Exercise: A Fuzzy Controller for the Pole Balancing Problem

Advanced Control lecture
at Ecole Centrale Paris

Anne Auger and Dimo Brockhoff

`firstname.lastname@inria.fr`

Jan 18, 2013

Abstract

After implementing the pole balancing problem and getting an intuition about how difficult it is to design the linear controller for it, we will now apply fuzzy logic to create a controller that does not use all possible input measurements of the cart.

Again, please keep your code for the next exercises!

1 Getting familiar with fuzzy implications

Let the temperature T be a fuzzy variable that can belong to the fuzzy set "hot" and to the fuzzy set "moderately hot" the membership functions of which are defined as follows:

$$\begin{aligned}
\text{Hot}(T) &= \begin{cases} 0 & \text{if } T < 25^\circ \\ \frac{T-25}{10} & \text{if } 25^\circ \leq T \leq 35^\circ \\ 1 & \text{if } T > 35^\circ \end{cases} \\
\text{ModeratelyHot}(T) &= \begin{cases} \frac{T-20}{5} & \text{if } 20^\circ \leq T \leq 25^\circ \\ -\left(\frac{T-30}{5}\right) & \text{if } 25^\circ \leq T \leq 30^\circ \\ 0 & \text{else} \end{cases}
\end{aligned}$$

Further, let us define the fuzzy set “Crowdedness” of an ice-cream shop as

$$\text{Crowdedness}(C) = \begin{cases} 0 & \text{if } C < 25 \\ \frac{C-25}{25} & \text{if } 25 \leq C \leq 50 \\ 1 & \text{if } x > 50 \end{cases}$$

where C is the number of customers in the shop.

a) Draw/Plot the membership functions of the three above fuzzy sets. Interesting MATLAB functions to look at: `trimf`, `trapmf`, `plot`, `hold on/off`

b) Find the implication of the rule

```
IF the temperature is hot OR the temperature is
moderately hot THEN the ice-cream shop is crowded.
```

with respect to Mamdani’s rule and Larsen’s product rule if the temperature is 28°. In other words, how crowded is the ice-cream shop at a temperature of $T = 28°$ according to the above rule? To this end

I) plot first the membership function of the left-hand side of the implication (see `max` and `min`),

II) then write a function for each of the implication rules (Mamdani/Larsen), and third

III) evaluate the left-hand side membership function at the right temperature (see e.g. `evalmf`) and apply the two implication rules to find out how crowded the ice-cream shop is (plot the resulting membership functions).

IV) Finally, use the centroid defuzzification[1] to get a “crisp” answer whether the ice-cream shop is crowded or not. In which way does it matter which implication function you use?

[1] It might be helpful to look at the `defuzz` function from the Fuzzy Logic Toolbox in MATLAB.

2 A fuzzy controller for the pole balancing problem

Review the pole balancing problem from the previous exercise. All parameters and variables are essentially the same as before except that, for simplicity, we loose the constraints of the track length to $h = \pm 100$m and of the pole failure angle to $r = \pm\pi/2$ (90°). In this exercise, the controller does not have access to the direct measurements of all four input values (value and derivative of the position x of the cart and of the angle θ of the pole). Instead, we allow only for the two inputs θ and $\dot{\theta}$ for the moment.

a) Think about how to fuzzify the inputs and outputs of a fuzzy controller for this restricted pole balancing problem. As a recommendation, use three linguistic variables or membership functions per input and five or seven for the output.

b) Define the membership functions according to your fuzzification. For simplicity (and for the debugging), the sole use of triangular and trapezoidal membership functions is highly recommended.

c) Define the rule matrix for your fuzzy controller.

d) Implement the fuzzy controller with the help of MATLAB's Fuzzy Control Toolkit. Use Mamdani's rule for the implications and the centroid defuzzification.

   I) Start the Fuzzy Control Toolkit by typing `fuzzy` in MATLAB. In the upper half of the new window, you see an abstract view of your system with the inputs on the left, the implication rules in the middle, and the outputs on the right. Add a second input variable with choosing `Edit-->Add Variable...-->Input` from the menu. Advice: Save your fuzzy system regularly by choosing `File-->Export-->To File...` from the menu.

   II) By double-clicking on the input or output variables, a new window opens where you can name and enter your membership functions. The menu allows to add or delete functions. Change the membership functions' shape according to you above design choices by editing the values in the `Params` field or using the mouse.

III) In the initial FIS editor window, double-click on the white middle block with “mamdani” written in the brackets to enter your rules of the rule matrix.

IV) If you have not done yet, save your fuzzy controller as a `.fis` file that you can later on use within your script from last week to get the response for a certain input.

e) Test your controller on the pole balancing benchmark from the previous exercise. To this end, use the functions `readfis` and `evalfis` to load your controller and evaluate it, e.g. like this:

```
myController = readfis('NAMEOFYOURCONTROLLER');
F = evalfis([angle_theta angle_velo], myController);
```

Does your fuzzy controller allow to stabilize the pole?

f) What is not possible with the current problem formulation and why?

3 A better fuzzy controller

In the following, we would like to also stabilize the cart position around a value of $x = 0$. To this end, update your controller from above by adding the position of the cart x as an input and designing new rules controlling the output.

4 Non-Mandatory Questions

If you have more time, you can further play around with the system.

a) Is the centroidal defuzzification giving better results than other methods?

b) What about the influence of other parameters of the sytem such as the implication rule, the simulation accuracy, or the initial cart and pole positions?

c) Can you come up with a nice illustration of the system behavior in which you can follow the trajectory of the pole and the cart during the simulation in one single plot?