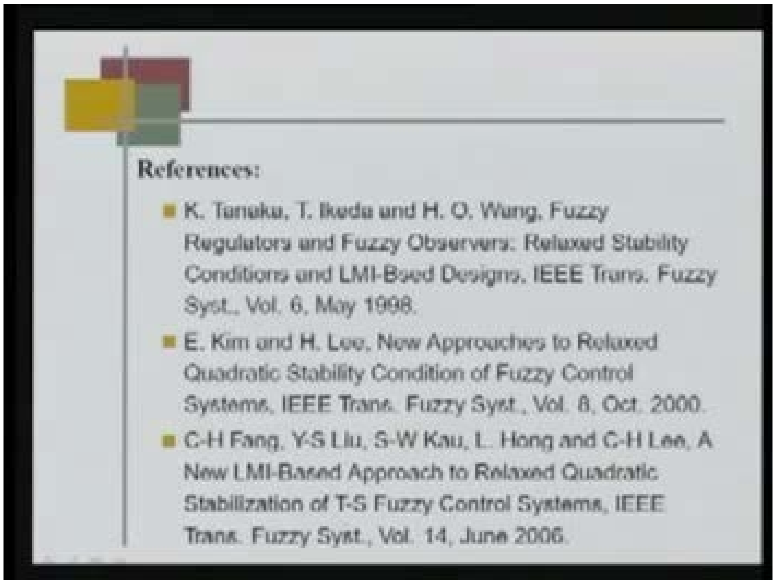

Similarly, for a discrete time case, we can follow the same method. The linear matrix inequality equations would look like this: $H_{ii}$ transpose P $H_{ii}$ minus P less than 0, and this quantity is less than or equal to 0. This gives you an idea of how the T-S fuzzy model based controller can be designed using LMI.

(Refer Slide Time: 58:16)



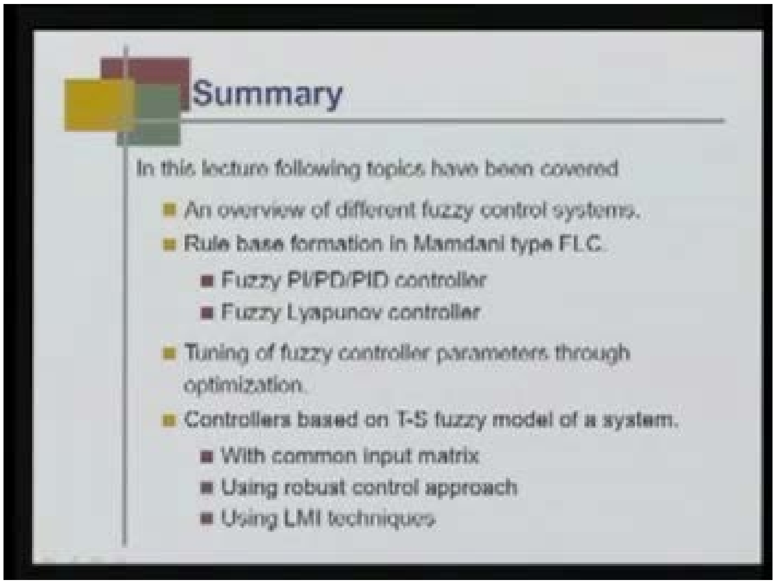
Some of the important works on this are by Wang, Tanaka and Griffin – An Approach to Fuzzy Control of Nonlinear Systems: Stability and Design Issues, IEEE Transactions on Fuzzy Systems, 1996; then Kim and Kim's Stability Analysis and Synthesis for an Affine Fuzzy System via LMI and ILMI: Discrete case, IEEE Transactions on Systems, Man, and Cybernetics, 2001; and Lam, Leung and Tam's A Linear Matrix Inequality Approach for the Control of Uncertain Fuzzy Systems, IEEE Control Systems Magazine, 2002.

(Refer Slide Time: 58:25)




Other works were by Tanaka, Ikeda, Wang – Fuzzy Regulators and Fuzzy Observers: Relaxed Stability Conditions and LMI-based Designs, IEEE Transactions on Fuzzy Systems, May 1998; then Kim and Lee's New Approaches to Relaxed Quadratic Stability Condition of Fuzzy Control Systems, IEEE Transactions on Fuzzy Systems, October 2000; and Fang, Liu, Kau, Hong and Lee's A New LMI-based Approach to Relaxed Quadratic Stabilization of T-S Fuzzy Control Systems, IEEE Transactions on Fuzzy Systems, June 2006.

(Refer Slide Time: 59:01)




Finally, we end this class by saying what we did in this class; we had an overview of different fuzzy control systems. We said that fuzzy rule base can be generated using the concept of PD/PI/PID type of response or using the notion of Lyapunov stability concept; tuning of fuzzy controller parameters through optimization using genetic algorithms, univariate marginal distribution algorithm or differential evaluation – any kind of evolutionary computation approach we can use to optimize the FLC parameters. When we express a nonlinear system using T-S fuzzy model, the controllers can be designed in three different cases, when each subsystem has a common input matrix, or in a generic case, we use the robust control theory to design the controller, or we can also use linear matrix inequality approach to design the fuzzy controller. Thank you very much.