Nonlinear Adaptive Control
Professor Srikant Sukumar
Systems and Control
Indian Institute of Technology, Bombay
Week 7
Lecture No: 38
Backstepping in Adaptive Control (Part-2)

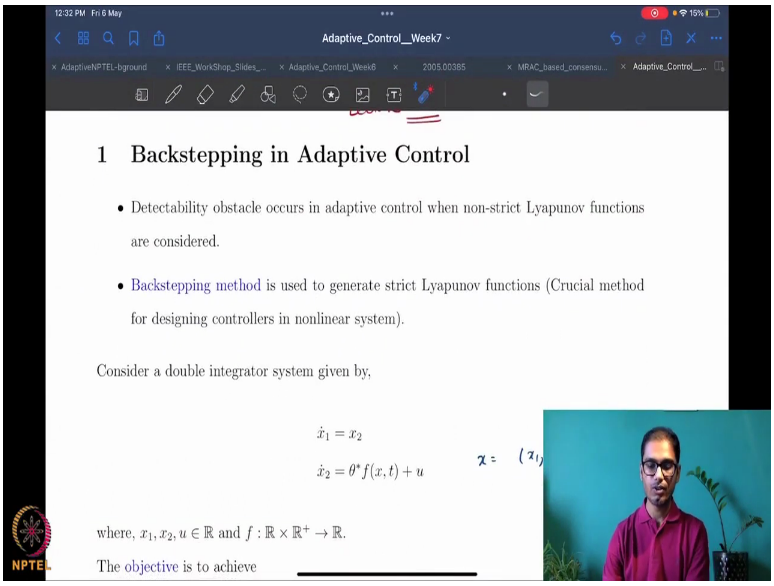
(Refer Slide Time: 00:16)

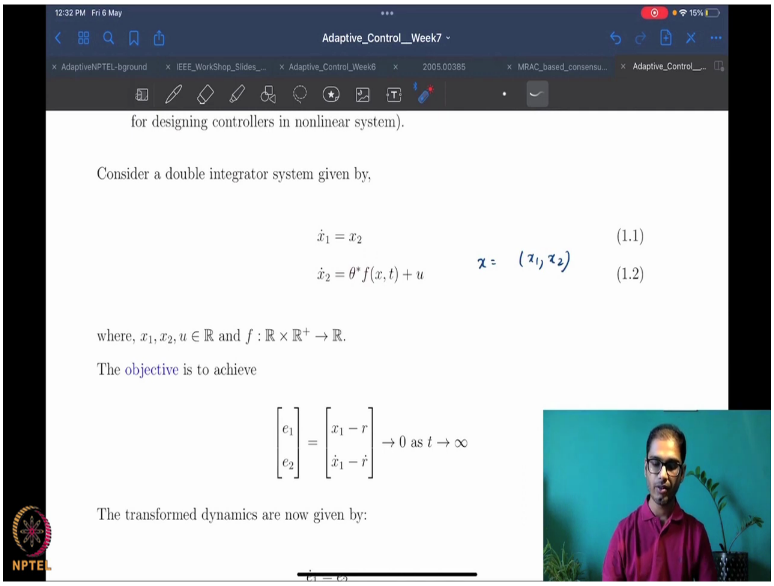
Hello, everyone. Welcome to yet another session of our NPTEL on nonlinear and adaptive control. I am Srikant Sukumar from Systems and Control, IIT Bombay. So we started with our excursion into this seven week of the course on Nonlinear Adaptive Control and we are well underway to designing adaptive control algorithms.

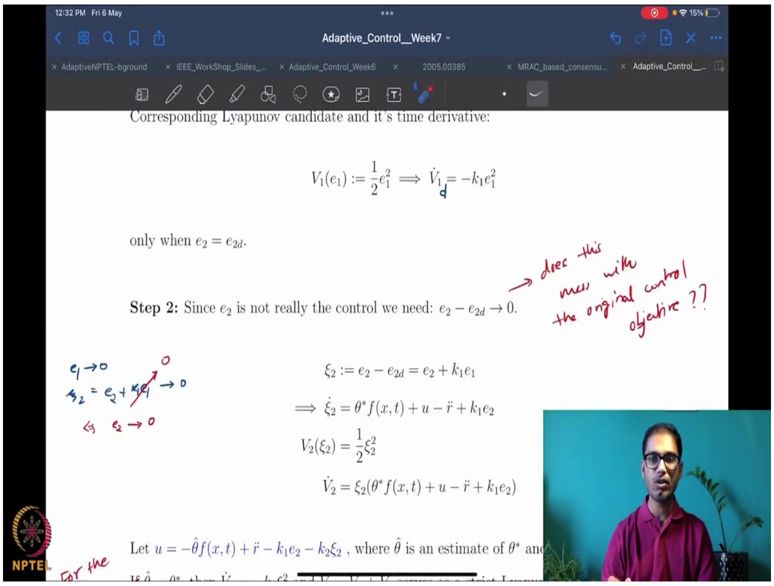
So we have already seen adaptive control algorithm design for first and second order scalar systems and this week, we have started the journey into a new method of adaptive control design which is the backstepping-based adaptive control design.

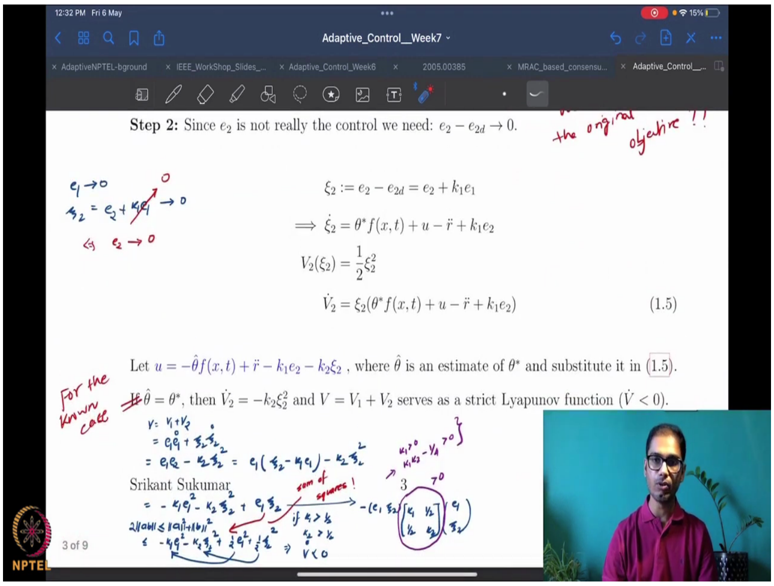
So we are as always motivated by this very, very nice background images which depicts the application of nonlinear adaptive control to space systems such as an autonomous satellite which is orbiting the Earth.

(Refer Slide Time: 01:32)









So what we were doing in this previous session was we had just started looking at backstepping in adaptive control. So as a first step, we had of course, introduced the problem setup which is still an adaptive control type problem setup.

However, the parameter was assumed to be known and therefore, we were simply doing a standard nonlinear control backstepping design. So I hope that what we did in this previous session gave you a pretty good idea of how backstepping control design looks.

So I keep emphasizing this regularly and I will do so again that this backstepping design is in fact, a way of constructing augmented Lyapunov functions. So it is, that is really what it is. It is a way of constructing augmented Lyapunov function.

So you start with the Lyapunov candidate for the first subsystem, then you construct a Lyapunov candidate for the second error subsystem, and then you sum the 2 individually Lyapunov candidate for the entire system.

So with this simple method, you are actually able to get a handle on constructing Lyapunov function because a lot of times many students asked me how to construct these Lyapunov functions because it looks like we are pulling it out of thin air and in a lot of cases, this is in fact true.

A lot of designing Lyapunov functions is in fact involving creativity. And of course, we are motivated by other Lyapunov function designs that are already existing in literature but fundamentally, yes, there is a lot of creativity.

And therefore, this kind of method like backstepping which allows you to build Lyapunov functions, piece by piece, is something that is, rather useful and might even be something that can be automated.

So, so we have already seen how backstepping control design works. What we want to do starting today is to look at the unknown that is the actual adaptive control scenario of using backstepping control. So let me mark this lecture as lecture 7.2. So this is the second lecture.

So now what happens when theta star is unknown? So what we do is, so if you notice, we were very careful and we actually defined the control law with the theta hat itself to begin with just like we do in certainty equivalence.

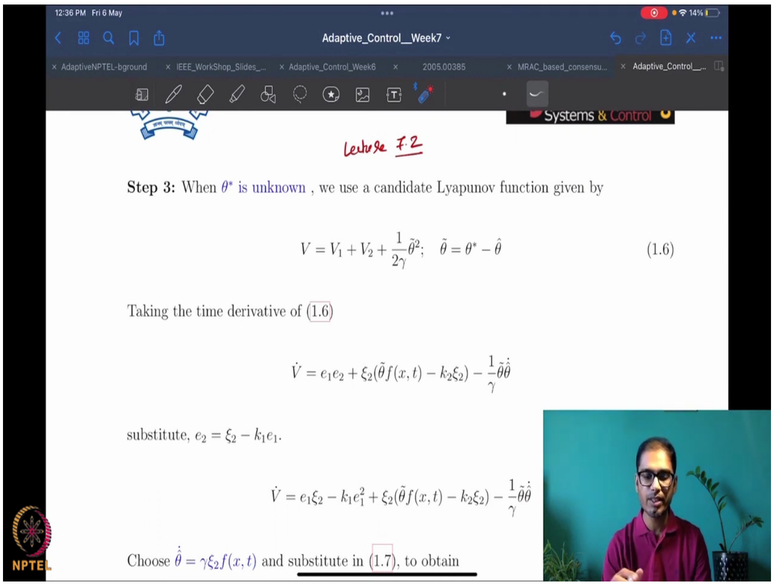
So the principle is still the same. We still use like the certainty equivalence principle. So we designed it with a theta hat, and we said that for the known case, we just substitute theta star for theta hat and then we did the subsequent analysis.

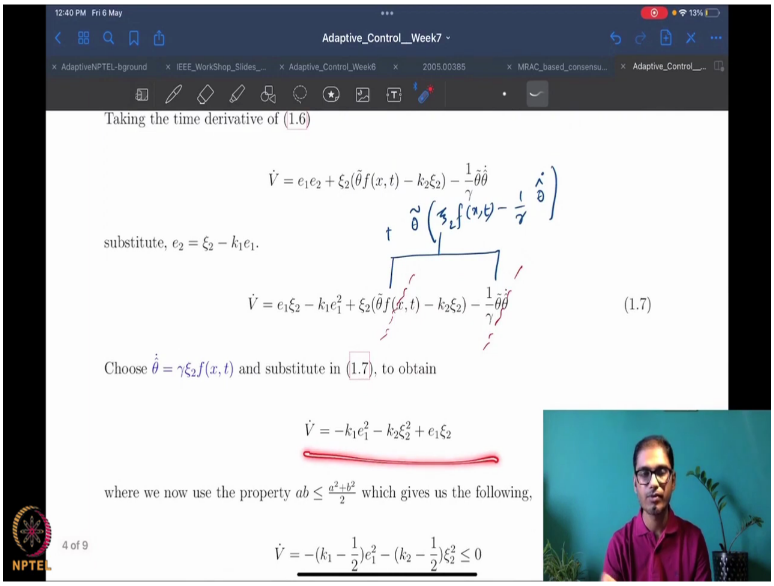
Now, when theta star is in fact, unknown, then nothing changes. The theta hat remains as it is because it is an estimate and as always we need to come up with an update law for the estimate.

We also noticed that the Lyapunov candidate that we had come up with in that the V2 piece was in fact exactly the Ortega construction and the V1 piece was additional. This additional piece, in fact, helps us to make the Lyapunov, I mean helps us to make the Ortega function a Lyapunov candidate function, great.

So the controller remains with an identical structure. The only thing is now this, this and this do not cancel out. This cancellation does not happen. So these in these forms are actual difference which is theta star minus a theta hat. Everything else cancels out because they are known and then this good term is introduced as before also. Excellent. Excellent.

(Refer Slide Time: 05:51)





Now, let us see what happens. So now what we do is as we are always used to doing whenever there is an unknown parameter, we simply add a term corresponding to the unknown parameter error and then it is theta tilde squared by 2 gamma again and as always theta tilde is defined as theta star minus theta cap.

So this is standard throughout the course. You always define the error as the true value minus the estimated value. I mean, you can always do the other way around, and so on and so forth but this is just a convention that we are following in this course. You are free to do it the other way as long as you are consistent and do not mess up theta tilde dot.

So for us whenever we compute theta tilde dot, it is minus theta cap dot because theta star is a constant. So this is the only thing where we need to remember how we define the theta tilde. So great.

So I took the same Lyapunov candidate which is what we have been doing in adaptive control. You take the known case Lyapunov candidate and add a term corresponding to the; quadratic term corresponding to the parameter and that is exactly what we are doing here.

And then we take the derivative, what is the derivative? We have already done this. We get an e1 e2 and we get a psi 2, psi 2 dot. Earlier there was only this term. But now like I said, because the parameter term does not cancel out, you also have this term coming and then of course, you have a minus 1 over gamma theta tilde theta cap dot just coming from here.

Now substituting for e2, again this is something we did in the previous session also. We substitute for e2 because now the new variables are e1 and psi 2. So e2 was not really a variable.

So we substitute for e2 in terms of psi 2 and e1 and we get e1 psi 2 which is the mixed term and you get a minus k1 e1 squared and then minus k2 psi 2 squared and so on and so forth.

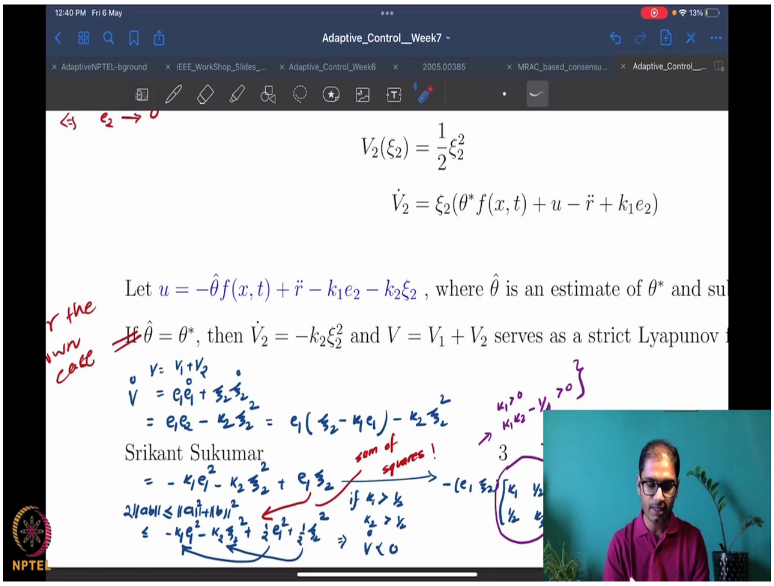
Now if you notice, this term and this term contains theta tilde in it. So these terms together can actually be written as theta tilde psi 2 f x, t minus 1 over gamma, theta cap dot.

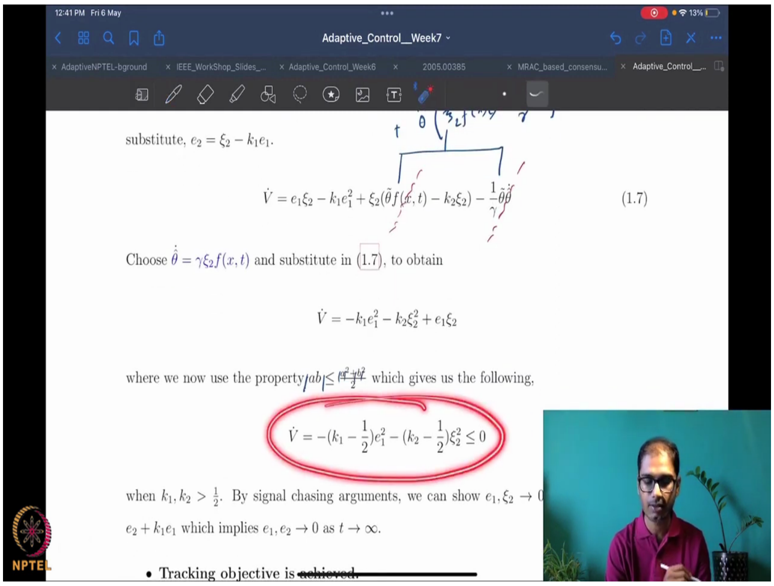
So what do we do? We know that we cannot really make this negative definite or anything because that would require having a theta tilde in my update law which is not allowed. Theta tilde is unknown to me. So I am not going to do anything ridiculous like that, right,

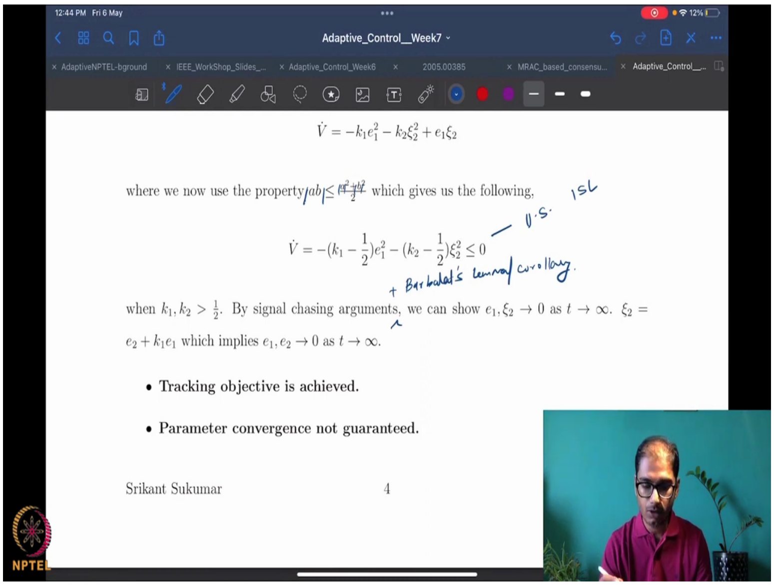
So I picked theta cap dot just so I can cancel this. I do the best I can. I cannot really introduce a negative term but I can at least cancel the bad stuff and all of this is known because we of course, again we are assuming that we have full state feedback in this entire course.

We know e1, e2. So we know x1, x2, r, r dot. Therefore, we know e1, e2 and so on and so forth. So we have full state feedback. So therefore, this entire thing is known, I can cancel it out by update law which looks like this. So once I cancel these out, I am left with just I cancel these out. So I am left with just these 3 terms.

(Refer Slide Time: 09:43)







And notice now that this looks exactly like this; exactly like this. Same. So what do we do? Sum of squares. Now again, this should not be surprising that this V dot. I mean, sorry I should have written as V dot here. This is V dot actually. That is fine. It is from the last lecture. No problem.

So it should not be surprising that V dot looks identical because this has been our experience in all adaptive control problems, our V dot always turns out to be the same as the known case. Although we start with a different V for the unknown case, our V dot always turns out to be the same as that for the known case.

This has always been the case with the adaptive controllers we have designed until now. So this should not come to you as a surprise. So as always, we do this, sort of use this nice property.

I mean, like I said, there is norms here and whatever I mean, you can use absolute values and norms here. But it is the same. In this case, these are all scalar quantities. And so once you do that, you remember from last time that you get this.

Now the only differences in the previous lecture, this was negative definite but now it is only negative semi-definite. Why? Simple, now, our V is no longer the same. Although the V dot turned out to be the same, V is not the same. V, in fact, has an additional state which is the parameter error state and so all states do not appear in V dot.

And if all states do not appear in a function, what do we know? Well, that it cannot be definite. So since only 2 out of the 3 states appear, it is only negative semi-definite, Great.

However, it is not difficult for us. I mean, we are not even doing these arguments and we are not writing these arguments explicitly anymore but we know that we can use signal chasing arguments and Barbalat's lemma.

So I am going to say signal chasing arguments plus Barbalat's lemma or and corollary. We can use signal chasing arguments, Barbalat's lemma and its corollary to show that e1 and psi 2 are going to go to 0. And we already know that psi 2 is e2 plus k1 e1 which implies e1 and e2 are both going to go 0. Great.

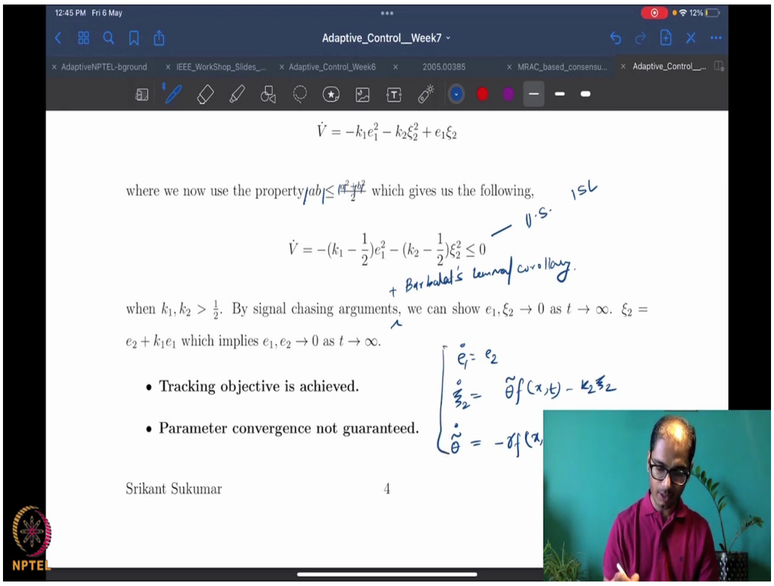
So as before e1 as in the case of the nonadaptive case or that is the case where the parameter was exactly known to us I still achieve tracking. Why? Because just the fact that V dot is negative semi-definite by standard Lyapunov theorems already gives me uniform stability in the sense of Lyapunov.

I already have uniform stability for the entire system like which means all the trajectories remain bounded, all nice things happen. If you start close to the origin, you remain close to the origin. If you start close to zero error, you remain close to zero error and all that nice stuff.

On top of it, I can also prove that asymptotically e1 and psi 2 go to 0 that is e1 and e2 goes to 0 equivalently which means that my tracking objective is achieved that is I tracked the desired signal while remaining bounded all the time while ensuring stability.

Parameter convergence as always is not guaranteed. Parameter convergence is of course not guaranteed. So I mean, if you again look at the closed loop system, I mean, let us try to formulate it in again in terms of our persistence type results. Let us see what it looks like.

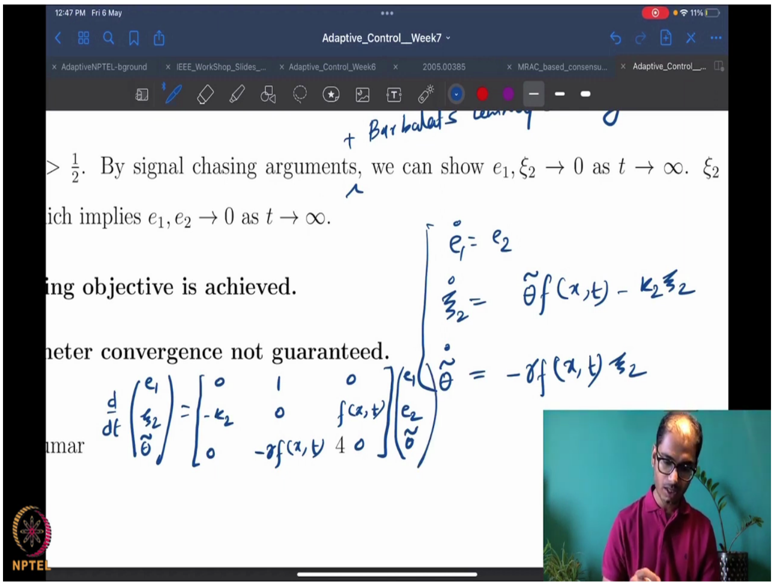
(Refer Slide Time: 13:54)



So you have your closed loop system is e1 dot is e2 and e2 dot is let us see well, actually, I am going to write this in terms of psi 2, if you do not mind because those are the new states. Psi 2 dot is minus theta tilde f x, t. Is it a minus or is it a plus?

No, it is just a plus and then I have a minus k2 psi 2 and then I have theta tilde dot as minus of theta hat dot and theta hat dot is this. So this is minus of gamma f x, t, gamma f x, t time psi 2.

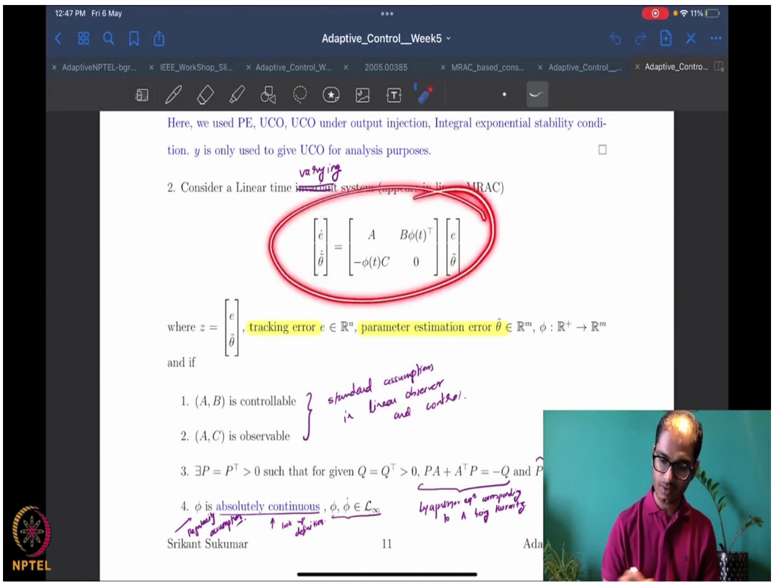
(Refer Slide Time: 15:13)

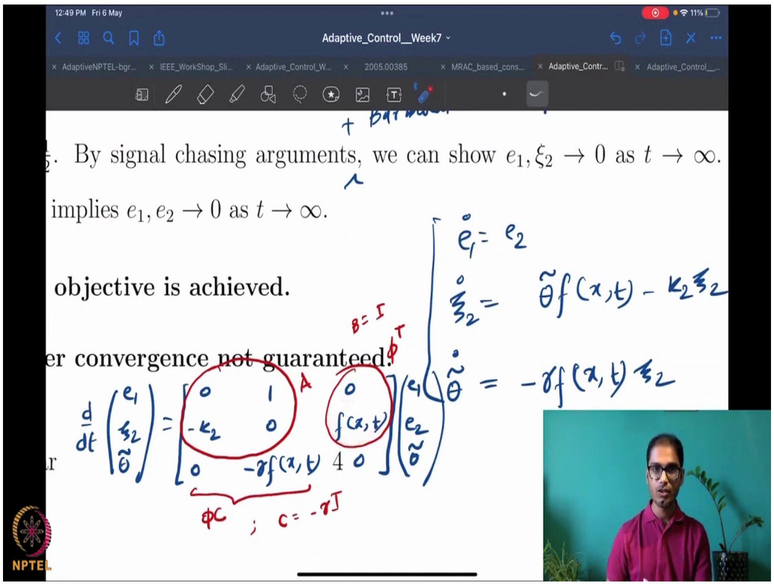


So the thing that you will notice is if I write it again in this nice state space form, let us see. This will come out to be e1 e2 d dt. Why do I keep saying e2? My bad. e1 psi2 d dt is equal to 0, 1. No, no. This is not how I want to do this. I apologize.

I want to write the whole thing d dt of e1 psi 2 and theta tilde as you have this nice block diagonal structure which is 0, 1 and then you have a 0 and then here you will have minus k2, 0 and theta tilde. I apologize. This is f x, t and here I have similarly 0, minus gamma f x, t, 0; e1 e2 theta tilde.

(Refer Slide Time: 16:57)





Now, after all of this work, it should not be difficult for you to see; unfortunately, I closed this. Let me look at I think week 5 maybe we have this. It starts to look again like it has this structure, like it has the structure. Because the error is e1 psi 2. e is equal to e1 comma psi 2.

e is actually e1 psi 2 in this case. It is a vector and then, so that is this system. So this I will even mark this, this entire thing is the A matrix. Then you have this as 0. Same. And then you have the connection to the theta tilde via this phi.

So the phi is of course your this piece is phi. Oh, I am sorry. I should do undo. Your phi is in fact, this piece. This is phi. And so, well, fine, I guess this is what you call phi transpose and not phi. So this is phi transpose.

So in this case your B is just the identity matrix. So B is just the identity matrix here. So B equal to identity and this is C phi where C is equal to minus gamma identity matrix. It is, I mean you can just call it a scalar in this case. There is no identity matrix but that is okay. That is fine.

These, a scalar is 1 and the C is, so it is phi times and C is still minus gamma. So you see that it still has the similar structure and we keep we kept saying that this structure is rather useful. And now remember that here again phi was only a function of time and here it is a function of time and the state.

So as always, we have to use the integral lemma type results. The more general integral lemma results. But, we can we have a good hope that we actually mimic this. Because (A, B) being controllable, (A, C) observable is pretty easy and A being stable.

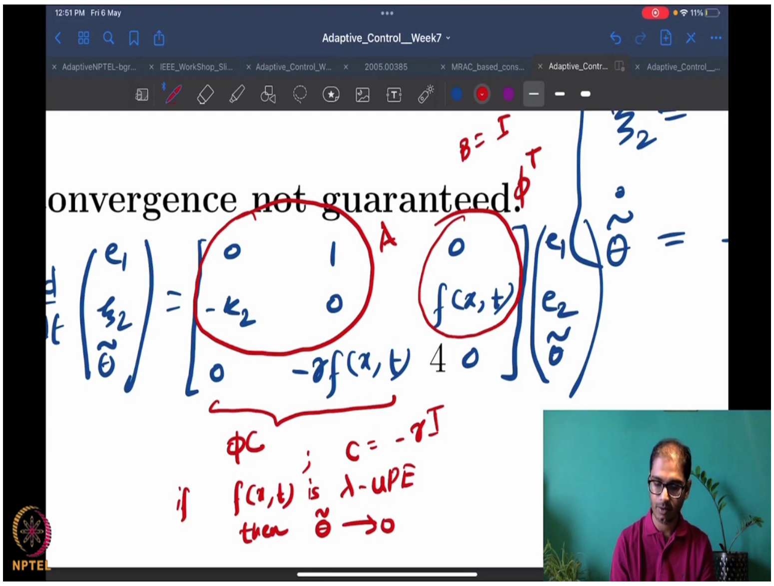
So A as you can see is already a stable matrix. This is a stable matrix, very obvious. (A, B) being controllable is not difficult. (A, C) being observable is also not difficult to prove.

So if we use the general integral lemma even if f is a function of the state and time, we will be able to claim that this system is exponentially converging as long as this is has this, in this case it should have lambda uniform persistence of excitation.

As long as this function has a lambda uniform PE condition satisfied, you will have this entire system to go to zero exponential. And that is essentially what you will need.

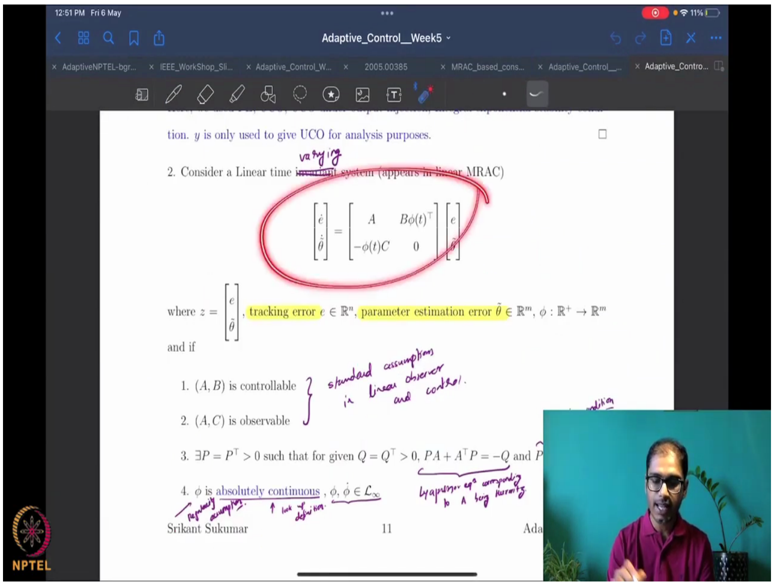
And that is essentially what you will need. You need that all these states go to 0 and when all these states go to 0, until we only prove that these 2 goes to 0. But when we prove all these 3 states go to 0, we also have the parameters to converge.

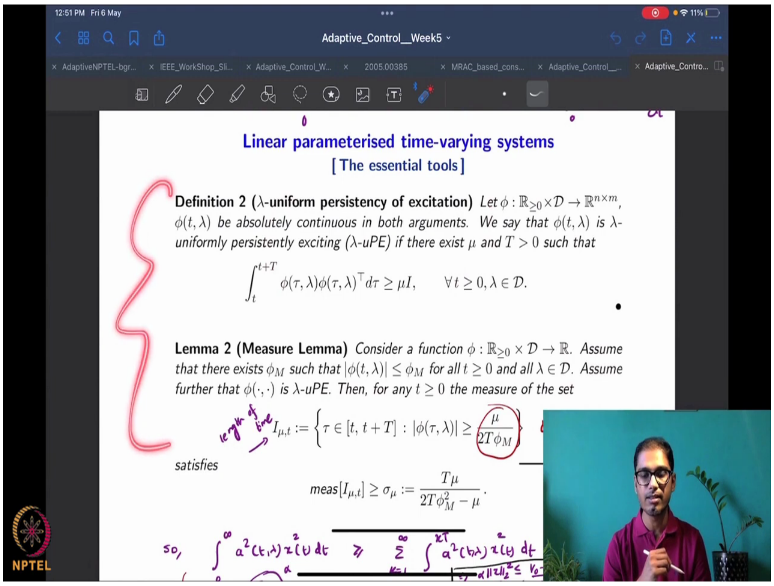
(Refer Slide Time: 20:30)



So in general, it is not difficult to say that if f x, t is lambda uniform persistently exciting then, theta tilde also goes to 0. So this is nice, so we started with a nonlinear problem but we are able to use some result here which looks like a linear result.

(Refer Slide Time: 20:54)



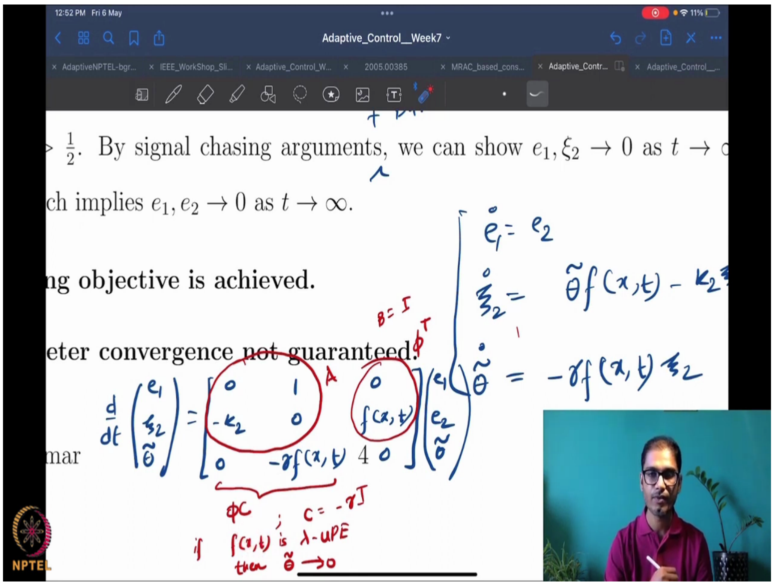


Again, we are not using exactly this result, remember but we are using the corresponding result via the lambda uniform PE and these results.

So we are actually not prove that result but the equivalent of this result exists when this phi is both state and time dependent. Because the state is just written as a solution with the parameter being the initial condition, initial time.

And therefore, you need notions of lambda uniform persistence, where lambdas are some parameters in this case, the lambda being t zero and x zero. So great.

(Refer Slide Time: 21:31)



So there is also conditions that one can talk about under which you will get convergence of the parameters. So, remember that all of this because this f, whenever you talk of persistence of the signal which contains the state, this is indirectly going to be connected to the persistence of the reference trajectory because the state is trying to track a reference trajectory.

So the state is trying to because those are the errors because the state is trying to track a reference trajectory, only if your reference trajectories persistent can you have this sort of a condition satisfied.

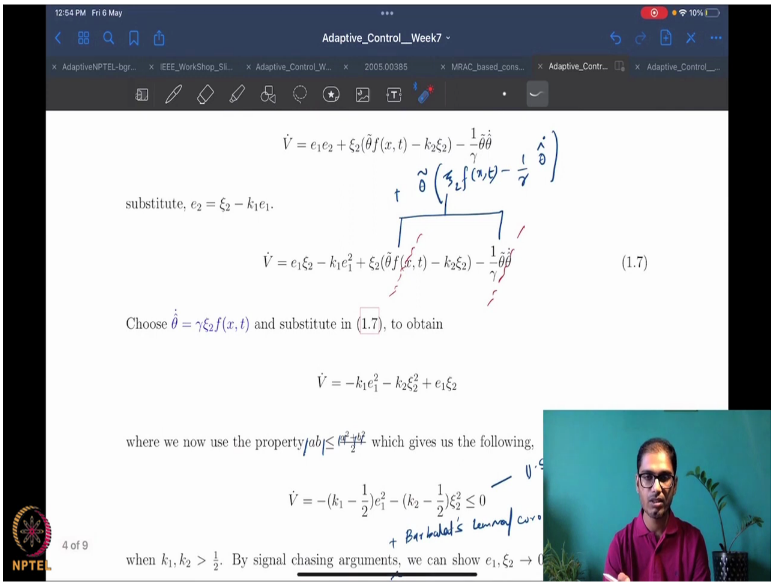
If your reference trajectory is just a constant horizontal line, then most probably, you will not get persistence and therefore, you will not get you would not be able to prove that the theta tildes are going to go 0.

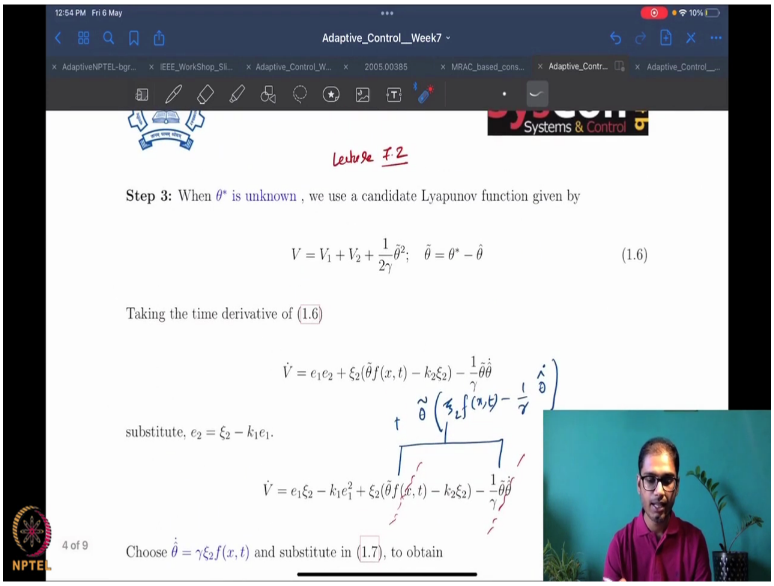
So just keep this in mind that that parameter convergence or true parameter identification is directly related to your persistence of your trajectory. And again, what is persistence of a trajectory?

It means that you have sufficient number of frequencies. It is the standard identification question. It has to have sufficient number of frequencies. Only then it is going to be a persistent reference trajectory.

So if your interest is not just in tracking but also in identifying a parameter, then you must use a persistent trajectory with several frequencies. Excellent. Good, good. I hope that is evident.

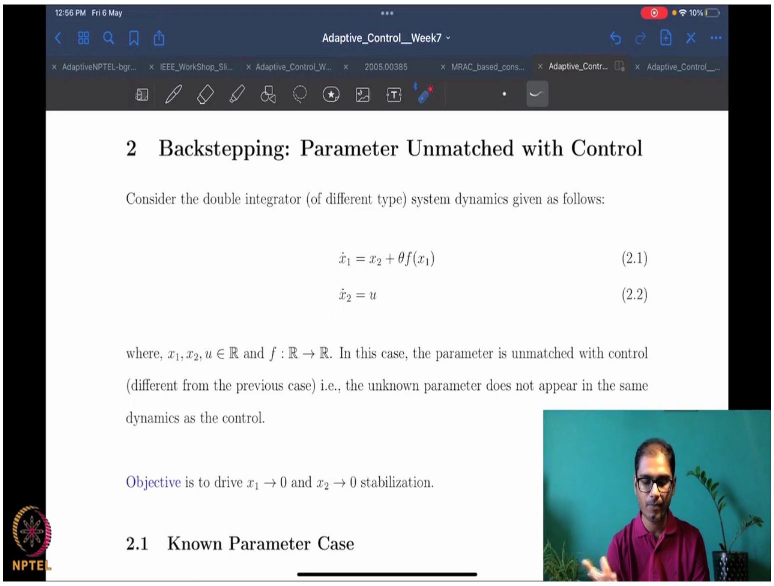
(Refer Slide Time: 23:21)





Okay so, we have this adaptive result then which is the using this adaptive backstepping.

(Refer Slide Time: 23:30)



So now, one of the things that we of course also are able to do is of course to claim that the detectability obstacle was avoided. Why? Because we for the known system we constructed a strict Lyapunov function.

So backstepping method not just helps you construct a Lyapunov candidate or any Lyapunov candidate, it helps you construct a strict Lyapunov function. So this is important. So once you do that for the known case, that for the unknown case things are going to be nice. Not a problem.

So, now, I mean, we want to introduce another different kind of problem. Now we want to introduce another different kind of problem, these are the kind of problems where now Ortega construction and such may or may not be usable. And when backstepping still gives you a very format or very clear path on how to design an adaptive controller.

So this is the case of parameter which is unmatched with the control. Until now, your parameter was always appearing in the same dynamics as the control. So here we have sort of, change the problem. I mean for a good reason,

I mean this, it may not always be the case that your unknown parameter always appears in the same dynamics as the control. It may so happen that the parameter appears in a different piece of the dynamics. So we have sort of flipped the problem.

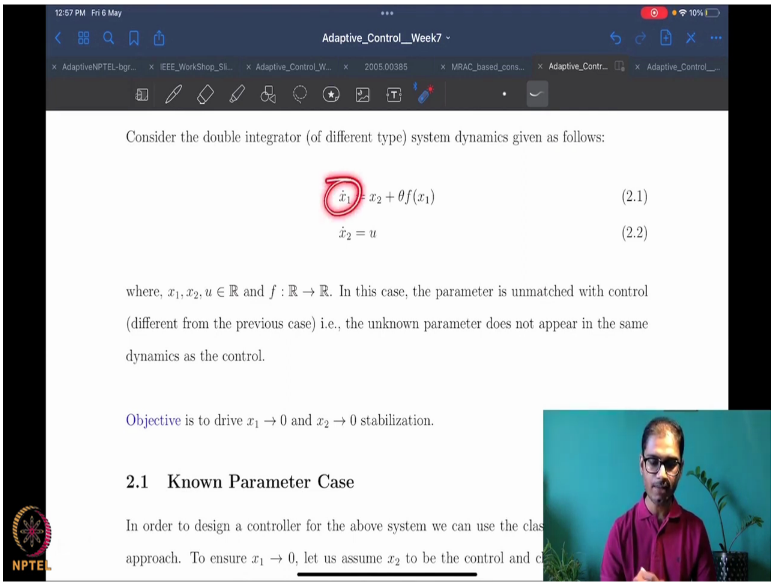
So we still have x1 dot as x2 and x2 dot as u like a double integrator type system but then you have theta f x1 appearing in the first piece that is x1 dot is x2 plus theta f x1. And now, this theta is as always denoting the unknown.

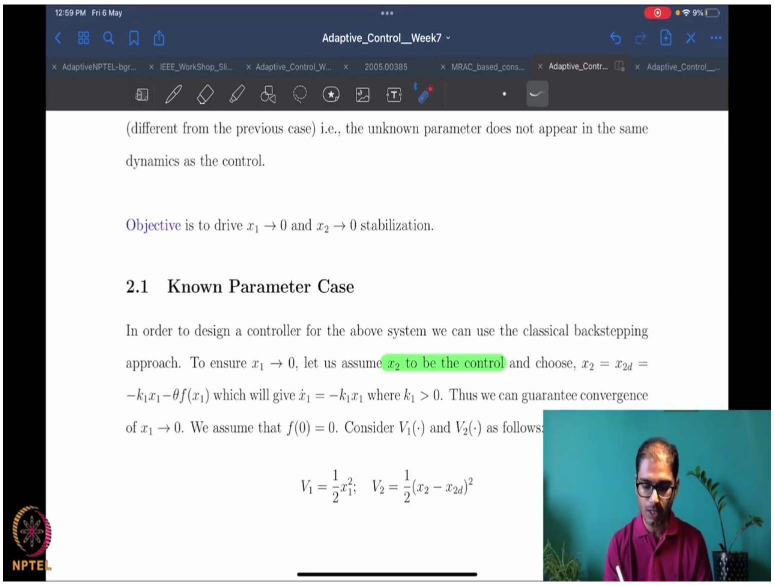
So that is why this is called parameter is unmatched with the control. The unknown parameter does not appear in the same dynamics as the control.

Now, because we are sort of just demonstrating how backstepping can be used, we are just doing the stabilization problem. That is we just want to drive x1 and x2 to 0. Now remember that the tracking problem is not different, honestly speaking.

It just involves you adding a trajectory term and so on and so forth. But it is not going to change anything significantly. So let us not worry too much about doing the tracking problem instead of stabilization problem. So the method that we illustrate here are going to remain exactly the same. So it is not going to significantly impact us.

(Refer Slide Time: 26:20)





So let us see what we would do in the known parameter case. How we would do the backstepping controller design. So, so first thing is we want to make sure, as always, we look at just the first piece and we want to ensure that x1 goes to 0 and we of course, assume that x2 is the control. This is what we do.

We basically think of the second state as the control for the first state and then we try to stabilize the system. What do we do to stabilize? We try to have a track an ideal system. And what do I do? Simple, cancel this introduce a good term. So that is it cancel this, introduce a good term. Simple.

So x2 is x2 desired. Sorry. We call it x2 desired because this is not really x2 but we want it to be this therefore, we call it x2 desired and we construct this as such. Once we construct it as such, you get x1 dot is minus k1 x1 and we get a corresponding V1 which is one half x squared, x1 squared.

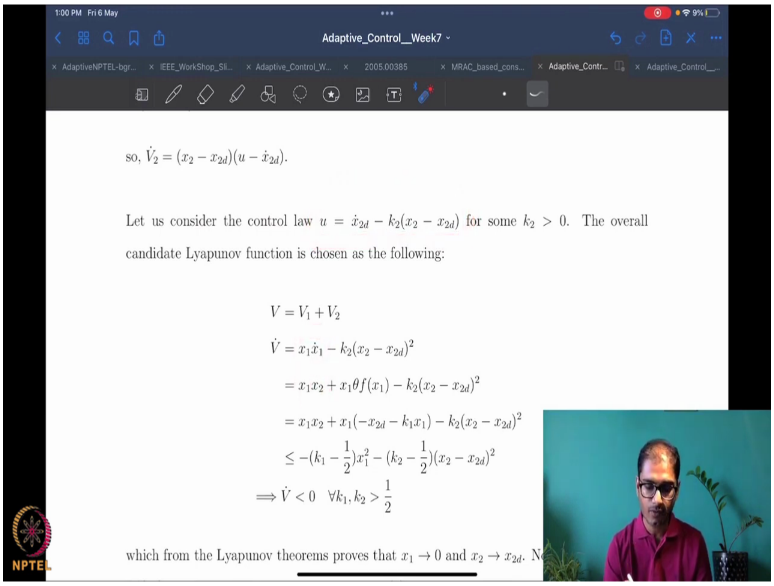
We do not even try to compute the derivative because we know what it will be. It will be minus k1 x1 squared. So this is corresponding to this. This is the same as last time. This, these 2 are not different. The only difference is earlier x2 was just this piece but now there is an additional piece because of the fact that there is an additional piece of dynamic state.

Again, this is the known parameter case. So theta is known, therefore implementable. So this is denoted as x2 desired. And what do we do? As we do all backstepping, we construct a V2 which is just the error between the true value of x2 and the desired value.

Because I know that I cannot actually have x2 to be identically equal to x2 desired. So what do we do? Do the next best thing. Try x2 to track x2 desired. And how do we do that? Just choose an V2 which is a quadratic function of x2 minus x2 desired.

Again, same as what we did last time. The only difference being x2 desired has a different definition. Slightly more complicated, in fact even as a non-linearity.

(Refer Slide Time: 28:41)



And what do we get for V2 dot? It is just x2 minus x2 desired times u minus x2 desired dot. So the x2 dot is of course, simpler. That is just u and x2 desired dot, I have written as it is. We have not really expanded because they are known quantities, derivatives of known quantities. You can implement them. No problem.

So what happens here? So what do we do? We choose the control as something that cancels this and introduces a nice negative term. That is it introduces a nice negative term here.

And then things are rather straightforward. I mean, we will again, continue this in the subsequent session but things are straightforward. We take V as V1 plus V2 and then you compute the derivative which is x1, x1 dot plus k2.

And then V2 dot is just minus k2 x2 minus x2 desired dot squared because of this choice; because of this choice. And then x1 dot is x2 and now because we have changed the states, we have to write x2 in terms of x2 minus x2 desired. That is what we do. That is what we do.

And we will of course look at this subsequently. So the Lyapunov essentially look very similar to what we have been doing until now for backstepping. So great, great. So the Lyapunov analysis is where we will continue from the subsequent session.

(Refer Slide Time: 30:23)

So what we have seen today is we sort of did the adaptive version of the backstepping control design from last time. And we also looked at the new problem where the control and the unknown parameter are unmatched and we are starting to look at how to do a backstepping based adaptive controller design for such systems.

So we will look at this unmatched problem again in the next session and we will try to see how the adaptive control design looks and possibly differs from the previous matched design case. Thank you.