**AEEM 5117/6117: Intelligent Robotics**

**Course Project #2**

**4. b) Design and implement an intelligent control system for the robots to accomplish the task. The control strategy must have learning (training) capability.**

**Solution:**

Use Fuzzy control and Genetic Algorithm.

1. Design Fuzzy system

Let one robot control the position of the rod and the other one control the orientation of the rod, we could design the fuzzy controller for each robot as below:

* The Fuzzy Controller for both robots has three inputs:
  + X component of the position of the robot
  + Y component of the position of the robot
  + Angle of the rod
* Outputs for position control robot are:
  + Change in X component
  + Change in Y component
* Out for orientation control robot is:
  + Change in angle of rod

As controller of each robot only takes the position of itself (which could be calculated as the length, the center position and the orientation of the rod are known to both robots) and the orientation of rod, they do not involve its partner’s information.

For each input and output, three member functions with undecided parameters were assigned and the inference rules are also undecided. Genetic algorithm will be used to tune the parameter during training the system.

So, the fuzzy sets and member functions for robots are defined as below

|  |  |
| --- | --- |
| Robot 1: Control Position   * 3 inputs: (robot position)   + T(x) = {mf1, mf2, mf3}   + T(y) = {mf1, mf2, mf3}   + T(θ) = {mf1, mf2, mf3} * 2 outputs: (position change)   + T(dx) = {mf1, mf2, mf3}   + T(dy) = {mf1, mf2, mf3} | C:\Users\Yufeng Sun\AppData\Local\YNote\data\syf_more@163.com\3768af03c9f544268d39e00052da554b\clipboard.png |

|  |  |
| --- | --- |
| Robot 2: Control Orientation   * 3 inputs: (robot position)   + T(x) = {mf1, mf2, mf3}   + T(y) = {mf1, mf2, mf3}   + T(θ) = {mf1, mf2, mf3} * 1 output: (orientation change)   + T(dθ) = {mf1, mf2, mf3} | C:\Users\Yufeng Sun\AppData\Local\YNote\data\syf_more@163.com\97742dc40586472988f109adbb3aab0c\clipboard.png |

1. Physical constraint

The two robots must maintain the distance between them to be the length of the rod. To achieve this with only considering the geometric representation, once the position control robot moves, we let the other robot move the same distance in same direction, while the orientation control robot moves, it must move along the circle path which center is position of other robot and the radius is the length of the rod.

They might be other designs for complying this physical constraint.

1. Cost function for Genetic Algorithm

The cost function is the key for using GA to tune the parameters of the fuzzy controller. The GA will try to find the minimum cost during training the sample data. So, the cost of the job could be the step of successfully moving the rod out of the room. While in each step, the position of each robot will be input to its own fuzzy controller and updated with the output change.

By setting a maximum step for the job (to avoid dead loop case), there could be some unsuccessful job. In this case, we need to add certain penalty value to the cost, so that the GA could identify this job as a bad one.

Another problem we should consider is that the robots and the rod might hit the wall, in real world, we know how to manage this situation, while in this simulation, we must design a solution for this problem. You could stop the job or change the robots’ position and let the job continue, and with both solutions, you might need to give a penalty as neither of them is expected.

1. Parameters for Genetic Algorithm

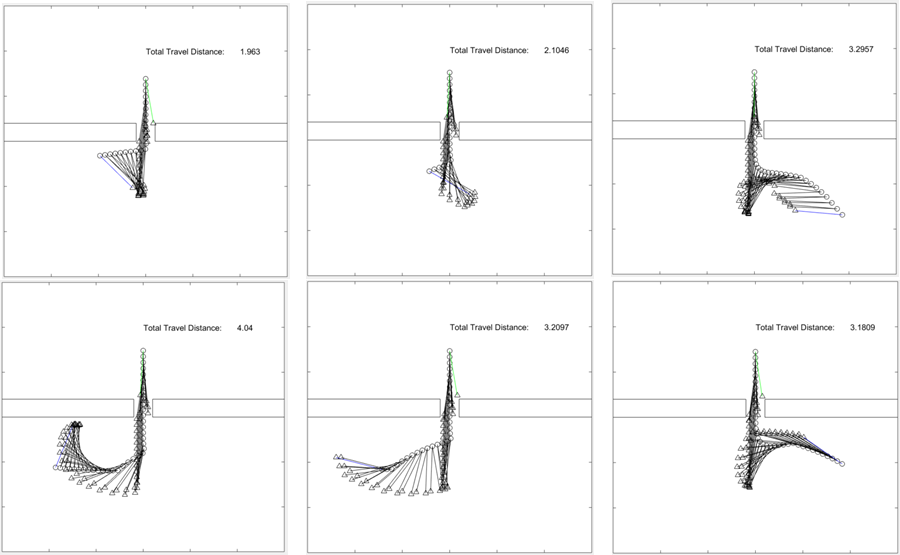
As we have 3 inputs for each fuzzy controller and 2 outputs for position controller and 1 output for orientation controller, in total we will have 99 parameters for GA to tune. Including 18 parameters for member functions and 81 parameters for rules.

We will give generation number as 100 and population size as 100 for GA, but these values could be varied in different proposals.

1. Simulation in MATLAB

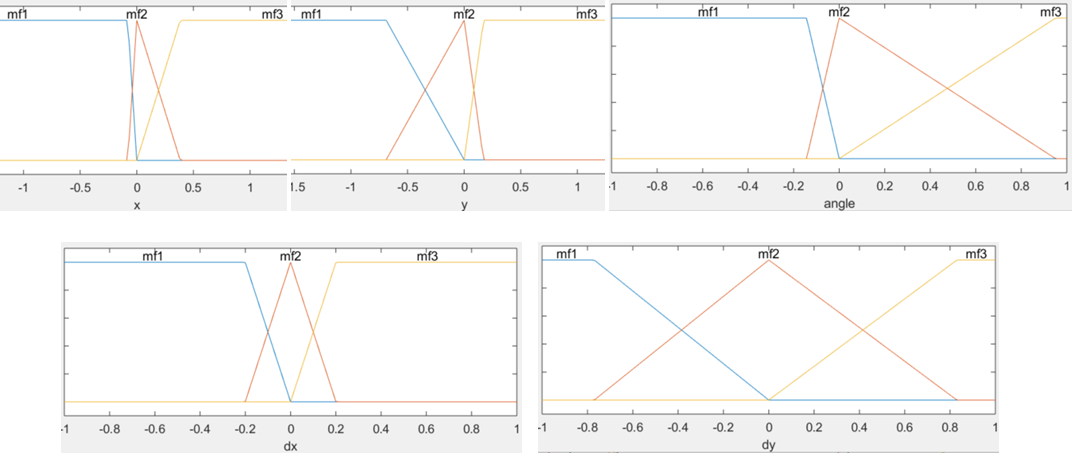
Given width, depth of the opening as 0.2, 0.2 respectively, and the length of rod as 0.5 in a room where the origin (0, 0) is located at the center of opening. the inside room is defined as (y < 0) and outside room is defined (*y > 0.2*), the opening is defined as *(-0.1 <= x <= 0.1 && 0 <= y <= 0.2).*

Given different samples of position of rod in the room for training, and the trained system behaves as expected during test.



**(blue line: initialize position of the rod; green line: end position of the rod; circle: position control robot; triangle: orientation control robot)**

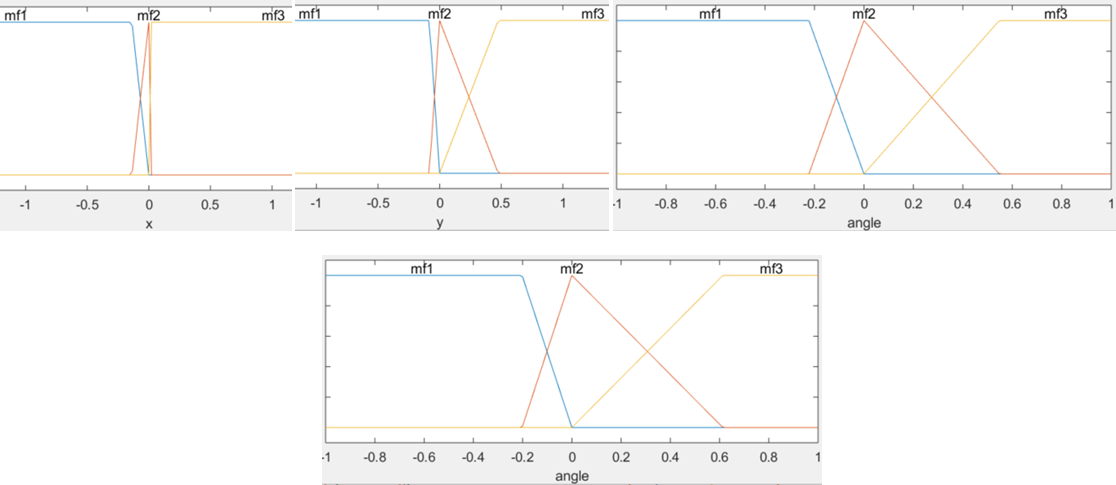
The tuned member functions for position controller are



And the tuned rules are



The tuned member functions for orientation controller are



And the tuned rules are



**Future Work**

The training is pretty time consuming due to the 99 parameters, one of the future works is to reduce the parameters. It seems that the input “y” for both fuzzy controllers could be simplified with less member functions.