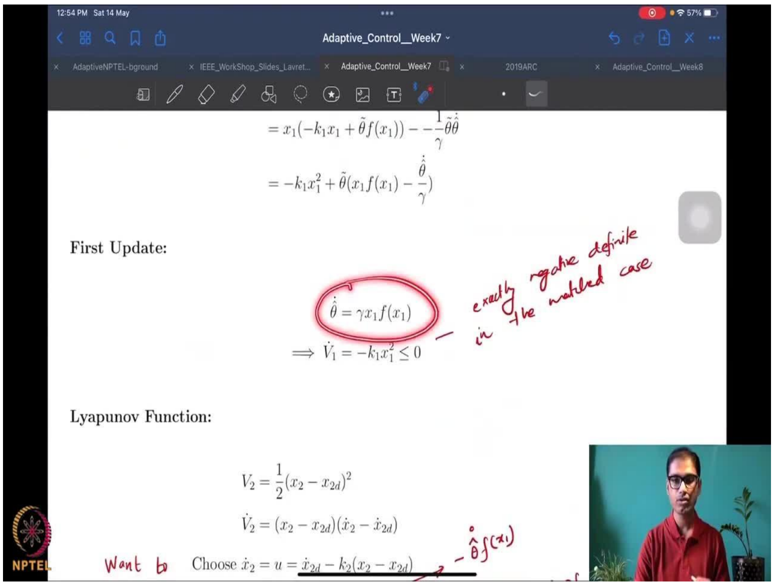
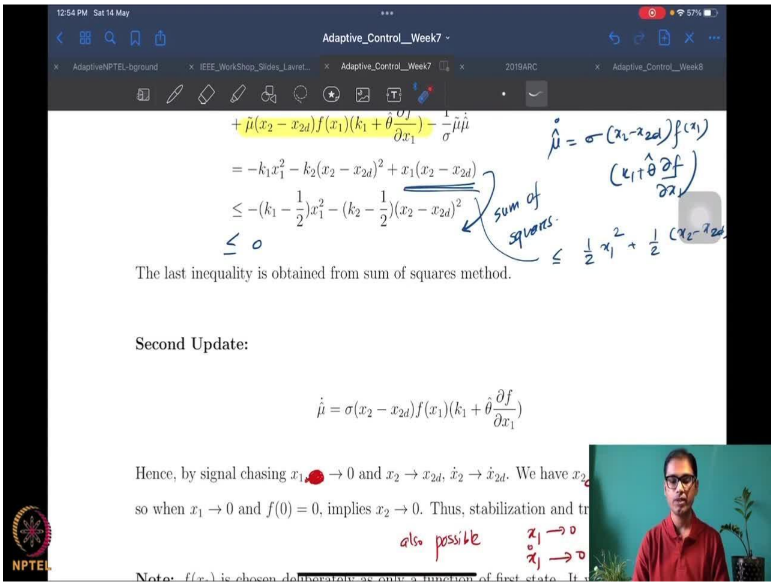
Nonlinear Adaptive Control
Professor Srikant Sukumar
Systems and Control
Indian Institute of Technology, Bombay
Week 7
Lecture No: 41
How to Deal with Unknown Gains in Control

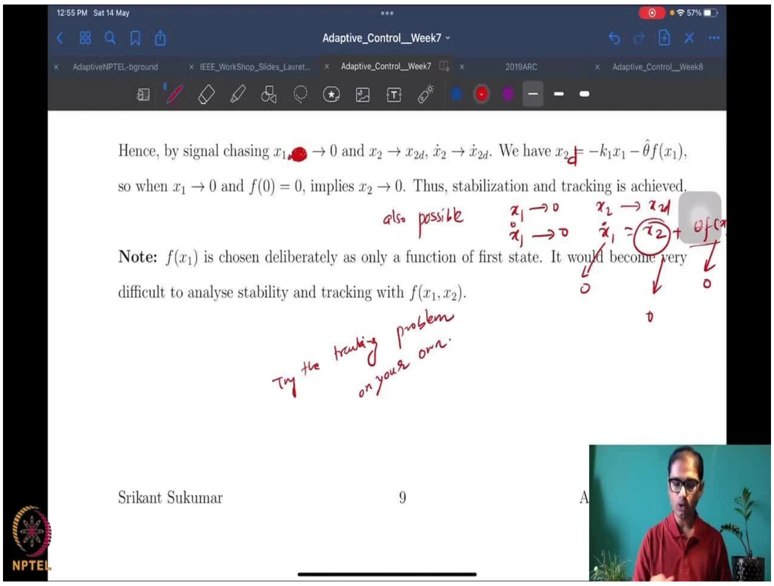
(Refer Slide Time: 0:16)

Hello, everyone, welcome to get another session of our NPTEL on nonlinear and adaptive control, I am Srikant Sukumar from systems and control IIT Bombay. So, we are well into the seventh week of this NPTEL course. And we are already looking into algorithms for designing adaptive controllers for single and double integrators matched and unmatched uncertainty systems. So, the sort of algorithms that we are designing are not so far from the kind of algorithm design that will be required to drive autonomous systems such as the satellite in orbit that you see in the background.

(Refer Slide Time: 1:06)





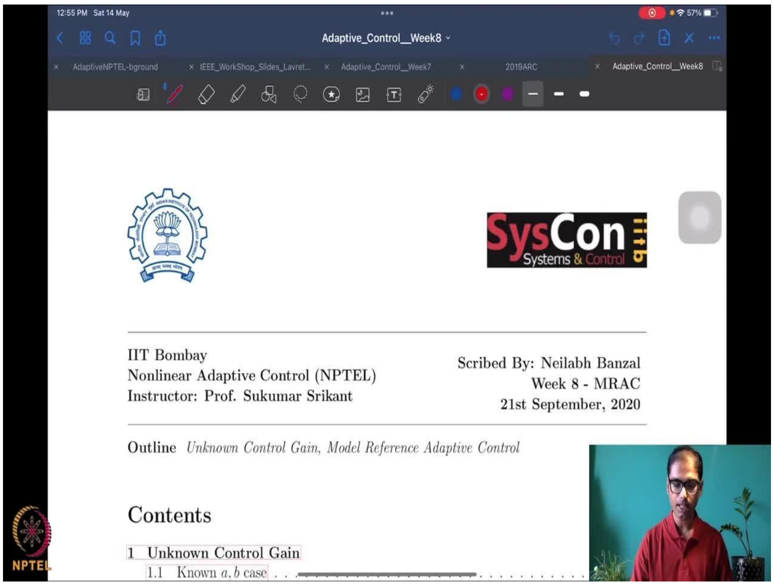


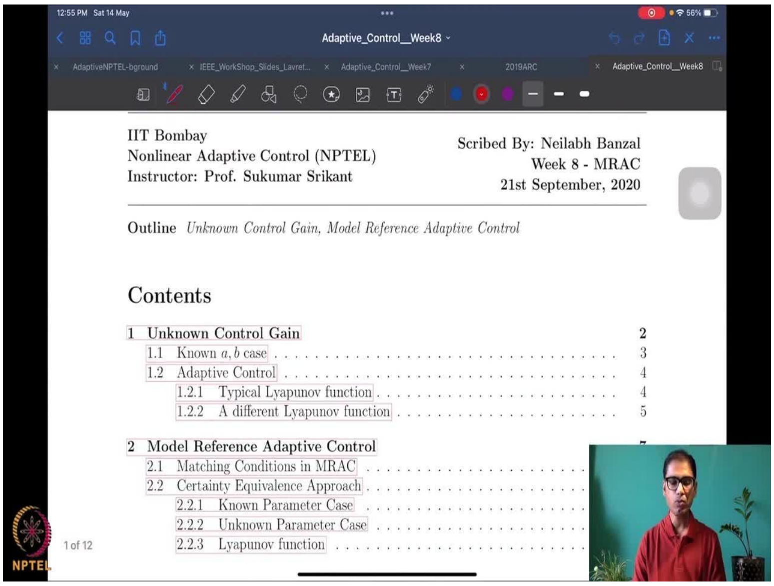
So, what we were looking until last time was the design of the adaptive controller for the unmatched uncertainty case. So this we were, we had essentially looked at how to use backstepping based methods in order to design an adaptive controller for systems that look like this, where the uncertainty is now in the state that is different from the state in which the control appears. So, this is what is called an unmatched parameter system. So, we of course, saw the backstepping design, and it was a little bit more complicated than what you would expect for the matched case.

And we also saw that one of the most important or key things to remember is that we had to design 2 different parameter update laws, which is the theta hat, and also the mu hat. So, we had to design the theta hat, and the mu hat, that is we had to over parameterize the system. So, we had to over parameterize the system in order to be able to sort of complete the adaptive control proof. So, in this case, although the uncertain parameter was only one scalar parameter, we had to design 2 different estimates for the same parameter.

So, of course, then we were able to prove tracking, or we did the stabilization problem, but as we mentioned, the tracking problem also would be a very straightforward extension of the stabilization. In fact, I would encourage all of you to try the tracking problem. So, that is something I would write here, try the tracking problem on your own.

(Refer Slide Time: 3:15)

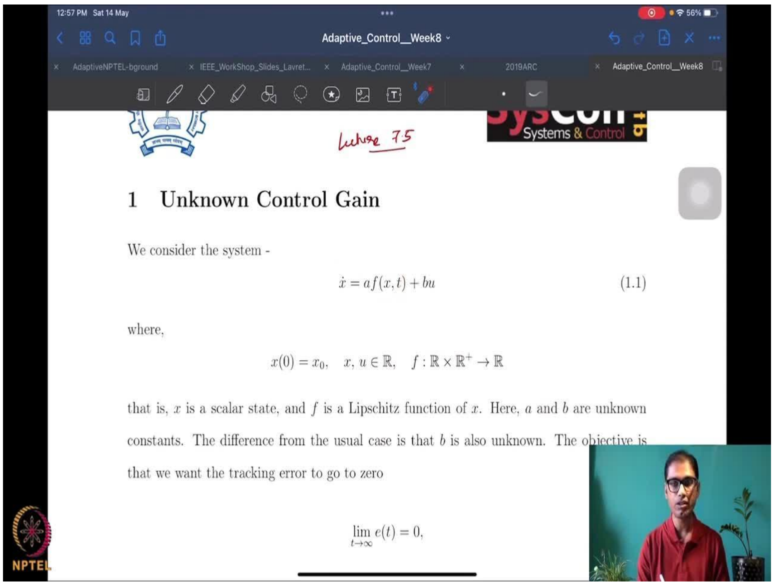


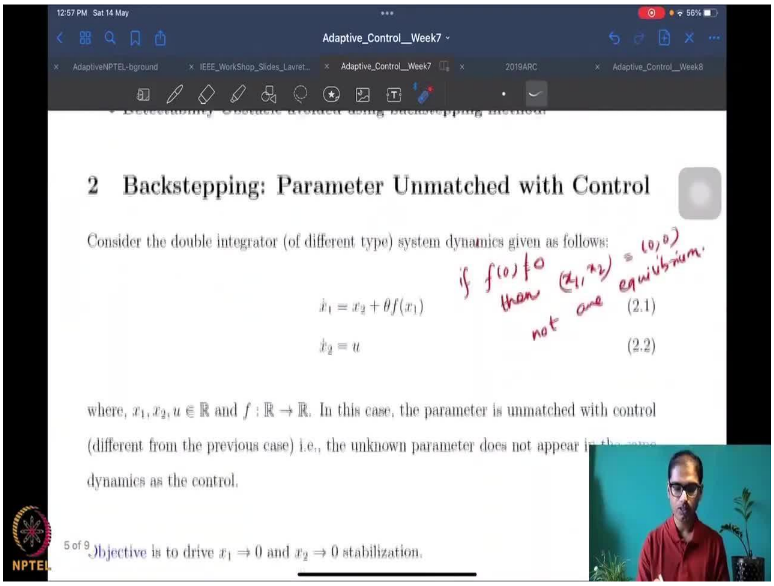


So, now we move on to the next set of notes. I mean, again, please do not mind that it says, week 8 and so on, we have been doing this. So, this is merely to sort of organize the material so we are sort of once we have completed one set of notes for that particular week, we have moved on to the next set of notes and we will continue to do this because there is enough material to cover till the end.

So we will move on to a slightly different topic or set of topics now. So, we want to start off with the unknown control gain problem. So, we will look at what is the unknown control gain problem, so this is what we want to do in hopefully the next couple of sessions. And we want to look at what is the unknown control gain problem, and how do you handle it? And, and as always, we look at the scalar versions of things here. And once we have done this unknown control gain problem, we will be in a very good position to actually introduce and look at the model reference adaptive control, which is probably one of the most popular adaptive control frameworks for linear system, so that is sort of the outline of what is in this subsequent lectures.

(Refer Slide Time: 4:38)

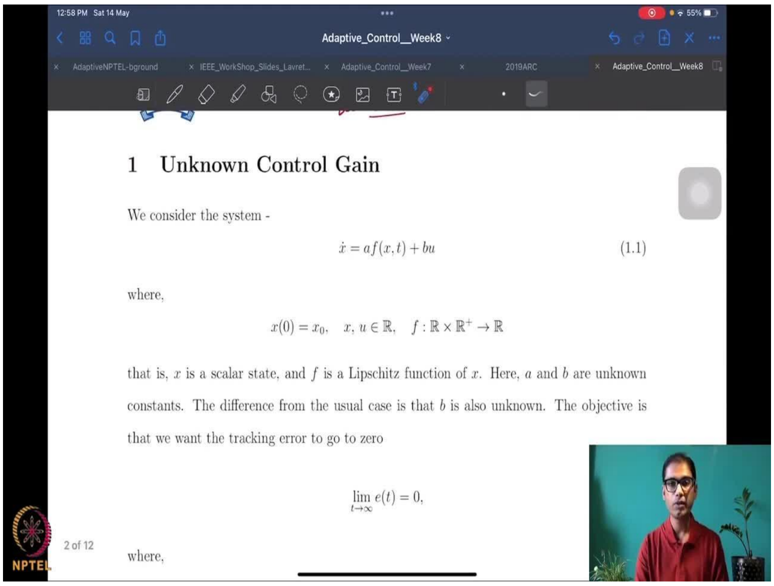


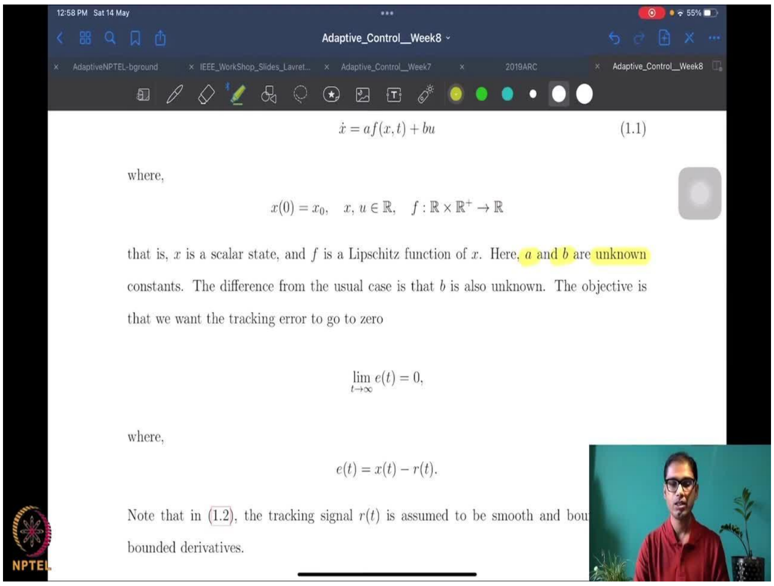


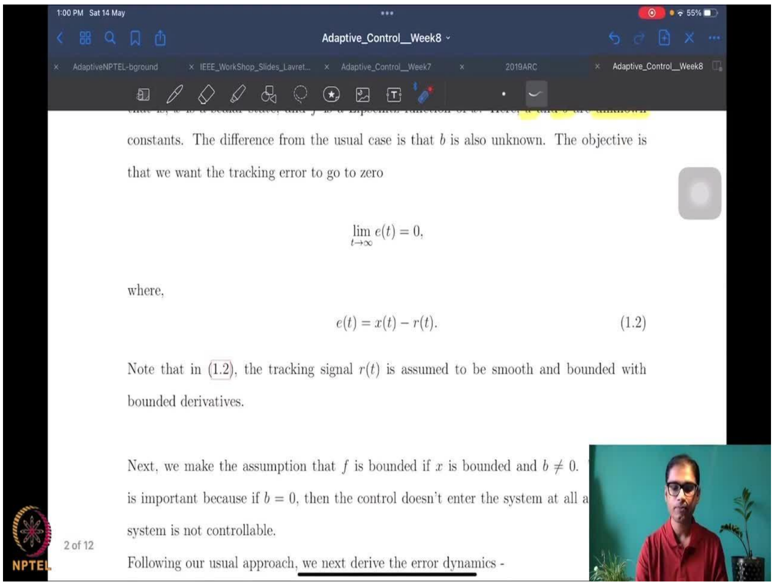
So, let us start looking at this. So I am going to mark this lecture as lecture 7.5. So this is the fifth lecture of the seventh week, so do not worry that it says week 8 in the beginning here. And that is merely for organization sake. So, what is the unknown control gain problem? So, until now, we have been looking at unknowns that are not in the control term, we will be looking at the unknowns in the drift term. So, like in the previous week we had I mean in the unmatched case also we had the unknown which was connected to the drift term or the state terms, there was never any unknown connected to the control term and that is what changes here, that is what changes here.

Now, we are again looking at the scalar single order system, just to make our treatments simple. Again, we will immediately after look at the model reference adaptive control case, which is the entire vector case, so, we do not have to worry about, missing any details as such, or this would be not applicable to any real system you are work working on. This is in fact, with minor little extensions applicable to a lot of vector systems also.

(Refer Slide Time: 6:06)







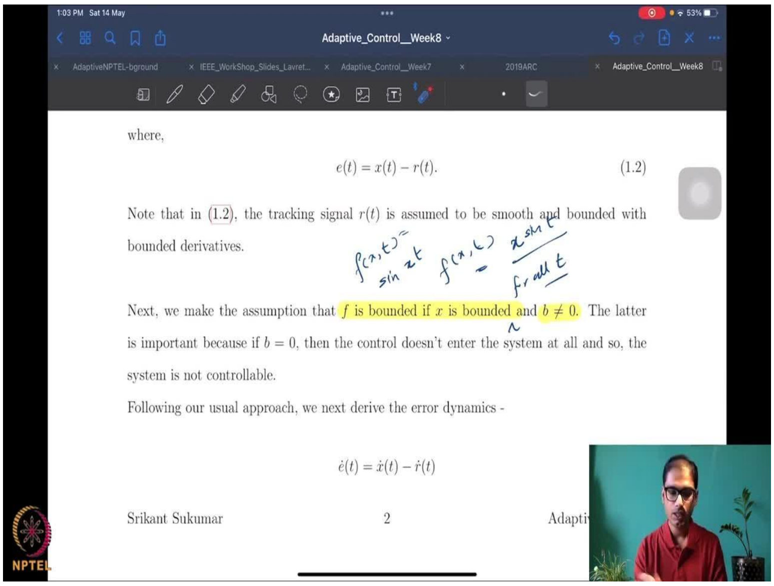
So, let us look at this system. So, what is this? This is x dot is a f x, t plus b u. Now, x0 is given to x0 is the initial condition, of course, and both the state and the control are in real numbers. And the function f is of course, taking the state and the non-negative time and giving you a scalar quantity R. And here, the important thing to remember is that both a and b, a and b are unknown constants. In this case, there is an unknown connected to the state, which is what we have been seeing until now, but there is also an unknown which is connected with the control. So, this is new.

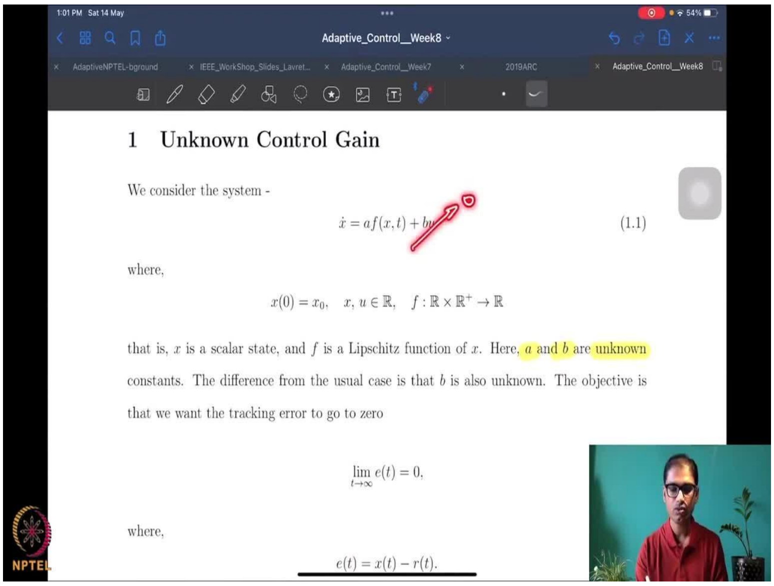
And so then the question is, can we solve the same problem? So, what is the same problem? The objective is of course, that we want the tracking error to go to 0, that is, we want t goes to infinity e of t goes to 0, where e is defined as the tracking error, which is the state minus the r. So, there may be some non-uniformity in us putting the time arguments with the states and r and so on and so forth. Please do not worry too much about this notational inconsistency if you may, we have just put in the time arguments wherever we think it is not very clear from the context.

So, when it is clear from the context, we do not put any arguments, but if it is not, we are just putting in some arguments of time. So, it is not nothing to worry about. If you want to be of course, you want to write articles, and journal papers and conference papers in the field, then you have to be more careful and be very consistent with your notation, but here the purpose is instructional and so we are keeping notation so that everybody can follow. So, the aim is more that you understand the material then be very precise and exact and consistent with the notation. So, the purpose of the notation is, so you can follow carefully. So, this is the tracking error, we always define this tracking error, because this is the quantity we want going to 0.

So, as usual we assume that our, the signal that we want to track is in fact, smooth bounded with bounded derivatives, so this is the standard assumption. So, it is infinitely differentiable, it is bounded, and its derivatives are also bounded, so its a very nice signal. We do not want to deal with non-nice signals and so on, I mean, simply because of that, because in most physical circumstances, you would not want your robot or any other mechanical or aero mechanical system, or maybe even an electrical system to follow very bad trajectories. So that is really, what is the aim.

(Refer Slide Time: 9:05)



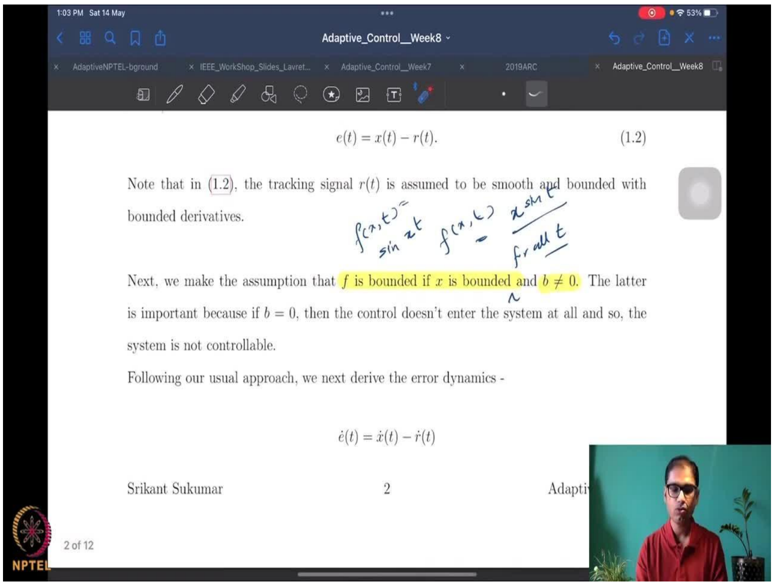


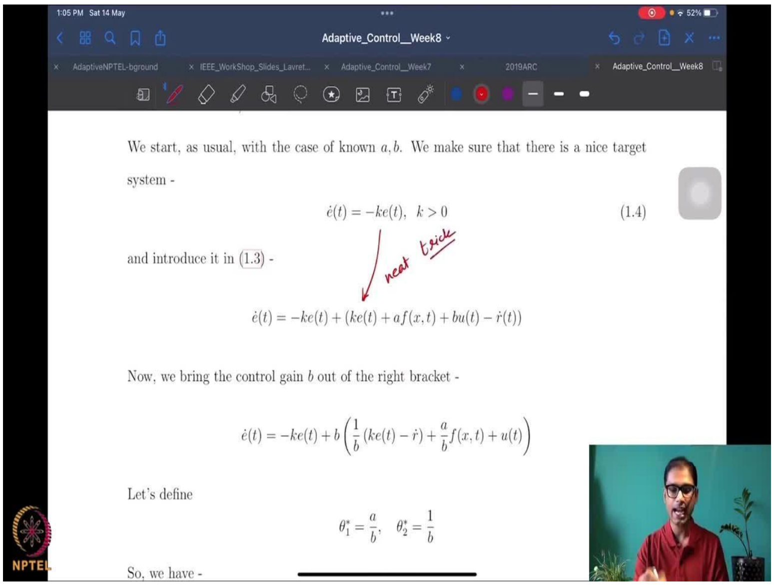
We also make some assumptions on f, so what is the assumption? Is that f is bounded if x is bounded. So, there are 2 assumptions, one is that the function f is assumed to be bounded if x is bounded. So, it is a function of both state and time, but we do not worry about time, so I would in fact, even say that for all t, and we also assume that b is nonzero. That is the gain that is connected to the control is assumed to be nonzero and this is a very reasonable assumption, because if this is zero, then there is no term here, there is no control here, so there is no problem to solve nicely. We cannot do anything, the system becomes uncontrollable. So this is very, I mean, this assumption is a very reasonable and a very fair assumption, so you do not have to worry about making it very infeasible assumption.

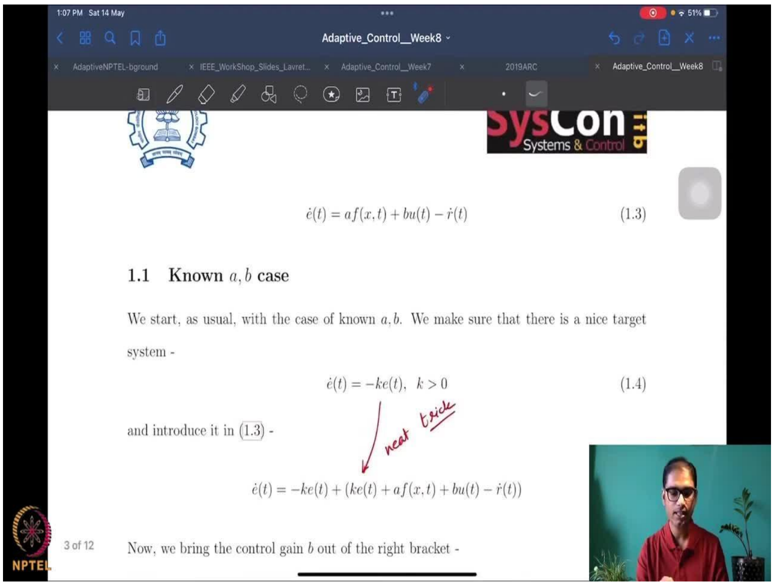
And so is the assumption on f. So, notice that we are already assuming that f is bounded when x is bounded, we are not saying that f is bounded for all states, if x goes to infinity x, the function f is also allowed to go to infinity. So, it is not something like a sinusoid or something, it is not something as simple as a sinusoid. Because if I had something like sine x, whatever, I means, sine x times t or some such thing, this will be always bounded. This f will look will always be bounded, irrespective of, what is x and t, because it is inside a sinusoid.

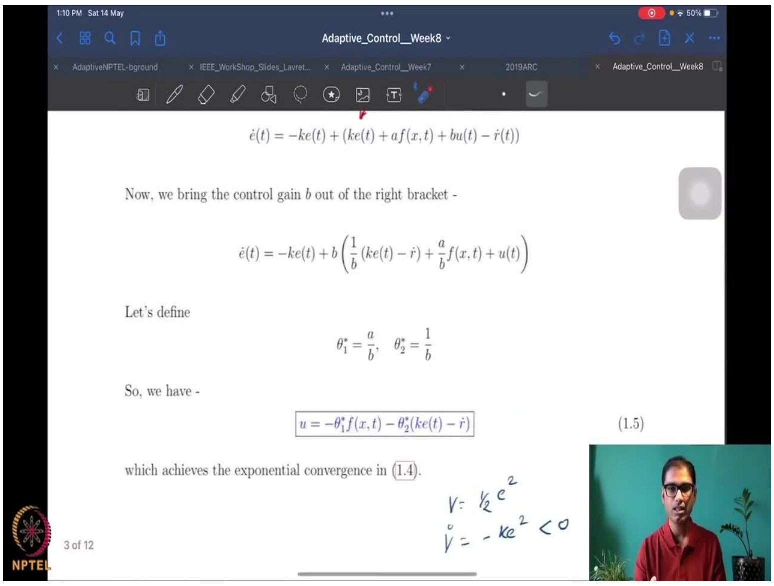
But we do not care for, we are not restricting ourselves to functions like these, we are even allowing something like say, linear in x, we even allow something linear in x. So, the only thing is, you have to remember that it should not grow unbounded with time. So, that is one thing that we sort of have. So, I mean, it could be something like x sine of t, something like this is all also allowed. So, this is a fairly, general enough set of functions, it is a fairly general enough set of functions. So, I would not worry about whether we are restricting ourselves to a very-very simple class of functions or anything like that. So, whenever we talk about the problem set up, when we talk about assumptions, it is sometimes or in most times, it is critical to understand that these assumptions are not very-very restrictive in the system.

(Refer Slide Time: 11:51)









So now, what do we do? We again, do the standard steps, whatever we are used to doing until now, we compute the error dynamics. So, that is e dot is x dot minus r dot. And what is that? It is a f x plus b u minus r dot. So, this is just obtained by plugging in the dynamics of x. So now, when as usual we do our design with the known case first, this has always been, I mean, method of how we have been doing things. And so for the known a, b, case we want to have a very nice target system, we have always been doing that too, we will be choosing a nice target system.

And what is a nice target system if I want to drive e to zero asymptotically? It is e dot equals minus k e for some k positive. And this is the simplest possible linear, in fact, exponentially decaying system that I can think of. So, we have also already worked with a system like this. And if you want to do this, what would I do? I would, so what we sort of do is we introduced this term, the minus k e term in the dynamics here. Earlier, we simply chose the control, but now we do not directly choose the control, just bear with me, because of what how we want to use the structure.

So, what we do is this minus k e is the nice term, so I introduce it in the e dot. And then I also add a plus k e, so that these 2 just cancel out, and I have not really changed anything in the right hand side. So, this minus k e is there and then I look at all these terms together because I want to sort of cancel this also. So, what do I do? I take the control gain b outside, so this is a trick. So, this is a, I mean I would qualify this as a neat trick. And lot of nonlinear control design is about such nice, neat, cute little tricks. So, do not, I mean, one should not be feel that this is just, a magical wand that I am waving these tricks are based on very good intuitions.

So, it is a neat little trick, and lots of nonlinear control. In fact, a lot of mathematics is based on doing such neat little tricks. In fact, a lot of proofs in mathematics that you will find are accomplished because somebody came up with a trick. So, just because it is we use the word trick is not derogatory, it should have actually come up with these tricks often you will be a very-very good nonlinear control theorist.

So, this is a nice trick, we just add and subtract this k e, and then this k e is clubbed with this guy. How do we club it? We take the control gain b outside the bracket and then I look at 2 different terms, there is one term where it is just a 1 by b. Because there is no unknown here in these 2 terms, so there is no unknown here, it is just a 1 by b. And there is another additional unknown here. So there is an a by b, f. and then there is just a u because the b was pulled out of the bracket.

So, this is a sort of rewriting of this term by pulling the b out of the bracket, this is also another trick that we pulled the b out of the bracket, technically, you would have to do it, even for the known case, because if I prescribe the control out of this, my control will have been the denominator. And so we are essentially doing that which is a writing it as such. Because we want to rewrite or redefine the unknown parameters that is the end, although as of now, we are saying that this known a, b case, but we still want to redefine the unknown parameter redefine the parameters of the system.

And how do we do that? I now choose a over b as theta 1 star, and 1 over b as theta 2 star. So, instead of a and b being my unknowns, I have 2 new unknowns, which is theta 1 star and theta 2 star defined as such. Now, remember that this is not over parameterization, I had 2 parameters to begin with and I have 2 parameters now. So, I am not doing any kind of over parameterization, I am simply doing a redefinition of the parameters. In fact, it is very easy to see in this particular case, may not always be the case, in this particular case, it is very easy to see that if I exactly estimate theta 1 star and theta 2 star, then a and b can be uniquely computed, a and b can be uniquely computed, so not a big deal.

So, we understand all this. So some tricks, and some redefinition of parameters. And what do we have, I choose my control in terms of these new parameters now, because these are known, and that is minus theta 1 star f to cancel this out, and minus theta 2 start k e minus r dot, cancel this out. Remember, I do not need my control to introduce any new term. Because I already have a nice negative term here. So all I need to do is cancel these guys out. And once I do that with the control, I am only left with this much. So, I achieved the target system for the known case.

This is exactly the same expression of control, the control expression is the same if I had started here, with the original system, this guy. and I had this target system is exactly the same control you would have obtained. I have just rewritten the control in terms of these predefined parameters and this is going to be very helpful to me subsequently.

This is subsequently going to be very-very helpful, that is the whole reason why we are doing this redefinition. So, this is the control and with this control, I know that I am left with only this. So, of course, I choose a, I have not written that here. But I will choose a V, which is one half e square, and I will get a V dot, which is minus k e square, which is negative definite. So, this is what we have been doing. So, this known case does not seem to significantly different except for some redefinition of parameters.

(Refer Slide Time: 18:40)

Now, we bring the control gain b out of the right bracket -

$$\dot{e}(t) = -ke(t) + b\left(\frac{1}{b}(ke(t) - \dot{r}) + \frac{a}{b}f(x,t) + u(t)\right)$$

Let's define

$$\theta_1^* = \frac{a}{b}, \quad \theta_2^* = \frac{1}{b}$$

So, we have -

$$\boxed{u = -\theta_1^* f(x,t) - \theta_2^*(ke(t) - \dot{r})} \tag{1.5}$$

which achieves the exponential convergence in (1.4).

$V = \frac{1}{2} e^2$

$\dot{V} = -ke^2 < 0$

1.2 Adaptive Control

Now, applying CE, we have -

$$u = -\hat{\theta}_1 f(x,t) - \hat{\theta}_2(ke(t) - \dot{r})$$

for the unknown parameter case, where, $\hat{\theta}_1, \hat{\theta}_2$ are estimates of θ_1^*, θ_2^*.

Now, the closed loop dynamics of the tracking error is -

$$\dot{e}(t) = -ke(t) + b\left(\tilde{\theta}_1 f(x,t) + \tilde{\theta}_2 (Ke(t) - \dot{r})\right)$$

where,

1.2 Adaptive Control

Now, applying CE, we have -

$$u = -\hat{\theta}_1 f(x,t) - \hat{\theta}_2(ke(t) - \dot{r})$$

for the unknown parameter case, where, $\hat{\theta}_1, \hat{\theta}_2$ are estimates of θ_1^*, θ_2^*.

$\dot{e} = -ke + b\left(\theta_1^* f + \theta_2^*(ke - \dot{r}) + u\right)$

Now, the closed loop dynamics of the tracking error is -

$$\dot{e}(t) = -ke(t) + b\left(\tilde{\theta}_1 f(x,t) + \tilde{\theta}_2 (Ke(t) - \dot{r})\right)$$

where,

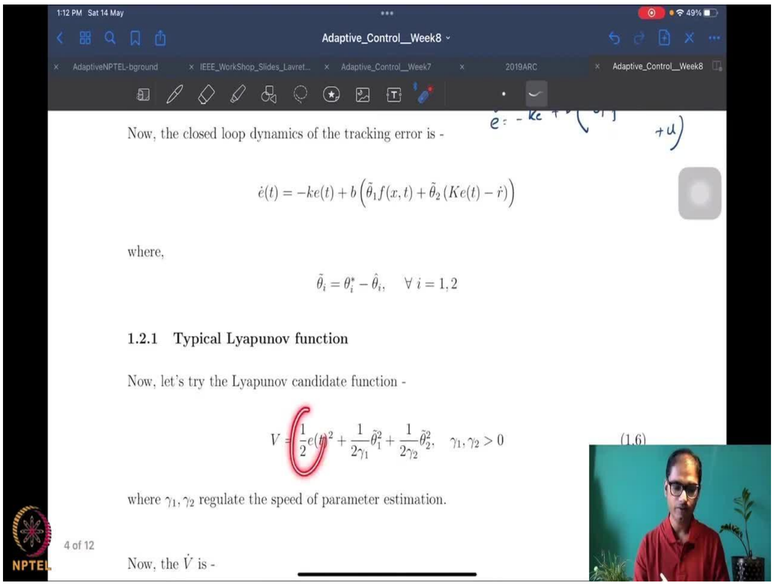
$$\tilde{\theta}_i = \theta_i^* - \hat{\theta}_i, \quad \forall\, i = 1, 2$$

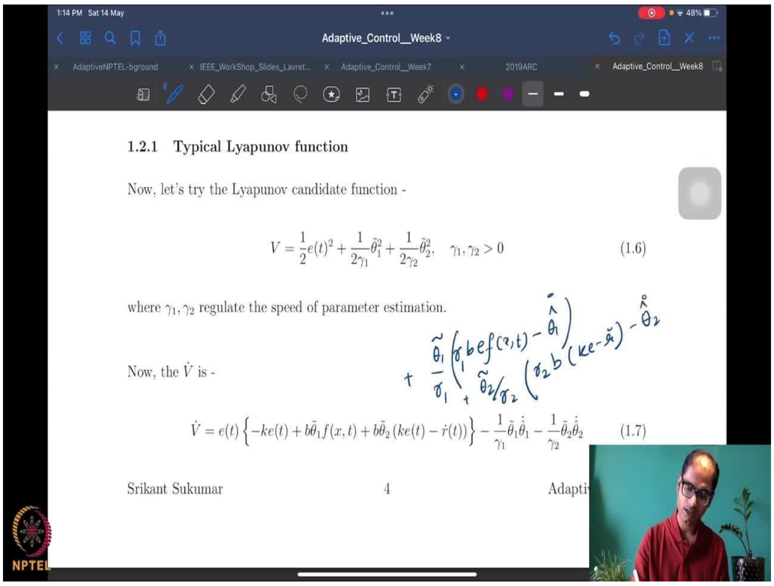
1.2.1 Typical Lyapunov function

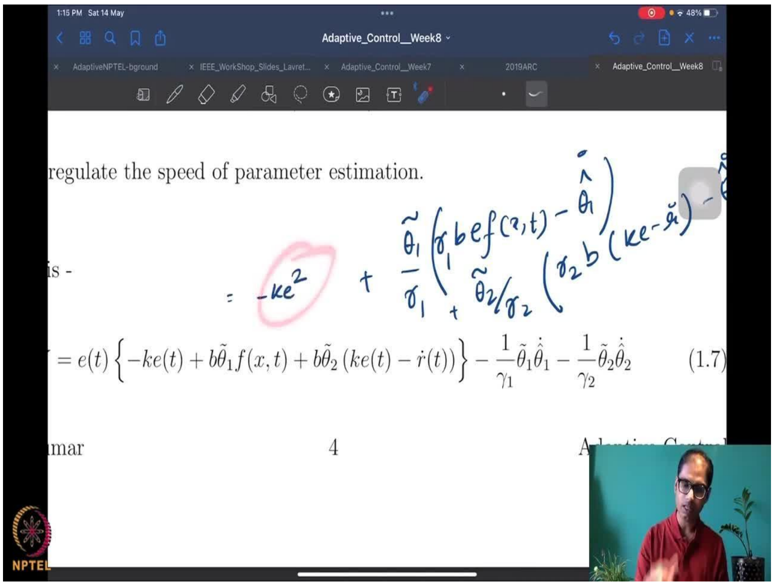
So now, we move on to the unknown parameter case. So what do we do? We take the control law here, in equation 1.5. And we replace the theta stars by the hats, because this is just what certainty equivalence principle has taught us. And this is what we have been using for so long, successful, so we do not want to change a successful formula or success formula. So, we replace the theta stars by the theta hats. And these are of course, the estimates.

Now what happens, of course, I do not get the desired e dot equals minus k e, but I do get that plus b times the tilde terms. This is I mean, this is very easy, I hope this does not confuse you. So, because if you look at this e dot is minus k e plus b times theta 1 star f plus theta 2 star k e minus r dot sorry, plus control. Now, if you plug in this control here and you keep in mind this definition it is very easy to see that the hats will give you theta 1 tilde here, this hat will give you theta 2 tilde here. And that’s it this terms remains as it is, this remains as it is and that is all we have.

(Refer Slide Time: 20:17)





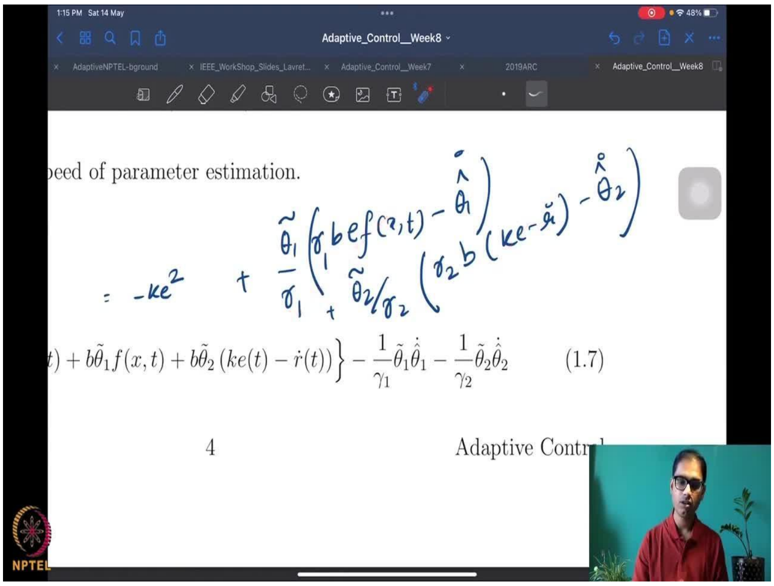


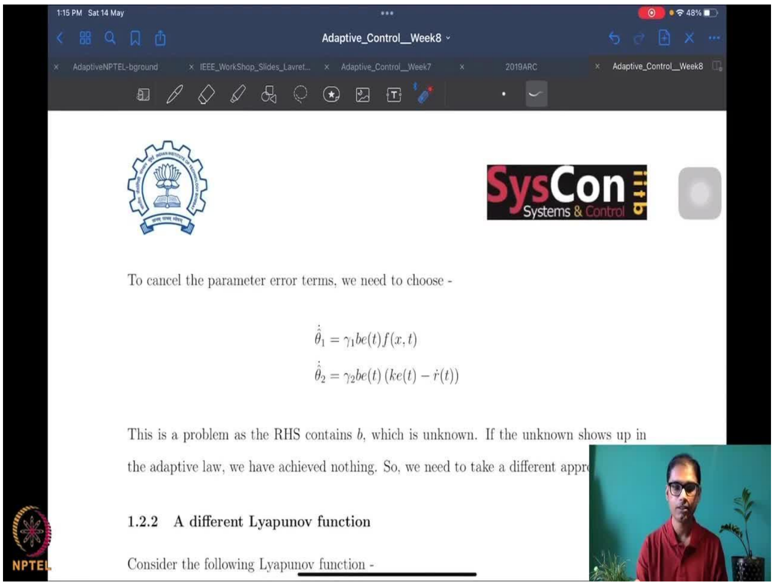
Now, the question is, how do I do the analysis now? Well we try and we try our usual idea, what do we do? We take the earlier Lyapunov of candidate, which was half e square, and we add to it quadratic terms in the unknowns, this is what we have been doing with some, of course, adaptation gain in each case, so this is what we have been doing. So, we want to continue doing something similar. So, then we start to compute the V dot very carefully. This contains e edot. And so e dot is simply this minus k e plus b times theta 1 tilde f plus b theta 2 tilde times this, this is based on the control law we have.

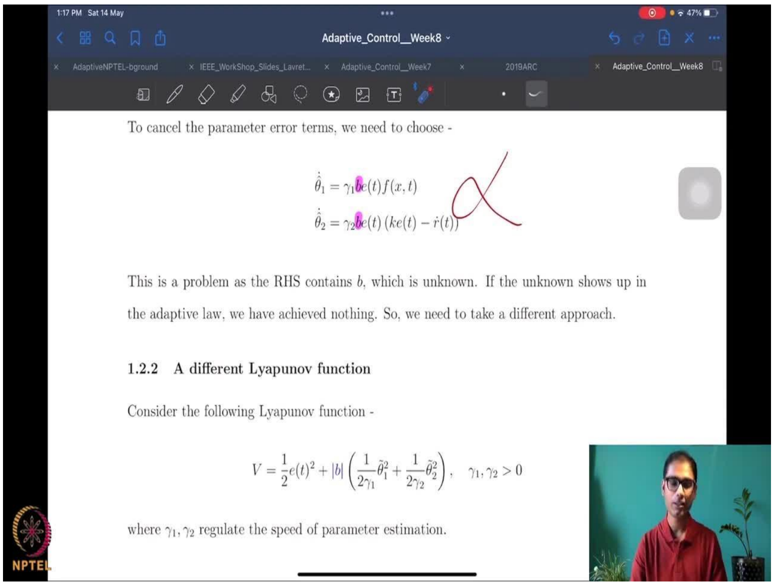
Then we have derivatives of these terms, which comes out to be minus 1 over gamma 1 theta 1 tilde theta 1 hat dot, and minus 1 over gamma 2 theta 2 tilde theta 2 hat dot. And we are of course, yet to define the update laws, we are yet to define the update laws. Now, again, what do we do we have this nice negative term coming from here, but then we have these theta 1 tilde and theta 2 tilde term. So, of course, we try to club these 2 guys and these 2 guys. That is what we have been doing again, we have just been clubbing the theta tilde terms and using the theta hat dots to cancel out the errors and whatever these terms.

So, in fact, I will write it out because it is not written here. So, what will this be? This will be theta 1 tilde times b e f x, t. I will take a gamma 1 in the denominator, so that I get a gamma 1 here and minus I get a minus theta 1 hat dot. And similarly, I have a theta 2 tilde divided by gamma 2 times gamma 2 b and k e minus r dot minus theta 2 hat dot. So, these are the terms I get in theta 1 tilde, and theta 2 tilde. I want to and the other term is of course, is equal to minus k e square. So this is, of course, the nice term, nothing to do, but I want to get rid of these 2 terms.

(Refer Slide Time: 23:02)







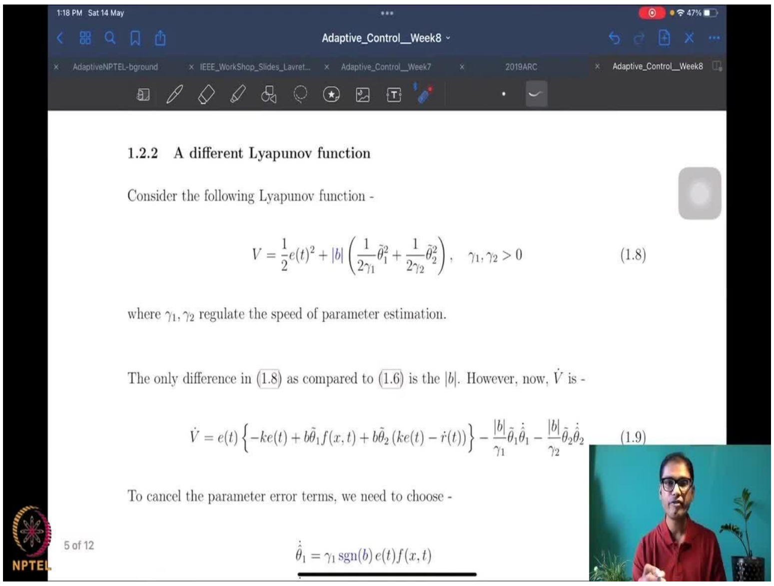
And if I do want to do that, I just simply use theta 1 hat dot and theta 2 hat dot here. But, look at what happens, something rather bad actually. Theta 1 hat dot is gamma 1 b e f x t and theta 2 ha dot is gamma 2 b e k e minus r dot. Everything looks nice except the b here, this b is bad, this b here is bad, this b here is a problem. It is a problem. Why? b is in fact, an unknown. And theta 1 hat and theta 2 hat are essentially being designed and defined so that we can identify this b. And so if we are still have an update law, and I notice that this theta 1 and theta 2 appear in the control so it is not like these are for show, these are not for show. These have to be actually implemented even if you have a microcontroller render on a robot, then these have to be implemented in the microcontroller, these have to be integrated in the microcontroller.

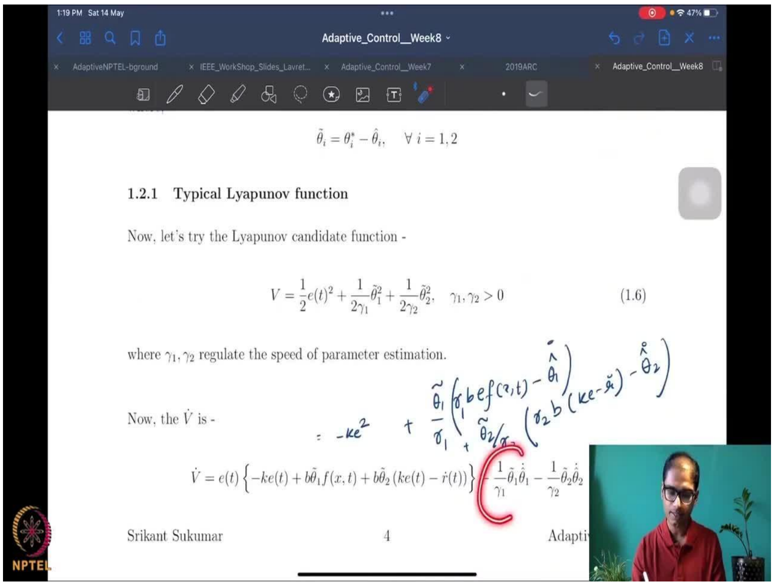
So obviously, these are not for show. And they contain the unknowns themselves. So this is a circular problem. That I am trying to identify the unknowns using this theta 1, theta 2, which contains the b and the a, but then the b appears in their derivative and that is not okay, of course, this is no adaptive control. So, this is not an adaptive control that can be implemented, so not implementable. So, that is why I have put a big cross.

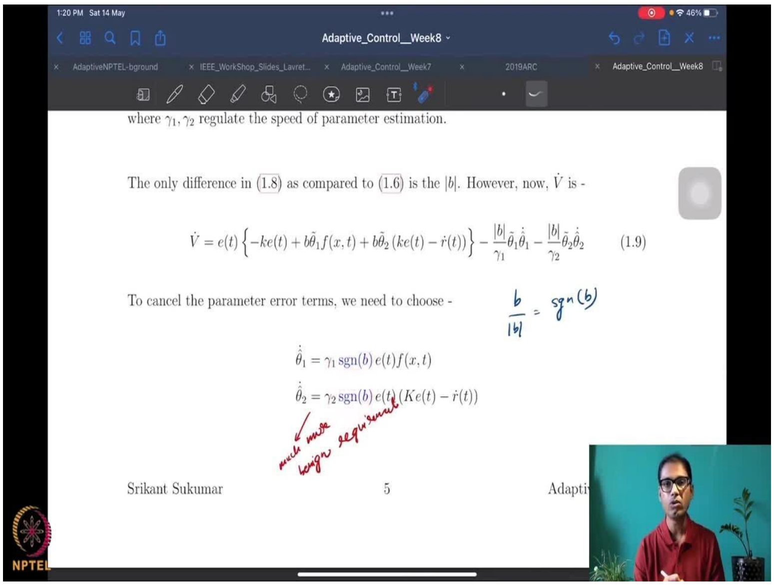
So whenever I get any adaptive control solutions from unfortunately several naive students and I find that and so the first thing I look for is if the unknown appears in the update or not. Yes I do not have to look at the rest of the solution at all, my checking becomes very easy, that, because if you ended up with an unknown in your adaptive laws, and you made the wrong design, your design is something that cannot be implemented. So, please do not make my life easy, please make it hard and design adaptive laws, which do not contain the unknown. So, this is wrong.

The question is what went wrong? Did we design the controller wrong, did we design the Lyapunov of candidate wrong, but it seemed like we had followed the steps that, we were doing until now. So, the problem is that the steps that we were doing until now, do not work for this situation, so we have to make a modification. The modification is not in the control law, which is still the certainty equivalence control law, so we do not make a change there. But we have to make a change in the Lyapunov function.

(Refer Slide Time: 26:12)





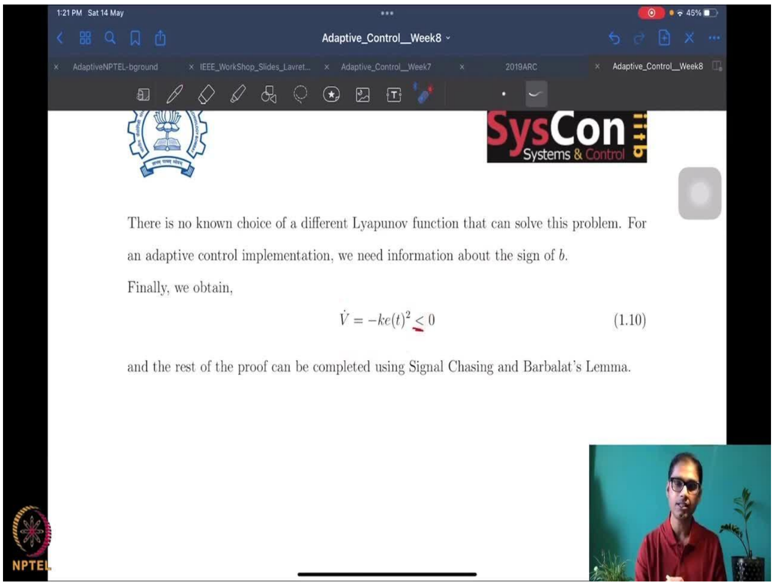


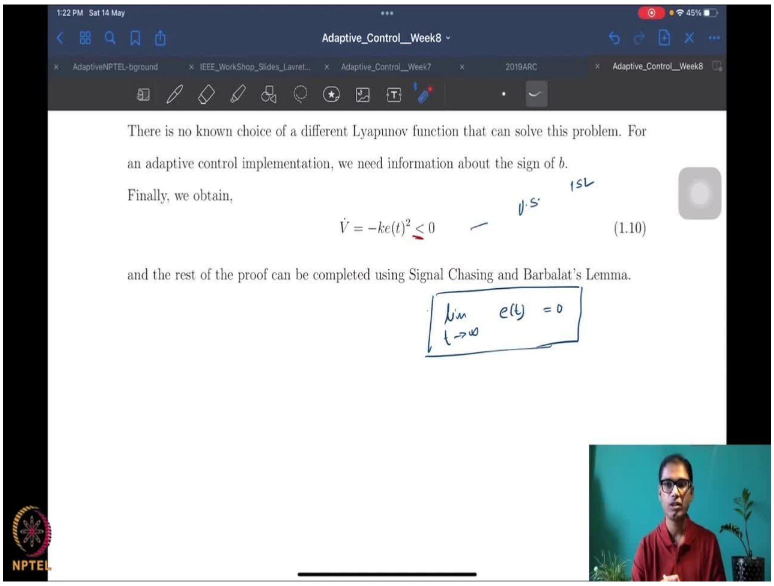
So here, we had chosen Lyapunov function, which is very much motivated by what how we have been doing things until now, with this just a quadratic term and adaptation gain here. But now we have to do something different, we actually have to introduce an absolute value of b here, we have to introduce an absolute value of b here, everything else remains the same and just this new term shows up, just this new term shows up.

So, what happens, everything else is same again, the only difference here, the only difference that we see here is that your because of this absolute value of b which is again a constant, remember, this is a constant, all parameters are assumed constant, there is no derivative with respect to it, but the second and third terms, which were simply these guys, in equation 1.7, now have an absolute value of b here. So, when I again combine this term and this term, my update law does contain something about b, but it is not the b itself because we use the fact that b divided by absolute value of b is simply signum b. I hope you understand this, b is simply equal to the absolute value of b times the signum of b.

So, although b does not appear now, which was wrong, signum b appears here. So, everything else remains the same just that instead of the b appearing which was absolutely not available, I have signum b appearing here, this I would say is a much more benign requirement, this is a much more benign requirement. What is the requirement? The requirement is that there is a signum of b that I need to know, in order to implement this adaptation law.

(Refer Slide Time: 28:35)





So, the amazing thing is that in the entire adaptive control community, there is no other known choice of Lyapunov function or control, which will help us to not know the value of signum b while implementing this control. So, the only solution that is available in the community is what I have shown you, and if you can come up with something that does not require the sign of b either, then you have done some magic and you will get a grade. So, this is actually something that is not possible yet. So, if you can come up with something excellent.

So anyway, so with the knowledge of the sign of this control gain, you can implement this adaptive controller and you will get V dot as minus k e square, which is of course now not less than but less than or equal to 0. It is not less than 0, this is only semi definite because there were 2 other states theta 1 tilde and theta 2 tilde. And therefore, you can complete the rest of the proof with Barbara's Lemma and signal chasing. In fact, you can show that limit as t goes to infinity, e of t is 0, the steps are standard, you will just show that e is integrable.

First, you obviously have stability from here I of course, uniform stability in the sense of Lyapunov, no problem, beyond that, I will be able to show that e is bounded that is l infinity, I will show I can show that V is integrable because it is non increasing and lower bounded and therefore, e is an l 2 signal when I integrate both sides. So, I obtain that e is both l infinity and l 2. And then e dot is also l infinity can be easily shown and therefore, by Barbalat’s Lemma corollary you will be able to claim that e goes to 0.

(Refer Slide Time: 30:41)

So, what we have seen today is how to handle unknown gains on the control in the adaptive control context. So, until now, we had been looking at unknowns only connected to the states that is in a drift terms. But now, we have also seen how one can handle controls, sorry unknowns in the control vector field. And this is again something that is very-very naturally occurring, if you have any sensor, or sorry, any actuator which is misaligned, or you do not know the exact position or the orientation for example, if I have a spacecraft with a control moment gyroscope has an actuator to help it rotate about its axis and then I do not know exactly where it is placed in the center of mass or if it is oriented exactly along the body axis or not, then there will always be an unknown control gain.

Or you can also think of a quadrotor type situation where the thrust is actually some coefficient multiplied by square root of angular velocity of the propeller. Now, this coefficient may again be an unknown quantity or not a well measured quantity and therefore, this also gets into the control gain and is unknown. So, this is a very common problem and we have given an adaptive control solution to control design in this context, so we can still achieve tracking. So, we could stop here, and I will see you again next time. Thank you.