FUZZY CLUSTERING TECHNIQUES FOR GROUND VEHICLE TRAFFIC MONITORING RADAR

Nicholas Zorka & M. K. Rao
Ford Research Laboratory
Ford Motor Company
Dearborn, MI 48121, USA

Ka C. Cheok and G. Edzko Smid
Electrical & Systems Engineering Dept
Oakland University
Rochester, MI 48309-4401, USA

Abstract

A potential impetus in automobile safety is the area of collision warning, avoidance and impact reduction. Devices such as forward looking radar have added the advantage of anticipating a collision and, if necessary, deploy safety counter-measures that can reduce the severity of impact. This paper discusses a neural and fuzzy logic approach to pre-crash sensing and estimation of time to impact, using radar clusters.

Introduction

Safety of vehicle occupants and others involved in traffic accidents is a major societal concern, which is receiving increasing attention from automobile manufacturers, and others involved in the transportation industry. Currently, factors for safety consideration include vehicle structural stiffness, high impact energy absorption, safety belts, front and side airbags. Recently, drivers-aid devices such as adaptive cruise control, lane departure warning and drowsy driver detection are beginning to appear in demonstration test vehicles and as options for more expensive vehicles.

Problem Description

Millimeter wave radar and laser radar are now available for measuring distances from 0.5 meters up to 100 meters. These radars are one of the key sensors for object detection and perception of potential collision. However, data obtained from the radar sensors are normally noisy and need extensive processing to extract useful information.

The radar reads signals that return from objects and roadside clutter. The objective is to separate and identify potentially collision-causing objects from radar clutter. Neuro-fuzzy logic based algorithm will be applied to radar data to track and monitor objects in front of the host vehicle.

Test Vehicle

A prototype vehicle pictured in figure 1 was equipped with four radar units to monitor the road directly in front of its path as part of a object detection system. These four radar units are connected to a CAN network and feed their output directly into an algorithm contained in an onboard PC housed in the vehicle.

Figure 1 Four radar units in front of the bumper

The purpose of the algorithm is to provide information of potential objects in the path of the host Vehicle. Each radar unit sensor is capable of sensing up to ten objects in its path at any given time step for recording information. A sensor cycle has duration of 20 milliseconds. The radar sensing units are aligned in such a

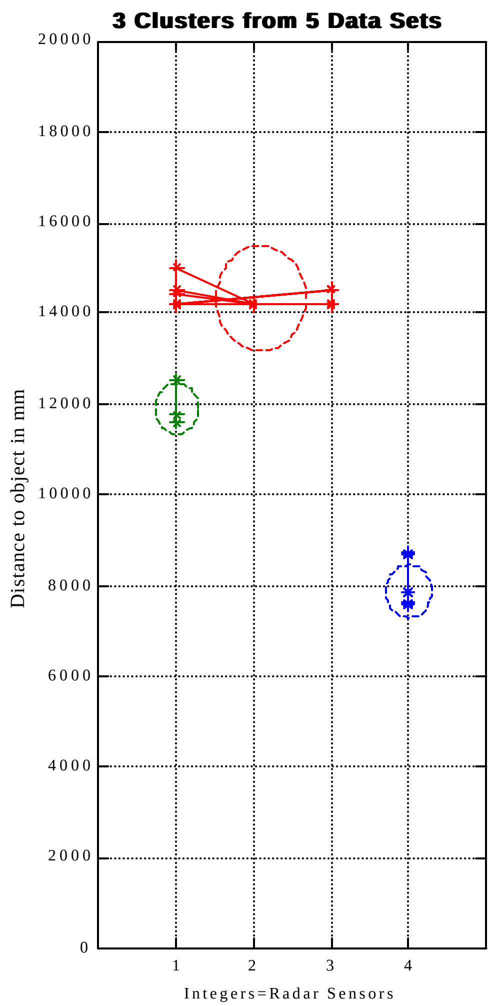

Figure 2 Matlab Output

way that they overlap each other in viewing the road way lane directly in front of the test vehicle for a distance of up to 20 meters.

The radar is part of a forward collision protection system to help monitor traffic on the road and to eliminate unsuspecting vehicle accidents and reduce its severity should they be unavoidable. This paper will discuss the impact of analyzing the large amount of data using artificial intelligence to determine if a real threat really exists as opposed to being indicated with false targets, which are also referred to as ghost targets

Fuzzy C-means Algorithm:

Pattern recognition is the search for structures in data. In pattern recognition a group of data is called a cluster. In practice the data are usually not well distributed, therefore the regularities or structures may not be precisely defined. To deal with the ambiguity, it is helpful to introduce some fuzziness into the formulation of the problem. For example, the boundary between clusters could be fuzzy rather than crisp; i.e. a data point could belong to two or more clusters with different degrees of membership. In this way, formulation is closer to the real world problem and therefore better performance may be expected.

The data used in this analysis was obtained in a test program as a means to aid in the development of algorithm to eliminate the potential of ghost targets appearing in the display. The raw data obtained from one test run was analyzed using a modified fuzzy c-mean algorithm. The test vehicle was driven on the road following a target vehicle directly ahead of it. Every 20-millisecond the four radar units transmitted and received information from objects directly in their paths. Over five hundred record samples were taken of the radar vehicle following a target vehicle. The output records were given in the following file format.

```
C
1393 0 63   0 53   0 51   0 59   0 60   0 59
0 3 10730 80 0   12020 84 0   16010 79 0
1 4 11030 79 0   12270 79 0   15800 105 0   16170 82 0
2 2 480 141 0   15830 100 0
3 1 15910 85 0
4 2 7880 106 0   8189 76 0
5 1 3510 103 0
B 1 -372 15805
C
1394 0 63   0 53   0 51   0 59   0 60   0 59
0 5 10550 79 0   11810 86 0   13270 76 0   14760 76 0
1 3 12090 76 0   15830 105 0   18080 76 0
2 3 480 141 0   15840 97 0   16190 76 0
3 1 15950 82 0
4 4 7850 103 0   8490 76 0   9490 81 0   10560 77 0
5 1 3540 91 0
B 1 -442 15834
```

The first line contains a capital C indicating the start of a new data collection cycle. The second line is the data counter of the sample being taken. The third line thru the sixth lines give the sensor number, the number of observed targets each sensor sees, the range to the observed targets as well as their individual amplitude.

Three major files were generated in the Mathworks program Matlab for analysis of the fuzzy c-mean clustering. A C-program was written to interpret the test data as a Matlab executable ('mex') `RdRadar.dll` file. The mex-file is then called by the main fuzzy c-means algorithm to pull the experimental data into the Matlab environment. The data is processed as a rolling average of five data blocks of information every 20 milliseconds. The rolling average data points were then plotted in a continuously updated graph, (see Figure 2). The x-axis shows the four radar sensor locations labeled one to four. The Y or vertical axis labeled as "distance to object in mm" displays the distance information that each radar unit recorded for each target

A procedure for the fuzzy c-mean clustering algorithm is to connect the data points in the clusters by straight lines to show the membership of each of the data points in the individual clusters. This procedure was somewhat modified by the use of circles along with the straight lines. The fuzzy c-means algorithm is actually called in Matlab as follows:

```
% Call the Fuzzy c-means algorithm to cluster the data
[centernonsort,U,obj_fcn] = fcm(data, nClusters,options);
% sort the clustering results on range of the centers
[cc,isort]=sort(centernonsort(:,2));
% extract the centers in sorted order
center = centernonsort(isort,:);
```

The circles enhanced the clusters with a larger radius increasing the confidence that cluster was indeed a real target. The remaining two clusters in general had significantly smaller confidence circles around their clusters. This could possibly be the result of only one sensor on each side of the vehicle picking up a target from the side of the road and showing it on an arc projected to the position directly in front of that radar sensor itself. Another concern is that these clusters tracked rapidly from a given distance of say 15 meters in front of the radar-equipped vehicle to virtually zero range in front of the radar vehicle. It was like seeing a stationary object at a distance and driving right past it.

Several methods on how to choose the optimal partition from the fuzzy c-mean space have been addressed in the literature [1,2,3]. The desirability of clustering candidates is established for each cluster and local minima of the objective function are defined as optimal clusters.

The objective of the fuzzy c-mean algorithm, using actual test data is to find membership functions contained in the M dimensional space, and the centers of these membership functions while minimizing the group sum of the squared errors defined by the cost function as;

$$J_m(U,V) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m \|x_k - v_i\|^2$$

Where J_m represents a spread function. $m \in [2,\infty)$ is an integer that shapes the spread and the membership functions. $U = \{u_{i,k}\}$ is defined as the set of individual membership values in the M space and V is the set of all centers of the these membership functions. The object of this cost function is to minimize the spread of the clusters subject to the constraint that the membership function values of a datum add up to one.

$$\sum_{i=1}^{c} u_{i,k} = 1$$

To optimize the width, we can take the partial derivatives of the Lagrangian of the function J_m with respect to $u_{i,k}$ and v_i and set them to zero.

This leads to

$$v_i = \frac{\sum_{k=1}^{n} u_{i,k}^2 x_k}{\sum_{k=1}^{n} u_{i,k}^2}$$

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\|x_k - v_i\|}{\|x_k - v_j\|} \right)^{\frac{2}{m-1}}}$$

Thus the optimization algorithm to find the optimal solution is a four-step process.

Step 1: Given n data points fix the number of centers c

Step 2: Set L=0,1,2 ,…, compute the C-mean vectors for I=1,…,c This computes the centers for each of the membership

$$v_i = \frac{\sum_{k=1}^{n} u_{i,k}^2 x_k}{\sum_{k=1}^{n} u_{i,k}^2}$$

Step 3: Update $U^{(L)}$ to $U^{(L+1)}$ using

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\|x_k - v_i\|}{\|x_k - v_j\|} \right)^{\frac{2}{m-1}}}$$

Step 4: Compare $U^{(L)}$ with $U^{(L+1)}$: if $\| U^{(L)} - U^{(L+1)} \| < e$ for small constant e, stop; Otherwise set L=L+1 and go to Step 2.

Enhance fuzzy c-mean Performance:

The quality of clustering can be significantly enhanced by placing a constraint on the distance between the cluster centers and the individual data points [4]. In other words the norm of the difference between the individual data points and the cluster centers must be constrained to meet a minimum distance.

This condition can easily be accomplished by specifying say six clusters or possibly more. Then invoke the fuzzy c-mean algorithm and generate the original six clusters and individual membership functions. The next step is to determine data point X_k and to which cluster it belongs. Take the norm of this data point with the different cluster centers and compare its value to a specified distance d_{ki}. If d_{ki} is less than d then make no changes to the program.

If d_{ki} is greater that d then eliminate X_k out of the Vi cluster. If d_{ki} is greater than d for all the values, eliminate X_k out of X, so that the total number of points decreases by 1. Also if all the $X_{i,k}$ for all I has the norm of the data points minus the cluster center greater that a given distance d eliminate Vi and change C=C-1. Now go back to the first step in the fuzzy c-mean algorithm and calculate a new set of membership functions based on the reduced order of the number of fuzzy clusters.

Another method to improve the output performance is to take the time derivative of the position to obtain velocity and then to take the time derivative of velocity to obtain acceleration. These two new values give added information to the system.

The major disadvantage of taking derivatives is thatit greatly enhances the noise on the signal itself. This problem can be controlled by the use of signal filtering with a Kalman filter. Kalman filtering can also be used for of predicting the next state of velocity and acceleration.

Providing velocity and acceleration information to the radar-equipped vehicle enables information on closing rates and enhances threat assessments and possible collision impact predictions.

Discussion:

Running the final program in Matlab displays the three clusters of the data obtained from the four radar sensors. The output is a series of animated lines and circles in three different colors one for each of the three clusters generated from the total output four radar sensors.

The graph of the computer generated output shows the four radar units on the horizontal axis and the range in millimeters for each of the data points taken in the actual vehicle test run. Since this was an actual vehicle test run the total number of target is known to be one target vehicle traveling at the same speed as the radar equipped vehicle. This makes analysis of the computer program much easier to resolve. The video like output is a series of colored stars that are connected by straight lines or large or small circles.

The red color cluster appears to remain at a constant distance of 17.5 meters from the radar equipped vehicle and, as the other two clusters green and blue appear to be moving down the vertical distance y-axis toward the radar equipped vehicle.

This type of action by the green and blue clusters is similar to the effect of seeing a stationary tree in the forward distance on the side of the road rushing toward the radar equipped test vehicle. With this in mind it appears that a majority of the objects picked up by the radar units are for the most part stationary objects with sufficient radar cross section.

Conclusions:

This initial work on the application of Neuro-fuzzy logic techniques to radar sensors data appears promising but more work needs to be done in the algorithm area. Possibily a new approach for better clustering of the individual radar unit data needs to be developed to eliminate the effects of strong membership of target from far away sensors

References

[1] Ka C. Cheok, G. E. Smid & D. J. McCune, "A multisensor-based collision avoidanceSystem with application to a military HMMWV", IEEE Intelligent Transportation Systems Conference Proceedings, pages 288-292, October 2000.

[2] J. C. Bezdek, S. K. Pal, Fuzzy models for pattern recognition background, significance, and key points: "Fuzzy models for pattern recognition," IEEE Press: New York, 1992; pp 1-25.

[3] LX. Wang, The fuzzy c-mean algorithm: "A course in fuzzy control," Prentice Hall: Upper Saddle River, NJ, 1997 Chapter 27.

[4] S. L. Chiu, "Fuzzy model identification based on cluster estimation," Rockwell Science Center: Thousand Oaks, California, June 1994.