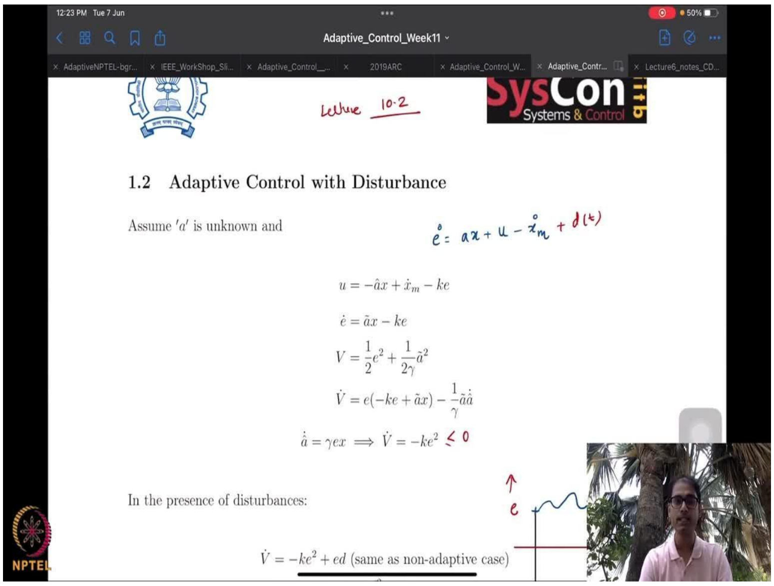
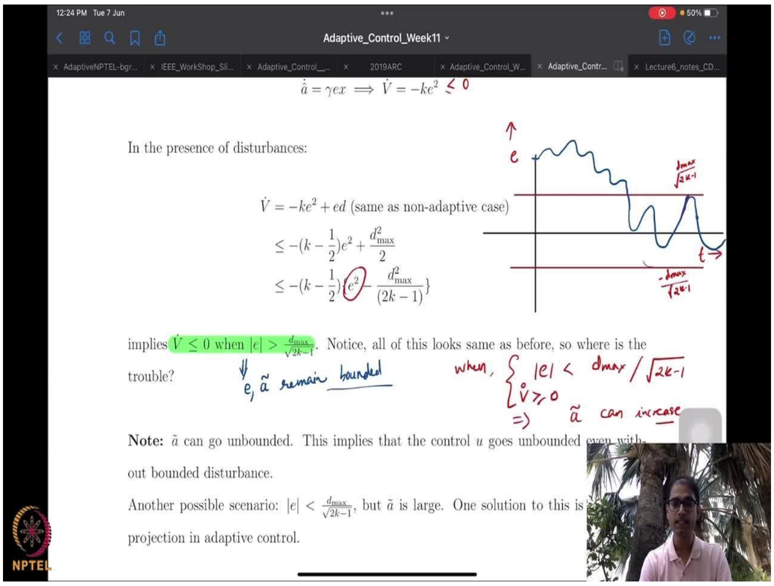
Nonlinear Adaptive Control
Professor Srikant Sukumar
Systems and Control
Indian Institute of Technology, Bombay
Week 10
Lecture No: 58
Parameter Projection in Adaptive Control (Part 2)

(Refer Slide Time: 0:18)

Hello everyone, welcome to yet another session of our NPTEL on nonlinear and adaptive control, I am Srikant Sukumar from Systems and Control, IIT Bombay. So, we are into the 10^{th} week of this course on nonlinear and adaptive control and I hope that all of you now have a very fair idea of how to design algorithms that will drive autonomous systems such as the Space X satellite orbiting the earth that you see in the background.

(Refer Slide Time: 1:02)

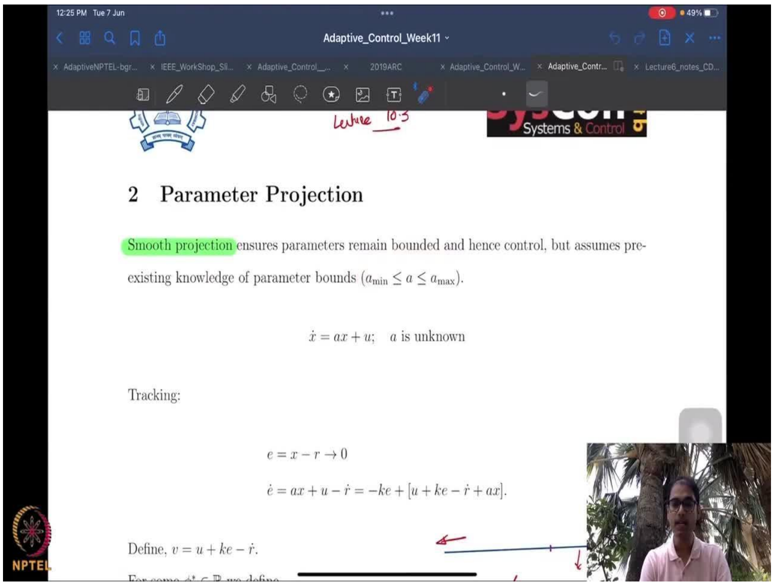




So, what we were doing in this previous session was basically look at adaptive control in the presence of disturbance and how it impacts the robustness of the overall closed loop system. And we realized that 1 of the big issues that occurs because of adaptive control is the fact that though your errors might remain bounded, under the action of an adaptive law, but what can happen in this is that your a tilde that is your parameter estimation error can very well go unbounded.

And now, this is a rather serious issue, because this a tilde that is a parameter estimation error going unbounded means, only 1 thing and that is that your parameter estimate a hat is going unbounded and since this enters the control law in a very prominent way, this also means that in order to maintain this kind of bound on e. We will be forced to have an unbounded controller, which is of course, something that is not acceptable in general, not in general, but in every situation because obviously, there is no way that you have unbounded controls available at your disposal.

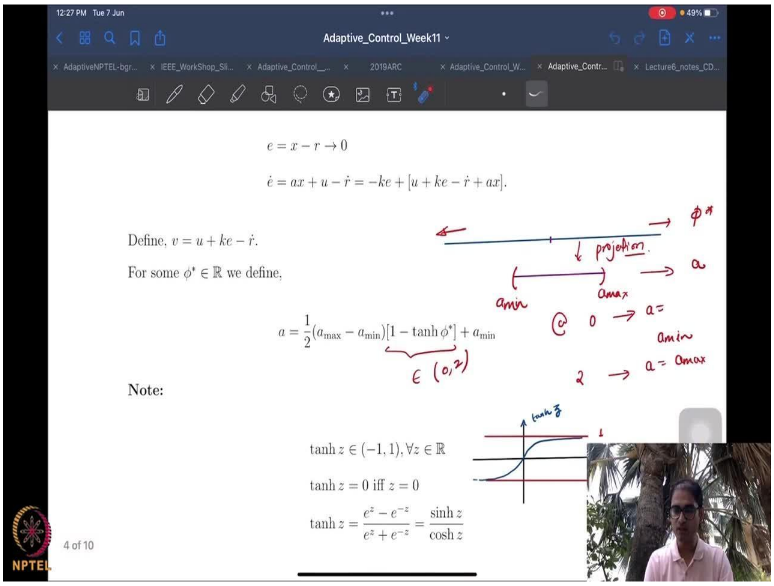
(Refer Slide Time: 3:01)



So, one of the ways that we wanted to solve this was using projection of the parameter estimate. So, the idea is if I do know, what the bounds on my parameter are, then I implement a projected adaptive controller, which means that you sort of ensure that your parameter update stays within these bounds, because of course since the true parameter value is within these bounds, it makes very little sense to, try to search for this true parameter value outside these bounds correct?

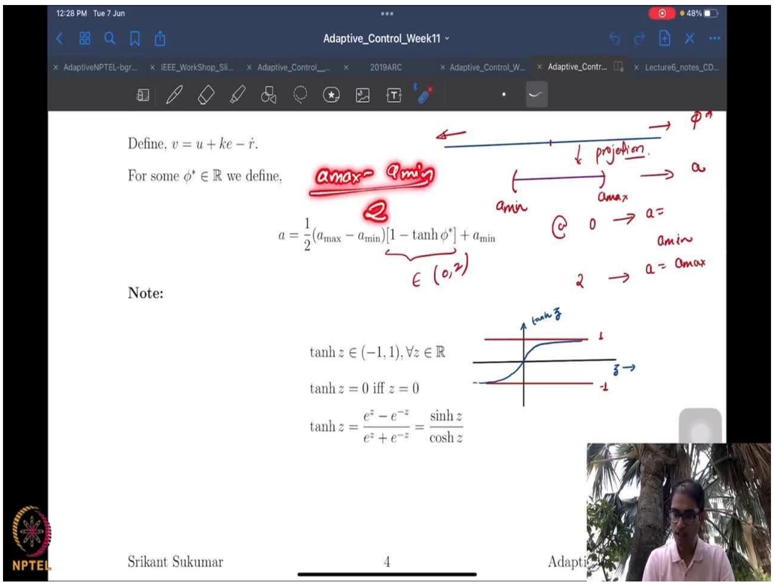
So, we were starting to look at projections. So, this is the direct most direct way to attack the problem, if you do know these bounds, then you can attack this problem in a very direct way that is, in the earlier situation, the lack of robustness is because of the parameter going unbounded parameter estimate going unbounded. So, now that is essentially what we do, we prevent the parameter estimate from going unbound. So, it is like a direct way to attack the issue.

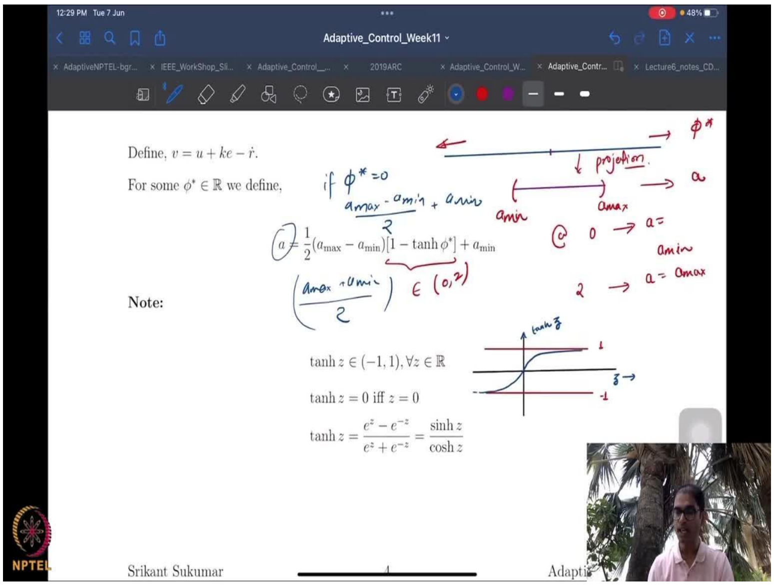
(Refer Slide Time: 4:04)



How did we do that? We essentially formulated a tracking results. So we also have a tracking error dynamics right as always, nothing very new here. And then we redefine the control as this is not a big deal because I mean, k e and r dot are both known. So obviously, I can implement u if I know v and vice versa.

(Refer Slide Time: 4:30)





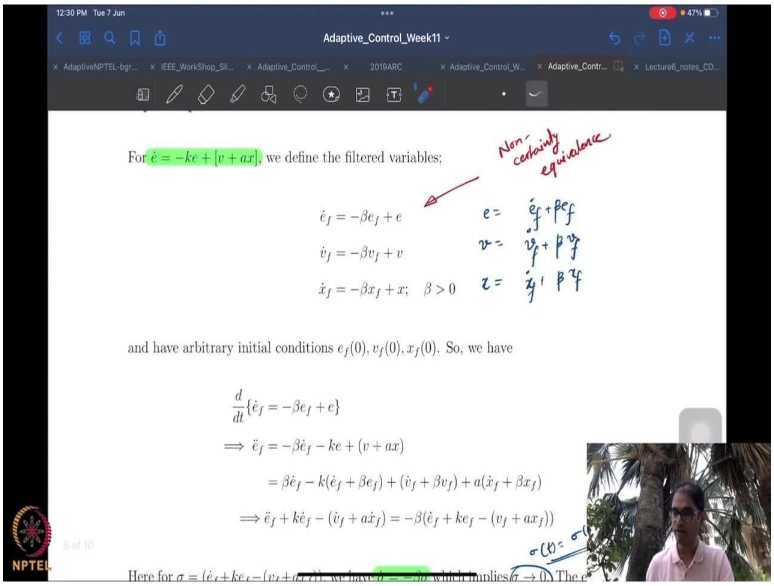
So what is the idea of doing projects? How do we do projection is that we change the parameter that we estimate, we, we start to estimate this parameter phi star instead of a, how do we do that we can write a in terms of this phi star using a tan hyperbolic function, which lies between minus 1 to 1 right? And because this function lies between minus 1 to 1 and it is 0 exactly when the argument is 0, that is phi star is 0.

So if you put 0 here, you will see that this outcome is, this is essentially, if you essentially try to see the bounds on this, you can see that this lies between 0 and 2, like this 1 minus tan hyperbolic lies between 0 and 2. And if you substitute 0 you get a min on the right hand side, we substitute 2 you get a max on the right hand side. So, this is the rather, nice property, that we are, in a sense looking and we will of course, look at how to make the error 0, and so on and so forth. And what is the right value of phi star and all, so obviously, you want to see what is the right correct value of phi star and all that.

I mean, the correct value of phi star may not be 0, obviously, as you can see, because that is not what we are looking for. In fact, if I put this equal to 0, then what I will get is, something like a max minus a min over 2 a max. So, let me write this out. If phi star is 0, I mean, this is not the ideal value, remember, I am just doing it for like an illustration, so if phi star is 0, then you will get something like a max minus a min over 2 plus a min. So this is going to be some value and this is going to be a max plus a min over 2.

So, and this may not necessarily be the right value of a, so phi star equal to 0 is obviously not the true value, phi star has some true value, which we do not know, what we are doing essentially is that we are moving the unknown from a being a to being phi star and this is what helps us and we can do this because, of course, the tan hyperbolic function which is available to us. Just give me a moment.

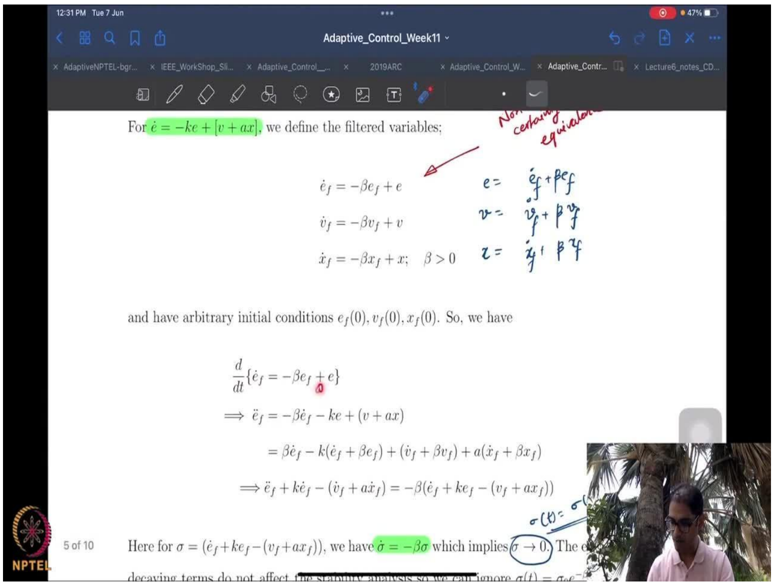
(Refer Slide Time: 07:33)



So, how did we go about this? We started to define what is called filtered variables? I mean, and they essentially are true to its name, it is just filtered variable means that I define a filter on everything that appears on the right hand side. So, the right hand side is simply, I am sorry, the right hand side is written here. It is e dot is minus ke plus v plus ax.

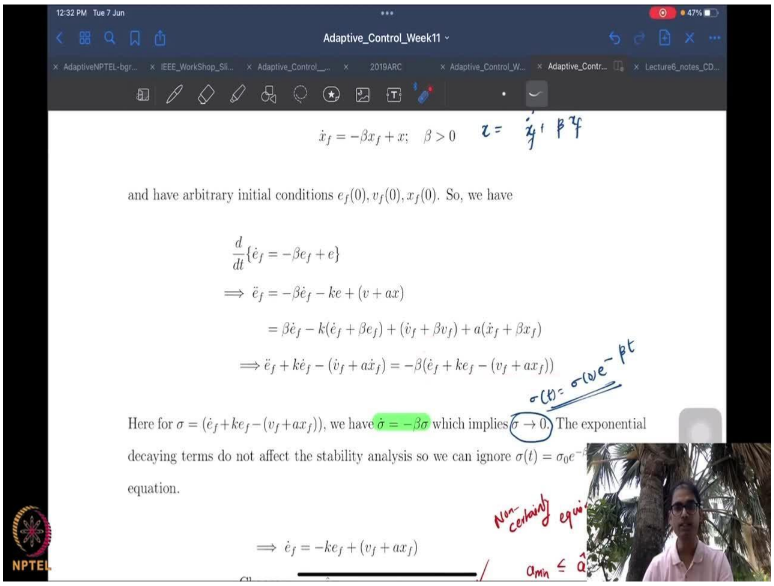
And so whatever quantities I know, on the right hand side, I define filters for that. So I know a, I know v, I know x. I do not define any filter for a, because does not make sense. I do not know a. Any filter that we define in typical adaptive control setting would need to be implementable. So therefore, we define these filters, which is e f dot is minus beta v f v f dot is minus beta v f plus v and x f dot is minus beta x f plus x.

(Refer Slide Time: 8:31)



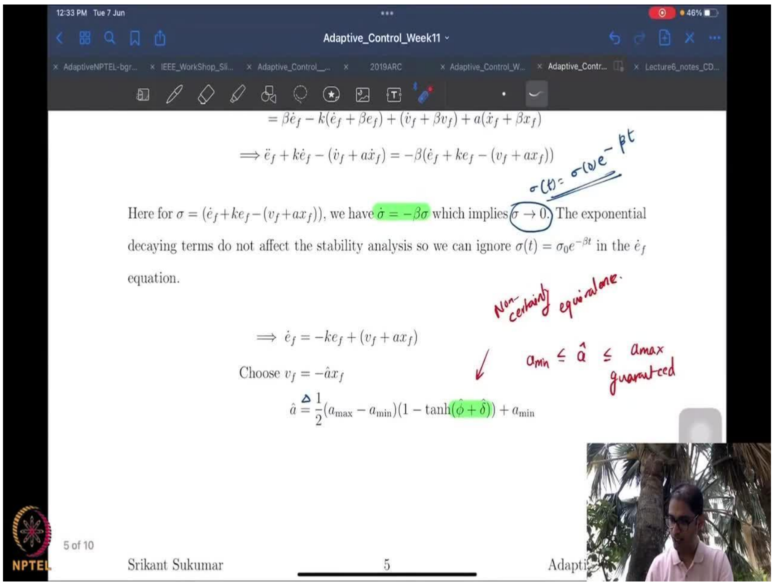
So the important thing to remember is that we defined all the filters with the same gain, beta, this is what we need to remember. Of course, we have arbitrary initial conditions, initial conditions are not very important. And now what did we want to do? We wanted to define the dynamics in terms of our filtered variables. How did we do that? We simply took a derivative of this this guy, because that brings in e dot and e dot can be substituted, frankly, that is simply the idea.

(Refer Slide Time: 9:12)



So that is what I do, I take derivative of both sides of this equation right here. And then I get e f double dot and then I get a minus beta e f dot and e dot and that e dot get substituted from here which is this guy. Now, because I want the dynamics just in terms of filtered variables, I substitute for e from here, that is e f dot plus beta e f, v as v f dot plus beta v f, x as x f dot plus beta x f. So just substituting everything in terms of my filtered variables, wherever I have written.

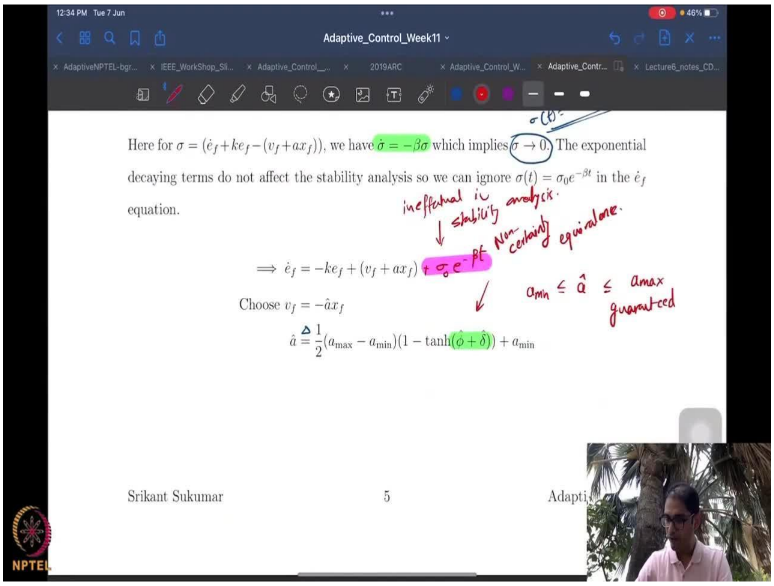
(Refer Slide Time: 9:49)



And then I separate out terms with beta on one side, and without beta on the other side. And if I define this quantity, that is multiplying beta on the right hand side as sigma right here. It is very easy to see that the equation that I have now is sigma dot equals minus beta times sigma. So, now, what do we do?

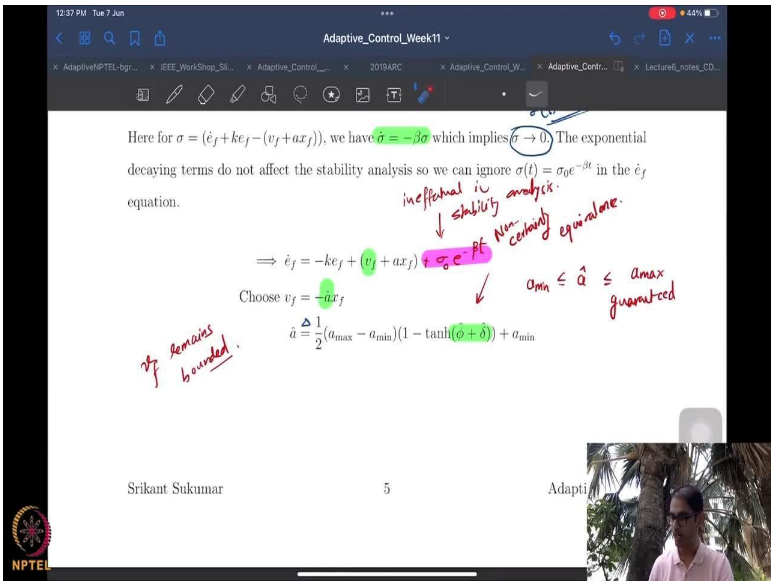
So, because we know that this is an exponential decay, that is sigma is equal to some sigma 0 e to the power minus beta t, what we can say is that this is pretty much saying that instead of looking at this exponential decay term, which is well known in stability analysis to be not have any impact, we simply and we directly put sigma equal to 0 and if you put sigma equal to 0 this is the dynamics that we get for e f dot.

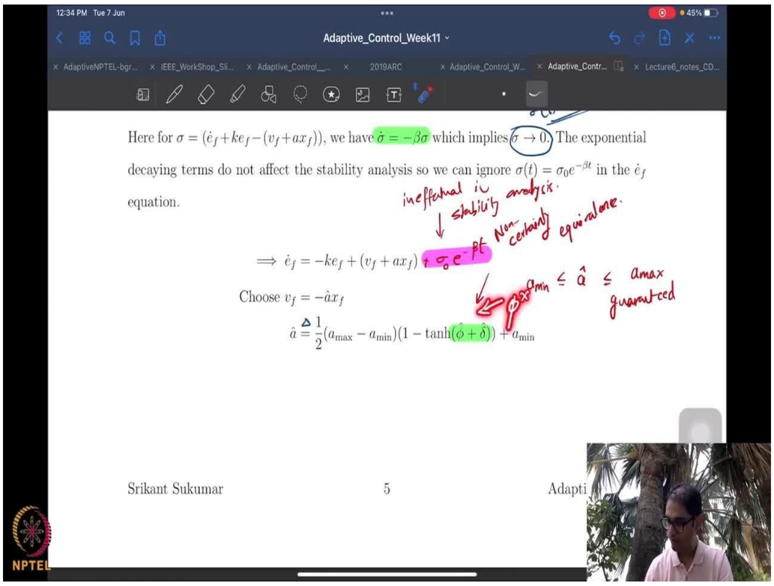
(Refer Slide Time: 11:01)



So, of course, I could be very, very precise and say that this is actually plus sigma 0 minus beta t. So, basically sigma is not 0, but sigma is exponentially decaying, but this term like I said does not affect the stability analysis. So, ineffectual in stability analysis, this is ineffectual instability analysis, and therefore, we can ignore it and we choose to do that, we choose to do that.

(Refer Slide Time: 11:40)





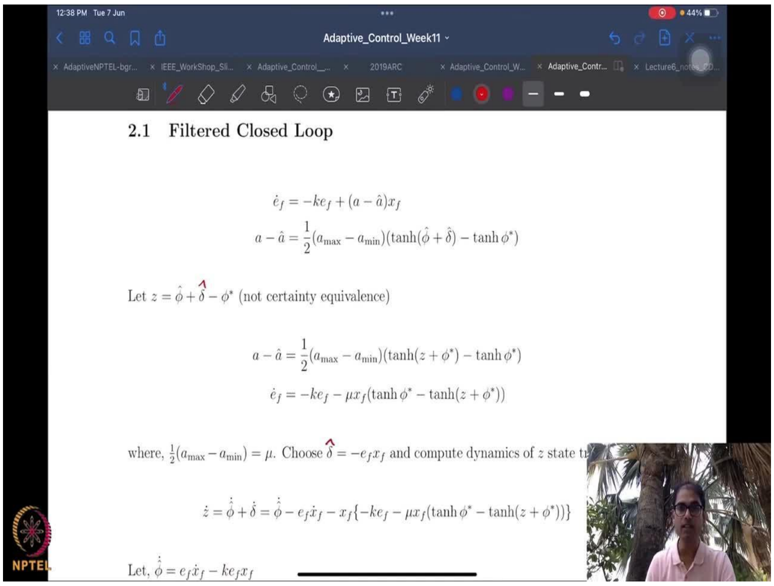
So, let us see so then what do we do? We define we implement what is called a non certainty equivalence type of adaptation law. So, we just like a, we define an a hat, a was defined with a phi star here, remember. So, but we define a hat with a phi hat plus delta hat. So, there are 2 terms, not just a phi star or phi hat corresponding to phi which would have been certainty equivalence but a non certainty equivalence so, we have a phi hat and add it to it, we have a delta hat. So, we have two terms and we will see how we use each of these terms.

So, because of this sort of using a and phi and phi hat delta hat instead of adapting for phi, phi hat and so on. We do not have to worry about a remaining bounded, because our control if you notice our control is now in v f, of course, given terms of the filtered variables, we will all of course, take it back to the original variable, not a big problem, easy to implement.

So, the control is now containing a hat. And it is not difficult to see that this a hat is guaranteed to remain within this nice bound, a hat is guaranteed to remain within this nice bound. So there is no scope of v f becoming unbounded. Because we will of course, prove that we of course prove that x f will also be bounded. So there is no scope for v f be unbounded. So v f remains bounded.

So the other thing to remember is because we have defined a filter and v is defined via v f and beta v f. And v is what we really implement on the system, we have to also claim that v is bounded. And that is not difficult either, because this is a stable system. So, basically, v f is bounded, the derivative is bounded. So v, which is the sum of v f dot and beta v f is also going to be bounded. So boundedness of the control is guaranteed. And that is essentially what was the key issue in standard adaptive control and which we can avoid now.

(Refer Slide Time: 14:27)



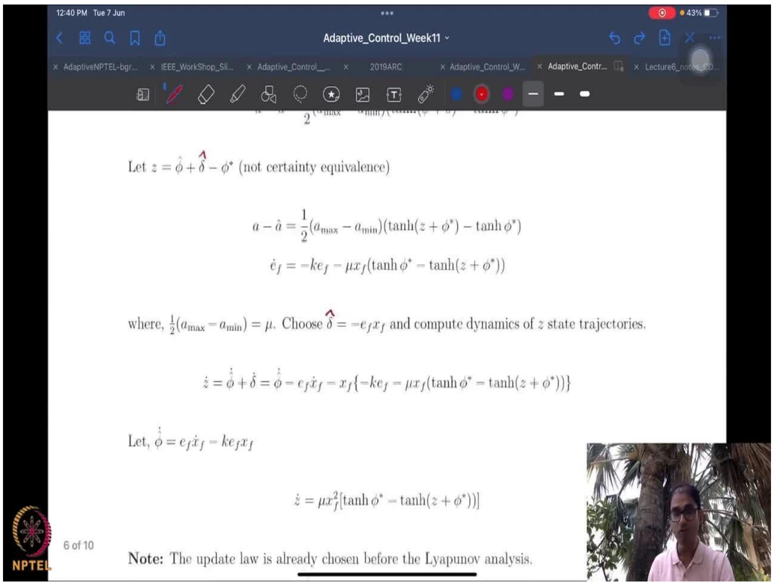
So, I am sorry, seemed like a long introduction, but I sort of re explained a few things. So this is where our lecture 10.4 begins, technically. But again, the explanations that I gave for the earlier material were rather important. And I hope you can keep those in mind.

So what does the filtered closed loop system look like? It looks essentially like this, it looks like this I mean, the expression looks simple, but in reality, this a and a hat are complicated. So we will of course write it out. So a minus a hat is actually this expression. You notice that the a min cancels out. And you are left with this guy. So not a big deal, pretty straightforward.

Now, we define an a variable, sort of parameter error, if you may as z. We use a different notation here and not the tilde notation, because this is not certainty equivalence. That is it.

And the z is defined as phi hat plus delta, actually, because we are using hats everywhere, this should be a hat. So this is actually phi hat plus a delta hat minus phi star. So this is what is your parameter error if you may.

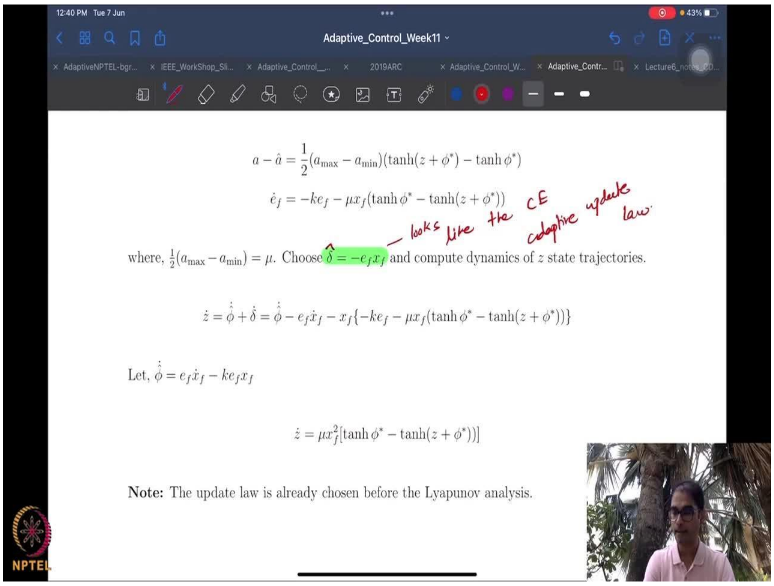
(Refer Slide Time: 16:01)

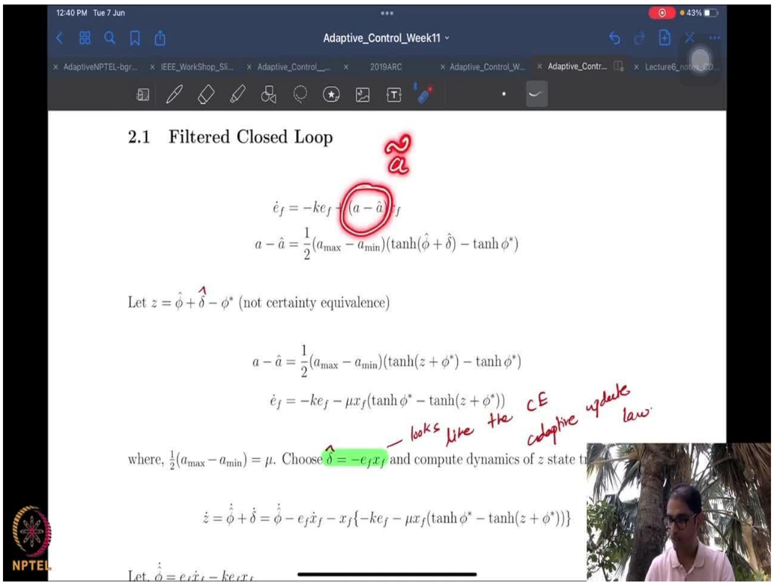


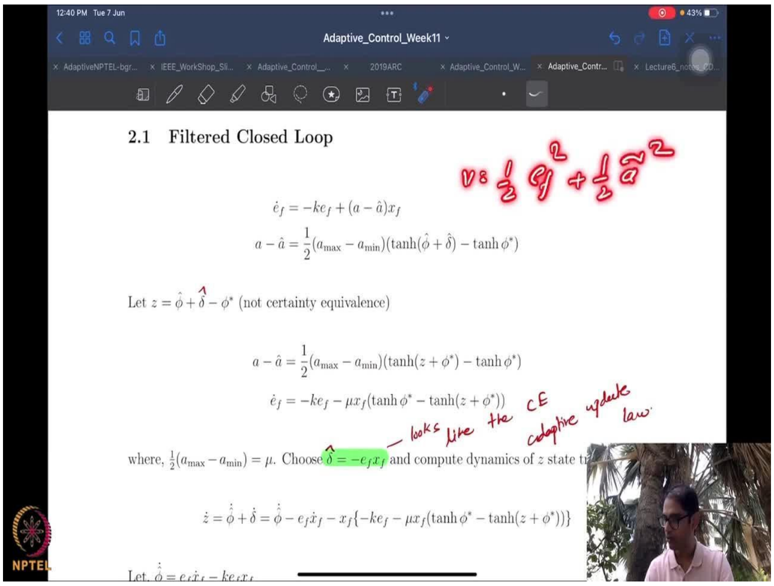
And now we want to write this term a, I am sorry, we want to write this using the z. So of course, we write phi hat plus delta hat as z plus phi star minus a tan hyperbolic phi star. And so of course, we substitute this quantity right here, we get e f dot is minus k e f plus this whole thing. And where what have we done, we have simply use this shortening notation, a shorthand notation for this half a max minus a min, and call it mu. And so what we have is minus k e f minus mu x f times this guy. We have just flipped the sign. That is not a big deal.

Now, what do we do? We have to choose 2 things, one is delta, how delta hat is obtained in one, and the other is how phi hat is obtained. So delta hat is not given any dynamics, it is simply defined as minus e f x f, it is minus e f x f.

(Refer Slide Time: 17:19)

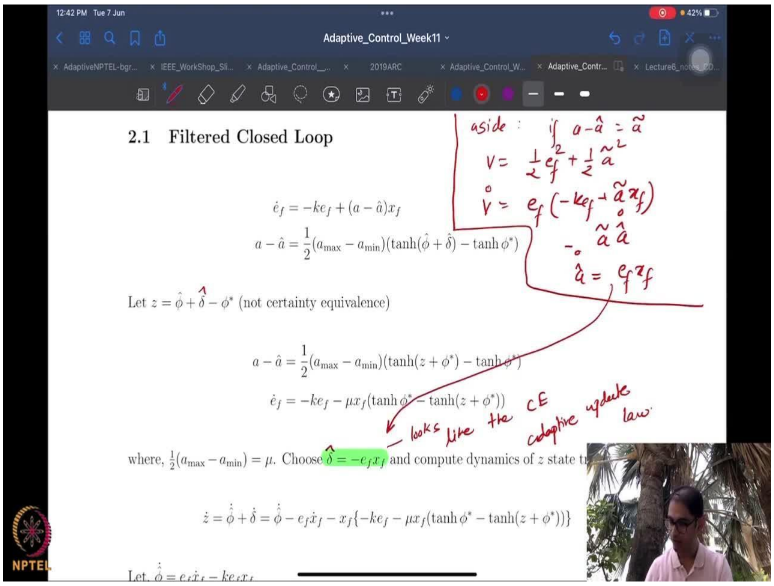






Interesting, I mean, if you do wish to do this, if you did want to do this essentially the delta hat expression looks like the certainty equivalence. Yeah, adaptive update law. This looks like the certainty equivalence adaptive update law, why because if you took the filtered system, which is this guy and you and you thought of this say, thought of this is say, a tilde and you took your V here half e f squared, plus half a tilde squared. You will get your a tilde. I mean, in fact, it is a very quick calculation, I can simply do this.

(Refer Slide Time: 18:12)

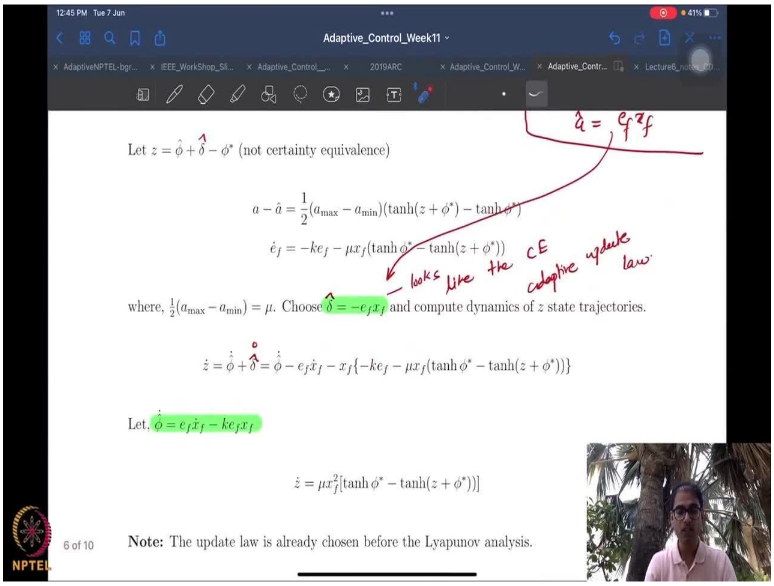


This is an aside. If a minus a hat equal a tilde, and take my V as half e f square plus half. Well, actually, I do not need a gamma. So I take half a tilde square. So V dot will be what e f e f dot which is minus k e f plus a tilde x f minus a tilde a hat dot. So if you wanted to choose a hat dot here, what would you do? a had dot will be? e f times x f, a hat dot would be e f times x f.

So let me see if there is only a sign issue here that we might need to resolve. But I think the sign issue is because we are defining things the other way. So, the expression is exactly the same. If you see, you get e f x f, and you get e f x f and the sign issue is just because of how you choose a minus a hat and all that.

Let us not worry too much about that. But the expression for delta hat pretty much is motivated by this. So interesting for you to observe I hope so, I hope you can sort of connect to this, connect to this.

(Refer Slide Time: 19:51)



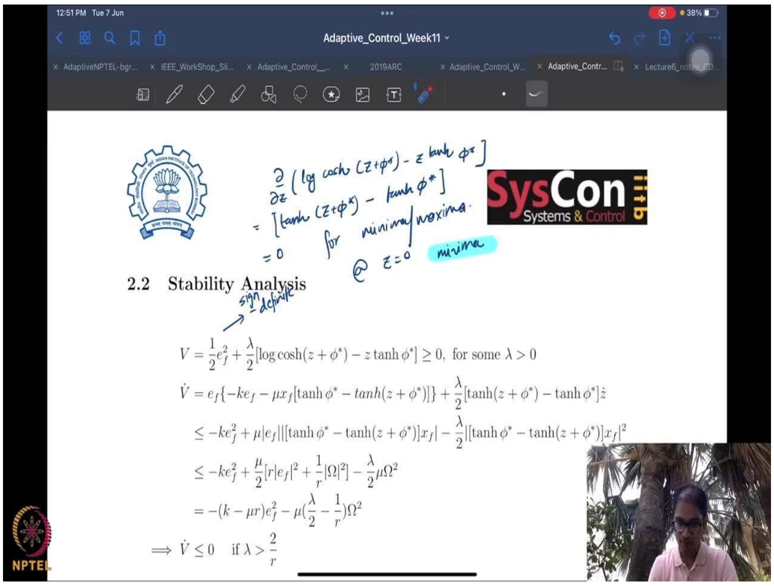
Now, what do we do we start to we actually do not use a Lyapunov function in this case to get a phi hat dot. At least but not yet, we do not go to the Lyapunov function at all. So, we directly compute a z dot and z dot is just phi hat dot plus delta hat dot phi star is a constant of course, unknown but still a constant. So phi hat dot remains as it is and then you compute delta hat dot sorry, compute delta hat dot. So phi hat dot is as it is and then delta hat is minus e f x f dot minus x f e f dot, which is this whole thing.

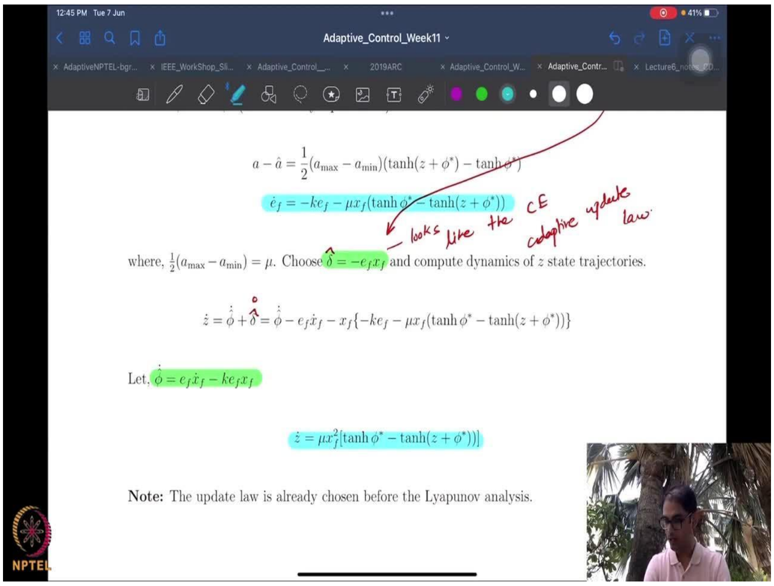
Now, what do we do we take phi hat dot so, as to cancel everything we can cancel. So notice I can cancel this guy, I can cancel this guy. And that is it. I cannot cancel either of these terms. So I choose phi hat dot to cancel this term with this and this term with this guy. And so what am I left with? I am left with a z dot, which is this term, so this is the important thing to remember. The update law is already chosen before the Lyapunov analysis is already chosen before the Lyapunov analysis.

So, that is important to remember. So that was the there are 2 pieces in the update in trying to find the parameter, there is the delta hat and there is the phi hat, so the delta hat is this and phi hat is and delta hat is given by a static term and the phi hat is an update with this sort of an expression. Important to remember, unlike the certainty equivalence method, which we were doing, until now, the phi hat or the delta hat, to be honest, are chosen using a Lyapunov function they just chosen separately just intuitively.

I mean, we are choosing a delta hat comes from is motivated by the CE adaptive update law, as we just saw, and phi hat is obtained by trying to cancel whatever we can cancel in this z dot.

(Refer Slide Time: 22:15)





Now, we can or we can sort of move on to the stability analysis. So we have now are 2 pieces of the dynamics. We have the let us see, we have the e f dot which is written in this form and we have the z dot, so these two are the critical ones, that I will mark them with a different color. So this is the e f dot and this is the z dot equal. So these are the two equations that are required anyway, because, we have essentially these 2 variables e f and z, x f and all are of course functions of v. So we do not have to worry about that right now.

So we want to do a stability analysis of this closed loop system. So let us see, how we do it. I mean, I will do a few steps first. The first thing to look at is the sort of the Lyapunov function we chose. The Lyapunov candidate is a rather interesting one, it is of this form half e f square, which is just the normal quadratic in the first element, but in the second term, this is rather interesting there is a lambda positive lambda divided by 2 times a log cosh log cosine hyperbolic z plus phi star minus z tan hyperbolic phi star.

Now, what we want to of course, we want to know and claim and we anyway want to do typically Barbalat's Lemma type analysis. So we are more than happy for V to be semi definite positive. So we know that the first term is of course nice. So no problem first term is, it is sign definite, so no doubt, but what about the second term?

So this requires a little bit of careful analysis. So what we will do is, let us take partial of log cosine hyperbolic z plus phi star minus z tan hyperbolic phi star. So how do I take the partial, just whatever take the derivative. So this will be, there is only one variable here. So therefore, we have to take partial with respect to only this one variable. And so this is because this essentially becomes the derivative of this is 1 over cos hyperbolic times sin hyperbolic works exactly like the standard logarithm.

So, this is tan hyperbolic, z plus phi star minus, if I take partial with respect to z, just tan hyperbolic phi star. And so this is what we get as the partial with respect to z. And of course, you want to note a few things. The first thing to note is that this term looks very similar to the term here. In fact, just, it is just the flip sign version of this term, you have tan hyperbolic phi star minus tan hyperbolic z plus phi star. And I have taken the Lyapunov candidate in a smart way so that it is partial, actually has the same term. Obviously, this will help us in the analysis.

But the other important thing to see is that this term is always greater than equal to 0. Again, this term is always positive, non negative. Why we say that is because, again, this is something you can verify. And it is something you can verify. So, this is because you have a phi star here and it basically when z is equal to 0, these 2 are the same so it is equal to 0, for any other value of z, this is non negative, that is the whole premise on which you sort of say that.

So in fact, let me see, let me think about this a little bit carefully. A little bit more carefully so basically, what I am trying to do is to take this V function and compute a sort of minima for this function that is the idea. And we claim that so how do you compute a minima, you take partial with respect to the variables and equate them to 0, and then whatever you get is the minimum.

So in fact, let us see, because I took the partial, I want to equate it to 0, I apologize, I did not complete the steps. So I am taking the 2 terms separately, because you can see that decoupled there is only e f here and there is only z here. But I can, do them separately decoupled them not a big deal. So that it is obvious that the this term is minimum at e f equal to 0, because I can take partial and equate it to 0.

Now to find the minimum of this term, I take the partial I took the partial here, and I will equate it to 0 equate to 0, for minima, maxima of course, it can be maxima or minima. So when this equal to a minima maxima? It is evident that at z equal to 0. And I claim it is, in fact, the minimum, I am not going to do this computation. You can check it out on your own. But it turns out that this has a minimum at z equal to 0.

And what is the minimum value? It is exactly what is the minimum value? It is exactly going to be 0. It is exactly going to be 0, so that is what you have to verify. So you will have if you put z equal to 0 here at z equal to 0, this will be log cosh this is just log of cos hyperbolic of phi star I believe, sorry, the minimum is not the minimum is not minimum value is not 0. The minimum value is not 0. It is a positive quantity.

So it is a positive quantity, so the minimum is not 0. I apologize. That is my mistake. It is a positive quantity. It is exactly in the log cos hyperbolic in fact is lambda times log cos hyperbolic phi star. If I plug in z equal to 0, z equal to 0 here and z equal to 0 basically gets rid of this term. I think that this is okay.

So the important thing to remember is that minima of this function is strictly positive. Again, I am not verified, it is a minima, you have to take a second derivative and verify but it is easily verified that this is in fact a minimum. So, minimum value I was thinking would be 0, but it is not it is actually log cos hyperbolic phi star. So, z goes z is 0, because if I check here carefully, you have if I equate this to 0, then tan hyperbolic z plus phi star is equal to tan hyperbolic phi star only when z is equal to 0.

So, what did we look at today, we have sort of proceeded further towards this proof of this projection based adaptive control. We are yet to complete the proof of course, but we are we have done the filter design and we have also, sort of given the update laws, which is the phi hat dot and also there is a delta hat in this non certainty equivalence type design. In the subsequent session, we will be able to complete the stability proof for this set of system and we will also try to understand what the implications of such a this kind of an adaptive controller is. Thank you. And I will see you again soon.