INTELLIGENT CONTROL OF ROBOTIC SYSTEMS

Prof. M. Felix Orlando

Department of Electrical Engineering

Indian Institute of Technology Roorkee

Lecture 08: Fuzzy Logic Control-I

Good afternoon, everyone. Today, we are going to see fuzzy logic control, part 1. The outline of this lecture will be: First, an introduction to fuzzy logic control. Then, we will be seeing the Mamdani type of fuzzy logic controller.

And in that, one example of a two-wheel mobile robot for trajectory tracking tasks. And then, finally, we will conclude our talk or lecture with Lyapunov function-based fuzzy control. Now, coming to fuzzy logic control introduction. A fuzzy logic controller is an approach to represent human-like thinking in a control system. It can be designed to imitate human deductive thinking to deduce conclusions or infer conclusions from past experiences. We can see when a fuzzy controller can be applied: it can be applied when there is a problem associated with the classical control strategy, such as unavailability of data, parameter variation problems, highly complex plants,

and inaccurate models. During these circumstances, one can apply a fuzzy logic controller in order to get good results. Now, coming to the types of fuzzy controllers, there are two types. One is called the Mamdani type fuzzy controller. It is basically a direct adaptive control strategy that has been designed directly based on the fuzzy rule base. The fuzzy rule base is nothing but a set of fuzzy rules. These fuzzy rules can be derived or designed based on intuition or based on a Lyapunov function candidate. Now, the other type of fuzzy controller is the Takagi-Sugeno type fuzzy logic controller. It is an indirect adaptive type controller in which the system to be controlled has to be identified in terms of a TS fuzzy model. And based on this TS fuzzy model, the controller is designed for the system.

Now, let us see about the Mamdani type fuzzy controller, which is the first type in our lecture today. So, the block diagram of the Mamdani type fuzzy controller is that first it has a fuzzification process where the crisp input is converted into a fuzzy output. The crisp value is converted into a fuzzy value through fuzzification, which is done by a fuzzifier. This is done basically by involving membership functions. Because membership functions are the ones that convert a crisp value into a fuzzy value.

And then the fuzzy output from the fuzzifier goes to the fuzzy inference system, which basically involves a rule base, a set of rules. And the inference engine or inference mechanism is Choosing which rule it is. Or to get the consequence. Of each rule.

We can use the Minimum composition. To get the consequence of each rule. And these consequences of each rule. Have been combined.

To form the output distribution. From the output distribution. We can go for. Defuzzification. That is the output of this.

Fuzzy inference system. Goes to defuzzifier, where defuzzification happens, which means a fuzzy input has been converted to crisp output, which is helpful for the control action of a system or a plant. So, there are three fundamental blocks. Fuzzifier meant for fuzzification. Rule base associated with the rule base is the fuzzy inference system.

And the defuzzifier meant for defuzzification, converting fuzzy to crisp. Because in the real world, we need crisp data. That is why crisp input is given as input to the fuzzy logic controller. So that we get a crisp output as a control signal for the plant which has to be controlled. Now here is the flowchart of the fuzzy Mamdani-type controller.

It shows that there is a plant whose current position or actual position has been sensed by a sensor. That value has been given to the fuzzy inference for fuzzification, which converts it into a fuzzy value, and there is a fuzzy inference mechanism that produces the output in terms of fuzzy values so that the fuzzy value has been converted into a crisp value by defuzzification. There are two popular methods in defuzzification. They are the weighted average and the center of gravity approach. And then the crisp output of the defuzzification block goes to the actuator, and the actuator actuates the plant.

Accordingly, we get a response that has been sensed by the sensor, and the process is iterative in nature. And you can see here in the Mamdani type fuzzy controller, this fuzzy inference mechanism has a fuzzy rule base with output state variables also represented as linguistic variables. That means we have if X_1 is A_1 and X_2 is A_2, then Y_2 is B_1. So, in this way, the output is also represented by

a linguistic variable directly coming as a linguistic variable. That is the salient feature here. Whereas in terms of the TS fuzzy controller, the output is given by an expression, not directly as a linguistic variable. It can be derived; the output can be derived in terms of an expression in the TS fuzzy controller. Now, let us talk about the fuzzification.

Here, the crisp value is converted into a fuzzy value by the fuzzification process. Let us consider two input states, E and delta E, as the two input states. Precisely, they are the input variables. They are converted into fuzzy sets using triangular membership functions. Here, you can see in the schematic, rule 1, the membership function value for E and the membership function value for delta E. So, based on the crisp value of E, we get a corresponding membership function value of 0.8.

In rule 1. Similarly, the crisp input of delta E in its membership function gives 0.3. Corresponding to the crisp value delta E, it gets a 0.3, and you can take here, okay? So, the minimum of this will give the output membership function value. This is u. That means these two values are correlated by the AND function, so that we get the minimum, which is the output of this rule. Similarly, for rule 2, we have the crisp value of E considered with the fuzzy value 0.7.

Accordingly, the crisp value of delta E has the fuzzy value 0.8 by the membership functions. Here, it is triangular membership functions. This way of converting the crisp value of the states into fuzzy values is called the fuzzification approach. And as I mentioned to you, how we are getting the output membership function values is based on the AND operations of the membership function values or the fuzzy values. Next, we move on to the fuzzy inference mechanism.

It chooses a suitable rule from the rule base. It deals directly with the rule base. The rule base is nothing but a set of rules. And we are going to see how many rules a system has if it has an n-dimensional state space. Okay, so now coming to the fuzzy inference mechanism, it establishes a logical connection between the input and the output fuzzy sets. It performs operations on rules as a decision-making unit for the inference system. It performs operations on rules as a decision-making unit.

It establishes the rule strength by combining the fuzzified inputs according to fuzzy rules. Again, it establishes the rule strength by combining the fuzzified inputs according to the fuzzy rules, rule 1, rule 2, rule 3, because rule 1, rule 2, rule 3 definitely are going to have different fuzziness. Different rule strengths. Then the consequences are obtained based on combining the rule strength and the output membership functions. Okay.

Now the fuzzy inference mechanism continues with what will happen with these consequences. By combining the consequences of each rule, the output distribution is obtained. Okay. That is why from rule 1 to rule 10 or rule 1 to rule 5, there are 5 rules, 10 rules, 25 rules. Each rule has been operated or comes under the operation of composition based on the OR function.

So, the maximum function has been applied here. With the membership functions of the output of each or the consequence of each rule, next, the final block is the defuzzifier where the defuzzification process happens. It converts the fuzzy value to a crisp value using defuzzification methods such as the center of gravity approach or the weighted average approach. Any one of these approaches is generally used to convert a fuzzy value into a crisp value. So, that the crisp value is the output of your fuzzy controller block. That block, that value is called the control input signal to the plant.

Now, let us take an example which is basically a Mamdani type fuzzy controller for a two-wheeled differential drive mobile robot. This is basically example one; it is not two, it is one. So, the task is to track a given trajectory, say circular trajectories or Lissajous trajectories, through a Mamdani fuzzy controller. We need to make this robotic system track these types of trajectories: circular trajectory and Lissajous pattern trajectory. Let us see. So, the steps involved In order to perform this task through a fuzzy Mamdani type controller, involves giving the desired circular trajectory in terms of x_d and y_d because it is a planar circular trajectory. So, x_d, y_d; if it is 3D, then x_d, y_d, z_d. Then, give the initial position of the robot where it is placed initially, and then we have the for loop to take forward. First, find the desired orientation angle using the current position of the robotic system. Say, for example, this is the situation, and the robot is here, which is x, y, and here is theta, which is orientation

The orientation is obtained with respect to the global frame. The robot frame, body frame, is compared with the global frame. So, G represents the global frame, or represents the robot frame, which is in the center of mass of the chassis robot. or of the body of the mobile robot. So, here we have x, y, and theta of the center of mass of this mobile robot.

Now, the task is to reach here and track these wave points of a circle, then come back here. Now, As a first step, we need to find the desired orientation because we have $[x_d, y_d$

of all the waypoints, and we also need the orientation to be given as input to this robotic system because the robot has to track in this fashion. Here are the two wheels. So, in this fashion, and it continues here, okay.

So, if the orientation is not taken into account, it will not trace the trajectory of a circular path properly, which means that the center of the mass will be the position that will be tracked because of $(x_d, y_d$

If we do not give x_d and θ_d, if we do not give θ_d, this Circular trajectory will be tracked like this. Let us say So, you can see where the wheels are. The wheels should be facing the forward direction, as in this case.

That is why we need to specify the θ_d as well. Then only that will be taken into account by the controller, and we can have the two-wheel robotic system to track the circular trajectory in a proper orientation. Here, the orientation suffers, whereas here, the orientation is proper. Next, find the error in distance, find the error in distance, say

$$e_d = x_d - x$$

Likewise, find the error in orientation, which $e_\theta = \theta_d - \theta$ is

And then, find the input membership functions for the input variables. What are the input variables here? e_d and e_θ So, find the membership functions for the input variables.

Because the membership functions are the ones that will give the values meant for as well as for defuzzification. The membership functions convert the crisp values into fuzzy values. And these fuzzy values are utilized in defuzzification steps as well. Then finally, use the rule base to find the outputs.

The rule base is the building block in a fuzzy controller. Then Do the numerical integration in order to obtain the actual position of the center of mass of the robotic system. X and Y, as well as the orientation. And end of the loop, and this loop continues as we come to the end of the trajectory point.

You start here and end at the same point. So, the robotic system completes its circular path. So, the detailed explanation starts now. Before that, we need to have the kinematic model of this two-wheel differential drive robotic system. First of all, the configuration of the mobile robot is characterized by the position x, y as well as the orientation in a Cartesian coordinate system, capital C here.

Now, let us see the kinematic model is given

$$\dot{x} = v\cos(\theta)$$

$$\dot{y} = v\sin(\theta)$$

$$\dot{\theta} = \omega = \frac{r}{2b}(\dot{\theta}_R - \dot{\theta}_L)$$

Where $\dot{x}$ is the velocity along the x-axis and

$\dot{y}$ is the velocity along the y-axis of the robotic system.

And $\dot{\theta}$ is the angular velocity of the mobile robot. Then theta r dot and theta l dot are nothing but the right and left wheel angular velocities respectively. And r is the radius of the wheels and 2b is the distance between the centers of the wheels. This is a mobile robot, the wheel and this distance is 2b.

Now, the kinematic model in the discrete domain is given by

$$x_{k+1} = x_k + v_k\cos(\theta_k)\cdot T$$

$$y_{k+1} = y_k + v_k\sin(\theta_k)\cdot T$$

$$\theta_{k+1} = \theta_k + \frac{r}{2b}(\dot{\theta}_{RK} - \dot{\theta}_{LK})T$$

So, since it is in discrete form, definitely sampling time plays a vital role in taking it to the next sampling instant.

Now, coming to the design of the fuzzy logic controller based on the Mamdani type, it has this form: give the input as distance error and angular error theta e_{theta} and e_d, and the output of the fuzzy controller will give you the linear velocity and angular velocity of the robotic system. Linear velocity is represented by $\dot{x}$, $\dot{v}$ and and angular velocity is represented by ω of the robot body.

There are certain rules forming the rule base meant for the inference. Okay, so now let us see the input variables to this problem. The distance error $e_d = \sqrt{(x_d - x)^2 + (y_d - y)^2}$.

Similarly, the orientation error is given by $e_\theta = \theta_d - \theta$.

where $\theta_d = \tan^{-1}\left(\frac{y_d - y}{x_d - x}\right)$.

where y_d and x_d are the desired circular trajectory positions for this robotic system. Now, coming to the input membership functions.

Triangular membership functions are considered for the falsification process for this particular problem, the trajectory tracking problem of the two-wheel mobile robot. Since the variable D, which is the distance, is positioned in the universe of discourse from 0 to 20 centimeters, it is defined by membership functions. So, we consider five linguistic variables associated with the distance: zero distance, small distance, medium distance, long distance, and very long distance. So, five linguistic variables are associated with the

input variable or fuzzy variable D towards the distance. Similarly, the angular fuzzy linguistic variables are zero angle, positive angle, very positive angle, negative angle, and very negative angle V and A. And the angular value is positioned in the universe of discourse within the range of minus 180 degrees to plus 180 degrees. Now, let us talk about the output membership functions. The linear velocity and angular velocity are the output variables.

And again, we considered here 5 membership functions corresponding to the 5 fuzzy variables associated with the fuzzy linguistic variables linked to the output variables. So, let us consider zero speed, slow speed, medium speed, fast speed, and very fast speed—five linguistic variables associated with the output fuzzy variable. So, we consider the simplest straight-line membership functions for these five fuzzy linguistic variables associated with the output. Likewise, for the angular velocity output variable, we also consider five fuzzy linguistic variables, such as zero angular velocity, medium positive angular velocity, medium negative angular velocity, large positive angular velocity, and large negative angular velocity. Accordingly, we have five straight-line membership functions for this angular velocity output variable.

Now, we form the rules. Now, how many rules are there? In general, in a fuzzy logic controller, the number of rules is given by the expression Q to the power of N, where we consider an N-dimensional state space with dimension N. We have Q number of fuzzy zones associated with the states. So, the total number of rules is given by R equals Q to the power of N, fuzzy zone to the power of the number of states.

So, the number of states in our case is 2, and the fuzzy zones in our study are 5. So, 5 to the power of 2 equals 25 rules. Accordingly, we have developed 25 rules. Input is ZD 0 distance, and input theta E theta is 0 angle.

So, the output is 0 for linear velocity. Similarly, the output angular velocity is 0. Likewise, we go up to framing 25 rules, constructing the rule base. Now, coming to the defuzzification, we have utilized the center of gravity approach, which is given by the output value, precisely the crisp value of the output, say Z_crisp, is given by $Z_{crisp} = \frac{\int \mu(z) \cdot z \, dz}{\int \mu(z) \, dz}$

where Z_{crisp} is the output value, the crisp output value obtained by defuzzification, and μ(z) is the membership function of the fuzzy set at a particular value of z. So, this corresponds to

So, y* is called the crisp value in this case, where z is crisp. So, this is the combined or the distribution of the output. From this distribution, we get the crisp value by the center of gravity method using this expression. So, the results obtained by this implementation of the Mamdani fuzzy controller for a two-wheel mobile robot give this type of well-accurate trajectory tracking performance. This plot shows the error plot, considering both the distance error and the orientation error, converging precisely. Finally, we have the angular velocities and the linear velocities associated with the robotic system. We can now see the demo, the experimental demo of the robotic system in tracking the system. So, you can see here this is the demo of the mobile robot tracking a circular path obtained or controlled by this developed fuzzy controller.

Next, we have a Lissajous pattern tracked by this robotic system through this Mamdani fuzzy controller, and then this. Now, coming to the final portion of today's lecture, which is the stability analysis of a fuzzy logic controller system using the Lyapunov function. So, now consider we can understand this topic based on an example, considering a scalar non-linear system given by $\dot{x} = -x^3 + u$. So, the objective of this problem is to design a fuzzy logic controller so that the plant is stabilized around the origin, x = 0. The solution to this problem starts with selecting a Lyapunov function candidate, say $V = \frac{1}{2}x^2$.

So v dot becomes x x dot. Substituting the value of x dot here, we have v dot being minus x power 4 plus x u. We know that The Leibniz function candidate is a positive definite function, a scalar positive definite function. So V dot we want to be negative definite so that the system is stable. So we define two rules that will make V dot negative. So we consider, as per the expression of V dot, if x is negative then U is positive.

If x is positive then U is negative. So accordingly, we have the control input as per the condition of x. If x is positive then U is negative, and if X is negative then U is positive. So based on this condition of U, we have the system response of this one-dimensional system converging smoothly, and we can observe the stability of the system. So, coming to the concluding part of this talk, this lecture, we can say that we have introduced a fuzzy logic controller in that there are two types: Mamdani type and TS fuzzy model type.

The Mamdani type, which is a direct adaptive control type, we have seen the fundamentals of Mamdani type, the basic strategy of this Mamdani type fuzzy controller with an example of a two-wheel mobile robot tracking a circular trajectory as well as a Lissajous pattern, and then we have also seen the stability analysis of a fuzzy controller using a simple one-dimensional non-linear system. Thank you so much for your patience. In the next class, we will be seeing the TS fuzzy model along with fuzzy PI PD controllers. Thank you very much.