Nonlinear Control Design

Prof. Srikant Sukumar

Systems and Control Engineering

Indian Institute of Technology Bombay

Week 12 : Lecture 70 : Finite time stability: Part 3

Hello, welcome to lecture number 3 of the final week of this NPTEL course. So, we were talking about finite time stability. And we've already looked at sort of the conditions that are required to talk about finite time stable systems considering that there is some non-Lipschitz kind of properties that we desire to have finite time convergence. So, we essentially look at autonomous systems, we've been looking at autonomous systems, which are unique in forward time. Therefore, the requirement is that the functions be locally Lipschitz everywhere, but removing the origin. And then we of course, define finite time stability, which includes the notion of a settling time function, and a neighborhood on which this settling time function works.

So, the settling time function is a function of the initial conditions. And it essentially tells us beyond what time the system trajectories are going to go to zero and stay at zero. So, we of course had this proposition, which essentially says that beyond the settling time, the trajectories will remain at zero for all time beyond the settling time. Now, we obviously want a Lyapunov characterization because that's what we've been doing this entire course.

We've been looking to get some kind of a Lyapunov characterization for almost all of our results. And that's what we want to do here as well. So, we stated the Lyapunov finite time theorem, which essentially requires the existence of a continuous differentiable V as before and positive definite, you know that these two things actually make this a candidate Lyapunov function in the normal sense of the word. And then we would like V dot to be negative definite again in a removed origin removed domain. And so D removing the origin.

So, everything that we state here has been removing the origin because things are not uniquely defined at the origin. The solution exists but is not unique at the origin. Therefore, we remove the origin from the discussion. And further we require that we have this kind of not just negative definiteness, but something much stronger. So, we already state negative definiteness, but this is something much stronger, which is what gives us this finite time convergence type of condition for alpha that is between 0 and 1.

Now, given these three conditions, we have finite time stability. And on top of that, this theorem itself gives an expression for the settling time itself. In fact, an upper bound on the settling time, if you know the initial value of Vx0, right, the and we actually saw how to compute it, it's pretty straightforward, right? You just integrate this differential inequality, given initial and finite final values and initial and final time. So, you get something like this on the left hand side and this on the right hand side just by a standard simple integration. And then we have this time quantity on the left hand side and we take all these other quantities on the right hand side.

And if we equate this to 0, we know that this guy is also going to 0 because it is lower bounded at 0. So obviously, it can never be less than 0. So if the right hand side goes to 0, and the left hand side is less than equal to 0 means it's actually going to be exactly equal to 0. So that's what we sort of want. So once we equate this, you actually get a nice expression for the time, which is in fact, the settling time.

Yeah, I mean, this is of course, used as an upper bound, because of the typical conservativeness of the Lyapunov analysis. Yeah, you can see that there is a less than equal to already here. Therefore, it is possible that you might reach 0 faster than this time. But this is definitely an upper bound. And as you can see, this time, and as is typical for finite time stability, this time depends on the initial condition because v0 is nothing but v of x0.

So that's something that's already been highlighted here, v of 0, v0 is actually nothing but a notation for v x0. Now, next we state like a converse theorem, that's what we do now. We state a converse theorem. What is the converse theorem? It says that if origin is f t or finite times stable, right, and n is as in definition one, as in, it's not definition one, but as in in a finite time stability definition. So what was n? n was the neighborhood in which the settling time works.

That's it, nothing very special. Okay, then what we say the converse states that there exists c0 function, v a Lyapunov-like function, which maps n to real numbers, such that it satisfies all the conditions that were actually requirements in the previous case. So what were the requirements? One is that v is positive definite, right, then v dot is also c0 and negative definite on n again. So v dot being negative definite obviously just means that v is c1 again, exactly like we had in the previous theorem, right, you had v to be c1, yeah, v to be positive definite, v dot to be negative definite. Here there is d removing 0 is what we are considering, but here we are considering the set n, right, okay, which was the set that comes from the finite time stability definition.

And further there exists k positive and alpha in 0-1 open interval such that v dot plus k v to the power alpha is less than equal to 0 on n, okay. So the only difference between the converse theorem, yeah, and the main theorem and the finite time stability theorem is that there everything was in the domain, right, here we are talking about a domain removing origin, right. Actually there is an error here, this should also be the domain removing origin, right. But because a priori we are not given a set n from the finite time stability definition, the set n actually comes from the Lyapunov finite time stability theorem, right. In the converse theorem on the other hand we are already starting with the assumption that the system is finite, origin is finite time stable, therefore we have a settling time function and we have a set n on which the settling time function is valid and therefore we are everywhere using this set n, okay.

And so somehow you can see that this is a if and only if kind of a condition, okay. So if you have finite time stability you have such Lyapunov functions existing and if you have a Lyapunov function existing then you have finite time stability, right. You just have to satisfy these three properties, right. These three properties here and vis-a-vis these three properties here, very similar looking, okay. So again we are not going to prove this, you can look at these references, yeah.

If you want to sort of see and understand the proof we are simply giving an overview and so we are not really going to prove things here, okay. What we are instead going to do is do more fun things and actually look at an example, right. So what is the example? The example is rather important, right. It is the Spacecraft Angular Velocity Stabilization, right. You have already seen some spacecraft examples, right.

So we are looking at Spacecraft Angular Velocity Stabilization and what is the Spacecraft Angular Velocity model? It is something like j omega dot is minus omega cross j omega plus some control u and you know that j is basically three by three symmetric inertia tensor, right. You know that omega is the angular velocity in body frame and u is some external control. For example, a thruster, yeah. Thrusters are the most commonly used external actuators in satellite whether it, so this is an orientation angular velocity, it is an orientation control problem. So you still have thrusters to actually manage what is called reaction control system.

So you have thrusters as part of the reaction control system to manage the orientation speeds and so on, yeah. Suppose we have started at some speed omega zero and we want to actually drive the speed to zero. You can see that if there is no control then zero is an equilibrium of the system. So it is a fair thing to ask to go to zero equilibrium. Of course, we usually do this via Lyapunov functions, right.

If I wanted to do it in infinite time, I mean my standard Lyapunov function would be something like omega transpose j omega. This you can understand is basically the kinetic energy of the system, right. And actually half omega transpose j omega is the kinetic energy of the system, right. So we actually remove this half, right. It is, so we use twice the kinetic energy of the system just for simplicity.

And so if we take a v dot, we get twice omega transpose j omega dot and that's equal to twice omega transpose and j omega dot can be substituted from here which gives me minus omega cross j omega plus the control, right. Now you understand that this vector is orthogonal to omega, right. That's evident because the cross product is orthogonal to each of the component vectors. So you have omega and j omega as the two vectors in the body frame. So if I take the cross product of these two, then I'm certainly going to get a vector which is orthogonal to these two and the dot product of the vector with its orthogonal is obviously zero.

So omega transpose omega cross j omega is zero, right. So omega transpose omega cross j omega is actually zero. Also property of the scalar triple product, right. So this is actually the same as omega dot.

Yeah, that's identical. These are the same things. Yeah, we're just saying the same thing and therefore this comes out to be a rather simple expression and that's twice omega transpose u. Now if you wanted some kind of a finite time convergence, I would simply, in fact exponential convergence, why not, I would simply plug in u as minus j omega which would imply that I get v dot as minus, well I'll just take it as half j omega just to make my life easy and I will get something like minus omega transpose j omega as my v dot which is, actually let me modify this further and say this is minus k by 2 j omega and this, I'll do this more carefully, this is twice omega transpose minus k by 2 j omega. This is a scalar so I can move it anywhere and this cancels with this so I will get minus k omega transpose j omega and that's minus k times v, right. As you can see this is exponential decay, right.

So in fact I have obtained exponential convergence. I can obtain exponential convergence of the angular velocity dynamics to zero. I can exponentially go to zero. However, you know that exponential is also infinite time, right. So obviously that's not what we're interested in and you can also see that this control is also rather nice, right.

I mean it's not just that I got infinite time convergence and I'm rather sad. No, because my control is also rather nice and smooth, right. So that's something that's good, right. That's something that's good, right. I get a nice smooth infinitely differentiable controller u and so this is a infinite time convergent smooth controller.

So that's the great property that we have. It's infinite time conversion and the smooth controller, right. Now if I want finite time convergence, remember what is the property I'm looking at for ft convergence because I have most of the other properties already, yeah. What is it? I already have v to be c1, right. If you see I chose a v that's rather nice.

It's c1 in fact c infinity, right. It's infinitely differentiable and v dot negative definite is also rather easy. In fact even in this case you see that v dot was negative definite. So that also I've ensured. I can ensure what I need for finite time convergence primarily is then v dot plus v dot. I would write it rather like that v dot is minus kv to the power alpha, right.

Okay, so in order to achieve this I will actually prescribe a controller which is minus k over 2 but this time I will take all right this time I will take something like omega transpose j omega to the power alpha minus one multiplied by j omega. Okay, so it's almost a similar looking controller. It has this k by 2 j omega here. Still the only thing is I've scaled it with some scalar divisor and it's something that divides because notice this alpha minus one is less than zero, right.

Alpha is between zero and one. So important thing to remember is that this is less than zero. So it's actually in the denominator and that's important to remember that it's actually in the denominator. All right, now if I do this what I will get as v dot is actually equal to let's be careful again 2 omega transpose minus k over 2 omega transpose j omega which is a scalar again time to the power alpha minus one times j omega, right. So notice this is a scalar.

These are scalars. This is the only vector quantity that I cannot move around but this entire thing I can move around wherever. So obviously I move all of these out, right. I know that this cancels with this. So I'll have minus k omega transpose j omega alpha minus one times omega transpose j omega and this is actually equal to minus k omega transpose j omega to the power alpha, okay. Because these two multiply to give me alpha, okay.

So this was smartly chosen exactly so this product becomes alpha and because it's a scalar I could move it out no problem. So the important thing to note is that this is actually minus kv to the power alpha, okay, as required. So I wanted v dot to be equal to minus kv, in fact less than equal to minus kv to the power alpha. I've made it exactly equal to minus kv to the power alpha and this alpha can be any number between zero and one. Now the important thing like I said is that this control now is not smooth, right.

Yeah, important to remember that this is not smooth. Unlike before, why? Because as I said this is actually division, a division by some omega transpose j omega to the power one minus alpha, okay. So there is a division. So something funny is happening at the origin, like at omega equal to zero.

Everywhere else it's fine, right. Everywhere else it's actually fine. So in fact what you can claim about this is that this is locally Lipschitz everywhere except the origin. This is exactly the kind of controls we've been looking at and it's of course continuous, right.

So this we can say is c zero. This is c zero. This is continuous, right, everywhere. This is continuous everywhere. Just that it is locally Lipschitz everywhere but not at the origin, okay. So why is it continuous at the origin? It's evident that I mean as you can see the numerator is also going to zero as omega goes to zero. The denominator is also going to zero zero as omega goes to zero.

Therefore you do have continuity but you do not have the Lipschitz property at the origin, okay. And this is exactly the kind of controllers that we have studied, that we've been talking about that give us finite time stability and you can see that we have exactly this v dot equals minus kv alpha and therefore we know that from our finite time stability theorem that there is this time within which within exactly within which my states in this case the angular velocity will go to zero, okay. So that's pretty powerful, right. I mean it's saying something rather nice that you are going to go to zero in finite time, okay.

And that's very very important. All right, great. So that's sort of what we wanted to discuss on the finite time stability. What we want to do is start a new notebook and talk about sliding mode control, right. So I am going to do that in yeah in a new notebook, right. So that it's a new topic, right.

So we started in a new notebook, right. So this is sliding mode control. The interesting thing is you will see a lot of similarity between what we spoke about in finite time control and what we see in sliding mode control, yeah. In fact sliding mode control I would say is a kind of kind of finite time control which involves sliding modes and we will actually look at what these are. So again because of our time constraints we are not going to really look at you know we are not going to really look at a lot of proofs. We are sort of going to motivate the idea of sliding mode control as far as possible as much as our time permits, okay.

And so we will do this mostly through examples and ideas. That's what is our aim, okay. So suppose so sliding mode control again lot of nice deep history here and and it's it's Utkin is mostly credited to bringing sliding mode control to the sort of mainstream control and he's been active I mean he has written a large large section of papers and articles in the area of sliding mode control establishing the area of sliding mode control but there are many many researchers now in the area and it is sort of falls under the under what is called variable structure controllers, okay. This is because these controllers tend to change their structure depending on where they are operating. So anyway we will explore different facets of sliding mode control as I said using examples, okay.

That's primarily our idea, right. So let's look at a simple second order system first. So this is x1 dot is x2 and x2 dot is u plus f x1 t, right. And you have some initial conditions x1 0 is x1 0 and x2 0 is some x2 0, alright. Obviously u these are all scalars so x1 x2 is an r u is an r is the control and f x1 x2 t is again belongs to reals but is essentially a non-linear disturbance, okay.

It's a non-linear disturbance term, okay. So that's important. However we assume so this is something I will probably highlight that f x1 x2 t is bounded for all time. So there is a uniform bound, okay. So obviously the idea is construct or I will actually put it more formally as an objective construct disturbance rejecting u such that x1 x2 go to 0 asymptotically, okay. Notice to begin with we did not require finite time convergence. Although I said that sliding mode control is sort of a finite time control, sort of a method in finite time control but sliding mode control actually has its more novel features than the finite time control the way we have seen it, yeah.

So the aim is not immediately to achieve finite time convergence of both states but in fact the aim is more to reject disturbance, reject bounded disturbance like this, right. So you can think of this disturbance obviously as you know standard external disturbances but it can be disturbances that are coming also from some kind of reduced order dynamics or you know I mean model approximations and model truncations and several other things, right. I mean this could essentially basically comprise of all the non-linearities that you don't want to directly work with when designing the control. So as long as you can say that this is bounded. So one of the typical obviously a tough sort of question that is usually very common when you are doing sliding mode control is you know is how to deal with the unbounded cases, right.

I mean because if you think of how these disturbances I mean at least in these basic examples how we are what we are asking of this disturbance term is pretty heavy, is pretty heavy because as a non-linear control theorist you would immediately ask why would a function of states be bounded. I mean what you are essentially asking is not just a bound but a uniform bound. You are saying that this quantity has to be bounded has with this uniform bound for all x1, x2 and t, right. And that's a pretty serious ask because even if I think of something as simple a non-linearity as say f x1, x2, t is say some polynomial non-linearity x1 square plus x2 square, right. Or say t times I mean it's something as simple as t times x1 squared plus x2 squared.

You see that this is unbounded, right. It's not uniformly bounded, right. It's bounded if your states are bounded, yeah. So, unbounded but bounded if states bounded, okay. In fact, even if it's even the linear case, right, you know things like t x1 plus x2 has the same property, right. It's bounded if the states are bounded, yeah or for bounded time in examples like these.

I mean if you have actually I shouldn't say this is bounded if states are bounded. I will not suppose I have something like this, yeah. So, these even these simple non-linearities and even this linear function for that matter is bounded only if the states are bounded. Otherwise, it's unbounded.

Therefore, this is one of the tougher critiques, right. However, there are of course modern answers to this if you can a priori guarantee that your solutions that your states a priori guarantee with some control that your states are not going to escape some invariant set then some bounded invariant set then you are fine. Then you have these kind of guarantees on your states and you are you are more than happy to go along with that, okay. However, in general it is not easy to satisfy this, yeah. But these are the assumptions that the very very classical standard sliding mode control methods work with and that's what we are going to see as well, yeah. So, we will continue with this in the subsequent lecture. Thank you.