Intelligent Speed Adaptation Using a Self-Organizing Neuro-Fuzzy Controller

David Partouche, Anne Spalanzani, Michel Pasquier

▶ To cite this version:

Intelligent Speed Adaptation Using a Self-Organizing Neuro-Fuzzy Controller

David Partouche   Michel Pasquier   Anne Spalanzani

Abstract— The need to increase road safety is a major concern, with millions of road users and pedestrians being killed in traffic accidents each year. The Centre for Computational Intelligence (C^2i) at NTU has developed an intelligent driving system based on hybrid fuzzy neural networks, which is able to park autonomously, drive on highways, and take some decisions such as lane changing, car following, and overtaking. This paper presents a new approach to autonomously adapt the speed of a vehicle by learning from a human driver and using anticipation. The architecture of the system is a specific fuzzy neural network realized at C2i: the Generic Self Organizing Fuzzy Neural Network using the Yager inference scheme (*GenSoFNN(Yager)*). Experiments have been conducted in simulation to test the longitudinal control and the ability of the system to anticipate curves. Results found are very promising.

I. INTRODUCTION

A world report on road traffic injury prevention has been published by the World Health Organization, estimating that 1.2 millions people are killed in road crashes each year, and as many as 50 millions are injured [1]. Projections indicate that these figures will increase about 65% over the next 20 years. The global cost of road crashed and injuries is estimated to be US$ 518 billions per year. The causes of these accidents are attributed to human error, alcohol, bad weather, heavy traffic or bad infrastructures. Some of these causes can be managed by measures, but the primary cause is human error, which is independant with the problem of road security. Autonomous driving systems have resulted from researches to decrease the risk of human error.

The Intelligent Autonomous Vehicle focuses on modeling the human driver in aspects of perception of the environment, learning and reasoning. While several researches have been made and proved effective on highways, with simple scenarios (lane marking, slow changes in curvature, no crossing, . . . [2] [3] [4]), automated driving in more diverse and complex environments is still far away from being implemented. An average person can easily drive in a city with stops, pedestrian crossing, lane changing, accelerating or slowing down according to the environment changes, but trying to automate these tasks raises a lot of problems. Even on Highways Automated Systems, longitudinal control of the vehicle uses simple behaviors (such as accelerating until the speed limit, braking or decelerating if there is an obstacle

in front of the vehicle, accelerating for overtaking...) but more complex behaviors, such as anticipating the curves, have not yet been implemented.

Intelligent Speed Adaptation are systems which can regulate the vehicle speed on roads [5] [6]. They have been tested and proved efficient in several countries, but these systems only react to the speed limit of the road, some more complex systems being able to adapt their speed to the road and weather conditions. But none of them can slow down when the car dangerously reaches a curve, which can cause the vehicle to go off the road and create an accident. Though anticipating curves is important for the security of the driver, very few research has been made on this subject, this topic being quite innovative in the Intelligent Speed Adaptation domain.

Driving can be modelled as a continuous decision-making process involving a set of rules that relate sensory input to control output. But designing these rules is quite difficult, so the simplest way is to learn from human expertise for extracting the rules. Our approach uses a type of hybrid intelligent system, a Fuzzy Neural Network, developed by C^2i: the Generic Self Organizing Fuzzy Neural Network mapped with the Yager inference scheme *GenSoFNN(Yager)*. A fuzzy neural network is a combination of a neural network and a fuzzy system, which provides advantage of both: the learning and generalization of neural networks, and the reasoning strength and ease of interpretation of fuzzy systems. A simulator, also developped by C^2i, has been used to test the capacity of the GenSoFNN(Yager) to learn how to drive from a human [7] [8]. Successful manoeuvers achieved so far include reverse parking, U-turn, as well as automated driving [9] [10] [11].

This paper is organized as follow: section 2 presents the structure and caracteristics of the GenSoFNN(Yager). Section 3 describes how the GenSoFNN(Yager) has been adapted to autonomously control a vehicle. Results of generated rules and anticipation are then analysed in section 4.

II. PRESENTATION OF THE GENSOFNN(YAGER)

The GenSoFNN(Yager) is a specific Fuzzy neural network, based on the structure of the Generic Self-Organizing Fuzzy Neural Network (GenSoFNN, described in [12]), and using the Yager Inference Scheme to interpret the fuzzy relations of the rules.

Section A describes the structure and learning of the GenSoFNN(Yager). Section B introduces the Yager Inference

D. Partouche is PhD Student at Institut National Polytechnique de Grenoble(FR). david.partouche@inrialpes.fr

M. Pasquier is Associate Professor at Nanyang Technological University, Singapore. pasquier@pmail.ntu.edu.sg

A. Spalanzani is Associate Professor at Pierre-Mendès France University, Grenoble(FR). anne.spalanzani@inrialpes.fr

Scheme, and the mapping of the Yager Inference Scheme into the GenSoFNN(Yager) is described in section C.

A. Generic Self-Organizing Fuzzy Neural Network

The Generic Self-organizing Fuzzy Neural Network (fig. 1) is a fuzzy neural network with a generic connectionist structure [12]. This network is able to automatically generate fuzzy rules, from a training data set, and has a strong noise tolerance by using the *Discrete Incremental Clustering* (DIC) technique [13].
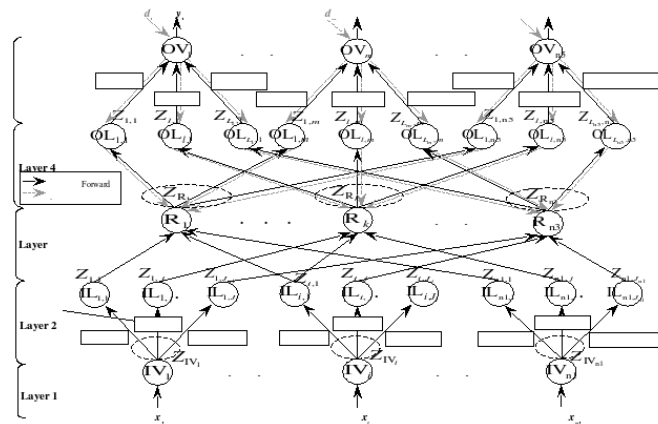

Fig. 1. Schematic structure of the GenSoFNN(Yager)

The GenSoFNN(Yager) consists of five layers of nodes. Each input node has a single input, the vector $X = [x_1, \ldots, x_i, \ldots, x_{n1}]$ represents the inputs of the Gen-SoFNN(Yager), and each output node computes a single output, the vector $Y = [y_1, \ldots, y_m, \ldots, y_{n5}]$ denotes the outputs of the GenSoFNN(Yager) with respect to the X input. In addition, the vector $D = [d_1, \ldots, d_m, \ldots, d_{n5}]$ represents the desired network outputs required during the parameter learning phase of the training cycle.

The input and output labels in layers 1 and 5 are determined by DIC [13] during the *self-organizing* phase. The nodes of other layers (2, 3 and 4) are created during the *rule formation* phase. The training phase of the GenSoFNN(Yager) consists of three phases : *self-organizing, rule formulation and parameter learning*. The *negative gradient descent back-propagation* algorithm is employed in order to tune the parameters of its fuzzy sets in label layers [14].

B. The Yager Inference Scheme

The original fuzzy inference scheme extends the conventional *modus ponens* rule which states that from the propositions:

P_1: *IF X is A THEN Y is B*
P_2: *X is A*

it can be deduced that *Y is B*.
The proposition P_1 concerns the joint fuzzy variable (X,Y) and is characterized by a fuzzy set over the cross product space $U \times V$. Specifically, P_1 is characterized by a possibly distribution, given in equation 1.

$$\Pi_{(X|Y)} = R \tag{1}$$

There are two approaches to interpret the fuzzy relation R. One is based on the *conjunctive* model of fuzzy relation and the other one is based on the *implication-based* model of fuzzy relation.
The *Compositional Rule of Inference* (CRI) scheme adopts the first approach [15]. This is illustrated in equation 2.

$$\mu_R(x,y) = min\{\mu_A(x), \mu_B(y)\} \tag{2}$$

$\mu_A(x)$ and $\mu_B(y)$ are the membership functions of set A and B.
On the other hand, the *Yager Rule of Inference* adopts the second approach, which is based on implication model of fuzzy relation [16] (as shown in equation 3).

$$\mu_R(x,y) = max\{(1 - \mu_A(x)), \mu_B(y)\} \tag{3}$$

It should be noted that both formulae correspond to the logical transition of P_1 interpreted in different ways. The second formula corresponds to the statement $\neg A \cup B$, which is essentially the same as $A \rightarrow B$ in crisp logic. The implication based model of fuzzy relation is exactly the core concept that underpins the *Yager Inference Scheme*.

C. Mapping of Yager Inference Scheme

The mapping of the Yager Inference Scheme into the Generic Self-Organizing Fuzzy Neural Network allows the allocation of the different operations used in the different layers.

- *Layer 1 (Fuzzyfication) :* This layer consists of input nodes, which fuzzify the crisp inputs the network receive.
- *Layer 2 (Antecedent Matching) :* The fuzzified inputs from layer 1 are then compared against their corresponding input labels that form the antecedent section of the fuzzy rules in the GenSoFNN(Yager). The antecedent matching between the inputs and the antecedent section is essentially to compute the negation of membership values of the inputs with respect to the input fuzzy sets.
- *Layer 3 (Rule Fulfillment) :* The third layer of the GenSoFNN(Yager) contains the fuzzy rule base of the network. Each rule node R_k computes the degree of fulfillment (i.e. the overall similarity) of the current inputs with respect to the antecedents of the fuzzy rules it denotes. In a fuzzy relation, the antecedent sections of a fuzzy rule R_k are connected by "AND" conjunctive and therefore operator min is used to compute the aggregated rule fulfillment of R_k.
- *Layer 4 (Consequent Derivation) :* Layer 4 contains output term nodes that represent the output fuzzy sets of the consequent of the rules in layer 3. Each output term may be connected to multiple fuzzy rules indicating that they may have the same consequent. GenSoFNN(Yager) uses implication-based model of fuzzy relation therefore conclusions of parallel rules will have to be combined *conjunctively*.
- *Layer 5 (Output defuzzification) :* The output nodes are responsible for the defuzzification of the derived fuzzy

Fig. 2. The throttle/brake subsystem scheme

outputs from the GenSoFNN(Yager) before presenting them as crisp outputs. For each output y_m, the derived fuzzy conclusions for all its output labels are aggregated using a modified *center of averaging* (COA) technique to produce the final output.

III. ADAPTATION OF THE GENSOFNN(YAGER) FOR CONTROL OF THE VEHICLE

The GenSoFNN(Yager) already proved its efficiency for highways driving and some tactical maneuvers, such as lane changing, overtaking, car following, and collision avoidance [17] [18].
But if lateral control has been a big focus of research during these last years [19] [20], longitudinal control is still in its first experiments [21] [22] [23]. The GenSoFNN(Yager) was used to implement longitudinal control of a vehicle, in order to autonomously adjust its speed according to the upcoming curve. The main advantage of using a fuzzy neural network instead of a classical neural network is the ability to extract the rule base from the system, and then to understand the functioning of the network.

A. Implementation of the Longitudinal Control

In order to implement the longitudinal control, two networks are used, one for the throttle, and another for the brake. The inputs used for the both networks are the same: the speed of the car and an anticipation variable (as shown in fig. 2). These inputs are based on experiments made on human drivers, in order to understand what processed information is acquired while they are driving [24] [25] [26] [27].

The speed variable vary between 0 and 200km/h is linearly normalized before being fed into the network.
The anticipation variable is used to memorize the shape of the upcoming curve. When humans are driving, they accelerate or brake according to their analysis of the road (they don't only use the last image they recorded). Consequently, a simple sensori-motor control can't be applied to anticipate the upcoming curves.
The main idea was to calculate anticipation in a variable, which is then fed to the GenSoFNN(Yager).

Every time step, the angle of the curve is calculated at a certain distance of the car (the distance depends of the speed, as shown in table I).

The returned value is then added to a list, which contains the description of the shape track for 1.6 seconds. This list is then used to calculate the anticipation of the curve (described in algorithm 1). The result value is an angular speed, normalized between $-PI$ and PI.

TABLE I
THE DISTANCE OF THE CURVE, DEPENDING OF THE SPEED

speed (in m/s)	distance (in m)
x < 4.7	5
4.7 < x < 7.8	10
7.8 < x < 10.9	15
10.9 < x < 14.1	20
14.1 < x < 17.2	25
17.2 < x < 20.3	30
20.3 < x < 23.4	35
23.4 < x < 26.6	40
26.6 < x < 29.7	45
29.7 < x < 32.8	50
32.8 < x < 35.9	55
x > 35.9	60

Algorithm 1 Calculate the anticipation variable

$anticipation \leftarrow 0$
for $i = 1$ to $length(curve)$ **do**
  if $curve[i > 0] \wedge curve[i+1 > 0]$ **then**
    $anticipation \leftarrow anticipation + \frac{curve[i+1]-curve[i]}{time\_step}$
  else if $curve[i < 0] \wedge curve[i+1 < 0]$ **then**
    $anticipation \leftarrow anticipation + \frac{curve[i]-curve[i+1]}{tim\_step}$
  else if $curve[i < 0] \wedge curve[i+1 > 0]$ **then**

    if $-curve[i] > curve[i+1]$ **then**
      $anticipation \leftarrow anticipation + \frac{curve[i]-curve[i+1]}{time_step}$
    else
      $anticipation \leftarrow anticipation + \frac{curve[i+1]-curve[i]}{time\_step}$
    end if
  else if $curve[i > 0] \wedge curve[i+1 < 0]$ **then**

    if $curve[i] > -curve[i+1]$ **then**
      $anticipation \leftarrow anticipation + \frac{curve[i+1]-curve[i]}{time\_step}$
    else
      $anticipation \leftarrow anticipation + \frac{curve[i]-curve[i+1]}{time\_step}$
    end if
  end if
end for

Given the algorithm, it is now possible to see if the curve is soft or sharp (the larger the value is, the sharpest is the curve), and also to know if this is the beginning or the end of the curve. At the beginning, the curve become sharper, then the result is positive, and at the end the curve become softer, and the result is negative.
Consequently, the vehicle does not react as soon as it detects a curve, but when a limit is reached (the variable *anticipation* become bigger and bigger while the car approaches a curve). So the variable depends not only from the sharpness of the curve, but also from the distance the car is from the curve. This allow the system to have the shape of the road for the next 1.6 seconds.
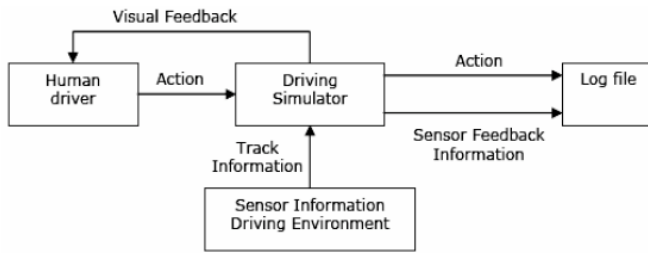
Fig. 3. Training data collection [7]

Fig. 4. schema of tracks used for the tests

B. The Driving Simulator

The first simulator used at C^2i has been developed and improved by Toh Mary et al. and has been involved in many projects [7] [28] [9]. The aim of this simulator was to provide a realistic environment to experiment on automated vehicles, using data from a human driver, and then learning with a fuzzy-neural network.

The main problem with this simulator is that the driving physics, such as the centripetal force that causes the car to slip from its turning course during high speed cornering, aren't well implemented. Or it is from the physics model that accurate experimental results can be obtained to justify the reliability of the GenSoFNN(Yager) to handle the driving of a real car. TORCS (*http://torcs.sourceforge.net*), an open-source racing car simulator, has been chosen and modified for our purpose.

The simulator is used to collect the training data that capture different driving maneuvers in a given road scenario. The drivers use the visual feedback to decide the subsequent action to be taken. For instance, if the driver see a turn ahead, he/she might respond by turning the steering wheel in a certain direction. Figure 3 summarizes the training data collection of human driving performance. The log file stores all the actions taken including the sensor information of each simulation time interval. The information is then used to train the neuro-fuzzy system within the simulation.

Once the human training set recorded, files are used to train the network. If the training set is coherent, the system should be able to automatically drive the car, reproducing the human driving. The network having a capacity of generalization, the system shall also be able to manage new situations, as humans do when they learn to drive: a monitor shows how to drive on a specific road, but then it can be generalized to drive on every road.

IV. EXPERIMENTAL RESULTS

This section describes the results obtained with the Gen-SoFNN(Yager), for the longitudinal control of the vehicle. Section A describes the experimental setup. Section B analyses the semantic of the generated rules. Section C studies how the rules are fired in the system, and section D analyses the results obtained with anticipation.
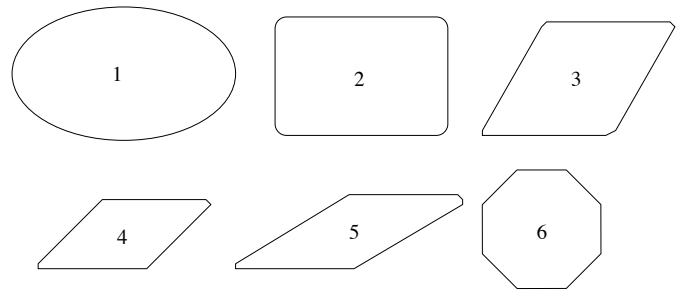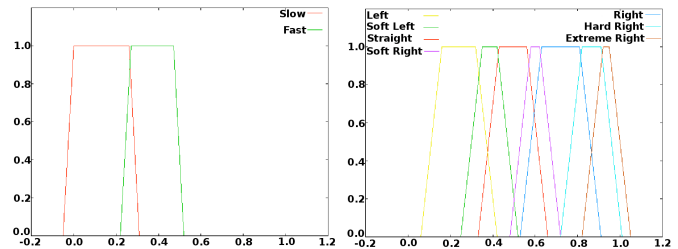

Fig. 5. Labels of the two inputs, absciss is the value of the input and ordinate is the degree of membership

A. Experimental Setup

During the test, 6 tracks were created (as shown in fig. 4). For the analysis of the generated rules (section B and C), only track 1 was used, the other tracks being used for the anticipation test. A driver was asked to drive 3 turns on this tracks, and data were recorded as described in the previous section. Then data was used for the learning of the network, which was able to reproduce the driving process. The aim of this test was to study the generated rules, which are described in the following section.

B. Rules Semantic Analysis

There is a total of 12 rules, linked to 9 input labels and 4 output labels. The inputs have two features: the speed of the vehicle and the degree of curvature of the road. The speed of the vehicle has 2 fuzzy sets, and the curvature has 7 fuzzy sets. The throttle output has 4 fuzzy sets.
The fuzzy sets of the antecedent layer described the different fuzzy values the inputs can have (as shown in fig. 5), we can attach linguistic terms to these values :

- input "Speed" (S) = {slow, fast}
- input "Angle Curvature" (C) = {straight, soft left, right, soft right, hard right,extreme right, left}

For the consequent layers, only the centroid of the kernel is needed, which is given by table II.
Values can be interpreted as follow:

- output "Throttle" (T) = {normal acceleration, slow acceleration, hard acceleration, no acceleration}

The corresponding fuzzy rule extracted from the rule base of the throttle system can be formulated with the linguistic terms as follow:
1) **IF** S is *slow* **AND** C is *straight* **THEN** T is *slow accel*
2) **IF** S is *slow* **AND** C is *soft left* **THEN** T is *slow accel*

TABLE II

CENTROID OF THE KERNEL

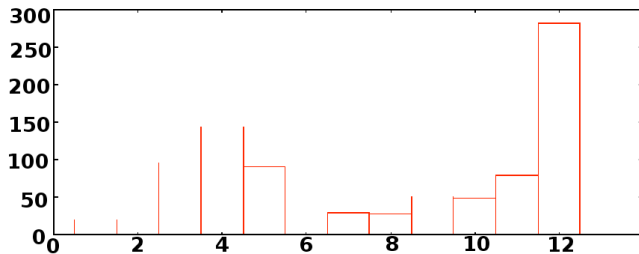| Neuron	Centroid
1	0,46
2	0,33
3	0,59
4	0,06

Fig. 6. Rule Firing Strength, absciss is rule number and ordinate is cumulative strength

3) **IF** *S* is *slow* **AND** *C* is *right* **THEN** *T* is *normal accel*
4) **IF** *S* is *fast* **AND** *C* is *hard right* **THEN** *T* is *no accel*
5) **IF** *S* is *fast* **AND** *C* is *soft right* **THEN** *T* is *normal accel*
6) **IF** *S* is *slow* **AND** *C* is *soft right* **THEN** *T* is *normal accel*
7) **IF** *S* is *fast* **AND** *C* is *soft left* **THEN** *T* is *fast accel*
8) **IF** *S* is *fast* **AND** *C* is *left* **THEN** *T* is *fast accel*
9) **IF** *S* is *fast* **AND** *C* is *extreme right* **THEN** *T* is *no accel*
10) **IF** *S* is *slow* **AND** *C* is *hard right* **THEN** *T* is *no accel*
11) **IF** *S* is *fast* **AND** *C* is *straight* **THEN** *T* is *fast accel*
12) **IF** *S* is *fast* **AND** *C* is *right* **THEN** *T* is *no accel*

The rule base is obviously coherent with the human way of driving, apart for the left turning (which is normal,as the tracks have no right turn the system could learn from). Rule 1 shows that if the car speed is slow, and the road is straight, acceleration is slow. This is due to the fact, that the car speed is slow on a straight line only at the start (then the car speed is always fast on straight lines). So when the system start driving, he first go slowly, and then increases the speed (such as driver does).

C. Rule Firing Analysis

A study of the rules fired in the network was conducted to analyze the network ability to maintain the consistency of the rule base. A concise and consistent rule base is significant to ensure good performances and a critical factor when applied to an embedded control system with limited storage capacity. A consistent rule base is shown by comparing the proportion of rules actually used compared to the total number of rules in the network. A consistent rule base is best displayed by a wide spread of the rules being fired over the total number of rules for all possible driving situations. The experimental results for the rule firing strength throughout the experiment process are summarized in fig. 6.

The rule base generated by the GenSoFNN(Yager) is consistent, as all the rules are used (the main rule being the number 12, and rules 3, 4, 5 and 11 being the auxiliary rules).

Comparing the rule semantics with the rule firing strength, the graph illustrates that the first rule is not often used (just when the car starts the race). The 3 left turning rules are also not often used, as for the slow and soft right rule (the car doesn't arrive slowly on a soft curve). The most used rule is the number 12, which means that the car often tries to go fast on right curves, when they are not too sharp (this rule is the most often used as the car more often drives on the right curves than on the straight lines).

Fuzzy rules are intuitive to human cognitive process. It is also very easy to extract the rules of the fuzzy neural network, and so to understand it, compared to the "black box" of the neural networks.

D. Anticipation Analysis

1) First test: driving on curves of same radius
During this test, a human was asked to drive on 3 turns tracks 2 and 4, and tests were reproduced on tracks 2, 3, 4 and 5. This test was done to study the capacity of generalization of the network (by interpolating and extrapolating the curves it learned), and to compare the autonomous driver with the human driver.
After training of the network, the autonomous driver was able to correctly drive on each of the test tracks. The results obtained for comparision between the human driver and robot driver on track 2 (with curves of 90 degrees) are described in the table III.

TABLE III

COMPARISION BETWEEN THE HUMAN AND ROBOT DRIVER

	human driver	robot driver
MSE	1.18	2.93
TIME (in sec)	169	164
MAX SPEED (in km/h)	152	128
Anticipation (mean in m)	38	45

Results show that the robot driver can drive faster than the human driver, without exceeding the limit (130 km/h). The driver exceeds it, particulary when he is concentrating on the road, and doesn't look at the speedometer. The human anticipation was also really differing in each curve (varying between 22 to 70 meters), but the robot anticipation is always around 45 meters before the curve.
The only driving skill the robot driver was not as good as the human driver was for the lateral driving (the robot was more driving near the left lane than in the middle of the lanes), this is shown by the Mean Squared Error (MSE) which is higher for the robot.
In conclusion, the robot drives in a safer way than the human driver. It doesn't exceed the road limit, slows down gradually quite in advance and has a smoothly acceleration/deceleration.

2) Second test: driving on the 45 degree curves

For this second test, the human drives on a track with 8 curves of the same degree (45 degrees), but with different radius (50,100,150 and 200 meters, track 6 on fig. 4).

The goal of this test was to see if the robot driver reacts differently according to the radius of the curve.

Data of the human driver was first recorded, then the network was trained with these one. Table IV shows the average of time the driver begins to slow down before arriving in the curve. Results show that the driver slows down according

TABLE IV

SLOW DOWN PERFORMANCE, ACCORDING TO THE RADIUS

Radius	next curve (in m)
200	0
150	32
100	44
50	53

to the radius. This means that the sharper is the curve (i.e. the higher radius), the sooner the robot driver begins to slow down.

The results found are the one that should be expected for a human driver. Coupled with the last test, they are very promising.

V. CONCLUSIONS AND FUTURE WORKS

Experimental results proved that the GenSoFNN(Yager) was able to correctly learn how to anticipate curves, and to adapt its speed according to the shape of this curve. Semantic analysis of the system was possible due to the capability of extracting the rule base created by the network. These tests, though being preliminary results of the project, are very encouraging for the future work.

Next step of the project is to improve the lateral control of the system by adding anticipation, such as on the longitudinal control. For the moment, only longitudinal control has been developed with this new method, and gave us very promising results. Improving the lateral control in the same way will allow the system to autonomously drive on any type of track.

REFERENCES

[1] M. Peden, R. Scurfield, D. Sleet, et al. World Report on Road Traffic Injury Prevention. *World Health Organization*, Geneva, 2004.

[2] D.A. Pomerleau. ALVINN: an autonomous land vehicle in a neural network. *In Advances in Neural information Processing Systems 1*, Morgan Kaufmann Publishers, San Francisco, CA, pp. 305-313, 1989.

[3] J. Forbes, T. Huang, K. Kanazawa, and S. Russell. Batmobile: Towards a bayesian automated taxi. *In Proceedings of the 14th Int. Joint Conf. on Artificial Intelligence*, Montreal, Canada, pp. 1878-1885, 1995.

[4] R. Sukthankar, S. Baluja, J. Hancock, D. Pomerleau, and C. Thorpe. Adaptive intelligent vehicle modules for tactical driving. *In 1996 AAAI Workshop on Intelligent Adaptive Agents*, pp. 13-22, 1996.

[5] K. Brookhuis, D. de Waard. Limiting speed, towards an intelligent speed adapter (ISA). *Transportation Research Part F: Traffic Psychology and Behaviour*, Volume 2, Number 2, pp. 81-90, June 1999.

[6] A. Van Loon, and L. Duynstee. Intelligent Speed Adaptation (ISA): A Successful Test in the Netherlands. *Proceedings of the Canadian Multidisciplinary Road Safety Conference XII*, London, Ontario, 10-13 June 2001.

[7] M. Pasquier, C. Quek, and M. Toh. Fuzzylot: a novel self-organising fuzzy-neural rule-based pilot system for automated vehicles. *Neural Networks*, Volume 14, Number 8, pp. 1099-1112, Oct 2001.

[8] M. Pasquier, C. Quek, W.L. Tung, D. Chen, and T.M. Yep. Fuzzylot II: a novel soft computing approach to the realisation of autonomous driving manoeuvres for intelligent vehicles. *"Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision*, 2002.

[9] R.J. Oentaryo, and M. Pasquier. Self-trained automated parking system. *Proceedings of the 8th IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV 2004)*, 2004.

[10] D.W. Chen, M. Pasquier, and C. Quek. Automating three-point turn maneuvers for autonomous vehicles. *Proceedings of the IASTED International Conference on Artificial and Computational Intelligence*, Tokyo, Japan, 2002.

[11] W.L. Tung, M. Pasquier, and C. Quek. A Generic Self-Organizing Fuzzy-Neural Network and Its Application to Automated Driving. *Proceedings of the IASTED International Conference on Artificial and Computational Intelligence (ACI 2002)*, Tokyo, Japan, 2002.

[12] W.L. Tung, and C. Quek. GenSoFNN: a generic self-organizing fuzzy neural network. *IEEE Transactions on Neural Networks*, Volume 13, Number 5, pp. 1075-1086, 2002.

[13] W.L. Tung, and C. Quek. DIC: A Novel Discrete Incremental Clustering Technique for the Derivation of Fuzzy Membership Functions. *Lecture Series in Artificial Intelligence 2417 - PRICAI 2002: Trends in Artificial Intelligence*, Springer Verlag, Berlin, pp. 178-187, 2002.

[14] D.E. Rumelhart, G.E. Hinton and R.J. Williams. Learning internal representations by error propagation. *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations*, MIT Press, Cambridge, MA, USA, pp. 318-362, 1986.

[15] L. A. Zadeh. Calculus of fuzzy restrictions. *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A. Zadeh*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, pp. 210-237, 1996.

[16] J.M. Keller, R.R. Yager and H. Tahani. Neural network implementation of fuzzy logic. *Fuzzy Sets Syst.*, Elsevier North-Holland, Inc., Amsterdam, The Netherlands, Volume 45, Number 1, pp. 1-12, 1992.

[17] R.J. Oentaryo, and M. Pasquier. GenSoFNN-Yager: A Novel Hippocampus-like Learning Memory System Realizing Yager Inference. *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN'06)*, Vancouver, BC, pp. 1684-1691, July 2006.

[18] M. Pasquier, and R.J. Oentaryo. Learning to Drive the Human Way: A Step Towards Intelligent Vehicle. *International Journal of Vehicle Autonomous Systems, Special Issue on Integrated Vehicle Control: Theories, Methods and Experimentation*, to appear in 2007.

[19] C. Taylor, J. seck, R. Blasi and J. Malik. A comparative study of vision-based lateral control strategies for autonomous highway driving. *International Journal of Robotics Research*, Volume 18, Number 5, pp. 442-453, 1999.

[20] R. O'Brien, P. Iglesias, and T. Urban. Vehicle Lateral Control for Automated Highway Systems. *IEEE Transactions on Control Systems Technology*, Voume 4, Number 3, May 1996.

[21] D. N. Godbole, and J. Lygeros. Longitudinal control of the lead car of platoon. *IEEE Transactions on Vehicular Technology*, Volume 43, Number 4, pp. 1125-1135, 1994.

[22] E.H.M. Lim, and J.K. Hedrick. Lateral and longitudinal vehicle control coupling for automated vehicle operation. *American Control Conference*, Volume 5, pp. 3676-3680, 1999.

[23] S. Huang, and W. Ren. Vehicle longitudinal control using throttles and brakes. *Robotics and Autonomous Systems*, Volume 26, Number 5, pp. 241-253, 1999.

[24] M.F. Land, and D.N. Lee. Where we look when we steer. *Nature*, Volume 369, pp. 742.744, 1994.

[25] J.B.J. Riemersma. Attention and Perception in Negotiation of Curves. *Vision in Vehicles III*, Aachen, Germany, 1989.

[26] E.R Boer. Tangent point oriented curve negotiation. *Intelligent Vehicles Symposium*, Tokyo, Japan, 1996.

[27] M.F. Land. The Visual Control of Steering. *Vision and Action*, Cambridge University Press, pp. 163-180, 1998.

[28] K.K. Ang, C. Quek, and M. Pasquier. POPFNN-CRI(S): pseudo outer product based fuzzy neural network using the compositional rule of inference and singleton fuzzifier. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, Volume 33, Number 6, pp. 838-849, 2003.