Fuzzy Control Project (100 points)

You are to design a fuzzy controller for the following robotics problem. You may use the Matlab fuzzy toolbox or write your own Matlab code (but I highly recommend you use the fuzzy toolbox). Cite sources where necessary.

You are a robot in an arena (assume that the arena has no boundaries). The two control (action) parameters you have are your rotation acceleration $\ddot{\theta}$ and your forward-drive acceleration $\ddot{r}$ (see the drawing below). The inputs to your system are the current rotational and forward-drive velocities $(\dot{\theta}, \dot{r})$, the Boolean inputs of whether your rotational and radial accelerations are positive or negative, and the relative angle ϕ_g and distance d_g towards the center of a goal, which is parameterized by two coordinates (x_i, y_i) and a radius denoting the target area. There will be multiple goals that your robot must enter; however, you cannot achieve the next goal until you pass through the current goal (i.e., the goals must be met sequentially). The time-step of your controller is 100 milliseconds (i.e., you can compute inputs and outputs every 100 msec).

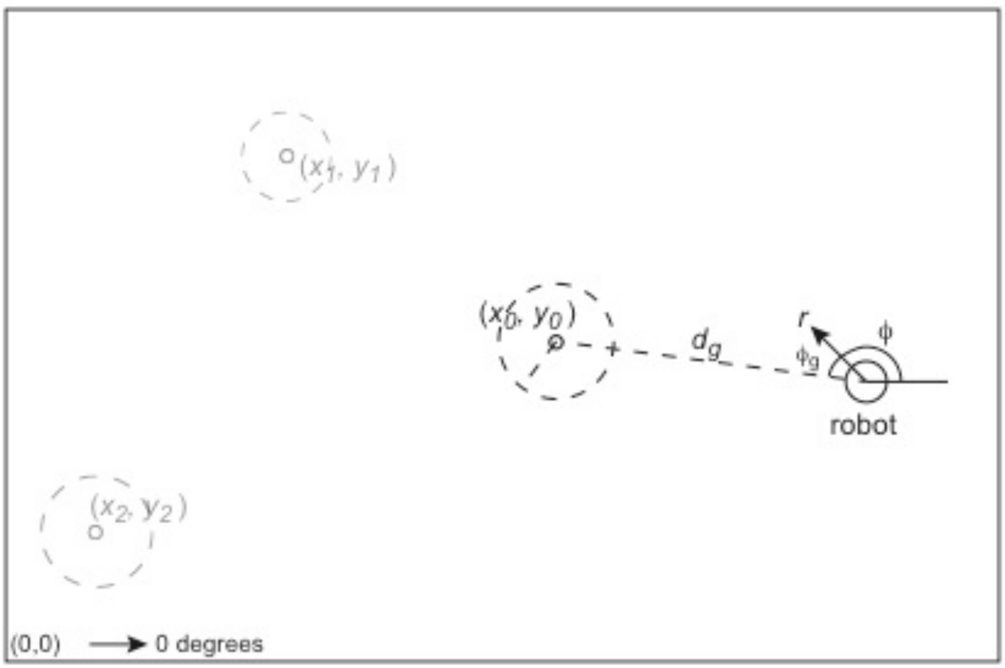

Fig. 1. Robot arena diagram.

Specifications of the robot:
Range of radial velocity – [0, 1] m/sec (robot only goes forward)
Range of radial acceleration – ± 0.2 m/sec^2
Range of rotation velocity – ± 60 deg/sec
Range of rotation acceleration – ± 20 deg/sec^2
Spatial size – none (assume it is a point)

Specifications of arena:
Origin is at (0,0) (bottom left) and 0 degrees is positive *x*-axis

Robot always starts at (0,0) with velocity = (0 deg/sec, 0 m/sec) and angle = 0 degrees, acceleration state = (0 deg/sec^2, 0 m/sec^2). So, it is pointed to the right and at a stand-still. Goals are considered met when the robot is within the radius of the goal at the end of a time-step. Goals are presented sequentially (only one goal is known to the robot at a time)

1) (6 points) Design a *fuzzifier* for your robot assuming the following inputs, relative angle ϕ_g between the forward motion vector of the robot and the center of the current goal and the distance d_g between the robot and the center of the current goal, and the current velocity state $(\dot{\phi}, \dot{r})$. See Fig. 1 (Positive angles are to the left of the robot, negative angles are to the right of the robot). Assume that an expert on your goal sensing system told you that the uncertainty on the angle sensor is about ± 5 degrees and the uncertainty on the distance detector is $0.1 d_g$ (i.e., the error is distance dependent; the farther the goal is from the robot, the more the error). Describe the rational behind your fuzzifier and plot a fuzzified input (the FACT) for the following inputs in degrees and meters:
1a) $(\phi_g, d_g, \dot{\phi}, \dot{r}) = (10{,}10{,}0{,}0.5)$
1b) $(\phi_g, d_g, \dot{\phi}, \dot{r}) = (-30{,}50, -50{,}1)$

2) (7 points) Map each input space (angle, distance, angular velocity, radial velocity) to a set of 7 fuzzy sets (use the sets we talked about in class). (Hint: you may have to have a scaling parameter for the distance sets that depends on the size of the arena). Plot your sets on four plots: one for each input. *Label your axes and the linguistic "name" of each set.*

3) (7 points) Map each output space (rotational acceleration, forward acceleration) to a set of 7 fuzzy sets (use the sets we talked about in class). Plot your sets on two plots: one for rotational acceleration and one for forward acceleration. *Label your axes and the linguistic "name" of each set.*

4) (10 points) Design a set of fuzzy rules for your system. Show your rules. Describe the rational behind your rules (you don't need to talk about each one, but describe the overall behavior you are going for).

5) (5 points) The last element for a fuzzy control system is the defuzzifier. Design code that takes fuzzy outputs for each acceleration output and turns them into real numbers. Describe your method (show an example). What method did you use? And why?

OK. You should have most of the elements for a fuzzy controller now. You only need to provide the connecting elements (mainly the Fuzzy Inference Engine, but that is just a bunch of *min*s and *max*s). Develop code that ties everything together, taking the crisp inputs of (ϕ_g, d_g), fuzzifies them, fires the rules in the fuzzy rule base, collects the rules together into a fuzzy output, defuzzifies the output, and then determines the next state of the robot by the defuzzified action $(\ddot{\phi}, \ddot{r})$. Hint: the Fuzzy Logic toolbox in Matlab does everything you need for parts 1)-5).

6) (10 points) Plot the location at each time-step of your robot for the following scenarios; i.e., show a plot of the path that the robot takes in one plot. For each scenario, show the total number

of time-steps necessary to reach all goals; show in a table. Don't use a different controller for each scenario. Try to find something that works best overall.

6a) One goal at (10,10) meters with a radius of 3 meters.
6b) One goal at (10,10) meters with a radius of 0.3 meters.
6c) Two goals: First goal at (20,20) with a radius of 1 meter; second goal at (5,20) with a radius of 1 meter.
6d) Three goals: (0,10) radius 4; (0,0) radius 1; (20,0) radius 5
6e) Five goals: (5,5) radius 0.5; (4,4) radius 0.5; (6,6) radius 0.5; (3,6) radius 0.5; (6,3) radius 0.5

(You may find that you need to edit your rules so that you get your robot to work or get to the goals faster. If you do this, describe your what happened and what you did to fix it. Show examples.)

7) (10 points)
7a) Given your results, what would you change / add to your system (such as other inputs) to get better performance?
7b) Do you really need 7 fuzzy sets for each input / output? If you reduce the number of input sets to 3 for each input and the number of output sets to 3 for each output, do you think you would get similar results? (Do an experiment that compares this to your results in part 6.)
7c) Write a short summary about your experiences with this project. What worked? What didn't?

8) (20 points) Submit your FIS file as FCproject.fis (EXACTLY THAT FILENAME). We will run your robot for five scenarios, two of which will be taken from the list of scenarios in part 6); the other three are unknown to you. We will add up the number of time-steps needed to accomplish the goals and score your robot according to the following rules:
1 point for each scenario in which the robot reached the goals within a reasonable amount of run-time (meaning that if your robot gets stuck, we will stop running it)
"about 5" points for "less than average" performance
"about 10" points for "average" performance
"about 15" points for "superior" performance
(We have a fuzzy inference system which will return the number of points you receive based on the number of time-steps your robot takes as compared to Dr. Havens's "basic" controller. His "basic" controller took the following number of time-steps for each scenario in 7. a) 166, b) 551, c) 622, d) 533, e) 1227)

9) (25 points) Write a clear and concise report that presents the solution to each of the problems above and describes the overall project. These points will be awarded for your writing and the appearance of the report. Submit as a PDF.