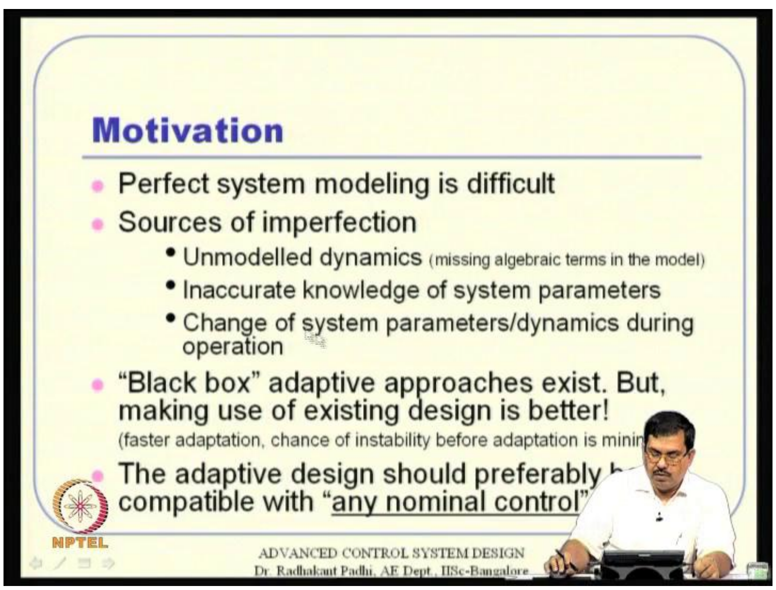
Advanced Control System Design
Prof. Radhakant Padhi
Department of Aerospace Engineering
Indian Institute of Technology, Bangalore

Lecture No. # 37
Neuro-Adaptive Design – II

Hello everyone, we will continue with our discussion on neuro adaptive design. Last class we have seen one design which has appeared from (( )), and all their colloquies and all that which is compatible with the dynamic inversion design. And as we as I hinted in the last class, we also want to discuss something which is little more generic, in the sense it is not only compatible with dynamic inversion design, it is also compatible with any other design. So, that is a technique that I mean anyway I proposed with my colloquies. So, sometime (( )), and then there are some revised persons and then there are some I mean implications and all that, we will see that some of these as we go along actually.

(Refer Slide Time: 01:24)

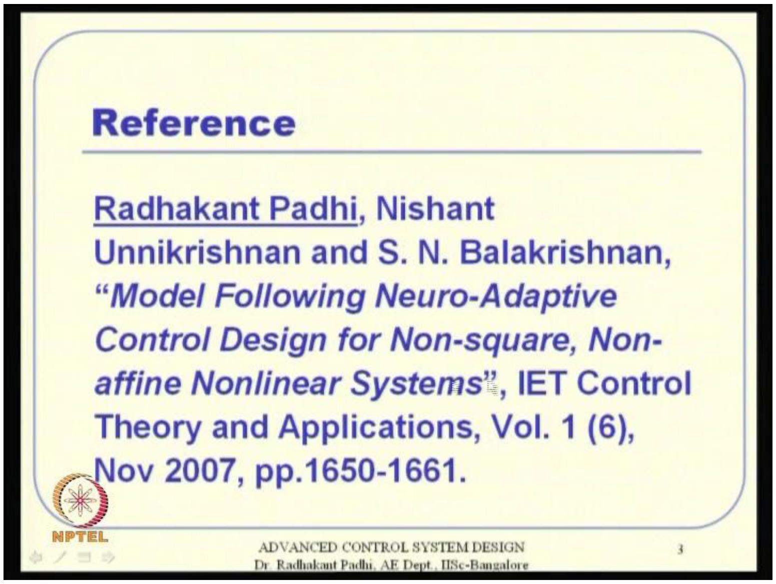


So, we will still study this neuro adaptive design and the frame work of some sort of a generic tool actually. So, motivation as we know perfect system modelling is difficult and the sources of imperfection can arise from unmodelled dynamics that is missing algebraic terms in the model. Then, inaccurate knowledge of the system parameter change of or it can also like change of the system parameters or dynamic during the operation itself. We also know that in this adaptive control when you talk about, then there are some black box adaptive approaches do exist; that means, the entire system dynamics.

So, somebody can think of capturing it through a neural network if possible and then we using this neural network model based on that, you can think of synthesizing the control. That means, the entire system model, there is no prior information actually. You can I mean, the system is identified as you go along and with that identifying model you keep on playing your control actually, but that what I consider? You take long time to adapt. And there is also chance of instability before adaptation, because you are starting with almost no information.

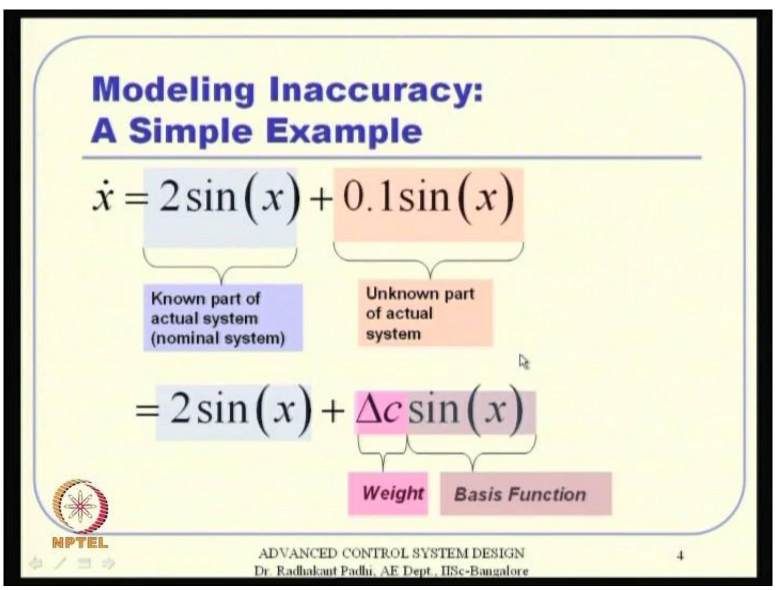
And then, so, what we really want? At least partial white box actually; partial grey box or something, what is known as like whatever is; whatever information is known to us, we will take that and whatever is not known to us we will try to identify. That way, it will lead to faster adaptation and the chance of instability is also minimum actually like before adaptation and all other. So, nominal control will have some robustness actually is based on the nominal model whatever you design, control we will call that is nominal control and that we are interested in improving its robustness. Now, as you go along, we will try to identify the unknown part and then use that unknown part for updating the control and all that.

(Refer Slide Time: 03:12)



So, and the as I told the adaptive design should be preferably compatible with any nominal design. So, reference as I told lastly, this is from our own research team and so, it is reported in the journal of I E T control theory applications in 2007. I will hopely take you through a little earlier version on that and then towards end of this lecture, I will tell you what it actually exist in this paper actually.

(Refer Slide Time: 03:30)

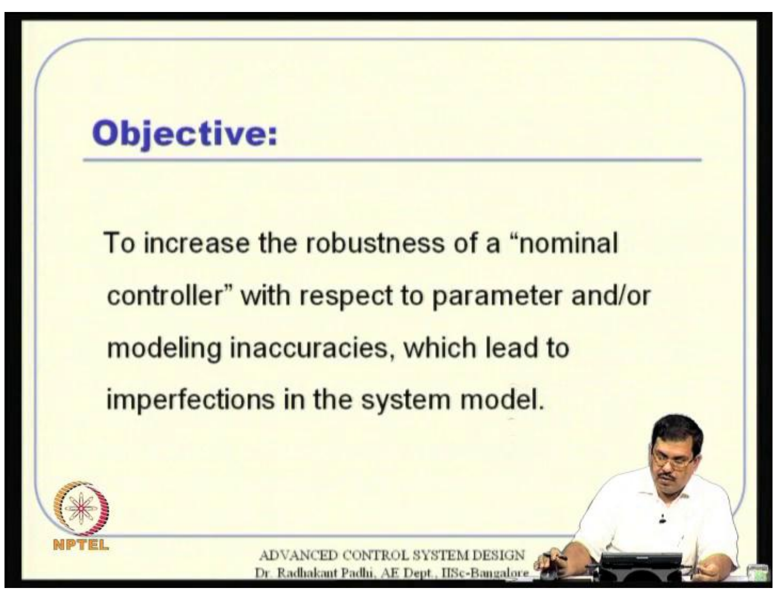


So, what is the simple example here that, we are bothered about partly I mean, we are largely bothered about modelling inaccuracy and one of the big reasons can be parameter in accuracy. But what happens is let us say, because of parameter in accuracy, you will be forced to deal with some sort of a unknown function. It is even, if you talk about a constant parameter number, that that number will throw some sort of an unknown function. So, let us see an example, let us talk about let say we have this x dot equal to 2 sin x, that is a small system dynamic where 2 appears say something like a system parameter.

But in reality, if it is really not 2, but 2.1 then, we can write it this way so, it is 2 sin x plus 0.1 sin x sort of things. So, this 0.1 sin x is unknown part, because whether it is plus 0.1or minus 0.1 or plus 0.2, we do not know about that. So, this entire part becomes some sort of a unknown. So, what is the approach here, let say for some reason, we can approximate this as something like 2 sin x plus some delta c coefficient, unknown coefficient term sin x. That means, this sin x part appear some sort of a basis function and this delta c appears on sort of a weight.

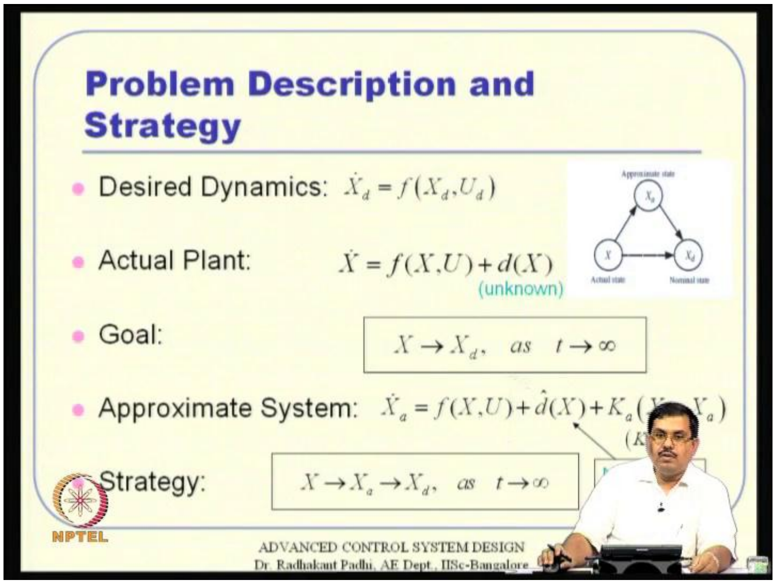
So, our objective here should be like and we do something so, that delta c will approach to this 0.1 in this case or if it is minus 0.1 it will go to minus 0.1 in that case. So, whatever is that unknown component this delta c, will go there and adaptive actually alright.

(Refer Slide Time: 04:58)



So, objective here is to increase the robustness of the nominal controller with respect to parameter and or modeling inaccuracies, which sincerely leads to the imperfection in the system model as I discussed here.

(Refer Slide Time: 05:14)



So, let us see, what to; what we do? And what is the objective in little more mathematical science actually? So, what we are assuming here is the desired dynamics is already known to us in the form of a nominal control design, so that subscript d stands for some sort of a desired objective and things like that. So, sincerely this part of the system dynamics we already know and using the system dynamic we have already designed a controller for whatever objective we want to achieve. That means U d is also available for us for further adaptive control design. So, what you are assuming here is, not only the dynamics is known, what we have already designed some sort of a nominal controller, I mean, with the help of that model that we already know actually.

However, this model is not known to us exactly, I mean this model does not represent the actual system exactly. That means, the actual plant can have some sort of a uncertainty which you are representing as some d of X actually, this term for some like disturbance and all that, but in general it is the kind of unknown part of it. So, what you are telling, at this particular system dynamics the behaviour of this is going to be different from what we think it should be there, because this is the our perception, that nominal control part of it. So, it is supposed to be here differently, because this part is unknown as long as this is non zero this dynamics is different.

So, what is our objective here is nothing but this particular X the states of the actual system should go to the states of the desired system so, that should simply like track this. That means, we know that the objective is met by X d already, because we designed a U d by that time. So, obviously the X d as it appears satisfied all the necessary things so, that means, X d is a good colour to talk actually that way. So that means, our objective should be like X should go to X d as soon as possible, because there are more, it will talk to it will follow X d anyway.

So, all the time we are in; we are interested in simply ensuring, that X should go to X d as soon as possible actually that that is all, but the problem here is because X dot contains d of X and we do not know that. So, whole idea is can we approximate that, that means, we use this neural network or something some function approximation tool to approximate that, but because this approximation can never be exact. So, this dynamics with the approximation let say, I talked that X d hat of X. So, with the approximation of d X as d hat of X, this dynamics is again going to be different. So, that is why, we need to denote it something else from other state.

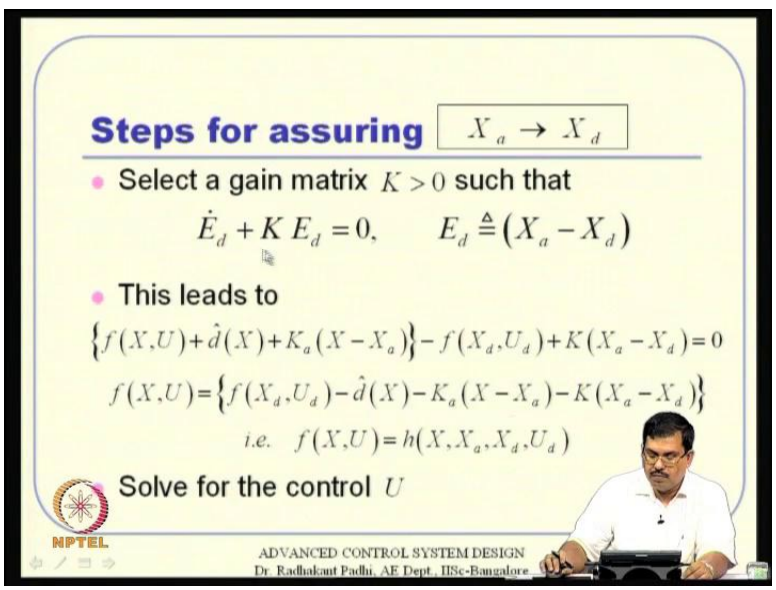
So, that is what I call X a; that means, this is like approximate system dynamic actually. So, if you if you see this d is replaced by d hat, which is neural network, I mean ideally speaking this is say supposed to be neural network approximation of d X. And this term is added later, I mean this is this additional term with K being a positive definite matrix is needed for ensuring the bound and all that, later we will see that how it helps us. So, what is our strategy, I mean objective is that X should go to X d, because X dot is not known to us completely we will not be able to do directly.

So, what you are telling here is X should go to X a first and X a should go to X d, because X a dot is known to us completely, because this dynamics is nothing unknown to us, because d hat is something that we are computing actually. So, whatever is X actually whatever X comes of in the reality. So, X should go to X a and X a should go to X t. So, essentially we are having this two loops now, one we have to ensure that this loop is satisfied. That means, X should go to X a, then we also should simultaneously makes sure that, this loop is satisfied; that means, X s will go to X t actually. So, that is the. So, these two loops we will study.

Now, also just to make a point here, that as soon as X goes to X a, this term is zero anyway, that term is 0, then d a; d is nothing but d hat of X. That is the way, we will identify that everything happens properly, then, X should go to X a very close at least. In that sense, this d should go to d hat actually, that is our objective anyway. So, we have to assure, I mean two loops we have to assure. First we have to assure that X a, should go to X t and then simultaneously X should go to X a and all that so, we will describe this two loop one by one.

Let us first discuss, so that, somehow X gone to X a, then how we will ensure that X a, should go to X t and after that actually. Also remember that we are not assuming anything like square system, anything like that; that means, this dimension of X and U can be different really. So, that brings in lot of additional difficulty actually alright.

(Refer Slide Time: 10:14)

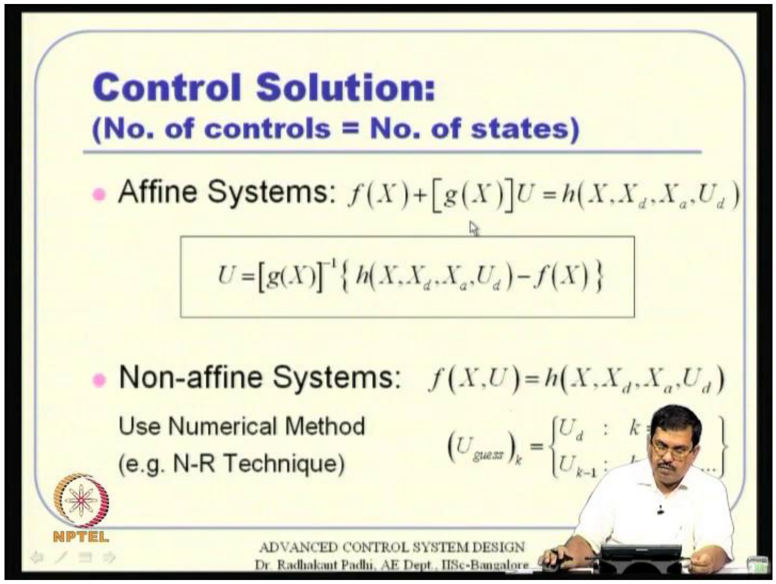


So, now let us assume, that X a is known to us and we have to just assure that X a should go to X d. So, let us try to kind of enforce some sort of a desired dynamics like the way that we do in dynamic inversion so, we define E d error actually. So, X a minus X t and enforce E d dot plus K E d equal to 0. So, if you enforce that, then X a dot minus X d dot and all that you can I mean this E d dot is X a dot minus X d dot and X a dot is known to you, X d dot is also known to us actually. So, that we substitute all that and then try to solve it as much as we can try to simplify actually. That means remember this U d is already available to us, that we can use it so; that means, you put everything in a right hand side, but you leave it is like f of (X, U).

So, this equation if I define this entire term whatever happens in the right hand side have something like h of (X, X a) all that actually. So, they will lend up with some sort of a equation like this. So, if you can solve this equation, then we have done actually, until now there is no approximation nothing there and all that. So, if you if you can solve this equation exactly; that means, this equation is satisfied exactly and hence X a, should go to X d and all that. Now, the thing is can we really do that and then it arises as I told this there is a dimensional in comfortability and all that; that means, in general the number of equations and number of variables are not same.

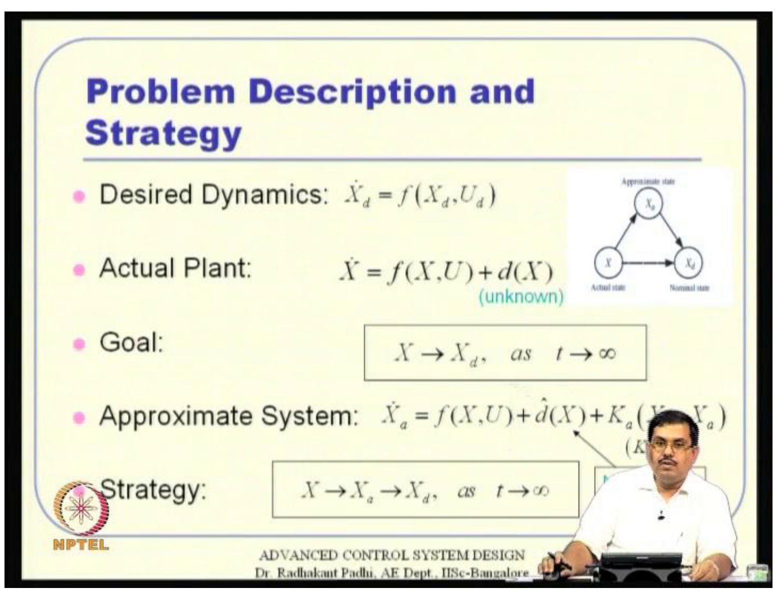
And here is a case of over constraint problem; that means, the number of equations are really dimension of f dimension of f is n and the variables, that we are talking about is you actually. So, it is not really a under constraint problem, in general it is a over constraint problem.

(Refer Slide Time: 11:52)



So, let us try to see a simplified case where you think, let us discuss case wise case or so, we discuss number of control is equal to number of states in that sense what happens. It is a very prevail case or I mean very unrealistic case rather, most likely it may not happen, but let us say in some problem it happens, you have sufficient control x n available. So, 3 x is stabilized and three I mean; three control and all that actually you can see that.

(Refer Slide Time: 12:14)



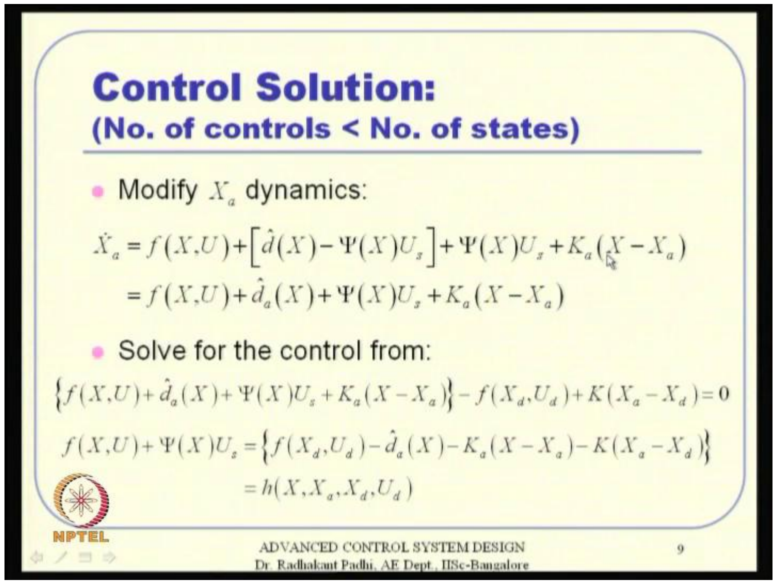
So, if it happens like that which is again this is very rare, but still if it happen sand it is also control affine; that means, the system dynamic is control affine and remember in genral we are not even assuming that. So, it we are talking about non-linear, non affine and general non-linear systems and all that. So, we are not talking that assumption either what you can still as introduce that assumption here; that means, system dynamics is control affine and number of control is equal to number of states that. Then we can write the above equation here something like this and in that in this particular case it has like because of this assumption this is a kind of equality constraint. So, we will be able to solve this control as something like this, provided g of X is never singular.

So, if g of X happens to be singular somewhere on the way, then there is same regular problem as what we had in dynamic inversion actually, so we have to take some precautions for that. Now, let us relax this assumption, one way one so; that means, first we will relax this affine system assumption actually. That means, in general this system dynamics can be like that, then somebody can think of using numerical methods something like Newton Raphson technique. I mean it must still be possible to solve it in a close form from. Even though it is non-linear function like this, provided you know what is the function that difference is its own dynamics is system dynamics.

For example if it is sin X sin of U is something there, then U is sin inverse of that I mean that, that particular thing is still the close form solution. But we will not be able to talk more, because we do not know the general form of this function, unless you know a particular system dynamics. So, in general somebody can think of using numerical method and it terms out that Newton Raphson technique has quadratic conversion and it actually converges very fast. And also, some the good thing about it is, the adaptive control if you see that, it all depends on how big is this d of X.

And if you are nominal system that dynamics is fairly good you know. That means, the accuracy part is fairly good here, inaccuracy is there, but it is not really a very bad inaccuracy I mean may be 10, 20 percent accuracy here or something like that, part of the system dynamics. Then it so, happens that your U of U will not be very different from U d either, the system dynamics are not very different. So; that means, you can think of guessing the value for U as U d, because you already have a good guess actually. So, not only the Newton Raphson technique converges fast you also start with the very good guess actually so that way, it is even faster alright. So, that is the way to talk about this non affine problem.

(Refer Slide Time: 14:54)

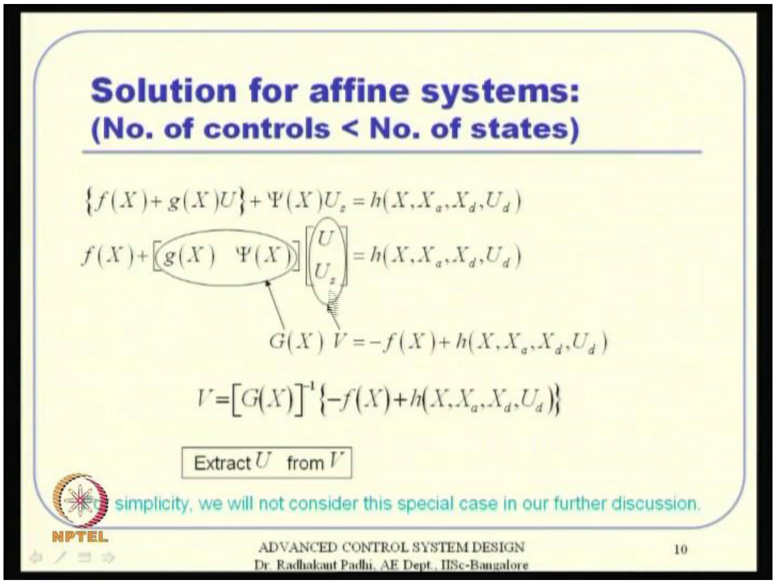


What about this generic case, where you have number of control, less than number of states, which is very general I mean in general that is the case. Then, what you have to do is, there is a idea of let say, we think of something like add and subtract operation here. So, what we do here is, we go back to this X a dynamics; X a dot dynamics here and then in that dynamics we want to add and subtract this term. Let us say, psi of X times U s, I mean subtract and then add it up. And then what you think is this particular thing, what you are seeing here is something like d a hat of X that. So, what happens here, this X a dot, I will not be able to kind of identify d hat, I will be able to identify the entire thing together basically.

So, by dong this addition and subtraction what I am doing here is, I am introducing free more free variables here and that is what I we interpret that is would have control. But this does not told is very different from what should have controlled, we discussed in the last class actually. So, that should have controlled in the setting of dynamic inversion and all that. Here is just an artificial variable that we simply call this would have controlled, because like you control variable actually, some sort of every variable here. So, you modify the system dynamic, in such a way that if you take U and U s together, this U and U s together, then it is same dimension as number of the dimension of state.

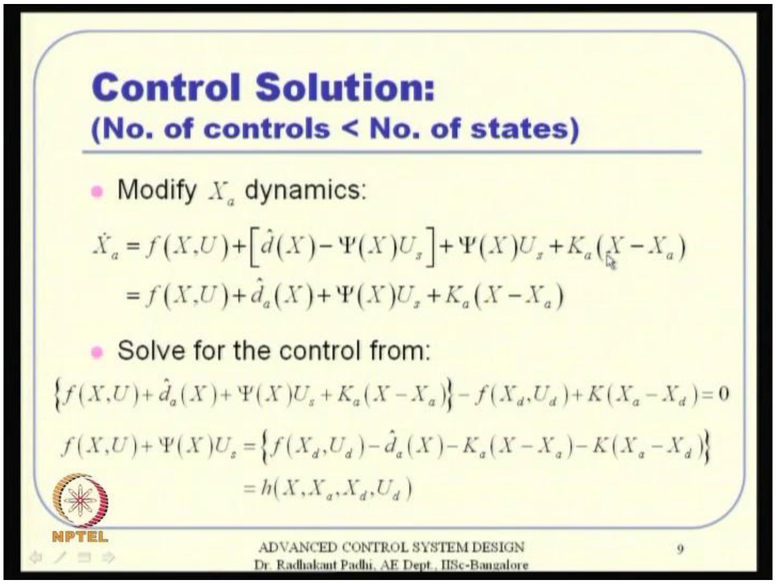
So, U is; let us say, X is n dimension, U is n dimension and U s is n minus n dimension. You have to add it in such a way, that U and U s will be like it. If I put them in below, if I put U and U s patrician vector sort of thing. So, that that V whatever V, I am talking about this is same dimensional X actually; that means, this equation that we are talking here, is still number same as number of variables and number of equations, this particular equation here.

(Refer Slide Time: 16:58)



So, let us in general again somebody can think of solving for you now U and U s together we had solved for U and U s together. So, you if it happens to be control affine again then again I have to I can write there like that and I can separate this out and put them together and think like that. So, U and U s, I will define at b now, this equation will have number of equation equal to number of free variables. So, I will be able to solve for V. So, once I solve for V, I know U actually. So, U is as inverse of that.

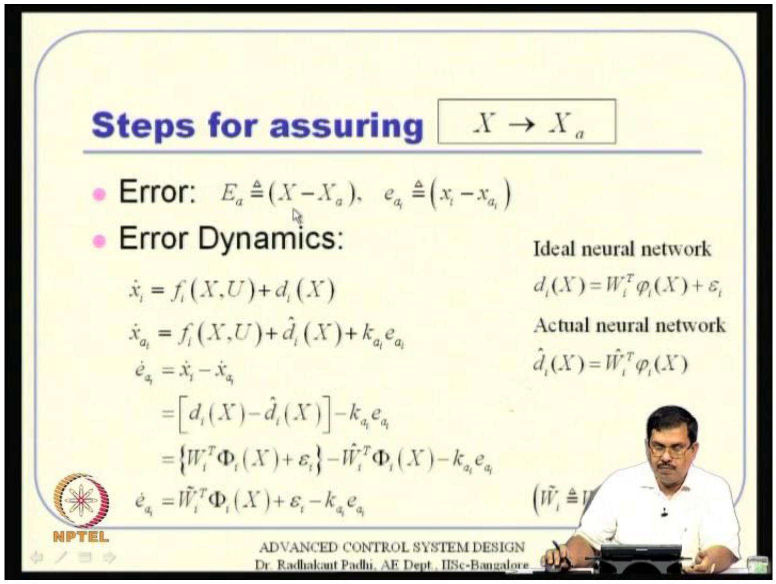
(Refer Slide Time: 17:15)



So, I mean, there is a little bit requirement here, because the entire solution will now depend, not only on the gain selection and all it will also depend on what function you select here. And later, I will give example where this form is not that difficult to turns out, because whatever experiment we have done. Normally, this psi of X terms out to be just like constant matrix sort of thing, with that will be able to get some solution. But in general that, what if the optimum value of psi of X? I think, that is still an often question actually. So, how do you fill it that?

Anyway, what your now we are not interested in an optimum solution let say, but we are interested in at least a solution. So, in that set of the solution is ready that way. So, once you get the solution for V we will extract U from V that first term first term element is nothing but U.

(Refer Slide Time: 18:32)

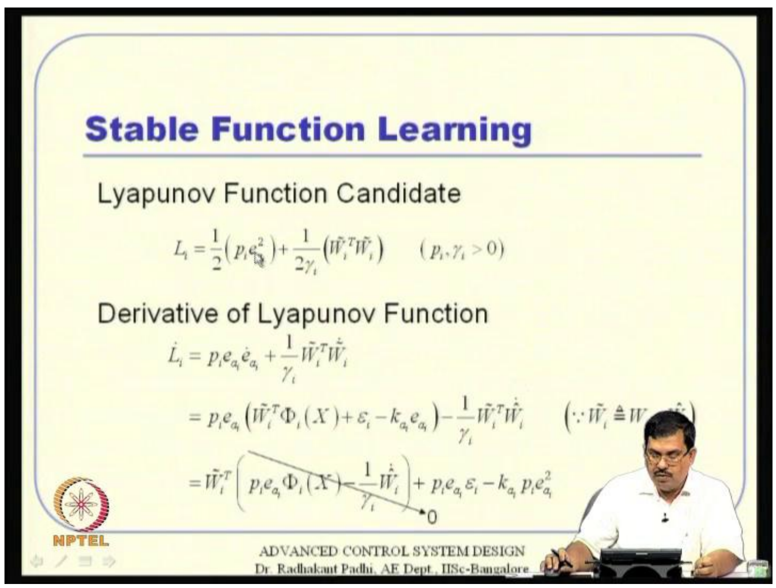


So, for simplicity we will not consider the special case for further discussion and all; obviously, that is requirement and there are, I will give an example of course and then first and further we that actually alright. So, what is; whatever about the next step so, we ensure that, that is now; we have to ensure, that should also happen actually simultaneously. So, for that, we will define some error term again X minus X a and this term will take the help of Lyapunov theory. So, if X minus X a, then the i th general like again like last class we will not discuss the entire state vector and all let us pick out the i th channel dynamics, which is X i dot is like that this is the original dynamics with unknown terms and all.

So, X a; X a i dot, the i th component of the approximate dynamics is like this, provided the I mean we are also assuming that this gain matrix K a is also diagonal and all that actually. So, that with the with the diagonal element of K i mean, if the K a is selected as a diagonal matrix here; sorry here and then we will this term is also like K a i, e a i. So, e a i is defined as X minus X a, E a is like that. So, with that i th component will turn out to be like that. So, Lyapunov theory, before using Lyapunov theory, we have to know this e a i dot, the error dynamics of that, what whatever dynamics we are talking about. So, e a i dot by definition is X i dot minus X a i dot and X i dot is available now, X a i dot is also available here.

So, we substitute all that and then try to simplify it and then tell my error dynamics turns out to be a function of error in the weight, error W tilde, W tilde is like W minus what exactly like what you discussed in the last class. So, there is an ideal weight, which I will call that as W, not discussing like W star and all that, I mean W is ideal weight for us and W hat is actually weight sort of thing. So, then you can put that all that and these two terms can be combined and I can write it this system dynamics. This is all with the assumption that this ideal neural network I mean representation is like that. d i the error, actual error can be represented something like that with epsilon i as error quantity there are ideal approximation error. And actual neural network is represented by W hat transpose times phi of X actually.

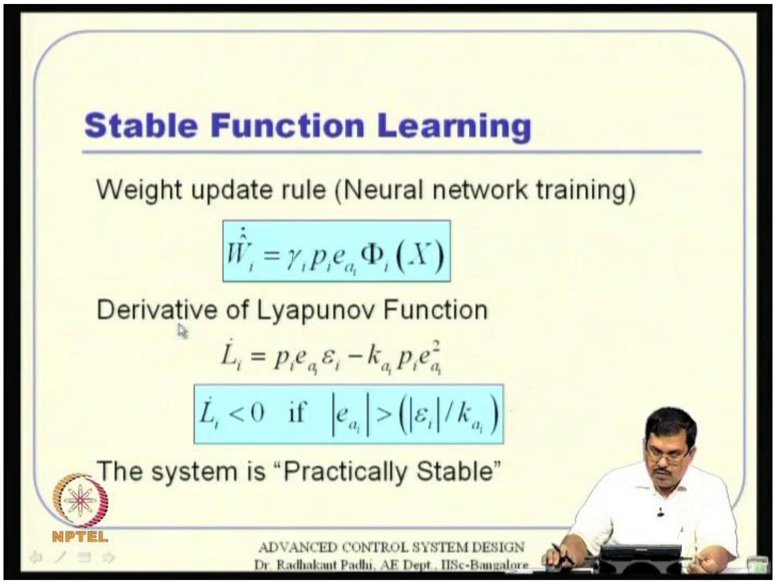
(Refer Slide Time: 21:02)



So, with that by the way all these phi's are same phi actually, there is no difference here. Probably, this is small mistake here, these are this phi, I mean what you are using in the left hand side and right hand side, this phi is same as that phi actually. So, it is alright. So, with that this is what is happening? Now, we have to select a Lyapunov function candidate. So, we will select that so, this is you want to have error minimum and then weight also should remain bounded and all that. So, we will take this Lyapunov function candidate, where p and gamma are tuning parameters both are selected as positive quantity.

Then, you take the derivative of that this particular i th channel Lyapunov can be the derivative, then substitute E i dot, whatever you have here and all. And then exactly proceeding p four and all that, this is something like W i tilde remember W i tilde is W i minus W hat, where W i is a constant number. So, W i tilde dot is nothing, but negative of W i hat dot.

(Refer Slide Time: 22:30)



So, using all that, this simplicity can be brought in and then W tilde transpose in W tilde transpose happens both sides and these are scalar quantity remember that. So, I can take it in the I mean inside actually. So, W i tilde transpose I have a common term and this is what is what I am left out. So, this W i tilde this coefficient thing, in case is enforced to zero like last class, because this particular thing we do not know. Because W i tilde is W i, this ideal weight is something that we never know, so because of that this difficulty arises. So, we cannot talk about whether it is positive negative all that. So, all that what you do is, that coefficient let us enforce it to 0 and once if the once you enforce the coefficient is 0, it also gives us a weight update rule W i hat dot is equal to like this.

So, then this Lyapunov function candidate will still have to analyze, because it is we have just ensured that one term is 0, we still left out, we are still left out with that. So, that is; that means, the L i dot is happens to be like this and this is quickly negative provided this condition happens I mean that is easy to verify actually. If you if you can if you think a little bit on that one it will happen, one is this L i dot less than 0 that is the question right. So, you simplify that and try to take some terms in the left hand side, right hand side of that and it turns out that if you I mean if this condition is satisfied then L i dot is negative.

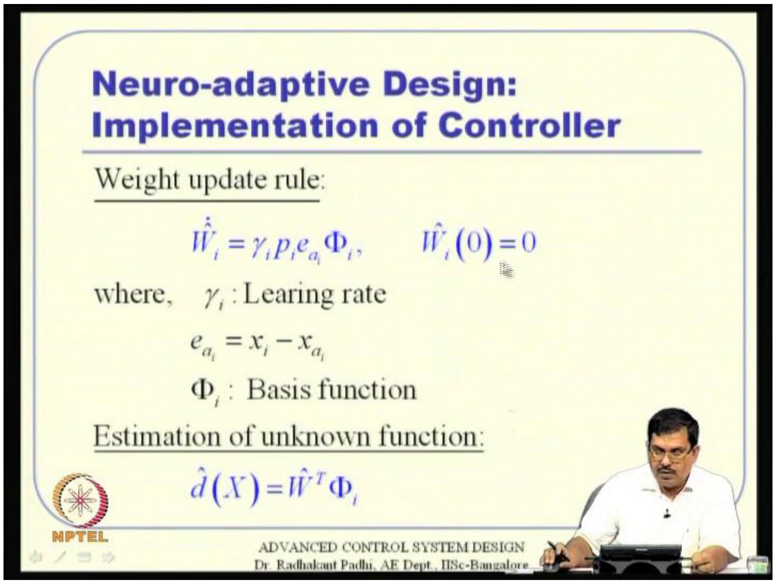
That means as long as this a dot a epsilon i is the neural network approximation error in the ideal neural network sense. So, that is anyway small already and then this is we do not have to really live with that, we also have a little bit training on; that means, we can talk about even reducing that further. So, that means, modulus of epsilon i divided by K a i and you remember K a i is a positive quantity. So, you select this K a i sorry; the training requirement, you select this K a i, I mean anyway positive quantity, but I will strongly recommend, that you select greater than 1.

If you select greater than 1, this quantity is (( )) actually like epsilon i quantity whatever you have, divided by some number, which is greater than one; that means, that quantity that you have in right hand side is further seen. But also, you can say that, you cannot keep on increasing forever I mean that if you do that, then if your system dynamics are clear this is actually some sort of high gain problem will start appearing. So, we will have very high gain all that do not try to do be make it infinity sort of thing then this dynamics will go into a stable actually. So, we have to be slightly careful about that.

Anyway so, again we; what we will end up with if the system is practically stable; that means, we are; we will be able to reduce it almost very close to zero, but theoretically speaking, we cannot claim that it is a asymptotically stable. So, that is the difficulty what results will show that, we are actually like result, consist almost close to asymptotic stable, where you are here actually alright.

Summary sense, what you really need to do and implement remember, adaptive control things are rather easy to implement and all. So, first of all we will initialize the weights to zero. And then this is the weight update rule available to us where E a i is something that needs to be computed that way and remember this in general this X i is supposed to come from sensors and filters. We are not supposed to do this system dynamic propagation, but in numerical experiment to verify our ideas and all. We will assume that there is some requirement some sort of a function and that function will be used only here, nowhere else. If you use that function here and just simply integrate in the background, whatever value comes out, that is of something that will assume that is coming out from the sensor actually.

(Refer Slide Time: 26:02)

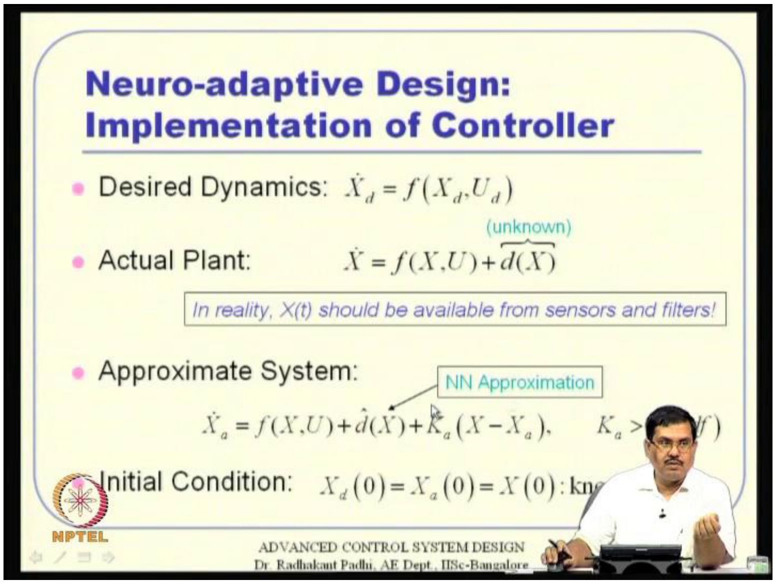


So, with that, what you really need to do is, you compute a i is something like this and gamma and p are the tuning quantities any way. So, both to be selected properly and remember gamma is supposed to be a large quantity compare to p, because see ultimate if you see here, this term should have lesser weightage compare to this term that is our objective actually. So, I mean both are there in together anyway, but given relativity sense this quantity should have; should be much less or weightage compare to this because our primary objective is to minimize this e a i basically. So, I select i gamma, then this term will be reduced actually that way.

(Refer Slide Time: 25:02)

So, this is; so, that is i select a initial weight as 0 and then this is my weight update rule available to me, where pi is the basis function that we have to select number of basis functions and all are also dependent on the problem. And then e a i is something available to the separation actually. So, once you have this at any point of time we have value for W hat t and using that W hat t will be able to compute or estimate this d hat of X this is our estimation of d hat of X.

(Refer Slide Time: 27:18)

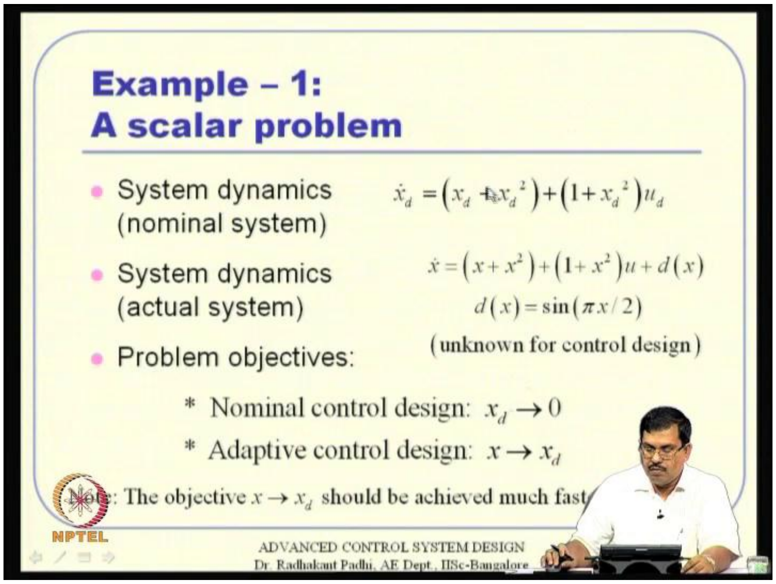


So, W i t is available from integration of this equation and then d hat of X can be computed that way. Once it is there available to us, then we have to I mean to compute the control and of those formulas, we have already discussed. How do you compute your like U out of that formulas and all these kind of things whether you compute this way or you compute that way and think like that. So, those things are available to us. On the way, we have to actually propagate this system dynamic equation. So, we have to have desired dynamics, then actual plant and then like approximate system dynamics and think like that we need to keep on propagating.

So, in implementation we are supposed to propagate this equation and this equation to get X d of t and in reality X of t should be available from sensors and filters. But for numerical experiment you can you can think of propagating this equation with some sort of an assumption of the X. And for initial condition, all these initial conditions should be preferably same, because that is where you start your adaptive control. And control formula calculation formula is, I have already given that before that is the way to implement the control.

I mean say integration of differential equation, including the weight with their all integrated forward in time actually, there is nothing like back and forth integration. And all that these are all one time integration starting from t 0. That is why, it is easy for implementation that is why, it is also computationally no issues at all.

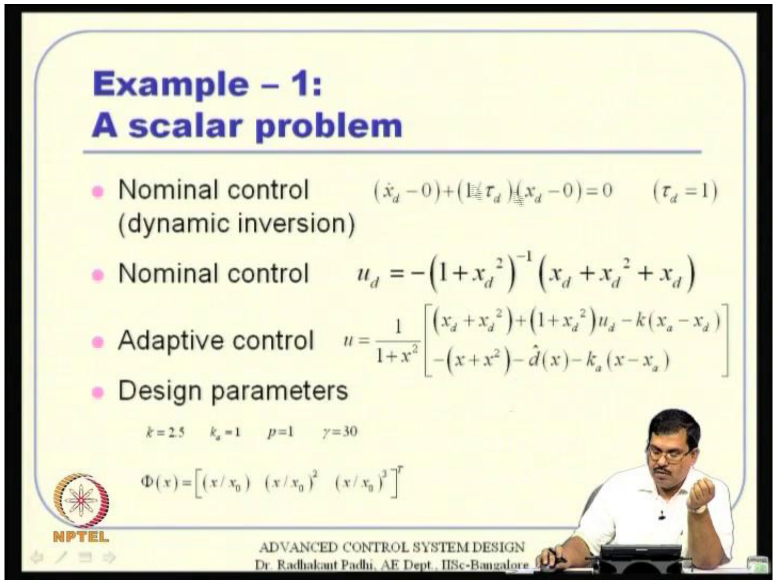
(Refer Slide Time: 28:48)



Little bit examples before we proceed further. So, first is a motivating scalar problem and this is very small scalar problem where X is; X and U are both scalar. And suppose this is a system dynamic that is given to us, but in general let us have this d of X contains a term like some sort of a sinusoidal, sinusoidal term, remember this term is unknown to us. So, we will not be able to assume anything anywhere other than integrating this equation, I mean just for numerical validation. So, you can go through the problem objectives and all tell first to design U d. So, for designing U d, you will tell we want a kind of a stabilizing controller.

So, X d should go to 0 and then after this the ending of U d, we have objective for X such that X go to X d that is all. And the objective of X should go to X t; I mean the objective of X going to X d should be achieved much faster than X d going to 0; obviously. So, pictorially speaking I mean this is something like this suppose you have this t and there is some X d and some X d going on actually there that is our X d. I do not know, then if I have some X somewhere, then X should go all to that one very fast, after that you should after that it will develop along with that first.

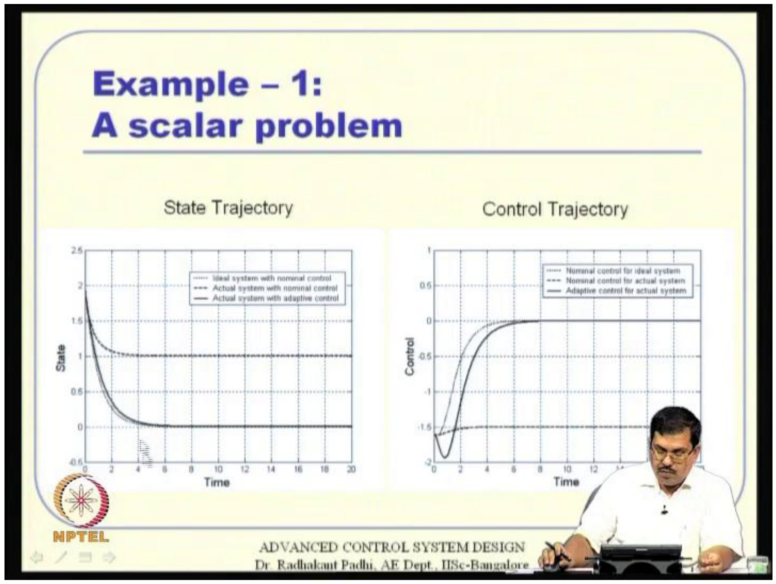
(Refer Slide Time: 30:40)



So, as far as X is concerned, this is our X, this is X of t. So, X d should go to zero, but X should go to X d and X d any way going to 0. So, X will also go to 0 later actually that way anyway. So, this is what will happen there and then using the; for checking this control validation and all will still use that dynamic inversion sort of filters for designing U d. So, you assume that error dynamics to be like this and nominal control happens to be like that, we have discuss this example in the dynamic inversion lecture also basically. So, U d turns out to be something like this and adaptive control, if you go through all that theory, that we discussed before, it will turn out to be like that where d hat has to be evaluated actually I mean that, that needs to be estimated first.

So, design parameter is selected is something like that remember again gamma is much larger than p remove that. So, that is why the error in weight is i less penalize compare to other one. And phi of X is selective something like a power series in case X, X square X cube and all that and three basis functions only so, that is sufficient for us in general. So, if you see this, I mean there is a little bit interesting thing here, if you if you just simply use this adaptive control formula. In the real system, you can very clearly show that this system dynamics will have something like multiple equilibrium point actually. So, we can I mean I will not go through that, but I will encourage you to do that yourself probably.

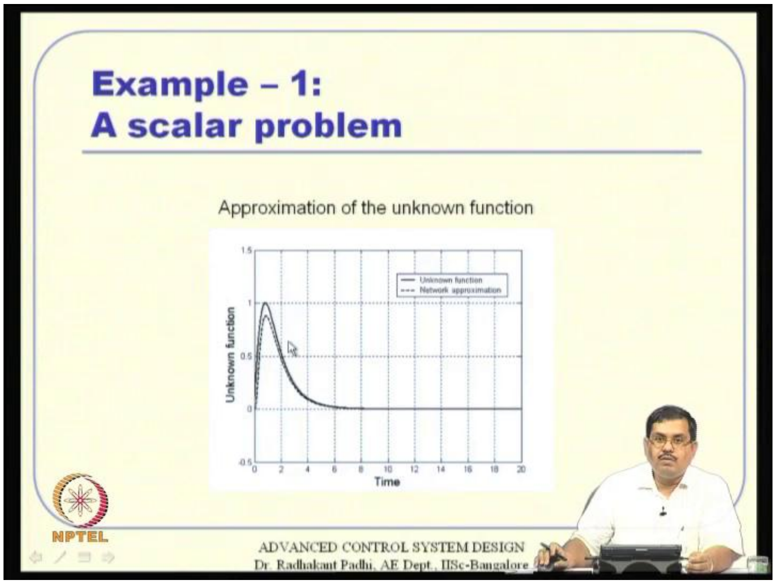
(Refer Slide Time: 32:22)



Then, if you if you simply the problem is, if you simply use the nominal control formula, which is like this here and do not do any adaptation. Then, because of the sin pi X; sin pi X by 2, because that term is here in the system dynamics, will not be able to do about so; that means, this system dynamics equilibrium sense. We will be I mean, you will have three equilibrium point actually. So, one is 0 equilibrium point, one is 1 and another one minus 1 also. So, when it turns out that, if you start anywhere between plus and minus 1, then you will go to 0, but if you start anywhere outside plus or minus 1, then we are supposed to be dropped at 1. So, if you start to beyond plus 1, then we will stabilize it plus 1, if you start beyond minus 1 and other side, then we will get dropped in the minus 1.

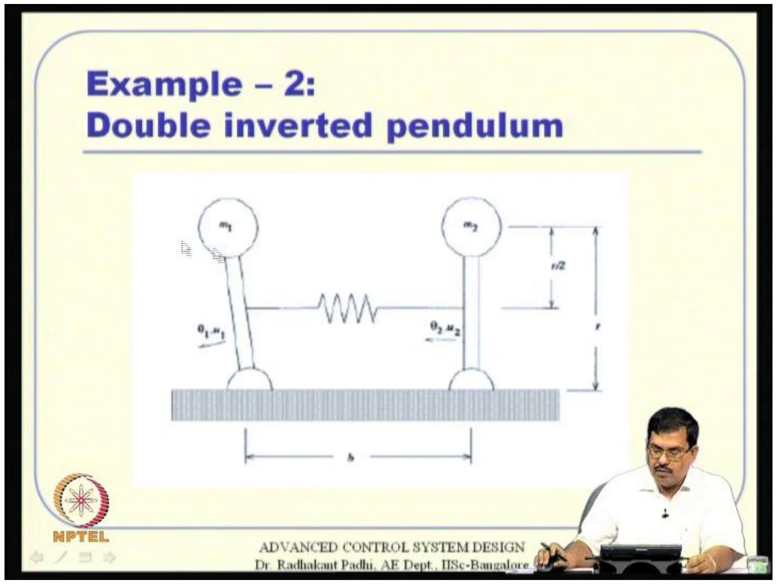
So, you will not be able to go to 0. So, now, if you really want to go to 0, apply the adaptive control. So, all that it will do is say it will be a as if it is a nominal system dynamics actually I mean the entire philosophy is we modify the controller in such a way that, the states of the actual system behave like the state of the nominal system. So, because nominal system is any way there, that problem is not there here. So, it will simply the nominal controller will ensure that the actual states should follow like a nominal state. So, you will never get dropped here, it will go to 0, so and also you can see that the control I mean; the difference of control is also there. So, I mean if you see this; the like this is your head of t control for the actual system; that means, this is the actual adaptive control that you are applying here. Otherwise, what will apply is, if you simply lye on the nominal controller, then you supply apply this controller and with that you will be able to get dropped here. So, that is also, obviously, not everything to.

(Refer Slide Time: 33:56)



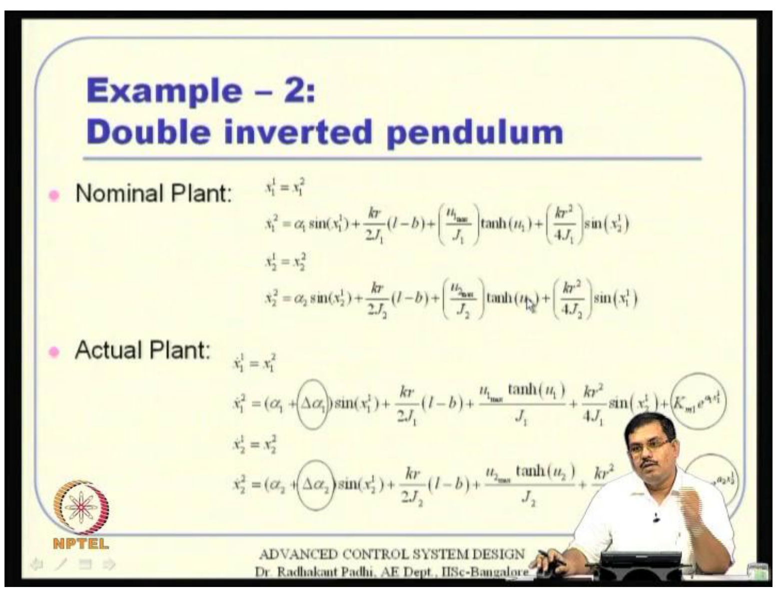
Unknown function identification sense general we are doing the d hat should go to d. So, that is the another validation part of it. So, whether d hat is actually going to d or not. So, that way this is what is plotted here, because d in this numerical exercise, we know d of X we can plot that and then you can plot d hat as it is computed from the formula that we have. The W hat transfer phi of X so, that is why, you can validate.

(Refer Slide Time: 34:34)



So, alright, so, this is a small problem, but what about the larger problem actually. So, we will take another bench mark problem, which is like a double inverted pendulum and all that. So, this system dynamic this problem is like that to inverted pendulum connected by spring, let us say both will oscillate.

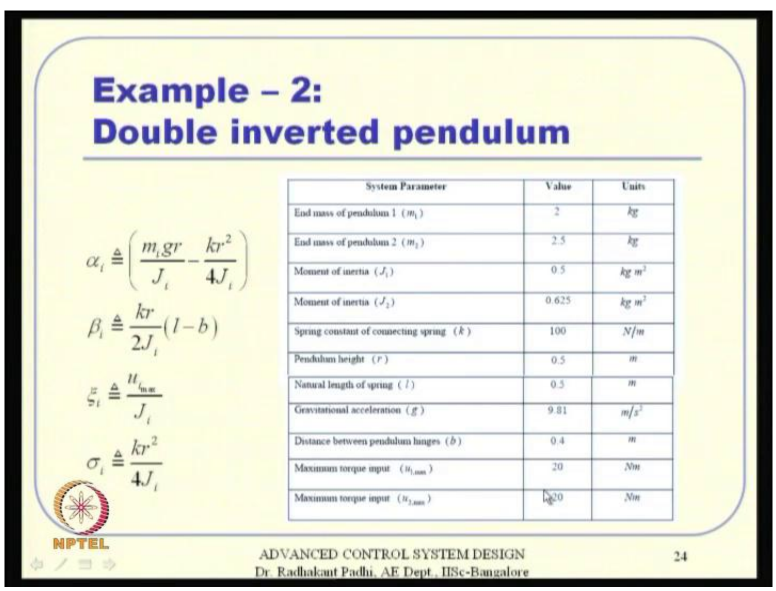
(Refer Slide Time: 35:04)



Now, we will apply this controllers u 1 and u 2, which are nothing but talks out there in harmonium such that, they will stabilize. So, can we do that and the nominal plant equation, here all like remember second order system and you have only one one control. So, we have; we will end up with the fourth order system dynamics with only two control X 1 c 1 and u 2, so this actually a system dynamics, where number of states are not equal to number of control. In reality, it is also a non affine system, because ten hyperbolic you appear. So, control does not really appear in a affine form, but I mean somebody can always argue that tan h of if this is a case, then I will assume that, this particular is like something like V1 and this particular is something like V 2.

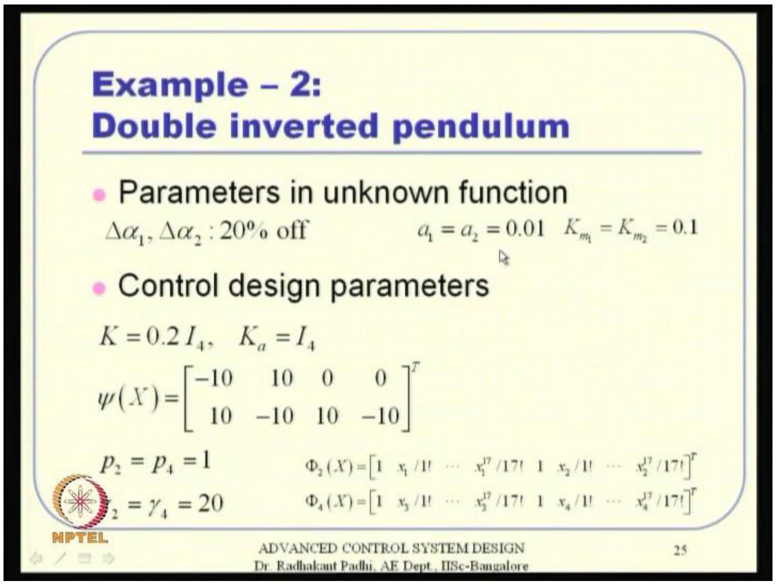
So, at least in V 1 and V 2, it is control affine. So, that problem is not that severe here, but still so, what you are assuming here is, let us assume that, this alpha 1 and alpha 2, which are parameters, where inaccurate with delta alpha 1 delta alpha 2. So, these values are not known to us and we also remember this entire design is not only parameter inaccuracy, we are also talking like inaccuracy in the system dynamics itself.

(Refer Slide Time: 36:14)



So; that means, we will let us assume that, we also have this in accuracy in the system dynamics itself and this inaccuracies are actually premising, because these are exponential terms. So, if you are adaptive control is not good, then these terms will actually take you away very fast, these are stabilizing exponential terms. So, in this system dynamic parameters and all that given here, as I have actually taken from a reference and the alpha i beta i is and all they define like that way. So that, the system dynamic appears in a little be the form.

(Refer Slide Time: 36:50)

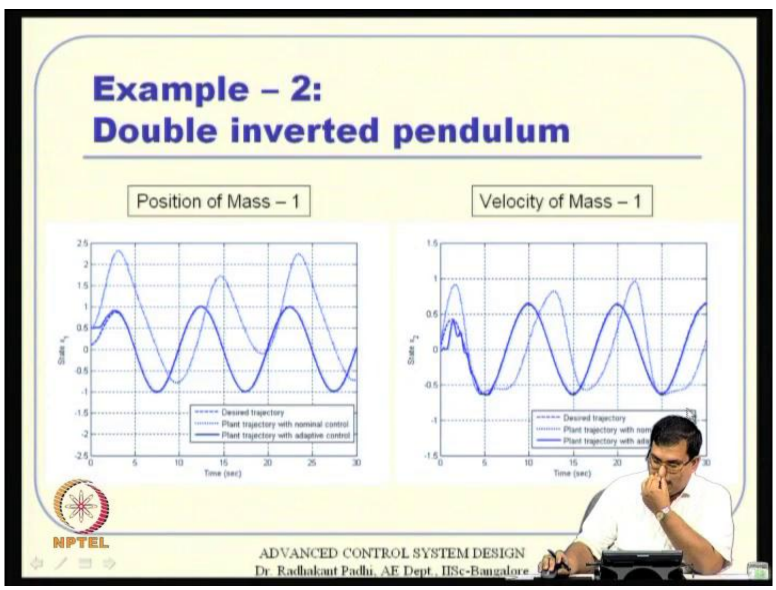


So, this alpha 1 alpha 2 and all that are not arbitrarily selected, they have the system dynamics meaning, these are related to system parameters really alright. So, these are like what we are assumed is like some delta alpha 1, delta alpha 2 can be 20 percent of and this exponential function K m 1 a one and all that those coefficients are also same I mean some values like this. Control design parameters are all like these are typically selected by sort of thing, but if you as you keep working on any problem will gain more and more experience also. That means, one or two problems if you solve the third problem will not appear that in that difficult anyway.

And here we will also change that, while designing this basis functions will become slightly more kind of, let us say smart or something. So why, because this is a particular sort of basis function, if you see, these are coming from Taylor series expansion. So, if you see this exponential term and all that, if I expand this e to the power anything, it will throw me some sort of exponential series. So, and sin X is also like that whatever happens here. So, using those things are little bit inside. So, using that we have selected that, but there is nothing like you have to select this. May be you can also select some of the thing that you want to.

But remember the more and more known information that you throw into the design, the lesser and lesser transient it will have actually. So, that is usually my recommendation, whatever you analyze the problem very well and whatever is a known information you want to pump into that, that way you will not lend up with another problem. This is also remember there is add and subtract term here, because this problem is not square remember that number of states are different from number of control. That is, in that sense, you will require psi of X times u s that should have control that you require sometime late. Then, this particular problem, we selected simply here constant matrix so, that will make the problems there.

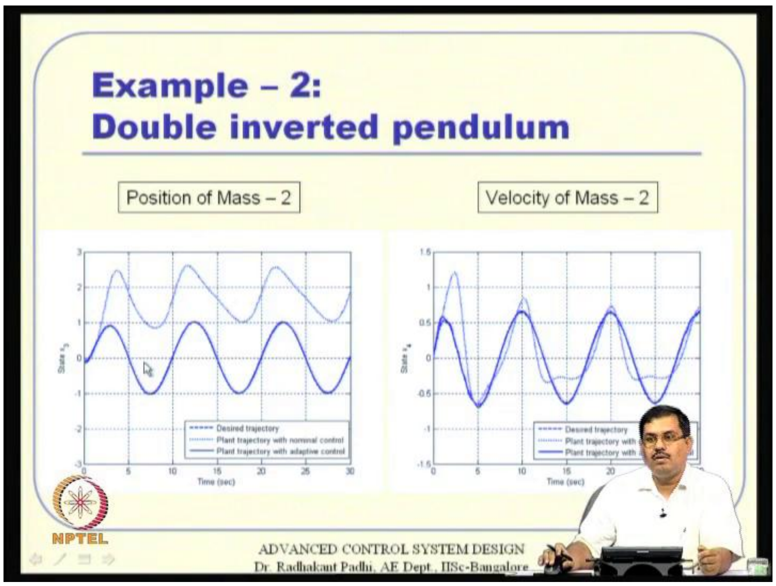
(Refer Slide Time: 38:32)



So, with those results, I have something like that and this is what you see is something like these dotted lines and all are what you really require for X 1 and X2, but if you simply apply nominal control, then it will not going to happen like that. Remember, when you are talking about nominal control application here, it is the nominal control formula as evaluated by the actual state. It is not just the nominal control u d is not a function of X d anymore. If the same function, what instead of X d, X will go there, because that will make the system dynamics operate in a feedback loop we are talking. If u d is a purely a function of X d then it is an open loop control so, we do not want to evaluate with that. You want to evaluate with close loop control so that means, nominal control formula as evaluated by actual state. So, that is what, you test the system really good.

By the way, if you simply use u d as a function X d, this dynamics that you see here will actually goes on stable, that is, lot more penalizing any way. So, the thing is it is nowhere close, what we desired? I mean, both magnitude by phase wise and all that it is a way and all that so. But if you apply the; this adaptive control, initially there is some sort of a transient behavior, these are not very close it will take some time to adapt both X 1 and X 2 is even more. Remember X 2 is a dynamic level, X 1 is kinematic level, kinematic level smoothness will open already, but X 2 is the dynamic part of it where parameter will make you these and all are directly reflected in the second and fourth component.

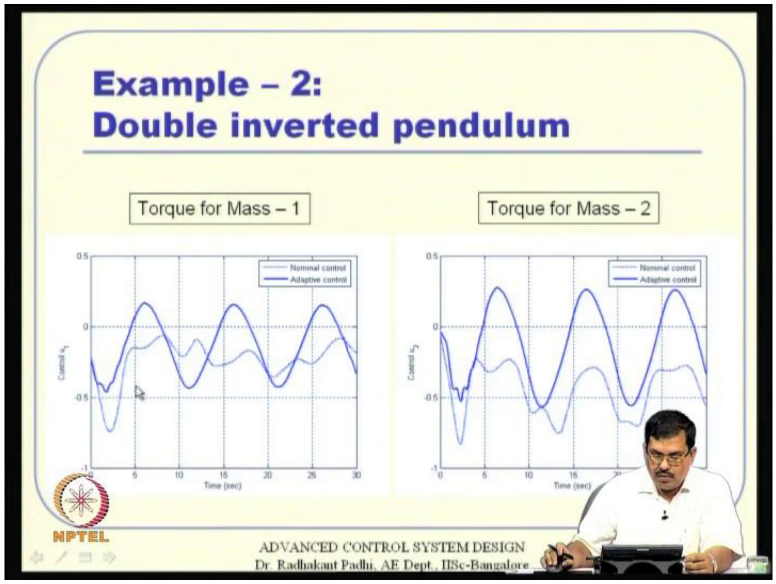
(Refer Slide Time: 40:18)



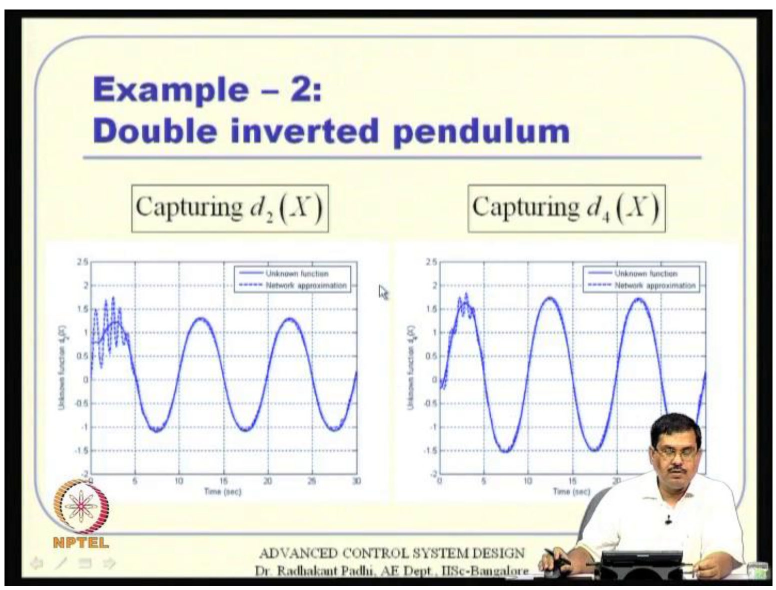
The second and fourth component of the states, you will see lot of more transient compare to first and third, first is here and third is here, these are integrated effects. So, we will not have too much difficulty there any way. So, the point here is when if you do not apply adaptive control things are not good, but if you apply adaptive control, very quickly you will adapt.

And later on we realize that there is almost like no difference between the actual system behaviour to the ideal or nominal system behaviour. So, that is the point.

(Refer Slide Time: 40:42)



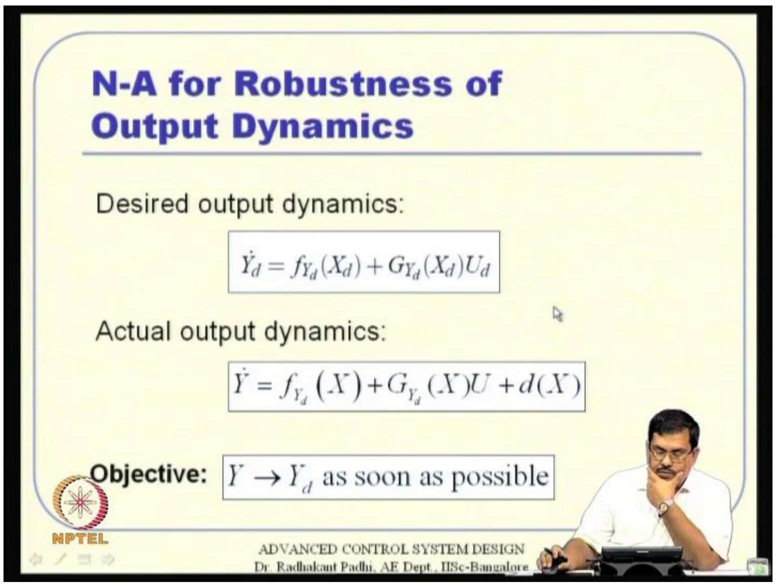
(Refer Slide Time: 40:58)



So, control behaviour you see that, the nominal controller and adaptive controller are very different and that is what, it is enforcing good behavior, adaptive control I mean enforce good behaviour there. How about capturing of d 2 and d 4, because these are things where there is unknown functions and all that. So, we see initially there are lot of transient, but afterwards, it will try to go along with that. Wherever there is a large change of derivative, again it will excide some sort of transient remember that. If you see the; if you amplify this peak and trough points, wherever there is a large sense of curvature derivative and all, it will not able to follow that.

Because the entire enforcement is based on first order derivative say what Lyapunov theory is all about first order derivative. So, when you have second order derivative is being large and second order derivatives are curvatures right. So, the curvatures being large, you will not be able to do. So, it will try to immediately adapt again and then try to follow there so that, that is the philosophy there. So, that is how it will happen there.

(Refer Slide Time: 42:22)



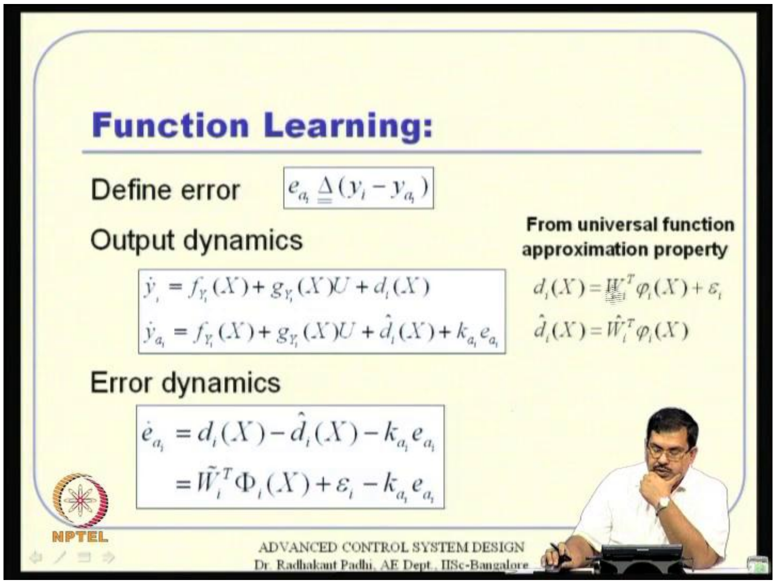
Now, this is all about, what we; what is there for the basic part of the design. Now, some sort of like further revisions, modifications all that, many times I recommend that, we implement this in terms of output robustness. So, we will not worry about the state robustness directly and if you are little bit clever, in the sense of you know the system dynamics very well. And if you know the objective very well think like that, this will typically actually. And what you are doing here is, instead of worrying about Y should go to

I mean X should go to X d and all will take some sort of output variable Y and make sure that, Y should go to Y d instead.

And why do you do that? Two reasons, one thing is the control effectiveness in Y should be very high compare to any other. Suppose you split that X actually in terms of let us say Y first and then, some other things here and the control effectiveness in y should be much higher here compare to other things. So, if you go back to the Y dynamics part of it, then this Y what you are talking about is nothing but the p q r dynamics. That means, the rotational Y dynamics, that is, where the control effectiveness is higher. So, if you want to make sure that, your control is doing a good job in the in p q r loop then, any other loop, that effect will not be that. And typically in the outer loop, I mean the outer part of it the U V W and other part will come there and that is where the guidance correction will also happen. So, that will also happen in a feedback sense.

So, and the; and when you guidance and control when you are together on a and things like that then things will not be that that period. So, you adapt your control in the inner most loop only and leave out to rest of inaccuracy to the guidance part of it. So, that is why, will be able to handle problems and why it is there, because by selecting a subset we will also make sure that dimension of Y is equal to dimension of U. That we are typically, if you see p dot, q dot, r dot are three variables and three control X are available to us. So, we will not be worried about those factors like that, add and subtract and the difficulties and all will be avoided.

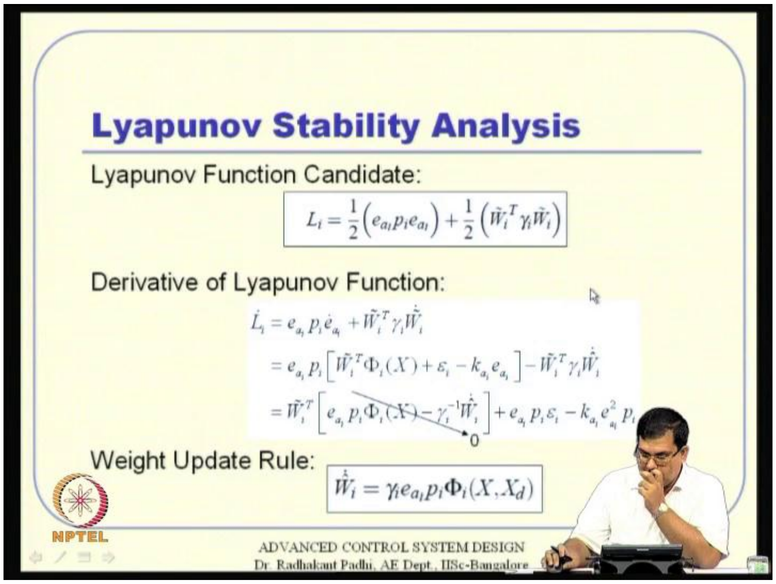
(Refer Slide Time: 44:38)



Ever then, that rest of the things are very similar, you can consider Y a dot is similar K a Y minus Y will be same and Y should go to Y a to Y d and samething. Y should go to Y d first and then that will be enforced like a earlier sort of thing. And if it is control Affine, we will be able to solve it, that way if it is not, then and we wrote the lyapunov equation. We defined here Y minus Y a and then go through this ideal neural network actual neural network all the exactly same parallel thing and then we will be able to tell d i is; d i of X is like this, then d hat of X is like this.

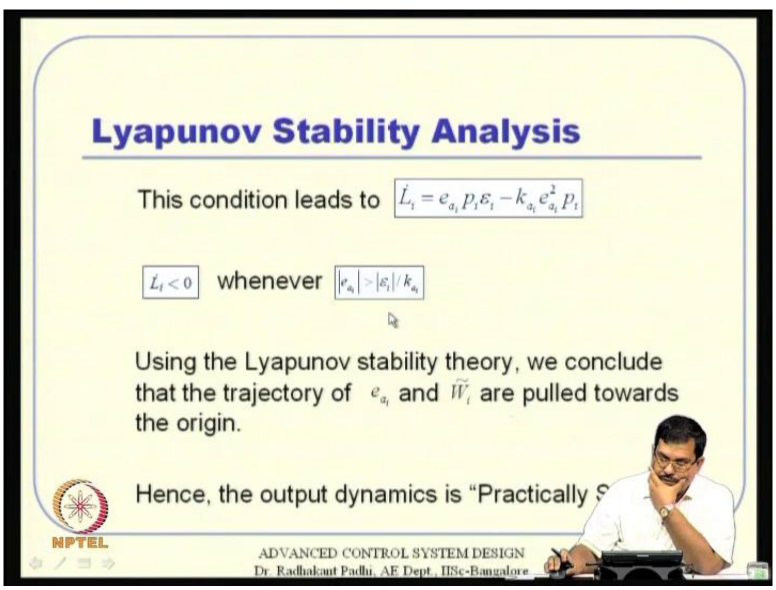
Remember even though, it is a Y dynamics the inaccuracy part can be a function X a actually. So, then the basis function will should be a function of X rather not necessarily only Y. You may try out that with Y only Y, Y a all that it may or may not work out that.

(Refer Slide Time: 45:08)



Now, aerodynamics is again same, whatever we have discussed before and then Lyapunov function candidate is very similar what we discussed before actually there is I think the small print mistake there, so the gamma i inverse actually. So, then you follow this gamma; in this is gamma inverse again that term and that mistake is already been corrected next line. So, this is gamma inverse anyway.

(Refer Slide Time: 45:52)

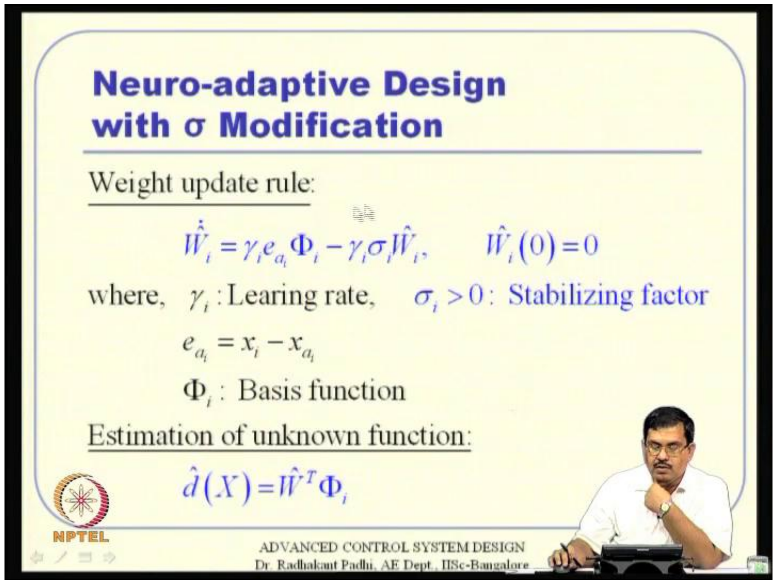


So, you do by exactly same thing, you start with that and then and E a i dot is something that you have derived here. So, put it back into the equation basically and then talk about simplification and then tell this is the cofficient. So, let me make sure that, that is 0, then we left out with that term and again this shows out the same sort of a equality there. So, instead of assuring that, all my state should behave like desired state and think like that like a nominal state. here. I am only worried about my I mean Y vector that I am selecting, that should go to Y d actually and most of the time our experience that is actually not a very bad thing to do. We have experimented several problems, when it turns out, it is a very good thing to do. As long as you operate that with the guidance loop, entire thing as you operated with the guidance loop in extent basically alright.

So, again we will end up with something like practical stability and all that actually. Now, here is a big difference, when we started developing all that anything like that, there was some objects from reviewers tell that this entire thing should be not having a stabilization term. I mean, if you see this, let us say go back to this weight update rule in general, this weight update rule is something like X dot equal to like some f of U sort of thing. However, so, W hat dot, what we are talking here? And right hand side is there is no w; W hat term actually. So, it is like the X dot is not a function of X it is a function of some other variable actually. So, it is strictly like an enforcing system strictly like a time varying and that X dot equal to like the sort of thing that that way.

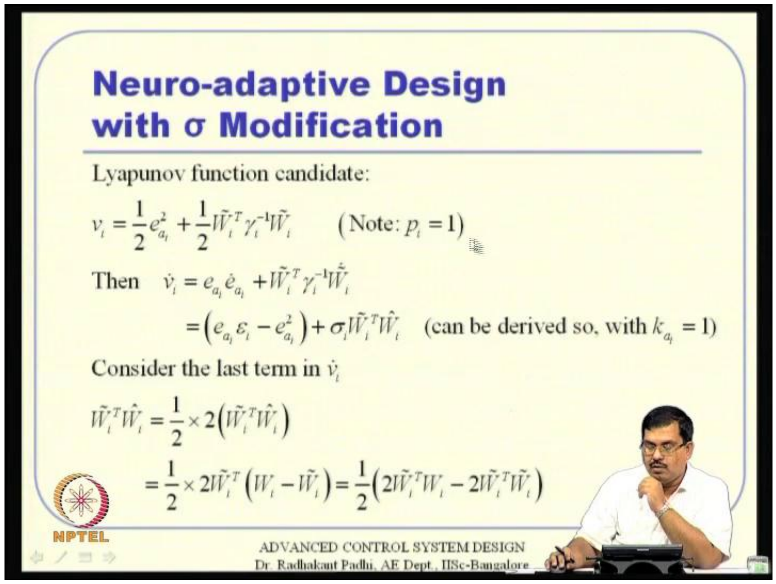
So, what happens here is, the stability analysis we come to difficult actually; that means, whether W hat will ever remain stable or not is not guaranteed here. and that essentially excides an issue of this parameter drift in adaptive control, in adaptive control theory there is an issue of parameter drift. And W hat is nothing but parameter, because W hat is something that, we are actually identifying d hat. So, W hat is nothing but the parameter for d hat and because it is d hat, it is also coming into the system dynamic, because d hat comes here. So, approximate system dynamic there is a w hat term here. So, W hat is a parameter and that parameter is adapted and the during the adaptation process that can drift away actually; that means, it can go to infinity sort of thing.

(Refer Slide Time: 48:44)



So, to avoid that there is a stabilizing term in one of the ideas are sigma modification term and all that. So, we are kind of twelve to do that and then we actually wanted and try to modify entire derivation based on that. So, let us say, I will quickly some sort of overview of that and then we tell you that there. So, what you do here is, with sigma modification term all that you are proposing here is, we have additional term like this. So, it is not just to this one, gamma e a I, phi i 0 actually sorry; p i is 1 here with the assumption that, I think we have; I have written on somewhere.

(Refer Slide Time: 49:02)



P i is some I mean; this entire exercise is we have assumed that all p i's are 1, I mean this is relative again whether you have p i term here or not and gamma here then, you have to fill it both. So, you can you can fix one as 1 and select the other one. So, that way, we select p i equal to 1 everywhere here. So, with that, it is like this term was originally what was there in here and we are in I mean now, we are telling, let us have this additional term out here. Remember gamma and sigma both are positive, gamma is any way positive, sigma is also positive term, which is something like a stabilizing fact.

So, what is happening here like. So, for whatever reason, suppose this term keeps on happening to be larger and all that actually; that means, these are ultimately non zero, like because of some control saturation or something happen then e a i is never zero. So, this is a forcing term, all the time W hat; W i hat dot is positive quantity, then if this is happens to be all those always positive quantity, then this supposed to grow actually. So, in that situation what will happen is even if you small quantity for sigma, very small number of sigma. Then ultimately W hat, if it is larger, then it will not enforce it to grow even further. this term is actually negative quantity, this x dot equal to like minus x sort of thing. So, that is actually stabilizing term. So, if this happens to be larger and this will this will give stability to that.

But now, if our experience shows that is a way great problem primarily, because perhaps that we should have, I mean we have never encountered like control and other issues. If adaptive control is there and control is saturated on the way, then this issues can I mean actually and if it happens in your problem. And what whatever problem you want to experiment, then it is better to have this term. Remember in general, synchronize very small quantity, something like 10 to power minus 6, minus 8 like that that kind of small quantity. Only to assure that this one over this quantity is grows large, very large, then it will not go to infinity. So, that is a primary factor primary reason for that.

So, parameter drift really a issue will be control and if it happens to be very small quantity anyway W hat though this term as if it is not there. So, that way it is not going to contraversary. Well, because of this little additional term, you think can be much simpler and think require too much of modification all that not to actually. So, saying that everything will remain stable from the Lyapunov theory was lot of exercise and let us see that. So, we will go it to the Lyapunov function candidate, let say, instead of L I, we will talk about V I, I mean this is one of the same thing we did. So, V i is something like this with p i equal to 1 and I also assume that K i equal to 1.

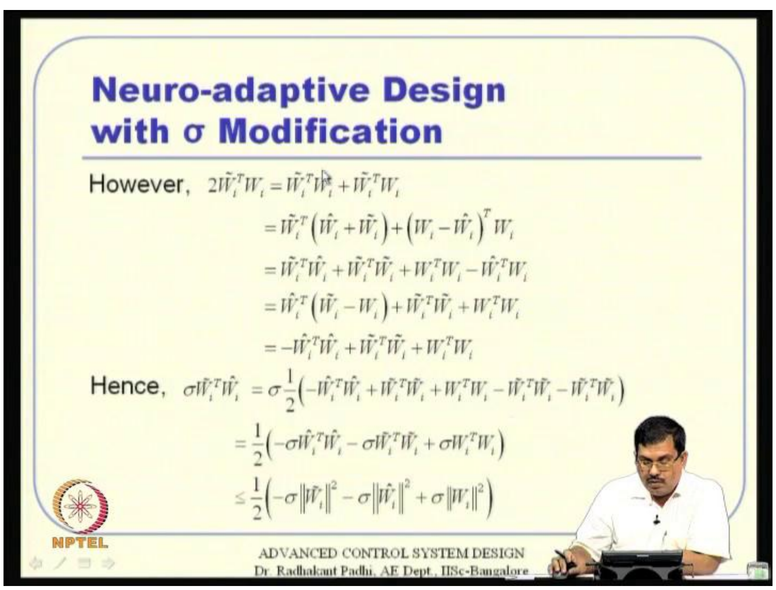
For simplicity, let us say that, K I, K a is again which goes to X a dot dynamics, in case of that, that is a diagonal element. So, all diagonal elements let us assume that is 1. So, with that assumption that turns out to be like that, this is V i is if it is like this, then that happens to be like that. And then with this with this equation that we are proposing here W hat is like this W hat sorry; W i hat dot is like this. So, W i tilde dot is actually what you I mean we all know that W i tilde is W i ideal weight minus W i hat. So, W i that dot is equal to this dot minus that dot and this dot is zero, because that is an ideal. So, it is minus W i hat dot and this W i hat dot is available to us already right.

So, we substitute this expression here, V i dot is already available to us before I mean that, the same expression that we have before. So, using all that and try to simplify little bit it will turn out to be like this. So, now, we have to caught up with a term to analyze something, what is happening really here? Now, remember this is after putting the weight update rule and if you go back and see that after putting this weight update rule, in previous case, we are left out with only this, if it is only this, then analysis are very straight forward actually right.

I mean if you let me just quickly go through that, here after we put all that we are left out with only this quantity, if I only this quantity, then this is very straight forward. Now, it is not going to be like that. So, we have to carry lot of analysis to address, what is really happening to the extra term, this extra term, where it is going? What it is giving often all that? So, let us try to analyze this extra term. So, before doing that let us analyze only this term here W i tilde transpose time W i hat. So, this one I can put it one way, 2 into 2 and all that and if I do that this into 2. Remember W i hat is again I put that expression W minus W i tilde basically, because this W i tilde by definition is W minus W i hat right. So, if i put W i hat this one is equal to this other side I am doing that and all that actually from using this relationship, I am doing that actually that way.

So, now this quantity I will put it inside, if I put inside, I will left out with two times that two times that and then I will carry further this exercise instead of writing all that, we will this is like a quadratic term already. So, we will not bother about so much on that, this is actually a quadratic term, W i tilde is there, W i tilde is that. So, this is W i tilde norm sort of thing.

(Refer Slide Time: 55:02)



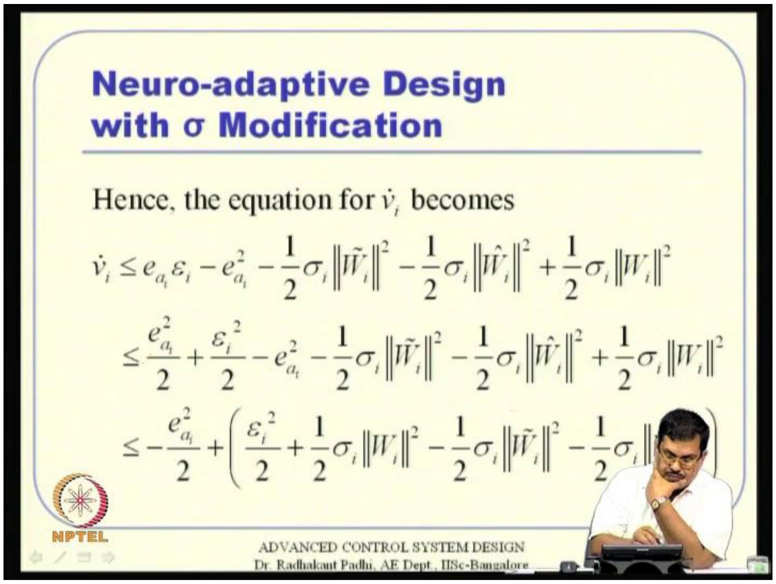
So, but what about this term? This is not quadratic term. So, we have to analyze little further on that. So, this term is where I will write it as something this one plus this one and then this particular thing will expand it again. And then try to, I mean expand the bracket and then simplify as much as possible and we will try we will lend up with something like this. So, sigma times this, what you; what we are really interested in, then we analyzed to be like that and then you can say there will be some sort of a plus minus cancellation quantity and thing like that. Ultimately, you can directly is come from here to there also.

What is happening here? Instead of a non quadratic term, that we started with this is, this is actually a nonquadratic term remember that we are writing that as once a quadratic term actually somehow. So, what the problem here is, this is not all are not with the same size. There is one two I mean two are positive sign one is negative sign and those you need to remember. So, if I all that, there and then do that proper algebra and thing like that, I will ultimately, I will end up with remember this is minus of that term, what you are analyzing this is positive of that term. So, all that thing has to be taken into account while analyzing this V i dot.

So, you put that and if there is any algebra mistake out here, you can also think of doing that exercise yourself and correct it out. And then I mean what it message here is this entire quantity, that we are bothered about in the in this part of it actually, turns out to be something like this. And with this particular quantity is actually less than equal to this quantity now I mean if I talk about norms and all that. Quadratic quantities, I will put that in a norm sort of a norm quantity and tell what it should be. And all this exercise by the it is all given like this, is the paper that I am talking here, this is like appear in this (( )) and all 2007 for all this things are also available here.

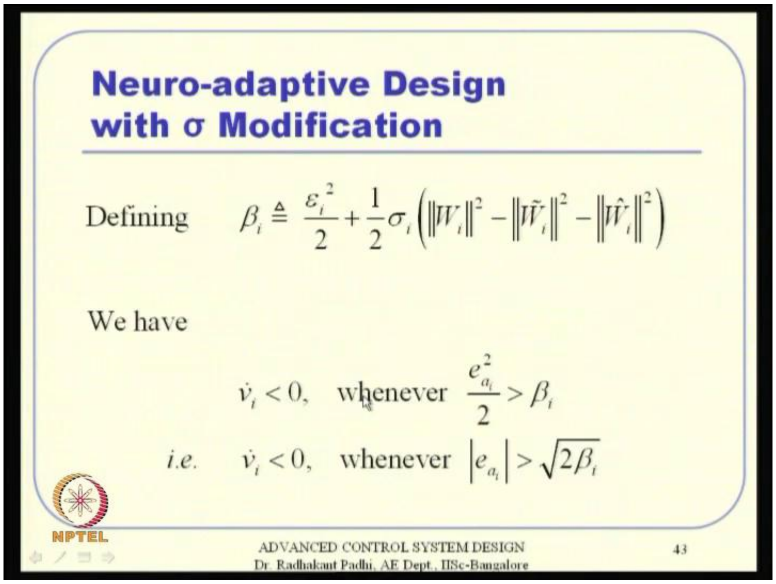
So, let me go back quickly, this entire exercise is what I am talking actually, all that things are available. So, if you I mean only mistake probably something like this should be W tilde dot sort of thing, it is not W hat. So, if you anybody wants to like read this, you can read this paper alright.

(Refer Slide Time: 57:40)



So, we are left out with something like this quantity and this quantity once you start putting everything everywhere. So, earlier we had only this quantity, now, we will have all this additional quantity. For this quantities can be combined together and thing like that. So, once you combine them, then it turns out, I will be able to combine it, let us say something like this.

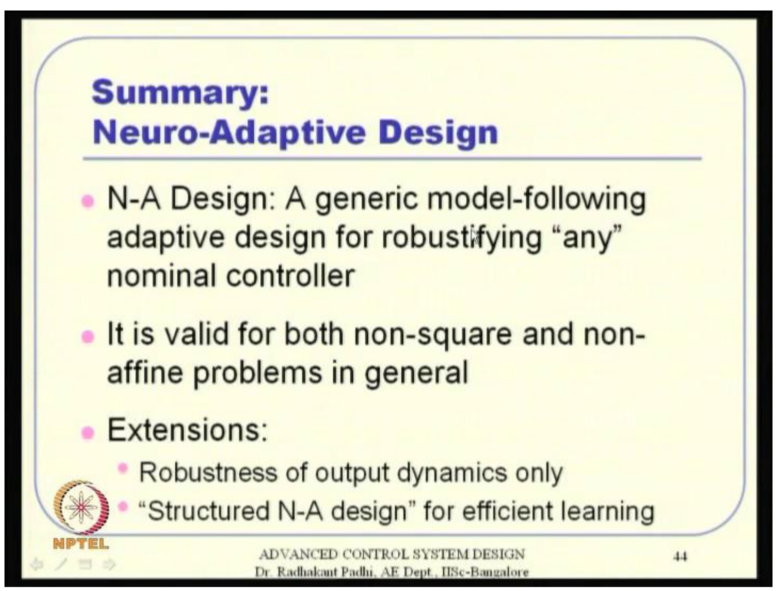
(Refer Slide Time: 58:10)



So, what is happening here? So, this V i dot is supposed to be like let me define this quantity as beta e sort of thing now. And with that definition beta I, we have V i dot is negative, whenever this, because you are left out with this negative quantity with this beta e term. So; that means, this beta e negative, whenever this is greater than equal to beta e whenever strictly greater than beta this is strictly less than 0. So, V i dot is strictly less than 0, whenever this is, this condition is satisfied. So, again we left out with a with a meaningful condition, where this we talk about norm of this error quantity obsolete scalar value.

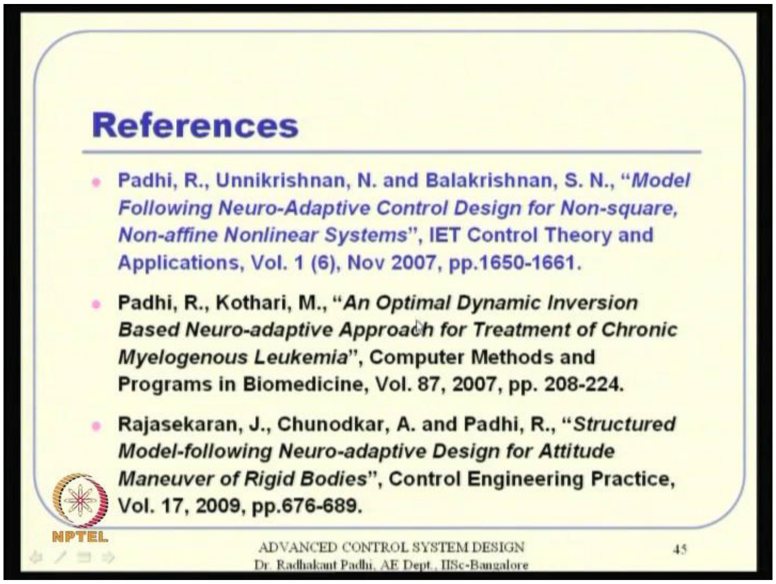
So, it is obsolete norm of error quantity is greater than this particular quantity and this quantity is also given something like this. And remember this W, what we are talking here, W is actually in general this is W remember W tilde is W minus W hat. So, W is equal to W tilde plus W hat. So, that to really speaking, if you have W like this so, then this is W weight let us say and W tilde let us say. So, what we are telling is, this I mean this length norm; that means, norm is vector quantity anyway. So, this norm minus this norm, minus that norm. So, it is a very small quantity, the beta; the definition of beta actually epsilon is anyway small quantity, because ideal error function approximation error. So, this is small quantity and this as long that this error norm is greater than this small quantity, then V i dot is negative. So, that is how it operates it.

(Refer Slide Time: 59:38)



So, with that modification, it was kind of complete and so, summary part of it in the entire design. It is a generic design with robustifying, it is capable of robustifying any nominal control, this is valid for both non square and non affine problems in general. And we have done others the modification, what is kind of structured neuro adaptive design for better efficient learning kinematics and dynamics parts done separately.

(Refer Slide Time: 59:58)



For references, largely I have taken from the first reference, but other references are also available. So, with that I will stop. Thank you.