

Assignment 5: Fuzzy Logic Report (Group 25)

1501CS54, 1821EE13, 1821ME18

The code is divided into 2 parts :

1. backend.py
2. frontend.py

Where **`backend.py`** implements the logic of fuzzification and find the current from the profile, then this value is transferred to the function in **`frontend.py`** which handle the modification of pendulum position and its display on the GUI.

Backend

Following contains a description of the various function that perform various operations from making profiles to doing the fuzzification and defuzzification.

- **def profile(epsilon1, epsilon2, epsilon3, x):** This is responsible for providing the membership of x to a profile which is defined by the parameters epsilon1, epsilon2, epsilon3, it outputs a membership value between 0 and 1
- **def fuzzication(epsilon: 'array', x):** This is responsible for fuzzification of a given values of x for different literal variables which include 'small negative', 'zero', 'small positive'.
- **def rules(theta, omega, epsilon_theta: 'array', epsilon_omega: 'array'):** This holds the rule for selection of current profile with a given member ship value
- **def defuzzify(epsilon: 'array of epsilon for curr', y):** Defuzzifier for a given profile and gives the area and the centroid of the trapezoid.
- **def compute_current(theta, omega, epsilon_theta, epsilon_omega, epsilon_curr):** Responsible for collecting current memberships from all rules and defuzzifying them then taking their weighted centroid to get a final current value.

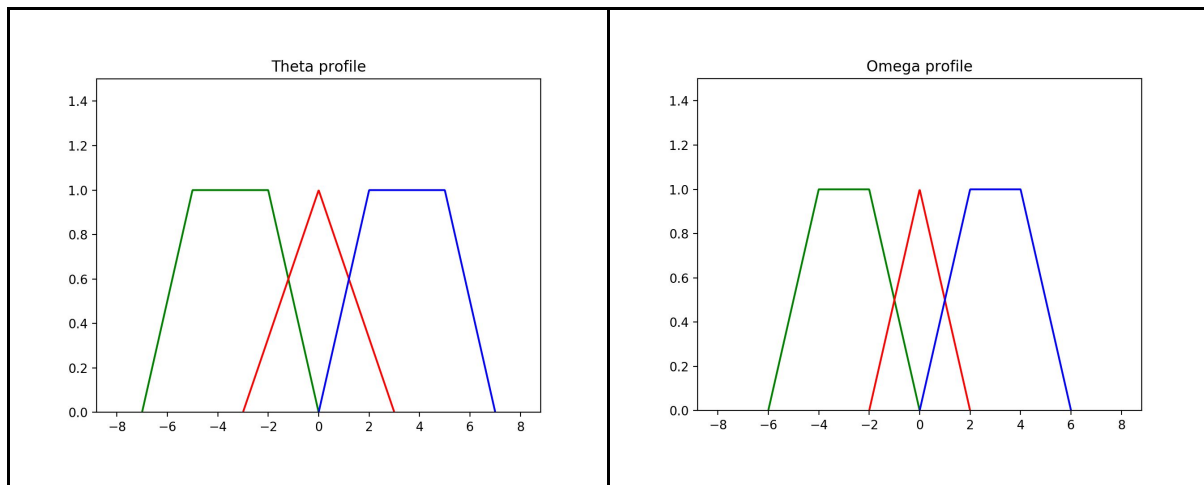
Frontend

This takes in the current value and uses the equations of motion to modify and get the new position and velocity of the pendulum.

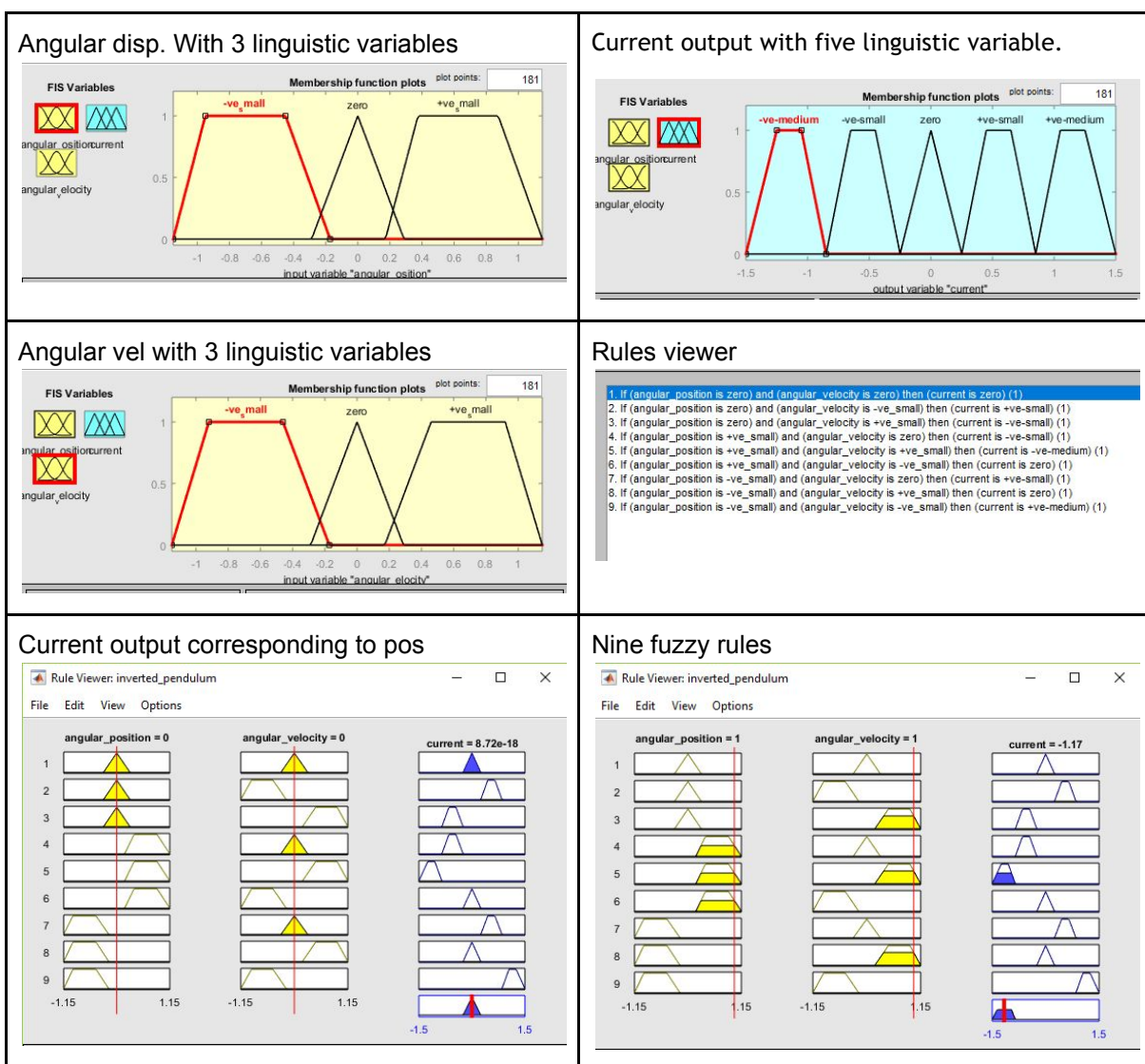
Requirements

- Python3
- Pygame

Profiles



Profile from Matlab



Acknowledgements : [Comments by 1501CS54] The python code was developed in collaboration with the members of group 17 i.e. 1501CS02, 1501CS36, 1501CS40