A Novel Fuzzy Inferencing Methodology For Simulated Car Racing

Duc Thang Ho and Jonathan M. Garibaldi

***Abstract*—This paper describes and further extends the fuzzy inferencing system which won the simulated car racing competition that was arranged as part of FuzzIEEE 2007 conference. The details of the winning non-stationary fuzzy controller and its results are presented. A novel approach to further improve the performance of the winning controller is described and formalised. We term the new fuzzy inferencing method a '*context-dependent fuzzy inference system*'. The concept of a '*context-dependent fuzzy set*' that is utilised by the fuzzy system is introduced. Finally, a comparison between context-dependent fuzzy inference system and various existing techniques are carried out on the simulated car racing application. The results show a better performance for context-dependent fuzzy inference systems in stochastic circumstances.**

***Index Terms*—Context-dependent Fuzzy Inference System (CDFIS), Context-dependent Fuzzy Sets (CDFS), Non-stationary Fuzzy Inference System (NSFIS), Non-stationary Fuzzy Sets (NSFS), Fuzzy Inference System (FIS).**

I. Introduction

TYPE-1 Fuzzy Inference Systems (FIS) were specially designed to represent uncertainty and vagueness and provide formalized tools to deal with the imprecision in many real world problems. Although FISs are able to map the typical non-linear relation of input-output model without a precise formula, they do not possess the capability to handle various uncertainties in practical applications [1]. These various uncertainties can be classified into non-stochastic uncertainties and stochastic uncertainties [**?**]. For example, random noise can be seen as stochastic uncertainty while the uncertainty about the plant dynamics can be described as non-stochastic. In an attempt to better model and minimize the effect of these uncertainties, the concept of fuzzy sets of type n, $n = 1, 2, 3, ...$, was introduced [2] and investigated by various authors [1][3][4][5][6]. It has been shown that these methods improve the ability to handle uncertainties especially non-stochastic uncertainty. Stochastic uncertainties, or random uncertainties, are frequently met in engineering fields such as robotic control systems [7] and signal processing [8]. These type of uncertainties are usually handled using probabilistic modelling methods which have been proven to be very effective in many applications. In 2005, Zhi Liu and Han-Xiong Li [9] incorporated the ability to processing stochastic uncertainty of probabilistic methods into the existing fuzzy methods by introducing the probabilistic fuzzy inference system that can be applied to engineering application directly. The initial application of probabilistic fuzzy inference systems has shown positive results. However, little work has been done to verify the usefulness of this form of fuzzy systems since then.

In the context of fuzzy expert systems, Garibaldi [10] motivated by the variations in the decisions which occur among a panel of experts, as well as in the decisions of an individual expert over time, has proposed the notion of *Non-deterministic Fuzzy Reasoning* in which variability is introduced into the membership functions through the use of random alterations. Later on, he extended and formalised the concept through the introduction of a notion termed *Non-Stationary Fuzzy Sets* (NSFS) [10]. Type-1 and type-2 fuzzy sets do not capture the notion of variability, in that a type-1 or type-2 FIS will always produce the same output(s) given the same input(s). Recent research [11][12][13] has shown that *Non-stationary Fuzzy Inference Systems* (NSFIS) can achieve good performance in both stochastic and non-stochastic circumstance.

With the aim of comparing the strengths and weaknesses of different fuzzy methodologies, a simulated car racing competition was run at FuzzIEEE 2007 [14]. Our non-stationary fuzzy controller won the first prize. In an attempt to further improve the performance of this non-stationary fuzzy controller, we have investigated different ways to make changes to the underlying type-1 fuzzy system. All the rules of system were retained and only the parameters of the membership functions of the inputs were altered. One of the best results was achieved when the membership functions of a term were adjusted in accordance to the value of another input variable. This means that for every distinct set of input variables, the FIS has a different set of membership functions and therefore the inference process will be affected. None of the existing fuzzy techniques can be used to model this kind of relationship. NSFS can only model changes over time while probabilistic fuzzy sets can only model random alterations. Type-1 and type-2 fuzzy sets are always fixed, even when circumstances change, and hence cannot model this kind of dependency. The use of linguistic modifiers or fuzzy optimization methods often result in a permanent shift of membership functions, while the changes of membership functions required in this case are temporary. The idea that the membership functions of a variable may change depending on other variables is novel and has motivated an investigation in the effect of context environment on the decision making process.

In this paper, the notion of *context-dependent fuzzy sets* (CDFS) is introduced and the design of *context-dependent fuzzy inference systems* (CDFIS) is presented in section II. In section III, the simulated car racing application in the FuzzIEEE 2007 fuzzy competition is described, and results are presented , which demonstrate that CDFIS can achieve better modelling performance, especially in stochastic circumstances. The discussion is given in section IV

978-1-4244-1819-0/08/$25.00©2008 IEEE

II. Context-dependent Fuzzy System

A. Context-Dependent Fuzzy Sets

Consider an FIS which consists of n linguistic variables. Each linguistic variable x_i, $i \in \{1, \ldots, n\}$, has a universe of discourse X_i. The context set $\mathcal{C}$ is defined to be a subset of the Cartesian product of the universe of discourse $X_1, ..., X_n$:

$$\mathcal{C} \subseteq X_1 \times \cdots \times X_n$$

For each context $c \in \mathcal{C}$, the context-dependent fuzzy set of a term $\bar{A}$ of a linguistic variable x_i is defined as:

$$\bar{A} = \int_{c \in \mathcal{C}} \int_{x_i \in X_i} \mu_{\bar{A}}(c, x_i)/x_i/c.$$

Note that the variables in the context set do not necessarily appear in the rule set of the FIS. For example, we could have a context set consisting of three variables, *distance, speed* and *heading angle*. However, only *speed* and *heading angle* are used in the rule set of the system and *distance* is used as a dummy variable which influences the fuzzy sets of the other two variables. The context set $\mathcal{C}$ can contain any number of variables in the FIS. Each specific instant c of the context set $\mathcal{C}$ consists of the states of the variables defined in $\mathcal{C}$: $c = \{x_1, \ldots, x_k\}$. We implicitly assume that the variable x_i of the context will not have any effect on the variable x_i of the system to avoid recursive definitions.

As an example, let us consider a FIS consisting of two input variables, $x \in \{0, \ldots, 10\}$ and $y \in \{0, \ldots, 10\}$. The context set $\mathcal{C}$ contains only the variable y ($\mathcal{C} = \{y\}$). The membership function of a term $\bar{A}$ of x is defined depending on the context c as follows:

$$\mu_{\bar{A}}(c, x) = \mu(y, x) = z = e^{-\frac{(x*r_1+3-y*r_2)}{8}}$$

where r_1 and r_2 are two random numbers with normal distribution, and z is the degree of membership of x in each context c.

Figure 1 shows pictorial representation of the context dependent fuzzy set $\bar{A}$. Figure 2 shows the 2D slice taken at $y = 2.2$ of the same context dependent fuzzy set. These two figures illustrate the point that, at each context, a context dependent fuzzy set reduces to an ordinary type-1 fuzzy set.

B. Context-Dependent Fuzzy Systems

CDFIS consist of the same four main components as an ordinary type-1 FIS: the fuzzifier, the rules, the inference engine and the defuzzification. Type-1 FISs use only type-1 fuzzy sets, whereas an FIS which uses at least one context-dependent fuzzy set is a CDFIS. The context set $\mathcal{C}$ should contain only those variables which are used at least once in the definition of other variables fuzzy sets, or in other words, independent variables can be removed from the context set. A CDFIS can be thought of as a dynamic type-1 system in which the membership functions of the system change according to the context set. All operations of CDFIS are the same as type-1 FIS.

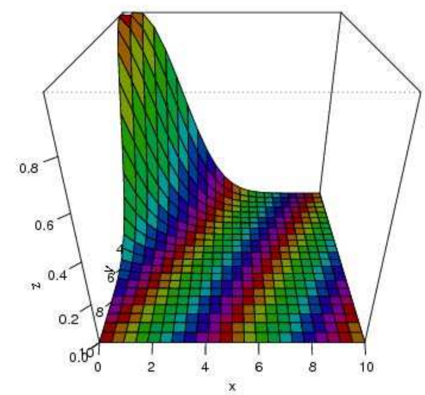

Figure 1. Context-dependent fuzzy set $\dot{A}$ with one-element context set

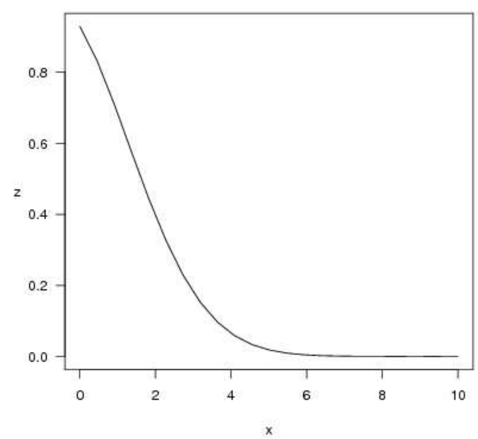

Figure 2. Context-dependent fuzzy set $\dot{A}$ at a specific context $y = 2.2$

III. Simulation Experiments

To verify the effectiveness of the proposed approach, the CDFIS is applied to the simulated car racing competition in FuzzIEEE 2007 conference and the performance of CDFIS is compared with some other types of FISs that took part in the competition.

A. Problem Formulation

1) General rules: The aim of the problem is to race a simulated car to a series of waypoints all within a unit square on a two-dimensional plane. The challenge is to find the best fuzzy controller for this. However, there are several constraints to the current problem that add more interest:

- The car competes against an opponent car.

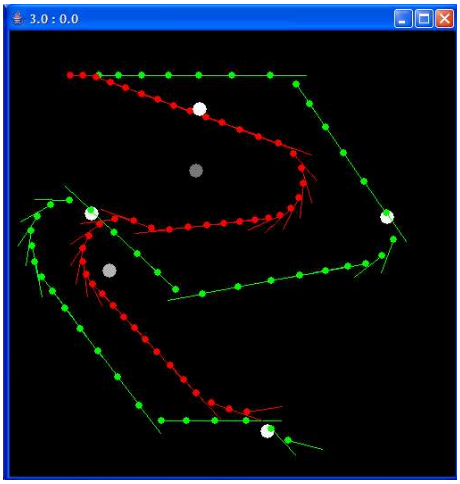

Figure 3. A screen-shot of a race

- At any point in time, each car only knows the position of the current waypoint and the next two waypoints. These must be visited in order by the racing cars, but once one a car has reached a waypoint, then the opponent car can miss this and go directly for the next waypoint.
- The waypoints are randomly distributed around a square area. Every track is different, and tests general driving ability rather than how well specialised a driver is to a particular track
- Only the first car to reach each waypoint scores a point for that waypoint.
- At each step, the position and status of the car are updated according to the returned values of the controller. The car can jump over the target. This means that the position of the car must exactly hit the waypoint in order to be counted as a hit.
- The performance of a controller is measured by the number of waypoints it can pass within a set time limit (500 time steps).
- Skidding and collisions are ignored. This allows for an entirely symmetric race. Each controller is fed identical information, and the cars start in identical positions.
- There are two types of tracks that were used:
  - Noisy track: Gaussian noise is added to the observed position and velocity of the cars as well as to the positions of the waypoints
  - Noiseless track: no noise is added to the observed positions of the cars and the waypoints, nor to the velocity of the cars.

Figure 3 shows an example run between two cars. A car is shown as a circle, with a line to indicate the current heading. In this example, the green car has a faster and more aggressive driving strategy, which has led it outperforming the red in this instance.

2) Controller requirements: At each step, the controller is given the following data:

- The positions of the next three waypoints. The position is represented as a two-dimensional vector.
- The current states of each car, which consist of a velocity vector, a heading vector and a position vector.

Other information such as the radius of the car (a car is represented as a circle) and the radius of the way point are constants. The controller of each car is provided with these inputs and is required to return the steering and acceleration that will be applied to the car at the next step. The steering and acceleration are restricted to given minimum and maximum values. Any values outside these bounds will be truncated. The underlying physics is fairly simple, all motions are based on Newtonian mechanics of point masses.

B. Strategies and implementation of CDFS

There are always two steps that a controller must do

- First, choose a waypoint as the target out of the three visible waypoints.
- Heading towards the chosen target without considering other waypoints.

A simple and fast heuristic controller is used to estimate the number of steps required for each car to hit the first waypoint. If our car takes fewer steps to get to the first waypoint than the opponent car, then the first waypoint is chosen as the target, otherwise the second waypoint is selected. In this paper, we will not focus on the details of the algorithm to choose the targets. From now on, we assume that the target has been selected and the aim of the controller is to find the best way to reach the target.

1) Strategies to reach the target: From experiments, we determined that the car turns faster at higher speed and, as skidding is ignored, it would generally take less time to make a quick u-turn at high speed than to reverse. Therefore we implemented forward motion only. Although there is a large variety of situations that the car can be in, they can be categorized into two alternatives:

- The angle from the car to the target is so small that no steering is needed. The problem simplifies to that of finding the optimal acceleration for the car to hit the target in a straight-line path given the initial velocity (speed) u of the car and the distance to the target s. The optimal acceleration can be determined as follows:
  - The estimate number of step required for the car to hit the target can be calculated by rearranging the equation $v^2 = u^2 + 2as$ as follows:

$$v = u + at = \sqrt{u^2 + 2as}$$

$$t = \frac{\sqrt{u^2 + 2as} - u}{a} \qquad (1)$$

The minimum number of steps must be an integer and is achieved when the acceleration is at its maximum value. Therefore:

$$t_{min} = \left[\frac{\sqrt{u^2 + 2a_{max}s} - u}{a_{max}} + 1 \right] \qquad (2)$$

where a_{max} is the maximum possible acceleration.

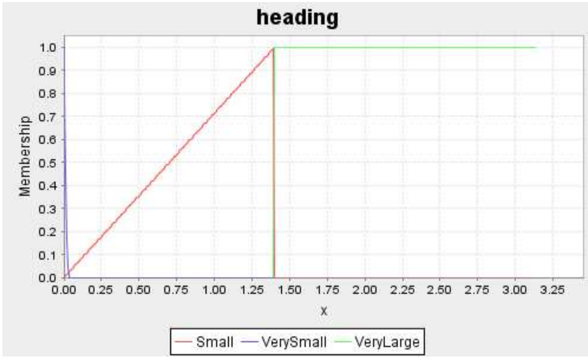

Figure 4. $heading$ and θ

– The acceleration requires to hit the target in t_{min} steps is:

$$a = \frac{2 \times (s - u \times t_{min})}{t_{min}^2} \qquad (3)$$

- The angle from the car to the target requires some steering by the car. The acceleration is set to its maximum value as it will make the car turn quicker. From experiments, the steering output should be roughly proportional to $\frac{heading}{speed}$:

$$steering \approx heading \times 1/speed \qquad (4)$$

where $heading$ is the current angle from the car to the target as shown in Figure 4

No more steering is needed when the current heading angle of the car is very small, or more precisely, when $|heading| \leq \theta$ (where θ is defined as in Figure 4). θ can be easily calculated given the distance to the target as follows:

$$\theta = arctan\left[(r_{car} + r_{waypoint})/s\right] \qquad (5)$$

where r_{car} and $r_{waypoint}$ are the radius of the car and waypoint respectively.

There are situations when the car tries to turn towards the target but ends up in a circular loop and will circle forever. It was determined experimentally that after 15 steps, if the car is not in a the position to go straight, it is likely to be trapped in a loop condition. To avoid loop condition, 15 steps ahead is calculated and if the car is trapped, reverse the current steering ($steering_{new} = -steering_{old}$)

2) Controllers: In this paper, three different types of controllers are implemented using the same strategy described above, each controller will compete with each other head-to-head and the one with the most wins is the winner. The three controllers are a fuzzy type-1 controller, a non-stationary fuzzy controller and a context-dependent fuzzy controller. See Figure 5 for an overview of the structure of the controllers.

As mentioned above, acceleration is set to maximum when the car is turning and set to a specific value when no steering is needed using a simple formula (3). The fuzzy system is used only to work out the steering angle of the car. Although the fuzzy system can only approximate the steering output in equation (4), it is surprised that the results achieved are even better than those using the formula directly, especially in noisy tracks. In this paper, only the results of fuzzy controllers are analyzed and discussed.

The structures of all three controllers are very similar. The inputs of the fuzzy systems are heading angle and inverse of speed, and the output is steering. All fuzzy systems use the same set of rules. Membership functions, parameters and the set of rules are chosen empirically. There is no training or optimizations involved. The rules are:

```
R1 : IF heading IS VerySmall
THEN steer IS VerySmall;
R2 : IF heading IS VeryLarge
THEN steer IS Large;
R3 : IF heading IS Large
AND inverseSpeed IS Large
THEN steer IS VeryLarge;
```

To simplify the problem, we consider only the case where $heading \geq 0$; for $heading < 0$, the sign is reversed. The second rule applies to the situation where the heading angle of the car is larger than the maximum possible angle that the car can turn, in which case the steering should be set to a maximum value. The third rule applies to the general situation where steering is approximately equal to the product of heading and $1/speed$. The first rule will set the steering to 0 in the case that the heading angle is very small, so that no more steering is needed. The definition of the term "*VerySmall*" encapsulates the meaning of that angle and the membership function of the term *VerySmall* is different for each fuzzy system. The defuzzication method used by the 3 fuzzy systems is LeftMax, which takes the left most maximum value of the output space. Three different types of membership functions are used: Gaussian, Triangular and Trapezoidal membership functions. The denotation of these membership functions can be formalised as follows:

- Gaussian membership function with centre c and standard deviation σ

$$gauss\ c\ \sigma = \mu(x, c, \sigma) \ = \ e^{-\frac{(x-c)^2}{2\sigma^2}}$$

- Trapezoidal function:

$$trape\ a\ b\ c\ d \ = \ \mu(x, a, b, c, d) \ = \begin{cases} 0, & x < a,\ x > d \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b < x < c \\ \frac{d-x}{d-c}, & c \leq x \leq d \end{cases}$$

- Triangular function:

$$trian\ a\ b\ c \ = \mu(x, a, b, c) \ = \ max\left(min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$$

Table I shows the details of all variables and their membership functions. Figures 6, 7 and 8 show the graphs representations of the variables *heading*, *inverseSpeed* and *steer* respectively. In figure 7, the scale of the figure is so

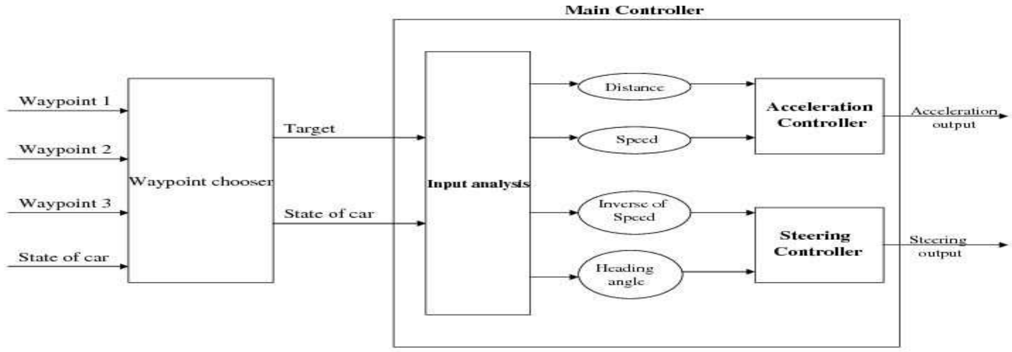

Figure 5. Structure of the controllers

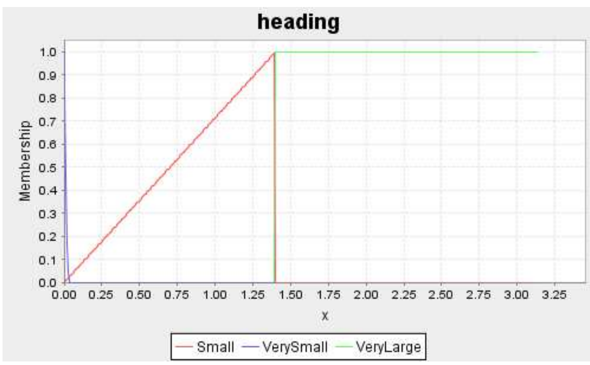

Figure 6. The underlying membership functions of fuzzy variable '*heading*' in case of NSFIS and type-1 FIS

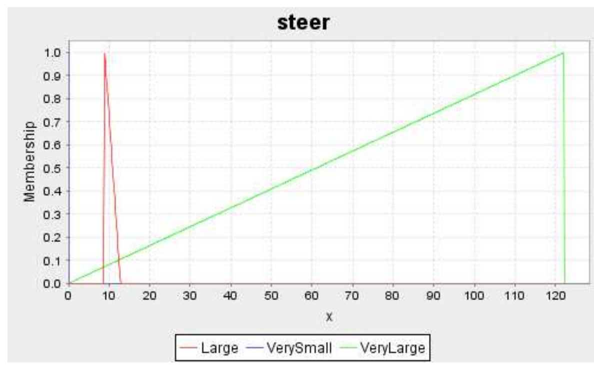

Figure 8. Membership functions of fuzzy variable '*steer*'

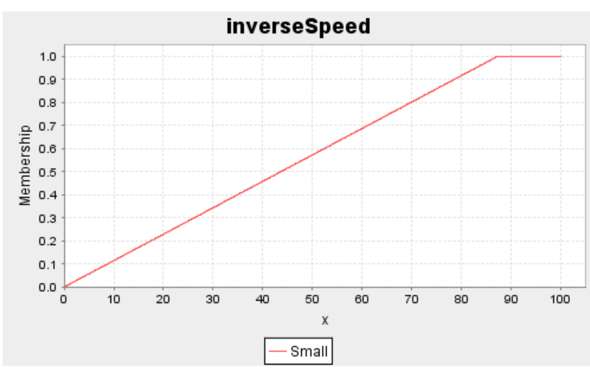

Figure 7. Membership functions of fuzzy variable '*inverseSpeed*'

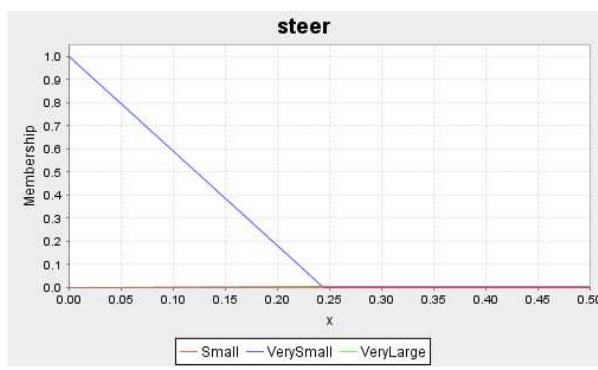

Figure 9. Membership functions of term *VerySmall* of fuzzy variable '*steer*'

```
FUZZIFY heading
TERM VerySmall := ?;
TERM Large := trian 0 1.4 1.41;
TERM VeryLarge := trape 1.38 1.4
3.14 3.14;
RANGE := (-3.14159 .. 3.14159);
END_FUZZIFY
```

```
FUZZIFY inverseSpeed
TERM Large := trape 0 87.267 100 100;
RANGE := (0 .. 100);
END_FUZZIFY
```

```
DEFUZZIFY steer
TERM VerySmall := gauss 0 0.01;
TERM VeryLarge := trian 8.6 8.9 12.8;
TERM Small := trian 0 122.173
122.173;
ACCU : MAX;
METHOD : LM;
DEFAULT := 0.0;
RANGE := (-122.1730 .. 122.173047);
END_DEFUZZIFY
```

Table I
DETAILS OF MEMBERSHIP FUNCTIONS

small that the blue Gaussian membership function of the term '*VerySmall*' is seen as a straight line. Figure 9 is the enlargement version which shows the membership functions of '*VerySmall*' in more details.

The membership function of the term '*VerySmall*' of the variable *heading* of the fuzzy type-1 FIS is defined as a Gaussian membership function with centre of 0 and width of 0.1

```
TERM VerySmall := gauss 0 0.1 ;
```

The NSFIS only used one instantiation, the underlying membership function is the same as the type-1 system. However, at each inference, both the centre and standard deviation are perturbed by a small Gaussian distributed random number. The NSFIS used in this paper is the winning entry at the FuzzIEEE 2007 competition at Imperial College, UK.

The CDFIS used the trapezoidal membership function with context-dependent parameters to define the term "*VerySmall*"

TERM *VerySmall* := trape 0 0 θ θ;

where θ is defined as in (5)

$$\theta = arctan\left[(r_{car} + r_{waypoint})/s\right]$$

where r_{car} and $r_{waypoint}$ are the radius of the car and waypoint respectively and s is the distance from the car to the target.

The context set $\mathcal{C}$ in this case contains only the variable *distance* ($\mathcal{C} = \{distance\}$). θ can be considered a function of distance and therefore will change when the distance changes. The context-dependent fuzzy set of a term '*VerySmall*' of a linguistic variable $heading$ is defined as:

$$\overline{VerySmall} = \int_{s\in\mathcal{C}} \int_{x\in X_{heading}} \mu_{\overline{VerySmall}}(c,x)/x/c.$$

where $\mu_{\overline{VerySmall}}(c,x)$ is the trapezoidal function with four parameters 0, 0, θ, θ

C. Results

The three controllers competed with each other in a round-robin manner, with 500 steps each match in the total of 500 matches in two rounds. Table II and III show the results of each pair in case of normal and noisy tracks respectively. The value at each cell represents the difference between the number of waypoints collected by the controller specified by the row compared to the controller specified by the column.

	Type-1 FIS	NSFIS	CDFIS
Type-1 FIS	-	-118	141
NSFIS	236	-	221
CDFIS	-115	-275	-

Table II
RESULTS FOR NOISELESS TRACKS

	Type-1 FIS	NSFIS	CDFIS
Type-1 FIS	-	-113	-391
NSFIS	-13	-	-443
CDFIS	285	248	-

Table III
RESULTS FOR NOISY TRACKS

NSFIS performs better than fuzzy type-1 FIS in both cases, although the difference between them in the case of noisy tracks is not significant. CDFIS performs slightly worse in noiseless case. However, it outperforms the other two fuzzy controllers when random noise is added. Overall, the CDFIS has the best score amongst the three controllers, and gets a better score than the other two fuzzy controllers in stochastic circumstances.

IV. DISCUSSION

This paper describes the simulated car racing competition that was arranged as part of last year FuzzIEEE 2007 conference. The details of the winning non-stationary fuzzy controllers are also included. In an attempt to further improve the performance of the fuzzy controllers, we subsequently devised a new methodology for fuzzy inferencing that we term a context-dependent fuzzy inferencing system utilising CDFS. These sets have been created with the specific intention of modelling the effects of the context environment on the inference process. While similar to non-stationary fuzzy sets in some regards, context-dependent fuzzy sets possess some important distinguishing features. The first is that a context-dependent fuzzy set does not have a fixed underlying type-1 fuzzy set. This allows the membership functions of a context dependent fuzzy sets to change freely instead of perturbing around the underlying type-1 membership function. The second feature is that context-dependent fuzzy sets could be used to model the co-relation among the inputs of a fuzzy system. Finally, the use of context-dependent fuzzy sets in fuzzy logic systems is similar to that of ordinary type-1 which would be more computationally efficient than non-stationary fuzzy systems. In fact, non-stationary fuzzy sets can be thought of as a special case of context-dependent fuzzy sets in which the context set contains the time variable t. Context-dependent fuzzy systems also differ from other methodologies which change the original membership functions such as fuzzy optimization, fuzzy learning or fuzzy linguistic modifiers, etc. These approaches often require permanent shifts of the membership functions and are intended to optimize the fitness of the system, while in context-dependent fuzzy systems, the changes of membership functions are temporary and serve the purpose of modelling the effects of contexts on the inference process.

Although the racing problem that we used in this paper is fairly simple, in that no skidding or collision is taken into account, it is enough to demonstrate the effectiveness of CDFIS in control applications. We have deliberately chosen a very good type-1 fuzzy system, which has better results than the CDFIS in noiseless tracks, to emphasize that a CDFIS can achieve very good performance in the case of stochastic noise being present. A much more complex simulated car racing problem, where the dynamics of the car is realistic, skidding and collision are taken into account and the output is the set of commands such as *left, right, forward, backward, forwardleft*, etc instead of exact acceleration and steering, has been used as a competition at the Congress of Evolutionary Computing 2007 conference (CEC07). A CDFIS based on the principles described in this paper won that competition. For more information on the CEC competition, please refer to [15]

In summary, more work needs to be done on context-dependent fuzzy sets before any definitive claims can be made in regard to their usefulness. However, even at this early stage, we believe that context-dependent fuzzy sets are novel and we have demonstrated their effectiveness in the presented application. Our research on applying context-dependent fuzzy sets in human decision making as well as control systems is ongoing and issues surrounding both will be further explored.

APPENDIX

Operations on Context-dependent Fuzzy Sets

In this section, the operators of *union*, *intersection* and *complement* of context-dependent sets are introduced. Suppose that we have two context-dependent fuzzy sets, $\bar{A}$ and $\bar{B}$, characterised by membership functions $\mu_{\bar{A}}(c,x)$ and $\mu_{\bar{B}}(c,x)$ where $c \in \mathcal{C}$

$$\bar{A} = \int_{c\in\mathcal{C}} \int_{x\in X} \mu_{\bar{A}}(c,x)/x/c.$$

$$\bar{B} = \int_{c\in\mathcal{C}} \int_{x\in X} \mu_{\bar{B}}(c,x)/x/c.$$

Definition 1: The *union* of $\bar{A}$ and $\dot{B}$ is a context-dependent fuzzy set $\dot{A} \cup \dot{B}$ such that:

$$\bar{A} \cup \bar{B} = \int_{c\in\mathcal{C}} \int_{x\in X} \mu_{\bar{A}\cup\bar{B}}(c,x)/x/c$$

where

$$\mu_{\bar{A}\cup\bar{B}}(c,x) = \mu_{\bar{A}}(c,x) \oplus \mu_{\bar{B}}(c,x)$$

Using the maximum t-conorm, this becomes:

$$\mu_{\bar{A}\cup\bar{B}}(c,x) = max\Big(\mu_{\bar{A}}(c,x), \mu_{\bar{B}}(c,x)\Big)$$

Definition 2: The *intersection* of $\bar{A}$ and $\bar{B}$ is a context-dependent fuzzy set $\bar{A} \cap \bar{B}$ such that:

$$\bar{A} \cap \bar{B} = \int_{c\in\mathcal{C}} \int_{x\in X} \mu_{\bar{A}\cap\bar{B}}(c,x)/x/c$$

where

$$\mu_{\bar{A}\cap\bar{B}}(c,x) = \mu_{\bar{A}}(c,x) \otimes \mu_{\bar{B}}(c,x)$$

Which, using minimum t-norm, becomes

$$\mu_{\bar{A}\cap\bar{B}}(c,x) = min\Big(\mu_{\bar{A}}(c,x), \mu_{\bar{B}}(c,x)\Big)$$

Definition 3: The *complement* of $\bar{A}$ is a context-dependent fuzzy set $\overline{\bar{A}}$ such that:

$$\overline{\bar{A}} = \int_{c\in\mathcal{C}} \int_{x\in X} \mu_{\overline{\bar{A}}}(c,x)/x/c$$

where

$$\mu_{\overline{\bar{A}}}(c,x) = \overline{\mu_{\bar{A}}(c,x)}$$

Which, using the standard complement becomes:

$$\mu_{\overline{\bar{A}}}(c,x) = 1 - \mu_{\bar{A}}(c,x)$$

REFERENCES

[1] J. Mendel and R. John, "Type-2 fuzzy sets made simple," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 117–127, 2002.

[2] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning - I," *Information Sciences*, vol. 8, pp. 199–249, 1975.

[3] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*. New York: Academic, 1980.

[4] J. Mendel and R. I. John, "Footprint of uncertainty and its importance to type-2 fuzzy sets," in *Proceedings of the 6th IASTED International Conference on Artificial Intelligence and Soft Computing*, 2002, pp. 587–592.

[5] J. Mendel, R. John, and F. Liu, "Interval type-2 fuzzy logic systems made simple," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 6, pp. 808–821, 2006.

[6] M. Mizumoto and K. Tanaka, "Some properties of fuzzy sets of type-2," *Information and Control*, vol. 31, pp. 312–340, 1976.

[7] K.P.Valavanis and G.N.Saridis, "Probabilistic modelling of intelligent robotic systems," *IEEE Trans. Robot. Autom.*, vol. 7, no. 1, pp. 164–171, Feb 1991.

[8] H. Deshuang and M. Songde, "A new radical basis probabilistic neural network model," *Proc. 3rd Int. Conf. Signal*, vol. 2, pp. 1449–1452, 1996.

[9] Z. Liu and H.-X. Li, "A probabilistic fuzzy logic system for modelling and control," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 6, pp. 848–859, December 2005.

[10] J. M. Garibaldi, M. Jaroszewski, and S. Musikasuwan, “Non-stationary fuzzy sets,” *IEEE Transactions on Fuzzy Systems*, vol. 1, pp. 396–410, 2006.

[11] T. Ozen and J. M. Garibaldi, “Effect of type-2 fuzzy membership function shape on modelling variation in human decision making,” in *IEEE International Conference on Fuzzy Systems*, Budapest, Hungary, July 2004.

[12] T. Ozen, J. M. Garibaldi, and S. Musikasuwan, “Modelling the variation in human decision making,” in *Fuzzy Sets in the Heart of Canadian Rockies, NAFIPS-04*, Banff, Alberta, Canada, June 2004.

[13] ——, “Preliminary investigations into modelling the variation in human decision making,” in *Information Processing and Management of Uncertainty in Knowledge Based Systems*, Perugia, Italy, July 2004.

[14] S. M. Lucas, *FuzzIEEE 2007 Simulated Car Racing Competition*, 2007. [Online]. Available: http://cswww.essex.ac.uk/staff/lucas/fuzzieee/FuzzyRace.html

[15] J. Togelius and S. M.Lucas, *IEEE CEC 2007 Car Racing Competition*, 2007. [Online]. Available: http://julian.togelius.com/cec2007competition/