159.302 **Artificial Intelligence**
Assignment #2

Fuzzy Controller for the Inverted Pendulum Problem

Maximum number of members per group: 3 students
Deadline for submission: October 11

Instructions

Your task is to implement and calibrate a fuzzy controller (Zero-order Sugeno Fuzzy Inference System) for balancing an inverted pendulum system. A written report detailing your system design and characterisation of its performance must accompany your program submission. In addition, fill-in the

A start-up program using second-order derivative physics equations and simple graphics library are provided, simulating the complete dynamics of the cart-pendulum system. In addition, it also includes a function for collecting data points for plotting a control surface, and a fuzzy logic engine that you can utilise to implement a complete fuzzy controller. A tutorial on how to use the engine is provided in the lecture slides (Lecture - Fuzzy Logic Engine.pptx – we discussed this in the lectures).

Details of the requirements:

Part 1: Fuzzy System Design

1. Use the following inputs (combine the inputs together, as suggested in Yamakawa's paper):
    - Combined inputs (Yamakawa)
      X = (**A** * **theta**) + (**B** * **theta_dot**)
      Y = (**C***x**) + (**D** * **x_dot**)

    - Definition of inputs:
      x – position of the cart
      x_dot – horizontal velocity of the cart
      theta – angle of the pole with respect to the vertical
      theta_dot – angular velocity of the pole
      A, B, C and D are positive constants; they are empirically defined

2. Use the fuzzy control rules defined by Yamakawa.
    - Reference: Takeshi Yamakawa's paper, titled "A Fuzzy Inference Engine in Nonlinear Analog Mode and Its Application to a Fuzzy Logic Control". Refer to page 517 of his paper to see what inputs were used in his design. This research paper is available for download in our Stream website.
    - Yamakawa defined 13 rules to solves the control problem. Optionally, you may extend Yamakawa's rules to 25 rules.

3. Define the rule outputs associated with each of the fuzzy rules (e.g. NL = -100, PL = 100, etc.). Note that we are implementing a Zero-Order Sugeno Fuzzy Inference System, and so the rule outputs are constants.

4. Define the fuzzy sets corresponding to the linguistic terms in your fuzzy rules.
    - The fuzzy sets need to be defined according to the range of possible values for the input variables.
    - Example: Input range of input variables:
        - **X**: [-4.0 – 4.0]

- **Y**: [-4.0 – 4.0]

5. Implement the fuzzy sets as membership functions in your program. You may use any of the membership functions we discussed in class. (The fuzzy engine contains the implementation of trapezoidal membership functions, if you want to use it.)
6. Define the defuzzification method for your system. (The fuzzy engine contains a centroid defuzzification method.)
7. Incorporate your fuzzy controller into the start-up program provided. (Tips on where to insert codes are provided in the start-up codes)
8. Note that in the start-up codes, there are blocks of statements that should not be modified as they are part of the implementation of the dynamics of the system. There are comments in the codes that identify these blocks of codes.
9. It is up to you to write and add any functions, classes or data structures that you may require to complete the system.
10. Your simulation system should demonstrate that the fuzzy controller is able to balance the inverted pendulum, given an initial pole angle and position.

Part 2: System Calibration

11. Calibrate your fuzzy controller by modifying the rules, shape of membership functions, etc. until it is able to balance the pendulum without exceeding the boundaries of the platform. Aim for a control solution that can balance the pendulum in a smooth fashion and can bring the cart-pole system at the centre of the platform at zero-degree angle with respect to the vertical axis.

Part 3: Results and Analysis

12. Generate the control surface data points using void ***generateControlSurface***().
    - The control surface comes with the following dimensions: angle of pole, angular velocity of pole, and Force calculated by the fuzzy controller.
    - Calling ***generateControlSurface***() will apply all the necessary physics equations to update the state of the world. It will also store the data points into a text file (data_angle_vs_angle_dot.txt) that you can use later for 3D surface plotting using MS-Excel.
    - Note that a statement calling ***generateControlSurface***() is already in place inside the main function
13. Plot the control surface using MS-Excel. Include the Excel file in your assignment submission.
    - MS-Excel requires a specific format for the tabulation of data points for 3D surface generation. Therefore, to plot a 3D surface, make sure that you delete the first zero value on the first row (upper-left corner) of the data points in the worksheet. The zero value is only there to align the columns properly, as required by the tabulation of data points by Excel.

14. Test the fuzzy controller system by setting the initial angle of the pole with different values.

    The bigger the initial angle is, the more challenging the problem becomes for the controller. Record the biggest angle magnitude that your system can successfully handle in the provided **checklist.xlsx** file.

    Characterise your control system by answering the following questions:
    - At x=1, what is the largest initial angle (most positive) and smallest (most negative) initial angle that your fuzzy controller can handle, without causing the cart-pole system to exceed the boundaries of the platform [-2.4m, 2.4m], and without dropping the pole on the ground?
        - You can find the answer to this question by experimenting with your fuzzy controller. Type the initial angle of the pole on the command prompt window, then press the <Enter> key (this is already in place in the start-up codes). The inverted pendulum simulation will run afterwards.
        - Note that the program automatically converts the input angle to radians.
    - For how long can your fuzzy controller successfully balance the pendulum?

Part 4: Documentation

- **Fuzzy Logic Controller**: Discuss the complete fuzzy system that you have designed
    - Show the details of the inputs, fuzzy rules, fuzzy sets, rule outputs and defuzzification method
    - Follow the algorithm documentation guide provided. Please see ALGORITHM DOCUMENTATION GUIDE.docx
    - Submit this as a type-written report (e.g. MS-Word/ OpenOffice/pdf file).
- **Control Surface:** Show the plot of the (3D) control surface (angle vs. angular velocity)
    - Submit the actual MS-Excel file.

Checklist: Please complete the **checklist.xlsx** file. Name your Excel file using the following format: **checklist_ID.xlsx**

Example: (e.g., **checklist_20298765.xlsx**).

ID number

Criteria for marking

- Documentation – 15%
- Fuzzy Logic system implementation and calibration – 85%

Submission Requirements:

1. Complete source code of your fuzzy controller and simulation system (*.cpp, *.h, makefile, etc.)
2. Checklist file (MS-Excel file).
3. Fuzzy System Documentation: (MS-Word/OpenOffice/pdf)

4. Control Surface (MS-Excel file).

Nothing follows.