Autonomous robot obstacle avoidance using a fuzzy logic control scheme

William Martin

Submitted on December 4, 2009

CS311 - Final Project

1. INTRODUCTION

One of the considerable hurdles to overcome, when trying to describe a real-world control scheme with first-order logic, is the strong ambiguity found in both semantics and evaluations. Although one option is to utilize probability theory in order to come up with a more realistic model, this still relies on obtaining information about an agent's environment with some amount of precision. However, fuzzy logic allows an agent to exploit inexactness in its collected data by allowing for a level of tolerance. This can be especially important when high precision or accuracy in a measurement is quite costly. For example, ultrasonic and infrared range sensors allow for fast and cost effective distance measurements with varying uncertainty. The proposed applications for fuzzy logic range from controlling robotic hands with six degrees of freedom[1] to filtering noise from a digital signal.[2] Due to its easy implementation, fuzzy logic control has been popular for industrial applications when advanced differential equations become either computationally expensive or offer no known solution. This project is an attempt to take advantage of these fuzzy logic simplifications in order to implement simple obstacle avoidance for a mobile robot.

2. PHYSICAL ROBOT IMPLEMENTATION

2.1. Chassis and sensors

The robotic vehicle's chassis was constructed from an Excalibur EI-MSD2003 remote control toy tank. The device was stripped of all electronics, gears, and extraneous parts in order to work with just the empty case and two DC motors for the tank treads. However, this left a somewhat uneven surface to work on, so high-density polyethylene (HDPE) rods were used to fill in empty spaces. Since HDPE has a rather low surface energy, which is not ideal for bonding with other materials, a propane torch was used to raise surface temperature and improve bonding with an epoxy adhesive.

Three Sharp GP2D12 infrared sensors, which have a range of 10 to 80 cm, were used for distance measurements. In order to mount these appropriately, a 2.5 by 15 cm piece of aluminum was bent into three even pieces at 135 degree angles. This allows for the IR sensors to take three different measurements at 45 degree angles (right, middle, and left distances). This sensor mount was then attached to an HDPE rod with mounting tape and the rod was glued to the tank base with epoxy. Since the minimum distance that can be reliably measured with these sensors is 10 cm, the sensors were placed about 9 cm from the front of the vehicle. This allowed measurements to be taken very close to the front of the robot.

2.2. Electronics

In order to control the speed of each motor, pulse-width modulation (PWM) was used to drive two L2722 op amps in open loop mode (Fig. 1). The high input resistance of these ICs allow for the motors to be powered with very little power draw from the PWM circuitry. In order to isolate the motor's power supply from the rest of the electronics, a 9.6 V NiCad battery was used separately from a standard 9 V that powered everything else. The continuous current demand on the op amps led to a small amount of overheating during continuous operation. This was remedied by adding small heat sinks and a fan to the forcibly disperse heat.

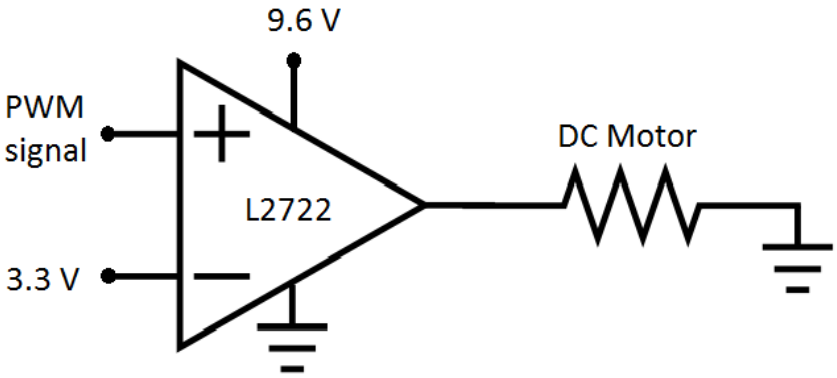

Fig. 1. The control circuit used for driving each DC motor. Note that the PWM signal was between 0 and 5 V.

2.3. Microcontroller

Computation was handled by an Arduino Duemilanove board with an ATmega328 microcontroller. The board has low power requirements and supports PWM signals with no modifications. In addition, it has a large number of versatile inputs which allowed for rapid prototyping of the control circuit. The programming language used for the Arduino is C-style and based on the Wiring language. This board provided an easy and low-cost platform to build the robot around.

3. FUZZY CONTROL SCHEME FOR OBSTACLE AVOIDANCE

In order to apply fuzzy logic to the robot's motion, a control scheme had to be developed to interpret measured distances. While the final algorithm depended critically on the geometry of the robot itself and how it operates, some basic guidelines were followed. Similar research projects provided both simulation results and ideas for implementing fuzzy control.[3,4,5]

3.1. Membership functions

Three sets of membership functions were created to express degrees of membership for distances, translational speeds, and rotational speeds. This made for a total of two input membership functions and eight output membership functions (Fig. 2). Triangle and trapezoidal functions were used exclusively since they are quick to compute and easy to modify. Keeping computation time to a minimum was essential so that many sets of data could be analyzed every second (approximately one every 40 milliseconds). The distance membership functions allowed the distances from the IR sensors to be quickly "fuzzified," while the eight speed membership functions converted fuzzy values back into crisp values.

3.2.Rule base

Once the input data was fuzzified, the eight defined fuzzy logic rules (Table I) were executed in order to assign fuzzy values for translational speed and rotation. This resulted in multiple values for the each of the fuzzy output components. It was then necessary to take the maximum of these values as the fuzzy value for each component. Finally, these fuzzy output values were "defuzzified" using the max-product technique and the result was used to update each of the motor speeds.

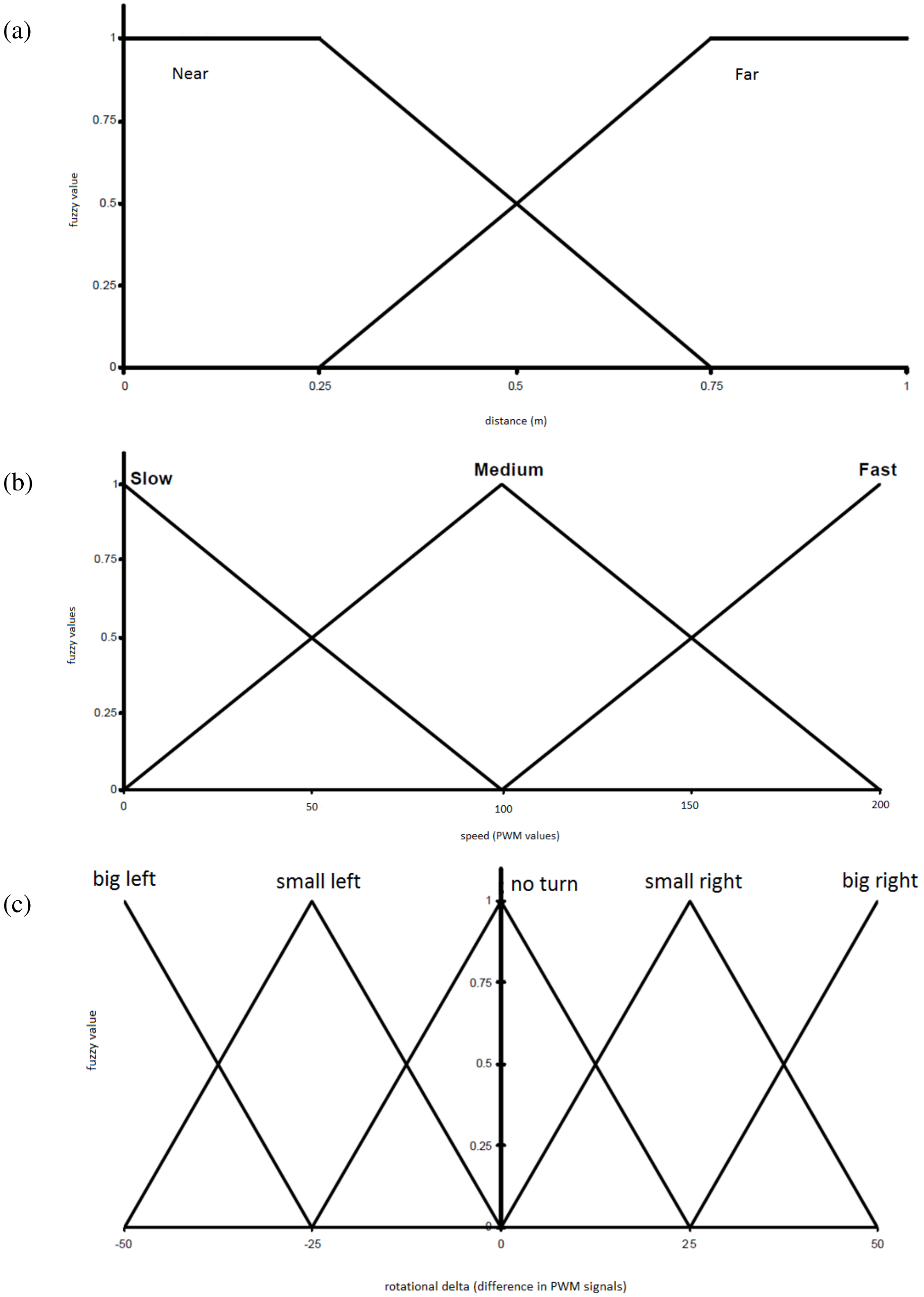

Fig. 2. The membership functions used for (a) distance, (b) translation speed, and (c) rotational speed. These functions were adapted from similar work done in reference 3.

Table I. The fuzzy logic rule base used for the control scheme

Rule	Left	Center	Right	Speed	Rotation
1	Near	Near	Near	Slow	Right big
2	Near	Near	Far	Medium	Right small
3	Near	Far	Near	Slow	No turn
4	Near	Far	Far	Fast	Right small
5	Far	Near	Near	Medium	Left small
6	Far	Near	Far	Slow	Right big
7	Far	Far	Near	Fast	Left small
8	Far	Far	Far	Fast	No turn

4. RESULTS

The fuzzy control scheme allowed for the robot to quickly respond to obstacles it could detect in its environment. This allowed it to follow walls and bend around corners decently without hitting any obstacles. However, since the IR sensors' measurements depended on the geometry of surrounding objects, there were times when the robot could not detect obstacles. For example, when the IR beam hit a surface with oblique incidence, it would reflect away from the sensor and not register as an object. In addition, the limited number of rules used may have limited the dynamics of the robot's responses. Some articles suggest as many as forty rules[6] should be used, while others tend to present between ten and twenty. Since this project did not explore complex kinematics or computational simulations of the robot, it is difficult to determine exactly how many rules should be used. However, for the purposes of testing fuzzy logic as a navigational aide, the eight rules were sufficient. Despite the many problems that IR and similar ultrasonic sensors have with reliably obtaining distances, the robustness of fuzzy logic was frequently able to prevent the robot from running into obstacles.

5. CONCLUSION

There are several easy improvements that could be made to future iterations of this project in order to improve the robot's performance. The most dramatic would be to implement the IR or ultrasonic sensors on a servo so that they could each scan a full 180 degrees. However, this type of overhaul may undermine some of fuzzy logic's helpful simplicity. Another helpful tactic would be to use a few types of sensors so that data could be taken at multiple ranges. The IR sensors used in this experiment had a minimum distance of 10 cm, so anything in front of this could not be reliably detected. Similarly, the sensors had a maximum distance of 80 cm so it was difficult to react to objects far away. Ultrasonic sensors do offer significantly increased ranges at a slightly increased cost and response time. Lastly, defining more membership functions could help improve the rule base by creating more fine tuned responses. However, this would again increase the complexity of the system.

Thus, this project has successfully implemented a simple fuzzy control scheme for adjusting the heading and speed of a mobile robot. While it is difficult to determine whether this is a worthwhile application without heavily researching other methods, it is quite apparent that fuzzy logic affords a certain level of simplicity in the design of a system. Furthermore, it is a novel approach to dealing with high levels of uncertainty in real-world environments.

6. REFERENCES

[1] Ed. M. Jamshidi, N. Vadiee, and T. Ross, Fuzzy logic and control: software and hardware applications, (Prentice Hall: Englewood Cliffs, NJ) 292-328.

[2] Ibid, 232-261.

[3] W. L. Xu, S. K. Tso, and Y. H. Fung, "Fuzzy reactive control of a mobile robot incorporating a real/virtual target switching strategy," Robotics and Autonomous Systems, 23(3), 171-186 (1998).

[4] V. Peri and D. Simon, “Fuzzy logic control for an autonomous robot,” 2005 Annual Meeting of the North American Fuzzy Information Processing Society, 337-342 (2005).

[5] A. Martinez, E. Tunstel, and M. Jamshidi, "Fuzzy-logic based collision-avoidance for a mobile robot," Robotica, 12(6) 521–527 (1994).

[6] W. L. Xu, S. K. Tso, and Y. H. Fung, "Fuzzy reactive control of a mobile robot incorporating a real/virtual target switching strategy," Robotics and Autonomous Systems, 23(3), 171-186 (1998).