Sliding Mode Control and Applications

Dr. Shyam Kamal
Department of Electrical Engineering
IIT(BHU) Varanasi

Week-04

Lecture-20

Welcome back. In the previous class, I was talking about robot control. In the class before the previous one, I was talking about integral sliding mode control. And what is our main philosophy that you can apply or design any classical control and If your system is highly non-linear, then you can design integral sliding mode control. So, somehow we are merging two philosophies: the philosophy of traditional or any kind of classical control, which is not able to compensate for uncertainty or disturbance. And after that, I am adding some kind of sliding mode control based on the integral, such that from the starting point onward, I am able to ensure sensitivity with respect to matched perturbation, and if we also have unmatched perturbation, then at least we will guarantee robustness.

And in all critical applications, we have already seen in the previous class that we have to actually give a guarantee that our robot will work with very, very high precision. Due to that reason, some kind of robust control is always essential whenever we are talking about either holonomic robots, nonholonomic robots, or actuated or underactuated robotics. So, in this lecture, I am going to explore the integral sliding mode control for fully actuated system and that system is also holonomic and due to that reason I am going to consider the two link manipulator and after that I am going to tell you that how to design the integral sliding mode control. So, purpose of discussion what we have seen in the previous class.

That robotics has several applications and manipulators are used for several medical purposes. So, what are we basically going to do? We are going to compensate for these kinds of challenges. What kind of challenge is presented by traditional robot control methods, which are based on PID controllers or other classes of controllers or optimal control? Most of the time, we assume that there is no uncertainty or perturbation, but in realistic situations, we always have uncertainty and parameter variation. Since these kind of manipulators are working in very harsh environment, so always there is some kind of perturbation and robot dynamics we have already seen that is very very non-linear or highly non-linear and due to that reason, we are lacking in terms of the sensitivity to the bounded perturbation and uncertainty. So, what is the proposed solution? One of the proposed solutions is the integral sliding mode control.

And since in a new era, people are exploring artificial intelligence, AI-based control, and data-driven control. So here, what can you do? You can design data driven control, artificial neural network based control or reinforcement learning based control, and then you can add another control loop that is based on this philosophy that is based on the integral sliding mode control and it is possible to show that your performance is going to improve. So, let us start this lecture. So, I am going to consider some kind of robot manipulator, and this is the generalized model of an n degree of freedom rigid robot manipulator. So, in the previous class, we have already seen that this class of systems is going to fall into the class of holonomic systems, where the constraint is only in the form of the position.

What am I going to do? I am going to apply the Euler-Lagrange method to obtain the mathematical model. I am going to calculate the kinetic energy, potential energy, I am going to take the difference, and after that, I am going to apply the principle of least action and then I will get the Euler-Lagrange equation. Okay. So, in this particular dynamics, this represents the inertia matrix, and we have already seen the property of the inertia matrix that it is symmetric and positive definite. So, what is the meaning of positive definite? It means that if I take some kind of vector y of the appropriate dimension. So, here suppose M_Q is $n \times n$.

So, here y is nothing but $n \times 1$. So, $y^\top$ is $1 \times n$, and y. Therefore, this should be greater than 0, and I am also assuming that it will satisfy the property of symmetry. So, this is the meaning of the first assumption, and all rigid robot manipulators are going to satisfy this kind of property. So, after mathematical modeling, how do you check that your mathematical model is okay? You can just check these two properties.

Now, the second is responsible for all kinds of Coriolis or centrifugal force, as well as the gravitational force. Our configuration variable is Q, $\dot{Q}$, and $\ddot{Q}$, which represent the joint position, velocity, and acceleration, and τ is torque. So, I have already discussed that in the robot control problem, I either have to apply force or torque to control the system. So, here I am assuming that τ is the torque vector. So, what is the key characteristic of this model? This model is highly non-linear; you can see from here and here also, and this dynamics is highly coupled.

After that, M_Q is nothing but what is responsible for the inertia matrix that depends on Q, and Q is changing. So, M_Q is changing with respect to time. And due to that, this is also varying. So now, for the control problem, whatever we are going to solve, I have to control those class of systems where the parameter is also varying. So, what is a control challenge? In order to design classical control, I have to make sure that Q and $\dot{Q}$ are considered because both depend on the position and velocity, which should be available in real time, as they are changing.

And whatever your control is, it is also able to compensate the non-linear coupling. And obviously, whenever we are designing some kind of control, we are designing for a realistic purpose. So, suppose that I have designed some kind of manipulator. So, I have to pick something; some kind of job I have to pick from this space. Once this manipulator picks some object, at that time, the load is different. Whenever it is moved from one point to another point, then the behavior of the load is going to change. It might be possible several times; this is also not constant. And due to that reason, payload variation is always present in realistic situations. So, particularly for medical use or several other uses, such as agricultural use, whenever you are going to apply some kind of manipulative task, there is always payload variation. So, your control should actually be good enough to handle that kind of situation, that kind of payload variation.

Now, since I am going to design the integral sliding mode control, what am I going to use? I am assuming that the nominal value of this model, the nominal value of the mass-inertia matrix, is known to us, and similarly, n_0 is known to us. So, these two things are exactly known to us, and due to that reason, whenever I am going to calculate the torque, at that time I have incorporated these two pieces of information. You have to understand that whenever we are designing control, the control you have to apply to the real system must be implemented through the actuator. So, I cannot apply any kind of unknown term for the control purpose, and due to that reason, whenever you are designing control action, you have to give a guarantee that everything is known to you.

So, here I am assuming that Q_e, where Q_e is nothing but $Q - Q_d$, and in the previous lecture I have already discussed that Q_d is the reference trajectory that is already given to us; some kind of desired path we have to move our robot along. I cannot move in an arbitrary way, and Q is nothing but the output; the current output from this model.

So, there is always a mismatch. That is nothing but the tracking error. And after that, I am also assuming that the tracking error or whatever design trajectory, I am assuming that the tracking error and the two derivatives are also known to us. Otherwise, I cannot design this kind of feedback. This is nothing but simple.

You can interpret this as a simple PID controller or a PD controller. My term is not there, so you can assume this to be a PD controller. So, here you can see that since during the design of the PD controller, I am assuming these two pieces of information are available, and for that reason, I am also assuming that the desired trajectory should be twice differentiable. If the desired trajectory is not twice differentiable, I cannot apply this kind of control.

Now, since I have selected the torque based on the nominal parameter, I am going to update that torque here. Once you update the torque, then now you can see that this is the nominal model, but there always exists some kind of variation on M, because M is depending on Q, and due to that reason, there always exists some kind of mismatch here. So, if M is exactly equal to M_0, it means that if I know, or in real time I have some way such that I can calculate M and M_0 and both are exactly matching, then our problem is very easy. I am able to design K_D and K_P, and then I have just second-order dynamics with respect to each state. Now, I can design any PD controller and I have used this kind of damping; you can use any other damping.

So, this is critical damping, and you are able to show that Q tends towards 0 as $t \to \infty$. And what is the physical interpretation? If Q is tending towards 0, it means that $Q - Q_d$ is actually tending to 0. So, you can track $Q = Q_d$, but this is valid if you assume that M_0 and n_0 are known to you; however, in a realistic situation, that assumption is not correct. And due to that reason, this dynamic also contains some kind of non-zero term here, and always if that term is bounded, then it is possible to show that Q_e is actually bounded by some kind of ε, where Q_e is a vector, and ε depends on the variation of the parameters $M_0 - M$ and $n_0 - n$.

For critical applications, this is not sufficient. Particularly, if I deploy a PD controller for medical purposes, you can see that if I have to move the tip of the robot along a specific path, I might accidentally damage some part which we do not want to cut. Due to that reason, I have to account for model uncertainty, which degrades performance, as well as payload variation, since whenever the tip is moving, there is always some variation. Therefore, I need a compensator. Even if I have inexact information, I can maintain $Q_e \to 0$ as $t \to \infty$.

In the best case, we hope that $Q_e \to 0$ as $t \to t_f$, i.e., finite time convergence. But in this lecture, I am assuming $Q_e \to 0$ as $t \to \infty$, which is enough for several practical applications. Due to that reason, I am going to design a unique feedback-based integral sliding mode control. This is a unique feedback, and here I am assuming that $s = 0$, then this term equals 0.

If $M = M_0$ and $n = n_0$, then obviously, we will get exponential convergence. K_D and K_P can be designed to ensure asymptotic stability. It is possible to show that more than asymptotic stability implies exponential stability. However, in practical situations, there always exists model uncertainty, and these terms are not going to match exactly. Therefore, the right-hand side of this second-order dynamics always contains some term that acts, and I cannot guarantee that the tracking error $Q_e \to 0$, i.e., $Q \to Q_d$. Hence, conventional PD control is inefficient for this application.

How do I combine sliding mode philosophy to maintain $Q_e = 0$ in the absence of exact parameters? I add extra components inside the control. This extra component compensates for parameter variation with τ_1. I am assuming that $\ddot{Q}_d$ is known, i.e., the second derivative of the desired trajectory is available and differentiable. Now I am going to design the integral sliding mode control. In order to design it, you remove all sets of uncertainty and then apply the nominal control, which is the nominal torque. I am assuming nominal parameters M_0 and n_0 are known, and then I design the integral sliding surface.

Suppose this is the surface defined as

$$S = \dot{Q}_e + \Lambda Q_e + \int_0^t (\dot{Q}_e + \Lambda Q_e)\, dt,$$

where $t = 0$, the integral is 0, and the surface term is 0. It means that from the starting point onward, I begin on this surface. I take the derivative of the sliding surface. That is the common philosophy. S is calculated as $S = \dot{Q}_e + \Lambda Q_e$. If you calculate $\dot{S}$ and substitute everything, using the system dynamics, it is possible to show that you will get the desired expression. Please check the calculation yourself.

Now you can design control using either a conventional form or a saturated form $S/\text{mod}X$, both are fine. Here C_i is responsible for the projection matrix in the integral sliding mode control. You can prove stability by taking the Lyapunov function

$$V = \frac{1}{2} S^\top M S.$$

If you maintain a τ_0 gain of the switching part greater than 0, and ρ is the minimum eigenvalue of the inertia matrix M, which is upper and lower bounded, then it is possible to show that the equivalent value of the sliding variable, by Utkin's equivalent control philosophy, satisfies $S = 0$.

If you substitute $S = 0$ into the system dynamics, the system is governed along the sliding surface. The sliding motion compensates for any uncertainty or variation. The nominal trajectory is maintained because τ_0 is designed based on PD control. In this way, the integral sliding mode control ensures that deviations due to uncertainties are compensated.

Now, consider a two-link manipulator with rigid links L_1 and L_2, each with point mass assumptions. If the point mass location changes, the dynamics also change. The world coordinates are defined with z along gravity. Our first goal is to calculate the dynamics of this manipulator.

How do you calculate the dynamics? So, I can easily calculate forward kinematics. What do you have to do? You just have to project this, and you can see that I will get this. So, this part is easy. It means that if you have to move in this particular space anywhere, you can adjust θ_1 and θ_2, that is, q_1 and q_2. So, this is q_1 and this is q_2, and then you can move anywhere in this space.

I have already told you that another dynamic is, suppose that if you have a similar manipulator and if you fix this position, then you have several ways to cover this position. This kind of configuration is called inverse kinematics, and you can see mathematically that it is not unique. Mathematics also tells you the real story. For the design of integral sliding mode control, I need the system dynamics.

So, what am I going to do? I have to calculate kinetic energy and potential energy. Due to that, $x_1 = l_1 \cos\theta_1$, and I have already told you that $\theta_1 = q_1$ and $\theta_2 = q_2$. So, this becomes $l_1 \cos\theta_1$, and this becomes $l_1 \sin\theta_1$. And what is the projection of $l_1 + l_2$? It is $\cos\theta_1 + \cos\theta_2$. Due to that reason, this is not very difficult to check.

Now, how do you calculate the kinetic energy? Kinetic energy is a scalar quantity. You can calculate the derivative, sum it, and then expand and simplify to obtain the inertia matrix from the kinetic energy. You have to verify these calculations.

Potential energy is

$$PE = m_1 g y_1 + m_2 g y_2.$$

Since our working space is perpendicular to g, it is later possible to show that this term does not appear in our dynamics. Again, I calculate $KE - PE$, verify calculations, and finally, I can also obtain n_1, n_2, and g. I have noted the absence of the gravity term because the manipulator is operated in the plane perpendicular to gravity. Due to that reason, $g = 0$ and this term is 0. The overall dynamics is then represented by the derived equations, which you should verify.

Once you get the mathematical model, you can verify all properties: skew-symmetric, symmetric, positive definite, etc. Verification ensures that the model is correct. For manipulators with 2, 3, or 5 joints, you can easily check all these properties.

Now, I will assign one homework. During the tutorial session, I will give you the simulation code, but it is better to try yourself. Take:

$$m_1 = 2, \quad m_2 = 1, \quad l_1 = 0.3, \quad l_2 = 0.2, \quad q_1 = \frac{\pi}{2}, \quad q_2 = -\frac{\pi}{2}.$$

Calculate m_{11}, m_{12}, m_{22}, then fix some trajectory and design a controller that combines

$$\tau = \tau_0 + \tau_s,$$

where τ_0 is the PD controller (K_P and K_D) and τ_s comes from the integral sliding mode controller. Show that you achieve the desired trajectory, which depends on your specifications.

During the design of sliding mode control, one piece of information required is the upper bound of the mass matrix and its minimum eigenvalue $\lambda_{\min}$. Using this, you can ensure

$$\lambda_{\min}(M^{-1}) \leq \text{controller gain} \leq \text{bound},$$

so that torque saturation is avoided when using actuators such as electric motors, pneumatic, or hydraulic actuators. Kinetic energy is also bounded.

The simulation part is homework. Now, to conclude today's lecture: I have taken an N-link manipulator and shown how to design integral sliding mode control, which is a combination of the PD controller plus integral sliding mode (on-off) control. For a two-link manipulator, I showed how to obtain the mathematical model, verify all criteria, and provided parameters for a homework assignment to simulate your own manipulator.

Integral sliding mode control is compatible with other control types. You can also explore adaptive integral sliding mode control from the literature. Learning-based control can be combined with integral sliding mode control to improve performance. When applied together, the performance of learning-based control is enhanced.

With this remark, I am going to end this lecture. Thank you very much.