NEFCOP: A Neuro-Fuzzy Vehicle Collision Prediction System

K Venkatesh, Archana Ramesh, Alagusundaram M, J V Sahana
Department of Computer Science & Engineering,
Sri Venkateswara College of Engineering, Sriperumbudur, India
kvenkatesh85@gmail.com, archana275@gmail.com, alagus@gmail.com, jvsahana@yahoo.com

Abstract

Given that road accidents occur in a real-time environment, simple crisp functions would barely provide an estimate of the gravity of the life situation. Fuzzy-based systems are able to establish complex non-linear relationships between variables with ease, making them perfect in the domain of vehicle collision prediction. However a sole fuzzy-based system, would fail to provide the user-specificity and intuition needed here. Neural networks, through their intelligent learning capabilities are ideal for this requirement. We propose NEFCOP, a neuro-fuzzy vehicle collision prediction system. NEFCOP uses laser ranging to obtain information about the road environment, which is then passed to a two-stage prediction system. The first stage clusters these data in order to prioritize them based on their relevance. The second stage is a neuro-fuzzy sub-system, which processes these data analyzing the possibility of a collision, following which the driver is warned accordingly. Thus, NEFCOP achieves adaptive, realistic accident prediction.

1. Introduction

Current collision detection and accident prediction systems, work on simple functions, and monitor only specific inputs from adjacent objects [1]-[3]. Such a simple probabilistic model cannot perfectly achieve realistic prediction of an accident, because the factors affecting the outcome vary with time. This process of prediction can instead, be effectively done through recursive approximation. Fuzzy-based systems are proven and modeled to provide ideal results under such circumstances [4]. However, although such systems can reason well with imprecise information and are good at explaining their decisions, they cannot automatically formulate the rules that they use for this reasoning. This can be solved by the introduction of neural networks. Neural networks, through pattern recognition, design and tune the membership functions, which quantitatively define linguistic labels in a fuzzy set [5,6]. Such neural network learning techniques, when applied to a fuzzy logic system substantially reduce development time and cost while improving performance [6].

Intelligent vehicle navigation systems can be primarily classified into two types- autonomous navigation systems, capable of functioning with minimal human assistance, and systems which assist vehicle-drivers in navigation, by assessing different potential collision situations. Unlike autonomous navigation systems, in the latter type of systems, the navigation is primarily controlled by the vehicle-driver, with the system providing assistance throughout. Therefore, such systems have to take into account the users' decisions and reactions. Existing approaches, fail to recognize the impact of the difference in user driving skills and judgment, on the working of the vehicle navigation system. We propose a solution: NEFCOP, which, through the use of neuro-fuzzy inference, achieves prediction that is realistic, intuitive and adaptable to the skill and expertise of the vehicle-driver.

Our paper is organized as follows: Section 2 briefly outlines the existing seminal approaches to intelligent vehicle navigation, their disadvantages and what NEFCOP intends to surpass. Following this, section 3 presents a detailed description of the proposed NEFCOP architecture and the associated algorithms. Section 4 constitutes the design and implementation specifications. Finally, section 5, records our real-time testing results and discusses the performance considerations for such a system.

2. Related work

Research in the domain of intelligent vehicle navigation systems, uses several interesting technologies. Here, we concentrate on those

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 **IEEE**

incorporating neural networks and fuzzy logic in the decision-making process.

We start with ALVINN: Autonomous Land Vehicle in a Neural Net [1], which proposed a neural network approach to vehicle navigation, by obtaining images of the road environment and then applying a neural network to this data in order to identify obstacles. Clear drawbacks of this system included the inability to be used in the absence of adequate lighting and large storage requirements. MANIAC: A Next Generation Neural Based Autonomous Road Follower [2] was a successor to ALVINN, which combined several ALVINN subsystems in order to incorporate transparent navigation between different road systems. Both the above systems did not address complex driving situations such as overtaking and turning, which were covered by SAVANT: Sentient Autonomous Vehicle Navigation using Advanced Neural Network Technique [3], a more recently proposed solution. SAVANT also minimized the space complexity, by using LASER sensors. However, as these systems were designed to navigate autonomously, none of them addressed the issue of user-adaptability. Further, they did not explore the possibility of a more realistic rendering of data through the use of fuzzy.

On the other hand most fuzzy-based systems achieve real time modeling, but fail to perceive the significance of learning algorithms applied to the process of prediction and navigation. Our solution, NEFCOP couples the two technologies and goes one step further by introducing a preliminary clustering stage. By clustering the data before it is passed to the neuro-fuzzy sub-system, the adverse effect of noisy data on the learning process is minimized and the number of inputs to the neuro-fuzzy subsystem are controlled. This ensures that the learning process is only influenced by relevant data. Equipped with a sixth sense, imparted by the neuro-fuzzy hybrid system, NEFCOP provides both improved accuracy and self-adaptability, leading to precise predictions, in tune with the driver's skill and judgment.

3. NEFCOP Architecture

3.1. System architecture

As shown in Figure 1, NEFCOP involves the following components.

- Scan phase: LASER sensors are used to scan the road environment for obstacles in the vehicle's vicinity. These sensors work at pre-determined angles such that the entire road environment is taken into consideration. The outputs provided by these sensors undergo preliminary processing, before being stored in the memory and passed on to the next phase. The utilization of laser ranging to obtain information from the environment enables NEFCOP to adapt to both night and poor visibility conditions.

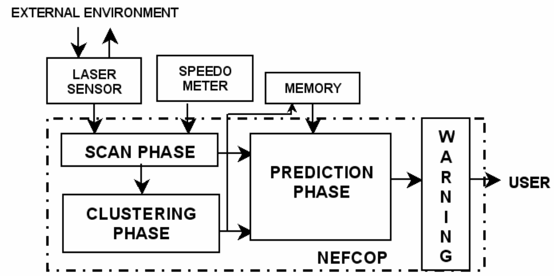

Figure 1. NEFCOP System architecture

- Clustering phase: During initial training, the clustering phase clusters the available database, based on the relevance of the data in two domains: one, the likelihood of collision and two, the user's driving skills. At runtime, the clustering phase compares the current data instance with the existing clusters, determines the associated data relevance parameter and passes this parameter to the next phase.
- Prediction phase: This phase uses a neuro-fuzzy inference system to provide an estimate of the possibility of a collision.
- Warning phase: The value obtained from the previous phase is translated into appropriate visual or audio warnings, which are used to alert the vehicle-driver, in time for corrective action.

3.2. Algorithm description

3.2.1. Initialization. This routine is performed when the system is started for the first time. This consists of the following steps.

- Dimensions

Obtain the length and width of the vehicle. These are used to compute functions (explained in Section 4), which predict the likelihood of a collision.

- Pre-determined Angles for Obstacle Direction

$$x = \{-n, -n+1, \ldots, -1, 0, 1, \ldots n-1, n\} \quad (1)$$

where n denotes the number of steps and α denotes the step angle

3.2.2. Scan phase. The outputs of this stage are the following :

Obstacle Distance(s)
Obstacle Direction(d)
NEFCOP Velocity(v)
Obstacle Relative Velocity (rv)

Operation: Given the time of flight of the light wave(t), from the laser ranger, the velocity of the

NEFCOP vehicle(v), from the speedometer and velocity of light($3x10^8$ m/s), the outputs are computed as follows.

Obstacle Distance is computed as in (2)

$$s = (3x10^8 - v) * t / 2 \quad (2)$$

- Obstacle Direction(d) is obtained by the expression (3) through predetermined angles.

$$d = x\alpha \quad (3)$$

where x and α are defined in (1).

- Obstacle relative velocity(rv) is obtained as in (4)

$$rv = 2*s / t \quad (4)$$

3.2.3. Clustering phase. The outputs of this stage are the following :

Cluster Feature(CF) Tree
Data Relevance Parameter (DRP)

Operation:

Data Relevance Computation

The outputs of the scan phase are used to obtain the data relevance parameter. The clustering performed in this phase is based on the *Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH)* algorithm[9].

The principal data structures used by BIRCH are the following:

Clustering Feature: a triplet CF=(N,LS,SS) where N: number of points in the sub cluster; LS: the linear sum on N points; SS: the square sum of data points.

Cluster Feature Tree: a height balanced tree formed in such a way that each non-leaf nodes stores the sums of the CFs of its children.

The LASER sensors provide the relative velocity, distance and direction of potential obstacles. This is stored in the memory, upon which the clustering is performed. When the system is trained for the first time, the database is scanned to build an initial in-memory CF tree and following this, the k-medoids clustering algorithm is applied in order to perform clustering of data points. During operation, for every scan by the LASER, data are passed to the clustering algorithm. Correspondingly a node is inserted into the CF-tree and the nodes are iteratively relocated. Based on the position of the current data instance in the tree, the data instance is assigned a DRP, which is then passed to the prediction phase. This parameter is used to identify the data relevant to the learning process (for e.g. data obtained from a vehicle ahead of the current vehicle, moving with a positive relative velocity would be redundant to the learning process). The DRP also indicates the relevance of the data instance to the occurrence of an accident and hence, the necessity for immediate and accurate prediction. This algorithm provides linear scalability and excellent performance when used with dynamic data.

3.2.4. Prediction phase. The output of this stage is the warning level.

Similar to the clustering phase, the prediction phase also involves an initial training routine, during which the neuro-fuzzy inference sub-system is invoked on the training data set and the membership functions are tuned.

The prediction phase receives the obstacle relative velocity, NEFCOP velocity, obstacle distance and obstacle direction from the scan phase, together with the data relevance parameter from the clustering phase. These data are first converted into the intermediate inputs (described in Table 1), required by the neuro-fuzzy inference sub-system. The following section explains the basis on which these inputs were selected.

Table 1. Intermediate inputs

Input	Description
v_o	Obstacle Velocity
d	Obstacle Direction; [0^o = Straight; -90^o = Left; 90^o = Right]
$v - v_o$	Velocity Difference
s sin(d)	Horizontal Component of Obstacle Distance
s / rv	Time to Collision
s cos(d) / rv	Horizontal Component of Time to Collision

In addition to these inputs, the width of NEFCOP vehicle (w), a constant, is also used during the application of the fuzzy rules. This neuro fuzzy inference sub-system consists of the following steps.

- Fuzzify Inputs: The first step is to determine the degree of membership of the inputs to the fuzzy rules. This decides the extent to which each input affects a rule.

- Applying Fuzzy Operators: Once the inputs are fuzzified, a unique value is computed, to represent all the antecedents of the given rule. This is done based on the fuzzy operator that governs the membership functions of the rule.

- Implication Method: Based on the value calculated in the previous stage, this step implies a certain degree of the consequent for each rule. Here the neuro-fuzzy module takes into account both the inputs and the fuzzy operator.

- Output Aggregation: An aggregate of outputs of all the rules based on the consequent value is computed for each rule.

- Defuzzification: The final step implements the weighted average method to convert the fuzzy output into crisp data that can be fed to an analog warning system.

Learning is accomplished through the Back propagation algorithm.

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 IEEE

3.2.5. Warning phase. The warning level obtained from the prediction phase is translated into analog audio/visual warning signals. This is then used to warn the user in time for corrective action.

4. Design and implementation

As explained above, the inputs, membership functions and rules of the neuro-fuzzy inference sub-system are formulated such that all possible potential accident situations are taken into consideration. This section explains how this is achieved.

Table 2 presents the different potential accident situations and the fuzzy rules they translate into.

Table 2. Neuro-fuzzy rules

Situation	Antecedent	Consequent
CASE 1: Vehicle-following Collision Direct		
Obstacle is directly in front	$d = 0^o$	compute s / rv
NEFCOP[a] and obstacle are moving in the same direction	$rv > 0$	
Obstacle is slower than NEFCOP	$v > v_o$	
CASE 2: Vehicle-following Collision Width		
Obstacle is in front, but at an angle.	$0^o < d < 90^o$ OR $-90^o < d < 0^o$	compute s cos(d) / rv
NEFCOP and obstacle are moving in the same direction	$rv > 0$	
Obstacle is slower than NEFCOP	$(v - v_o) > 0$	
Portion of NEFCOP width collides with obstacle	$s \sin(d) < w$	
CASE 3: Head-on Direct		
Obstacle is directly in front	$d = 0^o$	compute s / rv
NEFCOP and obstacle are moving in opposite directions	$rv < 0$	
CASE 4: Head-on Width		
Obstacle is in front, but at an angle.	$0^o < d < 90^o$ OR $-90^o < d < 0^o$	compute s cos(d) / rv
NEFCOP and obstacle are moving in opposite directions	$rv < 0$	
Portion of NEFCOP width collides with obstacle	$s \sin(d) < w$	

[a] NEFCOP indicates the vehicle in which NEFCOP is installed

These four distinct cases have been explicitly defined as rules, in order to aid the system adjust its weights and thereby learn faster.

Overtaking

Being a complex driving situation, we have split up the process of overtaking into the following sequence of situations

- Vehicle Following Collision Direct
- Vehicle Following Collision Width
- Head-On Direct
- Head-On Width
- Straightening

The rules to be applied in the first four cases have already been explored in Table 2. During straightening, the primary checks to be performed are to ensure that the NEFCOP vehicle does not collide with first, the vehicle it is overtaking, and second, the vehicle ahead of it while re-entering its original lane. This is achieved by obtaining the length available for the car to re-enter its original lane and comparing this with the length of the car.

5. Simulation and testing

We have simulated NEFCOP using the MATLAB Adaptive Network-based Fuzzy Inference System (ANFIS) Toolbox. As shown in Figure 2, the system structure consists of an input layer, three hidden layers and an output layer. The three hidden layers are utilized for the input membership functions, the rule base and the implication of the rules, respectively. We trained this network using data fully representative of the scenario in which it was ultimately to be used.

The test results obtained from this are shown in Figure 3 and Figure 4. Figure 3 shows the membership function plot of a given input variable, before training. Figure 4 shows how the system has learnt the membership function plot of the same input variable, after training.

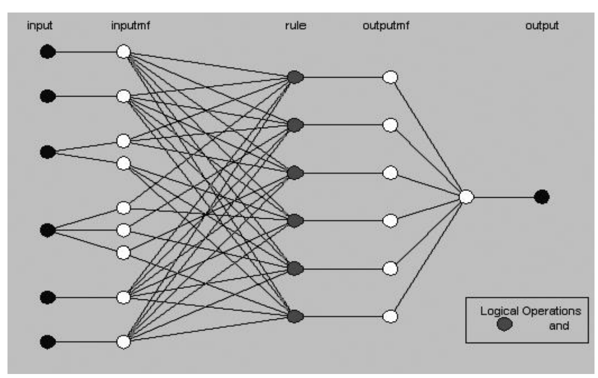

Figure 2. Structure of neuro fuzzy inference system after training

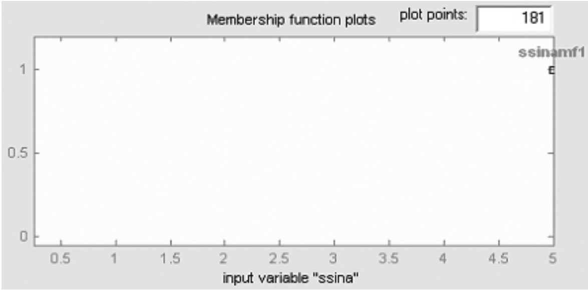

Figure 3. Membership function plot before training

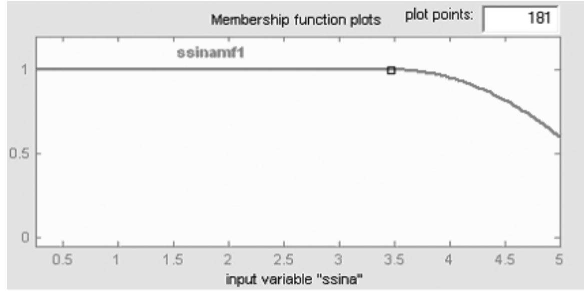

Figure 4. Membership function plot after training

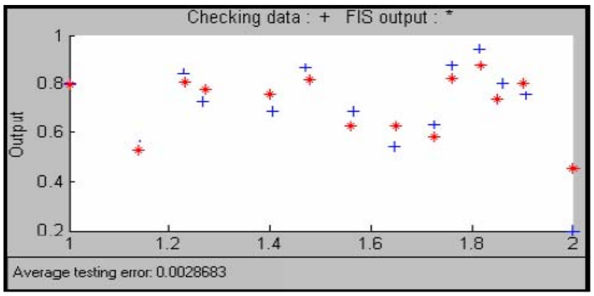

Figure 5. Testing error results

System stability was reached at the end of 230,000 epochs, with an average testing error of 0.0028683, as seen in Figure 5.

The primary requirement that NEFCOP should satisfy is reliability. Taking into account the user and vehicle's reaction time, the system must be able to warn accurately. From the error rate, we conclude that NEFCOP well satisfies this requirement. A second requirement is scalability, with respect to the area being scanned by the environment sensors. Through the use of predetermined angles and reduced step angles, NEFCOP is able to achieve this goal.

6. Conclusion

In this paper, we have proposed a solution to the problem of vehicle collision prediction: NEFCOP. Here we take into account, the dynamic nature of vehicle navigation. Existing road vehicle navigation systems are not modeled to work hand in hand with their human users. NEFCOP, on the other hand, adapts itself to the user, ensuring accurate real-time prediction. At the same time, NEFCOP is neither vehicle-specific nor vehicle-dependent.

By incorporating every possible collision situation and converging the learning data to only the relevant object instances, NEFCOP reduces computational overhead, while simultaneously handling complex driving situations.

Due to its focus on user-adaptability, NEFCOP revolutionizes the way intelligent vehicle navigation systems are designed, and paves the way for a completely new trend in intelligent vehicle navigation systems.

7. References

[1] C. E. Thorpe, M. Hebert, T. Kanade and S. Shafer, "Toward Autonomous Driving: The CMU Navlab," *IEEE Expert*, V 6 #4, August 1991.

[2] T. M. Jochem, D. A. Pomerleau and C. E. Thorpe, "MANIAC: A Next Generation Neurally Based Autonomous Road Follower," *Proceedings of the International Conference on Intelligent Autonomous Systems*, 1993.

[3] T.Srinivasan, J.B.Siddharth Jonathan and Arvind Chandrasekhar, "Sentient Autonomous Vehicle using Advanced Neural net Technology," *Proceedings of IEEE International Conference on Cybernetics and Intelligent Systems - CIS 2004,* IEEE SMC Society, IEEE Press, Singapore, Dec 2004, pp.826-831

[4] Lofti Zadeh, "Fuzzy Logic and its Application to Approximate reasoning," *Proceedings of IFIP Congress*, 1974, pp. 591-594

[5] B. Freisleben and T. Kunkelmann, "Combining Fuzzy Logic and Neural Networks to Control an Autonomous Vehicle," *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems*, IEEE Press, 1993, pp.321-326.

[6] Jyh Shing Roger Jang, Chuen Tsai Sun and Eiji Mizutani, *Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence*, ISBN-0-13-261066-3, Prentice Hall, 1997.

[7] Jyh Shing Roger Jang, "ANFIS:Adaptive-Network-Based Fuzzy Inference System," *IEEE Transactions on System, Man and Cybernetics*, 1993.

[8] Jaroubos, S Mollov, R Babuska, H B Verbruggen, "Fuzzy Model-based Predictive Control using Takagi-Sugeno Model," *Intl. Journal of Approximate Reasoning*, 1999.

[9] Tian Zhang, Raghu Ramakrishnan, Miron Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," *Proceedings of ACM SIGMOID Intl. Conference on Management of Data*, 1996

0-7695-2504-0/05 $20.00 © 2005 IEEE