Nonlinear Control Design

Prof. Srikant Sukumar

Systems and Control Engineering

Indian Institute of Technology Bombay

Week 6 : Lecture 32 : Control Lyapunov functions-Part 4

So, anyway I will repeat it for the benefit of the audience that here the small control property and the continuity of the control at origin are connected by an if and only if relationship which means that if the control is not continuous you do not have small control property, if you do not have small control property the control is not continuous at the origin ok. So, it works both ways, everywhere in fact yeah that is the power of an if and only if result alright. Anyway, so the key thing is you now have a formula for a universal controller which will work I mean it may look like a funny looking formula but it works ok. It gives you a stabilizing controller it is smooth everywhere but at the origin where it is continuous ok, still pretty good I would say. Now, before going to the rest of the piece of the proof which is now a little bit more mathematical and it really just talks about you know the fact that the controller that we have prescribed is smooth ok. So, that is really what we will prove but before we do that I want to look at this example ok and the fact that there are multiple controllers possible and so on and so forth ok.

So, if very simple example alright, very simple scalar example x dot is minus x cube plus u ok, this is the system that we are looking at. Now, one obvious control without doing any analysis or anything is to simply prescribe my control as this x cube minus x ok alright. Then you can see that your closed loop system is x dot is minus x. So, this is basically what you would call a feedback linearizing controller.

We have not looked at feedback linearization but basically what this controller is doing is linearizing the system in a sense. So, this is what this control does yeah. When you look at feedback linearization this is the kind of controls you will design, you will cancel, you will try your best to cancel the non-linearity and then introduce a nice linear term. In this case you got a linear exponentially decaying system ok. One thing that is obvious is if your when x is large control is large yeah that should be obvious right because of the x cube term which will dominate alright.

If x is small control is small yeah seems intuitive also large very far away from the origin large control close to the origin small control seems intuitive. Now let's do what we call typically called Lyapunov redesign. We don't design a control first like we did here which is sort of a feedback linearization based control. We just we choose a Lyapunov function first for the system. In fact what we choose is a control Lyapunov function but I don't say it like that.

Let's say it's a Lyapunov function. I know it's a candidate Lyapunov function because it's continuous and positive definite and all the nice things. So, this is a good function. So, I take V dot and I get this. Now V dot obviously contains the control right because I have not chosen it yet.

Now without going into any CLF theory or universal formula I just use this and look at this to design a control. What will I do? I know that this two multiplied are not giving me anything bad they are giving me a negative term. So, I don't try to cancel it. I don't try to cancel it. I just introduce another negative term.

Okay. So, my control is just u equal to minus x. In fact it's a linear controller. Okay. It's a linear controller.

Alright. So, I get V dot as minus x4 minus x square which is again negative definite. So, by Lyapunov theorem I have asymptotic stability. Great. Done. What is the nature of this control? Again large when x is large, small when x is small.

However, important thing is it is not never going to be as large as this guy. Alright. Never going to be as large as this guy. In fact I have made a small computation also just for our reference. If I take x equal to 10 this one comes out to 990 magnitude.

Magnitude is 990. Magnitude of this guy comes to be minus 10. Okay. This is positive 990 this is minus 10. When x is value of x is 10 units.

Whatever that unit is. Okay. Alright. Let's look at the universal formula.

Okay. Universal controller. We know that V equal to x square by 2 is already a CLF. Yeah. Why? Because you can see that if the control term is 0, if the control term goes away, I just verified like this. Then I still have a negative term here.

Yeah. That's all. CLF means is this. Till the control terms vanish or are 0 then this drift term is still negative which it is. Right. What is the drift vector field in this case? By the way, if I ask you what is the drift vector field and what is the control vector field? Everything is a scalar field here but still what is the drift vector field in this case? F0.

What is F0? Yes. You should be able to parse this. That what is F0, what is F1 if you want to apply these results. What is F0 you think? And what is F1 for this system? This is the system that we are looking at. So there is of course just F0 and F1 and nothing more. Because there is only one control.

Number of control vector fields is same as the number of controls. So there is only this much. So what is F0? Minus x cube.

What is F1? 1. 1. Okay. Great. So it is a CLF because del V del x F0 is negative. Yeah. When the other part is gone.

Okay. Alright. I will start the universal formula computation. Okay. So I first compute Ax which is what? Del V del x F0 x. Okay. So what is del V del x? It is x.

Del V del x is just x times F0 x which is minus x cube. So Ax is minus x to the power 4. What is B of x? It is del V del x which is x again times F1 x which is just 1.

Okay. So it is just x. Okay. So it should anyway it should be evident that I mean although I checked I said it is CLF and all that. It should be evident to you that if Bx is 0 the only way Bx can be 0 is if x is 0. Okay. And remember that we have to check all the CLF conditions only for non-zero x.

Okay. Only for non-zero x. Therefore I am saying that the condition 2 of the CLF definition is trivially satisfied which means you can never get in a situation where your drift where the control vector fields do not contribute and x is non-zero. Okay. If your control vector fields do not contribute then x is 0. That is the only possibility in this. So it is anyway trivially satisfied.

We don't have to check anything but it is okay. In this case you also see that it is minus x4 is there which is a nice helping term. Alright. So I use the universal controller.

Okay. I don't have to write this case because this means x is already at the equilibrium. Right. So this case is irrelevant for us. Yeah. Because if Bx is 0 x is 0 which means I am at the equilibrium so obviously I am not applying any control.

It is stupid to apply a control at the equilibrium. Okay. Right. So I just compute this formula. And what is that? I have just populated the terms.

Minus A plus square root of A squared plus B4. And that is what it is. A squared B4. And this whole thing multiplied by B over norm B squared which is just what is B over norm B squared? Sorry.

This is B. Norm B squared is this. Where B is a scalar so norm of B is absolute value. Okay. So this is B over norm B squared. Okay. So basically I have just you can just simplify this.

It just comes out to this guy. This is what is the control. This is what is the control. It is x cubed minus x square root of x4 plus 1. Notice that the expression for this control is significantly more complicated than both the other ones. One is a linear control which is minus x other one is x cubed minus x.

So simple expressions. Yeah. This expression way more complicated. But it is a very nice controller. Right. Better than the other ones. Why? If x is large what happens? This can be ignored.

This is playing a very small role. So then this is x cubed. This is almost zero for large x. Imagine for large value of state you are almost applying no control. For small value of x what happens? This is gone.

This is also gone. You are left with minus x. So for small value of control it behaves like this one. Like a linear controller. For small values of x it behaves like the linear controller. For large values of x it almost applies no control.

In fact you can compute. If you take x equal to 10 this is like 1000 minus 10 square root of 10 to the power 4 plus 1. This is almost zero.

Yeah. But it is evident anyway. Right. Because if I ignore the one both of these are x cubed. Ok. So this is a very cool controller. Right. Because it is like it is applying very small control for large values of state.

Ok. Now whenever I say things like this it is your job to tell me nothing comes for free. Ok. Nothing comes for free. Remember this these controllers both these controllers in fact this guy gives you a x dot equals minus x.

This guy gives x dot is minus x cube minus x. Both of these are beyond exponential rate of convergence. Super exponential. Exponential or super exponential.

Ok. Because x dot is minus x is already decaying at minus 1 t. Minus 1 rate exponential decay. This is even faster than that.

Ok. So both of these are converging rather fast. Ok. This guy there is no guarantee. Probably converging very slow. Ok. It is not necessarily converging very fast.

So nothing is for free. Yeah. We have designed controllers they are actually not that useless. It is not like we did a very shoddy job.

No. In fact more often than not we use this method. Sorry. We use this method of control design. Directly guess it from the v dot expression rather than go to the universal formula. Alright. But yeah. This is a nice control if you want to keep your control commands in check.

And you don't care about how fast you go. Especially there is a lot of applications when if the states are very large you don't care to apply very large control. One of the most common application is spacecraft de-tumbling. Ok. When a spacecraft is you have the launch vehicle right in multiple stages they get released then it leaves the earth's atmosphere then it says it gets close to the orbit then the spacecraft is released from the top.

Ok. When it is released of course it goes into the orbit but it is it is rotating at a crazy rate. Very very large rates of rotation. Ok. And remember in this case the states are angular position and angular rate.

Ok. Angular position is whatever 0 to 360 can't be more than that. But angular rates are super large and that is also state of the system. Right. So the states are very large in this case. Now for the de-tumbling manoeuvre if you start firing your engines like crazy to stop the de-tumbling.

Ok. Then you lost all your fuel in the first 10 minutes of your mission. Ok. Then what will you do after that. Yeah. So they don't usually folks don't care about the equipments are well enough well protected enough that for large angular rates also the equipment is not going to get damaged.

This is important. If your equipment is going to get damaged then you better de-tumble soon enough. But if your equipment is well stacked and you have done a good enough design so it is not going to create a problem for you even if you are rotating fast. All you want to do is make sure that it stops after say 2 days.

It is fine. Yeah. So the de-tumbling manoeuvres are really done with very small intensity jets. Very small. Or in fact a lot of times it is not even they try not to use jets so you find a lot of results on using magnetic torquers for de-tumbling. So they use the earth because if it is a low earth orbit you can imagine there is a small magnetic field on the satellite. And outside the earth orbit outside the earth's atmosphere there is no real atmosphere to stop it.

So even the small magnetic forces are enough to stop the satellite or slow down the satellite. So mostly your de-tumbling manoeuvres will be done with magnetic torquers. So very small torques very small. So these are the kind of controllers you want in such a scenario that you don't care when you stop.

You just want to stop. Ok. So but of course like I said if you are time critical application then yeah but then you better have enough you can see the trade off right. If you want fast you better have enough fuel or enough actuation ability. Ok. Because you are burning at this rate.

This is huge compared to this which is almost zero. Right. So lot of applications are there where you don't care about applying like you know huge torques just to stop a vehicle or something. Fine with stopping when it stops. Yeah. The only thing is you still have to apply something in the in orbit because there is nothing else stopping it.

So then you can't do a mission. Right. If you want earth pointing satellites then how will they point the earth if they don't stop.

Ok. So that's what is important. Alright. Ok. So I hope you understand that the universal formula though it gives funny looking controls it is a very useful control design. Secondly more often than not we don't use the universal formula. Yeah. We directly guess the controller from this kind of a Lyapunov redesign.

So we start with the CLF ideas. Right. But then we design the control using you know just by guessing at this stage. Ok. Now one of the other things that you know you folks should also see is that actually I can keep control to be zero in this case.

Then the system is still a stable system. Ok. The purpose of designing controllers for such systems would be to get a particular convergence rate. Ok. So this system is already asymptotically stable without any control. Yeah. The purpose of designing controller would be to get a rate, particular rate of convergence go at a particular speed.

Yeah. Alright. So there is a nice exercise for you guys. Find a control Lyapunov function and apply the universal formula to get a control.

This is the exercise. Any questions? Ok. Absolutely. Everything is dependent on the V. There is no except for feedback linearization there is no real method which gives you a control design without a V. I mean while you have things like model predictive control and so on where you sort of guess a control out of a optimization. But in those cases stabilization guarantees are not rigorous.

I mean there are guarantees but the guarantees are very conditional for stabilization. Yeah. They work more on I would say intuition. See it is more like I mean how would a typical predictive controller work is it would basically say that I will discretize this problem. So I will have say I look at say 20 time step horizon. So now I have a discrete problem right. I have a discrete problem meaning that I can write this whole problem as a if my state space is say 5th order 5 states.

Then I will write this whole problem as a 5 state and 20 time steps. So 100. So 100 by 100 order matrix and I can pose an optimization problem right. What will be my optimization requirement? It would be like say fuel consumption is low. So something like a U transpose U will be 1 and U transpose not U. U will also be the control at every stage every time step time step 1 time step 2 all the way to you know time step 20.

Ok. So you put one piece of optimization cost is this U squared or U transpose U. The other piece could be you want your state to shrink. So you can put another cost as X transpose X where X is again the stacked.

So it is like 5 states in 20 time step. So 100 is X transpose. So now you have an optimization problem. You solve this optimization problem and if the solution is good then you know that if you keep applying this this this is a control sequence right. You got a control sequence for 20 time steps. If you apply this control sequence for 20 time steps you know that you will hopefully reduce the X transpose X because that is what you pose as optimization problem. Then you apply the first two first two three steps of that control then you do the do it for the next horizon.

This is the receding that is what it says receding horizon. You switch the horizons first horizon second horizon. So you compute for this horizon apply control only until this point. Then you compute for this horizon apply control here.

This horizon apply control here. You don't apply all the 20 steps of control. You apply only the first two or so. And so there is some proof which which shows that yeah you will converge to the origin and all that. Proofs are very highly dependent on a lot of things yeah.

They are actually dependent on on existence of a controller. So it is very odd yeah. So so they are more optimization I mean and if you want to see such setup such a setup is way more useful if you want to put constraints on the states and things like that way more useful. Yeah. But if you are looking for stabilization type or you know tracking type results you rely on V have to rely on some V.

Yeah. And again it's also a nonlinear problem right. Optimization nonlinear optimization. So you don't know what comes out of it. Yeah. You put it into an engine then what.

You don't know. Just like a lot of you will also use learning algorithms these days right. I mean it's like I just have a hammer and I keep hitting all my nails with it. So you don't know that if you learned well enough or not and if it will then perform well in given us particular set of data. So that's a little bit of an issue. But anyway different context.

In this context we are looking at stabilizing controllers or tracking controllers hard to do it without a V.

Yeah. Very difficult to do it without a V. Absolutely. Yes. Yes. Yeah. But it sort of means that close to the equilibrium you will have less control. Doesn't necessarily say that away from the equilibrium what will happen. Close to the equilibrium these controllers will guarantee that your control magnitude will be small.

Away from the equilibrium no such guarantees. This is guaranteed. Yeah. That's not dependent on V. The small control property will hold. Yeah. Yeah. All right.

So what I want to do is to look at the proof is a little bit I'm not sure how many of you will follow but it's OK. There's a quick proof of why the controller is smooth. OK. How we do this is by invoking the implicit function theorem.

OK. If you don't know what is the implicit function theorem I just tell you in words you can look it up later. It basically says that if you have a function of two variables or two or more variables.

Yeah. Actually it's stated in two variables only. If you have a function f x y such that f x bar y bar is equal to 0 say. OK. And you also have d f I think one of the partials right. I would I think it's the partial with respect partial with respect to the y variable.

This is full rank. OK. Or maximal rank when I say full rank it means partial of f with respect to y maybe non square matrix. I hope you understand that. Yeah. f is not necessarily a scalar function could be a vector function.

OK. The implicit function theorem can be stated for a vector function. f is in general a vector valued function of two parameters two variables x and y. OK. Again different dimensions not necessarily scalars and the partial of f with respect to the y bar or the y y variable is maximal rank.

Yeah. Not full rank maximal rank. Then you can say that y bar in the neighborhood of so y can be written as a function of x smoothly in a neighborhood of x bar y bar. OK. Basically you can it's that's why it's called the implicit function theorem.

This function is implicit but you can it is possible to write it in an explicit way. OK. You can give an explicit relationship. OK. A lot of times that's not possible. If I give you an implicit function of variables then you may not be able to write an explicit expression. Yeah. In this case it says it is possible if the partial of the function with respect to the y variable is full rank then y can be written as a g of x.

A function of smoothly also smoothly. OK. All right. And of course we started f with f smooth. We also started with f otherwise of course nothing is possible. OK. All right. Thank you.