# Reactive Navigation for Autonomous Guided Vehicle Using the Neuro-fuzzy Techniques

Jin Cao, Xiaoqun Liao and Ernest Hall
Center for Robotics Research, ML 72
University of Cincinnati
Cincinnati, OH 45221

## ABSTRACT

A Neuro-fuzzy control method for navigation of an Autonomous Guided Vehicle (AGV) robot is described. Robot navigation is defined as the guiding of a mobile robot to a desired destination or along a desired path in an environment characterized by a terrain and a set of distinct objects, such as obstacles and landmarks. The autonomous navigate ability and road following precision are mainly influenced by its control strategy and real-time control performance. Neural network and fuzzy logic control techniques can improve real-time control performance for mobile robot due to its high robustness and error-tolerance ability. For a mobile robot to navigate automatically and rapidly, an important factor is to identify and classify mobile robots' currently perceptual environment. In this paper, a new approach of the current perceptual environment feature identification and classification, which are based on the analysis of the classifying neural network and the Neuro-fuzzy algorithm, is presented. The significance of this work lies in the development of a new method for mobile robot navigation.

**Keywords:** mobile robot, navigation, neuro-fuzzy, neural network, fuzzy logic

## 1. INTRODUCTION

The general theory for mobile robotics navigation is based on a simple premise. For mobile robot to operate it must sense the known world, be able to plan its operations and then act based on this model. This theory of operation has become known as SMPA (Sense, Map, Plan, and Act).

SMPA was accepted as the normal theory until around 1984 when a number of people started to think about the more general problem of organizing intelligence. There was a requirement that intelligence be reactive to dynamic aspects of the unknown environments, that a mobile robot operate on time scales similar to those of animals and humans, and that intelligence be able to generate robust behavior in the face of uncertain sensors, unpredictable environments, an a changing world. This led to the development of the theory of reactive navigation by using Artificial Intelligence (AI).

Reactive Navigation differs from Planed Navigation in that, while a mission is assigned or a goal location is known, the robot does not plan its path but rather navigate itself by reacting to its immediate environment in real time.

There are various approaches to Reactive Navigation, but the main concerned issue for all developer is that robust autonomous performance can be achieved by using minimal computational capabilities, as opposed to the enormous computational requirements of path planning techniques.

Designers of Reactive Navigation systems oppose the traditional robotics and Artificial Intelligence (AI) philosophy: that a robot must have a "brain", where it retains a representation of the world. Furthermore, they discard the idea that there are three basic steps for the robot to achieve its "intelligent" task: perception, world modeling and action. Robots based on this paradigm spend an excessive time in creating a world model before acting on it. Reactive methods seek to eliminate the intermediate step of world modeling.

Based on the above thinking, reactive methods share a number of important features. First, sensors are tightly coupled to Actuators through fairly simple computational mechanisms. Second, complexity is managed by decomposing the problem according to tasks rather than functions. Then, reactive systems tend to evolve as layered systems. This is where most disagreement occurs between the different researchers.

The autonomous mobile robot, the Bearcat II as shown in Figure 1., which has been used in this research, was designed by 99' UC Robot Team for the 1999 Automated Unmanned Vehicle Society competition (AUVS) sponsored by the Autonomous Unmanned Vehicle Society, the U.S. Army TACOM, United Defense, the Society of Automotive Engineers, Fanuc Robotics and others. The vehicle is constructed of an aluminum frame designed to hold the controller, obstacle avoidance, vision sensing, vehicle power system, and drive components. Two independently driven DC motors are used for vehicle propulsion as well as for vehicle steering. Also, all the subsystem level components have been chosen to be modular in design and independent in terms of configuration so as to increase adaptability and flexibility. This enables replacing of existing components with more sophisticated or suitable ones, as they become available.



Figure 1. The Bearcat II mobile robot

The navigation system takes range data as input, processes it in order to find regions that can safely drive over, and generates commands for steering the vehicle based on the

distributions of these untraversable regions. The system is set up as a reactive system in that it outputs steering commands frequently instead of planning long trajectories ahead.

Alessandro [1], M. Delgado et al, [2] gave an excellent discussion of how fuzzy computation techniques have been used in the mobile robot to address some of the difficult issues of autonomous navigation. D. Kontoravdis et al [3] present a fuzzy neural approach for the mobile robot navigation in their paper. 1999 UC Robot Team [4] describes the overall design of the UC mobile robot.

The purpose of this paper is to describe a Neuro-fuzzy control method for the navigation of an AGV robot. An overall system design and development is presented in the Section 2. The Neuro-fuzzy computation and it's application for mobile robot navigation are discussed in the Section 3. The conclusion and further work are given in the Section 4.

## 2. SYSTEM DESIGN

The system that is to be controlled is an electrically propelled mobile vehicle named Bearcat II, which is a sophisticated, computer controlled, and intelligent system. The adaptive capabilities of a mobile robot depend on the fundamental analytical and architectural designs of the sensor systems used. The mobile robot provides an excellent test bed for investigations into generic vision guided robot control since it is similar to an automobile and is a multi-input, multi-output system. The major components of the robot are: vision guidance system, steering control system, obstacle avoidance system, speed control, safety and braking system, power unit and the supervisor control PC.

By autonomous robot navigation we mean the ability of a robot to move purposefully and without human intervention in environments that have not been specifically engineered for it. Autonomous navigation requires a number of heterogeneous capabilities, including the ability to execute elementary goal-achieving actions, like reaching a given location; to reach in real time to unexpected events, like the sudden appearance of an obstacle; to determine the robot's position; and to adapt to changes in the environment.

Figure 2 is a brief description on the design and development of the navigation system of the mobile robot.

### Visual Sensor System

Visual information is converted to electrical signals by the use of visual sensors. The most commonly used visual sensors are cameras. Despite the fact that these cameras posses undesirable characteristics, such as noise and distortion that frequently necessitate readjustments, they are used because of their ease of availability and reasonable cost. Vision sensors are critically dependent on ambient lighting conditions and their scene analysis and registration procedures can be complex and time-consuming.
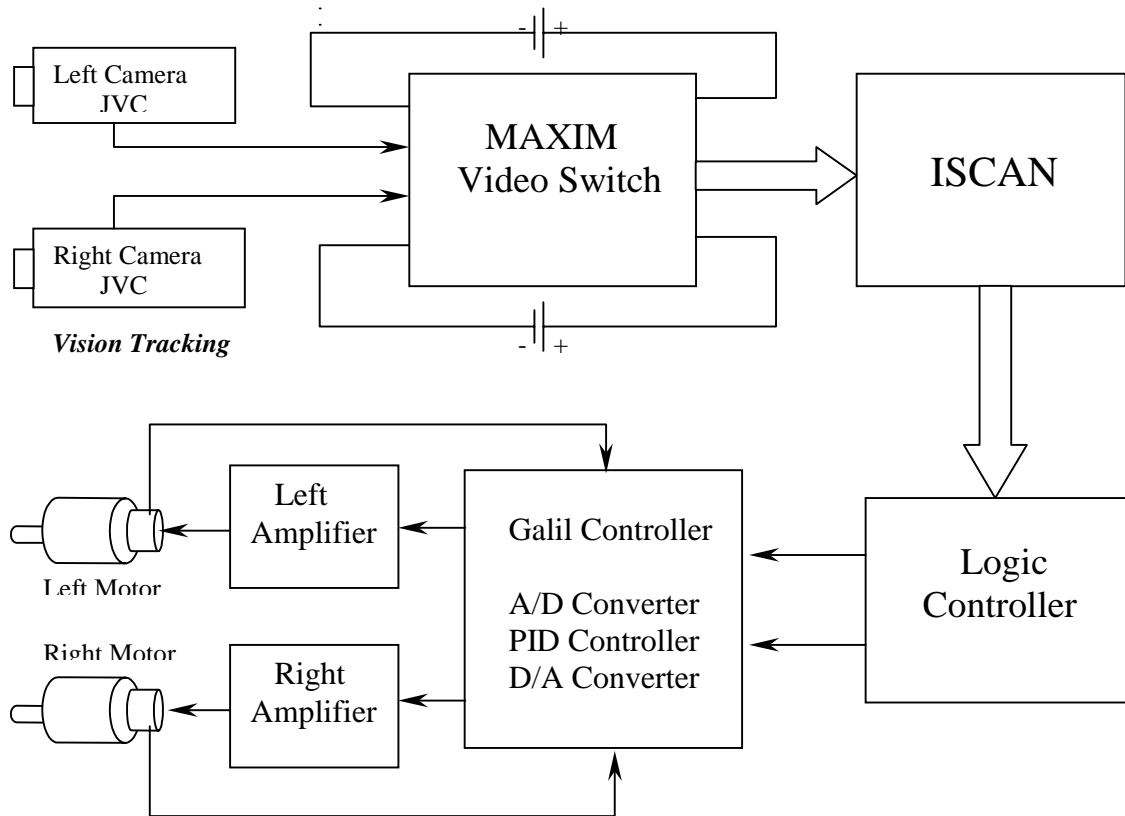
Figure 2.  The mobile robot navigation system

## The Control Of Mobile Robot

The control of mobile robots by means of sensory information is a fundamental problem in robotics. As shown in Figure 3, the robot is controlled by selecting desired angular velocity for each of two driving wheels. One castor wheel in the rear part of the robot can turn freely and thus does not determine the robot's movements.

The purpose of the vision guidance system is to obtain information from the changing environment of the obstacle course, which is usually bounded by solid as well as dashed lines. Two JVC CCD cameras are used for following the left and right lines. Only one line is followed at a time.
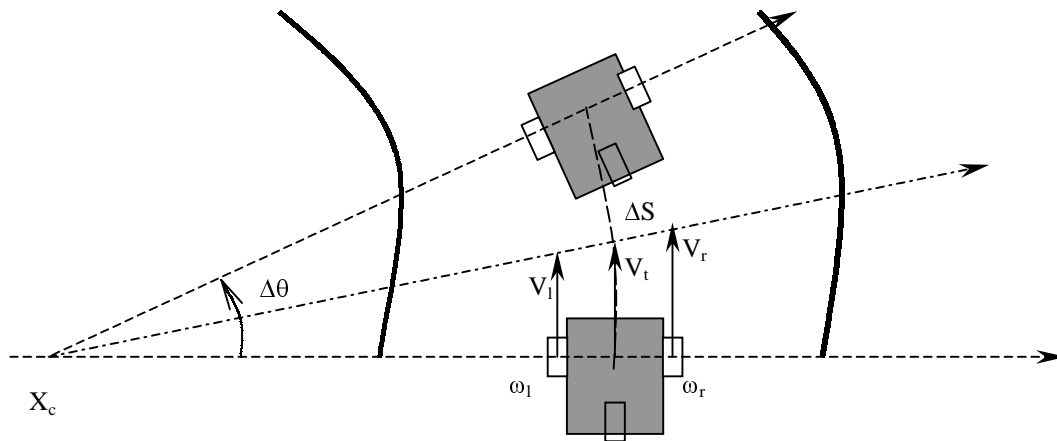
Figure 3.   The mobile robot motion

The ISCAN image-tracking device is used for image processing. This device can find the centroid of the brightest or darkest region in a computer controlled window and returns the X and Y coordinates of its centroid.

# 3. NEURO-FUZZY TECHNOLOGY

Neural networks and fuzzy systems (or neuro-fuzzy systems), in various forms, have been of much interest recently, particularly for the control of nonlinear processes.

A neural net can incorporate fuzziness in various ways:
- The inputs can be fuzzy. Any garden-variety backprop net is fuzzy in this sense, and it seems rather silly to call a net "fuzzy" solely on this basis, although Fuzzy ART (Carpenter and Grossberg 1996) has no other fuzzy characteristics.
- The outputs can be fuzzy. Again, any garden-variety backprop net is fuzzy in this sense. But competitive learning nets ordinarily produce crisp outputs, so for competitive learning methods, having fuzzy output is a meaningful distinction. For example, fuzzy c-means clustering (Bezdek 1981) is meaningfully different from (crisp) k-means. Fuzzy ART does not have fuzzy outputs.
- The net can be interpretable as an adaptive fuzzy system. For example, Gaussian RBF nets and B-spline regression models (Dierckx 1995, van Rijckevorsal 1988) are fuzzy systems with adaptive weights (Brown and Harris 1994) and can legitimately be called Neuro-fuzzy systems.
- The net can be a conventional NN architecture that operates on fuzzy numbers instead of real numbers (Lippe, Feuring and Mischke 1995).
- Fuzzy constraints can provide external knowledge (Lampinen and Selonen 1996).

### 3.1. Fuzzy Set Definition for the Navigation System

By defining the steer angel of the mobile robot is from –30 degree to 30 degree. The fuzzy sets could be defined as follows:

(a) The robot current angle with respected to the line direction: (Input Variable: AG)

| Fuzzy set | AG_2 | AG_1 | AG | AG1 | AG2 |
|---|---|---|---|---|---|
| Description | Lefter | Left | Middle | Right | Righter |
| Variation Range | -30~-10 | -20~0 | -10~10 | 0~20 | 10~30 |

(b) The robot offset with respect to the centerline: (Input Variable: OS) Suppose the road has a width of 3 meters, and we study the right-camera tracking system. So, the vehicle should keep its center from the right line for a distance of 1.5 meter.

| Fuzzy set | OS_2 | OS_1 | OS | OS1 | OS2 |
|---|---|---|---|---|---|
| Description | Lefter | Left | Middle | Right | Righter |
| Variation Range | 2.2~3 | 1.5~3 | 1.4~1.6 | 0~1.5 | 0~0.8 |

### 3.2. Input Membership
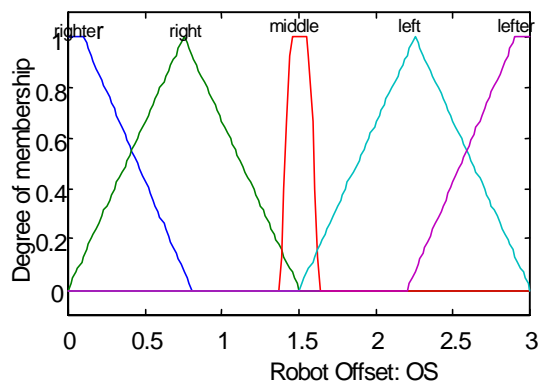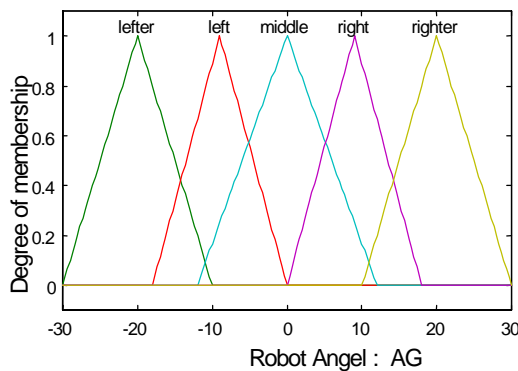


### 3.3. A Neuro-fuzzy Hybrid System

Neuro-fuzzy systems combine the advantages of fuzzy systems, which deal with explicit knowledge which can be explained and understood, and neural networks which deal with implicit knowledge which can be acquired by learning. Neural network learning provides a good way to adjust the expert's knowledge and automatically generate additional fuzzy rules and membership functions, to meet certain specifications and reduce design tie and costs. On the other hand, fuzzy logic enhances the generalization capability of a neural

network system by providing more reliable output when extrapolation is needed beyond the limits of the training data.

## 3.4. The Neuro-fuzzy Architecture

The Neuro-fuzzy system consists of the various components of a traditional fuzzy system, except that each stage is performed by a layer of hidden neurons, and neural network learning capability is provided to enhance the system knowledge.
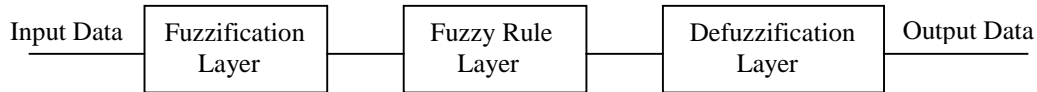
| Input Data | Fuzzification Layer | | Fuzzy Rule Layer | | Defuzzification Layer | Output Data |
|---|---|---|---|---|---|---|

Figure 4. The schematic of a Neuro-fuzzy System architecture

## 3.5. The implementation of Fuzzy Rules

Figure 5. shows a simple case of Neuro-fuzzy system for mobile robot navigation. A set of navigation rules is employed in the system.

For the Right Camera Line Tracking:

Rule 1: IF Offset is RIGHT and Angle is POSITIVE THEN the Steer_Angle is ZERO
Rule 2: IF Offset is RIGHT and Angle is ZERO THEN the Steer_Angle is LEFT
Rule 3: IF Offset is RIGHT and Angle is NEGATIVE THEN the Steer_Angle is LEFT
Rule 4: IF Offset is CENTER and Angle is POSITIVE THEN the Steer_Angle is RIGHT
Rule 5: IF Offset is CENTER and Angle is ZERO THEN the Steer_Angle is ZERO
Rule 6: IF Offset is CENTER and Angle is NEGATIVE THEN the Steer_Angle is LEFT
Rule 7: IF Offset is LEFT and Angle is POSITIVE THEN the Steer_Angle is RIGHT
Rule 8: IF Offset is LEFT and Angle is ZERO THEN the Steer_Angle is RIGHT
Rule 9: IF Offset is LEFT and Angle is NEGATIVE THEN the Steer_Angle is ZERO

The value at the end of each rule represents the initial weight of the rule, and will be adjusted to its appropriate level at the end of training. All the rules lead to three different subjects, which is the steer direction for the mobile robot. Then three output nodes are needed. They are TURN RIGHT, GO STRAIGHT and TURN LEFT correspondingly.
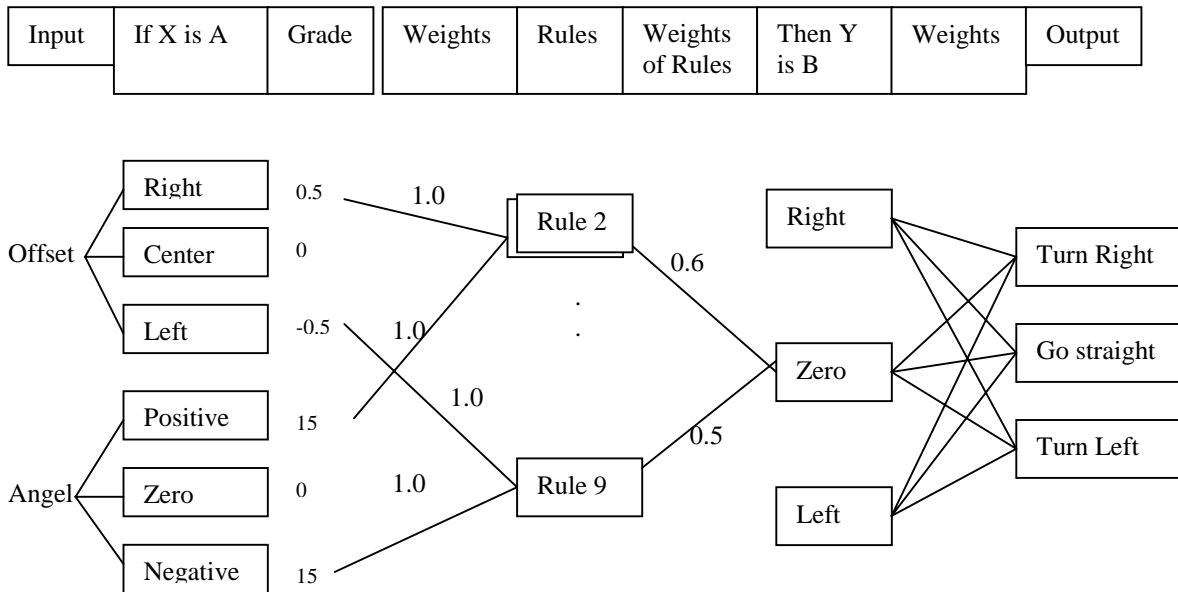
| Input | If X is A | Grade | | Weights | Rules | Weights of Rules | Then Y is B | Weights | Output |
|-------|-----------|-------|-|---------|-------|------------------|-------------|---------|--------|



Figure 5. Neuro-fuzzy Architecture

## 3.6. Training for Neuro-fuzzy System

The weight for each neural node is configured with an initial value specified by system experts, and then further tuned by using a training algorithm. A backpropagation algorithm is employed in this research as follows:

Step 1: Present an input data sample, compute the corresponding output
Step 2: Compute the error between the output(s) and the actual target(s)
Step 3: The connection weights and membership functions are adjusted
Step 4: At a fixed number of epochs, delete useless rule and membership function nodes, and add in new ones
Step 5: IF Error > Tolerance THEN goto Step 1 ELSE stop.

When the error level drops to below the user-specified tolerance, the final interconnection weights reflect the changes in the initial fuzzy rules and membership functions. If the resulting weight of a rule is close to zero, the rule can be safely removed from the rule base, since it is insignificant compared to others. Also, the shape and position of the membership functions in the Fuzzification and Defuzzification Layers can be fine tuned by adjusting the parameters of the neurons in these layers, during the training process.

# 4. CONCLUSION AND FURTHER WORK

The neuro-fuzzy system offers the precision and learning capability of neural networks, and yet are easy to understand like fuzzy system. Explicit knowledge acquired from experts can be easily incorporated into the navigation system, and implicit knowledge can be learned from training samples to enhance the accuracy of the output. A basic case is studied in this paper. Furthermore, the modified and new rules can be extracted from a properly trained neuro-fuzzy system, to explain how the results are derived. Some new technologies can be developed on the Bearcat II test bed to improve the learning speed, adjust learning and momentum rates, etc.

## REFERENCES

[1] Alessandro Saffiotti, "Autonomous Robot Navigation", Handbook of Fuzzy Computation, E. Ruspini, P. Bonissone and W. Pedrycz, Eds., Oxford University Press, 1998.

[2] M. Delgado, A. Gomez Skarmeta, H. Martinez Barbera, P. Garcia Lopez, "Fuzzy Range Sensor Filtering for Reactive Autonomous Robots", 5th International Conference on Soft Computing (IIZUKA'98), Iizuka, JAPAN, Oct 1998.

[3] D. Kontoravdis, A Likas, K. Blekas and A. Stafylopatis, "A Fuzzy Neural Network Approach to Autonomous Vehicle Navigation", Proc. EURISCON `94, Malaga, Spain, Aug. 1994.

[4] Karthikeyan Kumaraguru et al, "Design of a Mobile Robot Kit: Bearcat II", The 1999 UC Robot Team, Center for Robotics Research, University of Cincinnati, Cincinnati, OH, 1999.

[5] NIBS Pte Ltd Technical Report TR-960308, NIBS Inc., "NeuroFuzzy Computing", NewWave Intelligent Business Systems, http://web.singnet.com.sg/~midaz/.

[6] Ming Cao and Ernest Hall, "Fuzzy Logic Control for an Automated Guided Vehicle", SPIE international conference, November, Boston, 1998.

[7] Jin Cao and Ernest Hall, "Sensor Fusion for the Navigation of Autonomous Guided Vehicle Using Neural Networks", SPIE international conference, November, Boston, 1998.

[8] Jin Cao, Wen-chuan Chiang, T. Nathan Mundhenka and Ernest L. Hall, "Path Planning for Mobile Robot Navigation Using Sonar Map and Neural Network", SPIE international conference, November, Boston, 1998.

[9] Scott Pawlikowski, "Development of a Fuzzy Logic Speed and Steering Control System", Thesis of University of Cincinnati, Cincinnati, OH, 1999.

[10] General Electric, EV-1 SCR Control Manual, Charlottesville, Virginia 1986.

[11] Galil Inc, DMC-1000 Technical Reference Guide Ver 1.1, Sunnyvale, California 1993.

[12] Reliance Electric, Electro-Craft BDC-12 Instruction Manual, Eden Prairie, Minnesota 1993.