A Fuzzy Logic Based Navigation System for a Mobile Robot

Mark Leyden, Daniel Toal and Colin Flanagan
Department of Electronic & Computer Engineering,
University of Limerick,
Ireland.

mark.leyden@ul.ie, daniel.toal@ul.ie, colin.flanagan@ul.ie

Keywords: Mobile Robot, Fuzzy Logic, Obstacle Avoidance, Robot Navigation

Abstract

One of the problems associated with traditional behaviour based navigation systems for mobile robots is that of command arbitration. Having multiple behaviours which are all running concurrently leads to situations where several command outputs may be produced simultaneously. The decision about which output should be used to drive the robot is left to a separate command arbitrator. Using this method, however, does not always result in smooth motion control of the robot and may even cause the robot to move in a completely unintended direction. This paper describes the design of a fuzzy logic based navigation system for a mobile robot. The advantage of using fuzzy logic for navigation is that it allows for the easy combination of various behaviours' outputs through a command fusion process. The navigation system in this case consists of two behaviours - an obstacle avoidance behaviour and a goal seeking behaviour. The inputs to the fuzzy controller are the desired direction of motion and the readings from the sensor array. The outputs from each behaviour's rule base are integrated using the command fusion process and made crisp using a modified defuzzification technique. The end result is very smooth motion control of the robot.

1. Introduction

The traditional artificial intelligence approach to building a control system for an autonomous robot is to break the task into a number of subsystems. These subsystems typically include perception, world modelling, planning, task execution and motor control. The subsystems can be thought of as a series of vertical slices with sensor inputs on the left and actuator outputs on the right. The disadvantage of this approach, however, is that all of these subsystems must work correctly for the robot to function at all.

To overcome this problem, Brooks [1] developed an alternative behaviour based control system known as subsumption. In subsumption, the control task is broken down into a number of horizontal layers. Each layer consists of an individual competence or behaviour. Since each layer contains elements of all the vertical slices found in the classical system, each layer is capable of controlling the robot by itself. The behaviours of each layer perform one particular function; for example, moving from one particular location to another or avoiding contact with any nearby obstacles. Since the task of controlling the robot is broken down into a number of different behaviours, each behaviour does not have to tackle the whole problem of achieving the overall goal. This greatly simplifies the design of the architecture for the robot, since behaviours can be built up incrementally.

In subsumption, all of the behaviours run concurrently. Since many of the behaviours may produce outputs which are intended to be used by the same device, a motor for example, some sort of arbitration scheme must be provided to overcome the problem of conflicting outputs. If two behaviours both produce an output intended for the same device, the two behaviours will compete with each other for control. In subsumption, this arbitration scheme is achieved by arranging the behaviours into a hierarchy, in which a higher level behaviour can subsume control from a lower level behaviour.

From work that we have previously carried out on mobile robotics [2, 3, 4] using subsumption, we have found that on subsuming control, the controlling behaviour had no knowledge of the lower level behaviour's task. This has sometimes led to erratic behaviour under certain circumstances. To overcome this problem, the outputs of certain behaviours were fused using fuzzy logic techniques.

2. Limitations of Subsumption

One of the problems associated with subsumption is that the arbitration technique employed only allows a single behaviour to be active at any one time. While this is satisfactory in many situations, there are times when a combination of two behaviours is required. Take, for example, navigating towards a target and avoiding obstacles. Each of these could be implemented as a single behaviour each. So long as no obstacles are detected, the robot will gracefully head towards its target location. If an obstacle is detected, however, the obstacle avoidance behaviour becomes active and steers the robot away from the obstacle. The problem with this is that the obstacle avoidance behaviour has no knowledge about the target location, thus it could steer the robot in any direction to avoid the obstacle. In many situations, this may work perfectly well, but there are times when it may be desirable for the robot to steer in a direction which takes it closer to its desired path. This can be achieved by combining the output of the two behaviours, referred to as command fusion. The output from the path following behaviour and the obstacle avoidance behaviour are combined to produce a heading which takes it towards its target location while avoiding obstacles.

One method used to perform this task was developed by David Payton and Ken Rosenblatt [5]. In their system, instead of each behaviour outputting a single control value, they output a set of nodes. Each node corresponds to a possible control decision. A certain activation level is assigned to each of the nodes, which represents the confidence regarding the control decision. To combine the outputs of two behaviours, a weighted sum technique is used to combine the activation strengths of corresponding nodes.

3. Enhancing Subsumption

The navigation system employed on this robot is based upon a fuzzy logic control system. Both the path following and obstacle avoidance behaviours are implemented using a fuzzy logic based architecture. Each behaviour consists of a set of fuzzy control rules and a fuzzy inference module. The output from each behaviour is not be a crisp control value, but is instead a fuzzy set. These sets are then combined through a command fusion module and defuzzified to produce a crisp output value (Figure 1). The output of the path following behaviour produces a fuzzy set representing the desired turning direction while the output from the obstacle avoidance behaviour produces a fuzzy set representing the disallowed turning directions.

The above technique was first suggested by John Yen and Nathan Pfluger [6] as an extension to Payton and Rosenblatt's command fusion method. The method has only been tested in simulation, but has produced very favourable results.

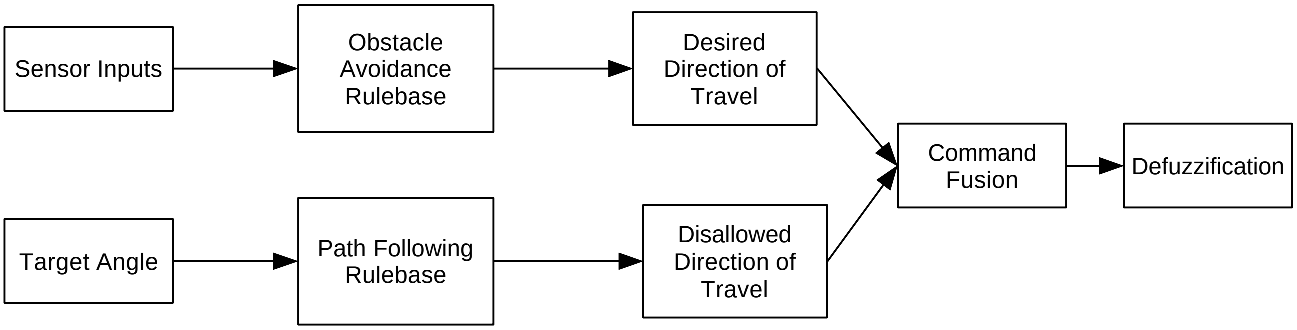

Figure 1: A fuzzy logic based navigation controller

The input to the path following behaviour is a value representing the angle between the robot's current heading and the location of the target. In order to give the robot a certain amount of flexibility in reaching its target, this specific angle is broadened into a more general desired direction by using a set of fuzzy rules. If this were not done, the robot would not be able to turn in order to avoid any obstacles. A few examples of the fuzzy rules employed by the path following behaviour are shown below:

If Target Angle is Around 0° **Then** Desired Direction is Forward
If Target Angle is Around 45° **Then** Desired Direction is Right Forward
If Target Angle is Around 90° **Then** Desired Direction is Right

The obstacle avoidance behaviour takes its inputs from a ring of ultrasonic sensors located around the robot. These provide the distance to any nearby obstacles located at the front of each sensor. Each sensor has it's own fuzzy set which represents how close it is to an obstacle. The fuzzy set for the forward-looking sensor considers objects further away to be more hazardous than the side looking sensors. This is due to objects having less influence on the side than ones at the front. The fuzzy rules for this particular behaviour produce an output which represents the direction in which the robot should not travel. Some rules employed by this behaviour are shown below:

If Forward Sensor Distance is Near **Then** Disallowed Direction is Forward
If Right Forward Sensor Distance is Near **Then** Disallowed Direction is Right Forward
If Right Sensor Distance is Near **Then** Disallowed Direction is Right

The outputs from each of the two behaviours are not crisp values but are instead fuzzy sets. One represents the desired direction in which the robot should travel while the other represents the disallowed direction of travel due to nearby obstacles. The next task is to take these two fuzzy sets and combine them using a command fusion process. This produces a single fuzzy set which is then defuzzified.

The direction for the robot to turn is both desired from the viewpoint of the path following behaviour and not disallowed from the viewpoint of the obstacle avoidance behaviour . To achieve this in fuzzy logic, the MIN operator is used:

$$\begin{aligned}\mu_{\text{Turning Direction}} &= \mu_{\text{Desired AND NOT Disallowed}} \\ &= \min(\mu_{\text{Desired}}, \mu_{\text{NOT Disallowed}}) \\ &= \min(\mu_{\text{Desired}}, 1 - (\mu_{\text{Disallowed}})\end{aligned}$$

This results in a single fuzzy set representing the turning direction for the robot. To convert this into a crisp output value, the fuzzy set must be defuzzified. Ideally, the Centre of Area defuzzification technique would be used. However, this is not an ideal method for a mobile robot since it can lead to bad output values being produced. This can be explained by a simple example. If the robot wants to move straight ahead and there is one obstacle directly in front, the robot has two options. It can either move left or right around the obstacle. The output fuzzy set in this case would consist of two peaks – one on the left and the other on the right. If this fuzzy set were defuzzified using the COA method, an output value would be produced which would take the robot even closer towards the target.

To overcome this problem, a modified defuzzification technique is employed. If a number of peaks exist in the output fuzzy set, then the set is divided into a number of separate subsets. The subset which has the largest area is then selected and defuzzified separately using the COA method. The crisp output value produced is the turning direction for the robot.

4. Conclusion and Further Work

At the present time, we have done much of the work in developing the fuzzy logic based navigation system as described by Yen & Pfluger [6]. We have successfully verified the system in simulation. We are currently in the process of building a mobile robot for the purpose, amongst others, of testing this fuzzy logic based navigation system on a real robot. The simulation has shown that the system works reliably, and continuously produces a smooth path around obstacles while travelling towards a target location. The system offers considerable advantages over the standard subsumption architecture. The ability to be able to combine the output of two behaviours results in much smoother control of the robot. If a standard subsumption based approach was used, a zig-zag motion could result.

References

[1] Brooks, R. A., "A Robust Layered Control System for a Mobile Robot." IEEE Journal of Robotics and Automation, Vol. RA-2, No. 1, 1986, pp. 14-23.

[2] Toal, D., Flanagan, C., Jones, C., Strunz, B., "Subsumption Architecture for the Control of Robots." IMC-13, Limerick, 1996.

[3] Flanagan, C., Toal, D., Strunz, B., "Subsumption Control of a Mobile Robot." Polymodel 16, Sunderland, 1995.

[4] Fitzpatrick, P., Flanagan C., "A New Approach to Control Architectures for the Control of Robots." Expersys97, Sunderland, UK.

[5] Payton, D., Rosenblatt, J., Keirsey, D., "Plan Guided Reaction." IEEE Transactions on Systems, Man and Cybernetics, Vol. 20, No. 6, pp. 1370-1382, Nov. 1990.

[6] Yen, J., Pfluger, N., "A Fuzzy Logic Based Extension to Payton and Rosenblatt's Command Fusion Method for Mobile Robot Navigation." IEEE Transactions on Systems, Man and Cybernetics, Vol. 25, No. 6, pp. 971-978, June 1995.