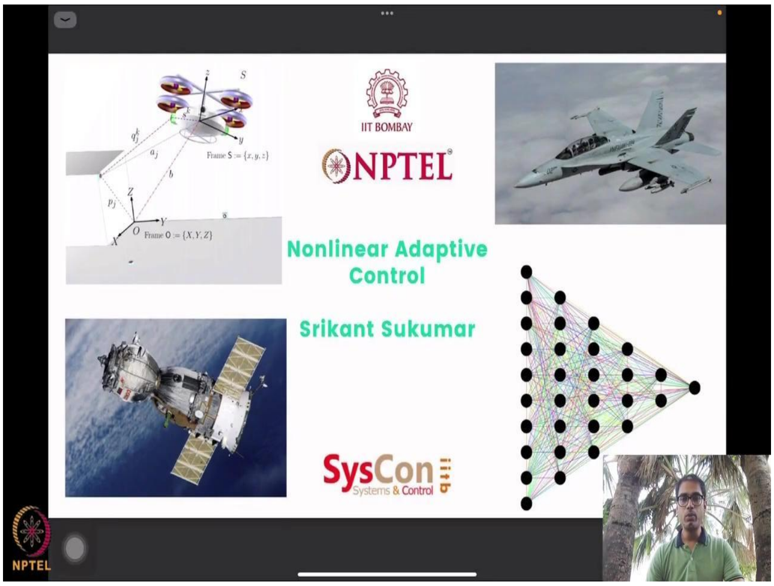
Nonlinear Adaptive Control
Professor Srikant Sukumar
Systems and Control
Indian Institute of Technology, Bombay
Week 11
Lecture No: 65
Initial Excitation in Adaptive Control (Part 5)

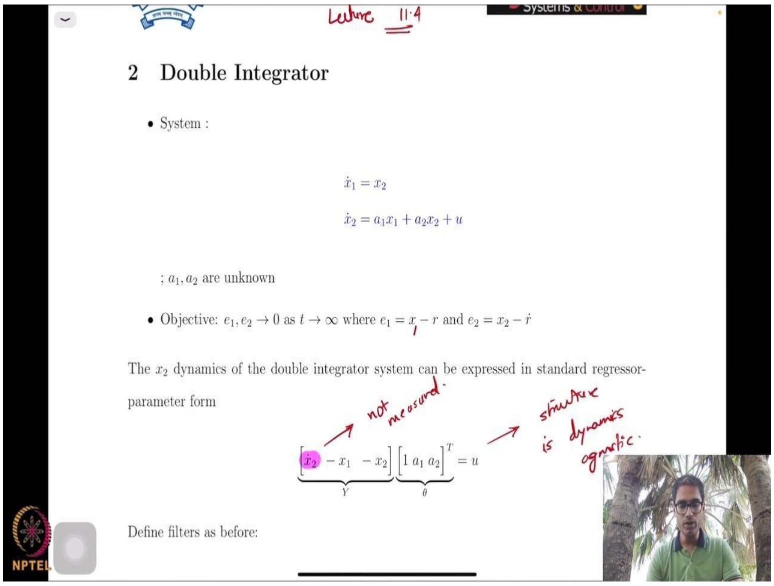
(Refer Slide Time: 0:16)



Hello everyone, welcome to yet another session of our NPTEL on nonlinear and adaptive control. We are sort of in the last lecs of this course. And we are close to the end of week number 11. And we sort of hope that whatever we have learned in this course will help you to design algorithms for systems such as what you see in the background. So, what we were doing until this last time was talk about start to discuss a double integrator problem using the initial excitation method.

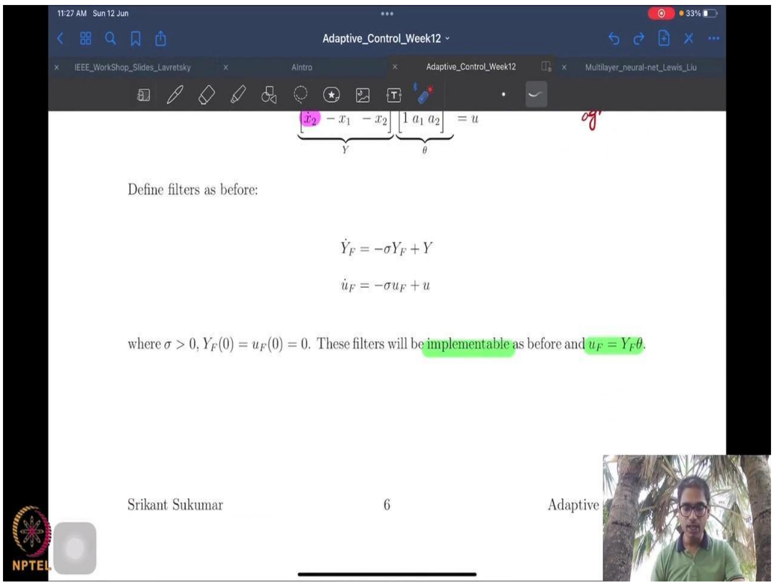
Now, this method helps us to avoid the need for persistent excitation in parameter learning. Now, remember that learning is one of the key sort of paradigms or requirements in neural network based adaptive controllers. Some what you see in this in this picture also so, other systems that you see here, like drones, and aircraft and space crafts are real applications of adaptive control and have in fact being tried at a large scale research level and at some implementation level also.

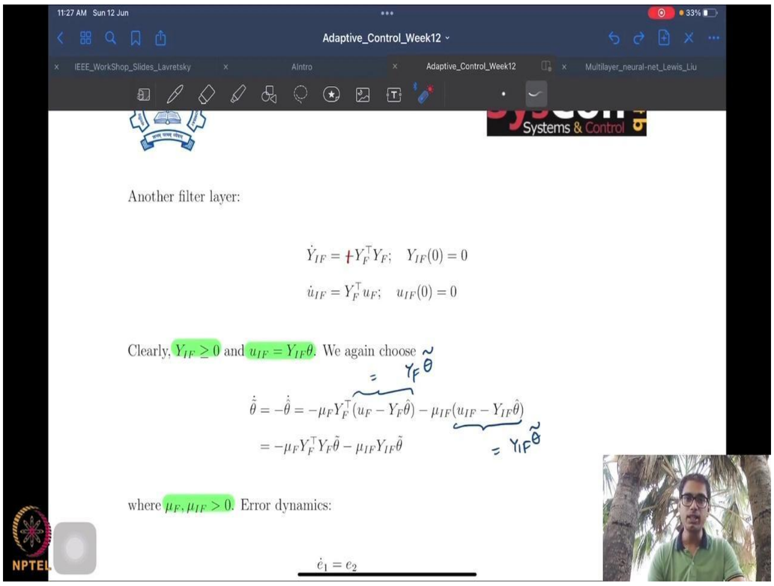
(Refer Slide Time: 1:53)

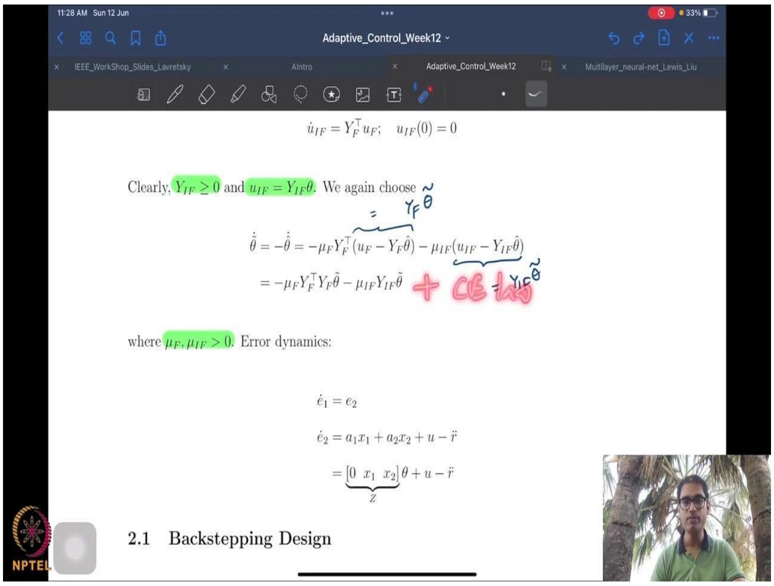


Now, in this double integrated type system, what we sort of emphasize or we were starting to emphasize is that the construction of your adaptive law does not get impacted significant. And the reason for this is that this regressor parameter structure that we use, in order to design our update law is agnostic to the dynamics of the system, it does not matter what is the order, it does not matter what kind of dynamical system it is as long as you can write the system in a regressor parameter form that is Y theta equal to u, if you can do that, then you can sort of design your adaptive controller for any system. In fact, even if there were nonlinearities, here, these nonlinearities would just get inside the Y equation and that it, it will not really effect how you design your adaptive update law.

(Refer Slide Time: 2:58)







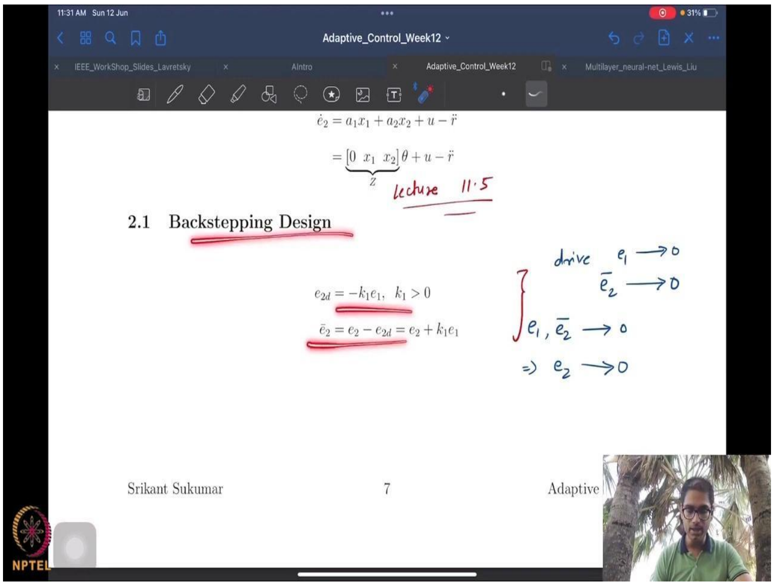
And so, of course, we designed these 2 layer filters, YF uF and YIF uIF and the structures are kept in such a way that the regressor parameter equation structure same remains identical for the first layer and the second layer filter also. And so, this is a rather important feature, which is what helps in subsequent analysis, and we already seen this kind of analysis for the single integrated system.

So, adaptive law is of course chosen again in the standard way it is, and you can theta tilde dot is nice your non-positive terms that is your minus mu F Y F transpose Y F theta tilde and minus mu IF Y IF theta tilde, so, we know for sure for a fact that Y IF is of course, positive semi definite. So, is Y F transpose Y F but you also know that in the presence of initial excitation, Y IF is positive definite. So, therefore, this becomes a legitimate negative term.

So, what we are not doing at this stage is that we have not introduced the certainty equivalence adaptive law yet, but, we can talk about that, but again, because you have seen it for the single integrator, it should be possible for you to implement also in the double integrator. So, we will look at that later as the need arises. But for now, we are just looking at the standard initial excitation based adaptive law.

So, we construct the error dynamics of course, using e1 and e2, and the dynamics turns out to be e1 dot is e2 and e2 dot z theta plus u minus r double dot great.

(Refer Slide Time: 4:55)



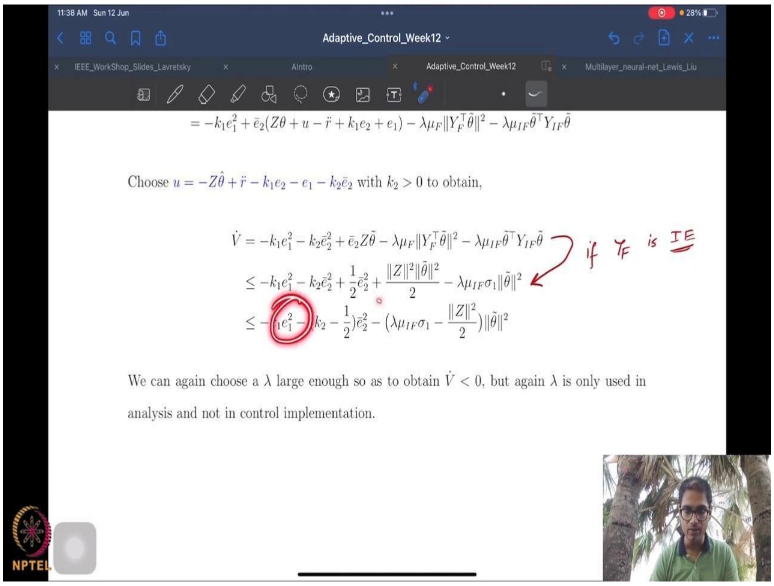
So, now we know we start our standard Backstepping Design, fine. How do we do it? We look at the first subsystem. Let me mark the lecture first. So, we are on to lecture 11.5. we are on lecture 11.5. So, how do we do the backstepping design, we start by looking at the first sub system and assume that e2 is a controller, if e2 was the control, then what is the good control to drive e1 to 0, simply minus k1 e1 for some positive gain k1. So, that would be your desired value of e2.

Now, it is evident to us that e2 cannot just be made to follow a desired value or cannot just be equal to desired value. But we can make it track the desired value. And that is how we define the backstepping error, which is e 2 bar which is e2 minus e2 desired, and that becomes e2 plus k1 e1 in this case, and again, we have seen this backstepping design even before. So, this should not be something new for you, just a couple of weeks old this material. So, I mean, I hope that this is not coming to you as any kind of surprise if it is, then I would strongly encourage you to go back and look at our backstepping design lectures.

So, a new system is essentially going to be written in terms of e1 and e2 bar. Where e1 is of course, your usual error variable and e2 bar is the backstepping error variable. So, of course, what we try to do is drive e1 and e2 to 0. So, our aim will now be sorry to drive e1 and e2 bar to 0. And remember that if e2 bar goes to 0 implies e2 goes to 0, why because e1, if you look e2 bar, and if e2 bar is going to 0, k1 e1 is already going to 0 by the previous fact that e1 is going to 0. So, we are left with the fact that e2 is also going to 0.

So, this was already established. So, making the backstepping error variable go to 0 is equal into actually making the original variable also track 0. So, this is what we want, this is what is the backstepping design.

(Refer Slide Time: 7:40)



Now, we of course want to do the analysis. So, we do the control design using Lyapunov function, the candidate Lyapunov function in this case, because it is not evident without doing this analysis as to what the control should. So, we as usual choose our this is our backstepping works, the first piece of the Lyapunov candidate is e1 squared by 2 from the first piece of the dynamics. And the second term is just the quadratic in the backstepping error.

And the third term now is our usual initial excitation based sort of term. So, there we introduced the lambda. And then as the theta tilde norms square, remember that e1 and e2 are scalars, but theta tilde has three terms so it's a r 3 its a vector in r 3. So, therefore there is a norm here.

Now we of course, expand this I get e1 e1 dot and e1 dot is, just e2 which I am writing in terms of the new variables e2 is just e2 bar minus k1 e1 and then e2 bar e2 bar dot is just e2 dot plus k1 e1 dot is this one and then of course, I am left with this nice set of terms from my parameter update law. Now what I have this nice negative term from e1 which I keep, then I have this term e1 e2 bar which I club with this guy. That is what I do that the nice term club, this guy here, So, I get z theta plus u minus r double dot plus k1 e2 as before, then I get this additional term e1.

So this term is of course coming from here. And these terms are of course, same as before, not made any change here, except for writing this term as the norm squared term that is it, that is the only difference. Now what will be the logical value of the control? As always try to cancel all the funny looking terms and introduce a good term. This is the basic logic. So, what are the five terms? I can completely cancel this. So, I introduced an update a parameter estimates here. So, I tried to cancel it with my estimate, I can cancel this. So, I do, I can cancel these 2. So, I do, and then I introduce a nice negative term with a positive gain K 2.

And once I do this, here, I get V dot as minus k1 e1 squared minus k2 e2 bar squared. And since I could not completely cancel this term, I am left with this theta tilde term. I hope that is sort of evident. Now, we, of course, have this 2 nice terms here, which is what we try to use to dominate. So, what we do is, we start to use our usual sum of squares method to split this into 2 pieces, this is getting split into this guy and this guy. Using a b is less than equal to a squared plus b squared by 2.

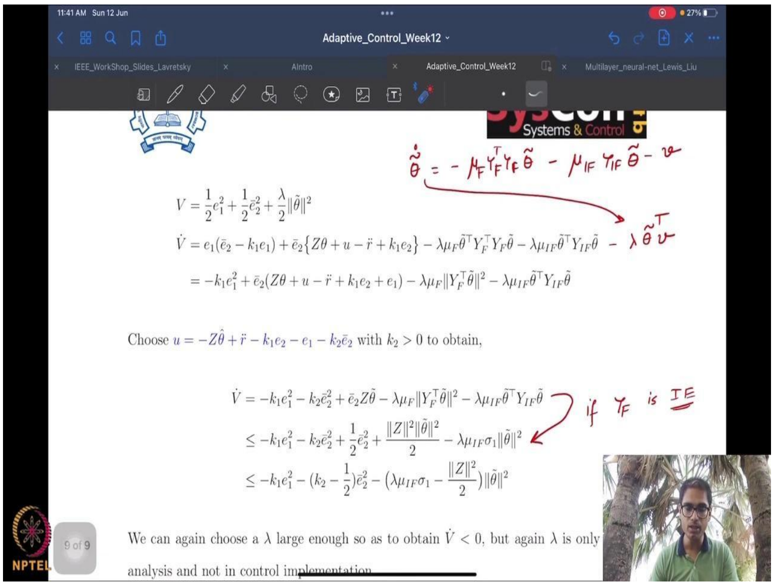
And now it is evident that this can be combined with this term, I do that and this term can be combined with this term. So, now notice that as usual, when I came from here to here, I ignored this. I remove this term, because this is not definite. Its atmost negative semi definite and so, it is not very clear if I can use it to dominate a term. So, therefore, it is not really useful for me in the Lyapunov analysis. Important thing to remember is that it does not harm my analysis, therefore, I can drop this term.

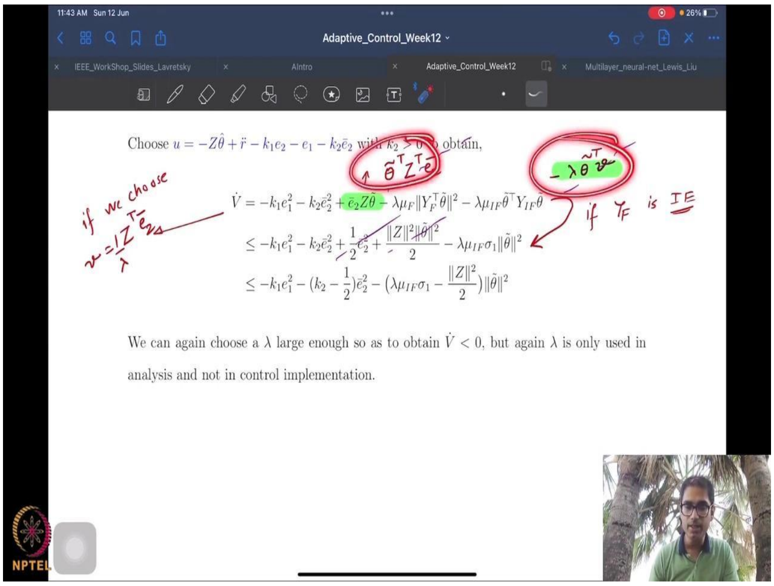
And so, this is because I dropped the negative semi definite term and guaranteed that this is less than equal to this term after dropping. So, if I now look at what is left with me, I know that this term gets combined with this guy give me that and this term gets combined with this guy to give me that. Now, as usual, there is this question of choosing lambda. So, remember that this mu IF and mu F and all these things are usually fixed beforehand. At this stage, I am only left with the ability to choose lambda sigma 1 I is also sort of is sort of chosen.

And So, I will say that, let us see, I can go from here to here if Y F is niche in IE. Remember, I cannot go from because I replace Y IF by the sigma 1 identity. And I can do this only if Y F was initially exciting. And so, this is where I have used the assumption on initial excitation, again, similar to before. And now because this Z is a function of the state, also, we can choose but it is okay. It is complicated, and not completely straightforward to say that this is the particular value of lambda, that will work but the important thing for us is the existence of such a large lambda. I just have be choose lambda large enough.

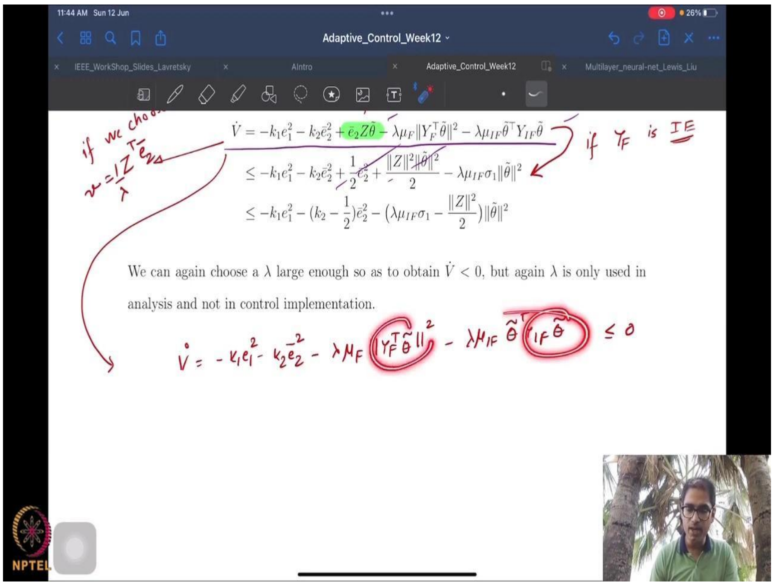
And if I do choose lambda large enough, I am fine. And most importantly, I get these nice negative terms here. And of course, in the presence of initial excitation, I can show convergence of e1 e2 bar and theta tilde. So, in the presence of initial excitation, everything is great. No problem just like before, I have convergence of all 3 parameters, I only require initial excitation and not persistent excitation. So, my performance is really nice. I can dominate this term with a large lambda, a lambda, which is not going into my control implementation, so, it does not matter. It does not matter to them.

(Refer Slide Time: 14:30)

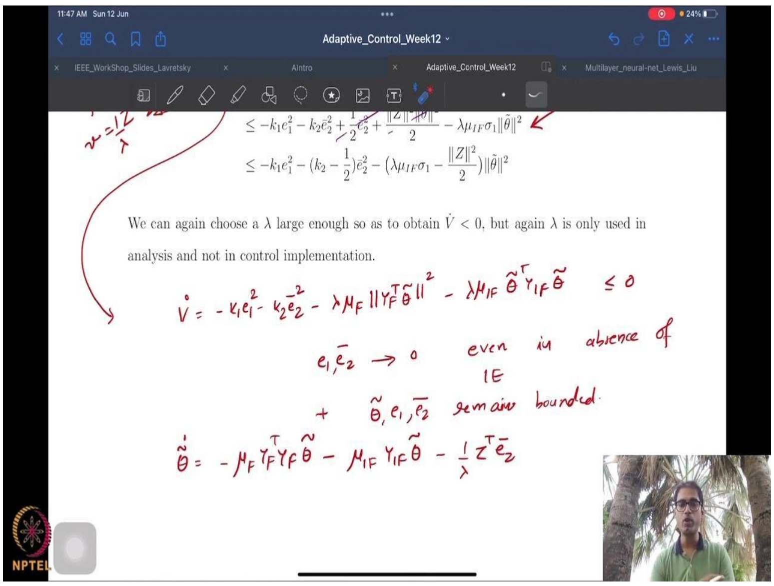




11:43 AM Sun 12 Jun
Adaptive_Control_Week12
IEEE_WorkShop_Slides_Lavretsky
AIntro
Adaptive_Control_Week12
Multilayer_neural-net_Lewis_Liu
Choose $u = -Z\hat{\theta} + \ddot{r} - k_1 e_2 - e_1 - k_2 \bar{e}_2$ with $k_2 > 0$ to obtain,
$\tilde{\theta}^T Z^T \bar{e}$
$-\lambda \tilde{\theta}^T v$
if we choose $v = \frac{1}{\lambda} Z^T \bar{e}_2$
$\dot{V} = -k_1 e_1^2 - k_2 \bar{e}_2^2 + \bar{e}_2 Z \tilde{\theta} - \lambda \mu_F \|Y_F^\top \tilde{\theta}\|^2 - \lambda \mu_{IF} \tilde{\theta}^\top Y_{IF} \tilde{\theta}$
if Y_F is IE
$\leq -k_1 e_1^2 - k_2 \bar{e}_2^2 + \frac{1}{2}\bar{e}_2^2 + \frac{\|Z\|^2 \|\tilde{\theta}\|^2}{2} - \lambda \mu_{IF} \sigma_1 \|\tilde{\theta}\|^2$
$\leq -k_1 e_1^2 - (k_2 - \frac{1}{2}) \bar{e}_2^2 - (\lambda \mu_{IF} \sigma_1 - \frac{\|Z\|^2}{2}) \|\tilde{\theta}\|^2$
We can again choose a λ large enough so as to obtain $\dot{V} < 0$, but again λ is only used in analysis and not in control implementation.
NPTEL



11:44 AM Sun 12 Jun
Adaptive_Control_Week12
IEEE_WorkShop_Slides_Lavretsky
AIntro
Adaptive_Control_Week12
Multilayer_neural-net_Lewis_Liu
if we choose $v = \frac{1}{\lambda} Z^T \bar{e}_2$
$\dot{V} = -k_1 e_1^2 - k_2 \bar{e}_2^2 + \bar{e}_2 Z \tilde{\theta} - \lambda \mu_F \|Y_F^\top \tilde{\theta}\|^2 - \lambda \mu_{IF} \tilde{\theta}^\top Y_{IF} \tilde{\theta}$
if Y_F is IE
$\leq -k_1 e_1^2 - k_2 \bar{e}_2^2 + \frac{1}{2}\bar{e}_2^2 + \frac{\|Z\|^2 \|\tilde{\theta}\|^2}{2} - \lambda \mu_{IF} \sigma_1 \|\tilde{\theta}\|^2$
$\leq -k_1 e_1^2 - (k_2 - \frac{1}{2}) \bar{e}_2^2 - (\lambda \mu_{IF} \sigma_1 - \frac{\|Z\|^2}{2}) \|\tilde{\theta}\|^2$
We can again choose a λ large enough so as to obtain $\dot{V} < 0$, but again λ is only used in analysis and not in control implementation.
$\dot{V} = -k_1 e_1^2 - k_2 \bar{e}_2^2 - \lambda \mu_F \|Y_F^T \tilde{\theta}\|^2 - \lambda \mu_{IF} \tilde{\theta}^T Y_{IF} \tilde{\theta} \leq 0$
NPTEL



11:47 AM Sun 12 Jun
Adaptive_Control_Week12
IEEE_WorkShop_Slides_Lavretsky
AIntro
Adaptive_Control_Week12
Multilayer_neural-net_Lewis_Liu
$\leq -k_1 e_1^2 - k_2 \bar{e}_2^2 + \frac{1}{2}\bar{e}_2^2 + \frac{\|Z\|^2 \|\tilde{\theta}\|^2}{2} - \lambda \mu_{IF} \sigma_1 \|\tilde{\theta}\|^2$
$\leq -k_1 e_1^2 - (k_2 - \frac{1}{2}) \bar{e}_2^2 - (\lambda \mu_{IF} \sigma_1 - \frac{\|Z\|^2}{2}) \|\tilde{\theta}\|^2$
We can again choose a λ large enough so as to obtain $\dot{V} < 0$, but again λ is only used in analysis and not in control implementation.
$\dot{V} = -k_1 e_1^2 - k_2 \bar{e}_2^2 - \lambda \mu_F \|Y_F^T \tilde{\theta}\|^2 - \lambda \mu_{IF} \tilde{\theta}^T Y_{IF} \tilde{\theta} \leq 0$
$e_1, \bar{e}_2 \to 0$ even in absence of IE
+ $\tilde{\theta}, e_1, \bar{e}_2$ remain bounded.
$\dot{\tilde{\theta}} = -\mu_F Y_F^T Y_F \tilde{\theta} - \mu_{IF} Y_{IF} \tilde{\theta} - \frac{1}{\lambda} Z^T \bar{e}_2$
NPTEL

Now the issue that we are left with as before, is that what happens if you do not even have initial x. So, we are still left with that same question. In order to answer that questions, I would again go back and try to modify my adaptive law. So, what I would do is, I would add here in another terms, lambda theta tilde dot. Sorry there will be a negative sign minus lambda theta tilde dot v. And this is assuming that theta tilde dot is minus mu F Y F transpose Y F theta tilde minus mu IF Y IF theta tilde minus v. So, this is what we assume. So, this is actually I am sorry this is theta tilde transpose v. So, this is what we have as an additional term if I think of introducing another additional term in the parameter update law.

So, why am I trying to do this, because we already saw that everything is nice if there is initial excitation, but if there is no initial excitation, both these terms are gone. I cannot really use them to dominate anything, but then I still have this mixed term, which I do not know if I can dominate in the absence of excitation, because these 2 terms are gone. So, then the only solution I would have is to somehow try to get rid of this term. And that is what we are trying to do.

So, if there is excitation everything was nice, excellent result, but in the absence of excitation, you want to make a modification to the adaptive law. So, that I do not have to deal with this next term, which I cannot cancel or which I cannot dominate anymore. So, then I proposed this additional term, and I tried to see what this additional term in the additional term has to come from the Lyapunov analysis. So, this term remains as it is here, this term remains as it is, and here too this term remains as it is.

Now, at this stage, if you notice, I can choose, so at this stage. I can look at these 2 terms together, because this term is if I take a transpose, it is theta tilde transpose Z transpose, e2 bar, e2 bar is irrelevant, because it is a scalar. So, I can move it around wherever, but I do have to take a transpose of these sets, so this theta tilde transpose and this theta tilde transpose is sort of matching. So, if I choose if we choose your v to be let us see, Z transpose, this is capital Z, Z transpose e2 bar divided by lambda if I do that, then this term and this term cancels out. So, this term does not exist.

I hope this makes sense. Essentially, I am able to choose a v such that I can cancel these 2 term. And once I cancel these 2 terms, I have no more mixed terms. So, this term is not there. This is also missing, so my e dot after this point beyond this point my V dot becomes different. So, my V dot with this new adaptive law becomes minus k1 e1 square minus k2 e2 bar squared minus lambda mu F Y F transpose theta tilde square minus lambda mu IF theta tilde transpose Y IF theta tilde. So, there is no more mixed term.

So, this is already less than equal to 0. So, if there is no excitation, then I can ignore these terms or I can still prove that Y IF theta tilde and Y F transpose theta tilde are going to go to 0. And you will also be able to prove that e1 e2 bar goes to zero, e1 e2 bar goes to 0 even in absence of initial excitation plus theta tilde e1 e2 bar remain bounded.

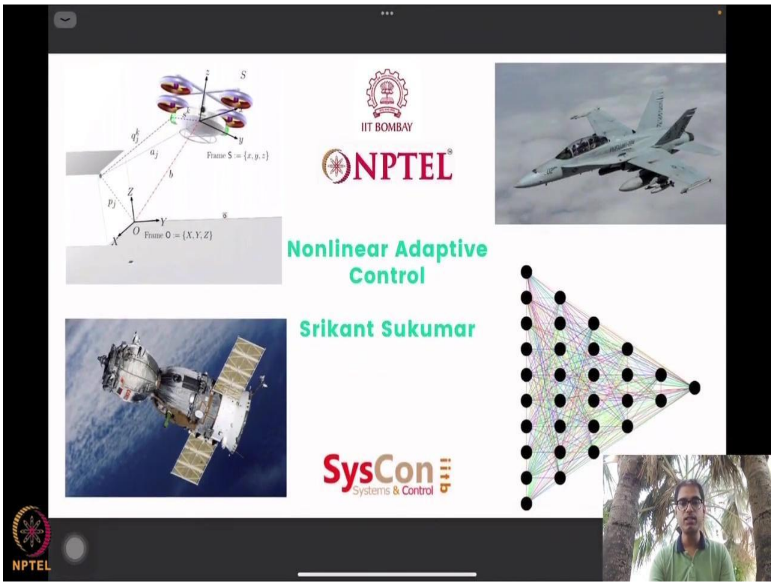
So, all these quantities also remain bounded. So, this is rather nice. So, if I make a small change, so, my adaptive law as you can see becomes this if I write it out here, so, theta tilde dot is minus mu F Y F transpose Y F tilde minus mu IF Y IF theta tilde and then minus 1 over Lambda z transpose e2 bar. So, if you see this is like the certainty equivalence adaptive law and this is the usual initial excitation method, if I make this modification with this additional term here, then I do get nice performance even in the absence of initial excitation.

So, basically you have now seen that it is very much possible to do this adaptive control design this initial excitation for double integrators also, in fact, I would strongly encourage you to try it for the unmatched case and so on and so forth to see that this is in fact applicable to a large variety of dynamical systems. Of course, we would also see the certainty equivalence modification and so on and so forth.

And, but I would encourage you to do that. So, if you do have initial excitation, which is a significantly weaker requirement, you are in very good shape, you can actually deal with many many different dynamical systems and your update laws generation is recoupled from sort of the dynamics. I mean, it is hidden, it is not like it is decoupled in the sense, it is not independent of the dynamics, it depends on the dynamics through the regressor and the filtered regressor and filter control and so on. But the fact is, you are still, you are not seeing it in the expressions.

And that is rather nice in terms of the construction, the construction has a standard structure, just your regressor keeps changing. So, you can pretty much use the same update law and plug it into a different system, just by modifying the regressor. And this is of course, very useful in implementations. Excellent.

(Refer Slide Time: 23:15)



So, what have we sort of seen in this week is that we started with this idea of initial excitation based adaptive control, we understood that the persistence of excitation is a very stringent requirement, we have, of course, seen how persistent excitation is used to prove parameter learning. And now, we wanted to do something better. So, therefore, we looked at initial excitation based adaptive controller, the idea was relatively nice, simple, straightforward, in that you write your system in a regressor parameter form. That is the first thing and then you start to design 2 level of filters, with 0 initial conditions.

And once you do that, you have this standard regressor parameter form in the filtered variables also. And because of this, you can construct a very straightforward parameter update law. With this you then want to design your controller like you would do and your parameter update was of course, already designed. We also talked about the parameter dependent version of initial excitation just like we have for persistent excitation, because more often than not, your regressor has to depend on the state, which means it depends on the solutions and which means it depends on the initial conditions.

Which are effectively going to function like parameters. Therefore you do require to define parameter dependent notions of initial excitation also. But then once you do that the proof goes on in our proof of convergence and all is significantly simpler in this case, because you even in the Lyapunov analysis directly, you start to see negative terms, if you have initial excitation or parameter dependent initial excitation.

So, things are significantly nicer in terms of the proof. So, you do not need more complicated results, like the integration integral lemma like that you did require in the parameter dependent version of persistent excitation, we then, of course, realize that a couple of things, one is that choosing lambda is not easy. Which is this lambda, which has it is used to dominate these mix terms. And we also saw that if there is no initial excitation either.

So, if you do not even satisfy this weak requirement, then things may not work very well for this kind of adaptive controller. So, what we saw was that a simple modification of this adaptive controller, when we add the original CE adaptive law, with this update, also helps alleviate this issue. That is one, you do not need to choose a lambda anymore, because there is no mixed terms, there is nothing to dominate. So, there is no need to choose a lambda and 2 in the absence of excitation also you get bounded trajectories and convergence of tracking error to zero which is what most adaptive control theorist promise anyways.

So you do not go back on that fundamental promise of adaptive controllers. And on top of that, you add this cool feature of having you only of requiring only excitation at initial time, and not requiring anything for infinite time, like you do when you talk about persistent excitation. So, I hope you found this new method rather interesting and impressive. If you already designed some adaptive laws for your dynamical systems, I would recommend that you do the same with this initial excitation based method.

And I would also recommend that you compare the performance So, that would be interesting for you to report and for me to see and so do let me know if you can see any difference one way or another. So, in a subsequent week, we are going to focus a little bit on a neural network adaptive control. So, learning ideas, and how we can do some provable sort of results in learning. So, we will essentially follow pretty much an article from Frank Lewis in the subsequent week in order to get a feel for our adaptive control and learning are intrinsically connected.

So, I really hope you will join me in the last week which is sort of excursion into more modern areas. Of course, the paper is not very modern, but because computations et cetera have become significantly cheaper now, learning and deep learning has become more popular tool for many systems engineers. So, we will look at some of that in this upcoming final week of our NPTEL. So, I hope you all enjoyed the sessions, and I hope to see you again in the next week. Thank you.