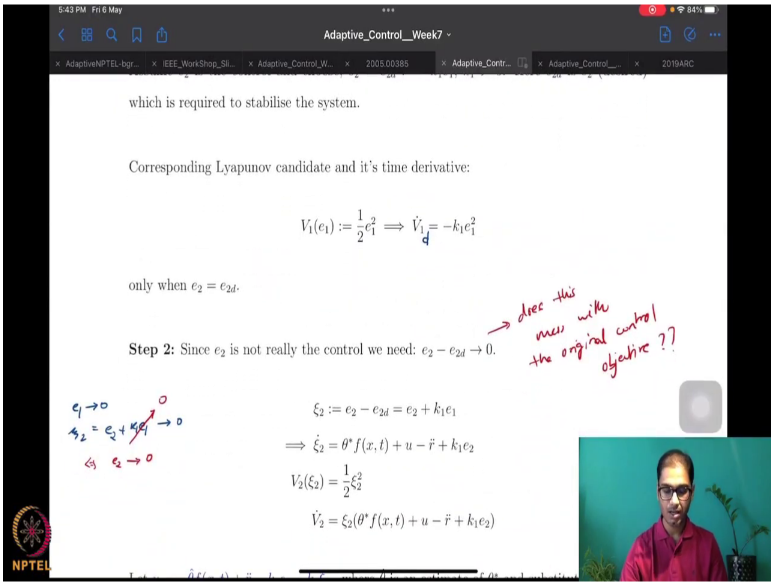
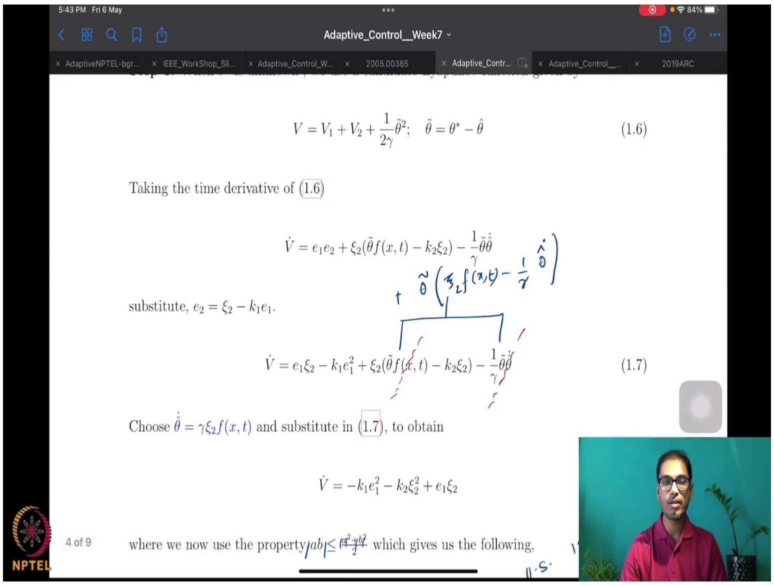
Nonlinear Adaptive Control
Professor Srikant Sukumar
Systems and Control
Indian Institute of Technology, Bombay
Week 7
Lecture No: 39
Backstepping in Adaptive Control: Parameters Unmatched with Control
Part 1

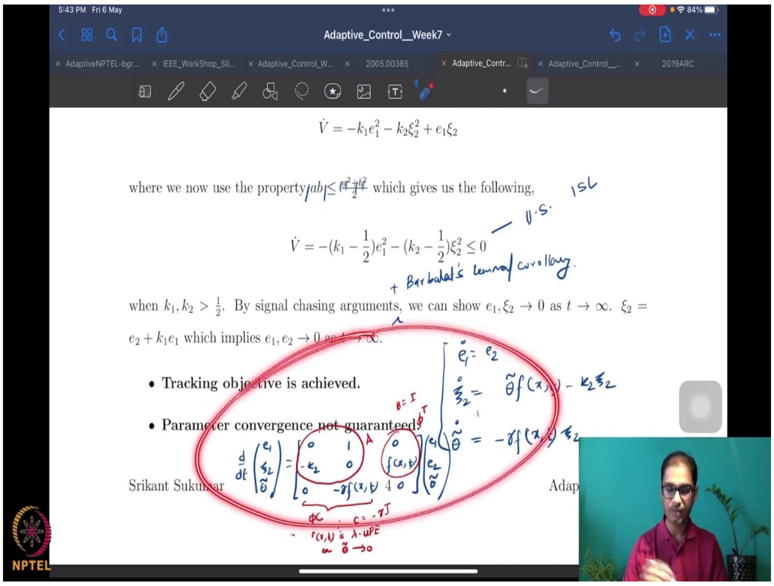
(Refer Slide Time: 00:15)

Hello, everyone, welcome to another session of our NPTEL on Nonlinear and Adaptive Control. I am Srikant Sukumar from Systems and Control, IIT Bombay. In this week number 7, we are already underway into learning algorithms that are expected to drive systems such as what we see in our background an orbiting spacecraft around the Earth.

(Refer Slide Time: 00:44)

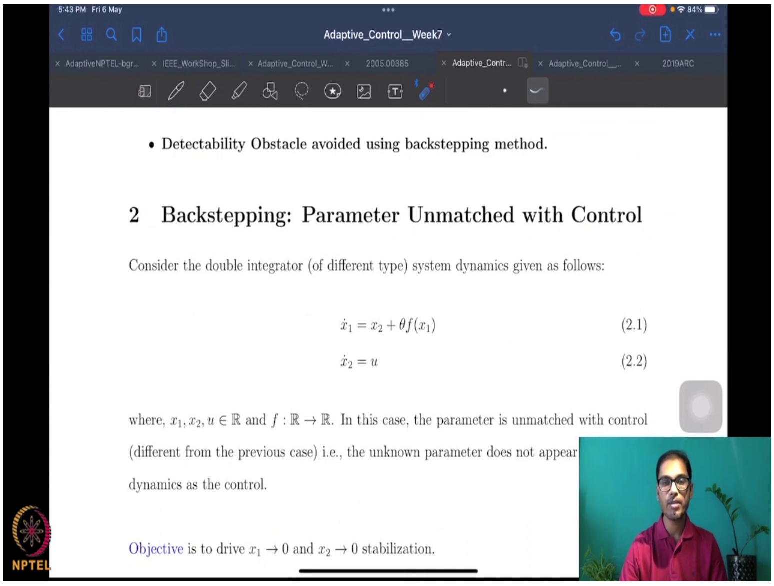


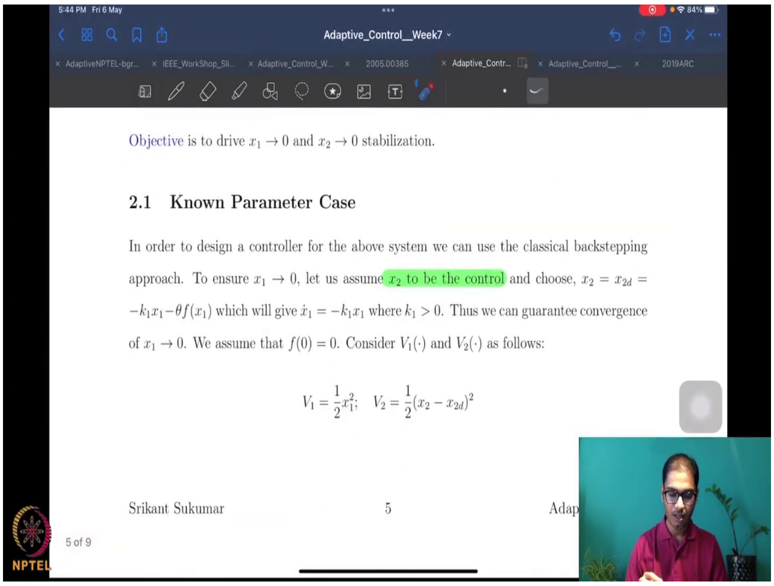




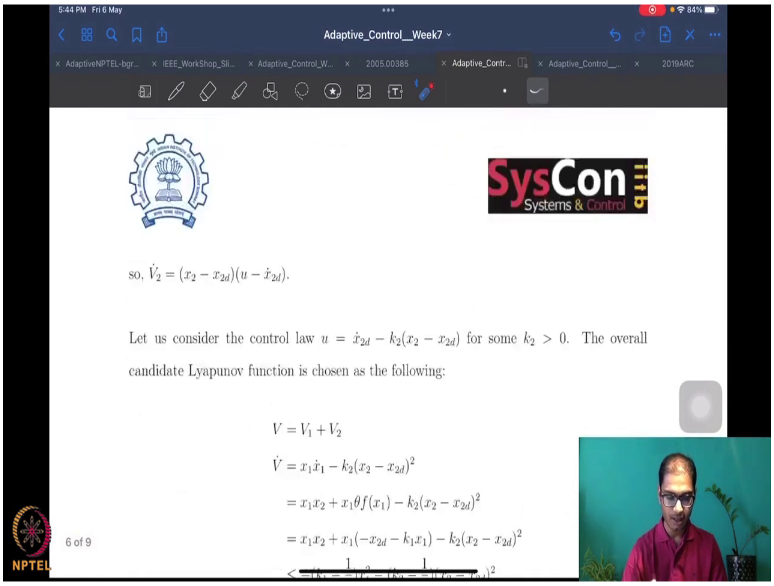
So, what we were doing until this last time is to look at first the matched uncertainty case, that is when the parameter was in fact, matched with the adaptive controller. And we actually learned how to use a backstepping based Lyapunov design to construct control and an update law and therefore an adaptive controller for such a system. And we also, of course, just the stability analysis, we also looked at a little bit of how to talk about, persistence. So, so basically, we did all that for the matched case.

(Refer Slide Time: 01:29)

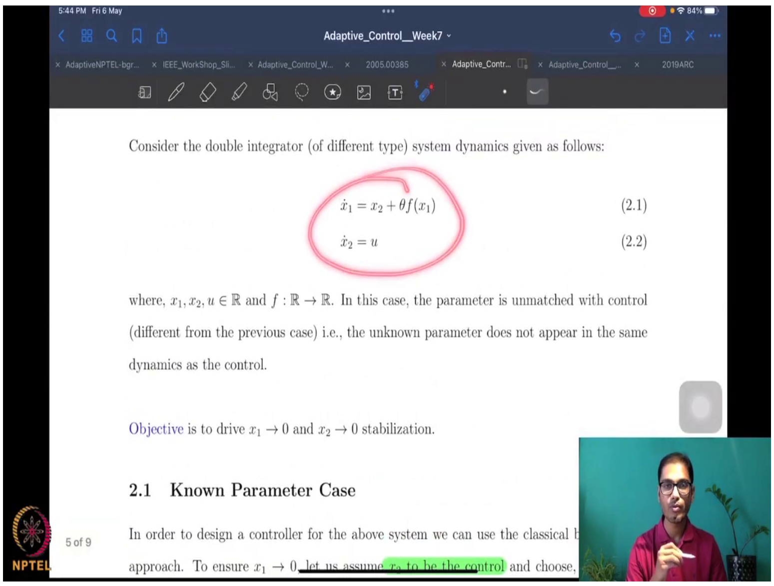




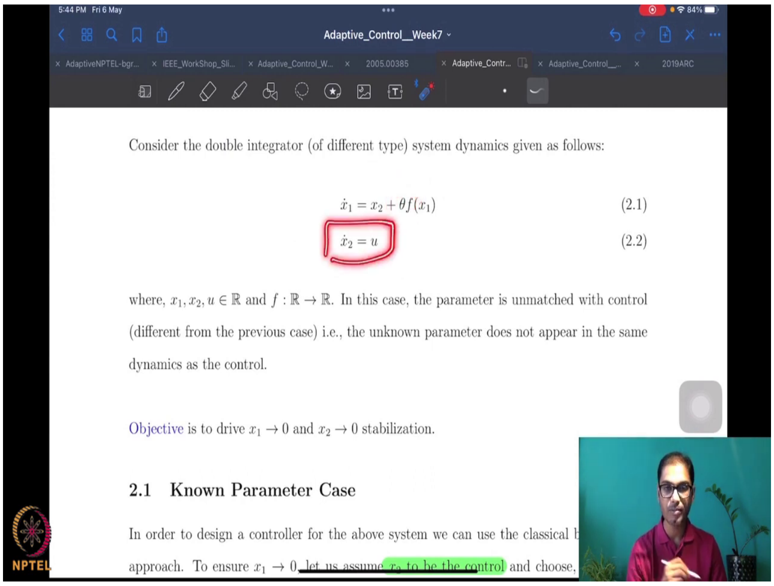
5:44 PM Fri 6 May
Adaptive_Control__Week7
Objective is to drive $x_1 \to 0$ and $x_2 \to 0$ stabilization.
2.1 Known Parameter Case
In order to design a controller for the above system we can use the classical backstepping approach. To ensure $x_1 \to 0$, let us assume x_2 to be the control and choose, $x_2 = x_{2d} = -k_1x_1 - \theta f(x_1)$ which will give $\dot{x}_1 = -k_1x_1$ where $k_1 > 0$. Thus we can guarantee convergence of $x_1 \to 0$. We assume that $f(0) = 0$. Consider $V_1(\cdot)$ and $V_2(\cdot)$ as follows:
$V_1 = \frac{1}{2}x_1^2; \quad V_2 = \frac{1}{2}(x_2 - x_{2d})^2$
Srikant Sukumar
5
Adap
5 of 9
NPTEL



5:44 PM Fri 6 May
Adaptive_Control__Week7
SysCon
Systems & Control
so, $\dot{V}_2 = (x_2 - x_{2d})(u - \dot{x}_{2d})$.
Let us consider the control law $u = \dot{x}_{2d} - k_2(x_2 - x_{2d})$ for some $k_2 > 0$. The overall candidate Lyapunov function is chosen as the following:
$V = V_1 + V_2$
$\dot{V} = x_1\dot{x}_1 - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1\theta f(x_1) - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1(-x_{2d} - k_1x_1) - k_2(x_2 - x_{2d})^2$
6 of 9
NPTEL

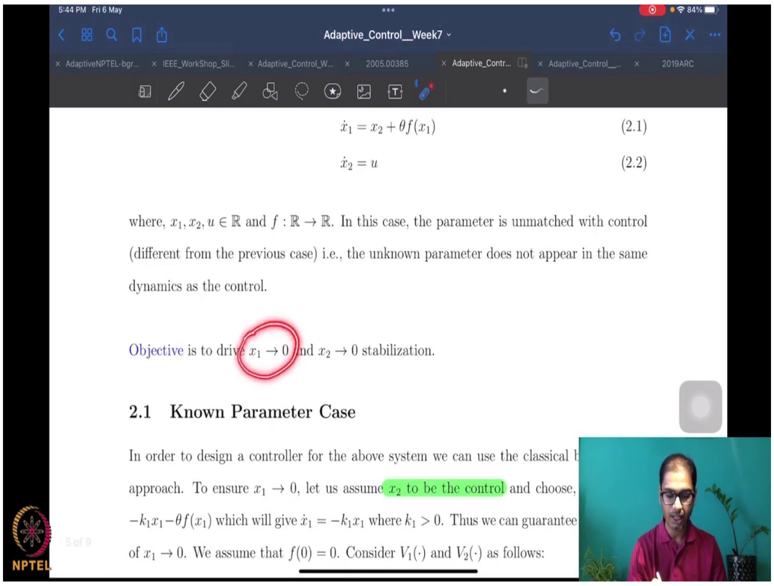


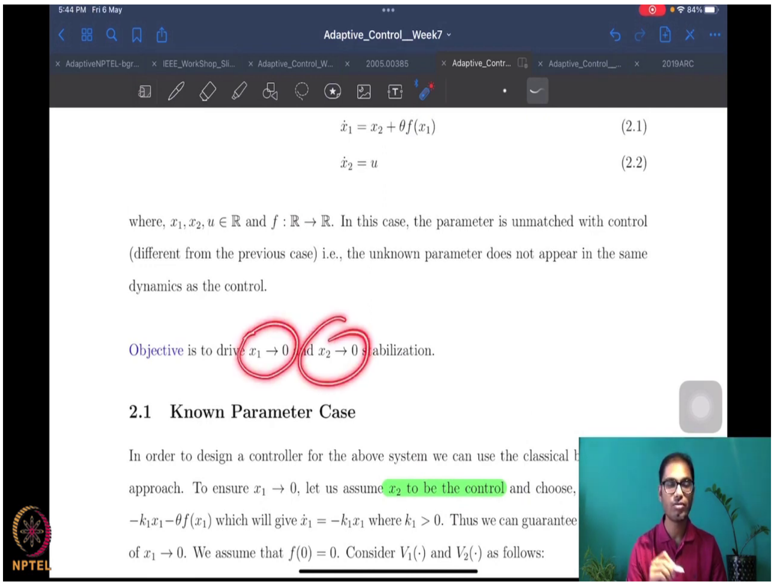
5:44 PM Fri 6 May
Adaptive_Control__Week7
Consider the double integrator (of different type) system dynamics given as follows:
$\dot{x}_1 = x_2 + \theta f(x_1)$ (2.1)
$\dot{x}_2 = u$ (2.2)
where, $x_1, x_2, u \in \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$. In this case, the parameter is unmatched 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
In order to design a controller for the above system we can use the classical b
approach. To ensure $x_1 \to 0$, let us assume x_2 to be the control and choose,
5 of 9
NPTEL



And then, we moved on to the case of parameter unmatched with the control, And the first again, looked at the known parameter case, in order to understand how the backstepping design will work. So we got a fair idea of how the control would be. So basically, we just started doing this, so for the known parameter case, we in fact, first looked at the model, where we have the parameter now in the first state, instead of the second where the control appears.

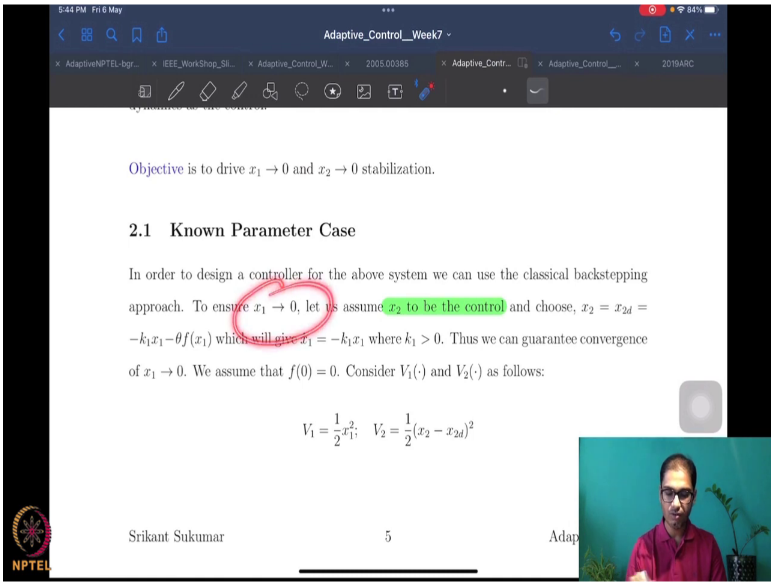
(Refer Slide Time: 02:07)

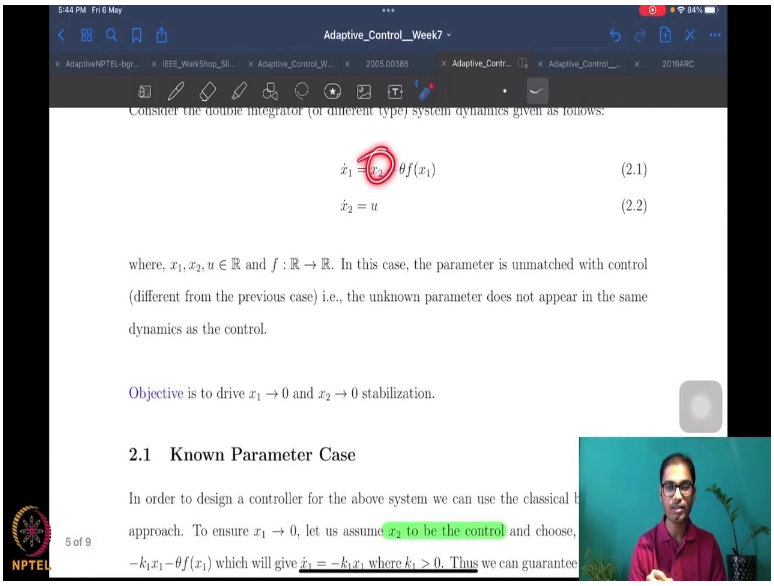


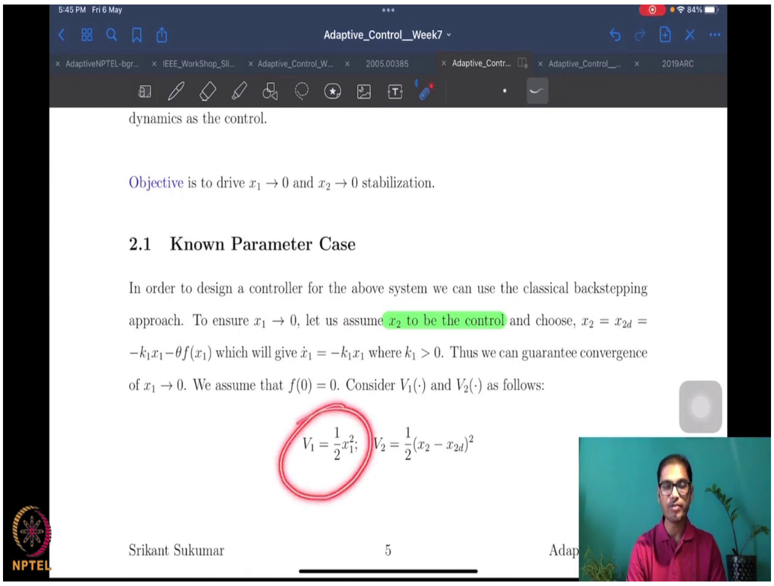


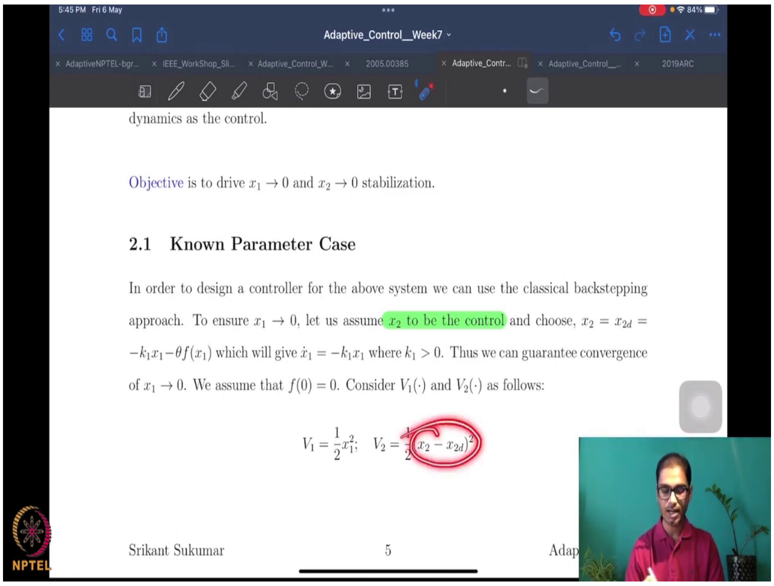
And the aim is the same, which is basically to drive x1 and x2 to 0. So, again, slightly different, because we are not looking at the tracking error problem. But as I said, last time, it says this is exactly the same, even if I was considering a tracking error problem, the methods that we are using would apply, almost identically.

(Refer Slide Time: 02:32)



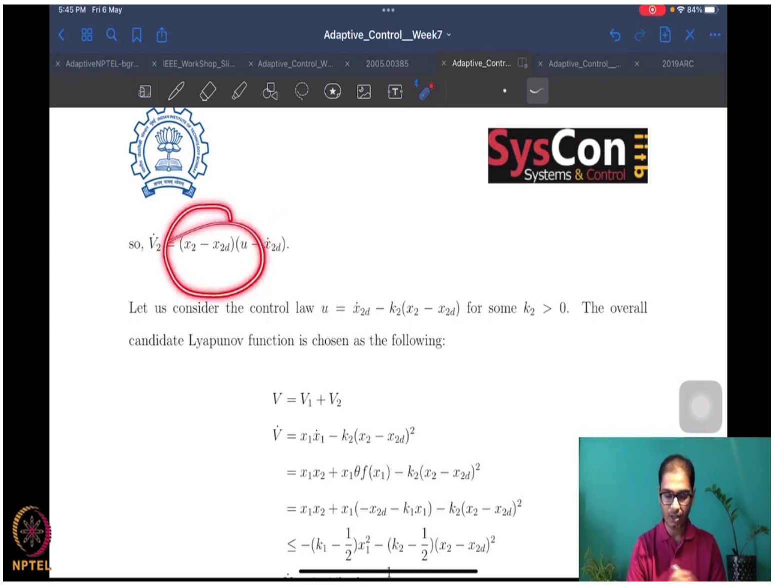


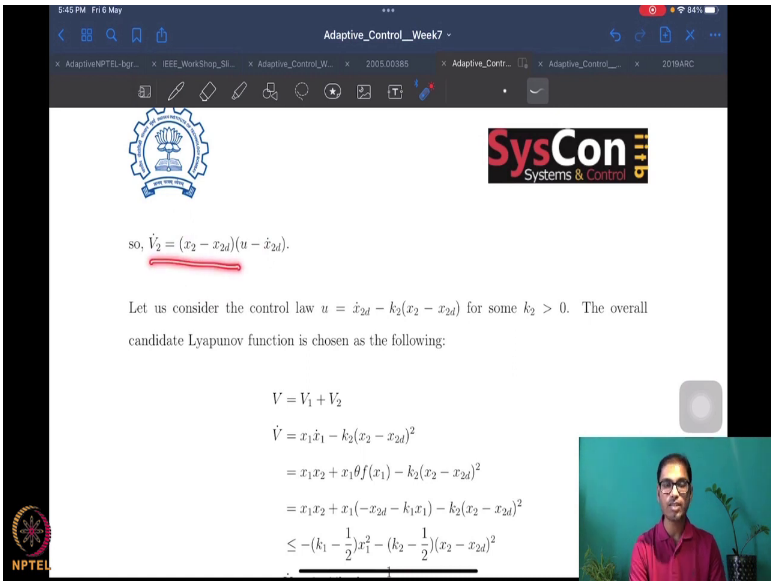




So in order to ensure x1 to, go to 0, we designed a controller, we are assuming that x2 is the control for the x1 state, this is again standard in the backstepping based method. And since the parameter is known, I can apply this kind of a control to get this nice ideal system. And we also choose this nice V1. Now, because we know that x2 is not identically equal to x2 desire, we try to do the next best thing that is to drive x2 to x2 desire. And this is where we get the motivation for choosing the second piece of our Lyapunov function, or Lyapunov candidate function, and that is the error between x2 and x2desire and a quadratic constructed out of it.

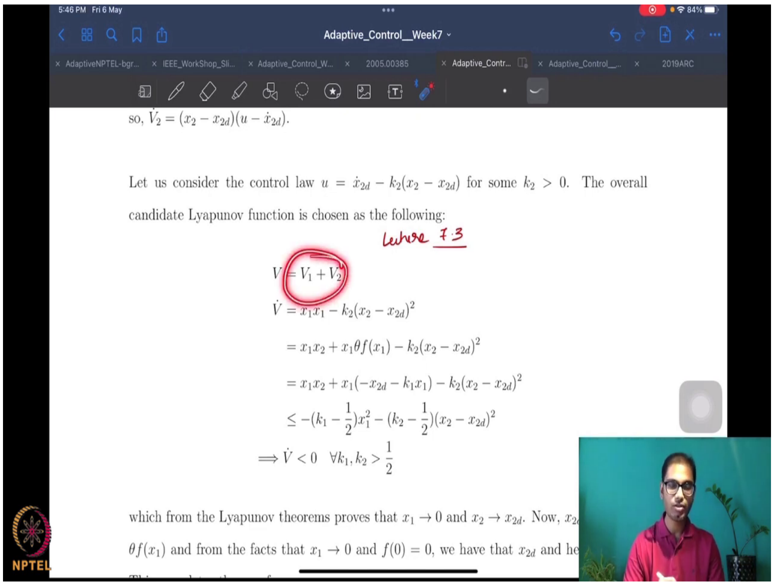
(Refer Slide Time: 03:17)

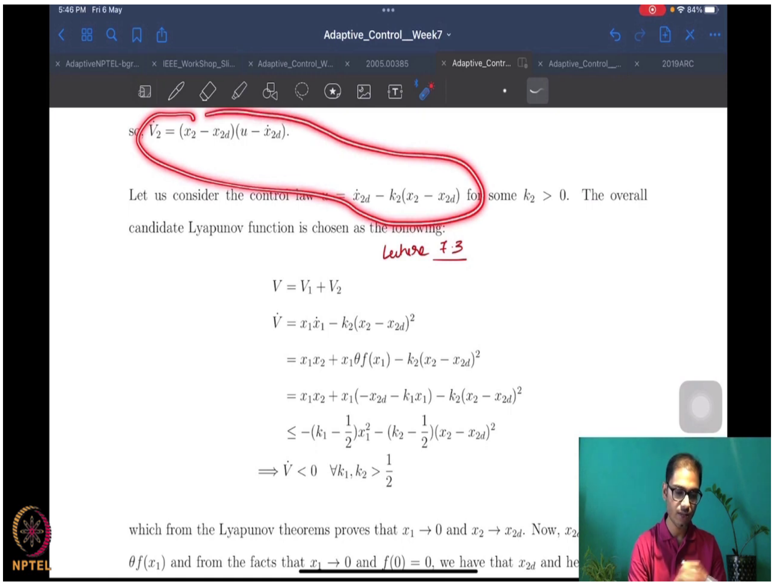


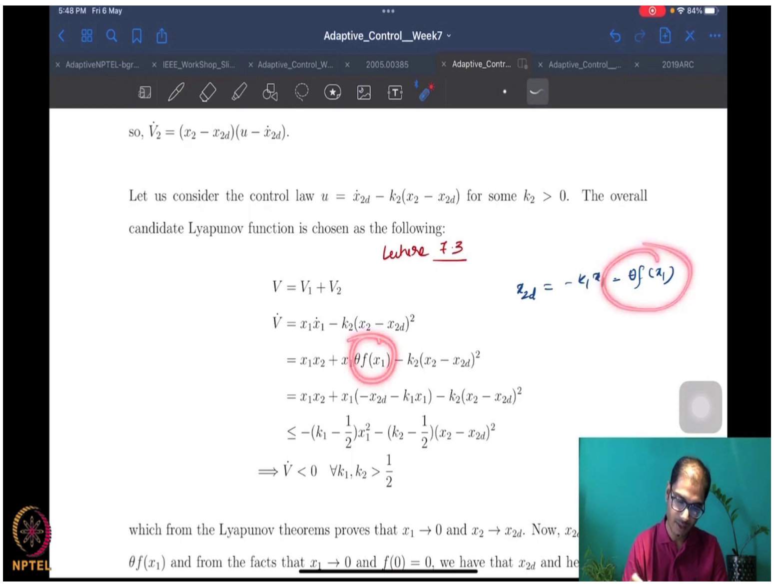


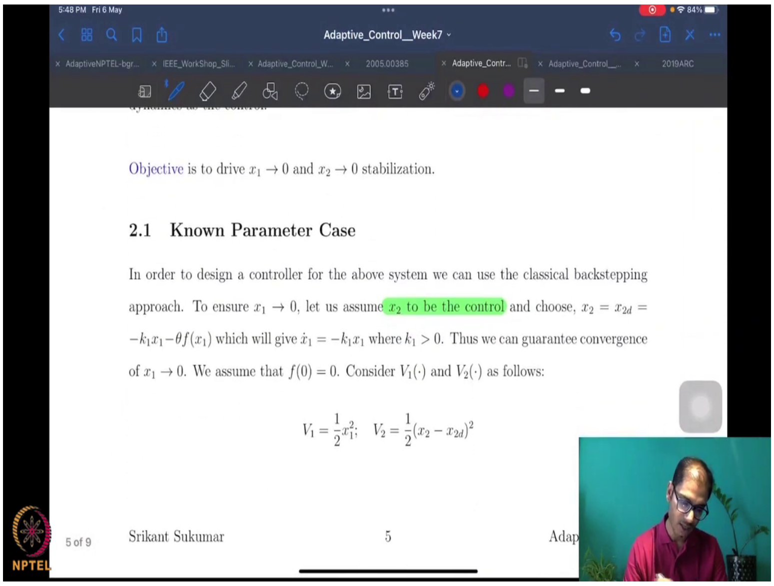
And now we know very well that V 2 dot turns out to be of this structure. And if we choose u to be x2 desired dot minus k2 x2, x2 desired, so then I know that this gets canceled out and I get a nice negative term. So I get basically V2 dot as minus k2, x2 minus x2 desired whole square. So, this is where we were. And we are going to start from here.

(Refer Slide Time: 03:49)


Adaptive_Control_Week7
so, $\dot{V}_2 = (x_2 - x_{2d})(u - \dot{x}_{2d})$.
Let us consider the control law $u = \dot{x}_{2d} - k_2(x_2 - x_{2d})$ for some $k_2 > 0$. The overall candidate Lyapunov function is chosen as the following:
$V = V_1 + V_2$
$\dot{V} = x_1\dot{x}_1 - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1\theta f(x_1) - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1(-x_{2d} - k_1x_1) - k_2(x_2 - x_{2d})^2$
$\leq -(k_1 - \frac{1}{2})x_1^2 - (k_2 - \frac{1}{2})(x_2 - x_{2d})^2$
$\implies \dot{V} < 0 \quad \forall k_1, k_2 > \frac{1}{2}$
which from the Lyapunov theorems proves that $x_1 \to 0$ and $x_2 \to x_{2d}$. Now, x_2
$\theta f(x_1)$ and from the facts that $x_1 \to 0$ and $f(0) = 0$, we have that x_{2d} and he
NPTEL


Adaptive_Control_Week7
so, $\dot{V}_2 = (x_2 - x_{2d})(u - \dot{x}_{2d})$.
Let us consider the control law $u = \dot{x}_{2d} - k_2(x_2 - x_{2d})$ for some $k_2 > 0$. The overall candidate Lyapunov function is chosen as the following:
$V = V_1 + V_2$
$\dot{V} = x_1\dot{x}_1 - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1\theta f(x_1) - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1(-x_{2d} - k_1x_1) - k_2(x_2 - x_{2d})^2$
$\leq -(k_1 - \frac{1}{2})x_1^2 - (k_2 - \frac{1}{2})(x_2 - x_{2d})^2$
$\implies \dot{V} < 0 \quad \forall k_1, k_2 > \frac{1}{2}$
which from the Lyapunov theorems proves that $x_1 \to 0$ and $x_2 \to x_{2d}$. Now, x_2
$\theta f(x_1)$ and from the facts that $x_1 \to 0$ and $f(0) = 0$, we have that x_{2d} and he
NPTEL


Adaptive_Control_Week7
so, $\dot{V}_2 = (x_2 - x_{2d})(u - \dot{x}_{2d})$.
Let us consider the control law $u = \dot{x}_{2d} - k_2(x_2 - x_{2d})$ for some $k_2 > 0$. The overall candidate Lyapunov function is chosen as the following:
$x_{2d} = -k_1x_1 - \theta f(x_1)$
$V = V_1 + V_2$
$\dot{V} = x_1\dot{x}_1 - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1\theta f(x_1) - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1(-x_{2d} - k_1x_1) - k_2(x_2 - x_{2d})^2$
$\leq -(k_1 - \frac{1}{2})x_1^2 - (k_2 - \frac{1}{2})(x_2 - x_{2d})^2$
$\implies \dot{V} < 0 \quad \forall k_1, k_2 > \frac{1}{2}$
which from the Lyapunov theorems proves that $x_1 \to 0$ and $x_2 \to x_{2d}$. Now, x_2
$\theta f(x_1)$ and from the facts that $x_1 \to 0$ and $f(0) = 0$, we have that x_{2d} and he
NPTEL



So, I am going to mark our lecture as lecture 7.3, starting this point. So as always, just like before, we did not backstepping, we just add up the 2 pieces to get a candidate Lyapunov function for the consolidated x1 x2 system. And we actually start taking the derivatives carefully. So the x1 V1 was just x1 squared by 2. So V1 dot is x1 x1 dot. And V2 dot by virtue of this analysis. It is just minus k2 x2 minus x2 desired squared.

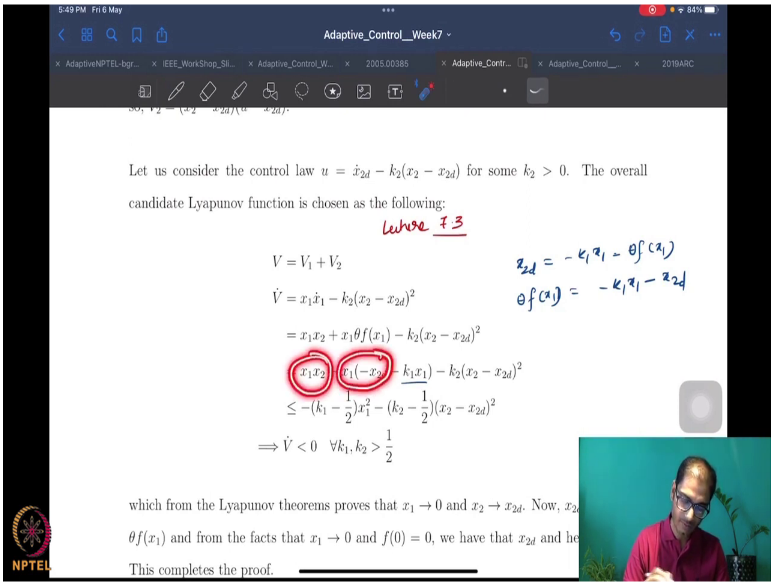
Now we plug in for x1 dot of course. So what does it turn out to be? x1 dot is x2 plus theta of x1. So I get x1 x2 plus x1 theta of x1. And now we also know that, of course, we want to write x2 in terms of this variable. And that is what we do. So that is what we do. So what is this? So basically, I have I mean, we just write, let me look at this carefully if this is done correctly or not.

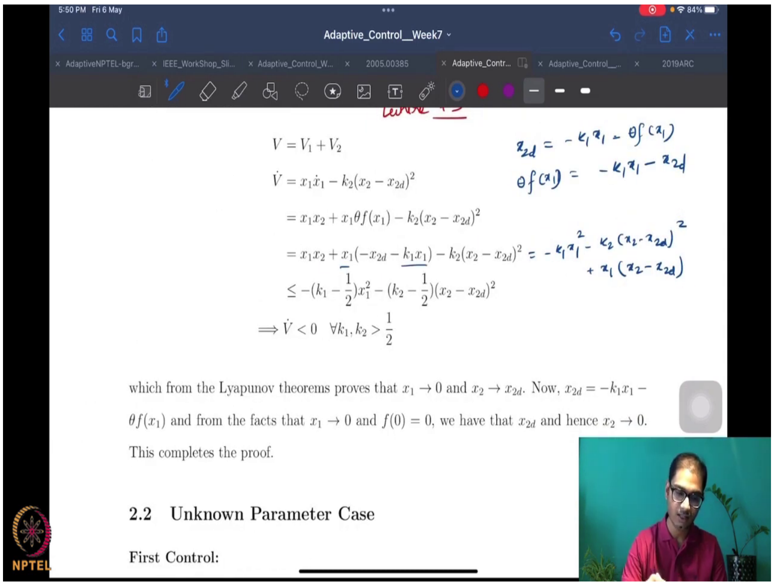
So, if we go back and look, in fact, I am going to reproduce the expression right here. So x2 desired was defined to be minus k1 x1, plus, I believe minus theta f x1. So in fact, the way this manipulation is done, it is fine, that is okay.

So, so what we do is, if I actually do not write x2 as this, but I simply replace theta f x1 from here. So if I do that, theta, f x1 is actually equal to minus k1 x1 plus x2 desired. So this will actually be a plus sign. This will actually be a plus sign. Let us see, if I got this correct. So, just a second. Let me go back and check.

So, x2 result is minus k1 x1 minus theta of x1. And if I go here, which is what I wrote here, x2 desired is minus k1 x1 minus theta times f x1. So if I replace for theta times f x1 from here, I get minus k1 x1. I made a mistake. This should be minus k1 x1 minus, minus x2 desired. This is what I was sort of looking for. And this is correct. And this is correct. That is what I was wondering. So now, if you combine, so this term is of course nice. It is minus k1 x1 squared.

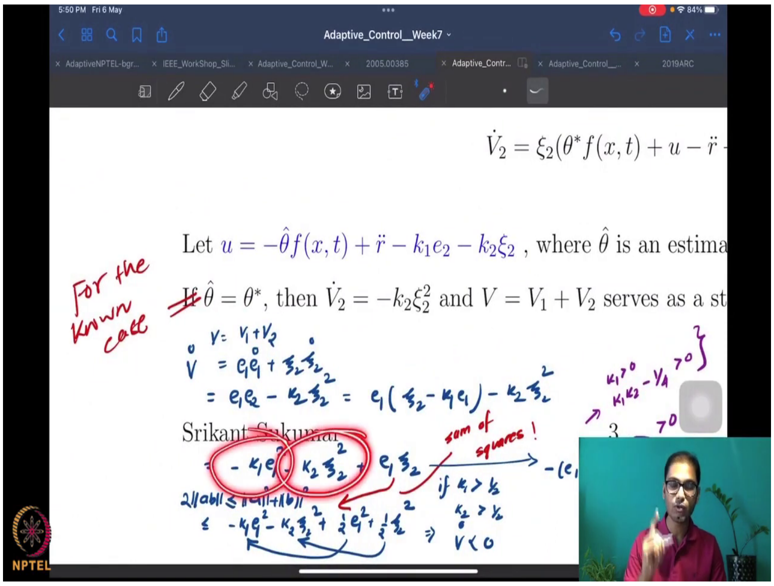
(Refer Slide Time: 07:15)

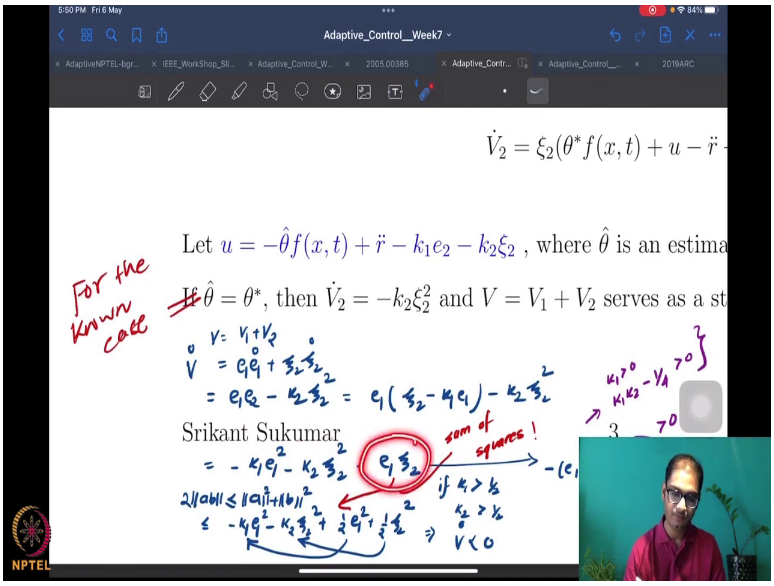




And if I combine these 2 terms, I get x1 times x2 minus x2 desired. So, essentially, there is a step that is missing here. So I am going to write it. So this is actually equal to minus k1 x1 squared minus k2 x2 minus x2 desired squared, plus x1 times x2 minus x2 desired. This is very similar to what we had in the previous backstepping also, this is very similar, because you have a nice 2, you have nice 2 negative terms and then 1 mixed term, this is very standard, what happens in backstepping data analysis.

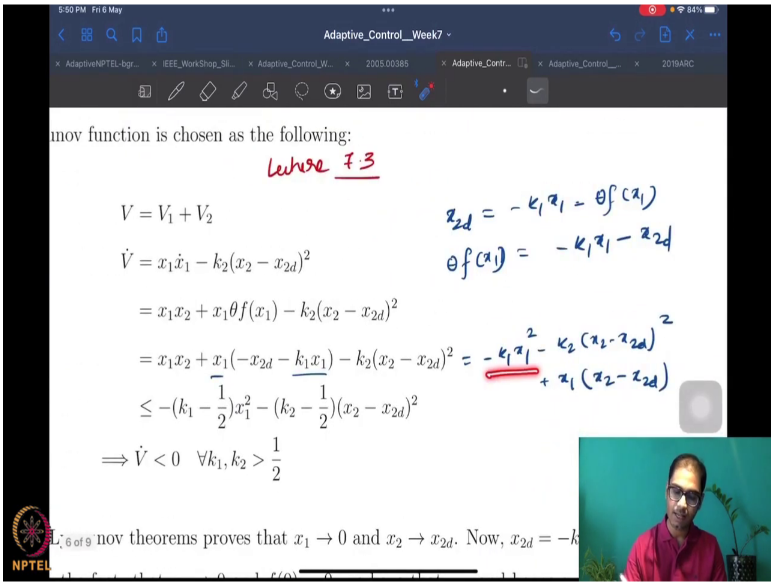
(Refer Slide Time: 08:08)

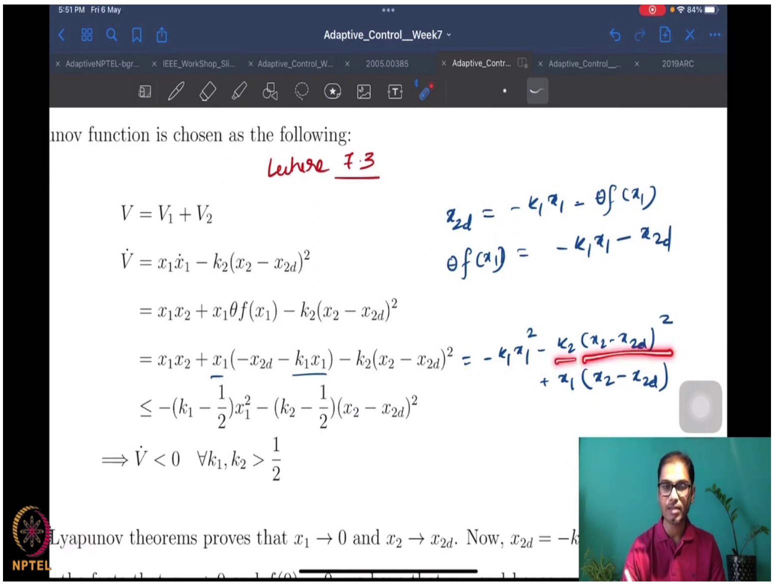


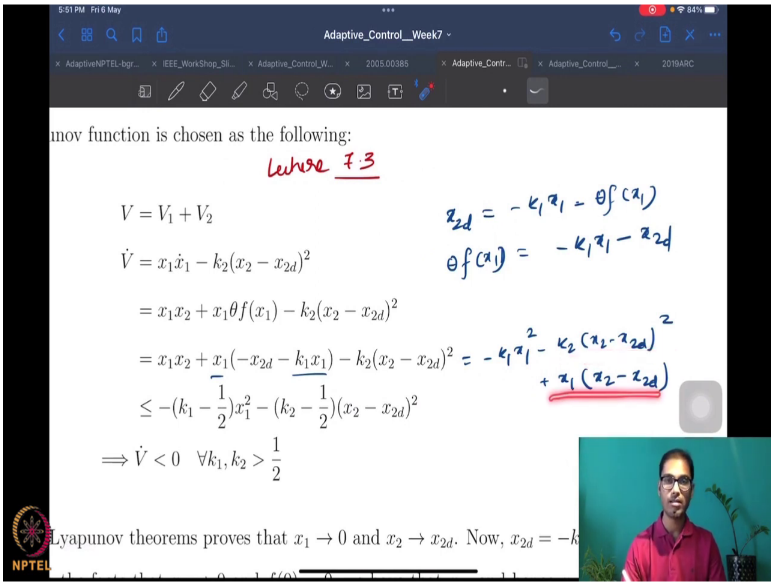


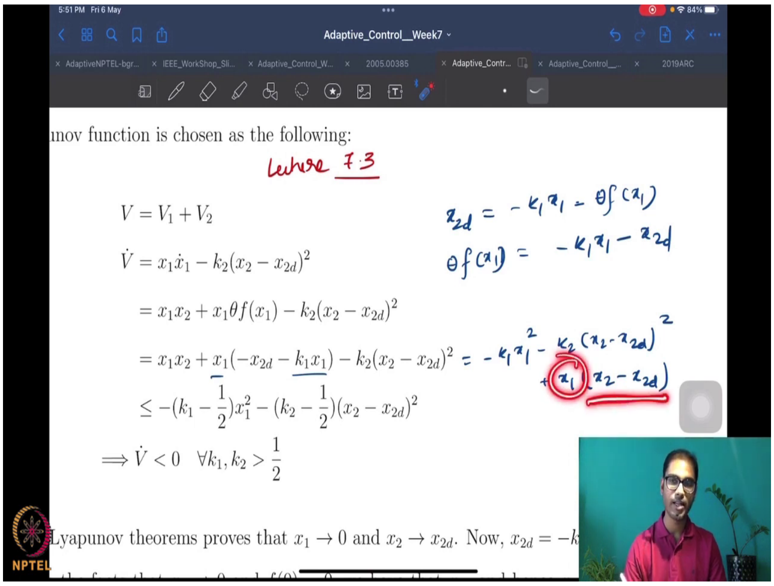
If you go back, I am going to try to actually demonstrate that similarity. If you see, I had minus k1 e1 squared minus k2 psi2 squared, which is the backstepping error variable. And then I had a mixed term in the e1 and a backstepping error variable.

(Refer Slide Time: 08:27)



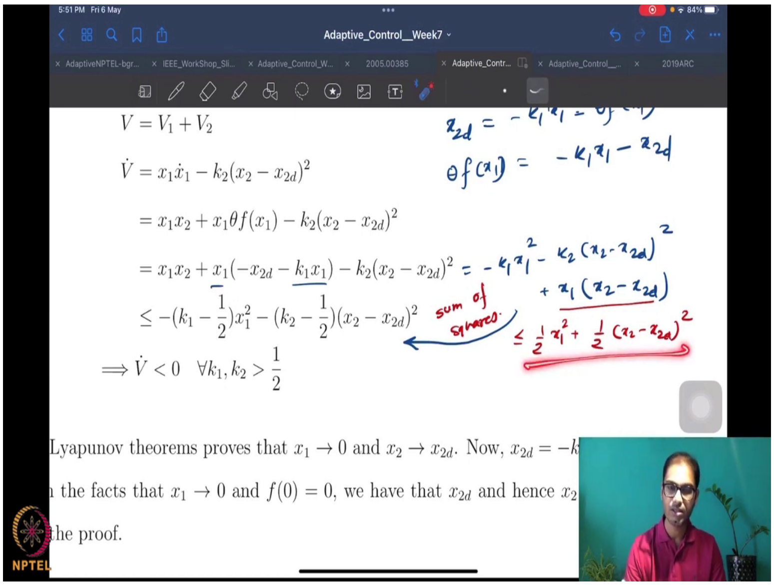




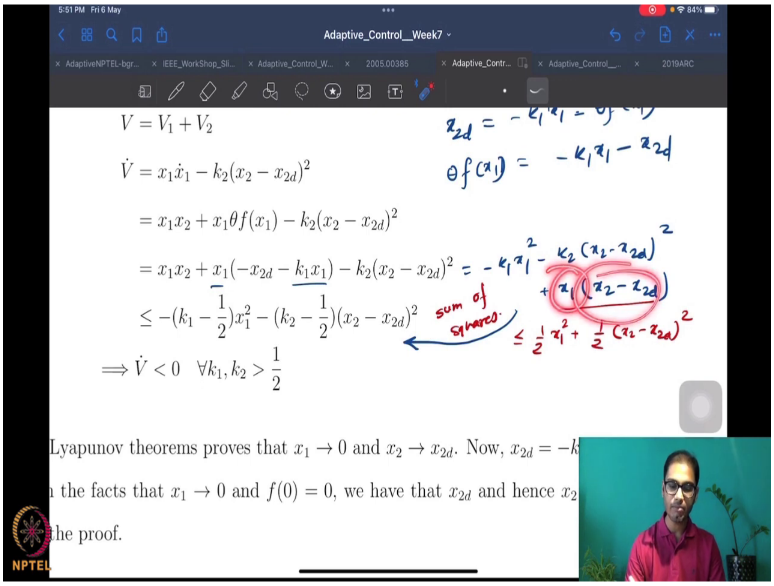


So if you look at this, I have the exact same scenario here, I have one term in minus k1 x1 squared, another term in the backstepping error variable, quadratic, and third, mixed term in the x1 and the backstepping error variable. So, this is exactly what you tend to get.

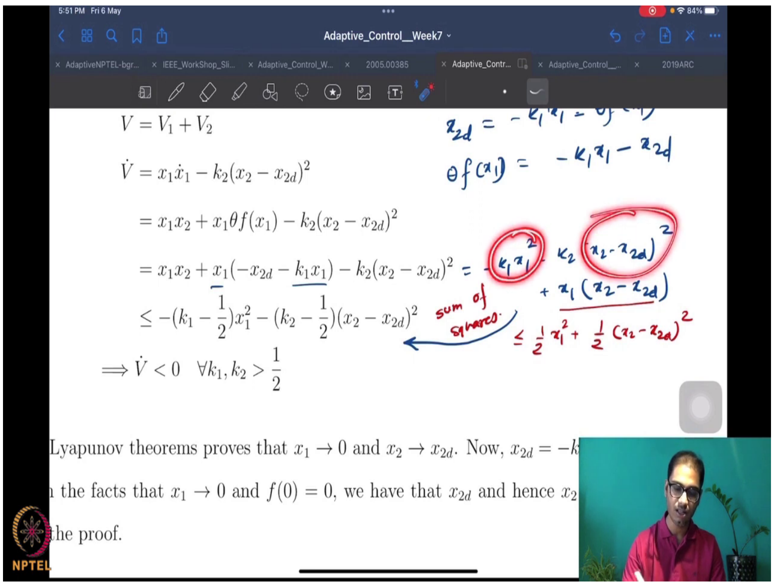
(Refer Slide Time: 08:44)



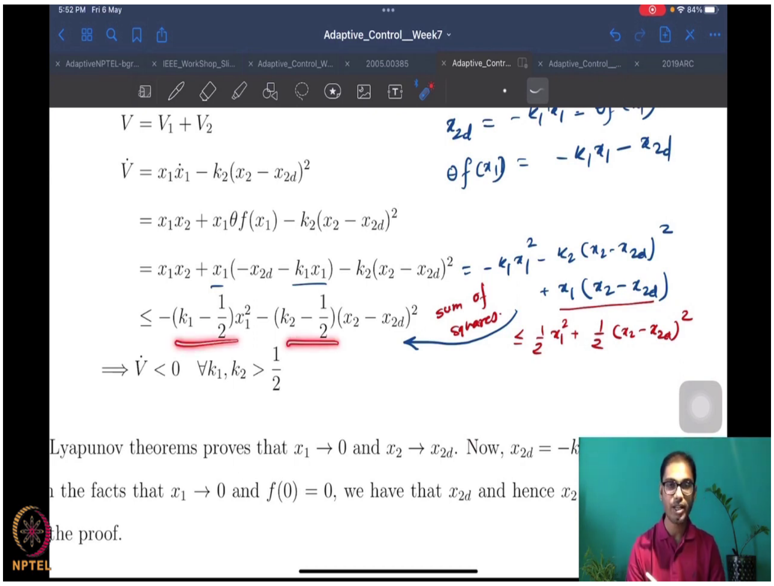
Adaptive_Control_Week7
$V = V_1 + V_2$
$\dot{V} = x_1\dot{x}_1 - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1\theta f(x_1) - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1(-x_{2d} - k_1x_1) - k_2(x_2 - x_{2d})^2$
$\leq -(k_1 - \frac{1}{2})x_1^2 - (k_2 - \frac{1}{2})(x_2 - x_{2d})^2$
$\implies \dot{V} < 0 \quad \forall k_1, k_2 > \frac{1}{2}$
sum of squares
Lyapunov theorems proves that $x_1 \to 0$ and $x_2 \to x_{2d}$. Now, $x_{2d} = -k$
the facts that $x_1 \to 0$ and $f(0) = 0$, we have that x_{2d} and hence x_2
he proof.
NPTEL

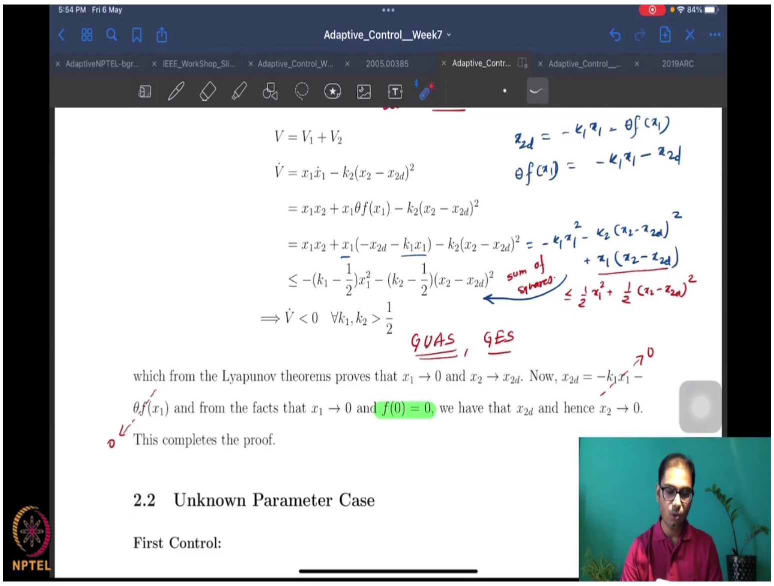


Adaptive_Control_Week7
$V = V_1 + V_2$
$\dot{V} = x_1\dot{x}_1 - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1\theta f(x_1) - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1(-x_{2d} - k_1x_1) - k_2(x_2 - x_{2d})^2$
$\leq -(k_1 - \frac{1}{2})x_1^2 - (k_2 - \frac{1}{2})(x_2 - x_{2d})^2$
$\implies \dot{V} < 0 \quad \forall k_1, k_2 > \frac{1}{2}$
sum of squares
Lyapunov theorems proves that $x_1 \to 0$ and $x_2 \to x_{2d}$. Now, $x_{2d} = -k$
the facts that $x_1 \to 0$ and $f(0) = 0$, we have that x_{2d} and hence x_2
he proof.
NPTEL



Adaptive_Control_Week7
$V = V_1 + V_2$
$\dot{V} = x_1\dot{x}_1 - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1\theta f(x_1) - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1(-x_{2d} - k_1x_1) - k_2(x_2 - x_{2d})^2$
$\leq -(k_1 - \frac{1}{2})x_1^2 - (k_2 - \frac{1}{2})(x_2 - x_{2d})^2$
$\implies \dot{V} < 0 \quad \forall k_1, k_2 > \frac{1}{2}$
sum of squares
Lyapunov theorems proves that $x_1 \to 0$ and $x_2 \to x_{2d}$. Now, $x_{2d} = -k$
the facts that $x_1 \to 0$ and $f(0) = 0$, we have that x_{2d} and hence x_2
he proof.
NPTEL





And from this point, you know that I can do, I can simply do what is called sum of squares. which just involves using the inequality that this guy, I will write it once more, is less than equal to half x1 square plus half x2 minus x2 desired square. So, once I use this inequality here, which is essentially the sum of squared process. Why is it called the sum of squared process? Because we are writing these mixed terms, a, b type terms in terms of squares, a squared and b squared, because those are the terms we have here.

And so once you combine these, you get exactly this expression, which is again very similar to what you had even in the matched case. So, with this backstepping type procedure, you are getting almost rather similar V dot as the matched case also. So, this is why backstepping is a rather universal, method of control design that can be use in many different contexts.

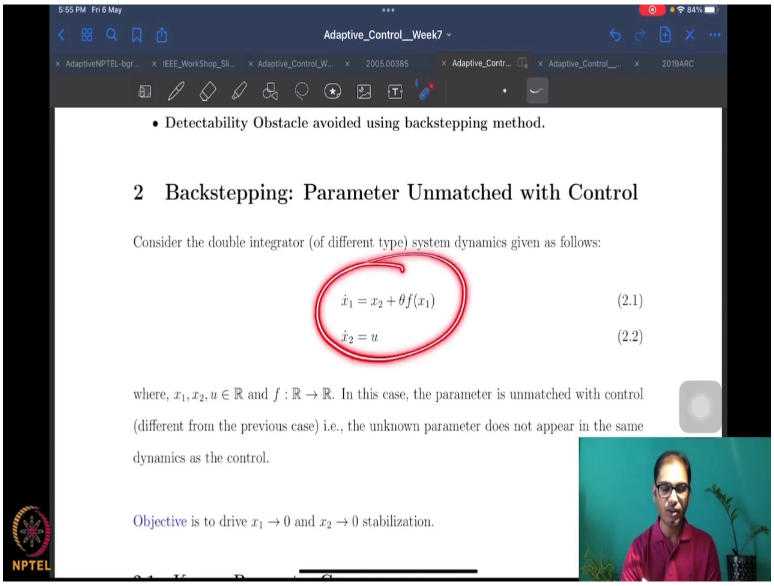
So, with this that if I choose k1 and k2 to be greater than half, then I have V dot negative definite, then of course, which immediately helps me to apply this Lyapunov theorem, which immediately proves that x1 goes to 0 and x2 goes to x2 desired. So, essentially, you have an all round, global uniform asymptotic stability and you have all essentially the best property that you can look for.

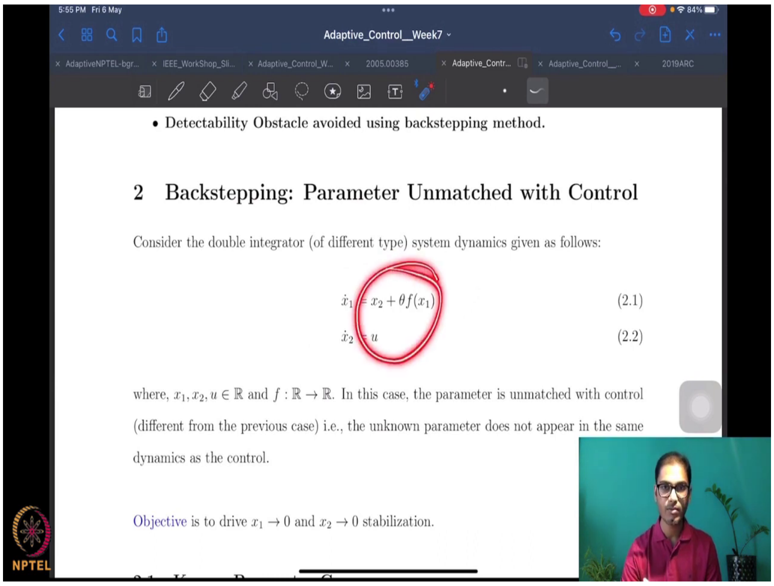
So, in fact, if you look at it very carefully, you can even conclude global exponential stability, if you so desired. You can even conclude global exponential stability, because you can see that it is the same model and all that. But, since we are dealing with nonlinear systems, mostly, we do not really mention this because more often than not, this is not possible to achieve.

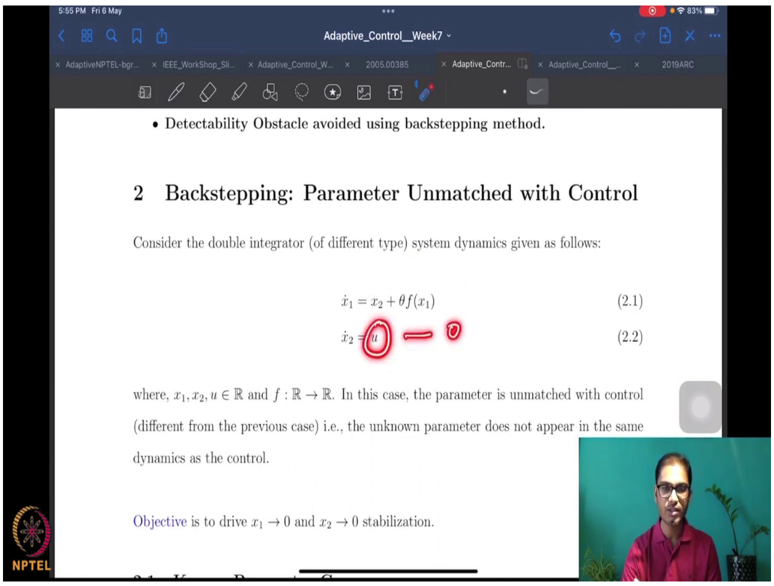
So now, what do we know? We have started off looking for x1 to go to 0 and x2 to also go to 0. So this always remains a question. Did the construction of the backstepping error variable, actually mess with our tracking or the stabilization control objective? In this case, also, the answer is no under certain assumptions. So what are those? So x1 goes to 0 is obvious, x2 goes to x2 desired, and x2 desired was this quantity. Now, what do I know? I know that this is going to 0.

So, if x2 desired is going to 0. So, if I want x2 desired to go to 0, because I want x2 to go to 0, so this will happen only if x2 desired is going to 0. So I want this to happen. And, so we sort of make this sort of an assumption. I mean, you can think of it as an assumption. This is an assumption, that because eventually, this is something beyond the system data. And this is an assumption. And if we do have such an assumption that f at 0 state is 0, then you have that x2 goes to 0. And we are done. Now, one might be tempted to argue about the reasonableness of this assumption.

(Refer Slide Time: 12:30)

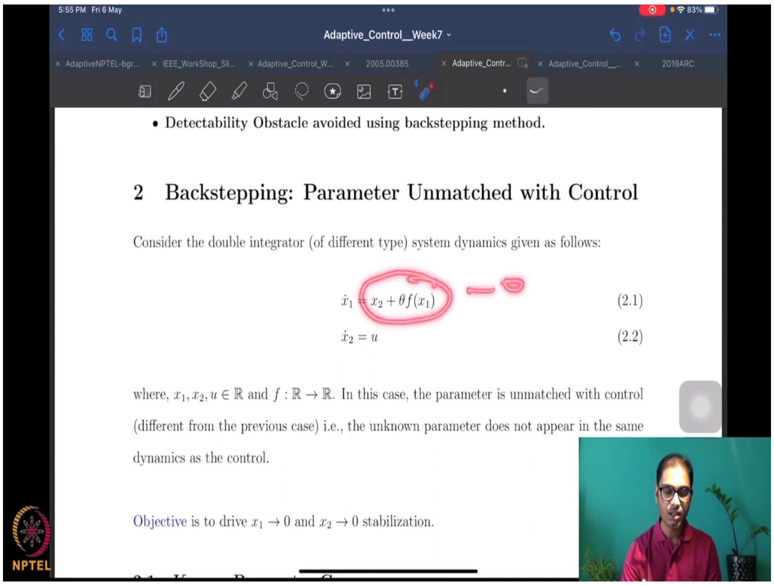

Adaptive_Control_Week7
• Detectability Obstacle avoided using backstepping method.
2 Backstepping: Parameter Unmatched with Control
Consider the double integrator (of different type) system dynamics given as follows:
$\dot{x}_1 = x_2 + \theta f(x_1)$ (2.1)
$\dot{x}_2 = u$ (2.2)
where, $x_1, x_2, u \in \mathbb{R}$ 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.
NPTEL

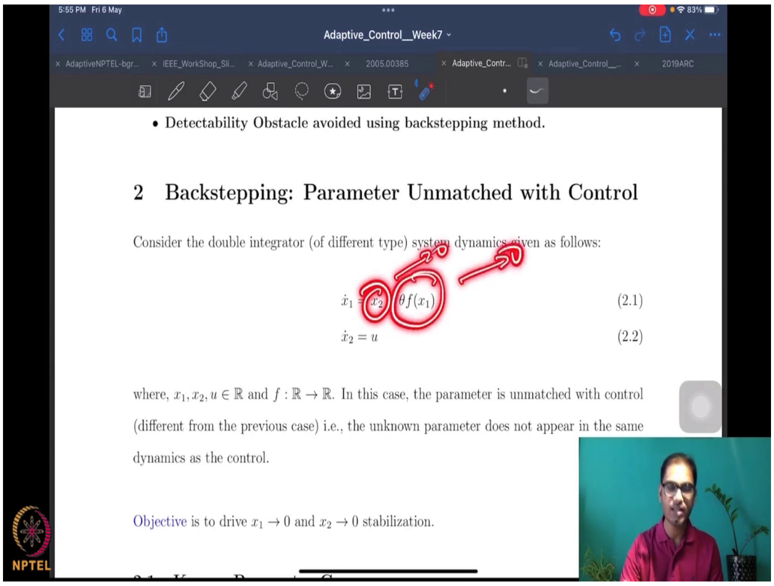

Adaptive_Control_Week7
• Detectability Obstacle avoided using backstepping method.
2 Backstepping: Parameter Unmatched with Control
Consider the double integrator (of different type) system dynamics given as follows:
$\dot{x}_1 = x_2 + \theta f(x_1)$ (2.1)
$\dot{x}_2 = u$ (2.2)
where, $x_1, x_2, u \in \mathbb{R}$ 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.
NPTEL


Adaptive_Control_Week7
• Detectability Obstacle avoided using backstepping method.
2 Backstepping: Parameter Unmatched with Control
Consider the double integrator (of different type) system dynamics given as follows:
$\dot{x}_1 = x_2 + \theta f(x_1)$ (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.
NPTEL

Now, why one would say that this assumption is in fact reasonable but at least reasonable? Is that if you have if you are looking for equilibrium, if you are looking for x1 and x2 equal to 0 to be the equilibrium of the system. So how do you find equilibrium, typically, you will assign the control to be 0. And you will try to find the right hand, where the right hand side is 0. So with the control 0, this is any way 0.

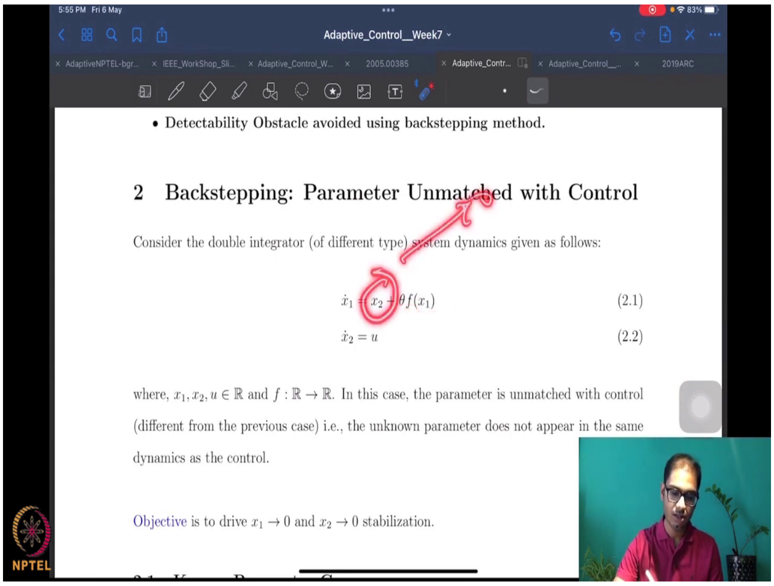
(Refer Slide Time: 13:02)

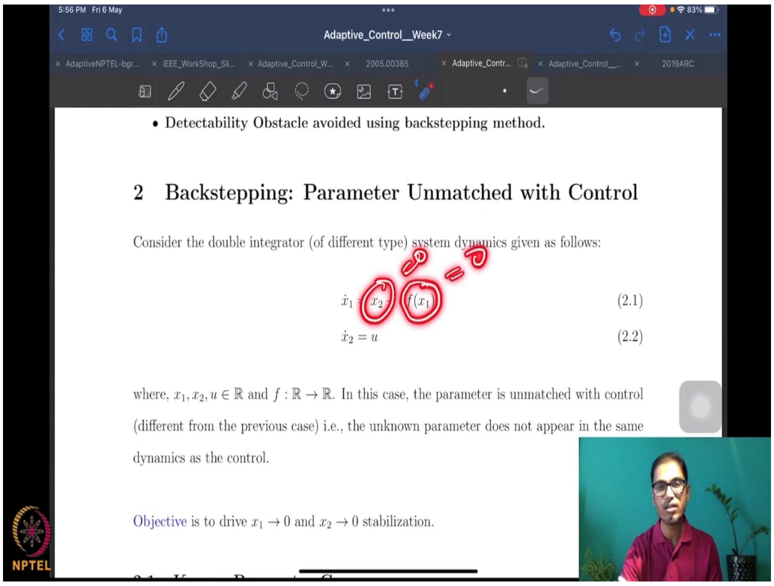


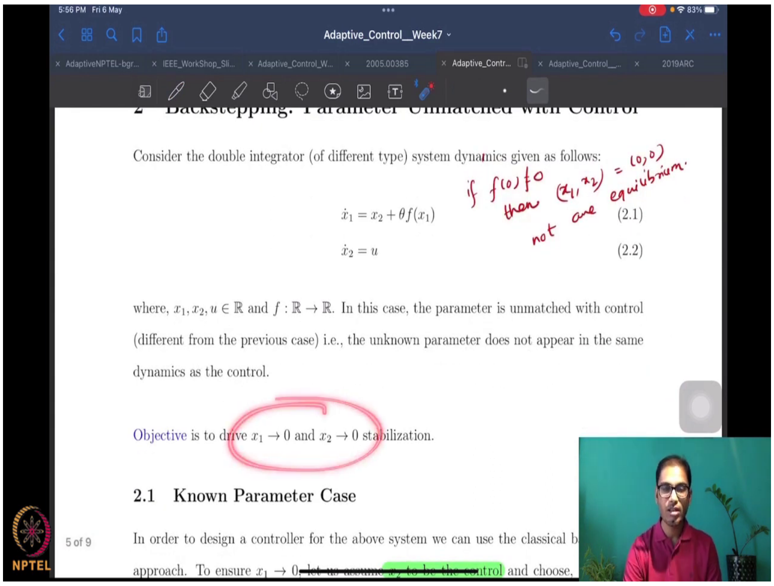


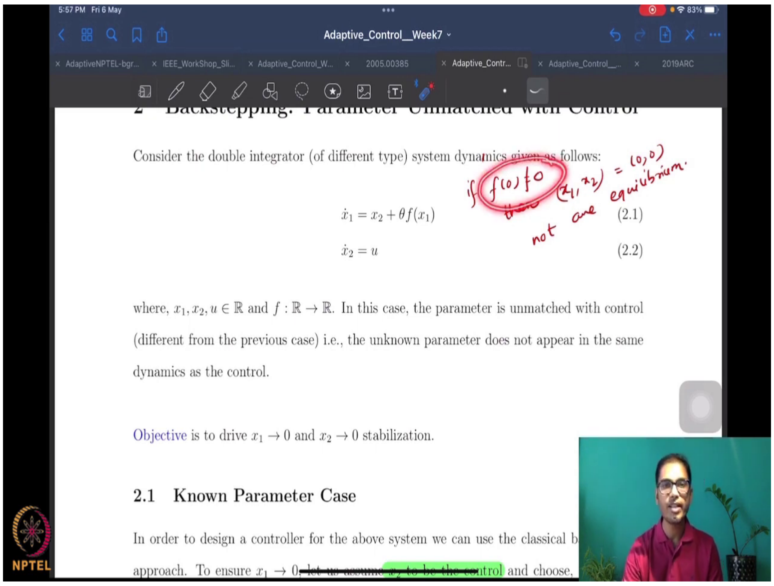
But if I want this guy to be 0, and I want x2 to also be 0, because I want x1 x2 equal to origin that is 0 0 to be an equilibrium of the system, then if this guy is 0, then this guy also has to be 0. And theta is some of course, constant non positive, sorry constant, non zero, unknown and so, constant and non zero. If it is 0, then there is nothing to do here.

(Refer Slide Time: 13:26)





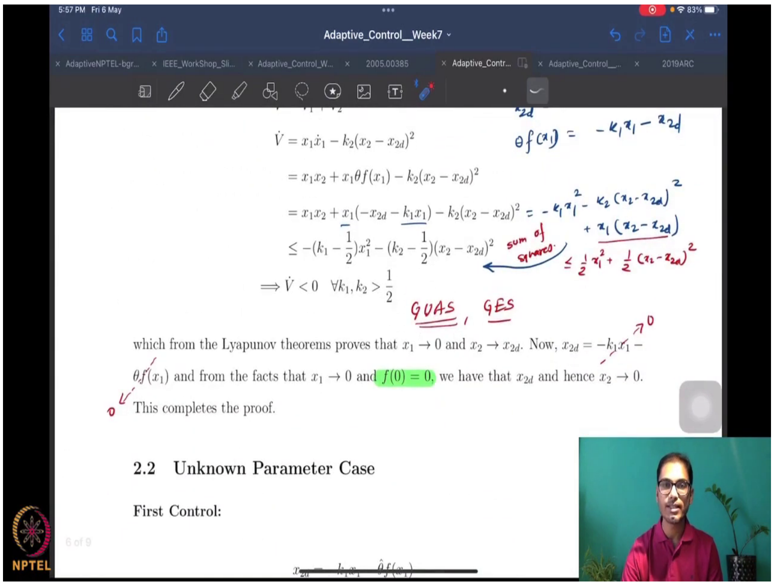


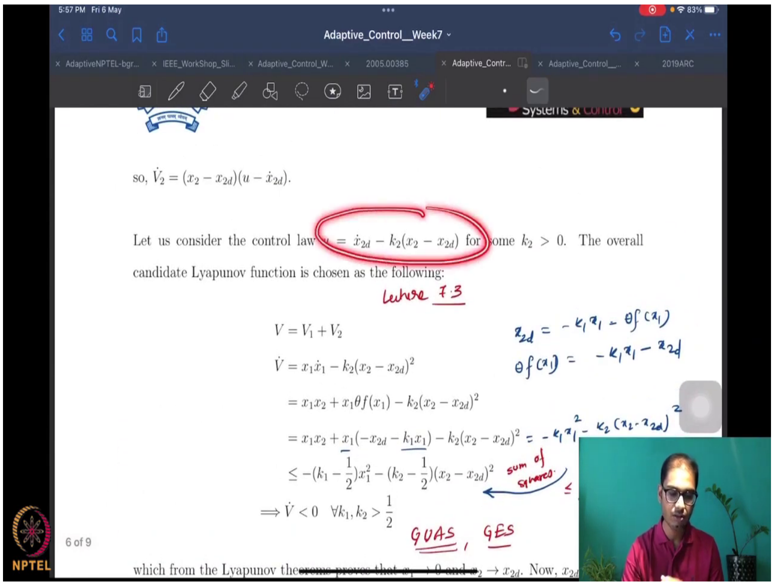


So if x2 is 0, or x2 is, x2 equal to 0 is an equilibrium. And then this also has to be 0 at x1 equal to 0. And otherwise x1 equal to 0 is not an equilibrium solution. If f x, if f 0 is not equal to 0, and so if, so the point is, that you are trying to make is, is f 0 equal to, not equal to 0, then x1, x2 0, 0, not an equilibrium.

So, and remember that we are always trying to analyze stability of an equilibrium, it does not make any sense. If you are trying to analyze the stability of a point, which is not the equilibrium. So therefore, if we are interested in going to a point 0 0, then it has to be, it better be an equilibrium of the system. And, so f 0 equal to 0 is a very reasonable assumption to ensure that x1 x2 equal to 0 0 is in fact an equilibrium system.

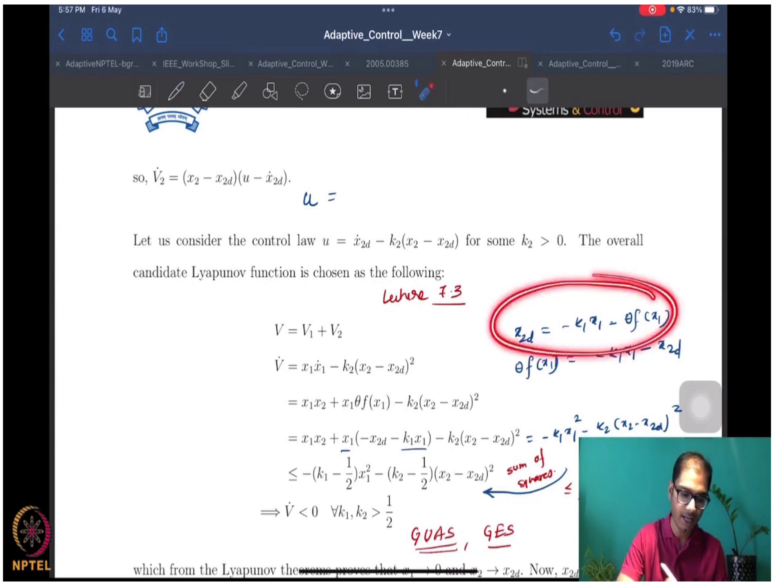
(Refer Slide Time: 14:42)

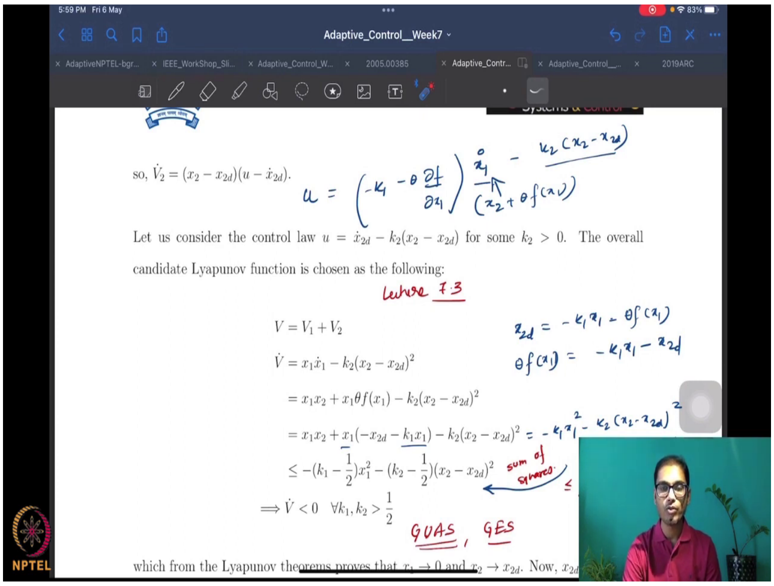




So, this is a fairly reasonable assumption, no problem. And, so we have been able to prove for the known case, this kind of control law. So, now one of the question that also arises is. What does this control law look like? Because there is an x2 d dot. So, you essentially have to take a derivative here.

(Refer Slide Time: 15:03)

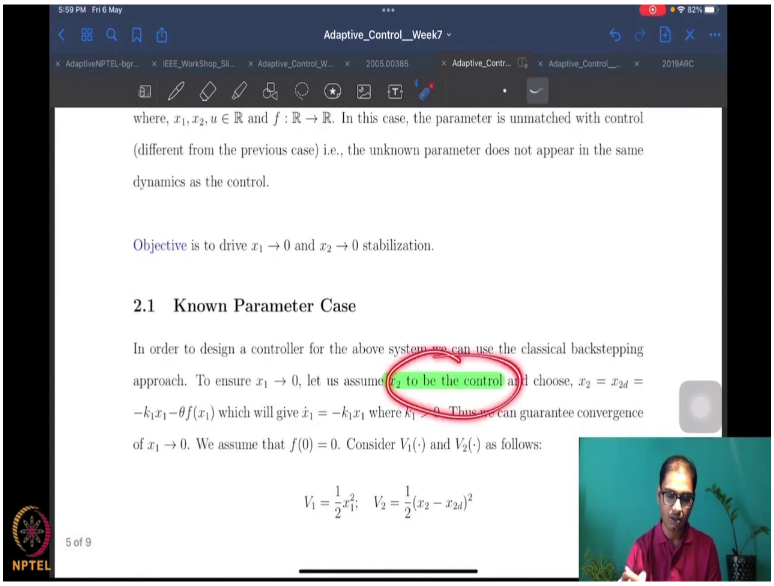


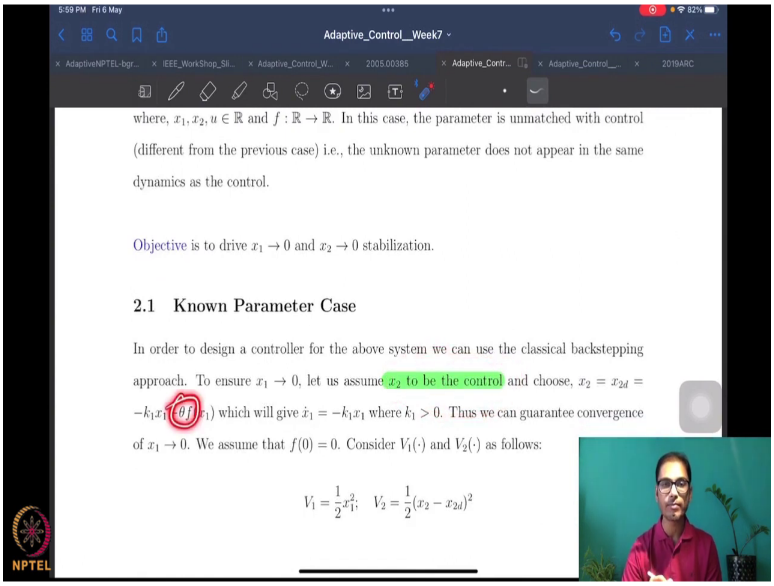


So it's , use this expression take a derivative. So, this is minus k1 x1 dot, which is minus k1 x2 and this is minus theta f dot, which is partial of f with respect to x1, x1 dot, sorry, I should just say, so there is x1 in both. So, this is like minus k1 minus theta partial of f with respect to x1 times x1 dot. So, this is essentially the control and if you want to substitute this you can of course substitute for this x1 dot as x2 plus theta f x1. So, this is a very implementable control. So, this is minus of course, k2 x2 minus x2 desired. So, that is, that is of course there, so x1 dot is just this. So, this is a very implementable controller, of course, no problem. It is just a function of the states, no issues. But of course, you notice that it contains the parameter, which is again something that you expect, in fact, x2 desired also contains the parameter, so the parameter appears here.

So the parameter is, it is complicated the parameter appears here, the parameter appears here. And the parameter also appears here. So, unlike the matched case, where the parameter appeared in only once place here the parameter seems to appear in a lot of different locations, and this is what you will see subsequently complicates the adaptive controller design. So, we will sort of start going into the adaptive control design for this same system.

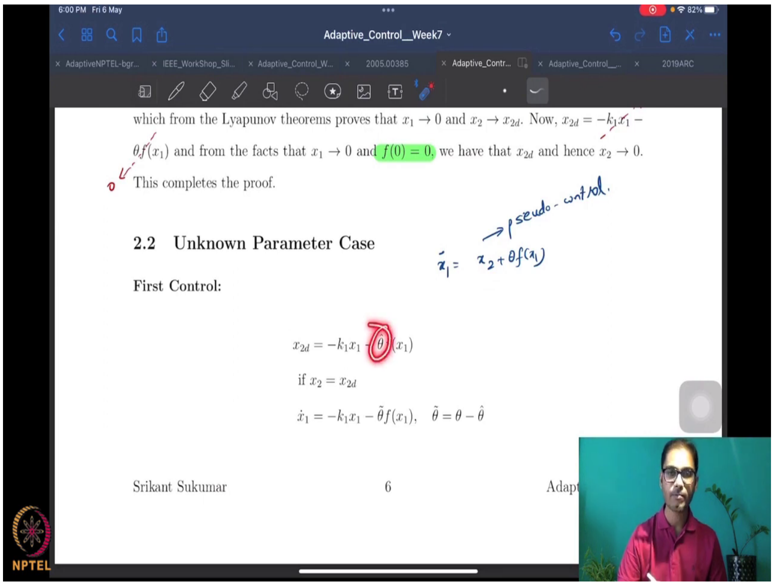
(Refer Slide Time: 17:03)

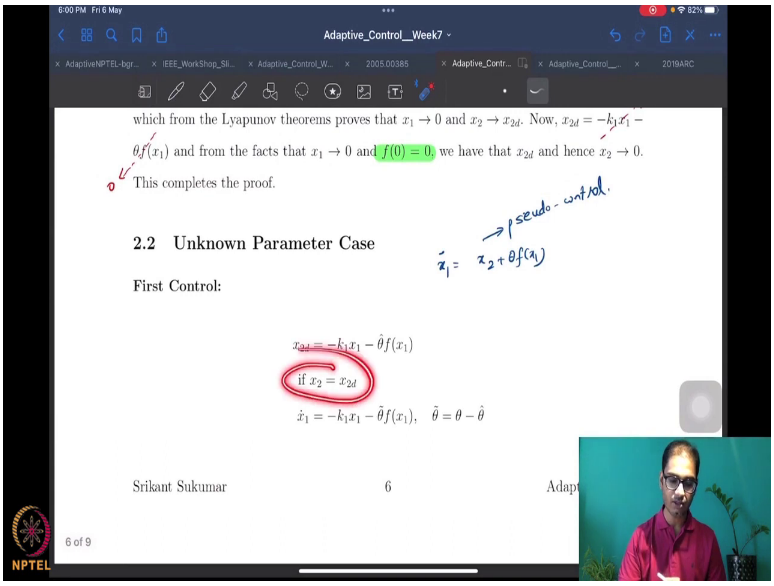


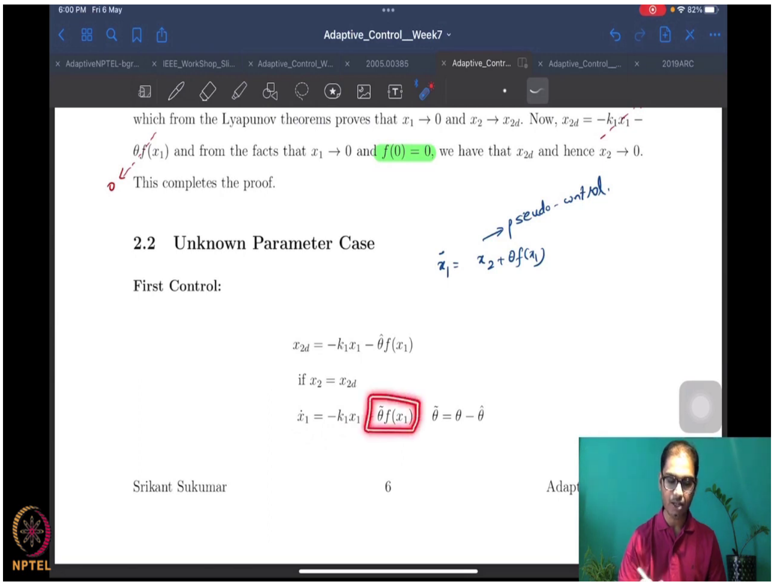


So, what happens in the unknown parameter case? So, of course, in the unknown parameter case, as you can see, I cannot have an x2 desired because I think of x2 as a control. So I cannot have x2 desired with the theta. Because, well, I do not know theta. So there is no question of implementing a theta.

(Refer Slide Time: 17:25)



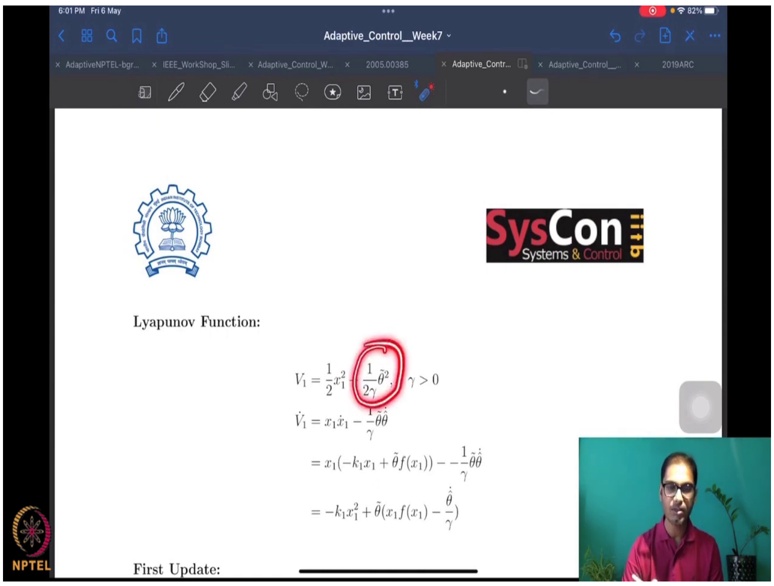


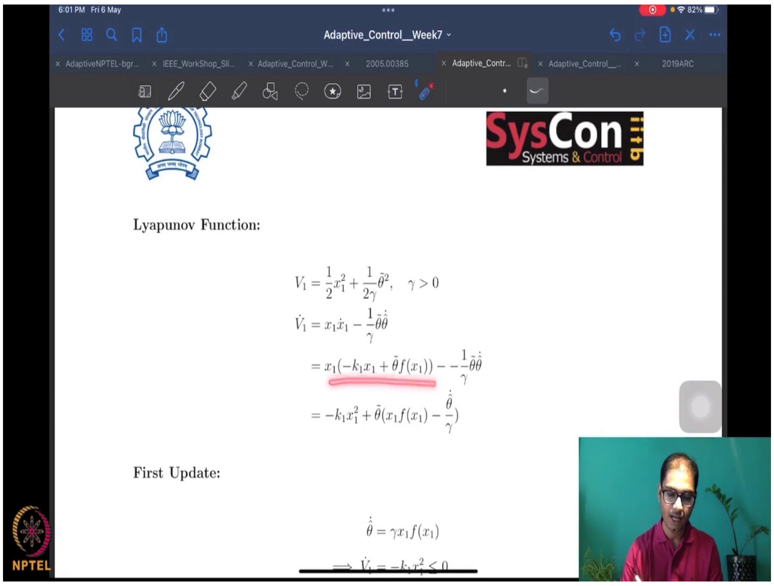


So therefore, the first step is, again, standard certainty equivalence. Because we are just looking at the first system. So the first system is x1 dot is x2 plus theta f x1 with this as some what one would call pseudo control. So, x2 desired is with a theta hat, which is basically an estimate of theta with f x1 and then of course, you have the nice negative term.

So, when f when x2 is in fact x2 desired, you will get x1 dot is minus k1 x1, k1 x1, but and now within additional theta tilde data. So this is unusual, you already start getting a theta tilde here, in the first piece itself. And this is where things start getting complicated. And now what do we do? Earlier, we took V1 as one half x1 squared, but now that is not sufficient. Because now I have a new state.

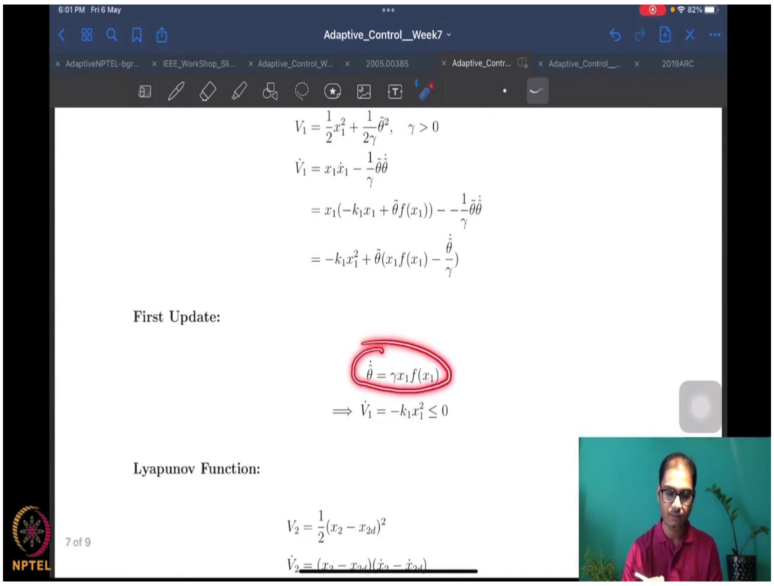
(Refer Slide Time: 18:29)

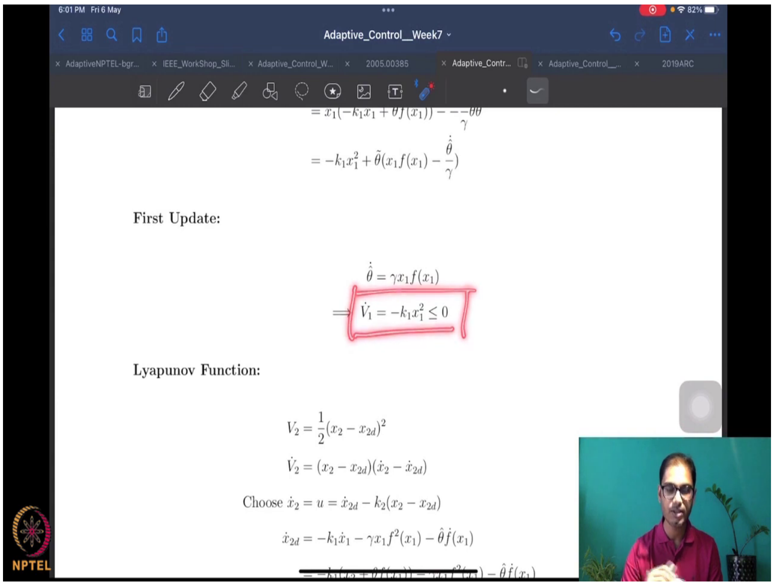


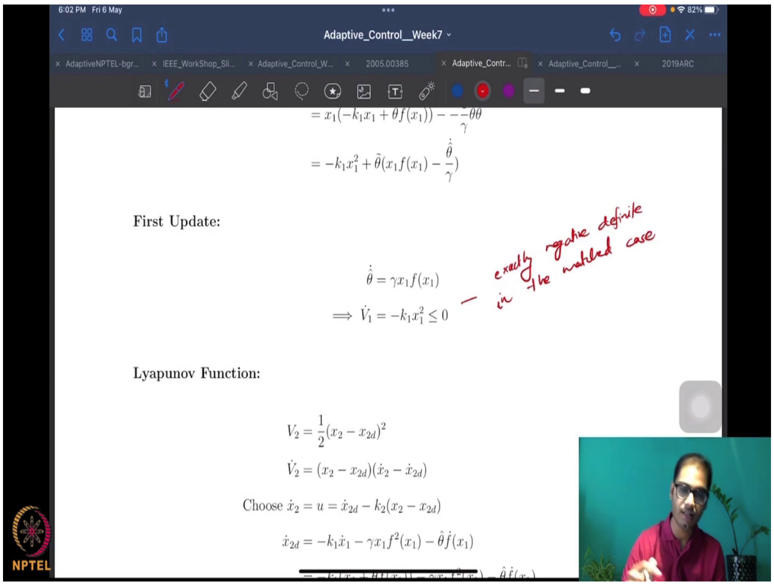


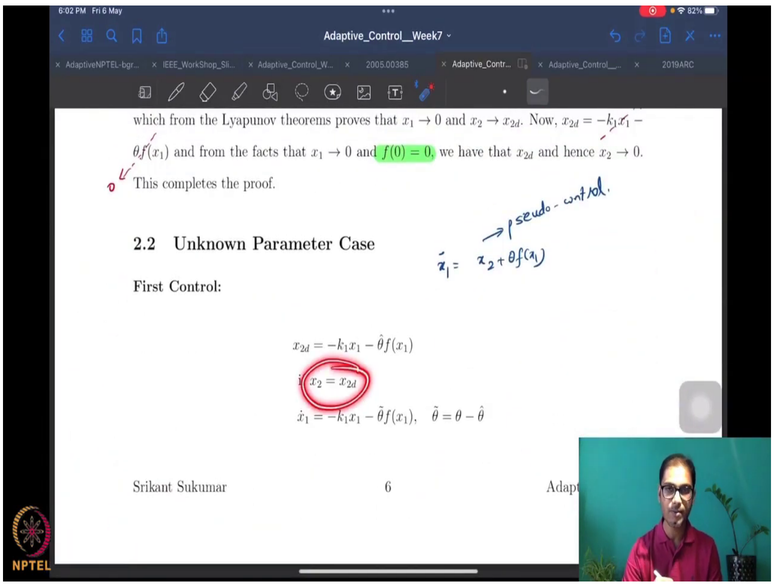
So, what do I do? I do what I do standard, typically in adaptive control, I add a quadratic term corresponding to the parameter error. And now I use that to construct an update law. So what do I get, I get here x1 x1 dot and one over gamma theta tilde, theta hat dot. And now I plug in for x1 dot, which is this and then retain this term as it is.

(Refer Slide Time: 18:56)




Adaptive_Control_Week7
$= -k_1x_1^2 + \tilde{\theta}(x_1 f(x_1) - \frac{\dot{\hat{\theta}}}{\gamma})$
First Update:
$\dot{\hat{\theta}} = \gamma x_1 f(x_1)$
$\implies \dot{V}_1 = -k_1x_1^2 \leq 0$
Lyapunov Function:
$V_2 = \frac{1}{2}(x_2 - x_{2d})^2$
$\dot{V}_2 = (x_2 - x_{2d})(\dot{x}_2 - \dot{x}_{2d})$
Choose $\dot{x}_2 = u = \dot{x}_{2d} - k_2(x_2 - x_{2d})$
$\dot{x}_{2d} = -k_1\dot{x}_1 - \gamma x_1 f^2(x_1) - \hat{\theta}\dot{f}(x_1)$
NPTEL


Adaptive_Control_Week7
$= -k_1x_1^2 + \tilde{\theta}(x_1 f(x_1) - \frac{\dot{\hat{\theta}}}{\gamma})$
First Update:
exactly negative definite in the matched case
$\dot{\hat{\theta}} = \gamma x_1 f(x_1)$
$\implies \dot{V}_1 = -k_1x_1^2 \leq 0$
Lyapunov Function:
$V_2 = \frac{1}{2}(x_2 - x_{2d})^2$
$\dot{V}_2 = (x_2 - x_{2d})(\dot{x}_2 - \dot{x}_{2d})$
Choose $\dot{x}_2 = u = \dot{x}_{2d} - k_2(x_2 - x_{2d})$
$\dot{x}_{2d} = -k_1\dot{x}_1 - \gamma x_1 f^2(x_1) - \hat{\theta}\dot{f}(x_1)$
NPTEL

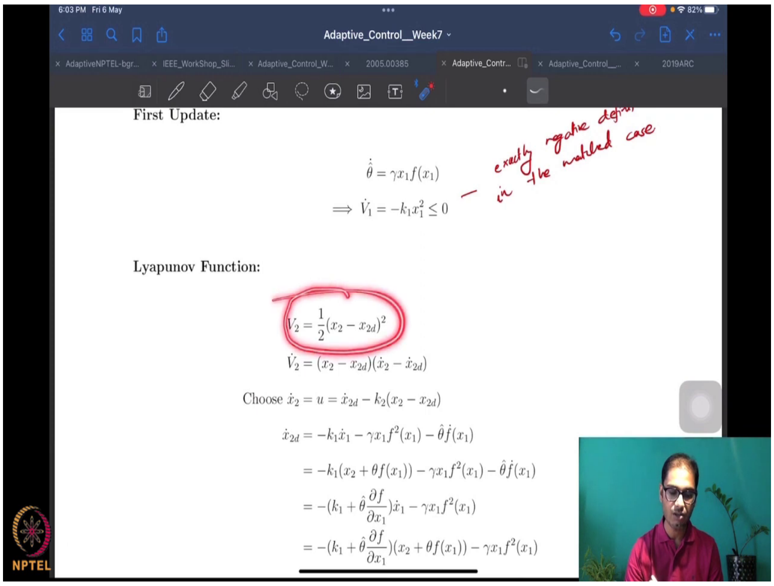

Adaptive_Control_Week7
which from the Lyapunov theorems proves that $x_1 \to 0$ and $x_2 \to x_{2d}$. Now, $x_{2d} = -k_1x_1 - \theta f(x_1)$ and from the facts that $x_1 \to 0$ and $f(0) = 0$, we have that x_{2d} and hence $x_2 \to 0$.
This completes the proof.
2.2 Unknown Parameter Case
pseudo-control.
$\dot{x}_1 = x_2 + \theta f(x_1)$
First Control:
$x_{2d} = -k_1x_1 - \hat{\theta}f(x_1)$
$x_2 = x_{2d}$
$\dot{x}_1 = -k_1x_1 - \tilde{\theta}f(x_1), \quad \tilde{\theta} = \theta - \hat{\theta}$
Srikant Sukumar 6
NPTEL

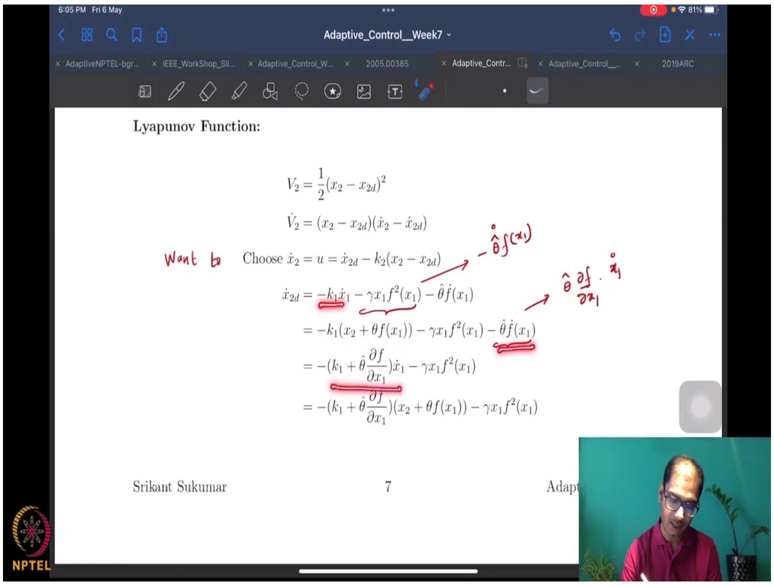
Now, I know that this theta tilde terms can be combined nicely. And if I choose theta hat dot to be this which is essentially cancelling this term out, which is the best I can do. We have already discussed this several times before, then we are left with V1 dot is minus k1 x1 squared, which is negative semi definite.

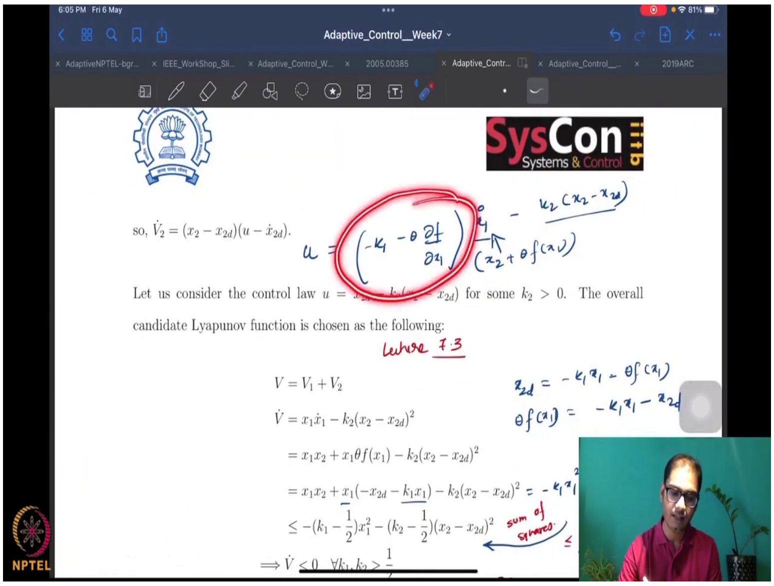
Again, try to remember the difference from the matched case, in the matched case, when we took the V1 dot, in the ideal circumstances, this is the ideal circumstance, because x2 is assumed to be exactly x2 desired. So even in the ideal circumstance, V1 dot is coming out to be only negative semi definite.

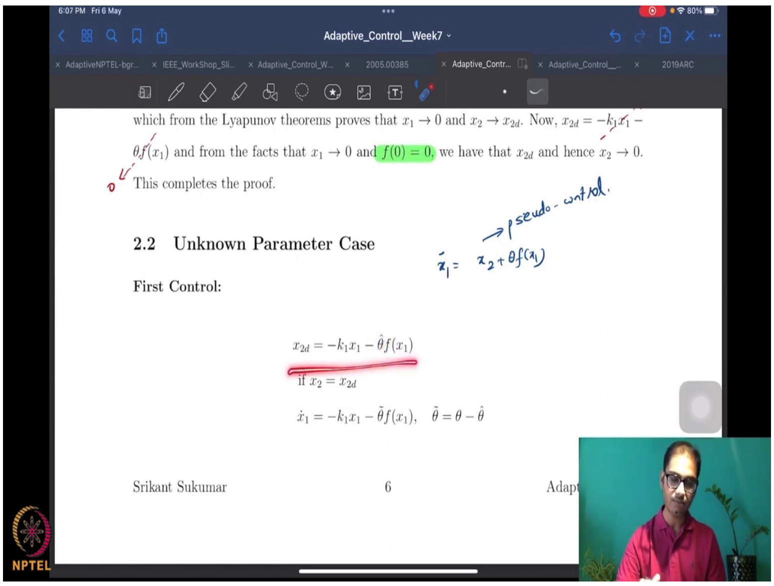
Whereas in the earlier version, in the matched case, V1 dot came out to be negative definite expression was exactly the same, but there was no theta tilde state. Therefore, the V1 dot was coming out to be exactly negative definite in the matched case. So that is what I will actually make comment on negative definite in the matched case.

(Refer Slide Time: 20:24)




Adaptive_Control_Week7
Lyapunov Function:
$V_2 = \frac{1}{2}(x_2 - x_{2d})^2$
$\dot{V}_2 = (x_2 - x_{2d})(\dot{x}_2 - \dot{x}_{2d})$
Choose $\dot{x}_2 = u = \dot{x}_{2d} - k_2(x_2 - x_{2d})$
$\dot{x}_{2d} = -k_1\dot{x}_1 - \gamma x_1 f^2(x_1) - \hat{\theta}\dot{f}(x_1)$
$= -k_1(x_2 + \theta f(x_1)) - \gamma x_1 f^2(x_1) - \hat{\theta}\dot{f}(x_1)$
$= -(k_1 + \hat{\theta}\frac{\partial f}{\partial x_1})\dot{x}_1 - \gamma x_1 f^2(x_1)$
$= -(k_1 + \hat{\theta}\frac{\partial f}{\partial x_1})(x_2 + \theta f(x_1)) - \gamma x_1 f^2(x_1)$
Srikant Sukumar
7
NPTEL


Adaptive_Control_Week7
SysCon
Systems & Control
so, $\dot{V}_2 = (x_2 - x_{2d})(u - \dot{x}_{2d})$.
Let us consider the control law $u = \dot{x}_{2d} - k_2(x_2 - x_{2d})$ for some $k_2 > 0$. The overall candidate Lyapunov function is chosen as the following:
$V = V_1 + V_2$
$\dot{V} = x_1\dot{x}_1 - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1\theta f(x_1) - k_2(x_2 - x_{2d})^2$
$= x_1x_2 + x_1(-x_{2d} - k_1x_1) - k_2(x_2 - x_{2d})^2$
$\leq -(k_1 - \frac{1}{2})x_1^2 - (k_2 - \frac{1}{2})(x_2 - x_{2d})^2$
$\implies \dot{V} \leq 0 \quad \forall k_1, k_2 > \frac{1}{2}$
NPTEL


Adaptive_Control_Week7
which from the Lyapunov theorems proves that $x_1 \to 0$ and $x_2 \to x_{2d}$. Now, $x_{2d} = -k_1x_1 - \theta f(x_1)$ and from the facts that $x_1 \to 0$ and $f(0) = 0$, we have that x_{2d} and hence $x_2 \to 0$.
This completes the proof.
2.2 Unknown Parameter Case
First Control:
$x_{2d} = -k_1x_1 - \hat{\theta}f(x_1)$
if $x_2 = x_{2d}$
$\dot{x}_1 = -k_1x_1 - \tilde{\theta}f(x_1), \quad \tilde{\theta} = \theta - \hat{\theta}$
pseudo-control
Srikant Sukumar
6
NPTEL

So, again, things are more complicated in the unmatched case. So now what we, of course, know that x2 is not identically equal to x2 desired, so we do the best we can, we try to have x2, track x2 desired, and for that we introduce a quadratic term in the backstepping error. So, we are trying to drive the backstepping error to 0. So, obviously, we introduce a quadratic term in the backstepping error in the form of V2, and this is again the same.

And now, what we would want to do is, so, in fact, I would say, want to choose but we would say is we want to choose the x2 dot, which is the control to be just again the same as before, which is cancelling these guys out. But now, remember, that x2 desired contains the theta hat. And other state no longer the theta, which is a constant and the derivative of the constant being 0.

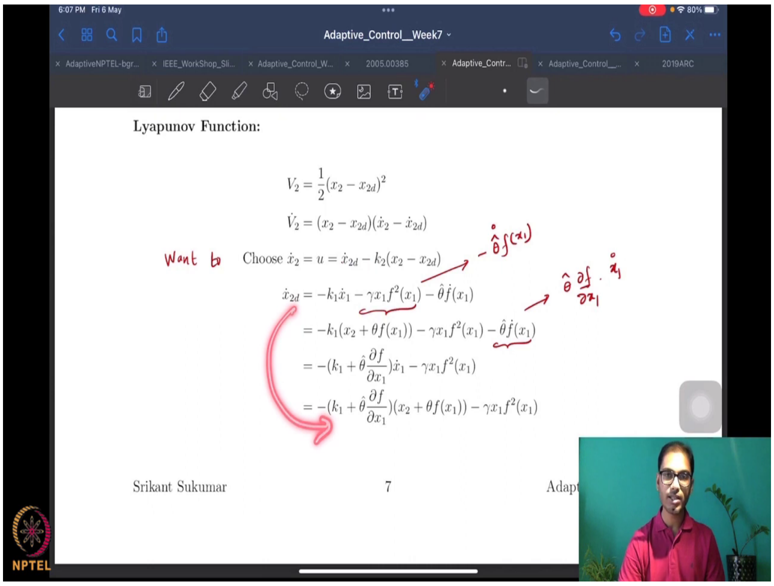
So, when I take the derivative of x2 desired, that is, I compute x2 desired for its implementation. What is it that I will get? I will get this minus k1 x1 dot, which is fine. Then I get this term is actually write it down. This term is minus theta hat dot, f x1. This term will minus theta hat dot f x1. And if I plug in this theta hat dot here, this is what you get here, this term is just minus theta hat dot times f x1, and then the next term is just I mean, because we are doing the product rule for the differentiation. So you have theta hat dot f x1 and theta hat f dot x1.

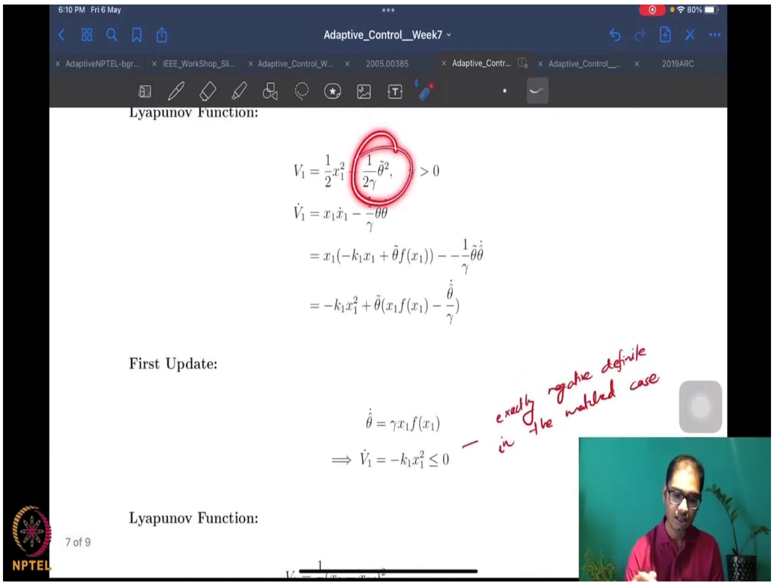
And, you know what happens is that, I mean, I, of course plug in for the x1 dot, I keep the theta hat f dot x1 as it is, but notice that theta hat f dot x1 is also theta hat partial with respect to x1 times x1 dot, so, not just an x1 dot here, but also an x1 dot from here. So that is what is written here. So this actually becomes, we do not need this step, this actually becomes k1 plus theta hat del f del x1, x1 dot. In fact this is what we had even here in the control law, minus k1 minus theta del f del x1, so this theta gets replaced by theta hat x1 dot. So that is the same.

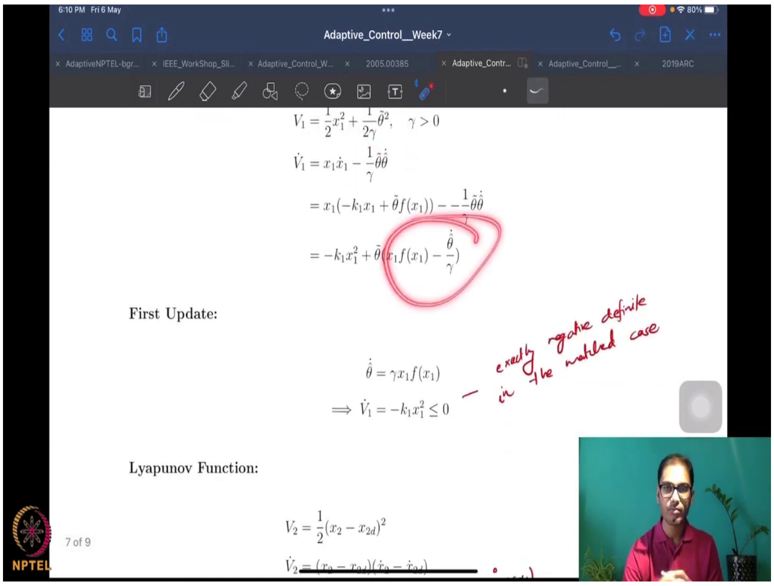
So that is what you have here. And you have minus gamma x1 f squared x1 which is just coming from this stuff, the derivative of the estimate. Now, you of course, substitute for x1 dot also, because we want to write the whole thing and we want to write the whole expression, excellent. So, what we are sort of trying to understand is that whether the control is implementable or not, and this is where things start to get wiry in this case in the adaptive case.

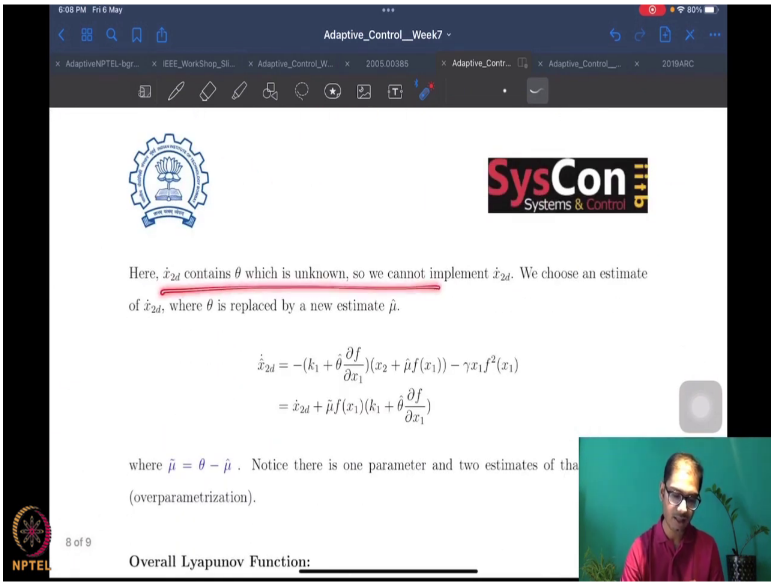
If you notice this x2 desired is of course implementable that is how we chose it, I mean, we introduced the theta hat in x2 desired just so that it is implementable and typically we assume all the states are known and theta is unknown, but then because theta is unknown, we introduce the theta hat here. So therefore, x2 desired is implementable.

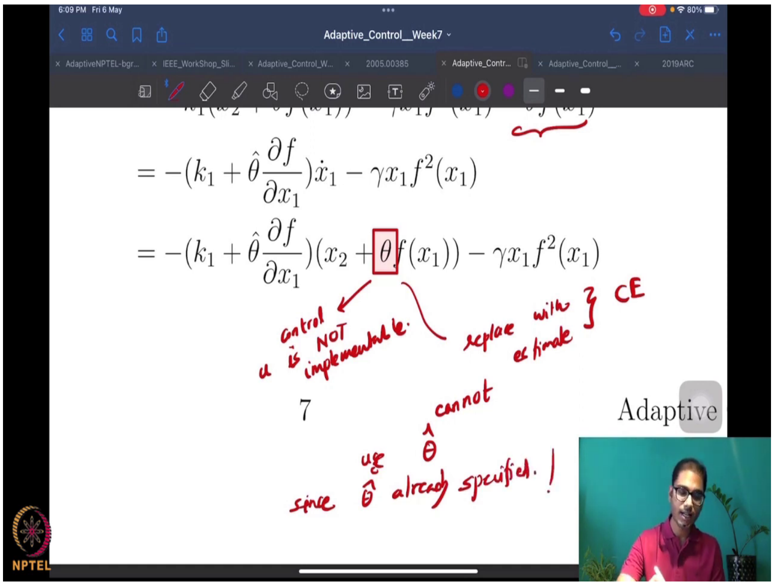
(Refer Slide Time: 24:37)






Adaptive_Control_Week7
$V_1 = \frac{1}{2}x_1^2 + \frac{1}{2\gamma}\tilde{\theta}^2, \quad \gamma > 0$
$\dot{V}_1 = x_1\dot{x}_1 - \frac{1}{\gamma}\tilde{\theta}\dot{\hat{\theta}}$
$= x_1(-k_1x_1 + \tilde{\theta}f(x_1)) - \frac{1}{\gamma}\tilde{\theta}\dot{\hat{\theta}}$
$= -k_1x_1^2 + \tilde{\theta}(x_1f(x_1) - \frac{\dot{\hat{\theta}}}{\gamma})$
First Update:
$\dot{\hat{\theta}} = \gamma x_1 f(x_1)$
$\implies \dot{V}_1 = -k_1x_1^2 \leq 0$
exactly negative definite in the matched case
Lyapunov Function:
$V_2 = \frac{1}{2}(x_2 - x_{2d})^2$
7 of 9
NPTEL


Adaptive_Control_Week7
SysCon Systems & Control
Here, $\dot{x}_{2d}$ contains θ which is unknown, so we cannot implement $\dot{x}_{2d}$. We choose an estimate of $\dot{x}_{2d}$, where θ is replaced by a new estimate $\hat{\mu}$.
$\hat{\dot{x}}_{2d} = -(k_1 + \hat{\theta}\frac{\partial f}{\partial x_1})(x_2 + \hat{\mu}f(x_1)) - \gamma x_1 f^2(x_1)$
$= \dot{x}_{2d} + \tilde{\mu}f(x_1)(k_1 + \hat{\theta}\frac{\partial f}{\partial x_1})$
where $\tilde{\mu} = \theta - \hat{\mu}$. Notice there is one parameter and two estimates of tha (overparametrization).
8 of 9
Overall Lyapunov Function:
NPTEL

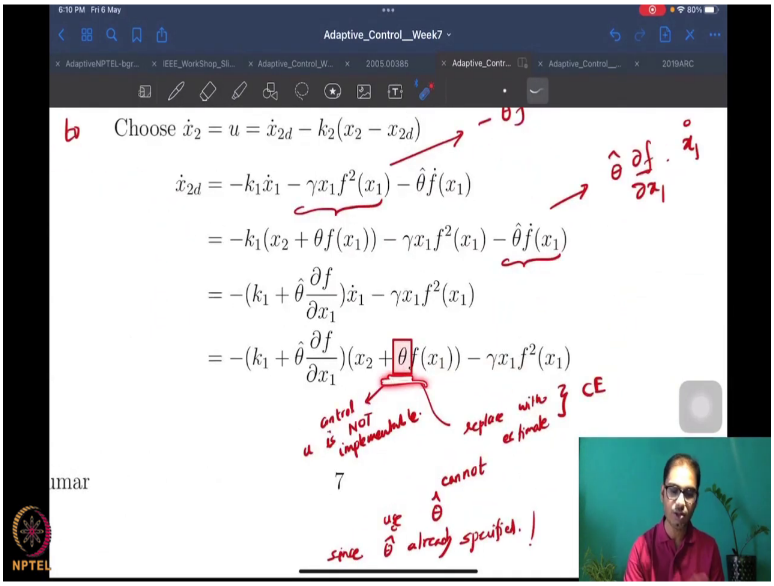

Adaptive_Control_Week7
$= -(k_1 + \hat{\theta}\frac{\partial f}{\partial x_1})\dot{x}_1 - \gamma x_1 f^2(x_1)$
$= -(k_1 + \hat{\theta}\frac{\partial f}{\partial x_1})(x_2 + \theta f(x_1)) - \gamma x_1 f^2(x_1)$
control u is NOT implementable.
replace with estimate } CE
cannot use $\hat{\theta}$
since $\hat{\theta}$ already specified.!
7
Adaptive
NPTEL

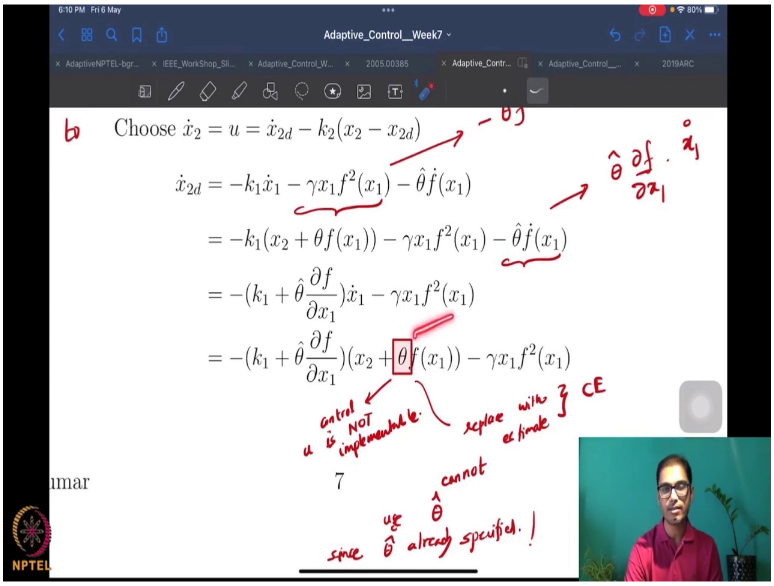
So, so in this control x2 minus x2 desired is of course implementable, but there is a problem with x2 desired dot. And, why, why is that a problem? Because if you note, when we took the derivative carefully, we came to this step, everything is implementable, but then we reintroduce the theta. How did we reintroduce the theta, which was an unknown? Because of x1 dot because the derivative of x1 shows up in the control law, because of this, therefore we reintroduce the unknown. So this is where we sort of start to have an issue with the implementation of the controller. So that is what I will say. So, in fact, I am going to highlight this.

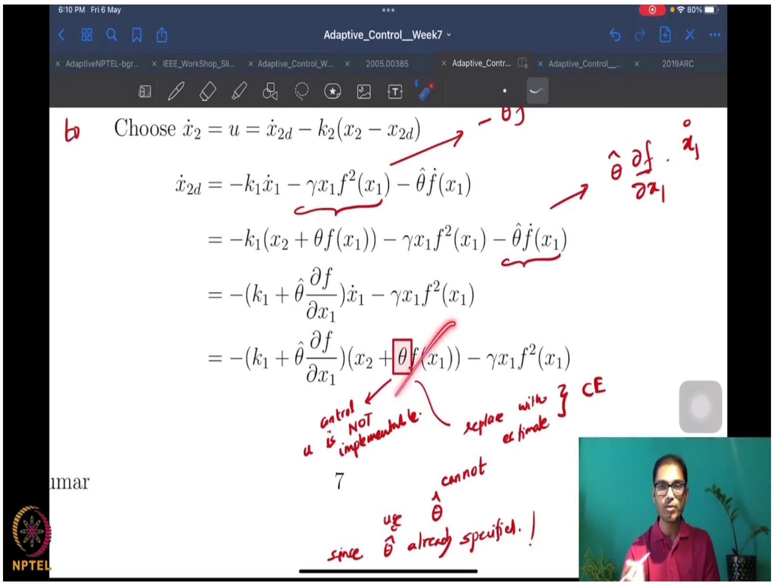
So, due to this control u is not implementable. The control u is not implementable because this theta, so that is what we say, contains the theta dot, which is unknown. So, that is what we want to do. Now, we want to figure out what we need to do. And we need to figure out what we need to do. We, of course, do the sort of obvious thing, instead of having a theta, we replace it with a new estimate theta mu hat.

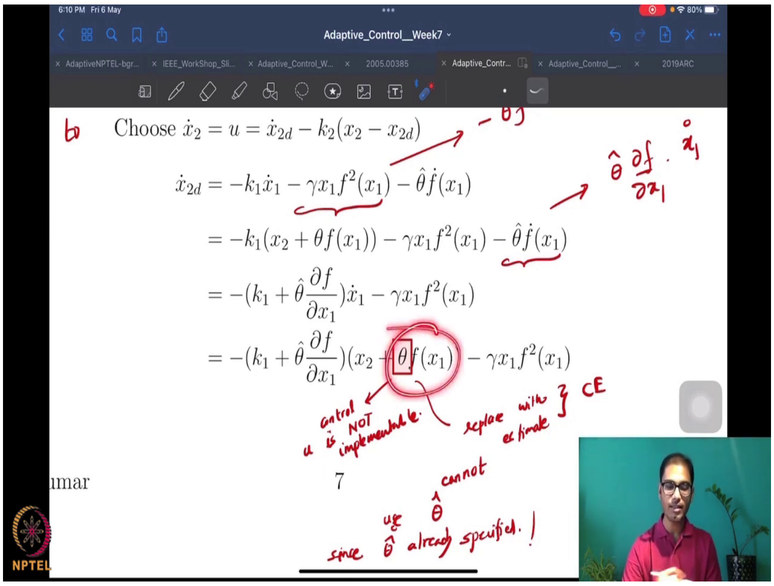
So because there is an unknown here, every time we get an unknown, that is what we do uncertainty equivalence, we replace it with the estimate. But the problem is the key problem to remember is that, so replace with estimate is the usual solution, is the usual solution via the certainty equivalence principle. But we cannot use theta hat since theta hat dot already specified. So, this theta hat dot is already specified, because theta hat dot is already specified when you do the Lyapunov analysis with this theta tilde squared. This guy, it will only cancel this one term, it will not cancel the theta tilde term that you get because of this change in the control.

(Refer Slide Time: 27:46)


Adaptive_Control_Week7
Choose $\dot{x}_2 = u = \dot{x}_{2d} - k_2(x_2 - x_{2d})$
$\dot{x}_{2d} = -k_1\dot{x}_1 - \gamma x_1 f^2(x_1) - \dot{\hat{\theta}} f(x_1)$
$= -k_1(x_2 + \theta f(x_1)) - \gamma x_1 f^2(x_1) - \hat{\theta}\dot{f}(x_1)$
$= -(k_1 + \hat{\theta}\frac{\partial f}{\partial x_1})\dot{x}_1 - \gamma x_1 f^2(x_1)$
$= -(k_1 + \hat{\theta}\frac{\partial f}{\partial x_1})(x_2 + \theta f(x_1)) - \gamma x_1 f^2(x_1)$
$\hat{\theta} \frac{\partial f}{\partial x_1} \cdot \dot{x}_1$
control u is NOT implementable.
replace with estimate } CE
cannot use $\hat{\theta}$
since $\hat{\theta}$ already specified. !
7
NPTEL

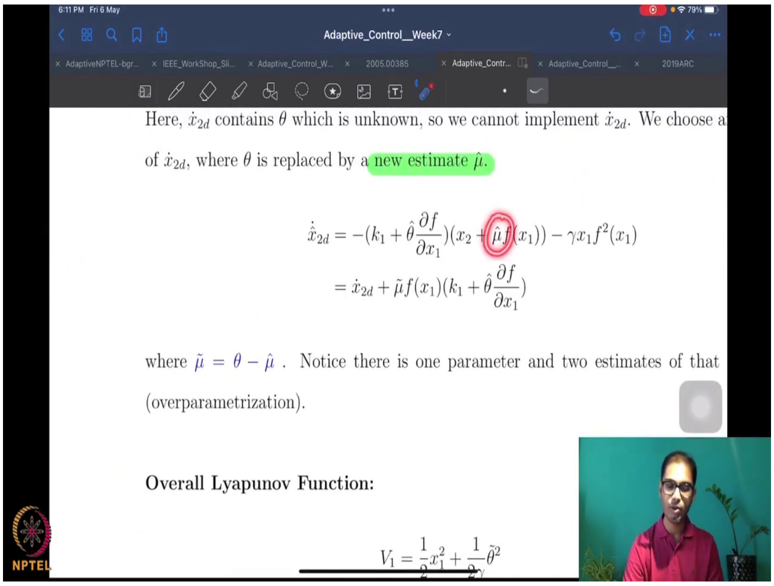

Adaptive_Control_Week7
Choose $\dot{x}_2 = u = \dot{x}_{2d} - k_2(x_2 - x_{2d})$
$\dot{x}_{2d} = -k_1\dot{x}_1 - \gamma x_1 f^2(x_1) - \dot{\hat{\theta}} f(x_1)$
$= -k_1(x_2 + \theta f(x_1)) - \gamma x_1 f^2(x_1) - \hat{\theta}\dot{f}(x_1)$
$= -(k_1 + \hat{\theta}\frac{\partial f}{\partial x_1})\dot{x}_1 - \gamma x_1 f^2(x_1)$
$= -(k_1 + \hat{\theta}\frac{\partial f}{\partial x_1})(x_2 + \theta f(x_1)) - \gamma x_1 f^2(x_1)$
$\hat{\theta} \frac{\partial f}{\partial x_1} \cdot \dot{x}_1$
control u is NOT implementable.
replace with estimate } CE
cannot use $\hat{\theta}$
since $\hat{\theta}$ already specified. !
7
NPTEL

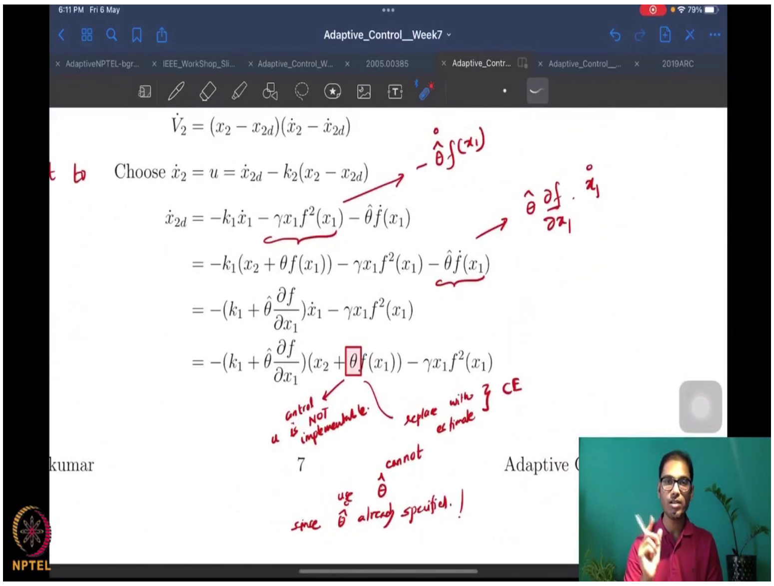

Adaptive_Control_Week7
Choose $\dot{x}_2 = u = \dot{x}_{2d} - k_2(x_2 - x_{2d})$
$\dot{x}_{2d} = -k_1\dot{x}_1 - \gamma x_1 f^2(x_1) - \dot{\hat{\theta}} f(x_1)$
$= -k_1(x_2 + \theta f(x_1)) - \gamma x_1 f^2(x_1) - \hat{\theta}\dot{f}(x_1)$
$= -(k_1 + \hat{\theta}\frac{\partial f}{\partial x_1})\dot{x}_1 - \gamma x_1 f^2(x_1)$
$= -(k_1 + \hat{\theta}\frac{\partial f}{\partial x_1})(x_2 + \theta f(x_1)) - \gamma x_1 f^2(x_1)$
$\hat{\theta} \frac{\partial f}{\partial x_1} \cdot \dot{x}_1$
control u is NOT implementable.
replace with estimate } CE
cannot use $\hat{\theta}$
since $\hat{\theta}$ already specified. !
7
NPTEL



This term will not get canceled. And, of course, you also want this term to be canceled, because I mean, in the ideal circumstance, this term goes away. But in the adaptive circumstance, that is how we cancel this theta tilde terms or the parameter error terms, is by introducing Lyapunov function and using the update law for the theta hat dot in order to cancel this stuff. Now, if we are unable to cancel the term corresponding to this, we will be left with a non negative definite term in Lyapunov function derivative. And that is a problem.

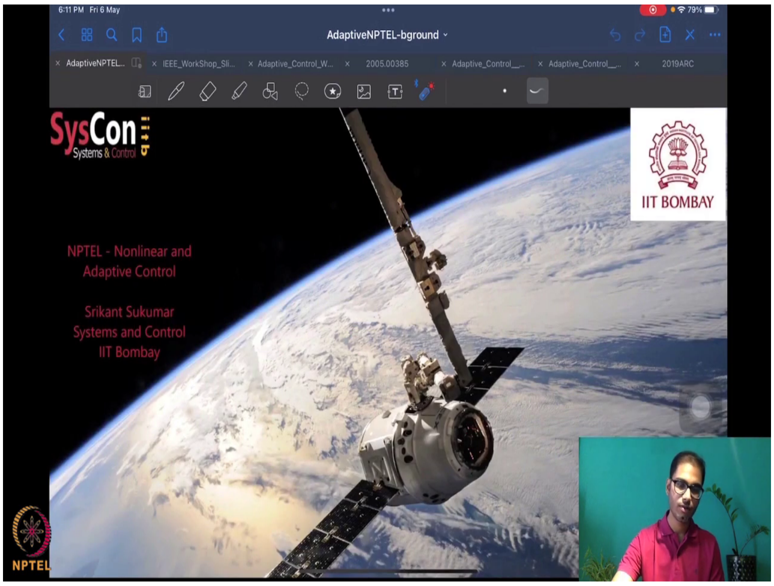
(Refer Slide Time: 28:24)





And therefore, we need to introduce a new estimate, mu hat. So, for the same quantity, for the same quantity, we have a new estimate. And that is what we do. So, this mu hat is what replaces theta hat theta. And we of course, construct an appropriate Lyapunov function and we will look at the Lyapunov function construction and the subsequent analysis in the upcoming session.

(Refer Slide Time: 28:54)



So, what is it that we looked at today, we sort of are continuing our analysis of this unmatched case, which as we can see is turning out to be much more involved than the matched case, these we saw, what the differences are, we saw that the x2 desired starts to contain the unknown parameter. And, as we go into the adaptive case, where of course, theta is assumed to be unknown, we land in some trouble in terms of how to define the update loss.

And then in the end, we come to a situation where the control which has this x2 desired dot also contains the theta again, so the theta seems to reappear and then we are looking for a way to deal with this. So we introduce a new estimate for this theta into the analysis.

So essentially, the only way we are able to resolve this issue at this stage is by having 2 estimates for the same unknown parameter theta. So, this is very standard in adaptive control by the way, in a lot of scenarios, we do what is called as over-parameterization. So, we will talk about this and how this unmatched state backstepping adaptive control will happen in the subsequent sessions. See you folks again. Thank you.