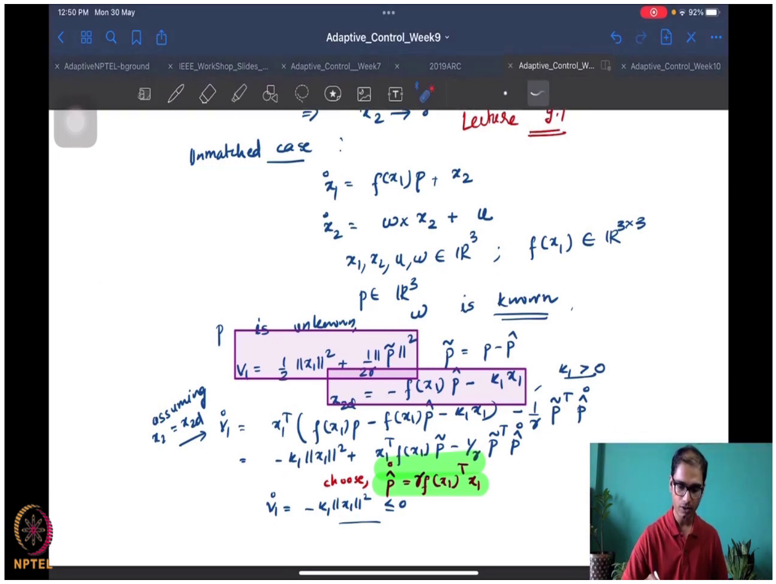
Nonlinear Adaptive Control
Professor Srikant Sukumar
Systems and Control
Indian Institute of Technology, Bombay
Week 9
Lecture No: 50
Extended Matching Design: An Example

(Refer Slide Time: 00:17)

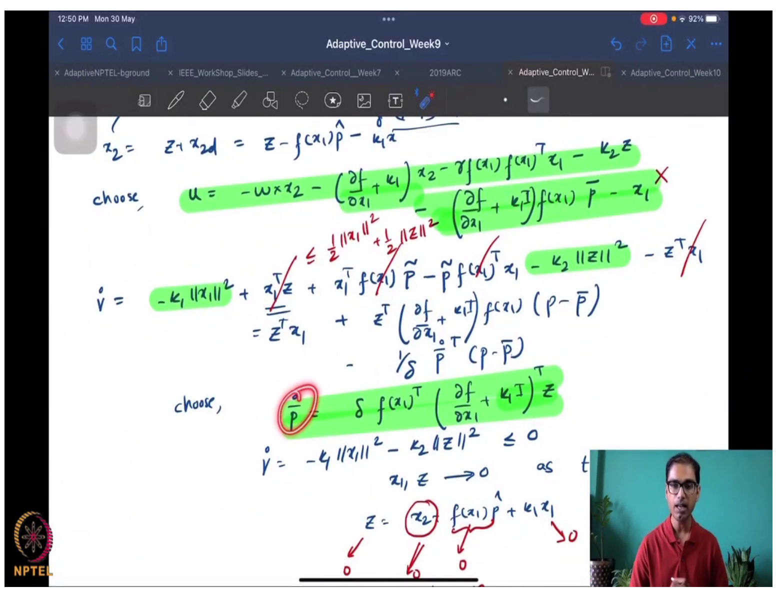
Hello, everyone. Welcome to yet another session of our NPTEL on Nonlinear and Adaptive Control. I am Srikant Sukumar from Systems and Control, IIT Bombay. We are into the ninth week of this NPTEL course on non-linear adaptive control, and I think by now, all of you have gained sufficient expertise and knowledge in order to be able to tackle adaptive designs for uncertain autonomous systems, such as the satellite that you see in the background.

(Refer Slide Time: 01:02)



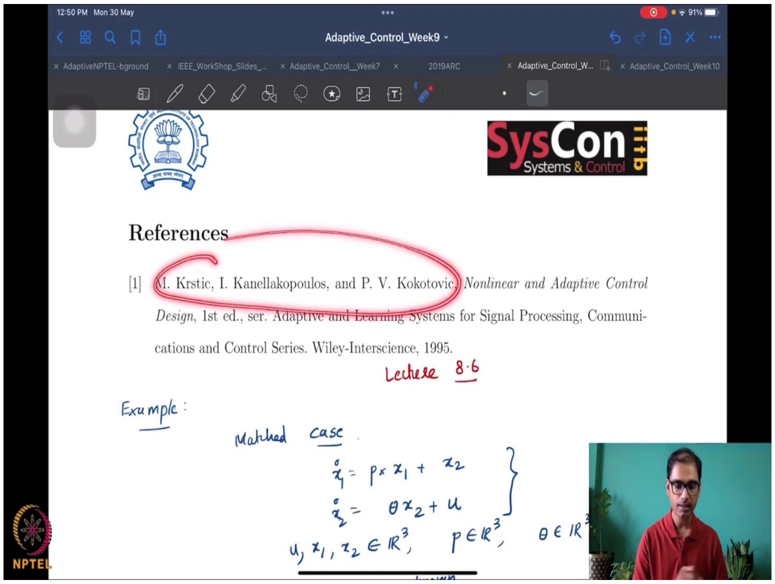
So, in the last week or on the last session we were still working on our unmatched design and what we had seen was essentially a cooked-up example for which we had these unknowns. The unknown was unmatched therefore we assumed that the p is unknown. And for this unknown p quantity, we actually designed an adaptive integrator backstepping based design.

(Refer Slide Time: 01:38)



So, we came up with two different parameter estimates, p hat and p bar with update laws and a feedback controller in order to stabilize the system, that is drive x 1 and x 2 to 0.

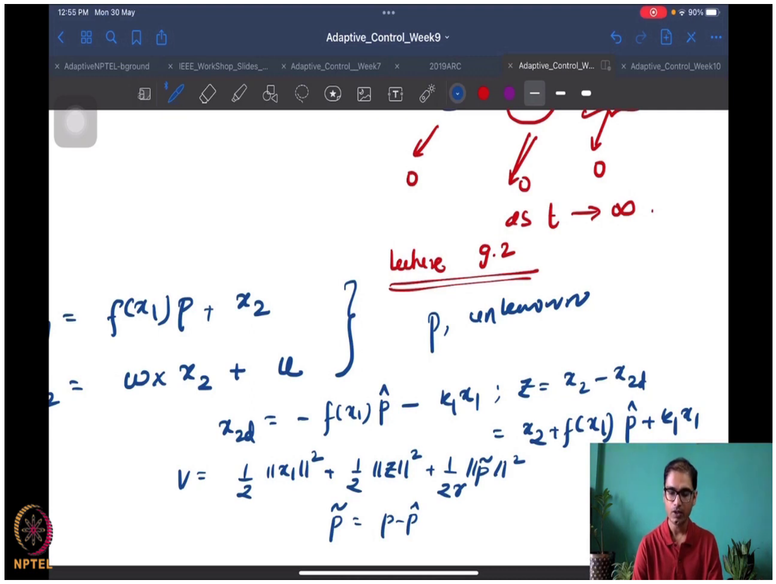
(Refer Slide Time: 01:58)

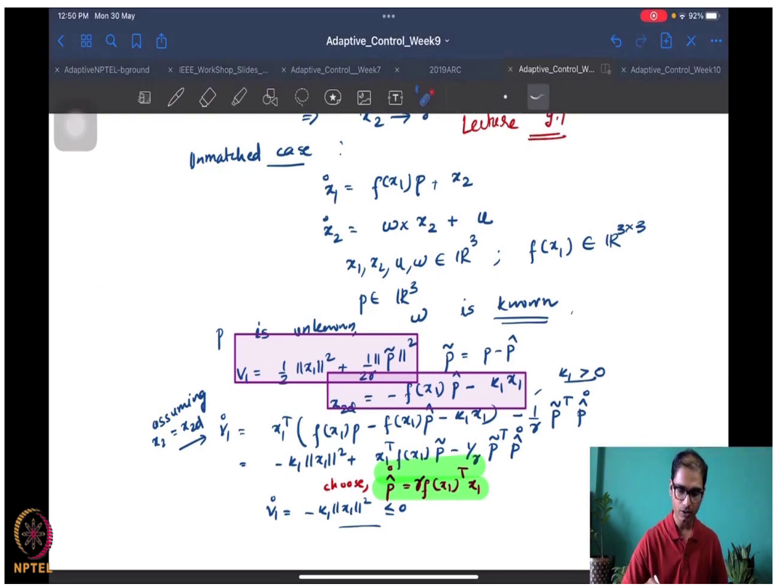


So, we continue to look at these stabilization problems. The tracking problems are, as I mentioned not significantly different. I would strongly recommend all of you to look at the K. K. K. book, that is the Krstic, Kanellakopoulos, Kokotovic book for more details on how to do this. So, this is rather important, that that all of you do use these book references. I am after all constrained by whatever time I have, and I cannot cover every single thing that the book contains.

So, the book is definitely more advanced material and extensions of whatever we are discussing in the class, and so, I would strongly urge all of you to look at the book because that is what will give you more handle on how to solve the more real, more practical problems. Although, whatever we are discussing here in the course itself is sufficient for you to be able to deal with a lot of practical systems.

(Refer Slide Time: 03:07)





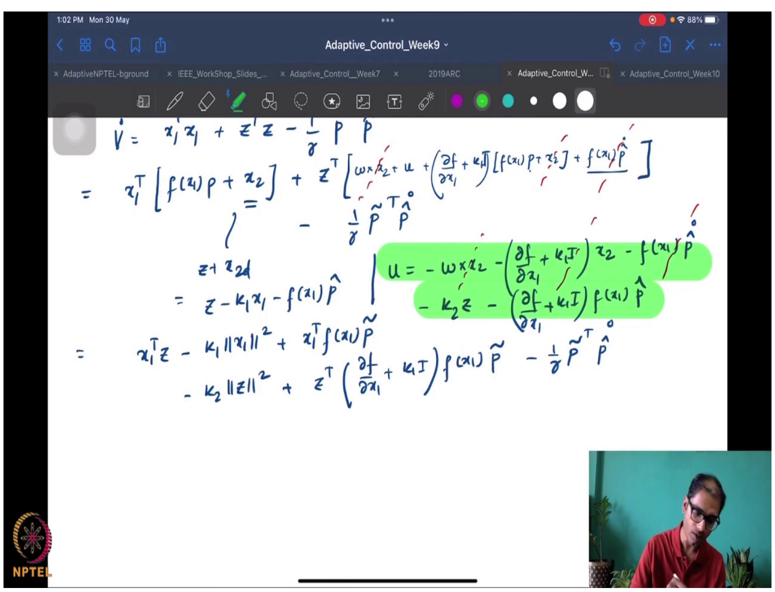
So, what I want to do now is that we understand our issue with the current design, that is we have to create two different parameter estimates corresponding to one parameter. And so, we want to alleviate this with the extended matching design. So, how do we do this? so, let me start here. What I am going to do is, first I am going to label the lecture number as lecture 9.2, and we start working our problem. So, I am going to rewrite the system, say, quickly, but I am going to cheat a little bit and copy. So, what do I do? I just paste it here. And for this system I know that p is unknown, of course. p is unknown. p is assumed to be unknown.

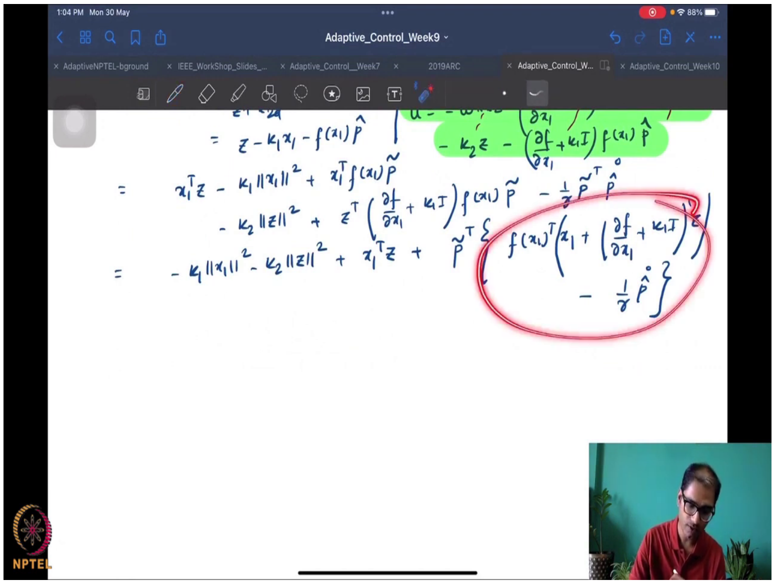
So, the first thing that we want to do is that we do not declare a first state, a first stage Lyapunov candidate, but we do define an x 2 desired which is essentially the same. If you see the x 2 desired, that is exactly the same. It is going to be exactly this guy, sorry, the negative of this guy. So, the x 2 desired is just motivated by the dynamics. You just try to cancel whatever you can and introduce an estimate. So, that is exactly what we do. So, x 2 desired is the same and so, that is going to be minus f x 1 p cap minus K 1 x 1, introduce a good term and try to cancel, as best as possible, the bad term.

So therefore, the estimate gets introduced. But notice that we are no longer going to declare a candidate Lyapunov function V 1, we are not going to declare, we are not going to do Lyapunov analysis to come up with the p hat dot yet. Instead, what do we do? We define the z, again, exactly in the same way. So, that is what I will do next. I will declare the z as x 2 minus x 2 desired, which is actually equal to x 2 plus f x 1 p cap plus k 1 x 1. So, this is exactly the same as before. Now, once I do this, so, now I declare my V. I declare my V not like this because I did, I do not have any V 1 at all, so, but in a rather simpler way, I would say.

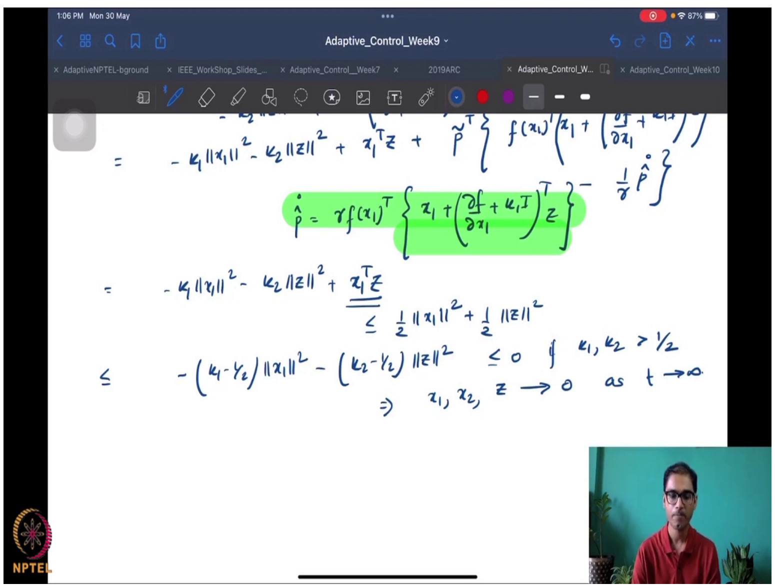
I declare my V as one-half norm x 1 square plus one half norm z squared plus 1 over 2 gamma norm p tilde square, where p tilde is p minus p cap. So, I never introduced a second estimate, I did not design a V 1, the first stage Lyapunov candidate, I did not already define a p hat dot. All of that will be done using this candidate Lyapunov function.

(Refer Slide Time: 06:50)





Adaptive_Control_Week9
$= z - k_1 x_1 - f(x_1)\hat{p}$
$- k_2 z - \left(\frac{\partial f}{\partial x_1} + k_1 I\right) f(x_1)\hat{p}$
$= x_1^T z - k_1\|x_1\|^2 + x_1^T f(x_1)\tilde{p}$
$- k_2\|z\|^2 + z^T\left(\frac{\partial f}{\partial x_1} + k_1 I\right) f(x_1)\tilde{p} - \frac{1}{\gamma}\tilde{p}^T\dot{\hat{p}}$
$= -k_1\|x_1\|^2 - k_2\|z\|^2 + x_1^T z + \tilde{p}^T\left\{ f(x_1)^T\left(x_1 + \left(\frac{\partial f}{\partial x_1} + k_1 I\right)^T z\right) - \frac{1}{\gamma}\dot{\hat{p}}\right\}$
NPTEL



Adaptive_Control_Week9
$= -k_1\|x_1\|^2 - k_2\|z\|^2 + x_1^T z + \tilde{p}^T\left\{ f(x_1)^T\left(x_1 + \left(\frac{\partial f}{\partial x_1} + k_1 I\right)^T z\right) - \frac{1}{\gamma}\dot{\hat{p}}\right\}$
$\dot{\hat{p}} = \gamma f(x_1)^T\left\{ x_1 + \left(\frac{\partial f}{\partial x_1} + k_1 I\right)^T z\right\}$
$= -k_1\|x_1\|^2 - k_2\|z\|^2 + x_1^T z$
$\le \frac{1}{2}\|x_1\|^2 + \frac{1}{2}\|z\|^2$
$\le -(k_1 - 1/2)\|x_1\|^2 - (k_2 - 1/2)\|z\|^2 \le 0$ if $k_1, k_2 > 1/2$
$\Rightarrow x_1, x_2, z \to 0$ as $t \to \infty$
NPTEL

1:08 PM Mon 30 May

Adaptive_Control_Week9

$$\dot{\hat{p}} = \gamma f(x_1)^T \left\{ x_1 + \left(\frac{\partial f}{\partial x_1} + k_1 I \right) z \right\}$$

$$= -k_1 \|x_1\|^2 - k_2 \|z\|^2 + \underline{\underline{x_1^T z}}$$

$$\underline{\underline{x_1^T z}} \leq \frac{1}{2}\|x_1\|^2 + \frac{1}{2}\|z\|^2$$

$$\leq -\left(k_1 - \tfrac{1}{2}\right)\|x_1\|^2 - \left(k_2 - \tfrac{1}{2}\right)\|z\|^2 \leq 0 \quad \text{if } k_1, k_2 > \tfrac{1}{2}$$

$$\Rightarrow x_1, x_2, z \to 0 \text{ as } t \to \infty$$

— x —

NPTEL

12:56 PM Mon 30 May

Adaptive_Control_Week9

$$z = x_2 - x_{2d} = x_2 + f(x_1)\hat{p} + k_1 x_1$$

$$\dot{z} = \omega \times x_2 + u + \left(\frac{\partial f}{\partial x_1} + k_1\right)\left[f(x_1)p + x_2\right] + \underline{f(x_1)\dot{\hat{p}}}$$

$$\underline{f(x_1)\dot{\hat{p}}} = \gamma f(x_1) f(x_1)^T x_1$$

design new estimate $\bar{p}$ for this term

$$V = V_1(x_1, \hat{p}) + \frac{1}{2}\|z\|^2 + \frac{1}{2\delta}\|(p - \bar{p})\|^2, \quad \underline{\underline{\delta > 0}}$$

$$\dot{V} = x_1^T\left(f(x_1)p + \underline{x_2}\right) - \underbrace{\tilde{p}^T f^T(x_1) x_1}_{\text{substituting for } \dot{\hat{p}}} + z^T\left[\omega \times x_2 + u + \left(\frac{\partial f}{\partial x_1} + k_1\right) f(x_1) p + \left(\frac{\partial f}{\partial x_1} + k_1\right) x_2 + \gamma f(x_1) f(x_1)^T x_1\right]$$

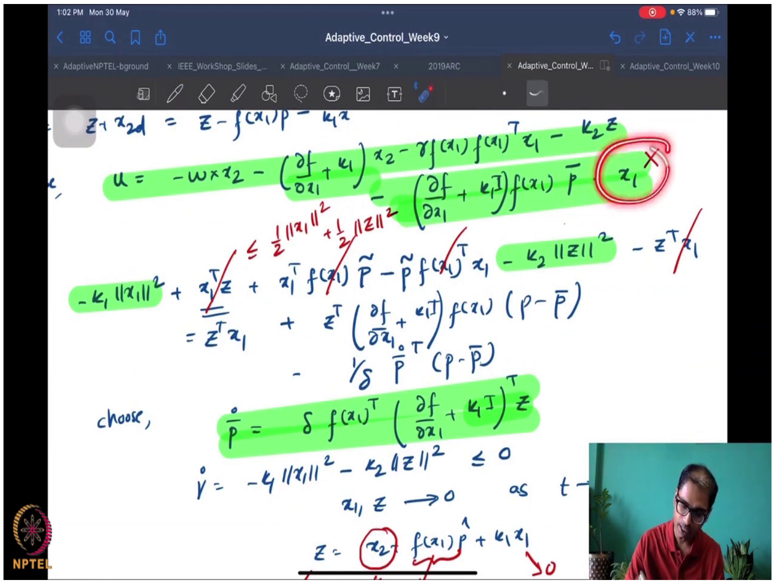
$$- \frac{1}{\delta}(p - \bar{p})^T \dot{\bar{p}}$$

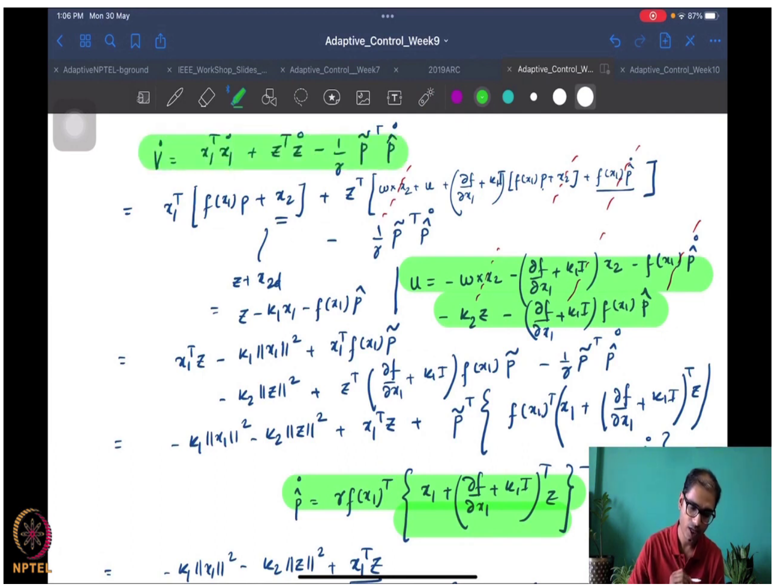
$$x_2 = z + x_{2d} = z - f(x_1)\hat{p} - k_1 x_1$$

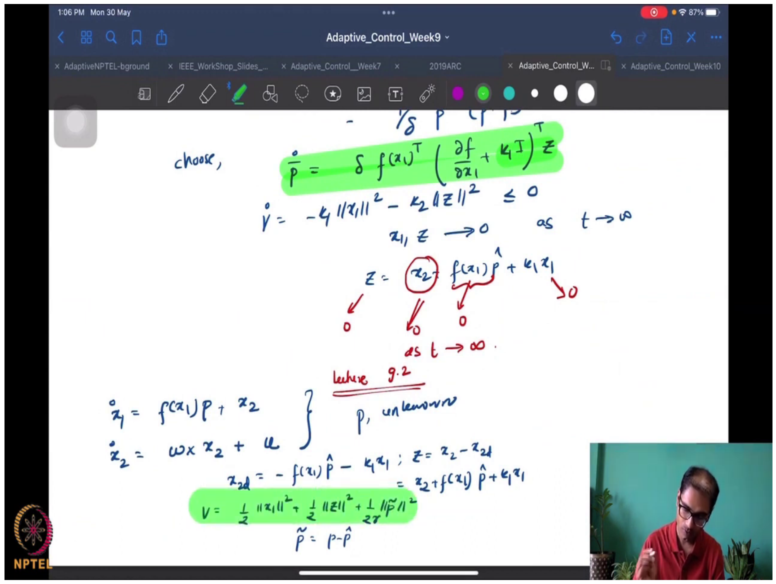
choose

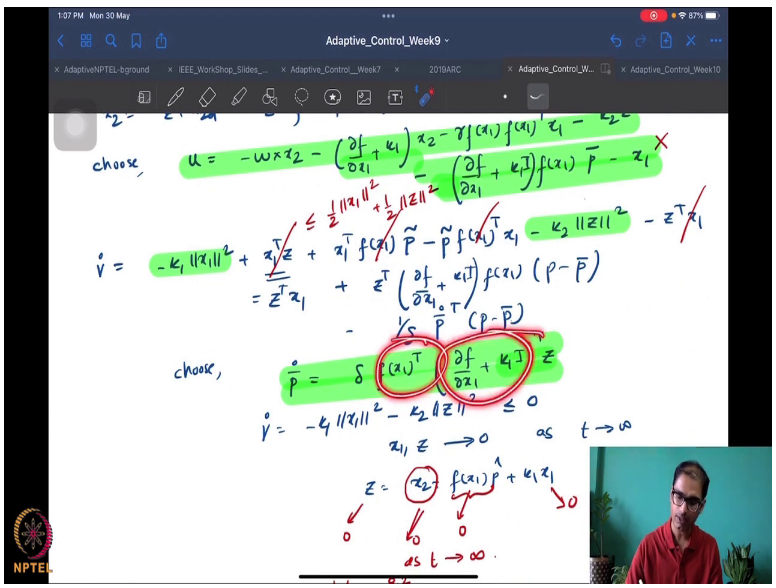
$$u = -\omega \times x_2 - \left(\frac{\partial f}{\partial x_1} + k_1\right) x_2 - \gamma f(x_1) f(x_1)^T x_1 - \left(\frac{\partial f}{\partial x_1} + k_1 I\right) f(x_1) \ldots$$

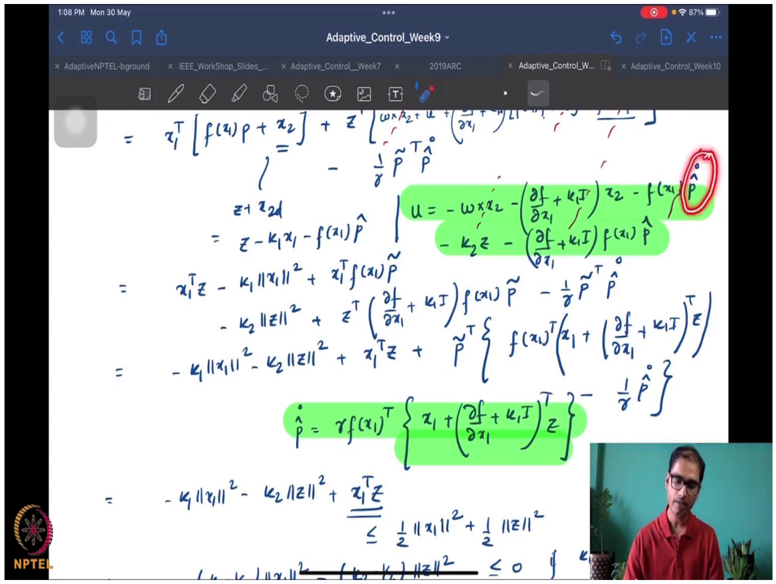
NPTEL


1:02 PM Mon 30 May
Adaptive_Control_Week9
AdaptiveNPTEL-bground
IEEE_WorkShop_Slides_...
Adaptive_Control__Week7
2019ARC
Adaptive_Control_W...
Adaptive_Control_Week10
choose,
NPTEL


1:06 PM Mon 30 May
Adaptive_Control_Week9
AdaptiveNPTEL-bground
IEEE_WorkShop_Slides_...
Adaptive_Control__Week7
2019ARC
Adaptive_Control_W...
Adaptive_Control_Week10
NPTEL


Adaptive_Control_Week9
AdaptiveNPTEL-bground
IEEE_WorkShop_Slides_...
Adaptive_Control__Week7
2019ARC
Adaptive_Control_W...
Adaptive_Control_Week10
choose,
p̄̇ = δ f(x₁)ᵀ (∂f/∂x₁ + k₁I)ᵀ z
V̇ = −k₁‖x₁‖² − k₂‖z‖² ≤ 0
x₁, z → 0 as t → ∞
z = x₂ + f(x₁)p̂ + k₁x₁
as t → ∞.
lecture 9.2
ẋ₁ = f(x₁)p + x₂
ẋ₂ = ωx x₂ + u
p, unknown
x₂d = − f(x₁)p̂ − k₁x₁ ; z = x₂ − x₂d = x₂ + f(x₁)p̂ + k₁x₁
V = ½‖x₁‖² + ½‖z‖² + 1/2γ ‖p̃‖²
p̃ = p − p̂
NPTEL


Adaptive_Control_Week9
AdaptiveNPTEL-bground
IEEE_WorkShop_Slides_...
Adaptive_Control__Week7
2019ARC
Adaptive_Control_W...
Adaptive_Control_Week10
choose,
u = −ωx x₂ − (∂f/∂x₁ + k₁) x₂ − γ f(x₁) f(x₁) x₁ − k₂z − (∂f/∂x₁ + k₁I) f(x₁) p̄ − x₁
V̇ = −k₁‖x₁‖² + x₁ᵀz + x₁ᵀ f(x₁) p̃ − p̃ f(x₁)ᵀ x₁ − k₂‖z‖² − zᵀx₁
≤ ½‖x₁‖² + ½‖z‖²
= zᵀx₁ + zᵀ (∂f/∂x₁ + k₁I) f(x₁) (p − p̄)
− 1/δ p̄̇ᵀ (p − p̄)
choose,
p̄̇ = δ f(x₁)ᵀ (∂f/∂x₁ + k₁I)ᵀ z
V̇ = −k₁‖x₁‖² − k₂‖z‖² ≤ 0
x₁, z → 0 as t → ∞
z = x₂ + f(x₁)p̂ + k₁x₁
as t → ∞.
NPTEL



So, let us move on and actually compute this V dot. So, V dot is going to be what? It is x 1 transpose x 1 dot plus z transpose z dot minus 1 over gamma p tilde transpose p hat dot. This is of course using the definition of p tilde. So, now if I substitute, I am going to substitute for the dynamics. So, x 1 transpose is f x 1 p plus x 2 plus z transpose. z dot is now this entire mess, I mean, this is entire thing. So, I am going to sort of copy this guy here. So, I am going to copy and paste this guy here. So, so, that is what is z transpose z dot. And of course, I have, still have my minus 1 over gamma p tilde transpose p hat. I still have that same term here.

Now, notice again that unlike the previous scenario, p hat dot is not defined yet. So, that remains as it is. p hat dot remains as it is, it is not defined. But again, if you remember the extended matching design discussion, the good thing is p hat dot is something that we specified. So, we know what it is. So, we know what it is. So, we are not quite worried about what is p hat. I mean, whether we can implement it or not. So, we can always use the control to cancel this p hat dot if we so, desire. Excellent. Now, we are going to carefully club all the terms in the unknown. Let us first see how we go about that, let us first see how we can go about that.

So, so, what do we want to do, what we want to do is we first want to write x 2, as always, in terms of z plus x 2 desired. So, that is again equal to z minus K 1 x 1 minus f x 1 p cap. So, this is actually going to turn out to be x 1 transpose z minus K 1 norm x 1 squared plus x 1 transpose f x 1 p tilde. So, once I substitute this guy here, this is what I will get from the first term.

So, before I put in the, let us see. I am wondering if I should, yeah absolutely, why do not I substitute for my control also. So, I will choose my control as minus omega cross x 2, basically try to cancel everything I can. So, this is for this guy. Then, I will have minus del f del x 1, and here it is a K 1 I plus K 1 I just so, that the dimensions match. x 2, this is to deal with this term, minus f x 1 p hat dot, that is to deal with this term.

Then I introduce a good term which is minus K 2 z. And then finally, I want to deal with this term which should give me, which for, which I will have what del f del x 1 plus K 1 I f x 1 p cap. Because I cannot implement a p, so, I put in a p cap. So, this becomes my control, I am sorry, this is not my control.

Well, I mean, that's ok, I think I will continue to retain this as my control, no problem. So, what do I get from the second term here after substituting this guy is a lot of cancellation. So, I will have this guy go away, I will have this guy go away, I will have this guy go away and I am left with a lot of nice terms.

So, I will be, I will end up with minus K 2 norm z squared. This is coming from this guy, this multiplying this, then I will end up with one more term due to this and this together. And that term is, now plus z transpose and del f del x 1 plus k 1 I. Just this I, the purpose of this I is to just ensure that the dimensions of the del f del x 1 and the second term match, remember that, times f x 1 times a p tilde.

So, this is what you get from the second piece. And finally, I am left with minus 1 over gamma p tilde transpose p cap dot. So, this was our control. I mean, our control is this guy. Notice that I had already mentioned that this x 1 term, additional x 1 term is not required because, so, we will also not use that x 1 term here. So, do not worry about that because I, I still have this additional term left but we do not have to worry about that term at all.

So, now what do I do is I take, I have my minus K 1 not x 1 squared minus K 2 norm z squared plus x 1 transpose z, these three terms. And then I take a p tilde transpose common and I have f x 1 transpose x 1 from taking transpose here. And, in fact, I can take something common here, plus, again, f x 1 transpose from taking common transpose here, I get a del f del x 1 plus k 1 I transpose z.

So, that is what I get. Minus a one 1 gamma p cap dot, minus a 1 over gamma p cap dot. So, what do I do? I just try to drive this guy to 0 because I cannot do anything better than that. So, that is what I do. I just try to drive this term to 0. And for that I choose my p hat dot as gamma times f x 1 transpose times x 1 plus del f del x 1 plus K 1 I transpose times z.

And once I make this choice, I am left with, so, this is my update law, just this. So, once I have made this choice, my V is basically minus K 1 norm x 1 squared minus K 2 norm z squared plus x 1 transpose z. And I know that this guy is less than equal to half norm x 1 squared plus half norm z squared by standard completion of squares, using a b is less than equal to a squared plus b squared divided by 2.

So, this entire thing is less than equal to minus K 1 minus half norm x 1 squared minus K 2 minus one half norm z squared. Pretty simple, pretty straightforward, it is pretty straightforward. And I know that this is negative semi-definite if K 1 K 2 are strictly greater than half. Let me write this properly.

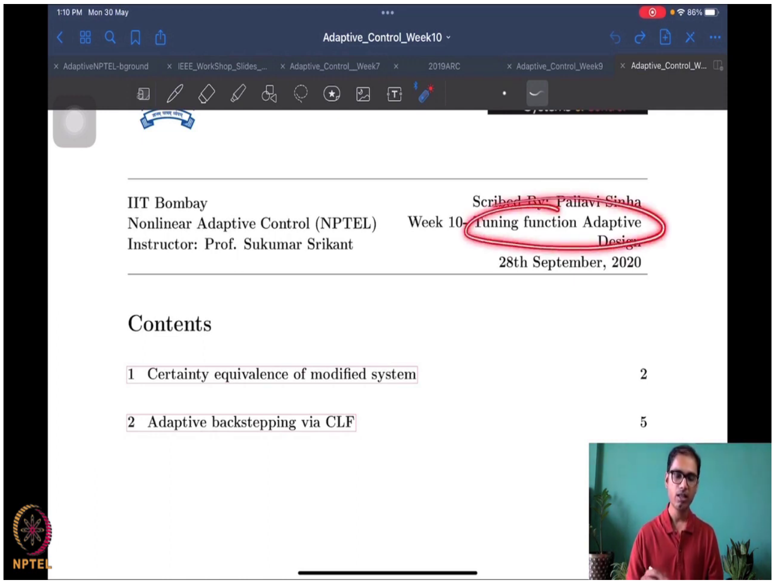
And so, I am done. As usual, I proved that x 1 goes to 0 z goes to 0 and because z goes to 0 and x 1 goes to 0, I can easily show that as x 2 also goes to 0 as t goes to infinity, just like before. So, nothing changes in those steps. Now, notice that we have only one control law, one Lyapunov function, sorry one Lyapunov function in the previous page here, and one parameter update law, which is what you would want.

Now, if you notice, so, these are the standard similarities anyway, that I always point out, even when we did the general extended matching. If you notice the p hat dot, you see two terms, gamma f transpose x 1, that is the first term and you see that is the same as gamma f transpose x 1 here for the p hat dot. And the second term is gamma f transpose del f del x 1 k 1 I transpose z gamma f transpose del f del x 1 k 1 I transpose z.

That is the second term. And that is exactly identical to this delta f transpose del f del x 1 k 1 I transpose z. So, what happened? Because we declared only one Lyapunov candidate, the p hat dot that we get is actually sum of the two derivatives, that is, it is the sum of the p hat dot and the p bar dot in the adaptive integrator backstepping case. So, that is one important thing to note.

The second important thing to note, which I, which we have already talked about is the sort of drawback of this method is that the control now contains the derivative p hat dot, control contains p hat dot. And as the distance between the control and the parameter becomes larger and larger, the derivatives also become larger and larger which can lead to implementation troubles and noise in the control implementation. So, this is essentially the idea of how to implement your standard adaptive integrator backstepping and also the extended design. So, we have seen both of them. So, I hope that all of you will be able to now actually use these design methodologies for your real applied problems.

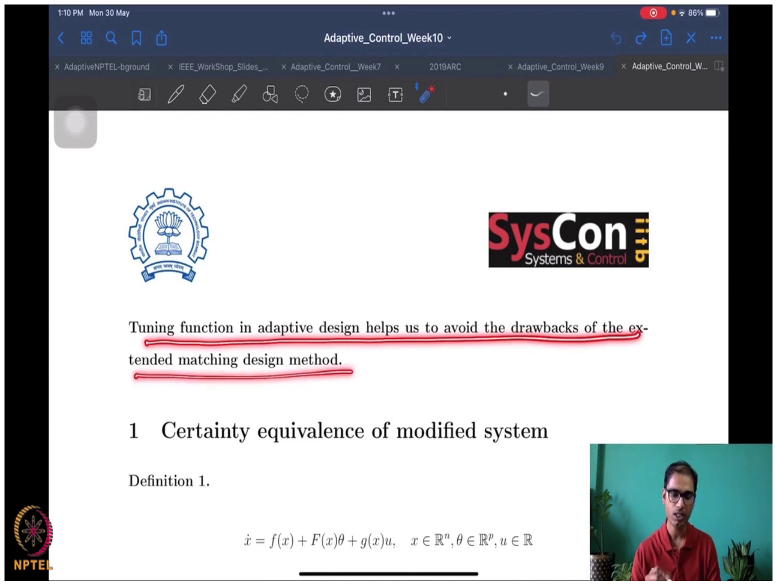
(Refer Slide Time: 20:11)

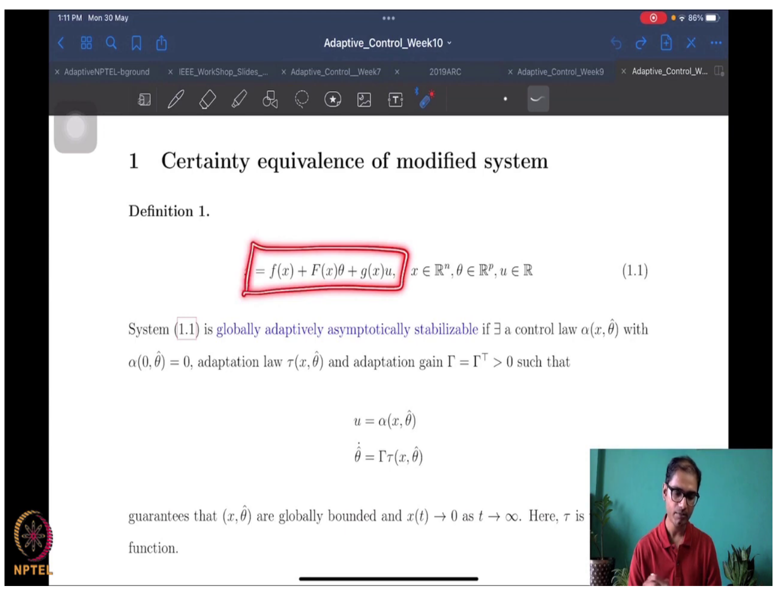


So now, we are sort of ready to move into week 10 lectures. Again, we are in week number 9 we seem to be always moving forward or we would always seem to be ahead. But we are not really ahead because we have some more additional material to cover. These weeks are more for homework reference than anything else. So, we are ready to move into week 10 lectures, and the key idea that we talk about is the tuning function adaptive design.

So, we already improved on one design flaw in the adaptive integrator backstepping which was that you have to sort of create two parameter, estimates per parameter and we were of course left with one issue in the extended matching which is that you have the derivatives of the parameter update, the parameter update law appearing in the control law. So, theta hat dot and p hat dot these things appear, which is not nice. So, we want to alleviate that issue also. And that is why the tuning function adaptive design method is a popular method or an improvement over what we have looked at until now. And so, that is what we sort of want to do.

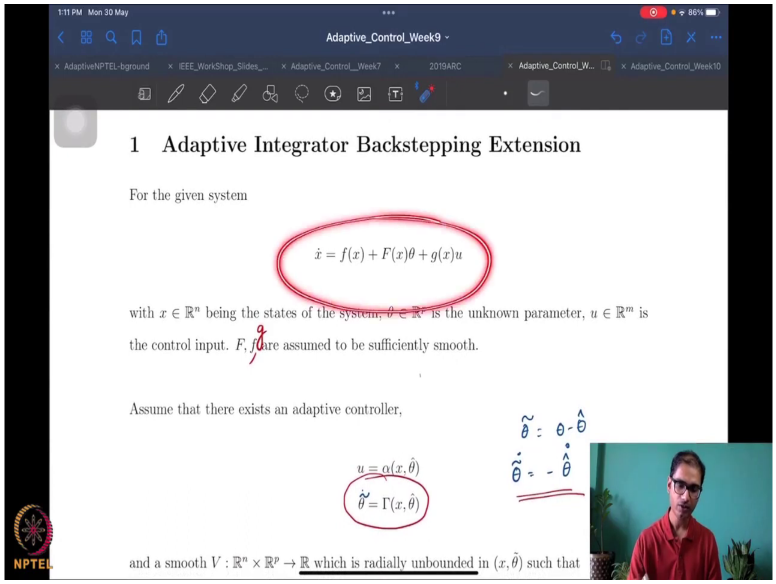
(Refer Slide Time: 21:31)





So, this is what we are saying tuning function in adaptive design helps us avoid the drawback of the extended matching design method. So, that is the idea. So, we have a few definitions first, so, we have a few definitions first. And I will go into some detail of nonlinear control in this set of lectures as for us to be able to understand these definitions better. So, for now I will sort of define one, one, couple of these definitions and then we will start with the more nonlinear control material in the subsequent session. So, we say that this system x dot is f x plus cap f x theta plus d x u.

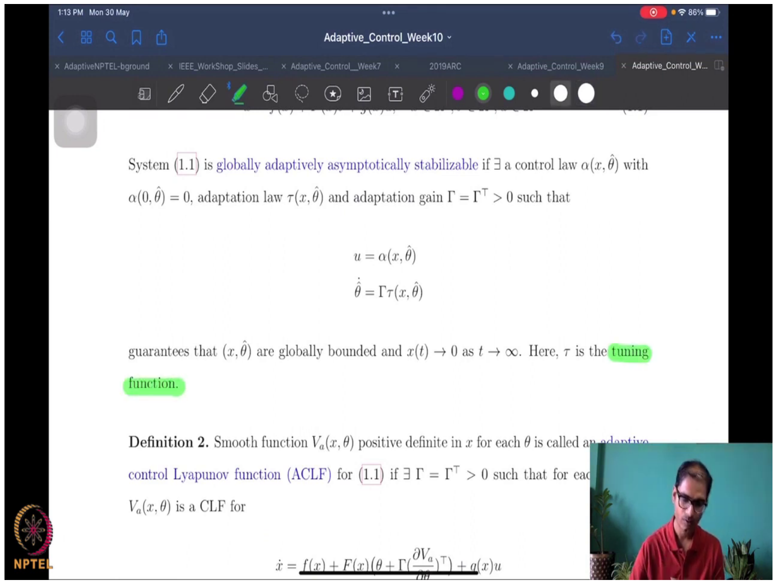
(Refer Slide Time: 22:42)



You are already used to looking at this system because we have been seeing this sort of a structure everywhere in adaptive integrator backstepping. So, this kind of structure is standard. Again, this is standard construct in the K. K. K. book because so, there is a drift term then there is a term depending on the unknown parameter then there is a control dependent. So, that is the same structure here too. The only assumption here is that u is now just a real number. It is a single input system. But again, these definitions can be generalized. But to make the treatment reasonable and easy to follow, you use the single input assumption here.

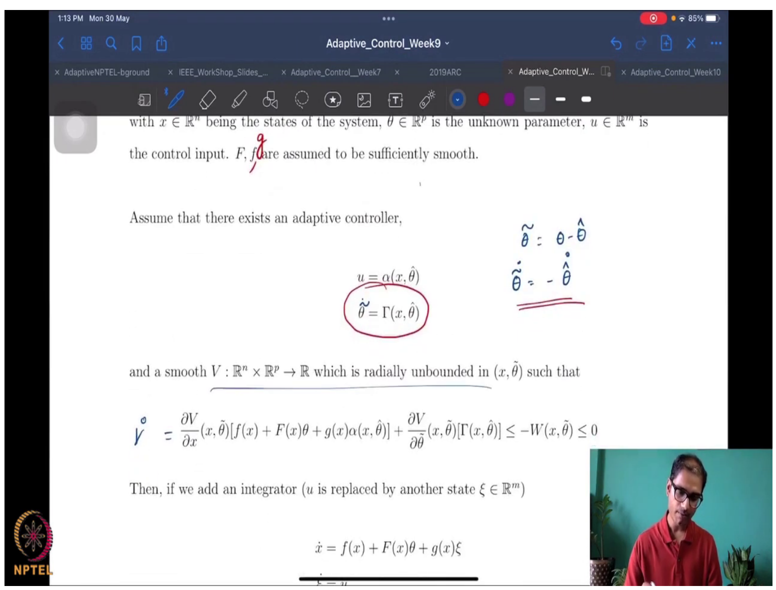
So, this system 1.1 is said to be globally adaptively asymptotically stabilizable. So, we have already seen global asymptotic stability, but here we are talking about global adaptive asymptotic stability. So, this system is called globally adaptively asymptotically stabilizable if there are two things, a control law, alpha, there is a feedback law alpha which depends on x and theta hat because theta is unknown, and a adaptation law which is a tau, which is given by tau and the gain gamma. Well that says that theta hat dot is gamma times tau, again depending on x and theta hat possibly such that the x theta hat states are globally bounded, and the x states go to 0 as t go to infinity.

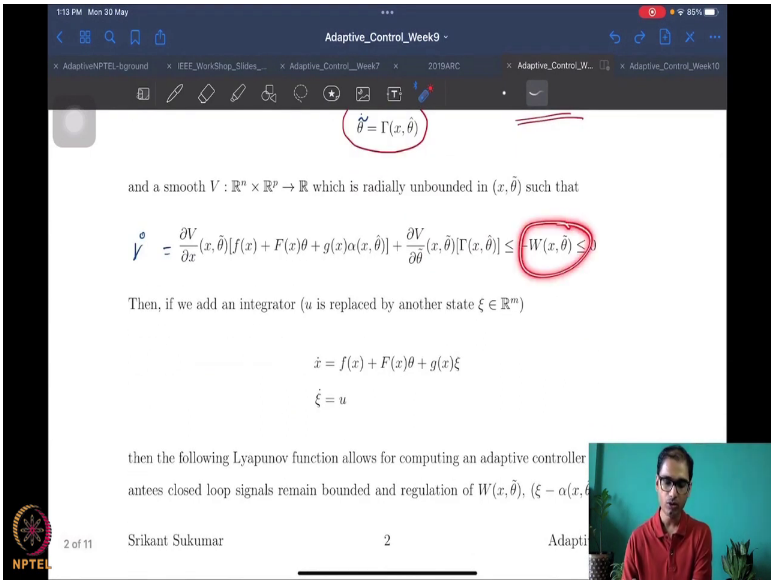
(Refer Slide Time: 24:14)



So, tau is of course called the tuning function, that is the term here. The tau is called the tuning function. And this is the important, a piece of terminology here. So, this is what you want, need for a system to be globally adaptively asymptotically stabilizing.

(Refer Slide Time: 24:37)





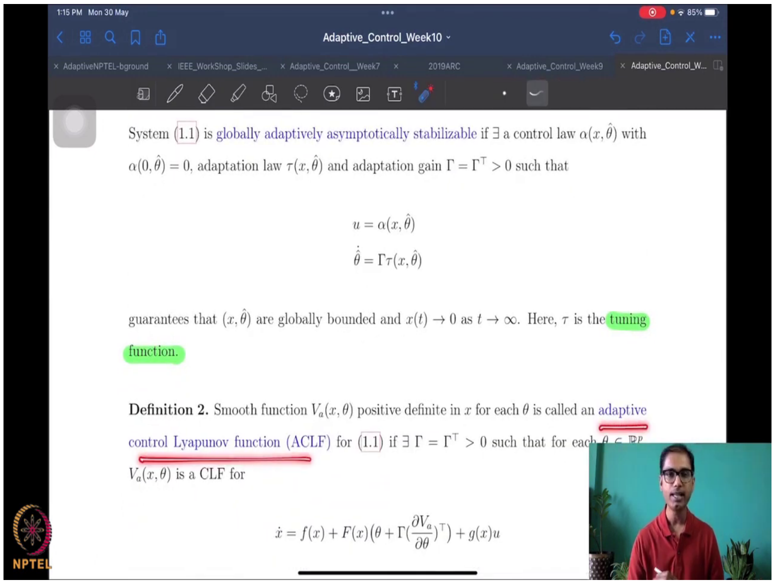
Now, remember that we did see something similar here also where we said that there exists a V readily unbounded and all that stuff such that the V dot is negative semi definite. Negative semi definite at least, and we essentially want to claim that W goes to 0 as t goes to infinity, and W goes to 0 as t goes to infinity. Here, it is a little bit more specific.

We are not just happy with W going to 0 because we do not know what kind of function of state and parameter update or parameter estimate W is. Here we are directly interested in claiming that x actually goes to 0. So, we want existence of an update law, sorry, a feedback law and a parameter update law and an adaptation gain such that the states remain bounded, x and theta hat remain bounded, and x goes to 0 as t goes to infinity.

So, this is slightly different from the previous construct. Also, this definition is not in terms of any Lyapunov candidate or anything like that. there is no V here in this definition. But of course, it should be evident to you that if such conditions hold, then there must be, there must exist such a. So, this is one of the key definitions which we will use, global adaptive asymptotic stability or globally adaptively asymptotically stabilizable.

Adaptive because theta is unknown, so, the, everything, there is a theta hat in the feedback law and the update law, and stabilizable because we want x to go to 0. We do not care about theta hat converging to theta as usual. So, this is a definition. So, remember this is a definition. We are defining this terminology. But of course, this is what we want in all our problems. We want the existence of such feedback and adaptive laws. So, this definition makes sense to us.

(Refer Slide Time: 26:43)



So, what we want to do in the subsequent session is talk about the ACLF, adaptive control Lyapunov function. But in order to do that I realize that all of you need a little bit of refresher at least on control Lyapunov functions themselves. So, that is what we will do. So, anyway, so, what did we look at in this session?

We had already done the unmatched adaptive control design via the standard adaptive integrator backstepping which leads to a two-parameter update loss or two parameter estimates per parameter. We wanted to get rid of these two parameters per parameter, two parameter estimates per parameter using the extended matching design.

And that is what we have completed today with the design and analysis of the same. We are now starting to look at the tuning function design which improves upon the drawbacks of the extended matching design also because the extended matching design brings in the derivative of the theta hats and the p hats which we do not like.

So, we want to move into this tuning function design method which of course, we started off with defining global, globally asymptotically, globally adaptively asymptotically stabilizable systems. And we want to move into defining adaptive control Lyapunov functions. And before we do that we are going to, in the subsequent session, look at a little bit of theory of what is control Lyapunov functions themselves.

So, we have already seen candidate Lyapunov functions and Lyapunov functions but we have not really talked about control Lyapunov functions, and the theory behind it. So, I am going to spend a little bit of time talking about control Lyapunov functions. So, that is it for this session. And I hope to see you folks again in the next session. Thank you.