Nonlinear Control Design

Prof. Srikant Sukumar

Systems and Control Engineering

Indian Institute of Technology Bombay

Week 11 : Lecture 65 : State constrained control: Part 1

Welcome to another session of SC 602, Non-linear control. So, what we were doing last time was actually giving you an intro of adaptive control. So, I think I of course did not go much further. So, there is of course an entire section you can see which is on backstepping parameter unmatched with control. I did not cover this because again this is not an adaptive control course. But of course you can see that the examples we were considering the control and the uncertainty appears in the same equation.

So, this is called a matched uncertainty. And as you can imagine these are much easier to handle. All that we did was instead of theta star we put a theta hat. So, again I did not say that at the time.

But this idea that whatever control I get I just substitute the unknown with an estimate of the unknown is called the certainty equivalence principle. So, this is very commonly whenever you hear somebody in adaptive theory talking about certainty equivalence this is what they mean. They basically design the control of assuming the parameter is known. And then in the adaptive control this is replaced the true value of the parameter by its estimate theta hat. So, this is the notion of certainty equivalence.

So, basically you are taking the certain controller and you are creating an equivalent controller. There is no you never change the structure of the controller. And this obviously gives us an easy way of constructing the control of course. So, the idea was pretty straight forward. All we did was we again constructed the known case control.

So, we of course we did it we are back stepping in this particular case. But it does not matter how you do it you construct a known case control. And then you basically replace the theta star in the control by the theta hat. And once you do that you basically add to your Lyapunov function a term in the parameter error. So, this is what we did.

So, you had this Lyapunov function for the system or the control Lyapunov function. Remember that we need a control Lyapunov function or a strict Lyapunov function. So, you should not even start doing adaptive control without a CLF or a or a strict Lyapunov function. Otherwise adaptive control is known to fail royally ok. Not even like you know in small ways it will basically really damage the system in the presence of disturbances.

So, make sure that for the known system you always start with a strictly Lyapunov function which is what we get by back stepping. That is why back stepping is so popular. Then we just added a term in the parameter error ok. This was the idea right. And then we did the analysis and the purpose of this analysis was to get a update law.

We got that. The really cool thing like I said was you do not care where you start this update law. You do not care about the initial condition. It can be anything. It can be as far from the true value as you want.

You will still get exact tracking. This guarantees that you get exact tracking ok. And that is pretty amazing if you ask me right because you made the system agnostic to the unknown however much it was. So, as you saw with the your typical signal chasing arguments which is what we did in the end here. We could prove that E1 and E2 both go to 0 correct.

And we of course only could prove that something like this you know. You only can prove this on the parameter errors right. So, the parameter errors are not guaranteed to go to 0 unless you have something called a persistence of excitation condition right. Of course we did not discuss what is the persistence of excitation condition. That is not part of the plan again right.

But finding the parameter is the so in adaptive control we do not particularly care about finding the parameter. Our aim is to track and we did it successfully. So, we do not care to find it ok. But in system identification and also in all learning all of learning all of deep learning finding the parameter is the requirement ok. And so in those cases persistence of excitation is inherently assumed yeah.

Although they may not use those same terms same ideas or same words sorry they may not use the same words but the idea is the same. Persistence of excitation is required for parameter learning ok. And that is the crux of it. You cannot you cannot learn without having this kind of a persistence yeah. But although typically all the learning algorithms are data driven.

So, nobody tries to verify these conditions. They just assume you did a good job ok. And then you run the algorithm yeah. But in reality if you do want to learn in a true sense you need persistence of excitation ok. That is really the key point here.

So, this is what we will do in terms of adaptive control. We are not going to do anything more. What I want to do is start a new topic and that is on constraint control. We want to talk about control under state constraints ok. So, a lot of you until now we have been talking about lot of scenarios although we have done non-linear control design and I hope you have learnt some methods at least pretty well.

But we have not seen a lot of realistic scenarios. We have definitely not seen disturbances. But I did say in passing yeah we did not cover it. But we did say in passing that you know that because we are doing Lyapunov based design it is naturally robust. In the sense if you add some disturbances your system is not going to become unbounded.

You are still going to have some kind of bounded performance which is governed by the size of the disturbances. And in a lot of cases you can even increase the control gains in order to make the residual set. Suppose you are supposed to go to the origin instead of going to the origin you will somehow you know circle around the origin alright. And whatever the size of the circle or the set is can be made can be shrunk by choosing larger control gains ok. So, this is really the advantage of Lyapunov based design that you are getting robustness for free ok.

It is easy to prove I am not going to really prove it here. But robustness is sort of something we have already handled. That is why I do not talk about it separately in this course ok. So, you can just blindly go and design a Lyapunov controller for your systems. Basic robustness is bounded noise bounded disturbance handled.

You do not have to think about it and worry about it yeah great. But the other scenario is that of constraints on the control and constraints on the state ok. Of course I am not going to talk about control constraints per say which can also be handled by the way in this using the same methods that I am going to talk about. But we want to talk about state constraints. So, if you see one of the issues with non-linear control is that if you have ever implemented any non-linear control and we have we have implemented these back stepping controllers on quadrotors and things like that.

Usually what will happen is you will get a lot of overshoots ok. Like in linear systems when you are talking about a linear system you design a linear controller. You can quantify how much is the overshoot right. You can actually say I want this much overshoot accordingly I will choose the PID gains using some transfer function ideas and all that. You can actually compute what is the overshoot.

You can actually control the gains so that the overshoot is minimized and things like that. In the non-linear case there is no such equivalent ok that you can that how much can you overshoot and it is a non-linear system right. It can actually throw you out before coming back in alright. So, it is a possibility that will be overshoot. So, we see a lot of overshoots and then gain tuning is not very easy.

So, this is another thing I will get a lot of questions on how do I tune gains and things like that ok. So, one of the key requirements in a lot of applications is that your state trajectories while they are trying the transient basically until now we are only talking about asymptotic performance ok. So, that is what so until now only asymptotic guarantees yeah. What does it mean? It means that I am only saying that I will do this as t goes to infinity or that as t goes to infinity. I am going doing something nice as time becomes large ok.

So, the big question is what about transients? Yeah. So, what happens to the transients alright. So, while I am converging to the good place how bad is my trajectory yeah. This as you as you can see none of the theory that we have done until now does not really cover this. Yes, you have to you have to understand that because again we are using Lyapunov based design we have always had something like v dot of x of t say is less than equal to 0.

At least we had this right. We had some kind of negative semi definite which means that we have v of x of t is less than equal to v of x of t 0 correct and this gave us some kind of ellipsoid right. I mean this is this what does this mean? This means that in terms of your right in terms of your yeah in terms of your real world systems what does it mean? It means that I sort of got a ellipsoid in which my states will remain ok. Now this ellipsoid could be of any size right. I mean it is not being yeah it is governed by your initial conditions right here right. But suppose your initial conditions were large you started with a large offset yeah not that uncommon.

Suppose you are very far from your desired trajectory. So, you started with a large offset. So, and this is your ellipsoid that is governed by this ellipsoid is actually nothing but v well let us see this is actually ellipsoid is if you like this terminology yeah it is weird notation. But all I am saying is it is like if you compute v x t 0 it is some constant value right because it is v is mapping to scalars it is some constant value right and if you take v inverse of that this is the shape you might get yeah you could get some other shape also I am not saying it has to be an ellipsoid. But I am saying it is some shape some closed set right because it is a inverse of some constant ok.

Because say for example if I think of v as x 1 square by 2 plus x 2 square by 4 right then I want so this is this set is something like equal to some constant right. So, you can see this is some kind of an ellipse equation right. So, ellipsoid ellipse equation and similarly if you do more you can get some general versions of ellipsoid. Now the point is if your initial conditions were rather large or initial errors were rather large because this can be in terms of x or it can be in terms of error depending on how the problem is posed right.

Then this ellipsoid is large right. Now what happens in the future later on all I am saying is I am restricted inside this set and nothing more right. So, my states can be anywhere here right I mean in the sense that all I am saying is at any instant in time I am not going to exceed this ellipsoid yeah which can if my trajectory if my desired was this yeah this is my desired right I want to sort of move here yeah usually you will take origin is the desired but do not worry about it I am just giving you a representation it may not be the truth in error vary in error variables typically origin will be your desired always that is how we have been working but if you think reference trajectory right you should x minus r is equal to the error so the r could be something in this blue something like this blue thing right. But I may be straying from this really far away right I could do this try to reach it but then I could go here try to reach it I could go here and then eventually slowly I can reach right. But I am going all the way to the end could I could potentially go all the way to the boundary right which means I am my states are growing larger large in an attempt to reach this set okay. So, yes there is some bounds on the transients but they are not bounds that you might like yeah.

Then there is another scenario where there is a lot of safety critical applications what is the safety critical application and one simple application is obstacle avoidance applications suppose I have some robot and I have these yeah so I have this starting point and I have this end point yeah I want to go but I do not want to hit anything okay. So, now the robot if I design a normal control I have no way of specifying that I have to avoid these sets right there is no such thing right how do you specify it there is no way typically your normal control will probably just go here which it will not because there is an obstacle so it will hit the wall yeah that is a problem. So, what do you want the controller to do? We wanted to go here and then figure out there is an obstacle I see yeah we wanted to go here we want and figure out say okay fine there is an obstacle then do this maybe go here then do this and do here okay. Whatever I mean I am not talking optimality here optimal path might be something like I am not talking optimality here but I am saying that I want to at least avoid the obstacles right. So, if I see an obstacle I should turn now it should not be like I have to of course you can do it the funny way where you can see a obstacle give a trajectory which is around it and then start rotating around it and all that right that is of course one way but the other ways of course you can include it in the control design itself all right.

So, safety critical applications are basically saying that you have some reach or avoid sets you have some avoid sets or possibly you do not want the states to grow too large in the transients itself while it is trying to reach okay you want the states to behave nicely while it reaches okay and that is a requirement for almost any control yeah anything you do any trajectory tracking you probably do not want to stray too far away from the desired yeah unless because it may go out of your operational range sensors might be an issue everything might be an issue if you have two large velocities in a spacecraft that is a problem. So, you do not want to even in the transient stage exceed this okay. So, this is what sort of motivates the notion of barrier function for control okay. So, this is what we want to look at and there is a little bit of theory we will cover yeah and let us see I will sort of try to motivate it initially if you look at this dynamical system is just a dynamical system okay not a yeah there is no control yet okay as of now no control we want to talk about what these barrier functions are right. So, the idea is make sets invariant okay or forward invariant, forward invariant is forward in time invariant okay.

So, until now aim has been what to have stability and reach a point and reach a equilibrium right. We did not try to make sets invariant although if you remember we did all this Lassalle theorem naturally there were some invariant sets okay. So, that is where so let us see let us use that example also okay. Suppose I take again some V of X of T alright. Now suppose again there are it has level sets of this kind this is one level set of V another

another and so on right.

Because so this will be something like V of X T equal to say C 1 V of X T equal to C 2 V of X T equal to C 3 these are the sets because I already showed you how you draw and of course you would expect that because it is smaller so you would expect by virtue of the fact that V is a Lyapunov function you would expect that what C 3 will be less than C 2 less than C 1 right. Because it is inside so you expect that it is the case right. Now what do we do? We start typically with Lyapunov function, Lyapunov candidate maybe right V X T right and then what do we typically get? We at least try to get something like V yeah because if we did not get this I guess we achieved nothing right. So obviously let us assume we got this. So therefore at this point this became a Lyapunov function itself okay.

Now like I showed you before the set which is defined by so we already know that V X of T less than equal to V X at T 0 holds right because of this right. So what does it mean? It means that if I define my set omega C or omega say C 1 as set of all X such that V well I will just say V X less than equal to C 1 yeah. What is this set? This set is just the outer ellipse right is the outer ellipse okay is invariant under what condition? What do I need for this set to be invariant using this Lyapunov function? I only need that V X T 0 right is that fine right because I already have that V X is less than V X T 0 and if V X T 0 is less than C 1 I am good okay alright alright alright great. Now now one of the issues so this is so I got invariance right this is actually invariant right because forward in time invariant if I start inside this V X equal to C 1 set I am going to stay inside it okay alright. Now suppose my initial conditions okay so this is assuming what my initial conditions are somewhere inside this way somewhere inside this outer ellipse.

Now suppose my initial conditions were inside the inner ellipse okay somewhere inside the inner ellipse right. So in that case you can see that omega C 3 is also invariant if V X T 0 less than equal to C 3 correct. So suppose I started inside the smaller ellipse then this becomes invariant which means I never escape this gate right never escape this gate. So basically by this logic if you keep using this you will see that depending on initial conditions how it is chosen or in fact even by scaling the V itself you cannot you don't need to actually change initial conditions by scaling the V itself you will be able to show that all these sets inside some larger ellipsoid are all invariant. So once you start inside them that's what is invariance right invariance means start inside a set remain inside a set okay that is invariance okay.

So whenever you start inside you remain inside this set okay but the point is this is not what we are necessarily looking for okay. So it might so be the case that our set this guy is our safety set that is I want to remain only inside this larger ellipsoid that is my safety set I am allowed to operate anywhere inside this larger region but because of this Lyapunov style of construction what happened is if I started inside this set I never escape this yeah I don't even utilize the larger space right I don't go here and come back to origin right whereas I am allowed to I am allowed to use the larger space larger state space because I only want to keep this guy invariant but because of the Lyapunov construction that I used right with positive definite function and then negative semi definite V dot what happens is if I start here I remain only inside this set which is governed by my initial condition not by a predefined set that I as a user you know defined right. Typically what how do you define your safety set or desired region of operation you decided predefined it you don't base it on your initial condition, initial condition can be anything right but it's you don't base your safety critical set on where you started so because of this kind of construction you will always stay inside this set so you are actually wasting the ability to work here yeah which might in some cases be what you desire actually yeah you might need more control effort to always stay inside yeah you might be allowed to get out and then come back in that's okay because as long as you are remaining in the larger set but this doesn't allow it okay. So we want to look at constructions which will allow us to make one set forward invariant but everything inside is not forward invariant necessarily okay so that is what we are looking to do with barrier functions okay. Thank you.