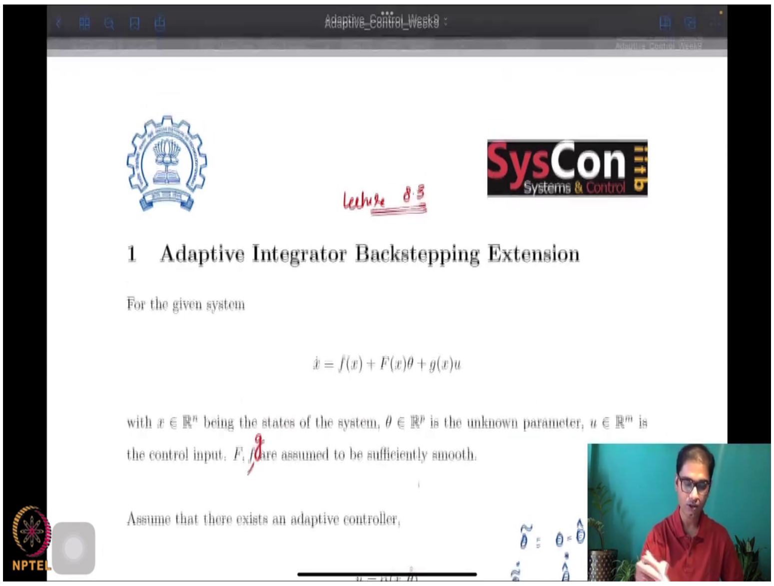
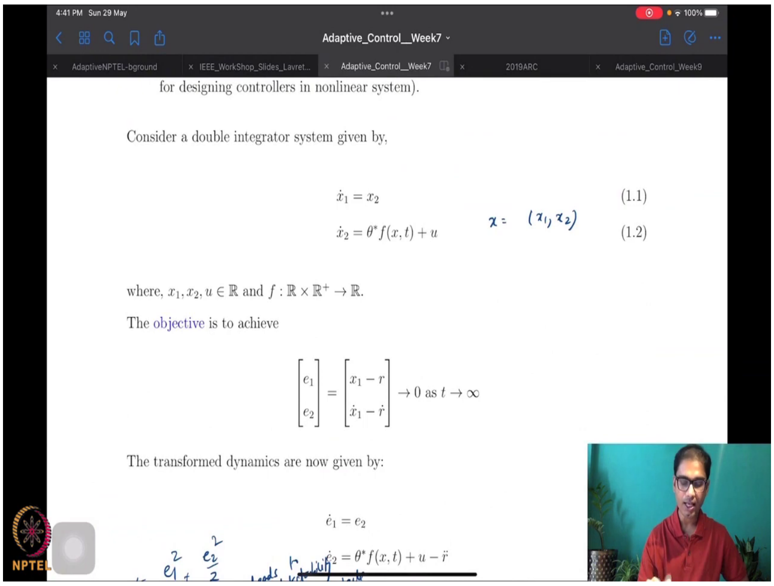
Nonlinear Adaptive Control
Professor Srikant Sukumar
Systems and Control
Indian Institute of Technology, Bombay
Week 8
Lecture No: 46
Generalization of Adaptive Integrator Backstepping Method (Part 2)

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. So, we are well to the 8-week lectures of this course, and we are already looking into several algorithm design methods through which we can actually control autonomous systems such as the Space X satellite that we see in the background hovering the earth.

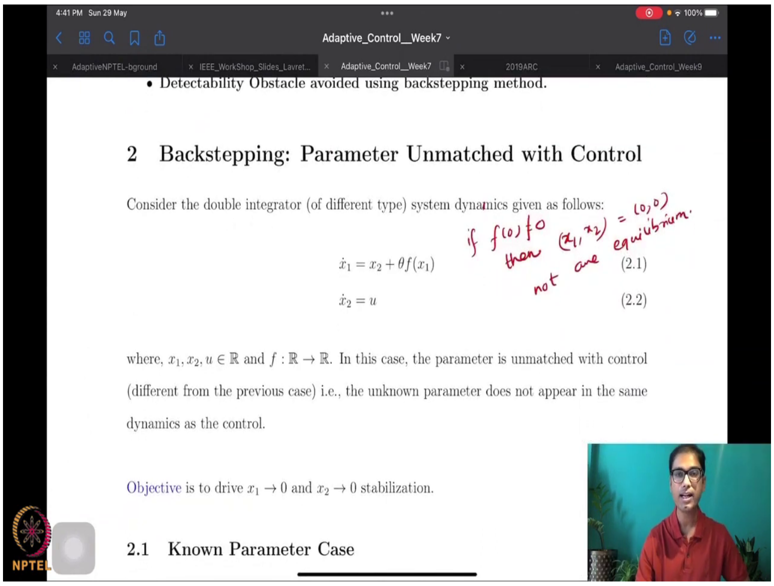
(Refer Slide Time: 00:58)





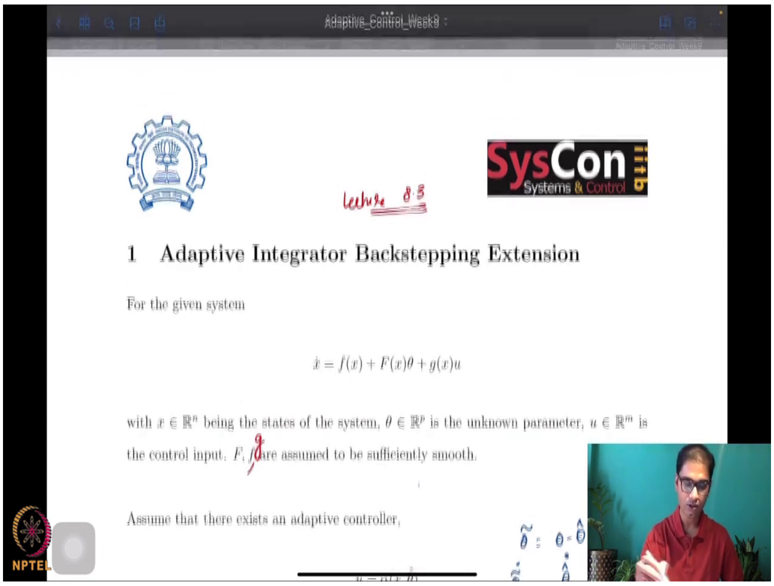
Now, what we were doing last time was essentially the beginning of the generalization of the adaptive integrator backstepping method. So, this was the method that we had, we had sort of looked at in the weak 7 lectures, if you remember. So, this was essentially the unmatched technique.

(Refer Slide Time: 01:24)

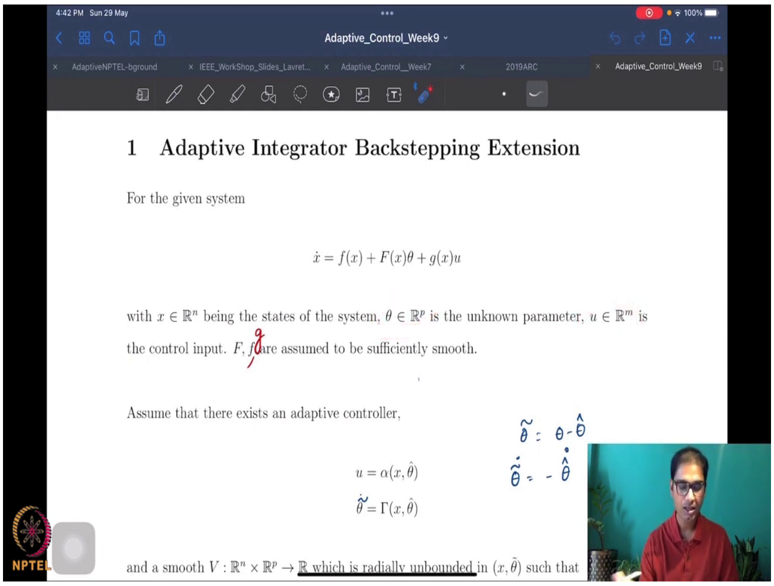


So, this was sort of what we had done in the week number 7, where we had the parameter which was actually unmatched with the control. So, and we had essentially devised how to use the backstepping method in order to design an adaptive controller for such cases.

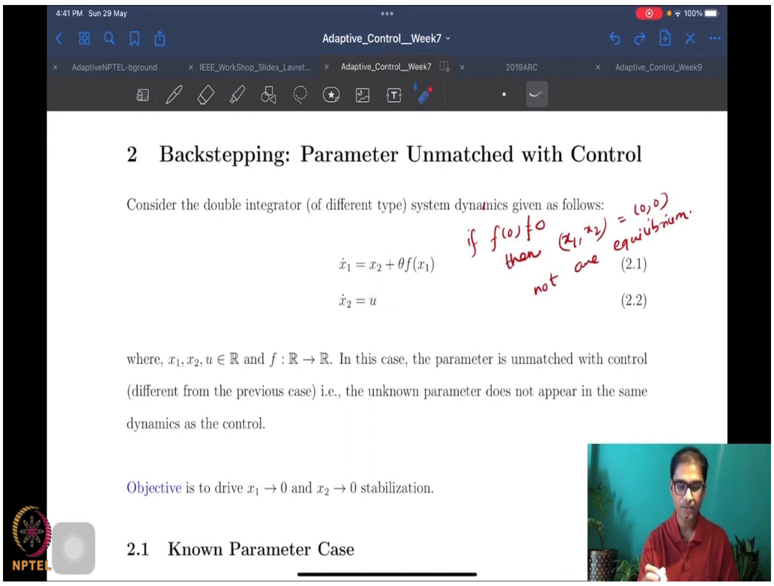
(Refer Slide Time: 01:49)



Adaptive_Control_Week9
lecture 8.3
SysCon
Systems & Control
1 Adaptive Integrator Backstepping Extension
For the given system
$\dot{x} = f(x) + F(x)\theta + g(x)u$
with $x \in \mathbb{R}^n$ being the states of the system, $\theta \in \mathbb{R}^p$ is the unknown parameter, $u \in \mathbb{R}^m$ is the control input. F, f, g are assumed to be sufficiently smooth.
Assume that there exists an adaptive controller,
$\tilde{\theta} = \theta - \hat{\theta}$
NPTEL



4:42 PM Sun 29 May
Adaptive_Control_Week9
AdaptiveNPTEL-bground
IEEE_WorkShop_Slides_Lavret...
Adaptive_Control_Week7
2019ARC
Adaptive_Control_Week9
1 Adaptive Integrator Backstepping Extension
For the given system
$\dot{x} = f(x) + F(x)\theta + g(x)u$
with $x \in \mathbb{R}^n$ being the states of the system, $\theta \in \mathbb{R}^p$ is the unknown parameter, $u \in \mathbb{R}^m$ is the control input. F, f, g are assumed to be sufficiently smooth.
Assume that there exists an adaptive controller,
$\tilde{\theta} = \theta - \hat{\theta}$
$\dot{\tilde{\theta}} = -\dot{\hat{\theta}}$
$u = \alpha(x, \hat{\theta})$
$\dot{\hat{\theta}} = \Gamma(x, \hat{\theta})$
and a smooth $V : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ which is radially unbounded in $(x, \tilde{\theta})$ such that
NPTEL



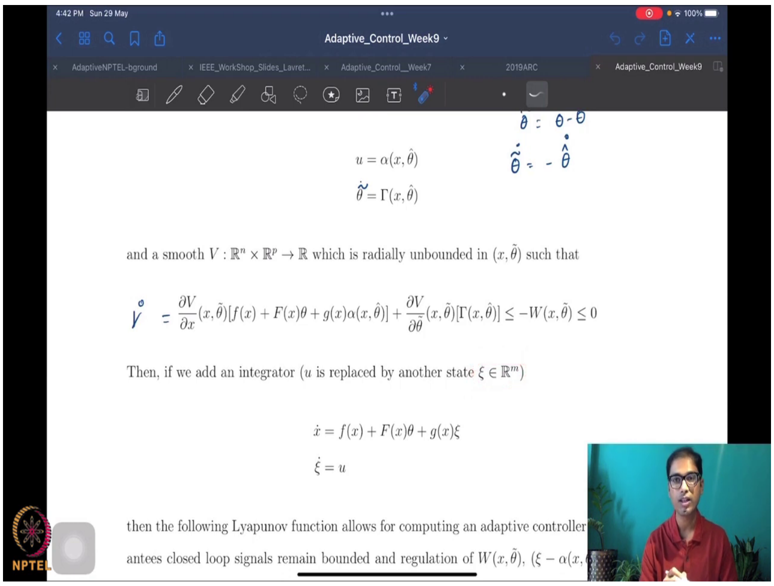
4:41 PM Sun 29 May
Adaptive_Control_Week7
AdaptiveNPTEL-bground
IEEE_WorkShop_Slides_Lavret...
Adaptive_Control_Week7
2019ARC
Adaptive_Control_Week9
2 Backstepping: Parameter Unmatched with Control
Consider the double integrator (of different type) system dynamics given as follows:
if $f(0) \neq 0$ then $(x_1, x_2) = (0,0)$ are not equilibrium.
$\dot{x}_1 = x_2 + \theta f(x_1)$ (2.1)
$\dot{x}_2 = u$ (2.2)
where, $x_1, x_2, u \in \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$. In this case, the parameter is unmatched with control (different from the previous case) i.e., the unknown parameter does not appear in the same dynamics as the control.
Objective is to drive $x_1 \to 0$ and $x_2 \to 0$ stabilization.
2.1 Known Parameter Case
NPTEL

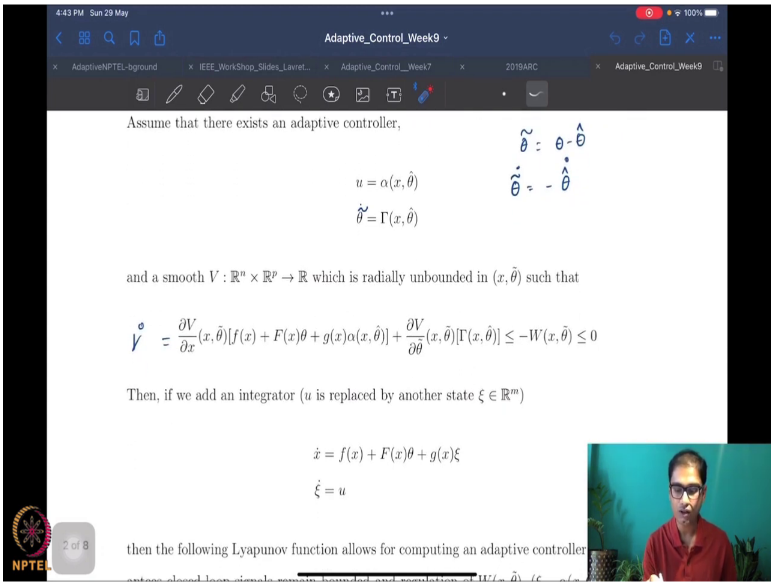
Now, what we are, we have started already to look at in lecture 8.3 is how to extend it to the general case because here everything was a scalar I mean, you had a scalar x1 and x2 and u and even the parameter was a scalar. So, now, we are looking at the general case of a system where you have an x which is in Rn, you have an unknown parameter, which is also in Rp control, which is in Rm.

And so, and then, of course, there is an extended state psi, which is also the same dimension as the control and therefore is in Rm. So, how did we start the setup of this system was that we said that, we have this form for the system and we are now used to this form for the adaptive integrator backstepping for the general case, I hope all of you remember that, here there is one term which is corresponding to the drift a second term which contains the unknown parameter which appears linearly and the third which contains the control which also appears linearly.

So, what we assume is that for such a system, where of course, the dimensions are stated and functions are sufficiently smooth, there exists an adaptive controller, which means that there exists a u and there exists a theta tilde dot such that the system has bounded states and some nice thing happens to the theta typically to the error states or say x states.

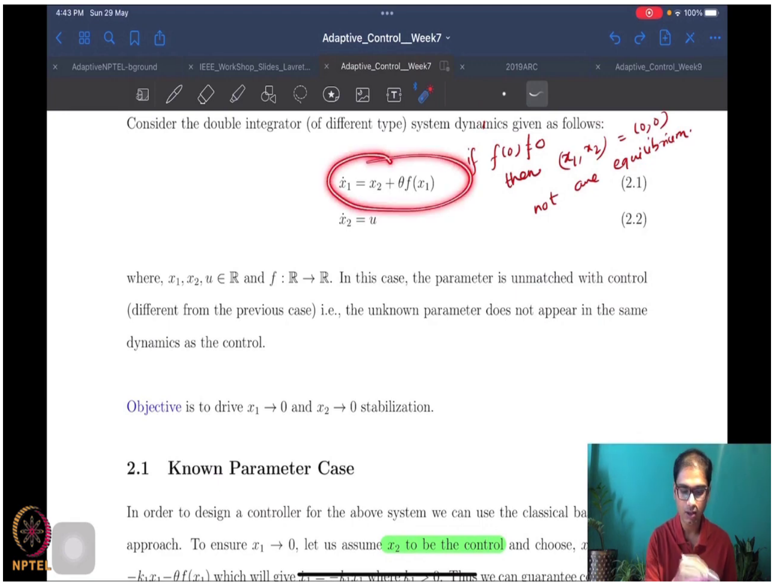
(Refer Slide Time: 03:31)

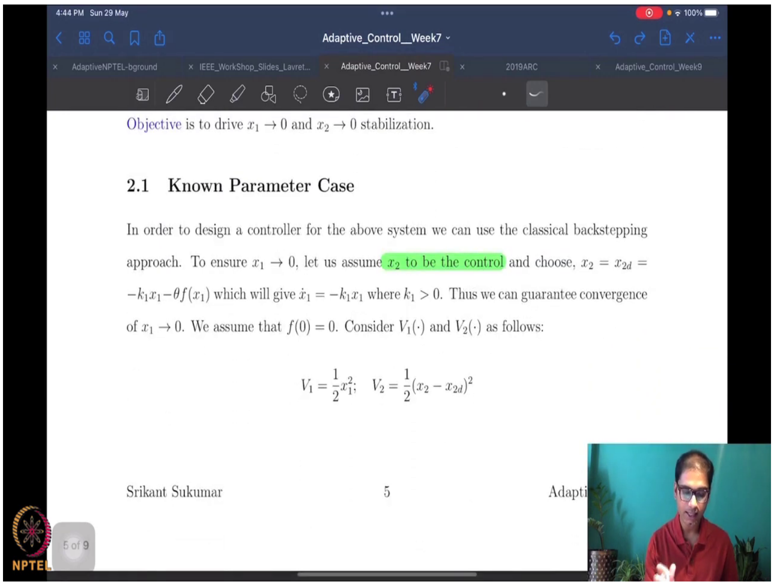


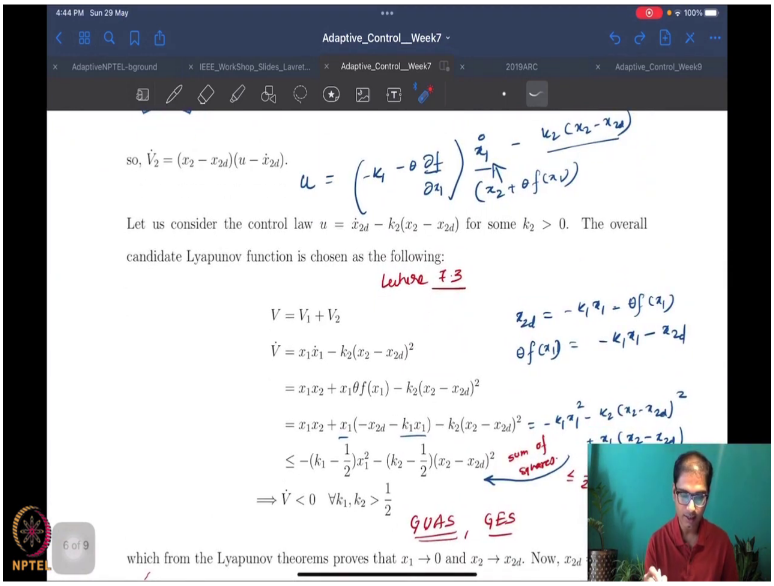


So, this is codified in terms of the existence of a smooth function V, in this case of x and theta tilde, such that its derivative is negative semi definite at least and from this negative semi definiteness, we can actually conclude more often than not, this function W x theta tilde, which is what you have on the right-hand side goes to 0 as t goes to infinity. This is exactly what we had here.

(Refer Slide Time: 04:03)

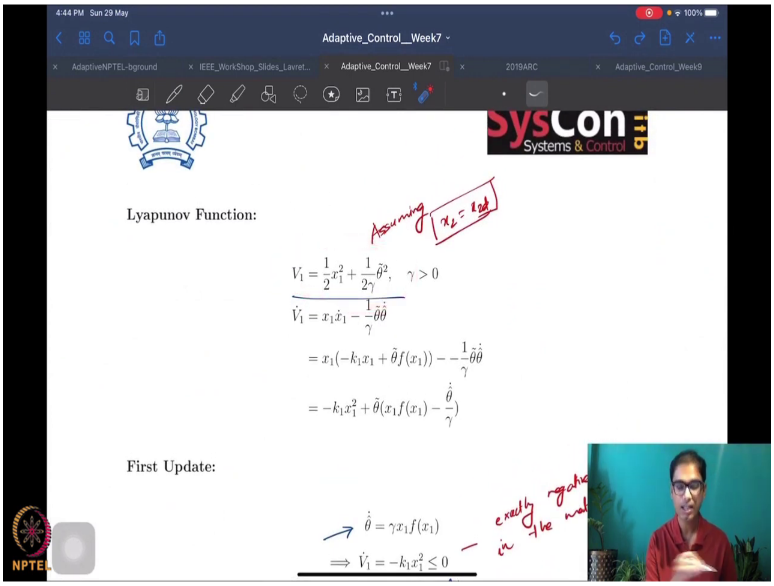


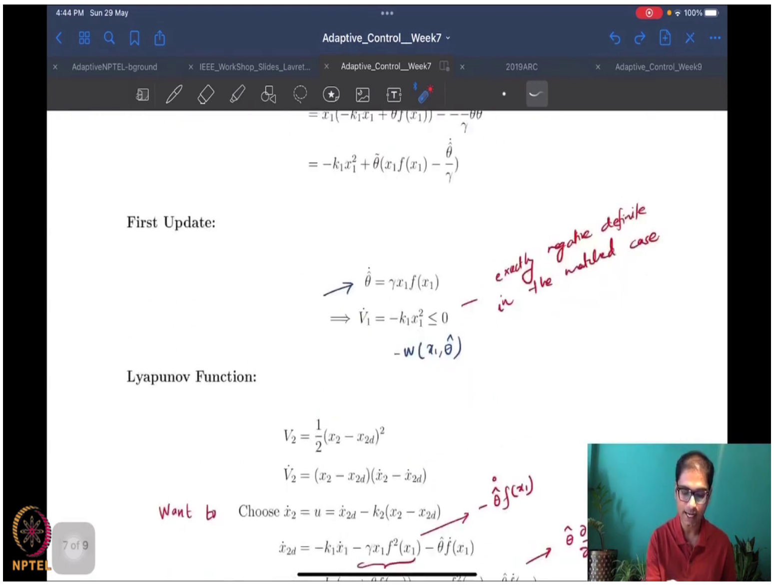




You have you had this sort of a system, and the assumption was that you had a function V in this case.

(Refer Slide Time: 04:17)

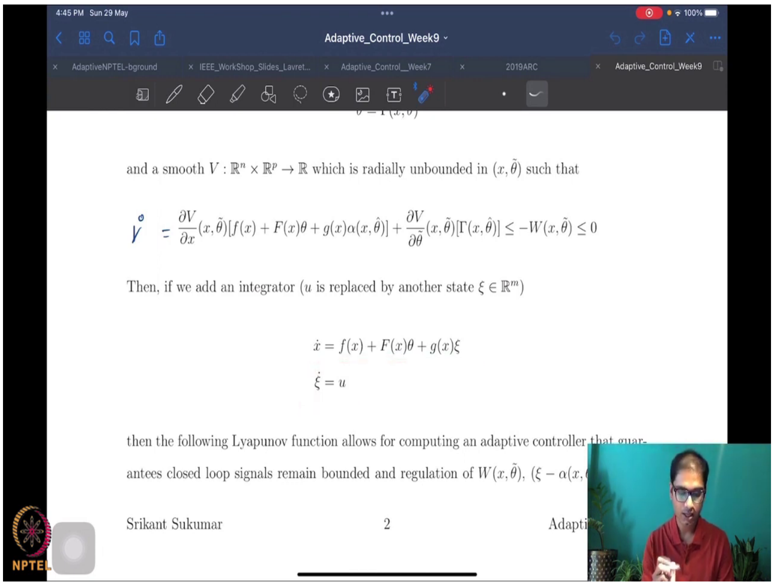




Let us, in this case, the function V is something like this, this V1 function, and the derivative of V1 after taking some theta hat dot, which is essentially the same as having a theta tilde dot came out to be this quantity, which is what is the minus W quantity, and we could prove that this goes to 0. And because this goes to 0, of course, we could prove that, you know, something nice happens to the x. I mean, I am sorry, there's is no x2 state yet.

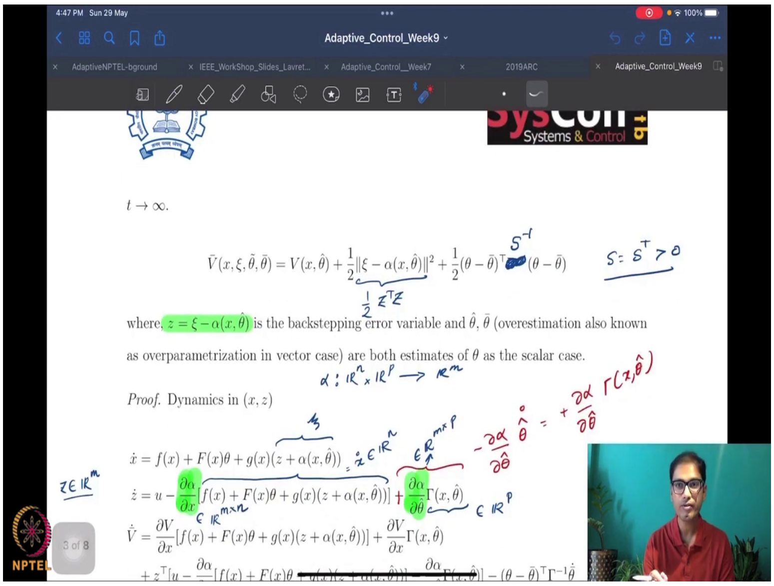
So, of course, we can prove this goes to 0, which means that x1 goes to 0. So, that is exactly what we also claimed here.

(Refer Slide Time: 04:50)



Remember again that we are looking at the tracking progress sorry the stabilization problem, but the tracking problem would have been exactly identical. Not there would not be significant difference in the tracking problem either. So, now if we had an integrator state, that is when the u gets replaced by the state psi, which is Rm, you, of course, had this psi dot equal to u sort of thing.

(Refer Slide Time: 05:32)



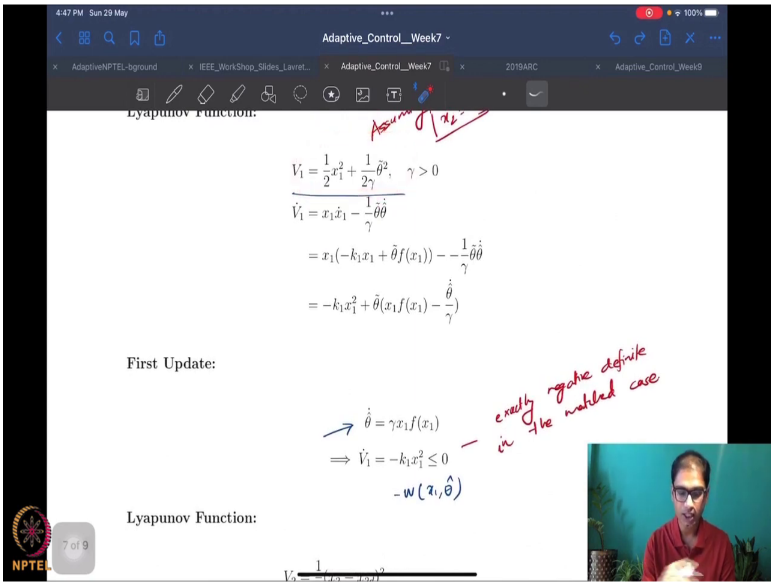
Then, what we claim is that this sort of Lyapunov function V bar now, which is essentially the previous V with an added backstepping error term, which is norms z squared and a new parameter error term, because, if you remember, there was a over parameterization. And because of the unmatched states, that is the new parameter estimate is called theta bar, the old parameter estimate was called theta hat.

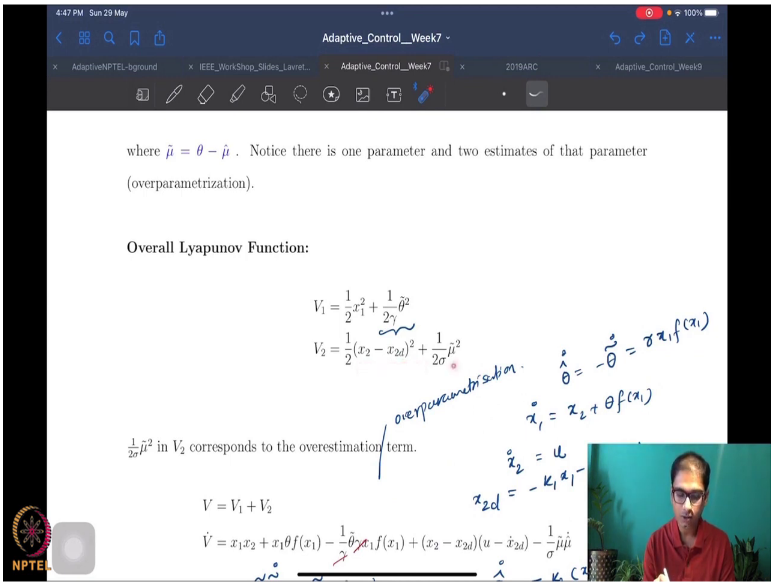
And so, with the quadratic term and the new parameter error, you can actually show negative semi definite V bar for the new system. That is the system with an integrator norm. Now, as you would imagine, because there was a nice adaptive law for the original system, we define them backstepping error, state z which is the psi minus the alpha, because, although psi is not the actual control of the system, we want it to follow the alpha because we know that is the good control. So, we try to sort of push this towards 0.

So, that is what we do. Now, since everything is vector now, instead of scalars, that is z squared, and theta squared and theta tilde squared, we have norms, that is the only difference, so now I have z transpose z, which is norm z squared, which is essentially this quantity. And then you have sort of a norm squared here, but with the scaling with this, this scaling is simply the adaptation gain.

That we already know about and the scalar case, also, we had an adaptation gain. Yeah, even in the matrix case. And now also in the vector case. So, this is the modified V bar, which will work.

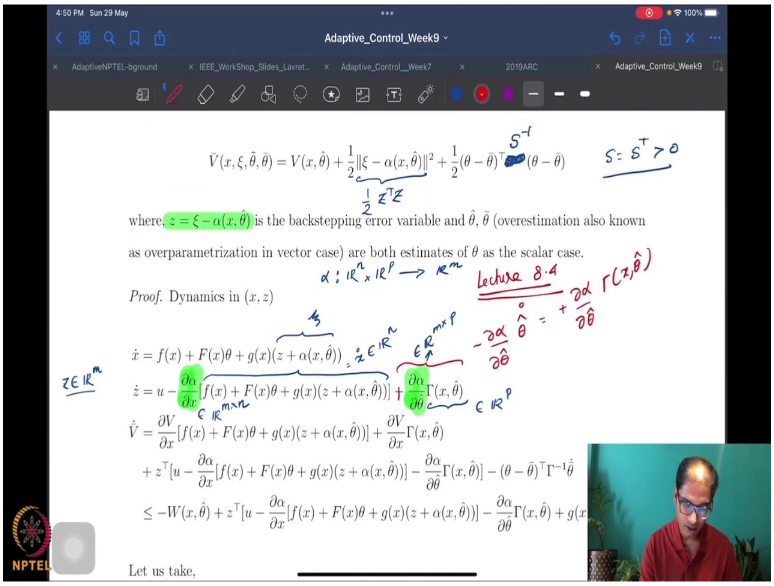
(Refer Slide Time: 07:28)

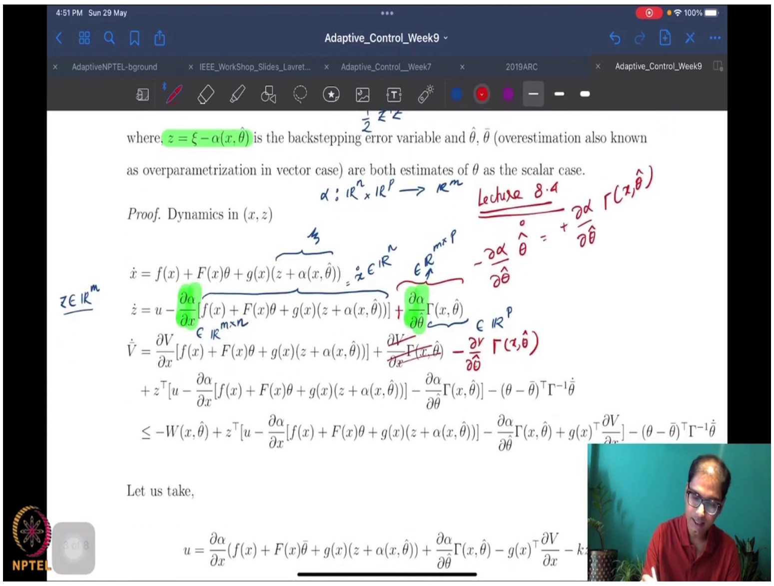


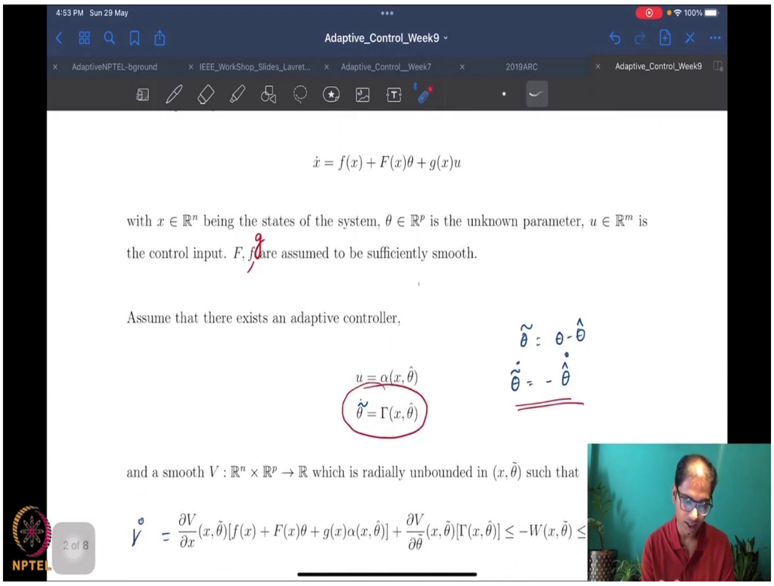


And this is, again, not very different. And because V1 if you remember, was the original function here. And so, the take our overall Lyapunov function, as just this guy added with this, which is exactly what we are talking about now, because this is the backstepping error term squared. And then there is a squared and the new parameter error.

(Refer Slide Time: 07:50)







So, squared in the new parameter error and the backstepping error squared. All right, that is it. So, it is essentially very similar to the scalar case, just taking into account the fact that we are now dealing with vectors. So, a lot of norms appear in place of simple squares. Great, so now, this is this is it. So, we essentially use this. And now, we simply started taking derivatives. That is what we had. That is where we had stopped.

So, I am going to mark my lecture, right here. So, we start our lecture 8.4. Right here. Also remember, again, that we are looking at the week 9 notes, we are not going to worry about it. The purpose of these markings is simply to sort of help align the homework’s, so let us not worry too much about this.

So, let us take the derivatives for this V bar in order to prove that this in fact, helps us claim that you know, our W goes to 0 and you know, our psi goes to alpha and all that nice stuff. So, this is the dynamic. So, what we do is we now start to write a dynamic in terms of the new variable psi. Sorry, in terms of the new variable z, which is the backstepping error.

So, x dot was the fx plus capital Fx theta plus gx psi. And psi is simply z plus alpha. That is what we do. And now, we also write the dynamics of the second state, which is psi minus alpha. So, psi dot is just a control and the derivative of alpha has two pieces, because it is depending on two different quantities and two different quantities.

So, this is u minus del alpha del x, u and this del alpha del x times x dot, which is essentially, this guy plugged in, and del alpha del theta cap theta cap dot. And so that is this with the plus sign. So that is what you will have here. That is what we verified here, because theta cap dot is minus theta tilde dot, and that is, gives me the plus sign. And that is what we have.

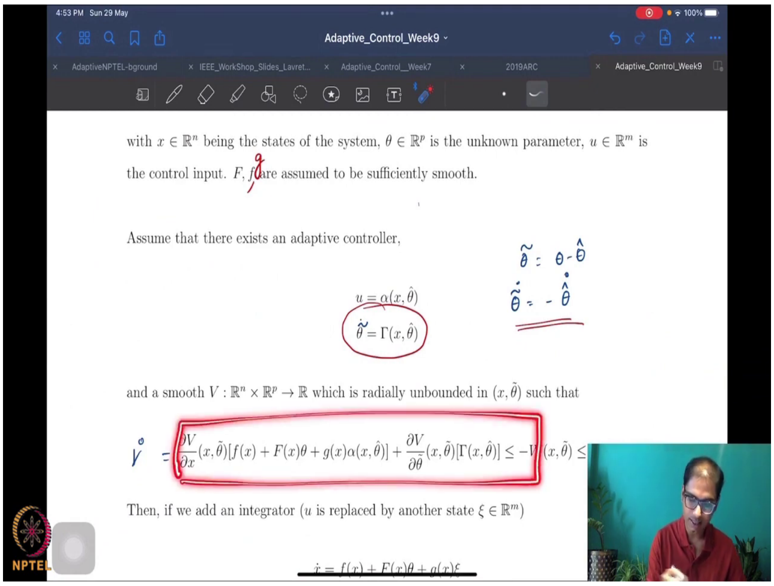
So now, when I write V bar dot, I get, again, two pieces from this partial, which is del V del x times this, plus del V del x times a gamma. Let us see, it is not del V del x this is. So, I need to check the signs here. So, this is actually equal to. So, V is a function V was a function of x and theta cap. So, this term is not this, but it has to be del V del theta cap, theta cap dot, so this is minus gamma x theta cap. Yeah, so that is what you have.

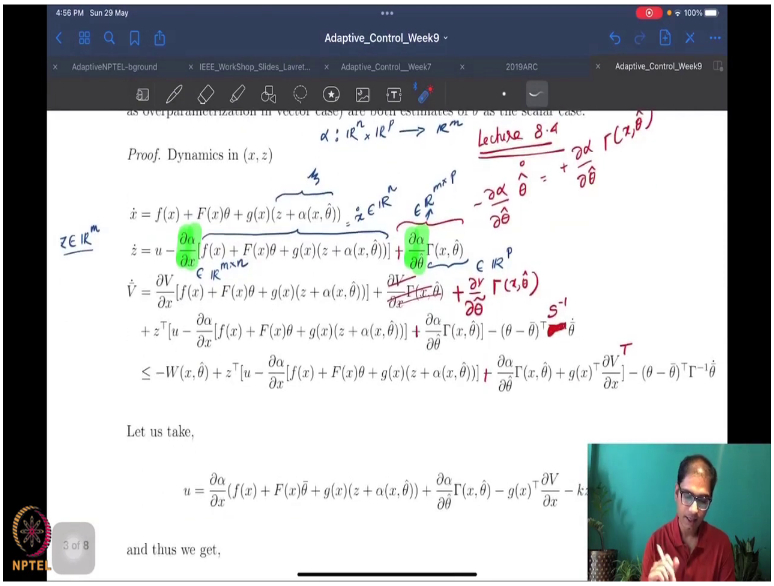
The second term will be minus del V del theta cap gamma x theta cap I am simply taking the derivative of V, which is a function of both x and theta cap. And remember that theta tilde dot was declared as gamma, and theta cap dot and theta tilde dot are related by just the negative sign. Let us just remember that much. And that is all we are doing.

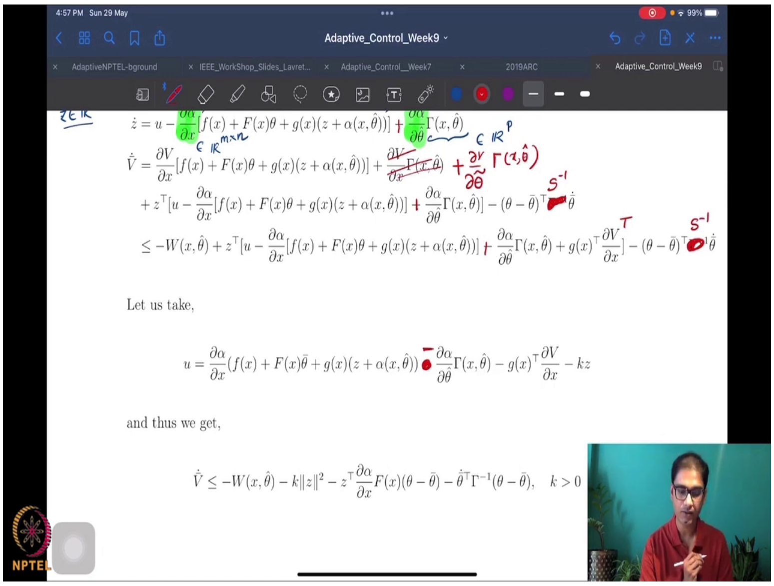
The second term is not del V del x again it is del V del theta cap times negative of gamma x theta. And then the second term is the derivative of the second term is just you know, this guy z transpose z dot and we already have written z dot here. So, z transpose z dot is this entire thing, this guy getting plugged in here, and the final term is this. So, I have theta minus theta bar transpose. So, I changed the S, gamma to S, because we already have used gamma.

So, you have theta minus theta bar transpose S inverse theta bar dot. So, I get the negative sign because theta bar has a negative sign here. So, therefore, I get a negative sign here all right. So that so it looks messy, but honestly, we are just doing careful book keeping, just careful. Now, look at this expression for the V carefully right here, if you look at this carefully.

(Refer Slide Time: 13:01)







I already know a few things I know that V dot from the previous case when there was no integrator has del V del x fx plus cap Fx theta plus gx alpha and del V del theta tilde I see. So, here it is written as a function of theta tilde I am sorry, this is this has gotten messy, this is actually V was a function of x and theta tilde. So, this was a mistake here.

I apologize. This was a mistake here. So, this is fine. So, V was a function of theta tilde in fact, so that is okay. All right, that was okay. This was so then the second partial is in fact with respect to theta tilde, and then this is theta tilde dot, which is just gamma. So, if you look at this expression, this guy is actually available here.

If you look at this, there is del V del x fx plus cap Fx theta plus gx alpha plus del V del theta tilde gamma. So, this entire thing is available, and so this entire thing can be written as minus W. And that is what we do. And then we pull this piece out. This piece is the only thing that is remained. So, if I take this out, I have a z transpose. And that shows up here. So, this piece is the only thing that is remaining is this piece and that becomes this guy.

All right, this is z transpose g transpose and del V del x transpose but del V del x being as, del V del x, will be. Since V is symmetric, so there is no need to transpose it. But if you want, you can just use the transpose here, no problem. Okay, that is not a problem let us just use the transpose here. So, let us look at this.

So, you have the rest of the terms as is, u minus del alpha del x times x dot minus del alpha, del theta cap, gamma, sorry, this signs had to be corrected here. This is a plus sign again. And this is a plus sign again. Correct? And so that is correct. And then that is it, this additional term comes in from here…. So, that is it, we have this additional term.

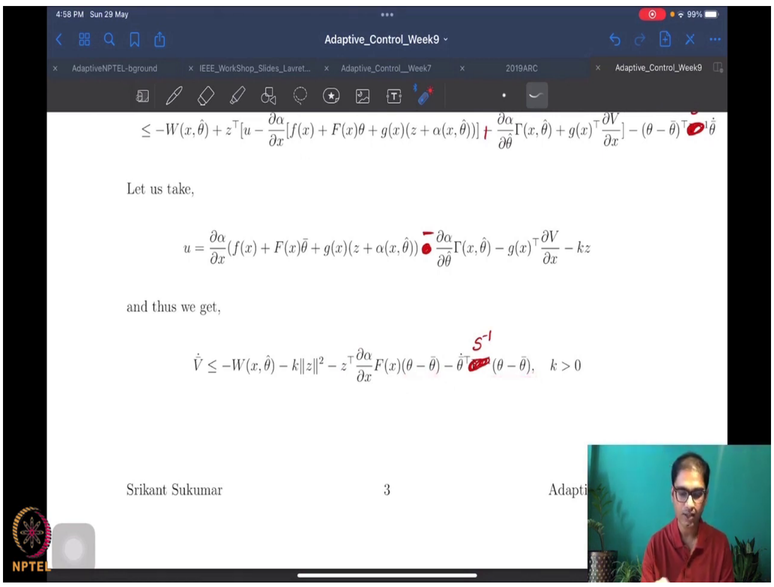
And everything else is accounted for. And then this entire term goes out. I mean, it is just reproduced here, as it is. Now, you start to see what is the advantage of you, you will start to see very soon, what is the advantage of this sort of a term? Yeah, because, because if you see Fx theta, which is the unknown, theta is the unknown has appeared once again, in the along with the u.

And if we do not have another handle here, I would be in trouble. I would be in some trouble. So, what do we do? We as usual, cancel as much as we, we can get rid of this, all of this great, this guy also. And this guy also. And then we introduce a good term, which is the minus kz. But then for this term, I have to introduce another cancelling term, which is not again, which is again, not the true value of the parameters, since it is not known.

But it is an estimate and a new one. That is what we do. And once we do that, what will we be left with all? Everything else cancelled out. Except for this guy. Yeah, corresponding to which I will have a theta minus theta bar vector. And that is what is this guy? Yeah. So, again, I need to be careful about the sign here.

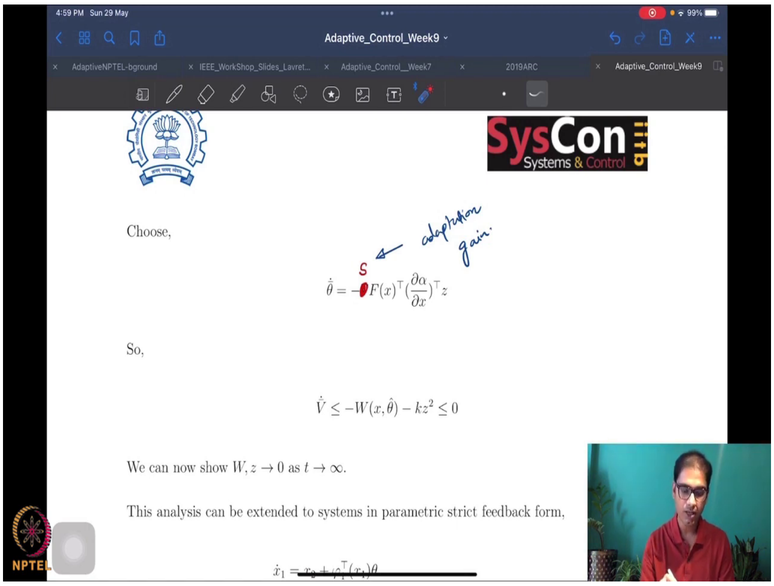
Because this was a plus, this has to be a minus sign. And so, this term yields this gain. And this is, as it is, of course, with the S inverse instead of the gamma inverse.

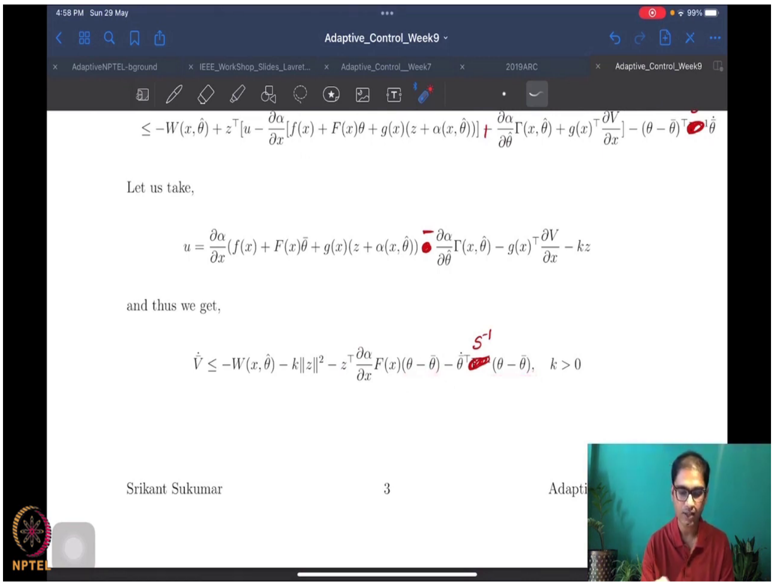
(Refer Slide Time: 17:58)

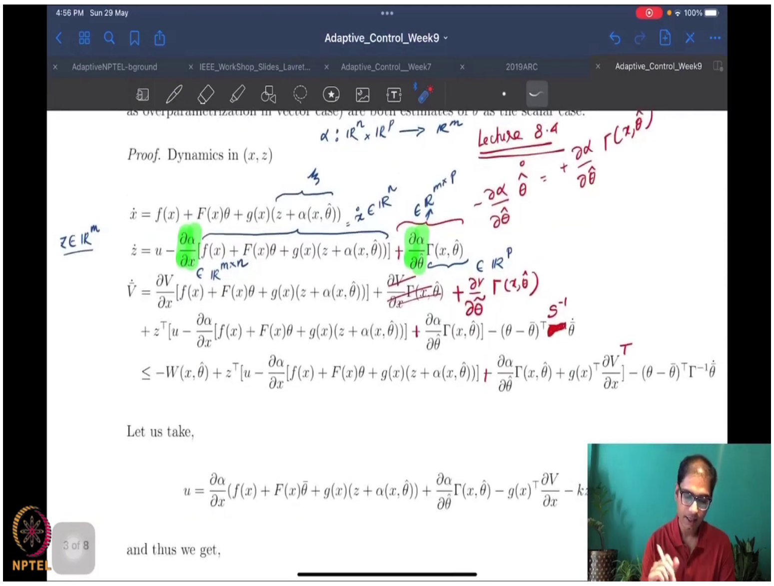


And similarly, the S inverse instead of the gamma inverse, so good things happen, as we expect, the you have a nice negative term in the z we have a nice negative term W, which we know has to be a nice negative term. And then we are left with these guys, which we know how to cancel.

(Refer Slide Time: 18:22)



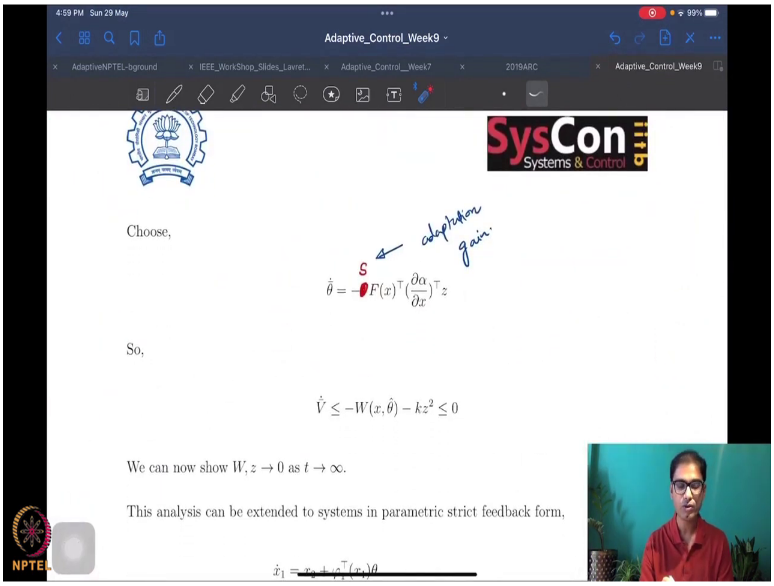


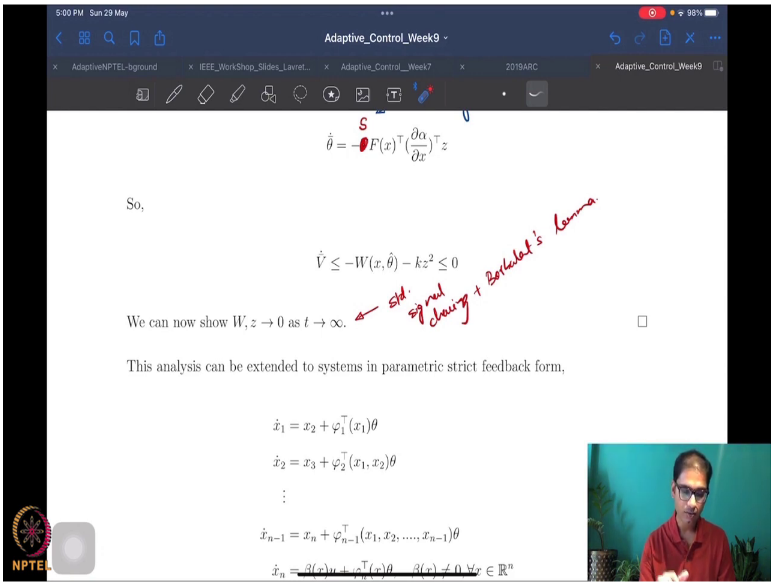


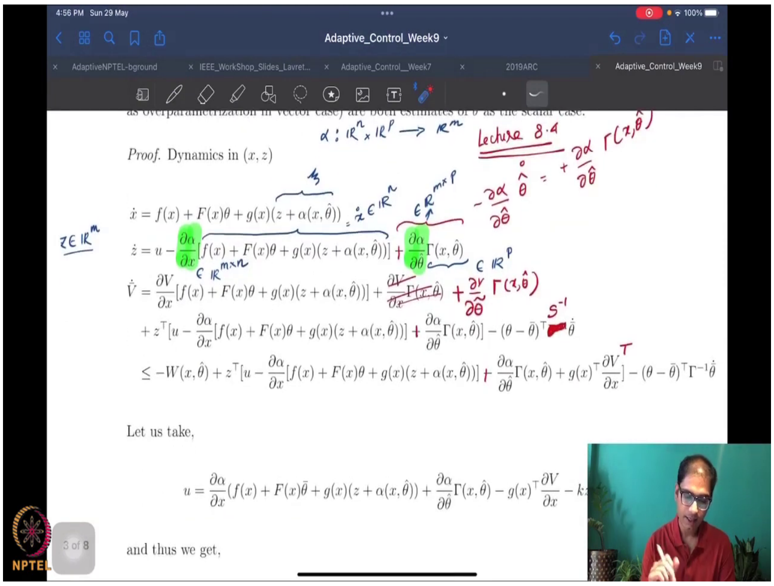
And how do we do that? We simply use this quantity to cancel this and how do we do that? So, we take theta minus theta bar common and then therefore, we have an expression here. And in this case, it will be S times F transpose del alpha del trans… del x transpose times z. Basically, it will be the transpose of this whole guy. Pre-multiplied by S. So, that is what it is. You have the transpose of that whole quantity, pre-multiplied by S, which is the adaptation gain. So, this is the adaptation gain.

All right. So, I hope that is, these steps are clear. See, all we did was very careful book keeping, and carefully clubbing terms and cancelling terms. Yeah, it is not very different from the scalar case at all. So, the purpose was to show that this method, in fact, has a very nice vector extension also, and also to show that dealing with vector states is not significantly different from the scalar counterparts.

(Refer Slide Time: 19:53)



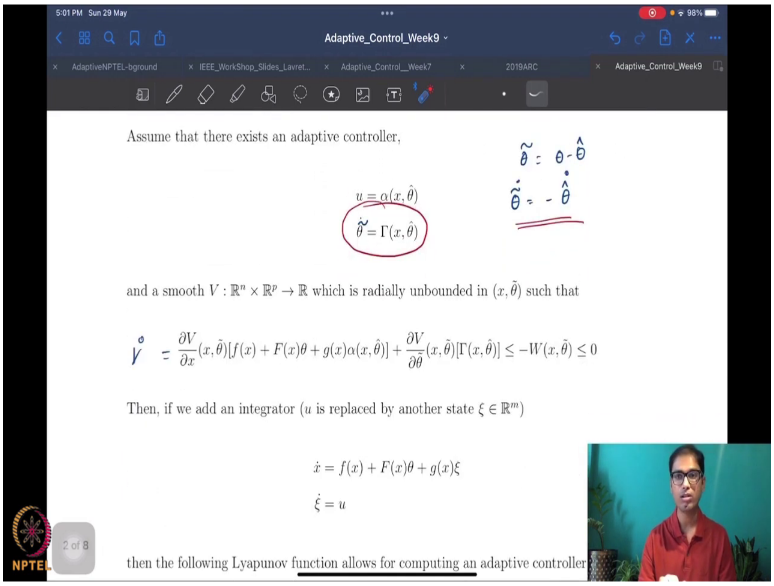


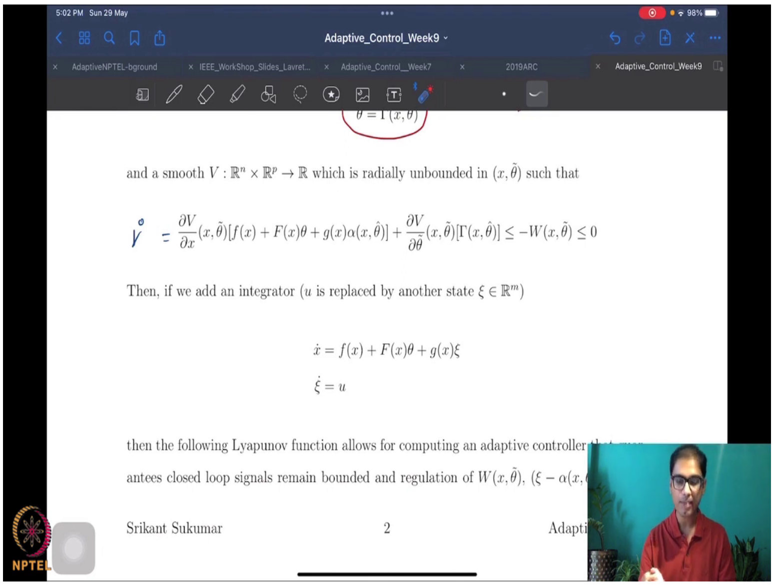


So, we have this nice expression norm. Now, after we have used theta bar dot in order to cancel this guy up not to cancel all the unwanted parameter terms that appear in the V bar dot V to get this nice V bar dot expression. And therefore, we can show very easily that W and z do go to 0 as t goes to infinity, and this is, of course, standard signal chasing plus Barbalat's lemma.

Notice that for a long time now, we have in fact stopped showing these arguments, because now we have assumed pretty much that all of you are experts in this since we did it quite a few times. And the steps were very, very standard. And therefore, we find no further need to keep repeating these steps. So, what so, what sort of is from a little bit of a philosophical not really philosophical, but actually an implementation concern is the fact that we have two different estimates. Which is theta bar and also theta hat.

(Refer Slide Time: 21:20)

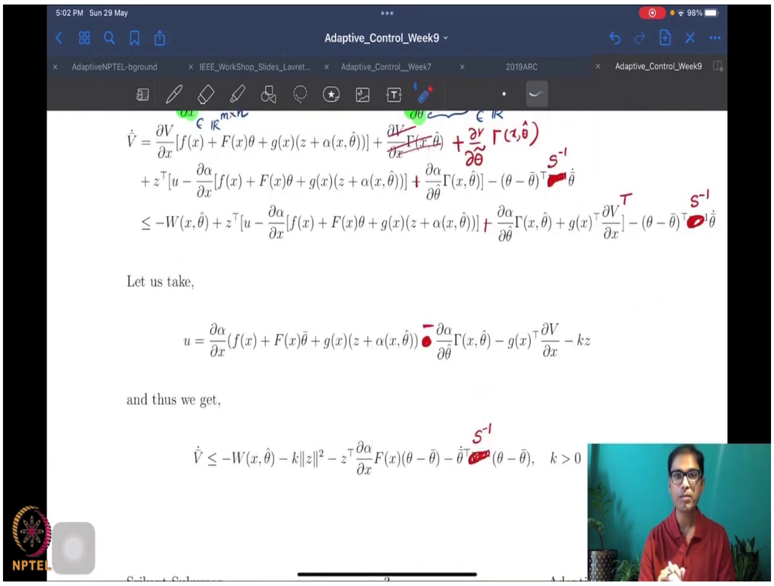


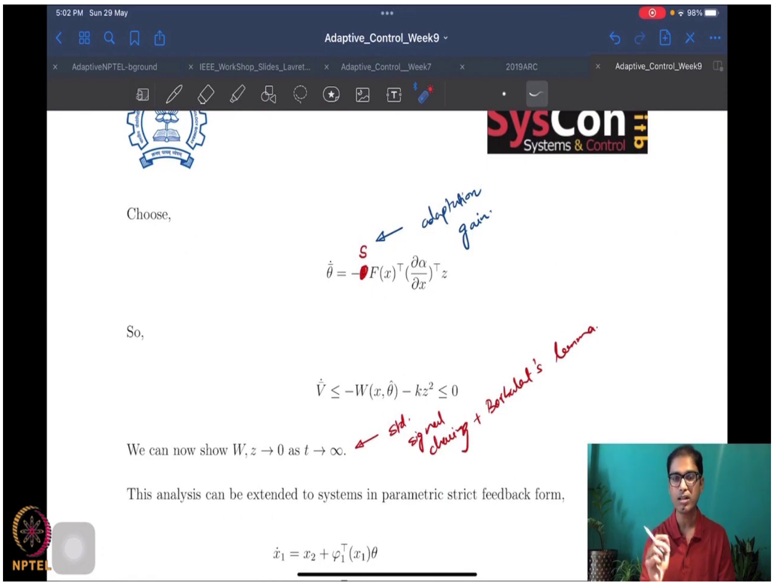


Also, the theta hat here, yeah, in order to you know, estimate the same parameter theta, yeah. And as the number of unknowns keep increasing, and just you know, Rp, in this case, you will of course, see that the number of excess unknown parameterizations or unknown parameter estimates also keeps increasing. And this is, of course, not a convenient situation to be in. Yeah, I mean, eventually, these are all going to be implemented in a microcontroller.

And you definitely do not want to load your real time control system by adding more and more states. Alright, so that is really one of the critical concerns here and we seek to address it of course, but anyway, the important thing is even for the unmatched case, we are able to design very nice adaptive control loss.

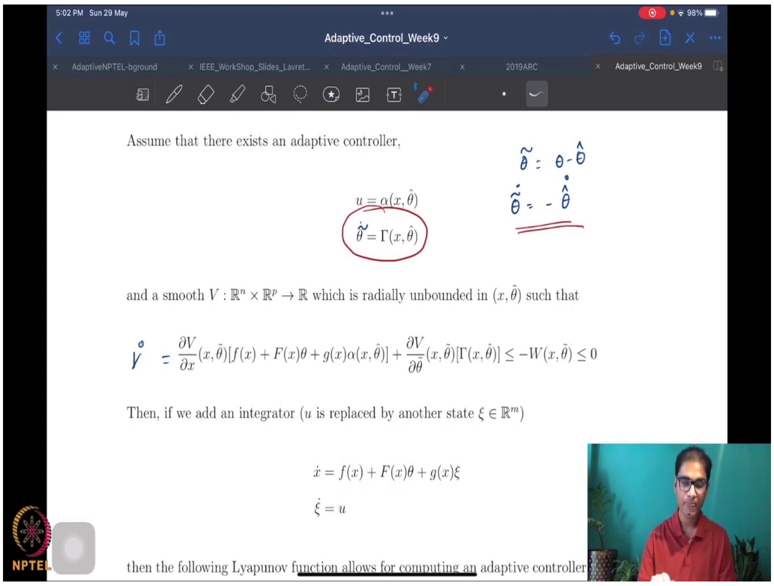
(Refer Slide Time: 22:21)

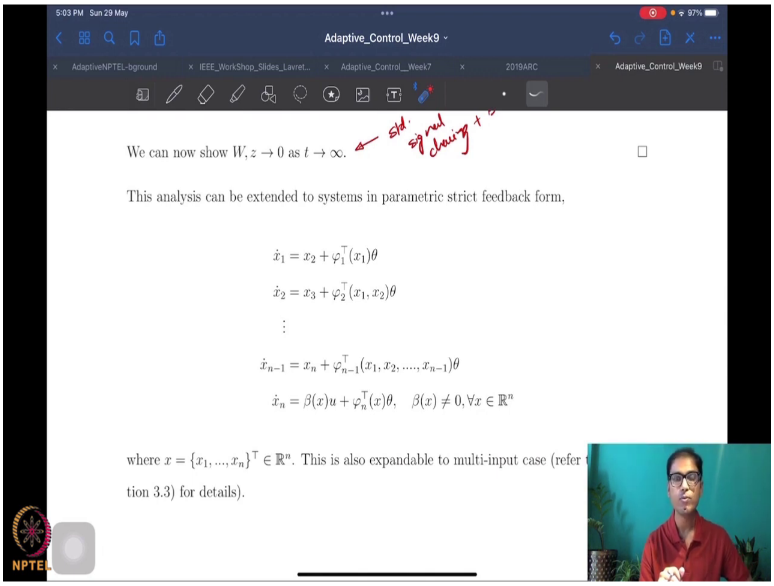




And which actually help us to deal with unknowns that are not matched with the control. Remember, this is the way it is set up right now, it currently works only when there is the unknown parameter one step above the control.

(Refer Slide Time: 22:37)



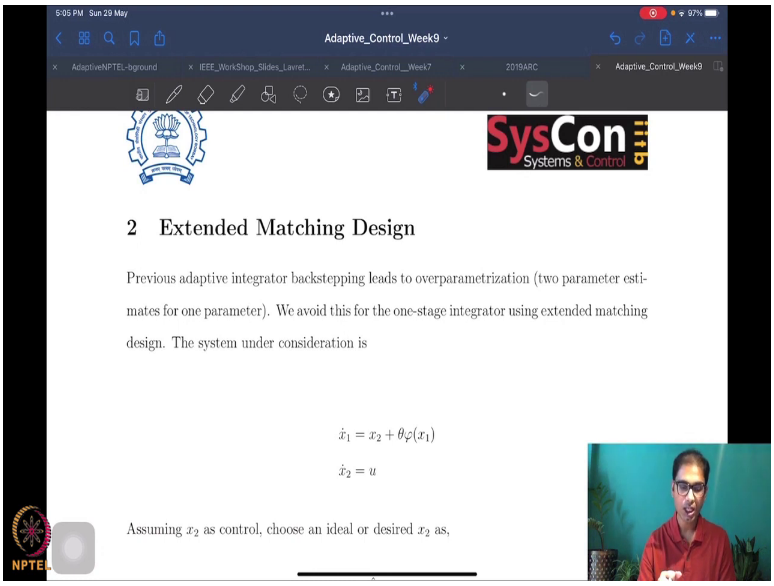


So, if you have a system like this, it is fine right because the control the control is one step below one integrator below if you may, from the unknown parameter. So, this method can also be generalized to the you know sort of the more complicated situation where you have unknowns you know, that sort of appear you know, in several levels above the control. And the way you would do it is to do it sequentially.

Of course, I mean, this is essentially what is called the parametric strict feedback form. And the idea here is that you would think of x like x3 as the control and work with x1 x2 and then so on and so forth. I mean, you move forward further down. And similarly, until you reach xn being the control and then you finally have u as the controller itself.

So, this is of course, these details are available in the K. K. K. book, which is the Kanellakopoulos, Kokotovic, Krstic adaptive control book which is one of the important key references for this backstepping based adaptive control design. In fact, the book has significantly more details and significantly more methods and examples and what we do in these lectures, I would in fact strongly recommend that all of you do look at that material.

(Refer Slide Time: 24:12)



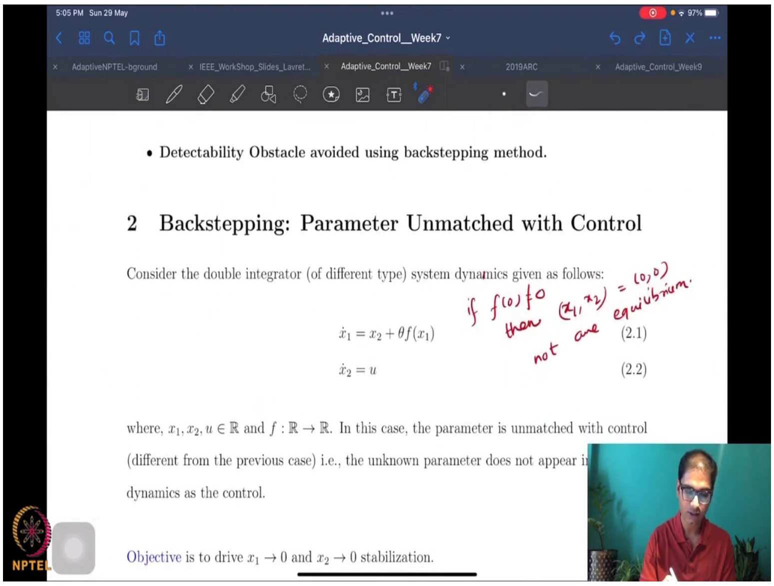
Now, as I already mentioned, one of the big issues is that requirement for extra parameters, extra parameter estimates. And this is a you know, significant constraint when we talk about real implementation, typically, if you have 100 parameters, for example, yeah, I mean, a typical engineering system may have even 500 unknown parameters.

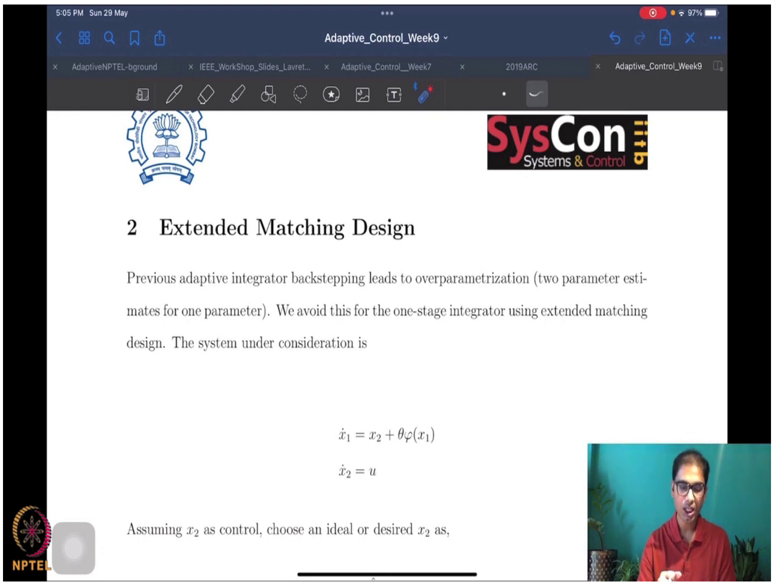
And if you are talking about such large numbers of unknown parameters, and you have double the number of states required to be estimated, so, if you 500 parameters, you actually estimated using 1000 states, then you can imagine that this is a real concern. So, you can imagine this is a real concern. So, that is the idea that of extended matching design is how to overcome this sort of over parameterization.

So, I will not do it, I will not start it now in this lecture. But the system that we consider is almost exactly identical. That is you have this kind of a setup, again, we go back to the scalar case, because everything is easier in the scalar case in the vector case, although it is the same, it is just a lot more book keeping, and it becomes more complicated to explain things.

So, otherwise everything is exactly the same. What we will try to do is do one vector example also at a later point. Yeah, so let us not worry about that too much. But the system is again the same scalar system that we saw in week 7, I mean, let me go back and try to match itself.

(Refer Slide Time: 25:50)





So, this is the exact same system. If you see these two looks exactly identical. And now, we want to do extended matching design, which means that we do not want to have multiple parameter estimates anymore, and just one, and that is what you will see in the subsequent lecture. Great.

So, what did we look at today, we had started the generalization of the backstepping design last time. Now, until the week number 7 lectures, we had looked at the backstepping method for the matched case, that is when the uncertainty appeared in the control dynamics. And also, had looked at the case where the uncertainty appeared one integrator level above the dynamics in which the control was present.

Now, of course, the case when the uncertainty appears in the control, it can be easily generalized to the vector case, it is not a big deal. Yeah, although we did not do that, again, we can try to do all of this at a slightly later stage. Now, the point is that here, we look at a generalization of the unmatched case, that is when the uncertainty appears one level one integrator level above the control.

And we want to look at vector states, vector controls and vector unknown parameters. And that is what we did in today's lecture and also the previous one. So, we completed the proof, we started with the assumption that the first layer integrator has a nice adaptive controller along with the stabilizing Lyapunov candidate construction.

And what we can actually show is the construction of the complete candidate Lyapunov function for the system when integrator layer is added to this, and we can also show that you know, our backstepping error goes to 0. And you can also show that you know, whatever your W that is this negative semi definite function we had from the first integrator layer will also go to 0. So, this is what we have been able to show today.

And in the subsequent lecture, we will look at what is called the extended matching design, great. So, I really hope that all of you are with me, I know that the book keeping part does make things look rather complicated, but I hope you understand that it is just a matter of closely following what is going on and not really any big innovation in terms of theory.

And so, you know, I hope that all of you enjoyed these lectures, and you are able to follow what we are doing here in these courses. Great, so this is where we will stop today. And I will see you again soon. Thank you.