Intelligent Systems and Control

Prof. Laxmidhar Behera

Department of Electrical Engineering

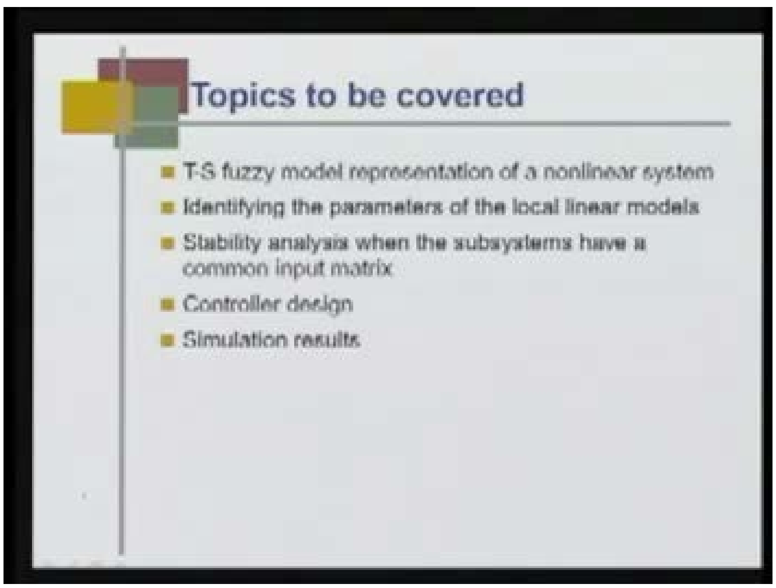
Indian Institute of Technology, Kanpur

Module – 4 Lecture – 5

Controller Design for a T-S Fuzzy Model (Common Input Matrix)

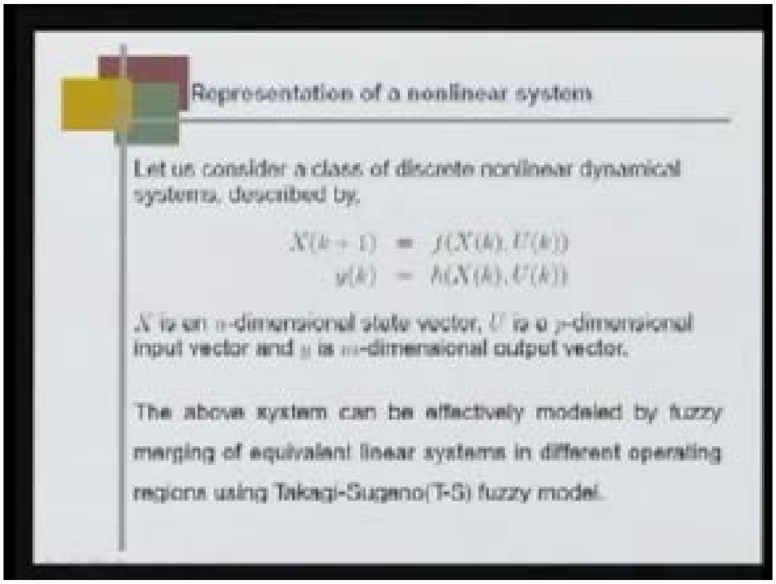
Controller design for a T-S fuzzy model, common input matrix, this is the fifth lecture in this fourth module on fuzzy control. In previous three classes we discussed mainly on Mamdani type of controllers. Today onwards, we will be talking mostly on T-S fuzzy model. What is T-S fuzzy model? What is common input matrix? They may be very new to you or some of you have some understanding but we will learn in detail what they are.

(Refer Slide Time: 01:14)



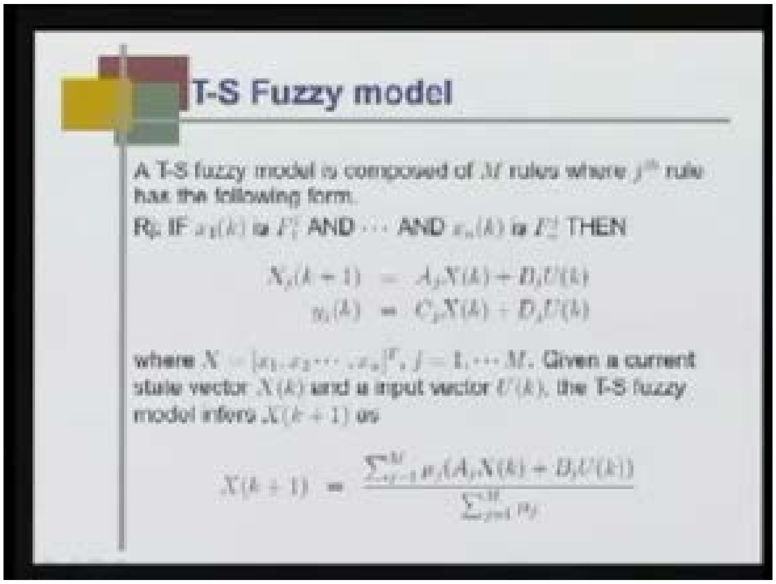
The topics that we will be covering are T-S fuzzy model representation of a nonlinear system; identifying the parameters of local linear models; stability analysis when the subsystems have a common input matrix; controller design; and simulation results.

(Refer Slide Time: 01:36)



Representation of a nonlinear system- if I am looking at a non linear system discrete time, the usual way approach is x is my state space, n dimensional state space X k plus one is f X(k) U(k) and y(k) is h X(k) u(k). This is our normal functional description, f is n dimensional function vector, h is again m dimensional output vector. So I have m output and n states and u is p dimensional input vector, so the above system can be effectively modeled by fuzzy merging of equivalent linear system in different operating regions using Takagi-Sugano (T-S) fuzzy model. What is the meaning of this Takagi-Sugano model fuzzy model of a nonlinear system is when although the system dynamic system non linear overall but locally system is linear. So by fuzzy merging of linear subsystems we can construct of the approximation of the actual nonlinear function of the system. This is the typical T-S fuzzy model.

(Refer Slide Time: 03:36)

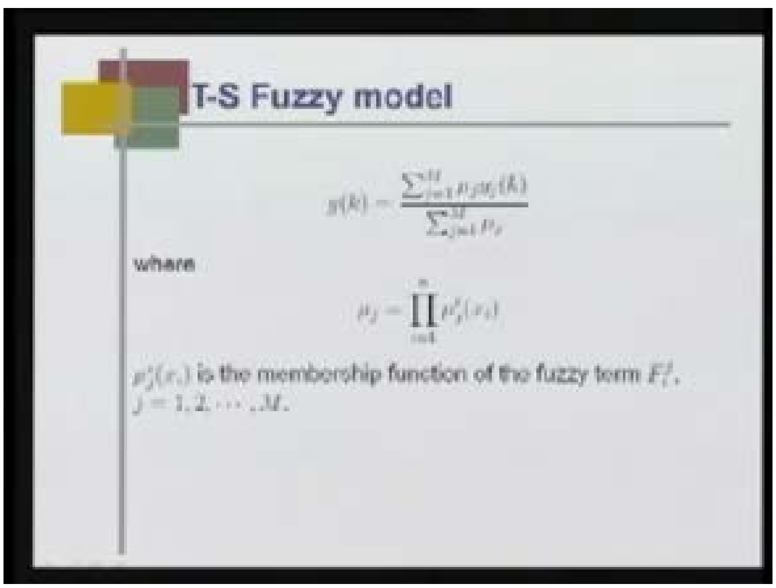


A T-S fuzzy model can composed of m rules where j th rule has following form R_j, the rule j, if $x_1(k)$ is F_1 j and so on this $x_2(k)$ is F_2 j until $x_n(k)$ is F_n j then, that means in the j th for the zone the system is represented by linear system, in this k time we have written, we can also be written in a continuous time, so x j (k plus 1) is A_j X(k) plus sorry this is mistake X (k plus 1) is a A_j X(k) plus B_j U(k) and y(k) is C_j X(k) plus D_j U(k) because X(k plus 1) and y(k) they are of the system states, we cannot put j there j is affiliated to the coefficient, the system matrix A_j, control matrix B_j, this is your output matrix C_j, where x is x_1 x_2 until x_n, n dimensional vector and j equal to 1 to m that means I have m capital M fuzzy rules, given a current state vector X(k) and input vector U(k) in T-S fuzzy model, infers X(k plus 1) as…..

How does it matter? It is because we have to finalize what is X (k plus 1). So X (k plus 1) is fuzzy merging or fuzzy blending of output of each system, output of each system is…. X (k plus 1) is here is A_j X(k) is B_j U(k) this is my j th system dynamics and I multiply with that the corresponding mu_j, the membership function. You know the membership function normally is derived the minimum of the membership function associated with F_1 j to until F_n j or product of that. We can select any one of them, but whatever it is that is mu_j you select to one principle and then you find out the mu_j, the membership function inferred for the rule. See membership function is always given a crisp value x_1 (k) I know what is the membership function F_1 j. Similarly, given a crisp $x_n(k)$, what is the membership function? The membership function associated with $x_n(k)$ is computed from F_n j but the membership function of the rule is mu_j and if mu_j either mean minimum of these membership functions that are compute or you can say mu_1 $x_1(k)$ into mu_2 $x_2(k)$ until mu_n $x_n(k)$ is mu_j. We can do that also. Once I do that and I saw that we form all the rules from 1 to m, divided by the summation of all this membership function mu_j that we have already computed j equal to 1 to m mu_j.

This is how we saw the first T-S fuzzy model. We have m rules this is my j th rule and for j th rule, this is my X(k plus 1), I multiply with corresponding membership function associated with the rule and then I sum such quantities for each rule from 1 to m and then I divide that quantity by summation of all the membership functions, that is mu_j.

(Refer Slide Time: 07:33)



And similarly, this was X(k plus 1), before 1 and y(k) is similarly mu_j y j (k) j equal to 1 to m until j equal to m, mu_j where mu_j in this case we have selected in this paper as a product but you can also take as a mean; it is all up to you the designer, so mu_j i x_i the membership function of fuzzy term F_i j, j equal to 1 to m.

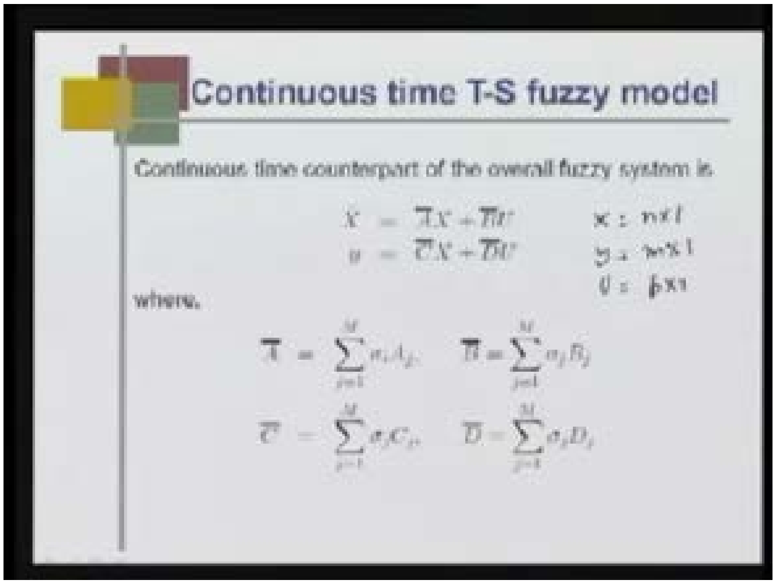
The overall fuzzy system can be simplified into…, overall fuzzy system was this one X(k plus 1) is this quantity, this is my fuzzy dynamics representation of the nonlinear system in terms of fuzzy dynamics. This fuzzy dynamics can be written in terms of X(k plus 1) is A bar X(k) plus B bar U(k) where A bar is sigma j $A_{j,}$ j equal to 1 to m where sigma j is mu_j upon this, you can easily do that, that is very simple if I rewrite this equation I can easily write down this, this is simply you see that, I can write this one as A_j so I write this one as A bar X(k) B bar U(k) where this A bar this is at sigma mu_j by this quantity.

I can define sigma j equal to one two and mu_j this quantity. So all that the mu_j I am dividing by this total summation which is this and the summation is j is equal to 1 to m A_j. This quantity is represented as sigma j, so I am writing A bar is j equal to 1 to m sigma j A_j. This is what exactly we have done.

X (k plus 1) is A bar X(k) plus B bar U(k) where A bar is sigma j A_j and we defined here sigma j A_j equal to 1 to m. Similarly, B bar is sigma j B_j, j equal to 1 to m, C bar is sigma j $C_{j,}$ j equal to 1 to m, D bar is sigma j D_j where j equal to 1 to m and I already defined what is sigma j before, mu_j upon total summation and you must recognize that sigma j from 1 to m, this is most important this is always true, the total sigma j has to be 1. The overall system is nonlinear, so you can easily see that this is very convenient form although I am representing this in a state space format, it appears to be linear but it is not linear because A bar is a function of sigma j which you see here and sigma j is the function of X(k). Sigma j is a mu_j and mu_j comes from x j this is a function of X(k).

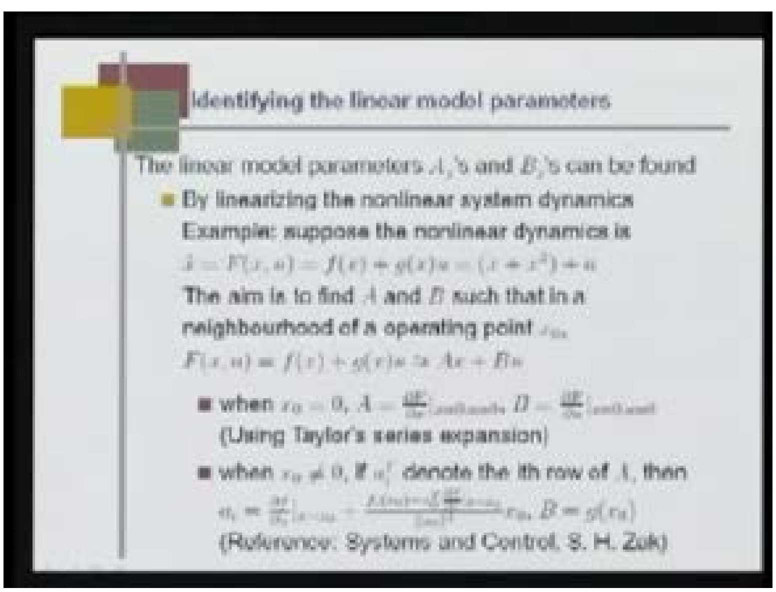
Just like we did for discrete times similarly also we can say for continuous time, instead of X(k plus 1), I will write X dot is A bar X plus B bar U where again X is n into 1 dimensional vector and y is your whatever we wrote here 1 into one dimensional vector and u is p into one dimensional vector.

(Refer Slide Time: 12:33)



The whole thing again is similar A bar is sigma j sigma j A_j and B bar is sigma j $B_{j,}$ C bar is sigma j C_j and so on. What we said is until now we just said what T-S fuzzy model is. Now given an actual non linear plant dynamics can I directly write what is T-S fuzzy model? It is actually simple. The linear model parameter, how do we find out for each rule what should be my individual linear system? So to find out what is individual linear system one of the methods is linear system.

(Refer Slide Time: 12:33)



The linear model parameter A_{js} and B_{js} can be formed by linearizing the non linear system dynamics. The simple example is, suppose the linear dynamics given as x dot is F (x, u) which f (x) plus g (x) into u, this one form and that can be written as x plus x square plus u. This is your f (x) and g (x) is one here. The aim is to find A and B is that in a neighborhood of an operating point x_0, F(x, u) equal to f (x) plus g(x)u which is Ax plus Bu. How do you find out that? When x_0 equal to 0, A is doe F and doe x, x equal to 0, u equal to 0 and B equal to doe F by doe a x equal to 0 u equal to 0 using Taylor series expansion.

This thing we already know, given this non linear function f, how can we find out this A and B? That is by simply differentiating the f by x. When x_0 equal to 0 and if A_i transpose the i th row of A then A_i is doe f upon doe x equal to f_0 plus f_i is not minus x_0 doe f and doe x like x equal to x_0 upon x_0 whole square into x_0 B equal to $g(x_0)$. The reference you can find out this reference system is in control, given written by Jack. There are many other classical text books you can follow, how to linearize on non linear system. I think we also covered this linear in the class.

Thus two rules of the T-S fuzzy model, using T-S fuzzy model our system was x dot was x plus x square plus u. You can easily see that x plus x square plus u. That is our system. Rule one if f equal to 0 x_0 is so I differentiae this with respect to x, so I get 2x plus 1 at x equal to 0. If I differentiate this quantity 2 x plus 1 at x equal to 0, so this will give you 1 x equal to 0 the value is 1, into x so x plus u because doe f by doe x at x equal to 0 into x plus, actually delta x but since the operating point is 0 so delta x is same as x. (Refer Slide Time: 16:43) is x plus doe f upon doe u. Since the coefficient here is 1, so doe f upon doe u is 1 and into u. So A_1 is 1 and B_1 is 1.

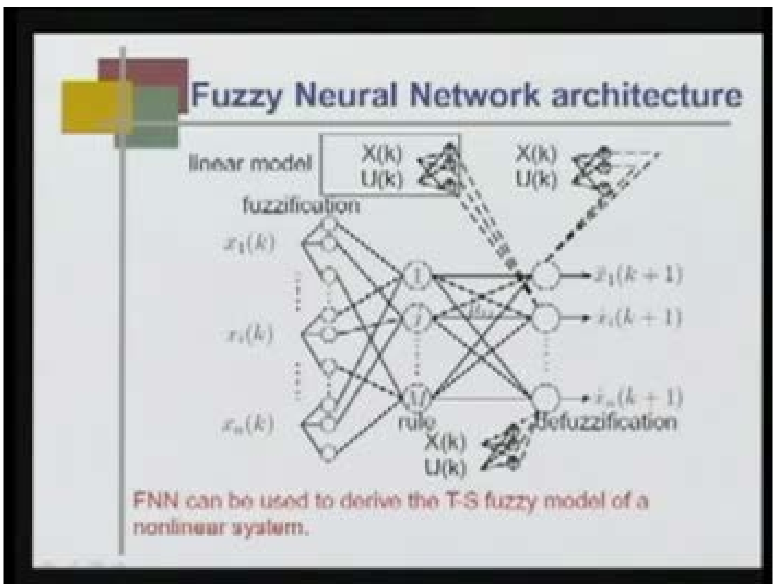
Similarly, if x equal to 1 then we implement this second one because when x_0 equal to 0 then we implement this rule and while implementing that the x_0 is 2 x plus u and A_2 that means A_2 is 2 and B_2 is 1. This is simply a scalar system, scalar differential equation. Similarly, this same thing can be also duplicated to the vector differential equation. The linear model parameters A_j s and B_j s can also be identified. This is the first method by simply linearizing. If I know a non linear system dynamics and I think that is an exact model the best way is not to waste system against identification, simply linearize the system around various operating zones and define the fuzzy partitioning and do the linearization. But this also can be identified using a fuzzy neural network.

From the input output data of the system, I have a given system I generate input output data train a fuzzy neural network. When using a fuzzy neural network the elements of A_j and B_j are the weights of the neural network.

Least square cost function is used to find the proper weights; weights are updated using the standard gradient descent algorithm as well.

So this is simple, what is that? These are my states $x_1(k)$ $x_2(k)$ then k until x_i. So these are my states and I have m rules.

(Refer Slide Time: 18:33)



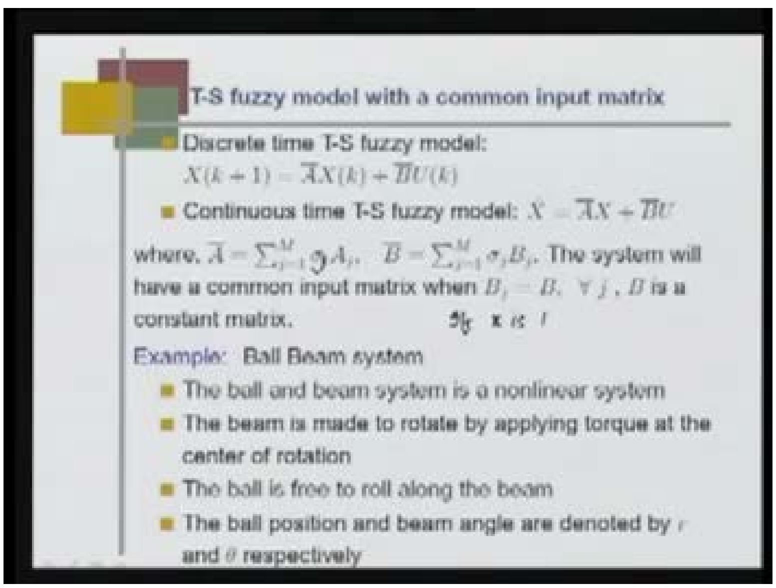
And corresponding to each rule, I have a fuzzy membership linear neural network, whose input is X(k) and U(k) and output is X(k plus 1). Similarly, corresponding to each rule I have a linear neural network and finally, we do the defuzzification to compute what is the actual X(k plus 1). This fuzzy neural network can also be used to derive the T-S fuzzy model of a nonlinear system. We identify that and this is my, if these neural network weights after identification they become difficult. For example this one is the linear model for the rule one, this one the linear model for rule j and this one the linear model for rule m.

Earlier in the beginning of the class we said about T-S fuzzy model, I hope that you understood by this discussion. Now we also talked in the beginning common input matrix. What is this common input matrix?

Now you see the discrete time T-S fuzzy model was given by X(k plus 1) A bar X(k) plus B bar U(k) and continuous time of T-S fuzzy model x_0 is A bar X plus B bar U, where A bar is sigma j A_j summation over all rules. Similarly, B bar is sigma j B_j summation over all rules. But for this system you will have a common input matrix when B_j is B for all j, where B is a constant matrix.

If I can define the system dynamics in terms of a common input matrix B, B is constant matrix and if I do that, then that gives us certain advantage.

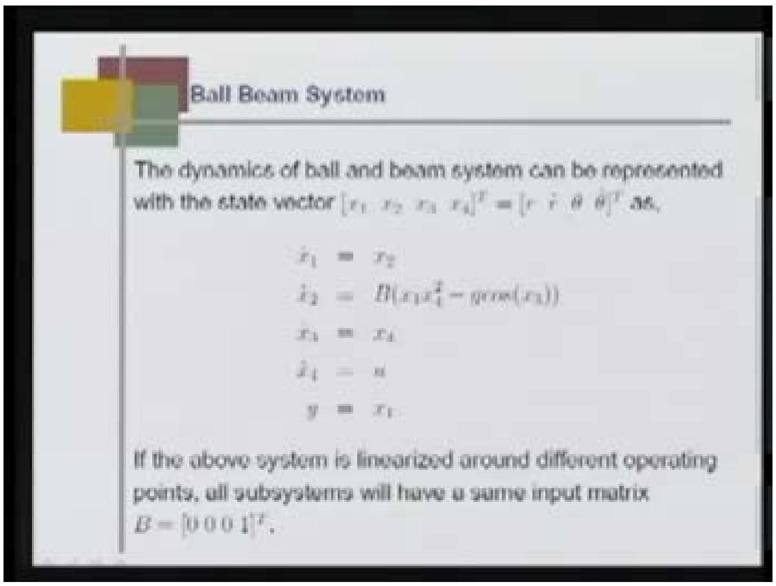
(Refer Slide Time: 21:12)



What I am trying to say is that for if x is some fuzzy zone then if my state vector in a specific fuzzy zone j then all that I am saying is that my x dot is A_j x plus B u, not B_j but B u and where B is a constant matrix. This is called common input and B matrix is same for all the rules. When this is done then such a system is known as T-S fuzzy model then common input matrix, input matrix is common for all linear subsystem. Why we are interested in such a system? Let us take a ball beam system.

In a ball beam system, the ball and the beam system is a nonlinear system. The beam is made to rotate by applying torque at the center of rotation. The ball is free to roll along the beam. The ball position and beam angle are denoted by r and theta respectively.

(Refer Slide Time: 22:39)



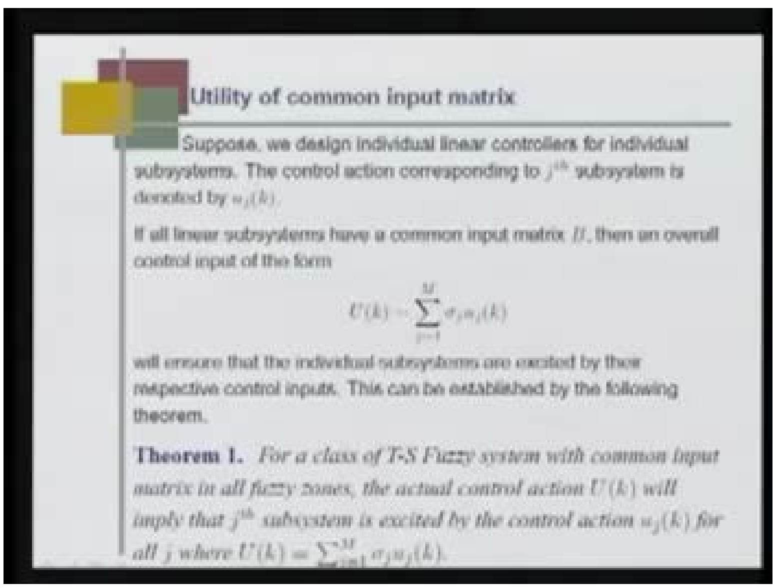
The dynamics of ball and beam system can be represented with a state vector x_1 x_2 x_3 x_4 as r r dot and theta theta dot as x_1 dot is x_2; x_2 dot is B x_1 x_4 whole square minus g x_3; x_3 dot is x_4; x_4 dot is u; y is x_1. So single output and single input, if the above system is linearized around the different operating points all sub systems will have a same input matrix which is B is 0 0 0 1, this is important.

While making B equal to 0 0 0 1, sorry, this linearization of the system dynamics will always lead to B matrix for all linear subsystems the mean matrix is always 0 0 0 1. This is a practical example, where exist a fuzzy control system for which for all linear subsystems the control matrix is 0 0 0 1, so this is common.

Utility of common input matrix, what is the advantage of common input matrix? Of force when B s are same that means, there must be some advantage because system is now simpler.

Suppose we design individual linear controller for individual subsystem, I have that to say because now I have n linear subsystems. Controlling the complete nonlinear system is they involve controlling M subsystems. Now I am saying you have the freedom to design because linear subsystem means, for any linear system we know where we have adequate technology of tools, adequate methodologies for designing controllers. What we are doing here is that let us design control action for each subsystem. The control action corresponding to the j subsystem is denoted by $u_j(k)$ that stabilizes that subsystem.

(Refer Slide Time: 24:48)



If all linear subsystems have a common input matrix B then an overall control input of the form U(k) is sigma j $u_j(k)$, this will ensure that individual subsystems are excited by the respective control system. This can be established by following theorem.

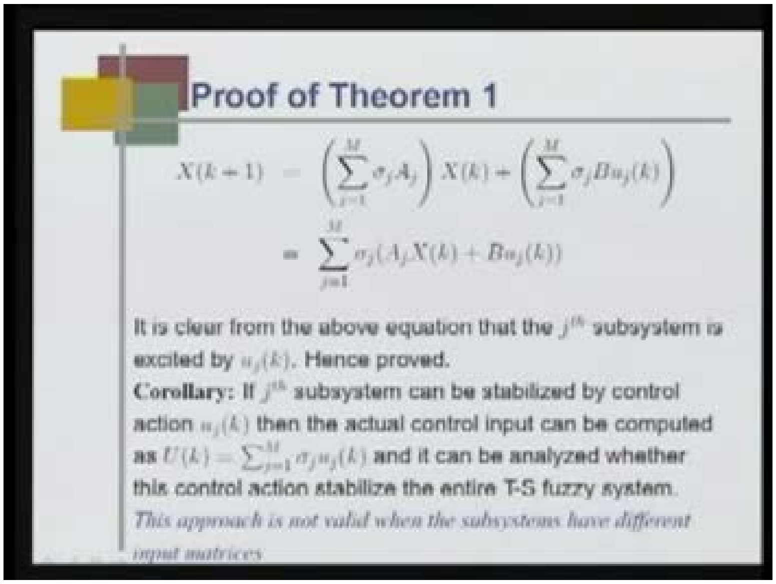
What I am trying to say is this is my actual plant, I give u and I get x the response right through the plant. Now I have, these are all the apparent subsystems linear subsystems and fuzzy merging of this linear subsystem gives us the actual plant dynamics. Controlling this u because I give to the plant of only u. So that is obviously my input to the plant to this system of B u but instead what I have done, I have stabilized this by giving a input u_j, but I cannot give this u_j to the plant, plant is given u.

But if I compute this term u in terms of u_j using this formula, it says that if I have a common input matrix for all the subsystems then such a control action U(k) which is written in terms of individual control action u_j, implies that individual subsystems are excited by $u_j(k)$ So this is the theorem for a class of T-S fuzzy system with common input matrix in all fuzzy zones, the actual control action U(k) will imply the j th subsystem is excited by the control action $u_j(k)$ for all j, where U(k) is the fuzzy blending of all individual control action. You must recognize this u_j is actually fake. We do not apply to any actual system. This u_j is simply a control action that is if I give u the actual plant which implies this plant is experiencing a control action not u but $u_j(k)$, this is important.

If I am giving a control action U(k) to the actual plant, individual plant is not being excited by U(k) rather $u_j(k)$. This is very important, only if all the subsystems of the T-S fuzzy model are having a common input matrix, otherwise not. This is very important. The fuzzy dynamics of common input matrix be represented as X(k plus 1) is that we have already shown, the sigma A_j j equal to 1 to m which is A bar; similarly, sigma B not B_j because common input matrix j equal to 1 to m, so we expanded that. Now you can write this one, I can clock this j equal to 1 to m.

Sigma j A_j X(k) plus B u U(k) this is the total thing. This is my one term and you can see this term is actually 1, sigma j j equal to 1 to m and hence this is simply B U(k) where B is the common for all linear subsystems.

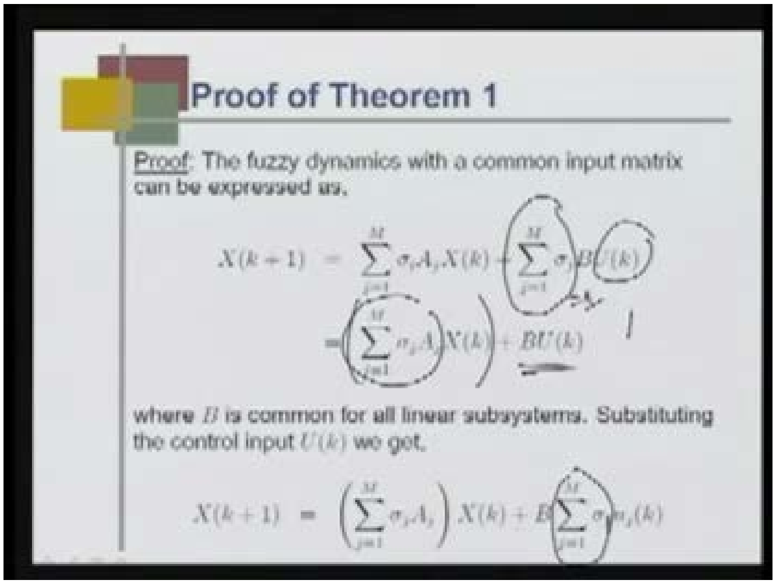
(Refer Slide Time: 27:48)



Substituting of a control input u(k), what is U(k) now? U(k) is j equal to 1 to m, sigma j u_j (k) and here I can separate this term, so j equal to 1 to m, sigma j A_j and X(k). This is my A bar and we started with, we gave a input U(k) which is summation of this individual thing; now we will rewrite this equation then you will see that X(k plus 1) is sigma j A_j X(k) plus here I can bring this quantity to this side and B inside, simply B I put inside because this is anyway in common, so B inside and u_j(k). So doing that what I am trying to do is that sure, this is actually gain, this quantity is 1.

Now, what we are doing here is that now we have this total thing. That is I can now rewrite j equal to 1 to m sigma j and A_j X(k) here and sigma j is common here and B u_j(k). I can rewrite that. What is meaning of this? The meaning of this is that, this is a fuzzy combination of A_j X(k) plus B u_j(k). Individual system is actually excited by u_j when actually I have a excited the global system by U(k).

(Refer Slide Time: 30:48)

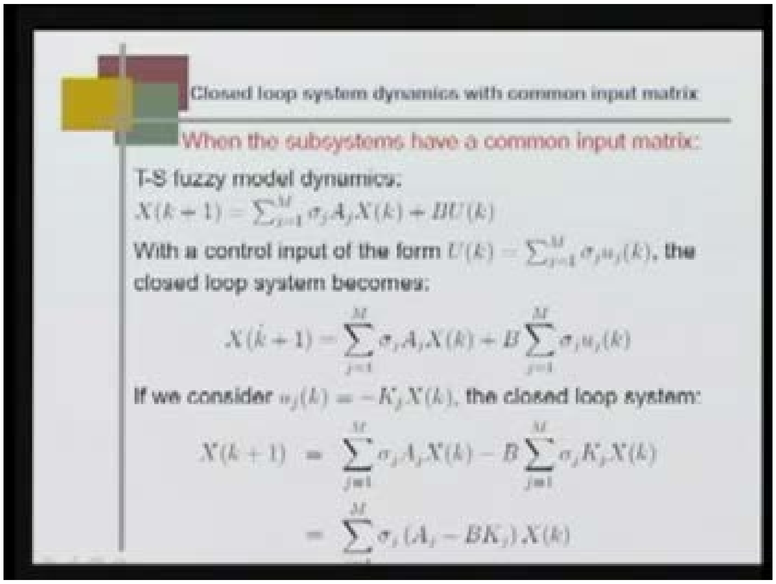


What I showed in this theorem is that if I give U(k) to my actual system plant this implies my linear subsystems being excited by u_j, that is the meaning of this of this method.

It is clear from the above equation that j th subsystem is excited by $u_j(k)$ and it is proved. The corollary: if j th subsystem can be stabilized by control action $u_j(k)$ then the actual control input can be computed as U(k) is this sigma j equal to 1 to m sigma j $u_j(k)$ and it can be analyzed whether the control action stabilize the entire T-S fuzzy system.

This approach is not valid when the subsystems have different input matrices. This is very important.

(Refer Slide Time: 31:48)

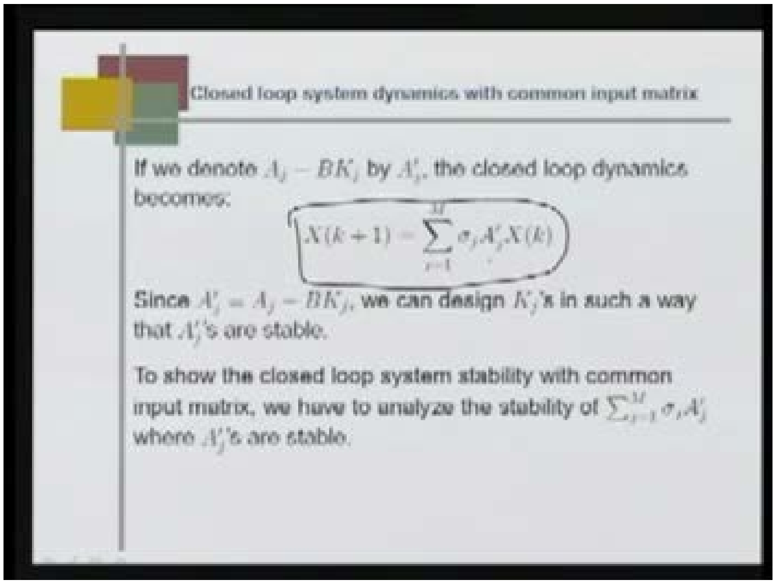


When the subsystems have a common input matrix, T-S fuzzy model dynamics is used. So closed loop system dynamics with common input matrix, I will explain to you what the meaning common input matrix is.

Common input matrix means that if the actual plant is excited by capital U(k), then the actual subsystems are excited by the individual subsystem as usage, that gives us a fairly good method or good tool for us to design linear controllers. Now for such common input matrix will be the closed loop dynamics? T-S fuzzy model dynamics is sigma j A_j X(k) which I have already told you, this quantity is A bar plus B U(k), so when U(k) is this type we have already said, the closed loop system dynamics is given by this. Simply I replaced U(k) by this particular term, this is input.

Now assume that u_j (k) is minus K_j X(k) such that j th linear subsystem is stable. This I am assuming such that j th system subsystem is stable. If j th subsystem is stable then the closed loop system dynamics is minus, this will retain, this quantity is same and this quantity is here because this minus is coming here, u_j (k) minus k, minus term is written here and sigma j, this quantity is retained here, instead of uj(k) minus K_j X(k). That gives you, if you look at that that gives you the final value here because this B K_j I can multiply and then it becomes A_j minus B K_j sigma j is common and this side X(k) and this side X(k), so this is my closed loop dynamics.

(Refer Slide Time: 34:48)



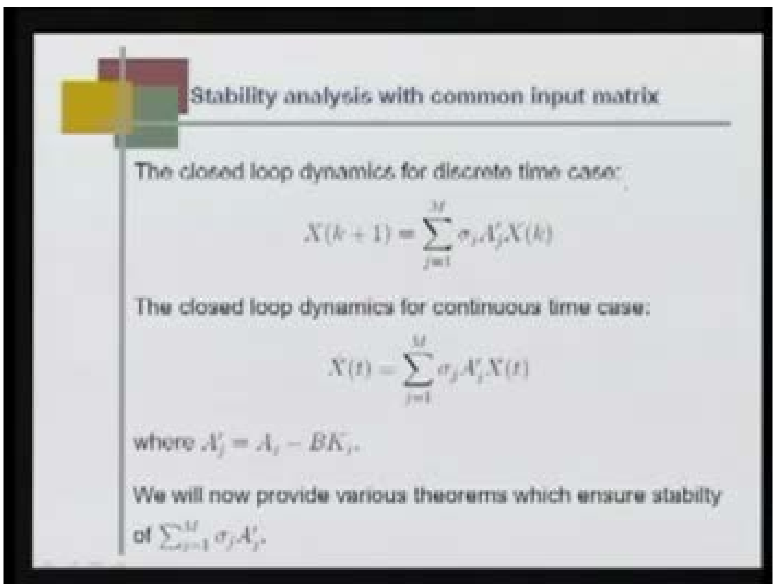
What implies that common input matrix mix our closed loop system dynamics is very simple. This is very important. If you denote A_j minus B K_j by A_j it is closed loop dynamics becomes, this particular thing where A_j dash minus B K_j we can K_j in such a way that A_j's are stable.

Now the point is that we have designed we can find U(k) in such a way A_j dashes are stable, but now X(k plus 1) this is the closed loop dynamics, my stability simply depends I know already A_j dashes are stable, so now given A_j dashes are stable, these matrices are stable, can I say that if I write this one as A bar X(k) where A bar is simply j equal to 1 to m sigma_j A_j dash, since dash is there I will put dash also here, so a dash bar if I say can I say if A_j dash's are stable then A bar is stable.

You may say, it should be but it is not, if A_j dash stable does not imply directly A bar is stable. We will find in this lecture what are the conditions for which if A_j dashes are stable then A bar is stable. That is the point. To show the closed loop system stability it comes with common input matrix to analyze the stability of the quantity and stability of the quantity where A_j's are stable. This part I will not talk, this simply says that when the subsystems have different input matrices then the closed loops are different then if I find out I designed $u_l(k)$ the individual system minus K_l X(k) then X(k plus 1) is given by this particular term and you see that here we cannot represent the way we represented, the reason is the above equation are shows that cross term in the closed loop dynamics when j is not equal to l. Because of this cross term that is coming here because it had it been B it is simple but it is B_j. this makes our life little tough.

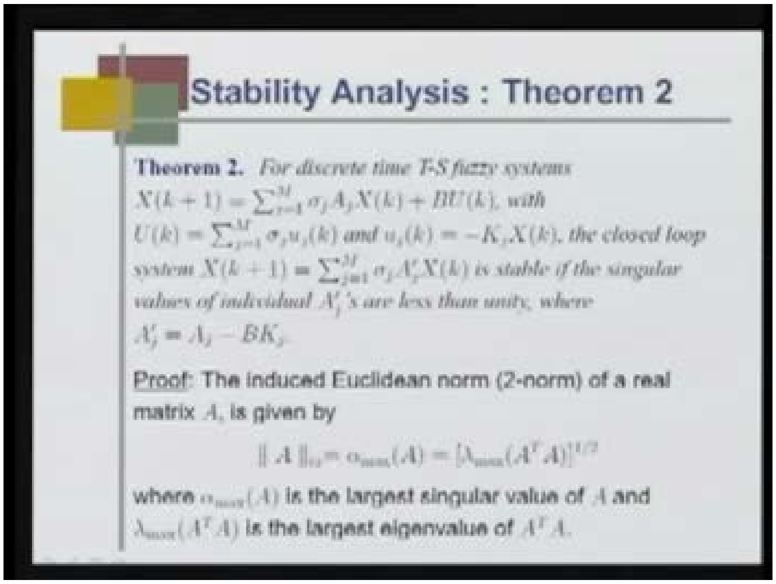
Now we will go to the actual stability analysis. The stability analysis is common input matrix that we have now simplified.

(Refer Slide Time: 37:00)



The closed loop dynamics for discrete time case is X(k plus 1) plus sigma_j A_j dash X(k) and where A_j dash is simply A minus B K_j. So K_j is associated state feedback for j th linear subsystem. The closed loop dynamics for continuous time case similarly, exactly for this is discrete time, this is the closed loop dynamics of the continuous time where A_j dash is A_j minus B K. We will now provide various theorems which we ensure stability of this quantity, that is we would like to say because now, this can be written as A bar X(k) and this can be written as A bar x(t). So if A_j dash is stable I can say A bar is stable, this is the theorem.

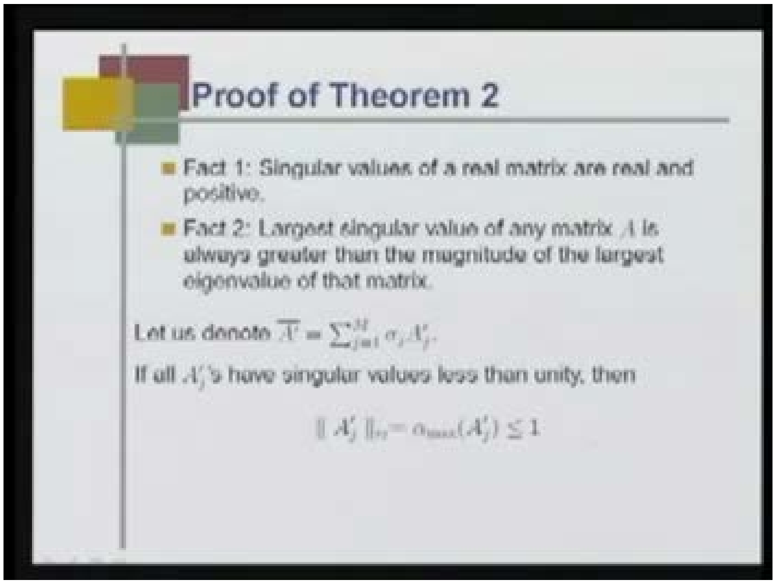
(Refer Slide Time: 39:00)



Stability analysis theorem 2- for discrete time T-S for the system k plus 1, this is my T-S fuzzy system common input matrix, if U(k) is given in terms of fuzzy blending of individual control action and the control action is taken in such a way the individual subsystems are stable then the closed loop system X(k plus 1) which is given by this stable if the singular values of the individual A_j dash or less than unity, where A_j dash is A_j minus U(k).

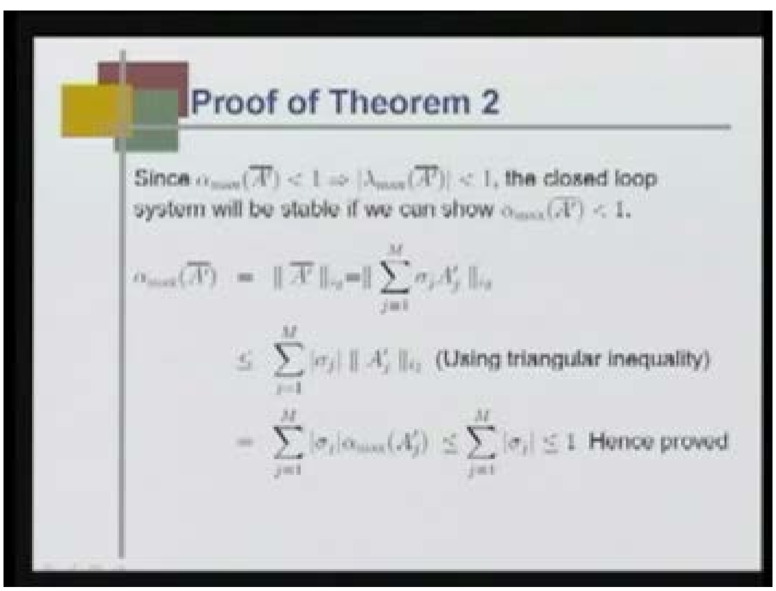
Here the proof is the induced Euclidean norm, second norm of a real matrix A is given by A second norm induce norm is the maximum singular value of a matrix and what is that the maximum singular value a matrix is computed as the maximum A value A transpose A and take the square root. I compute the maximum a value of A transpose A and take the square root then I find maximum singular value of A where $alpha_{max}$ is the largest singular value of A and lambda transpose A is the largest eigen value of A transpose A.

(Refer Slide Time: 40:23)



Fact 1: Singular values of a real matrix are real and positive, this is very important. Fact 2: Largest singular value of any matrix A is always greater than the magnitude of the largest eigen value matrix, this is very important. I hope that you know this from the matrix algebra. Now, let us denote A bar dash is j equal to 1 to sigma$_j$ A$_j$ dash if all A$_j$ dash has singular value less than unity then obviously the induce norm of A$_j$ dash which is maximum value of A$_j$ dash is less than 1.

(Refer Slide Time: 41:23)



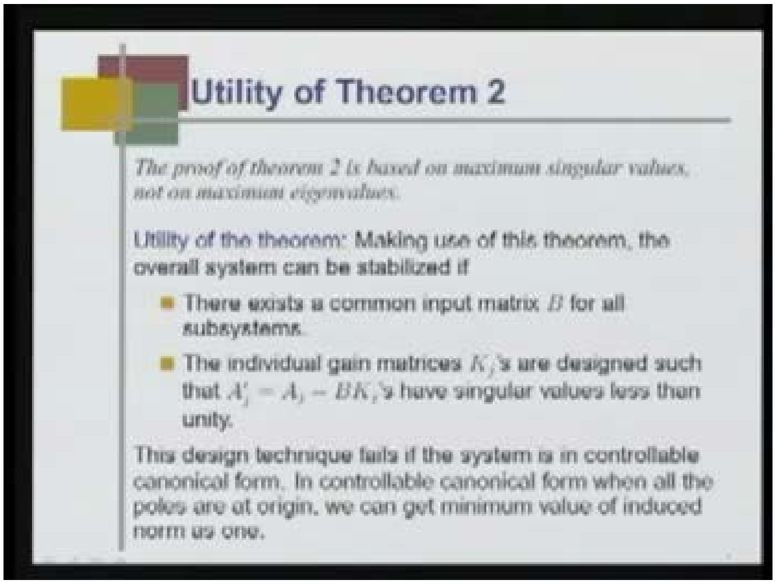
Since, $alpha_{max}$ since largest singular value is given because we are saying in the theorem if the largest singular value is less than 1 of all individual, if largest singular value individual matrix is less than unity, less than 1, then the system is stable. That is what we are saying, so you see that. Now let me find out what is the largest singular value of A bar. The largest singular value of A bar is induce norm of it, which is induce norm A bar dash is sigma dash A_j dash A bar is simply this and then you see that using triangular inequality, this is summation or this is summation of two products means by triangular inequality it is always less than equal to summation of individual absolute product, that is $sigma_j$ A_j norm. Now I know this A_j norm is represented by $alpha_{max}$ A_j dash.

And I know already that this quantity alpha max of A_j dash always less than 1, I am assuming that. I know that let us say all A_j in this T-S fuzzy model, they have maximum singular value less than 1 because of that I can write this because this is less than 1, so this is less than equal to $sigma_j$ and you know that j equal to 1 to m $sigma_j$ is 1, so this is less than 1.

So $alpha_{max}$ the largest singular value A dash is also less than 1 and we have already said the largest singular value is always which we said earlier, we see that, largest singular value of any matrix always greater than the magnitude of the largest eigen value of that matrix.

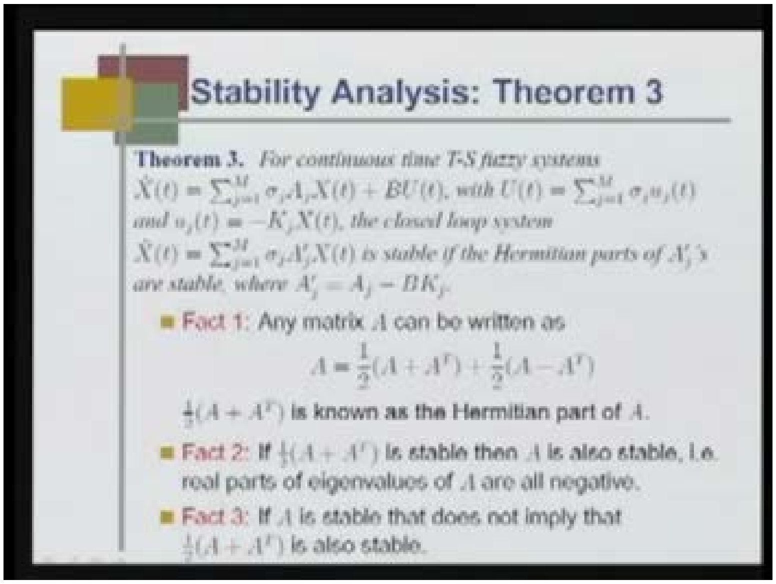
That is, since we proved the largest singular value of A bar is less than 1, means largest eigen value of A bar is also less than 1 and for discrete time case the largest eigen value the magnitude should be less than 1. It should be unity circle. You know already that given a discrete time system the poles should be within the unity circle. What we said in this theorem that if my individual subsystem in the closed loop form A_j dash they have singular value less than 1, the maximum singular value, then the overall closed loop system matrix A bar also will have the singular value less than 1, implying that this system is stable.

(Refer Slide Time: 44:50)



Now utility of theorem two the proof of theorem 2- is based on maximum singular value not on maximum eigen values. Utility of the theorem, making use of theorem, the overall system stabilized if there exists common input matrix B for all subsystems; the individual gain matrix K_j's are designed such that A_j dash which I have already told is A_j minus B K_j have singular value less than unity. This design technique fails if the system is in controllable canonical form. In controllable canonical form, when all the poles are at origin, we can get minimum value of induced norm as 1.

(Refer Slide Time: 44:50)



Stability analysis: theorem 3- Now earlier we have talked about discrete time, now we talking about continuous time. For continuous time T-S fuzzy system x dot t is this quantity which is our T-S fuzzy model, this is our normal fuzzy model, with again our control x and u(t) is sigma j.

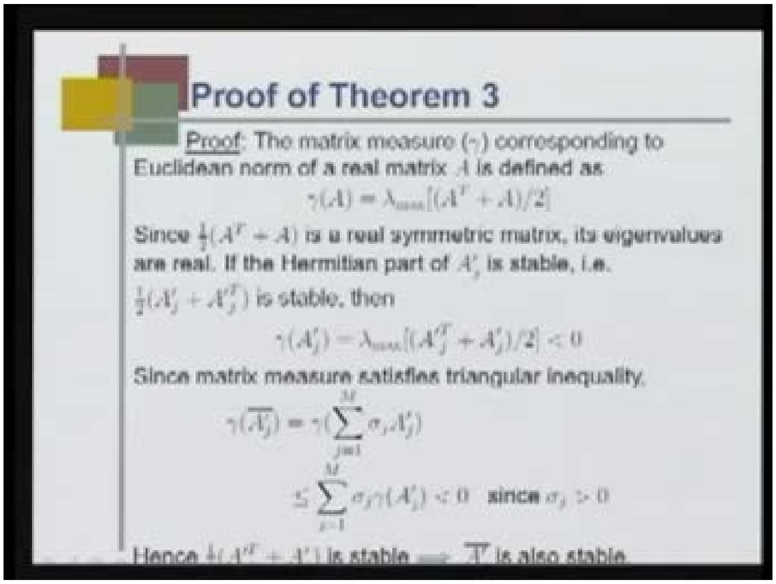
$u_j(t)$ is minus K_j X(t) the closed loop system x dot t is stable, the Hermitian part of A_j dashes are stable where A_j dash is A_j minus B K_j.

Fact 1- any matrix A can be written as A equal to half A plus A transpose plus half A minus A transpose, half A plus A transpose is called Hermitian part of A. This is the Hermitian part. You can easily see that, this total multiplication, this will cancel out so this is 2 A by A, 2A by 2 is A, but this actually is Hermitian part known about non Hermitian part.

Fact 2- If half A plus A transpose is stable then A is also stable. That is real parts of eigen values of A are all negative.

Fact 3- if A is stable that does not imply that half of A A transpose is also stable. The reverse is not true. If this is stable then I can say A is stable but if A is stable we cannot say A plus A transpose is still also stable.

(Refer Slide Time: 47:16)

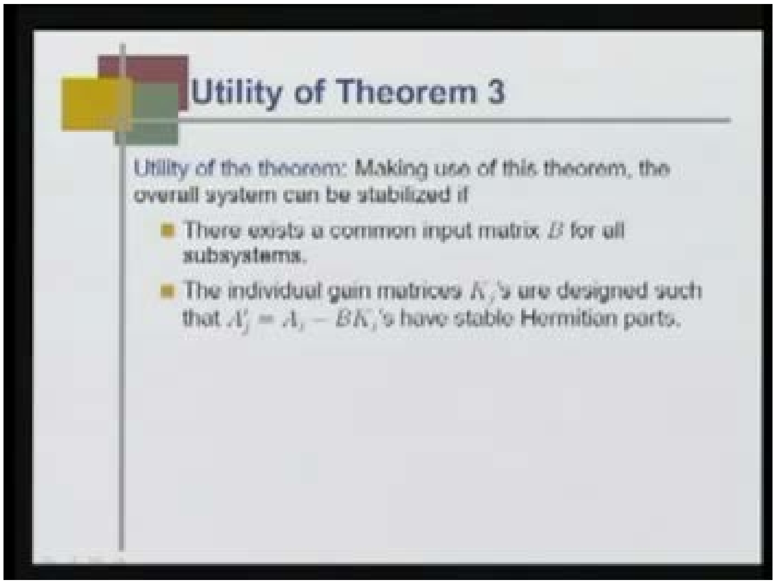


Proof: the matrix measured gamma corresponding to the Euclidean norm of a real matrix is defined as that maximum eigen value of the Hermitian part of A. Since A transpose A by 2 the Hermitian part of A real symmetric matrix if eigen values are real. So if the Hermitian part of A_j is stable that is half A_j dash plus A_j dash transpose is stable then the matrix measure of A_j dash which is lambda maximum of this matrix is always less than 0, the maximum eigen value.

Since the matrix measure satisfies triangular inequality, I can write the measure of overall matrix A_j bar is gamma into summation j equal to 1 to m this is the individual A_j dash. Now this can be written in terms of, this is less than equal to I can take this using gamma inside using triangular instability summation, so the individual is sigma_j gamma A_j dash, because matrix measure satisfies the triangular inequality. Using that theorem you can say that this is less than this and since I know that the matrix measure of this is the maximum eigenvalue of this quantity which is always less than 0.

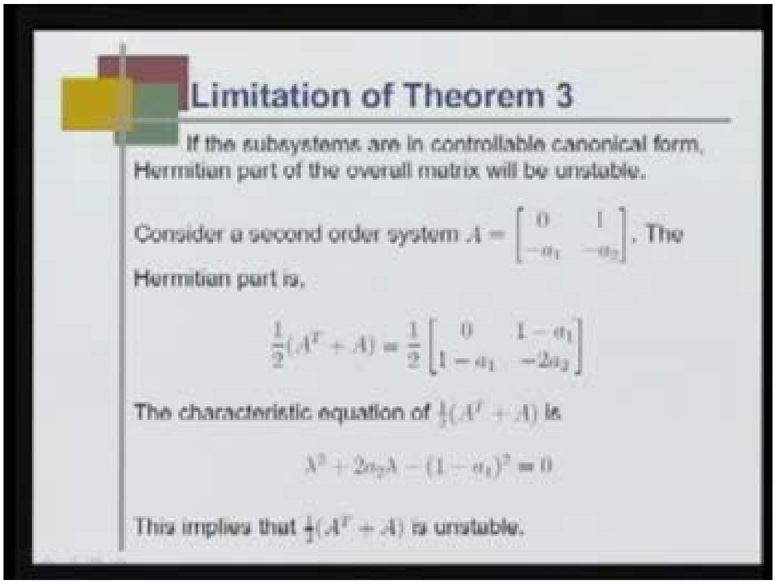
So sigma_j is always greater than 0 and this is the negative quantity and this is the positive quantity, the summation will be always negative. Hence the proof is if this is stable A_j bar is also stable. So the second theorem says for linear subsystem, for a linear continuous type system, the overall fuzzy blending of the system should be stable, provided the Hermitian part of the individual subsystems. This is my Hermitian part of the individual subsystem that is stable.

(Refer Slide Time: 49:41)



Utility of theorem: making use of this theorem the overall system stabilized if there exists a common input matrix B for all subsystems and the individual gain matrix K_j's are designed such that this has a stable Hermitian parts. We can always design this because I can always design such that this is a stable Hermitian part, is not a difficult thing.

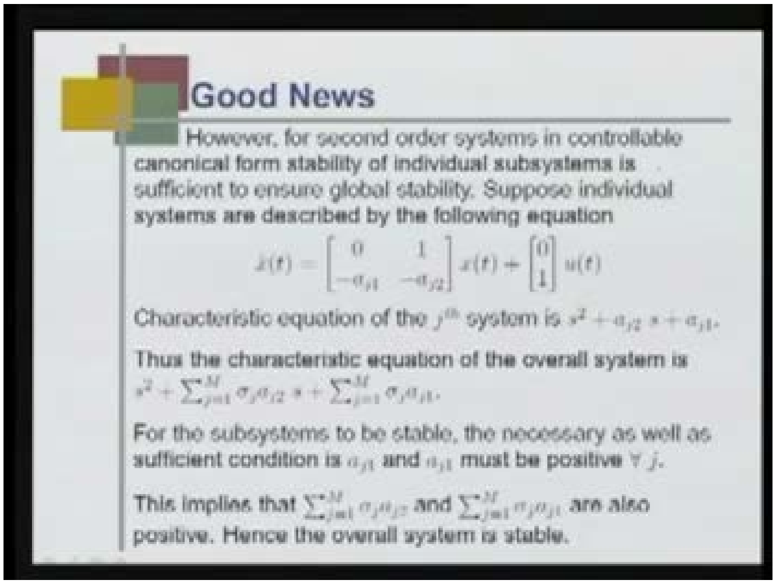
(Refer Slide Time: 50:00)



Limitation of theorem 3: if the subsystems are in controllable canonical form, Hermitian part of overall matrix will be unstable.

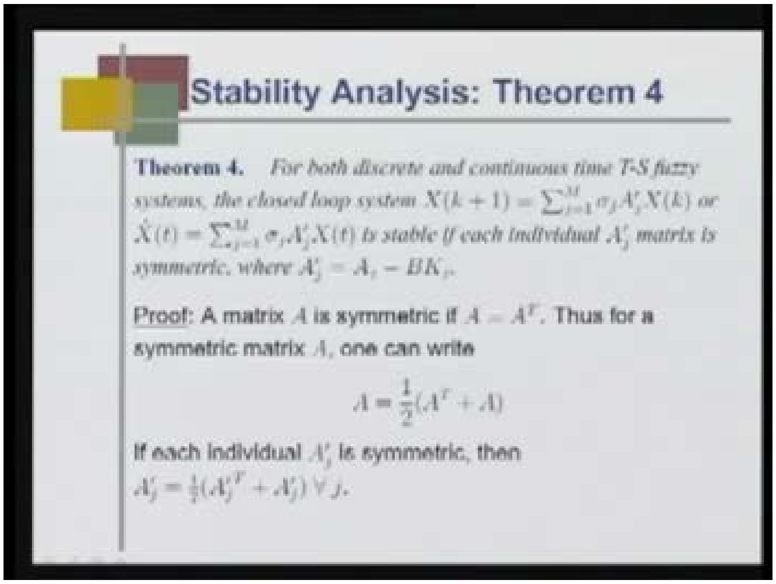
Consider a second order system like this, the Hermitian part is like this, the characteristic equation is given. This is very simple. This implies that half A transpose plus A is unstable is unstable system. If the subsystems are in controllable canonical form Hermittian part of overall matrix will be unstable.

(Refer Slide Time: 51:00)



Good News: However, for second order system in controllable canonical form stability of individual subsystems is sufficient to ensure the global stability. Suppose individual systems are described by the following equation, second order characteristics equation of the j th system is this, the characteristics equation of overall system is this. Subsequently, for the subsystem to be stable the necessary as well as sufficient condition is A_{j1} and A_{j1} must be positive. This implies that this particular summation is also positive. Hence, the overall system is simple on logic.

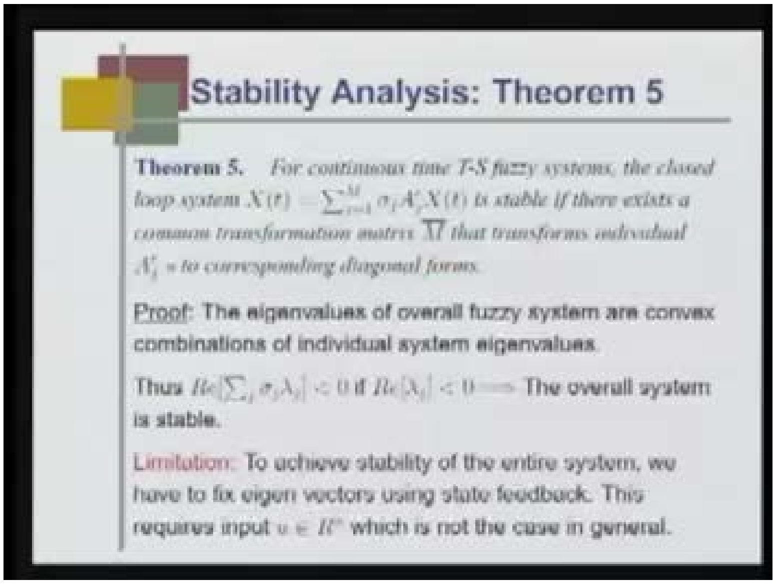
(Refer Slide Time: 51:35)



Now we will go to theorem 4: for both discrete and continuous time is fuzzy system the closed loop system X(k plus 1) which is given like this or in continuous time x dot is sigma A_j dash x(t) is stable, if each individual A_j dash matrix is symmetric where A_j dash is A_j minus B K_j. Proof- a matrix A is symmetric if A is A_j transpose. Thus for a symmetric matrix A one can write A is half A transpose A that means it does not have non Hermitian part. If each individual A_j dash is symmetric then A_j dash is half A_j dash transpose plus A_j for all j.

In this case the maximum eigen value of A_j dash is equal to the matrix measure of $A_{j.}$ Hence the proof of the continuous T-S fuzzy system is similar to the matrix measure approach we used in the previous theorem.

(Refer Slide Time: 52:00)

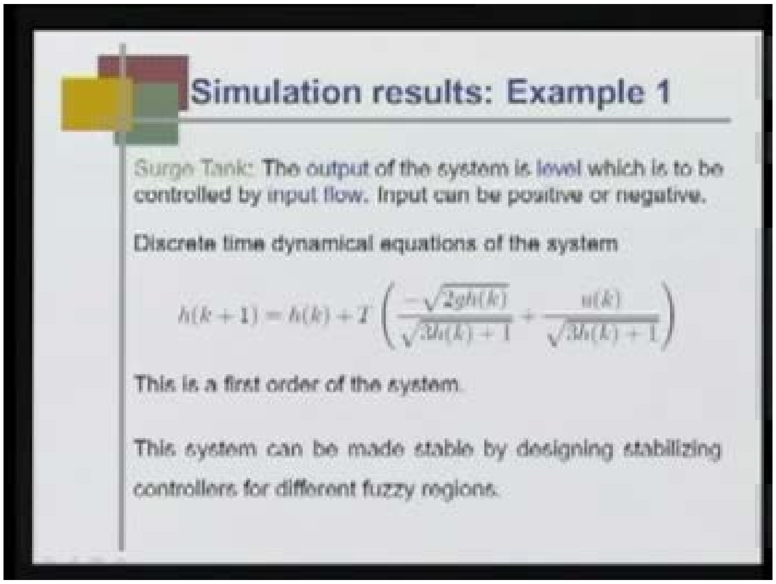


Again for symmetric matrix, singular value of equal to magnitude of eigen values. Therefore the proof for discrete time T-S fuzzy systems is same as that of induced norm approach.

Utility of the theorem- making use for the theorem the overall the system can be stabilized if there exists a common input matrix B for all subsystems an individual gain matrix K_j's are designed such that A_j dash equal to A_j minus B K_j are symmetric.

Now we will skip these things, now we will give some simulation results.

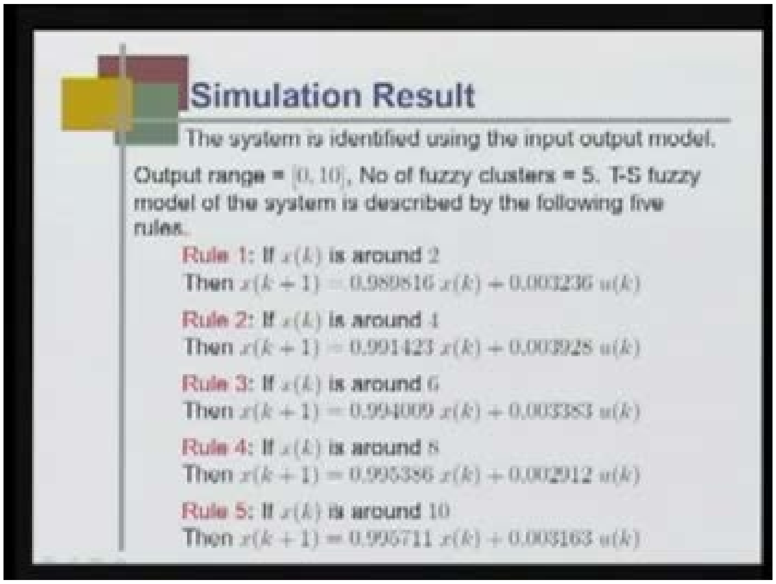
(Refer Slide Time: 52:57)



For surge tank, the output of the system is the level, the liquid level which is controlled by input flow. Input can be positive or negative.

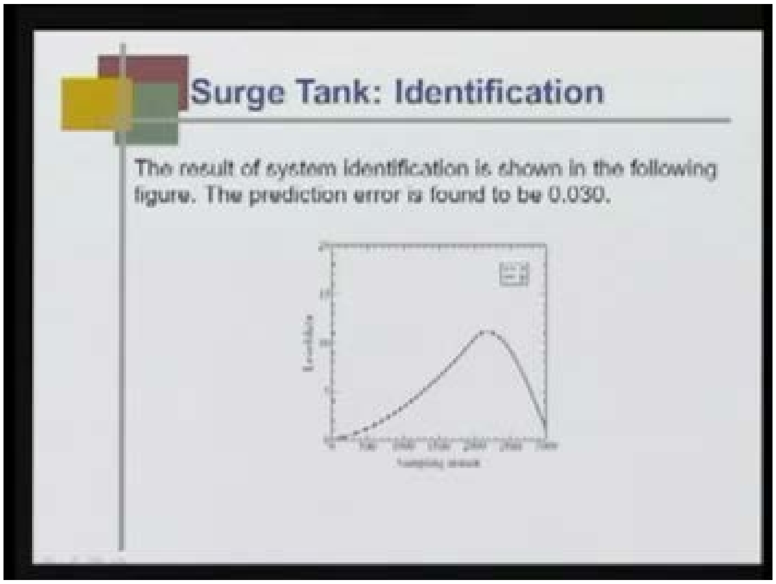
Discrete time dynamical equations of the surge tank is h (k plus 1) is h k plus t and you can see this is a nonlinear dynamics and also u(k) upon root over 3 h (k plus 1). This is a first order system; the system can be made stable by designing stabilizing control for different fuzzy regions. You see that?

(Refer Slide Time: 53:32)



What we did they used neural network to identify T-S fuzzy model. You see that error, the system is identified using the input output model, the output is the range for zero to ten; number of fuzzy clusters are five, so the five rules. The T-S fuzzy model is described by following five rules: if x(k) is around 2, then x(k plus 1) is 0.98 something plus 0.0032 u(k) and so on; rule two- is if x(k) is around 4 x(k plus 1) is 0.991423 x(k) plus 0.003928 u(k), so like that. You can easily see although B matrix is varying but almost similar, that is why the method we said is also applicable here. To the result of system identification we generated data from Surge tank model, trained a fuzzy neural network and you see that the data the desired trajectory, and the predicted trajectory, they exactly match.

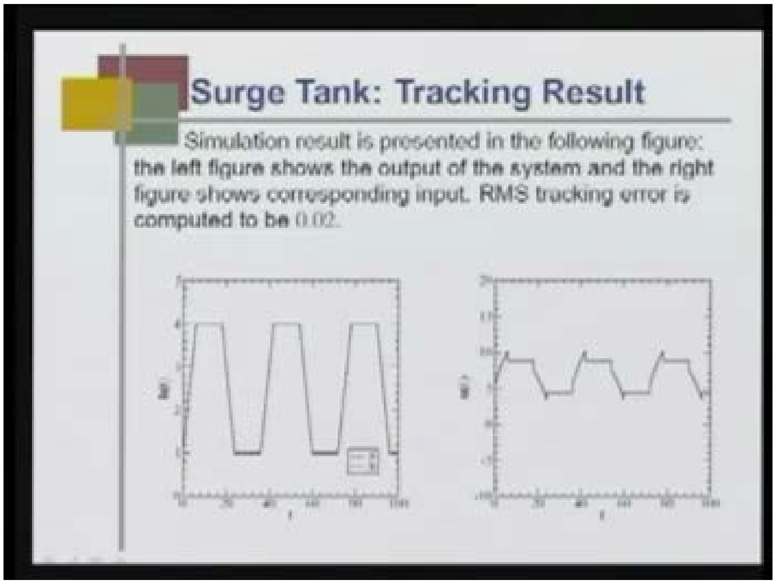
(Refer Slide Time: 54:40)



Applying the global control law, the overall system becomes like this, where the individual subsystems is controlled by state feedback and individual k a l designed for the system meets this criteria and control law stabilize the global system. Since the overall output h(k) is this quantity, the output here is h(k), h_j (k) is the j th sub system's output, if each individual system track the reference input r then the overall system will also track r.

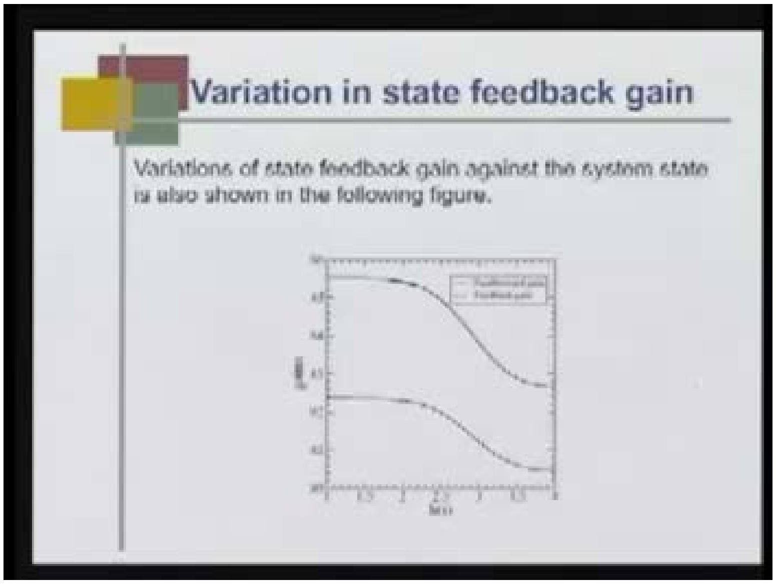
The controller thus becomes you see this is our state feed back part and this is my part for tracking where K_j and F_j are feedback and feed forward gains for j th system to track the reference input r.

(Refer Slide Time: 55:32)



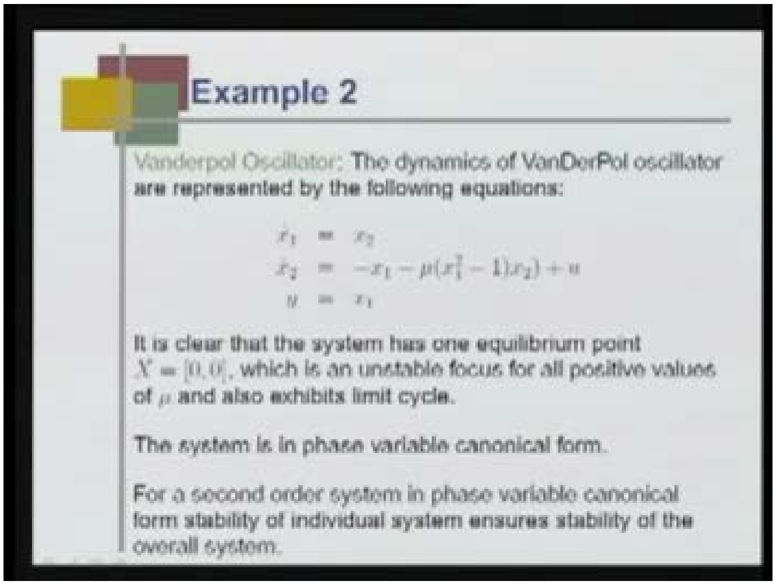
The simulation result is presented in the following figure; the left figure shows the output of the system and the right figure corresponding input. That means control action and this is my height so I am tracking a am set point tracking from 4 to 1 again 4 to 1 and this is very perfect tracking, you can easily see and controller is almost is very smooth.

(Refer Slide Time: 55:53)



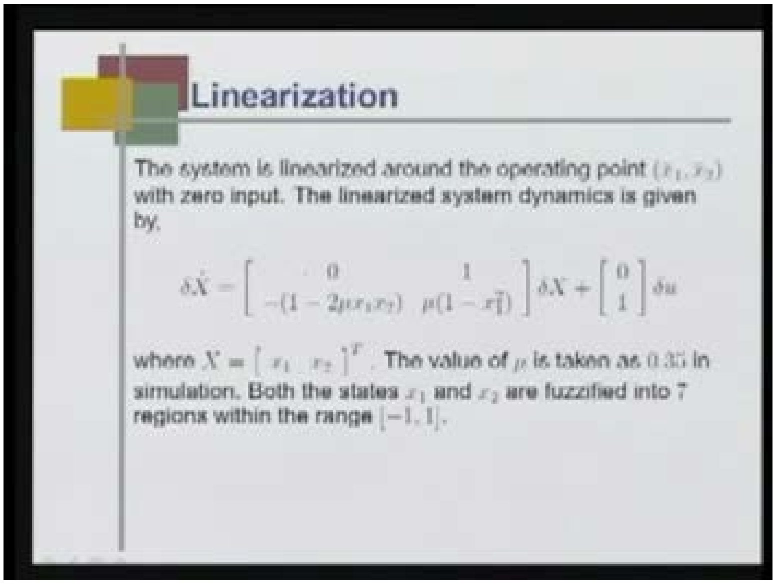
And you see that the gains that the feed back gain and feed forward gain, that are obtained, you can easily see here, they are of time bearing nature because this is a non linear system. It cannot have a fixed gain, constant gain it has a variable gain. So variation of state feed gain system state is also shown in the following figure.

(Refer Slide Time: 56:20)



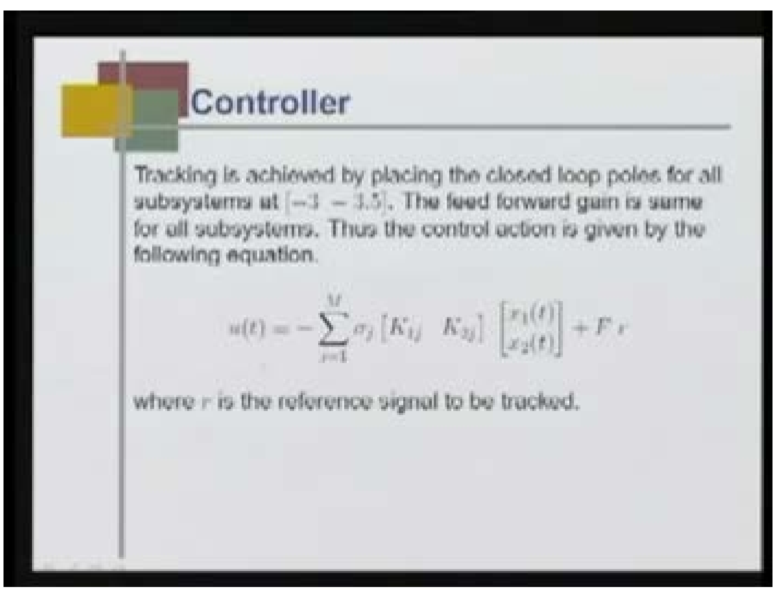
Now we take second example of Vanderpol oscillator. Most of you have studied this in the introduction to nonlinear system because this is very classic example of nonlinear system. x_1 dot is x_2, x_2 is the quantity and y is x_1, single input single output system and if I linearize this system, I have this particular form and you can easily see, my B matrix is 0 1 and hence this is common for all subsystems.

(Refer Slide Time: 56:56)



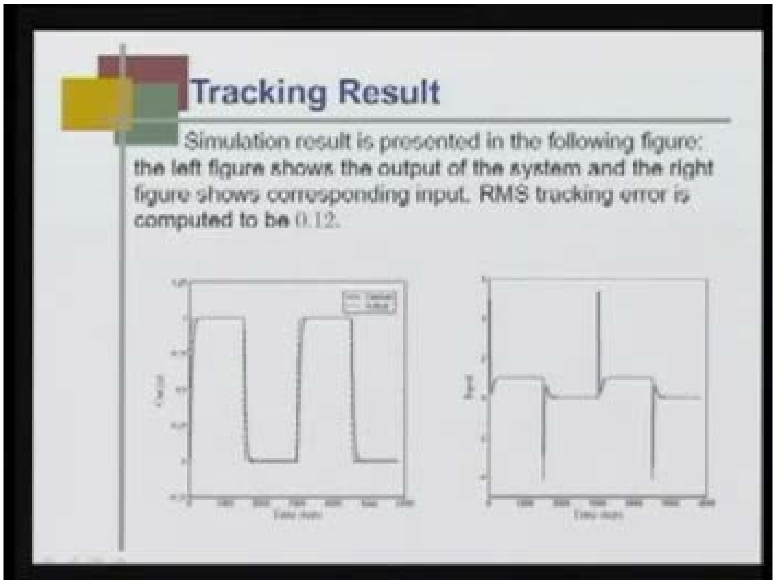
And in the A matrix also 0 1, for all subsystems only the second row vary according to x_1 x_2 and x_1 square. So we can linearize that and one example is if x t is the 0 then x dot is 0 1 minus1 2, x t 0 1 ut and similarly, we have forty nine rules. Using this the controller is designed, the tracking is achieved by placing the closed loop holes for all subsystems are minus 3 and minus 3.5, the feed forward gain is same for all subsystems. Thus the control action is given by the following equation.

(Refer Slide Time: 57:36)



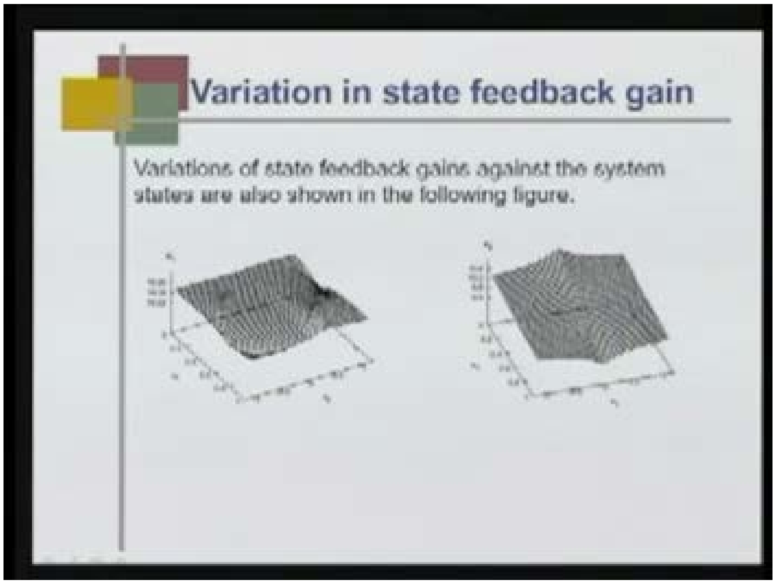
And you see this is my state feedback controller for regulation and this is my tracking the reference signal. So the tracking result you see the tracking is very very perfect.

(Refer Slide Time: 57:50)



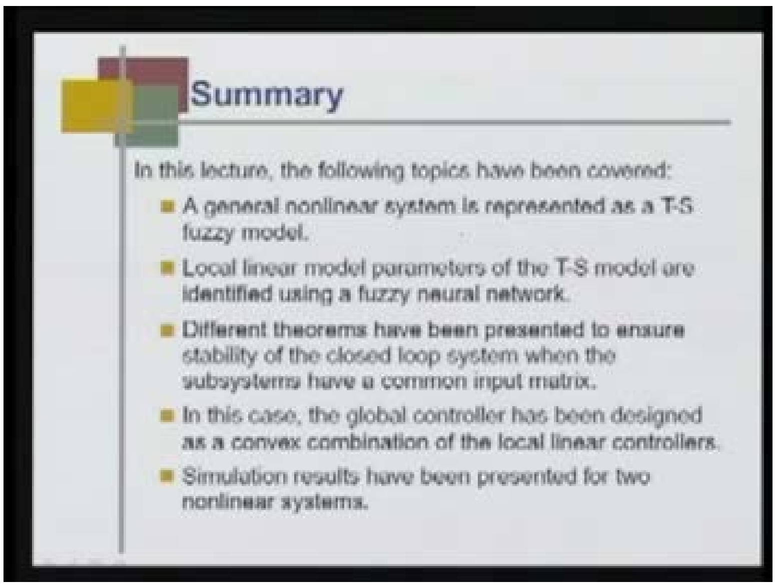
Very good tracking so given set point variation and this is my control input so the simulation result is presented in the following figure; the left figure shows the output of the system and the right figure shows the input of the system. This is my control action and this is my output. So variation of state feedback, you can easily see again, these gains are variable.

(Refer Slide Time: 58:17)



This is not a flat surface, the variable gain for k_1 and k_2.

(Refer Slide Time: 58:22)



Summary: in this lecture the following topics have been covered- a general nonlinear system is represented by T-S fuzzy model. Local linear model parameters of the T-S fuzzy model are identified using a fuzzy neural network. Different theorems have been presented to ensure stability of the closed loop system, when the subsystems have a common input matrix. In this case the global controller has been designed as a convex combination of the local linear controllers. Simulation results have been presented for two nonlinear systems. That is all. Thank you very much.