A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots

Hani A. Hagras, *Member, IEEE*

***Abstract*—Autonomous mobile robots navigating in changing and dynamic unstructured environments like the outdoor environments need to cope with large amounts of uncertainties that are inherent of natural environments. The traditional type-1 fuzzy logic controller (FLC) using precise type-1 fuzzy sets cannot fully handle such uncertainties. A type-2 FLC using type-2 fuzzy sets can handle such uncertainties to produce a better performance. In this paper, we present a novel reactive control architecture for autonomous mobile robots that is based on type-2 FLC to implement the basic navigation behaviors and the coordination between these behaviors to produce a type-2 hierarchical FLC. In our experiments, we implemented this type-2 architecture in different types of mobile robots navigating in indoor and outdoor unstructured and challenging environments. The type-2-based control system dealt with the uncertainties facing mobile robots in unstructured environments and resulted in a very good performance that outperformed the type-1-based control system while achieving a significant rule reduction compared to the type-1 system.**

***Index Terms*—Behavior-based mobile robots, hierarchical controllers, reactive navigation, type-2 fuzzy logic systems.**

I. Introduction

REACTIVE autonomous mobile robots navigating in real-world unstructured environments (i.e., environments that have not been specifically engineered for the robot) must be able to operate under conditions of imprecision and uncertainty present in such environments. The choice of adequate methods to model and handle such uncertainties is crucial for mobile robots navigating in unstructured environments.

The fuzzy logic controller (FLC) is credited with being an adequate methodology for designing robust controllers that are able to deliver a satisfactory performance in face of uncertainty and imprecision [20], as a result the FLC has become a popular approach to reactive mobile robot control in recent years [20], [21]. There are many sources of uncertainty facing the FLC for a mobile robot navigating in changing and dynamic unstructured environments; we list some of them as follows.

- Uncertainties in inputs to the FLC which translates to uncertainties in the antecedents membership functions as the sensors measurements are typically noisy and are affected by the conditions of observation (i.e., their characteristics are changed by the environmental conditions such as wind, sunshine, humidity, rain, etc.).
- Uncertainties in control actions which translates to uncertainties in the output membership functions of the FLC. Such uncertainties can result from the change of the actuators characteristics which can be due to the inconsistency of the terrain or due to the environmental changes. For example, a "fast speed" on a sunny day with dry ground may be different from "fast speed" on a rainy day with muddy ground as wheels may slip, etc.
- Linguistic uncertainties as the meaning of words that are used in the antecedents and consequents linguistic labels can be uncertain as words mean different things to different people [18]. In addition, experts do not always agree and they often provide different consequents for the same antecedents. A survey of experts will usually lead to a histogram of possibilities for the consequent of a rule, this histogram represents the uncertainty about the consequent of a rule [19].

All of these uncertainties translate into uncertainties about fuzzy set membership functions [18]. These uncertainties cause difficulty in determining the exact and precise antecedent and consequent fuzzy membership functions.

To date all the FLC implementations in robot control are based on the traditional type-1 FLC. There are different ways to construct type-1 FLCs for mobile robots, the most common way is to construct the FLC by eliciting the fuzzy rules and the (input and output) membership functions based on expert knowledge or through the observation of the actions of a human operator controlling the mobile robots [3], [15]. Several researchers [2], [4]–[7], [16] have also explored the use of learning techniques to learn the type-1 FLC for mobile robot control.

The type-1 FLCs (designed using human experience or using learning mechanisms) have the common problem that they cannot fully handle or accommodate for the linguistic and numerical uncertainties associated with changing and dynamic unstructured environments as they use precise type-1 fuzzy sets. Type-1 fuzzy sets handles the uncertainties associated with the FLC inputs and outputs by using *precise and crisp* membership functions that the user believes capture the uncertainties [17]. Once the type-1 membership functions have been chosen, all the uncertainty disappears, because type-1 membership functions are totally precise [8], [17]. The linguistic and numerical uncertainties associated with changing unstructured environments cause problems in determining the exact and precise antecedents and consequents membership functions during the FLC design. Moreover, the designed type-1 fuzzy sets can be suboptimal under specific environment and robot conditions, however because of the robot and environment changes and

Manuscript received October 1, 2003; revised June 2, 2004.
The author is with the Department of Computer Science, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: hani@essex. ac.uk).
Digital Object Identifier 10.1109/TFUZZ.2004.832538

the associated uncertainties (like operating in windy weather rather than still weather under which the designed FLC was suboptimal), the chosen type-1 fuzzy sets might not be appropriate anymore. This can cause degradation in the mobile robot FLC performance and we might end up wasting time in frequently redesigning or tuning the type-1 FLC so that it can deal with the various uncertainties faced in changing unstructured environments.

A type-2 fuzzy set is characterized by a fuzzy membership function, i.e., the membership value (or membership grade) for each element of this set is a fuzzy set in [0,1], unlike a type-1 fuzzy set where the membership grade is a crisp number in [0,1] [13]. The membership functions of type-2 fuzzy sets are three dimensional and include a footprint of uncertainty, it is the new third dimension of type-2 fuzzy sets and the footprint of uncertainty that provide additional degrees of freedom that make it possible to directly model and handle uncertainties [17], [18]. The type-2 fuzzy sets are useful where it is difficult to determine the exact and precise membership functions [13] (which are the case for mobile robots in unstructured environments). The type-2 FLC presented in this paper uses type-2 fuzzy sets to represent the inputs and outputs of the FLC.

Using an FLC (type-1 or type-2) for mobile robot control has the problem that rules increase *exponentially* with the number of variables involved and as the robot has a large number of inputs and outputs, this results in huge rule bases which cause problems for the robot's real time performance and the FLC design. To cope with this problem, a common strategy is to *hierarchically* decompose the control problem by breaking down the input space for analysis by sharing it amongst multiple low level behaviors, each of which responds to specific types of situations, and then integrating the recommendations of these behaviors via a high level coordination layer [21], [22]. Previous work [2], [5], [6], [20]–[22] had implemented the type-1 hierarchical FLC (HFLC), which used type-1 fuzzy systems for the implementation of the basic behaviors and the high level coordination layer. In this paper, we will present the type-2 HFLC which uses type-2 FLC to implement the basic behaviors and the high level coordination layer. This type-2 HFLC will have many advantages in terms of performance and rule reduction when compared with the type-1 HFLC as will be shown in this paper.

To the author's knowledge, no other work in the literature had investigated the use of type-2 FLC and HFLC to mobile robot real time control in unstructured environments. There is only one relevant paper produced by Wu [24] which used a type-1 FLC based on a type-1 fuzzy interval system and he did not address the use of type-2 fuzzy sets and how to implement a real time type-2 FLC or HFLC.

In Section II, we will review the type-2 fuzzy sets and their associated terminologies. In Section III, we will introduce the type-2 FLC and its various components. In Section IV, we will introduce the type-2 HFLC. In Section V, we will present the application of the type-2 HFLC to mobile robot control in indoor and outdoor unstructured environments. In Section VI, we will present the experiments and results. Finally, conclusions are presented in Section VII.

II. Type-2 Fuzzy Sets and Their Associated Terminologies

A type-2 fuzzy set, denoted $\widetilde{A}$ is characterized by a type-2 membership function $\mu_{\widetilde{A}}(x,u)$ [18], where $x \in X$ and $u \in J_x \subseteq [0,1]$, i.e.,

$$\widetilde{A} = \{((x,u), \mu_{\widetilde{A}}(x,u)) | \quad \forall\, x \in X \quad \forall\, u \in J_x \subseteq [0,1]\} \tag{1}$$

in which $0 \leq \mu_{\widetilde{A}}(x,u) \leq 1$. $\widetilde{A}$ can also be expressed as follows [18]:

$$\widetilde{A} = \int_{x\in X} \int_{u \in J_x} \mu_{\widetilde{A}}(x,u)/(x,u) \quad J_x \subseteq [0,1] \tag{2}$$

where $\int\int$ denotes union over all admissible x and u [18].

J_x is called primary membership of x, where $J_x \subseteq [0,1]$ for $\forall x \in X$ [18]. The uncertainty in the primary memberships of a type-2 fuzzy set $\widetilde{A}$, consists of a bounded region that is called the *footprint of uncertainty* (FOU) [18]. It is the union of all primary memberships [18]. Recently, it has been shown that regardless of the choice of the primary membership function (triangle, Gaussian, trapezoid), the resulting FOU is about the same [19]. According to [19], the FOU of a type-2 membership function also handles the rich variety of choices that can be made for a type-1 membership function, i.e., by using type-2 fuzzy sets instead of type-1 fuzzy sets, the issue of which type-1 membership function to choose diminishes in importance. For type-2 fuzzy sets there are new operators named the *meet* (denoted by $\sqcap$) and *join* (denoted by $\sqcup$) to account for the intersection and union [14]. According to [13], a type-2 fuzzy set can be thought of as a large collection of embedded type-1 sets each having a weight to associated with it.

At each value of x say $x = x'$, the 2-D plane whose axes are u and $\mu_{\widetilde{A}}(x',u)$ is called a vertical slice of $\mu_{\widetilde{A}}(x,u)$ [18]. A *secondary membership function* is a vertical slice of $\mu_{\widetilde{A}}(x,u)$. It is $\mu_{\widetilde{A}}(x=x',u)$ for $x' \in X$ and $\forall\, u \in J_{x'} \subseteq [0,1]$ [18], i.e.,

$$\mu_{\widetilde{A}}(x=x',u) \equiv \mu_{\widetilde{A}}(x') = \int_{u\in J_{x'}} f_{x'}(u)/(u) \quad J_{x'} \subseteq [0,1] \tag{3}$$

in which $0 \leq f_{x'}(u) \leq 1$. Because $\forall x' \in X$, the prime notation on $\mu_{\widetilde{A}}(x')$ is dropped and $\mu_{\widetilde{A}}(x)$ is referred to as a secondary membership function [18]; it is a type-1 fuzzy set which is also referred to as a *secondary set* [18]. Many choices are possible for the secondary membership functions. According to [17], the name that we use to describe the entire type-2 membership function is associated with the name of the secondary membership functions. For example, when $f_x(u) = 1$, $\forall u \in J_x \subseteq [0,1]$, then the secondary membership functions are interval sets, and, if this is true for $\forall x \in X$, we have the case of an *interval type-2 membership function* which characterizes the interval type-2 fuzzy sets [17]. Interval secondary membership functions reflect a uniform uncertainty at the primary memberships of x [17]. Since all the memberships in an interval type-1 set are unity, in the sequel, an interval type-1 set is represented just by its domain interval, which can be represented by its left and right end-points as $[l,r]$ [13]. The two end-points are associated with two type-1 membership functions that are referred to as *upper* and *lower*

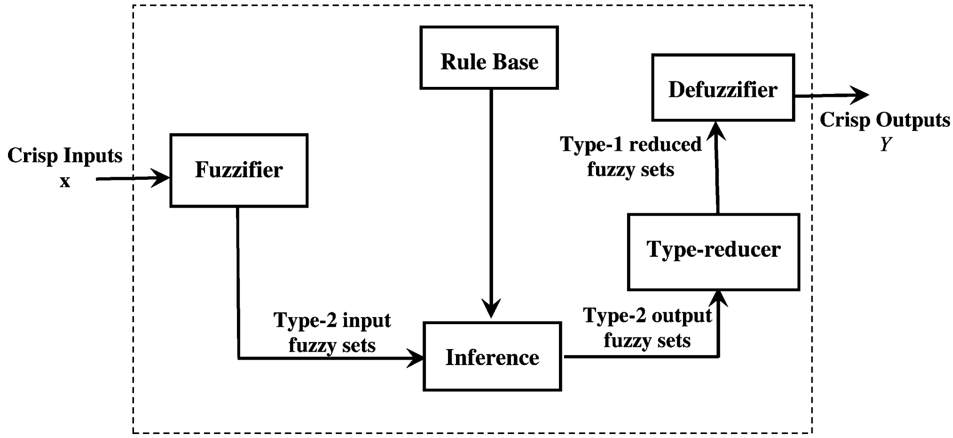


Fig. 1. Type-2 FLC.

membership functions which are bounds for the footprint of uncertainty $\mathrm{FOU}(\widetilde{A})$ [13], [14]. The upper membership function is associated with the upper bound of $\mathrm{FOU}(\widetilde{A})$ and is denoted by $\overline{\mu}_{\widetilde{A}}(x), \forall x \in X$ [17]. The lower membership function is associated with the lower bound of $\mathrm{FOU}(\widetilde{A})$ and is denoted by $\underline{\mu}_{\widetilde{A}}(x), \forall x \in X$ [17].

Using type-2 fuzzy sets to represent the inputs and outputs of a mobile robot FLC has many advantages when compared to the type-1 fuzzy sets, we summarize some of these advantages as follows.

- As the type-2 fuzzy sets membership functions are themselves fuzzy and include a FOU, then they can model and handle the linguistic and numerical uncertainties associated with the inputs and outputs of the robot FLC in changing and dynamic unstructured environments and hence they can handle the difficulty associated with determining the exact membership functions for the fuzzy sets [13]. Therefore, FLCs that are based on type-2 fuzzy sets will have the potential to produce a better performance than the type-1 FLCs.
- Using type-2 fuzzy sets to represent the FLC inputs and outputs will result in the reduction of the FLC rule base when compared to using type-1 fuzzy sets as the uncertainty represented in the footprint of uncertainty in type-2 fuzzy sets lets us cover the same range as type-1 fuzzy sets with smaller number of labels and the rule reduction will be greater when the number of the FLC inputs increases [17].
- Each input and output will be represented by a large number of type-1 fuzzy sets which are embedded in the type-2 fuzzy sets [13], [18]. The use of such a large number of type-1 fuzzy sets to describe the input and output variables allows for a detailed description of the analytical control surface as the addition of the extra levels of classification gives a much smoother control surface and response [9].

III. Type-2 Fuzzy Logic Controller (FLC)

A type-2 FLC is depicted in Fig. 1, it contains five components which are fuzzifier, rule base, fuzzy inference engine, type-reducer and defuzzifier. In developing the type-2 FLC for mobile robot control both the inputs and outputs will be represented by interval type-2 fuzzy sets. We will use interval type-2 fuzzy sets as they are simple to use [18] and they distribute the uncertainty evenly among all admissible primary memberships [17]. Furthermore, the general type-2 FLC is computationally intensive and the computation simplifies a lot when using interval type-2 FLC [14] (using interval type-2 fuzzy sets) which will enable us to design a robot FLC that operates in real time.

The type-2 FLC works as follows, the crisp inputs from the input sensors are first fuzzified into, in general, input type-2 fuzzy sets (however, we will consider only Singleton fuzzification) which then activate the inference engine and the rule base to produce output type-2 fuzzy sets. These output type-2 fuzzy sets are then processed by the type-reducer which combines the output sets and then performs a centroid calculation, which leads to type-1 fuzzy sets called the type-reduced sets [17]. The defuzzifier can then defuzzify the type-reduced type-1 fuzzy outputs to produce crisp outputs to be fed to the actuators.

The uncertainty is handled by the antecedents and consequents interval type-2 fuzzy sets which include FOUs to accommodate the linguistic and numerical uncertainties associated with changing unstructured environments. The interval type-2 fuzzy sets also include a large number of embedded type-1 fuzzy sets and thus according to [11] the type-2 FLC can be thought of as a collection of many different *embedded* type-1 FLCs [11] to deal with the different uncertainties. In the following sections, we will introduce each block in the type-2 FLC.

A. Fuzzifier

The fuzzifier maps a crisp input vector with p inputs $\mathbf{x} = (x_1, \ldots, x_p)^T \in X_1 \times X_2 \ldots \times X_p \equiv \mathbf{X}$ into input fuzzy sets, these fuzzy sets can, in general, be type-2 fuzzy input sets $\widetilde{A}_x$ [13], [17]. However, we will use Singleton fuzzification as it is fast to compute and thus suitable for the robot real time operation. In the Singleton fuzzification, the input fuzzy set has only a single point of nonzero membership [13], [17], i.e., $\widetilde{A}_x$ is a type-2 fuzzy Singleton if $\mu_{\widetilde{A}_x}(\mathbf{x}) = 1/1$ for $\mathbf{x} = \mathbf{x}'$ and $\mu_{\widetilde{A}x}(\mathbf{x}) = 1/0$, for all other $\mathbf{x} \neq \mathbf{x}'$ [17].

B. Rule Base

According to [17], the rules will remain the same as in type-1 FLC but the antecedents and the consequents will be represented by interval type-2 fuzzy sets. Consider a robot type-2 FLC having p inputs $x_1 \in X_1, \ldots, x_p \in X_p$ and c outputs $y_1 \in Y_1, \ldots, y_c \in Y_c$. The ith rule in this multiple-input–multiple-output (MIMO) FLC can be written as follows:

$$R^i_{\mathrm{MIMO}}: \text{IF } x_1 \text{ is } \widetilde{F}^i_1 \text{ and ..and } x_p \text{ is } \widetilde{F}^i_p, \text{ THEN } y_1 \text{ is } \widetilde{G}^i_1 \ldots y_c \text{ is } \widetilde{G}^i_c, \qquad i = 1, \ldots, M \tag{4}$$

where M is the number of rules in the rule base.

According to [17], as the type-2 FLC rule base will have the same structure as the type-1 FLC then we can use the results from [12] which states that R^i_{MIMO} can be considered as a group of multiple-input–multiple-output (MISO) $R^i_{k\mathrm{MISO}}$ rules, where $R^i_{k\mathrm{MISO}}$ is a rule relating the multiple p inputs and the kth single output, where $k = 1, \ldots, c$.

C. Fuzzy Inference Engine

The inference engine combines rules and gives a mapping from input type-2 sets to output type-2 sets [17]. In the inference engine, multiple antecedents in the rules are connected using the *Meet* operation, the membership grades in the input sets are combined with those in the output sets using the extended sup–star composition, multiple rules are combined using the *Join* operation [17].

Each rule in a MISO fuzzy rule base with M rules having p inputs $x_1 \in X_1, \ldots, x_p \in X_p$ and one output $y_k \in Y_k$ can be written as follows [17]:

$$R^i_{k\mathrm{MISO}}: \widetilde{F}^i_1 \times \cdots \times \widetilde{F}^i_p \rightarrow \widetilde{G}^i_k = \widetilde{A}^i \rightarrow \widetilde{G}^i_k, \quad i = 1, \ldots, M. \tag{5}$$

$R^i_{k\mathrm{MISO}}$ is described by the membership function $\mu_{R^i}(\mathbf{x}, y_k) = \mu_{R^i}(x_1, \ldots, x_p, y_k)$ and according to [17] $\mu_{R^i}(\mathbf{x}, y_k) = \mu_{\widetilde{A}^i \rightarrow \widetilde{G}^i_k}(\mathbf{x}, y_k)$ can be written as follows [17]:

$$\begin{aligned} \mu_{R^i}(\mathbf{x}, y_k) &= \mu_{\widetilde{F}^i_1}(x_1) \sqcap .. \sqcap \mu_{\widetilde{F}^i_p}(x_p) \sqcap \mu_{\widetilde{G}^i_k}(y_k) \\ &= \left[\sqcap^p_{a=1} \mu_{\widetilde{F}^i_a}(x_a)\right] \sqcap \mu_{\widetilde{G}^i_k}(y_k). \end{aligned} \tag{6}$$

As we are using the Singleton fuzzification, then the type-2 fuzzy input set $\widetilde{A}_x$ contains a single element $\mathbf{x}'$ and each $\mu_{\widetilde{X}_a}(x_a)$ is non zero only at one point $x_a = x'_a$. In our interval type-2 FLC, we will use the *meet* under product t-norm so the result of the input and antecedent operations, which are contained in the firing set $\sqcap^p_{a=1} \mu_{\widetilde{F}^i_a}(x'_a) \equiv F^i(\mathbf{x}')$, is an interval type-1 set, as follows [17]:

$$F^i(\mathbf{x}') = \left[\underline{f}^i(\mathbf{x}'), \overline{f}^i(\mathbf{x}')\right] \equiv \left[\underline{f}^i, \overline{f}^i\right] \tag{7}$$

where $\underline{f}^i(\mathbf{x}')$ and $\overline{f}^i(\mathbf{x}')$ can be written as follows, where $*$ denotes the product operation:

$$\underline{f}^i(\mathbf{x}') = \underline{\mu}_{\widetilde{F}^i_1}(x'_1) * \ldots\ldots.. * \underline{\mu}_{\widetilde{F}^i_p}(x'_p) \tag{8}$$

$$\overline{f}^i(\mathbf{x}') = \overline{\mu}_{\widetilde{F}^i_1}(x'_1) * \ldots\ldots.. * \overline{\mu}_{\widetilde{F}^i_p}(x'_p). \tag{9}$$

D. Type Reduction

Type-reduction was proposed by Karnik and Mendel [10], [11]. It is called type-reduction because this operation takes us from the type-2 output sets of the inference engine to a type-1 set that is called the "the type-reduced set." These type-reduced sets are then defuzzified to obtain crisp outputs that are sent to the motors of the mobile robots.

As in [14], we will use the center of sets type reduction, as it has reasonable computational complexity that lies between computationally expensive centroid type-reduction and the simple height and modified height type-reductions which have problems when only one rule fires [17]. The computation of center of sets type-reduction will allow for real time operation if the rule base is small, as we will see later. The type-reduced set using the center of sets type-reduction can be expressed as shown in (10) at the bottom of the page [17], where $Y_{\cos}(\mathbf{x})_k$ for kth output is an interval set determined by its left most point y_{lk} and its right most point y_{rk}. $i = (1 \ldots M)$ where M is the number of rules.

y^i_k corresponds to the centroid of the type-2 interval consequent set $\widetilde{G}^i_k$ of the ith rule for the for kth output, y^i_k is a type-1 interval fuzzy set determined by its left most point y^i_{lk} and its right most point y^i_{rk} [14]. f^i denotes the firing strength (degree of firing) of the ith rule which is an interval type-1 set determined by its left most $\underline{f}^i$ and right most point $\overline{f}^i$ [14].

The calculation of the type-reduced sets is divided into two stages. The first stage is the calculation of centroids of the type-2 interval consequent sets of each rule which is conducted ahead of time and before starting the robot FLC operation. The second stage happens each control cycle to calculate the type-reduced sets which are then defuzzified to produce the crisp outputs to the actuators. In the following sections, we describe these two stages.

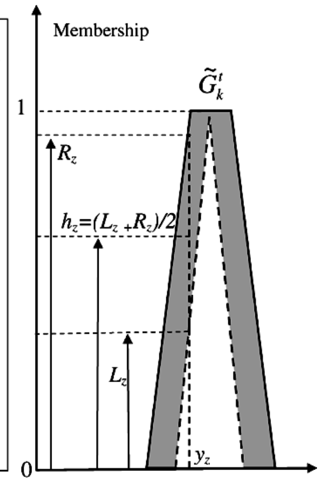
1) Calculating the Centroids of the Rule Consequents: For any output k, the type-2 interval consequent set of the ith rule $\widetilde{G}^i_k$ will be one of the output type-2 interval fuzzy sets $\widetilde{G}^t_k$ representing this output, where $t = 1, \ldots T$ and T is the number of output fuzzy sets representing this output. If for each output we calculated the centroids of all the type-2 interval fuzzy sets representing this output, then the centroid of the type-2 interval consequent set for the ith rule y^i_k will be one of the precalculated centroids of the type-2 output sets y^t_k which corresponds

$$Y_{\cos}(\mathbf{x})_k = [y_{lk}, y_{rk}] = \int_{y^1_k \in [y^1_{lk}, y^1_{rk}]} \cdots \int_{y^M_k \in [y^M_{lk}, y^M_{rk}]} \int_{f^1 \in [\underline{f}^1, \overline{f}^1]} \cdots \int_{f^M \in [\underline{f}^M, \overline{f}^M]} 1 \Big/ \frac{\sum_{i=1}^{M} f^i y^i_k}{\sum_{i=1}^{M} f^i} \tag{10}$$

Without loss of generality assume that y_z are arranged in an ascending order; i.e. $y_1 \le y_2 \le .. \le y_Z$ Then,

1. **Initialise θ_z by setting $\theta_z = h_z$ for $z = 1, \ldots, Z$ and then compute $y' = y(h_1, \ldots h_z)$ using Equation (12).**
2. **Find e $(1 \le e \le Z-1)$ such that $y_e \le y' \le y_{e+1}$.**
3. **Set $\theta_z = h_z - \Delta_z$, for $z \le e$ and $\theta_z = h_z + \Delta_z$, for $z \ge e+1$ and compute $y'' = y(h_1 - \Delta_1, \ldots h_e - \Delta_e, h_{e+1} + \Delta_{e+1}, \ldots, h_Z + \Delta_Z)$ using Equation (12)**
4. **Check if $y'' = y'$. If yes stop, y'' is the maximum value of $y(\theta_1, \ldots \theta_Z)$ and it is y'_{rk}, if no go to step 5**
5. **Set y' equal y''. Go to step 2.**

(a)



(b)

Fig. 2. (a) Iterative procedure to calculate y_{rk}^t. (b) Parameters needed by each y_z in the procedure in Fig. 2(a).

to the rule consequent. We must calculate all centroids y_k^t for $t = 1, \ldots, T$ ahead of time, and before starting the robot FLC operation, as they are needed for the computation of $Y_{\cos}(\mathbf{x})_k$ [17].

The centroid of the tth output fuzzy set y_k^t is a type-1 interval set determined by its left most point y_{lk}^t and its right most point y_{rk}^t which can be obtained from the following equation [17]:

$$y_k^t = [y_{lk}^t, y_{rk}^t] = \int_{\theta_1 \in J_{y1}} \cdots \cdot \int_{\theta_Z \in J_{yZ}} 1 \Big/ \frac{\sum_{z=1}^{Z} y_z \theta_z}{\sum_{z=1}^{Z} \theta_z}. \tag{11}$$

To compute the centroid of each output fuzzy set we will use the iterative procedure developed in [10] and [17]. Fig. 2(a) shows this procedure to compute y_{rk}^t. In this procedure for the kth output, we will discretise each output fuzzy set into Z points, $y_1, \ldots . y_z$ where $z = (1 \ldots Z)$. Let $J_{y_z} \equiv [L_z, R_z]$ and $h_z = (L_z + R_z)/2$ and $\Delta_z = (R_z - L_z)/2$. Fig. 2(b) shows for each y_z how to find L_z, R_z, h_z needed by the iterative procedure. We need to maximize and minimize y as a function of $(\theta_1, \ldots . \theta_z)$ to compute the end points, i.e., optimize [17]

$$y(\theta_1, \ldots . \theta_z) = \frac{\sum_{z=1}^{Z} y_z \theta_z}{\sum_{z=1}^{Z} \theta_z}. \tag{12}$$

To compute y_{rk}^t we will apply the iterative procedure shown in Fig. 2(a). The value of y_{lk}^t can be obtained using a similar procedure to the one used in Fig. 2(a) but doing one change ; In step 3, set $\theta_z = h_z + \Delta_z, for\ z \le e\ and\ \theta_z = h_z - \Delta_z, for\ z \ge e+1$ and compute $y'' = y(h_1 + \Delta_1, \ldots . h_e + \Delta_e, h_{e+1} - \Delta_{e+1}, \ldots, h_Z - \Delta_Z)$ using (12).

This iterative procedure converges in at most Z iterations to find y_{rk}^t and Z iterations to find y_{lk}^t [17]. Where each iteration consists of one pass through steps 2–5 as step 1 is an initialization step. We have chosen Z in our robot controller to be 100, however as mentioned above the calculations of the consequent centroids is done only once before the robot begins moving and is not part of the control cycle.

2) Calculating the Type-Reduced Set: For any output k to compute $Y_{\cos}(\mathbf{x})_k$ in (10) we need to compute its two end points y_{lk} and y_{rk}. For each rule, we need to attach to the firing strength f^i the centroid of the ith rule consquent y_k^i calculated in the previous step. According to [17], let the values of f^i and y_k^i that are associated with y_{lk} be denoted f_l^i and y_{lk}^i, respectively, and the values of f^i and y_k^i that are associated with y_{rk} be denoted f_r^i and y_{rk}^i, respectively. From (10), we see that [17]

$$y_{lk} = \frac{\sum_{i=1}^{M} f_l^i y_{lk}^i}{\sum_{i=1}^{M} f_l^i} \tag{13}$$

$$y_{rk} = \frac{\sum_{i=1}^{M} f_r^i y_{rk}^i}{\sum_{i=1}^{M} f_r^i}. \tag{14}$$

In order to compute y_{lk} we need to determine $\{f_l^i, i = 1, \ldots, M\}$ and its associated $\{y_{lk}^i, i = 1, \ldots, M\}$ and to compute y_{rk} we need to determine $\{f_r^i, i = 1, \ldots . M\}$ and its associated $\{y_{rk}^i, i = 1, \ldots . M\}$ [17]. This can be done by using the procedure developed in [14] and [17]. Fig. 3 shows the procedure to compute y_{rk} [14], [17], it is a four step iterative procedure where step 1 is an initialization step.

Observe that in this procedure the number R is very important. For $i \le R$ $f_r^i = \underline{f}^i$ where as for $i > R$ $f_r^i = \overline{f}^i$, hence, y_{rk} in (14) can be written as follows [14]:

$$y_{rk} = \frac{\sum_{u=1}^{R} \underline{f}^u y_{rk}^u + \sum_{v=R+1}^{M} \overline{f}^v y_{rk}^v}{\sum_{u=1}^{R} \underline{f}^u + \sum_{v=R+1}^{M} \overline{f}^v}. \tag{15}$$

The procedure for computing y_{lk} is similar to the one given to calculate y_{rk} just replace y_{rk}^i by y_{lk}^i and in step 2 find $L(1 \le L \le M-1)$ such that $y_{lk}^L \le y'_{lk} \le y_{lk}^{L+1}$. Additionally, in step

Without loss of generality, assume that the pre-computed y_{rk}^i are arranged in an ascending order; i.e. $y_{rk}^1 \le y_{rk}^2 \le .. \le y_{rk}^M$. *Then,*

1. Compute y_{rk} in Equation (14) by initially setting $f_r^i = (\underline{f}^i + \overline{f}^i)/2$ for *i=1,….M* where $\underline{f}^i$ and $\overline{f}^i$ have been previously computed using Equation (8) and Equation(9) respectively and let $y_{rk}' \equiv y_{rk}$

2. Find *R (1≤R ≤M-1)* such that $y_{rk}^R \le y_{rk}' \le y_{rk}^{R+1}$

3. Compute y_{rk} in Equation (14) using $f_r^i = \underline{f}^i$ for *i≤R* and $f_r^i = \overline{f}^i$ for *i>R* and let $y_{rk}'' \equiv y_{rk}$

4. If $y_{rk}'' \ne y_{rk}'$ then go to step 5. If $y_{rk}'' = y_{rk}'$ then stop and set $y_{rk}'' \equiv y_{rk}$

5. Set y_{rk}' equal to y_{rk}'' and return to step 2.

Fig. 3. Iterative procedure to calculate y_{rk}.

3, compute y_{lk} in (13) with $f_l^i = \overline{f}^i$ for $i \le L$ and $f_l^i = \underline{f}^i$ for $i > L$, then y_{lk} in (13) can be written as follows [14]:

$$y_{lk} = \frac{\sum_{u=1}^{L} \overline{f}^u y_{lk}^u + \sum_{v=L+1}^{M} \underline{f}^v y_{lk}^v}{\sum_{u=1}^{L} \overline{f}^u + \sum_{v=L+1}^{M} \underline{f}^v}. \tag{16}$$

This iterative procedure is proven [14], [25] to converge in no more than M iterations to find y_{rk} and M iterations to find y_{lk} where M is the number of rules. The procedure will be used each control cycle to compute the type-reduced sets which will be then defuzzified to give the crisp outputs to the actuators.

E. Defuzzification

From the type-reduction stage, we have for each output a type-reduced set $Y_{\cos}(\mathbf{x})_k$ determined by its left most point y_{lk} and right most point y_{rk}. We defuzzify the interval set by using the average of y_{lk} and y_{rk}, hence, the defuzzified crisp output for each output k is [14], [17]

$$Y_k(\mathbf{x}) = \frac{y_{lk} + y_{rk}}{2}. \tag{17}$$

IV. TYPE-2 HIERARCHICAL FUZZY LOGIC CONTROLLER

A. Hierarchical Fuzzy Systems

Single rule base FLC (type-1 or type-2) suffer from the serious limitation that the number of rules increases *exponentially* with the number of variables involved [23]. For our robots (indoor and outdoor) we have eight input sensors (seven sonar sensors and one goal detection sensor), if we represented each input by only three fuzzy sets then for a single rule base we need to determine $3^8 = 6561$ rules, which is very difficult to design, also this huge rule base translates directly to slower controller response.

For interval type-2 FLCs, computing the type-reduced fuzzy set is directly proportional to the number of rules and for a large rule base, it will take long time each control cycle to converge to the type-reduced set thus limiting the real time application of the type-2 FLC [25]. Real time is defined as producing a control action in time enough to be useful [1] (50 ms for our robots). For our mobile robots using a single rule base we might have a rule base of $3^8 = 6561$ rules. In this case, for each output k we might need maximum 6561 iterations to determine y_{rk} and maximum 6561 iterations to determine $y_{lk,}$, thus we might need for each output 13122 iterations each control cycle to finish the type-reduction and produce an output which will take long time and might disturb the type-2 FLC real time operation. Wu and Mendel [25] introduced a method to approximate the type-reduced set by the inner and outer bound sets, however this method needs training data to properly design a FLC based on minimizing a risk function to achieve similar outputs to the type-reduced outputs. However, this data is difficult to acquire for a mobile robot which encounters a lot of unforeseen circumstances and will navigate mostly in unknown environments.

To cope with *the rule explosion problem* and its effects on the design and real-time operation of the interval type-2 FLCs, we will hierarchically decompose the control problem by breaking down the input space for analysis by sharing it amongst multiple low level type-2 behaviors. Each behavior responds to specific types of situations, and then integrating the recommendations of these behaviors via a high level type-2 coordination layer. Each behavior is an independent and self contained interval type-2 FLC with a small number of inputs and outputs and a small rule base and it serves a single purpose (e.g., edge-following or obstacle-avoidance) while operating in a reactive fashion. The behaviors will typically (but not necessarily) map different inputs sensors to common actuators outputs [22]. Such primitive behaviors are building blocks for more intelligent composite behaviors, i.e., their capabilities can be combined through synergistic coordination by a high level fuzzy coordination layer to obtain an overall coherent behavior that achieves the intended task(s). Fuzzy coordination allows the ability to express partial and concurrent activations of behaviors; and the smooth transition between behaviors.

The hierarchical fuzzy systems have a nice property that the total number of rules increases linearly rather than exponentially as in the single rule base FLC [23]. For example in a type-1 system implemented in [5], the robot controller was divided into four cooperating behaviors namely obstacle avoidance, goal seeking, left, and right edge following [5]. We represented each input using three fuzzy sets (as in the case of a single rule base FLC) then the obstacle avoidance behavior, using three forward facing sonar sensors, will have a rule base of $3^3 = 27$ rules. The left-edge-following behavior, using two left side facing sonar sensors, will have a rule base of $3^2 = 9$ rules, the right edge following behavior will have the same number of rules. The goal seeking behavior, using a single goal detection sensor (more accurately represented by seven fuzzy sets) will have a rule base of seven rules. Thus the total number of rules in the low behaviors is $27+9+9+7 = 52$ rules and the total number of rules in the coordination layer is 4 (number of behaviors), thus we need a small number of rules which are much easier to be determined than *6561* rules in the case of the single rule base FLC.

There are many papers reporting implementations of type-1 HFLC which had produced good results [2], [5], [6], [20]–[22]. However, as the type-1 HFLC is composed from type-1 FLCs to implement the basic behaviors and the coordination between these behaviors, then the total coordinated system suffers from the uncertainty problems faced by type-1 fuzzy systems operating in changing unstructured environments.

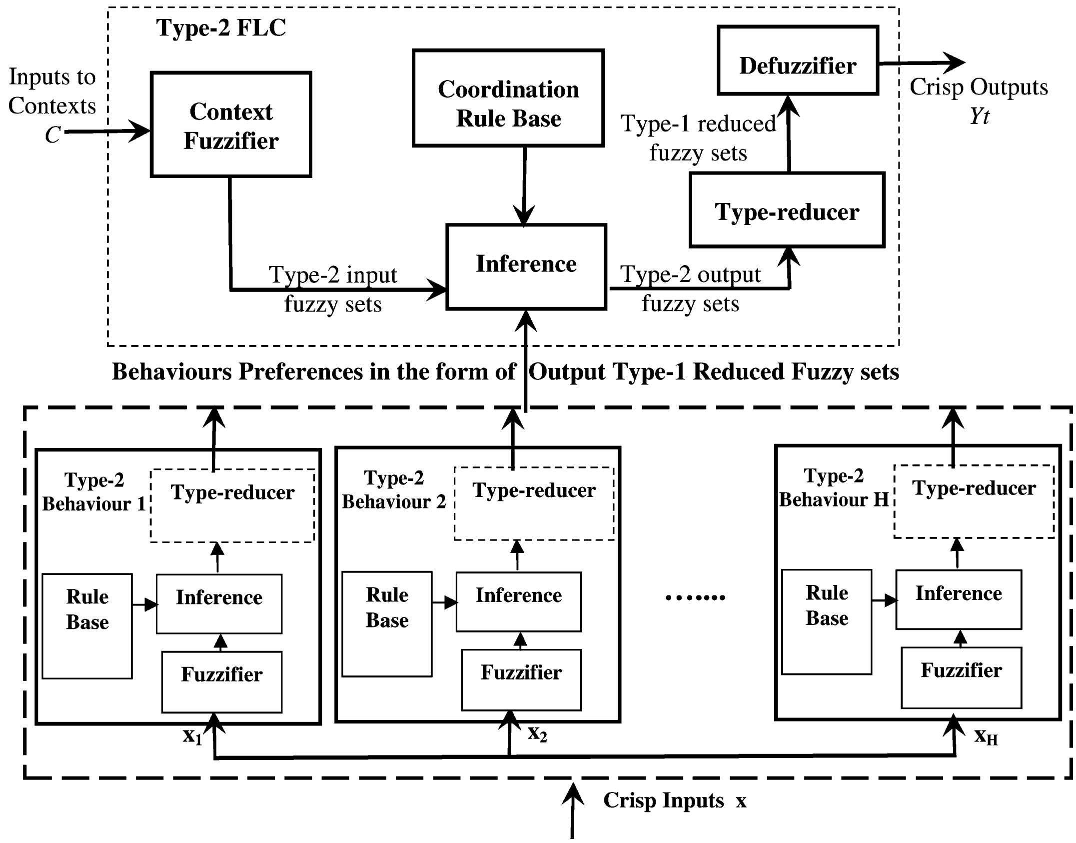


Fig. 4. Type-2 HFLC architecture.

B. Architecture of the Type-2 Hierarchical Fuzzy Logic Controller

1) Low Level Type-2 FLC-Based Behaviors: In our type-2 HFLC, each low level reactive behavior will be an interval type-2 FLC using interval type-2 fuzzy sets to represent the input and output variables of each behavior. Each low level behavior will receive a subset $\mathbf{x_h}$ of the total crisp inputs $\mathbf{x}$ available to the type-2 HFLC, all behaviors will produce *preferences* to the same common outputs, which are the outputs of the HFLC, so each behavior will map different inputs to common outputs. The low level type-2 FLCs will be the same as Section III except that there is no defuzzification block and the outputs from each low level behavior will be type-reduced sets to represent the *preferences* from the perspective of the behavior goals.

In our experiments, we will use four type-2 low level behaviors which are obstacle avoidance, goal seeking, left and right edge following. For the obstacle avoidance type-2 FLC, behavior using the three forward facing sonar sensors and representing each sensor by only two type-2 fuzzy sets will lead to a rule base of $2^3 = 8$ rules while for the left edge following behavior using two left side facing sonar sensors and representing each sensor by two type-2 fuzzy sets will lead to a rule base of $2^2 = 4$ rules, the right edge following behavior will be the same. For the goal seeking behavior, using a single location sensor which will be represented by three type-2 fuzzy sets will lead to a rule base of $3^1 = 3$ rules. Thus, the total number of rules required for the type-2 low level behaviors will be $8+4+4+3 = 19$ rules which is a rule reduction of about 64% from the type-1 hierarchical system which used 52 rules, while giving a better performance than the type-1 system as will be shown in the experiments section.

2) High Level Type-2 Coordination Layer: Fig. 4 shows the type-2 HFLC architecture, at the high level there is an interval type-2 FLC which is responsible for the coordination of the low level interval type-2 FLC-based behaviors. Each low level behavior has a *context* of activation $\widetilde{C}_j$ representing when it should be activated. The *context* is represented by an interval type-2 fuzzy set to handle the linguistic and numerical uncertainties associated with these *contexts*. Each behavior is activated with a strength given by the truth value of the *context*, i.e., the degree of firing of the interval type-2 membership function.

The high level type-2 FLC receives the crisp inputs to the *contexts* d_j $(j = 1, \dots, H)$, where H is the total number of behaviors. These crisp inputs are then fuzzified by matching each input to its *context* interval type-2 membership function. Again, we have chosen Singleton fuzzification. When the crisp inputs are fuzzified against the interval type-2 *context* membership functions we obtain for each crisp input d_j a lower membership value $\underline{\mu}_{\widetilde{C}_j}(d_j)$ and an upper membership value $\overline{\mu}_{\widetilde{C}_j}(d_j)$ for each context $\widetilde{C}_j$.

The high-level coordination type-2 FLC has a coordination rule base which contain coordination rules that describe in a fuzzy way when each behavior should be activated to influence the operation of the robot at each moment, the coordination rules take the following format:

$$\text{IF } d_j \text{ is } \widetilde{C}_j, \text{ THEN Behavior } \widetilde{B}_j, \qquad j = 1, \dots H. \tag{18}$$

Where d_j is the crisp input to the type-2 context $\widetilde{C}_j$ and H is the total number of behaviors. $\widetilde{B}_j$ is the behavior output type-2 fuzzy set. Note that we have a coordination rule for each behavior, thus we have a total of H coordination rules. From (18), we see that that each behavior output should be activated with a strength given by the truth value of its *context* $\widetilde{C}_j$.

In the inference engine of the high-level type-2 FLC, we use the *meet* under the product t-norm and the *join* using the maximum operation and the extended sup–star composition. We will use the center of sets type reduction which will allow for real time operation if the rule base is small, which is our case as the number of rules in the coordination rule base is equal to the number of behaviors H which is a small number. However to use the center of sets type-reduction we need to compute the centroid of the combined output type-2 fuzzy set for each low level behavior which must be pre-computed before the computation of the type-reduced sets of the HFLC. However if in each control cycle, we calculate the centroid of the combined output type-2 fuzzy set of each behavior, which is proportional to the number of discrete levels Z (which is in 100 in our case), this will take long time which might make the controller not computationally efficient for real time operation. An approximate solution for calculating the centroid of the output type-2 fuzzy set of each behavior is to calculate the center of sets type-reduced set for each behavior which gives very close results to calculating the centroid, as the center of sets type-reduction like any type-reducer combines all the rules output type-2 fuzzy sets and then performs a centroid calculation which leads to a type-1 fuzzy set [17]. The computation of the center of sets type-reduced set for each low level behavior is proportional to the number of rules in each behavior which is a small number. So, approximating the centroid of the output type-2 fuzzy set of each behavior by the center of sets type-reduced set will result in faster computation, which will result in a good real time performance.

At each control cycle, each low level type-2 FLC-based behavior will receive a subset $\mathbf{x_h}$ of the total crisp inputs $\mathbf{x}$ available to the type-2 HFLC and will generate a *preference* from the perspective of its goal which is represented by the type reduced interval set $[y_{lk}^j, y_{rk}^j]$ which approximates the centroid of the behavior output type-2 fuzzy set. Where $j = 1, \ldots, H$ and H is the total number of behaviors.

Note that the structure of the high level type-2 FLC will be the same as the type-2 FLC described in Section III, where it will use the context rule base and $\underline{\mu}_{\widetilde{C}_j}(d_j)$ and $\overline{\mu}_{\widetilde{C}_j}(d_j)$ will represent the left and right most points of the firing strength of the jth context rule. The consequent of each context rule is the behavior output type-2 fuzzy set $\widetilde{B}_j$ and its centroid is approximated by the type-reduced interval set $[y_{lk}^j, y_{rk}^j]$.

The type-reduced set for each HFLC output k *(*$k = 1, \ldots, c$*)* is defined by its left most point yt_{lk} and its right most point yt_{rk} which can be calculated according the following equations using the iterative procedure in Fig. 3:

$$yt_{lk} = \frac{\sum_{j=1}^{L} y_{lk}^j \overline{\mu}_{\widetilde{C}_j} + \sum_{j=L+1}^{H} y_{lk}^j \underline{\mu}_{\widetilde{C}_j}}{\sum_{j=1}^{L} \overline{\mu}_{\widetilde{C}_j} + \sum_{j=L+1}^{H} \underline{\mu}_{\widetilde{C}_j}} \tag{19}$$

$$yt_{rk} = \frac{\sum_{j=1}^{R} y_{rk}^j \underline{\mu}_{\widetilde{C}_j} + \sum_{j=R+1}^{H} y_{rk}^j \overline{\mu}_{\widetilde{C}_j}}{\sum_{j=1}^{R} \underline{\mu}_{\widetilde{C}_j} + \sum_{j=R+1}^{H} \overline{\mu}_{\widetilde{C}_j}}. \tag{20}$$

Note that the iterative procedure will converge in no more than H iterations to find yt_{lk} and no more than H iterations to find yt_{rk}. The final crisp defuzzified output Yt_k for the kth output can be found by finding the average of yt_{lk}, yt_{rk} as follows:

$$Yt_k = \frac{yt_{lk} + yt_{rk}}{2}. \tag{21}$$

V. Application of the Type-2 HFLC to Mobile Robot Control in Indoor and Outdoor Unstructured Environments

A. *Mobile Robots Descriptions*

In our experiments, we used indoor robots and outdoor robots. Both the indoor and the outdoor robots have a ring of seven ultrasonic sensors (with a covering cone of 30°), and a goal detection sensor which is an infrared scanner sensor for the indoor robots and GPS/compass for the outdoor robots. The indoor robots have two independent dc motors for driving plus differential steering. The outdoor robots have two dc motors one for controlling the speed of the front wheels and the other for controlling the steering of the front wheels. For our outdoor robots, the steering values are in percentage where positive steering values means steering to the right while negative steering values means steering to the left. Both the indoor and outdoor robots hardware is based on embedded Motorola 68040 processors running VxWorks Real-Time Operating System. The control programs are developed under the Tornado environment and then downloaded via an Ethernet cable or a wireless RF modem to the robots. After the program downloading the cable or the RF link can be disconnected and the navigation becomes autonomous.

Both the indoor and outdoor mobile robots will have four basic low level type-2 FLC-based behaviors which are left/right edge following, obstacle avoidance and goal seeking. The total number of available crisp inputs to the interval type-2 HFLC is eight inputs which are seven sonar sensors plus a goal detection sensor. The HFLC and all the behaviors have two common outputs (left and right wheel speeds for the indoor robots and the front wheels speed and steering for the outdoor robots). In the next section, we will describe the low level type-2 FLC-based behaviors and the high level type-2 FLC coordination layer.

B. *The Low Level Behaviors*

1) Right and Left Edge Following Type-2 Behaviors: Edge following behaviors are used to follow edges at a desired distance. We will use two edge following behaviors which are left edge following and right edge following behaviors.

The right edge following behavior type-2 FLC in both the indoor and outdoor robots receives only two crisp inputs from the two right side sonar sensors defined as right side front (RSF) and

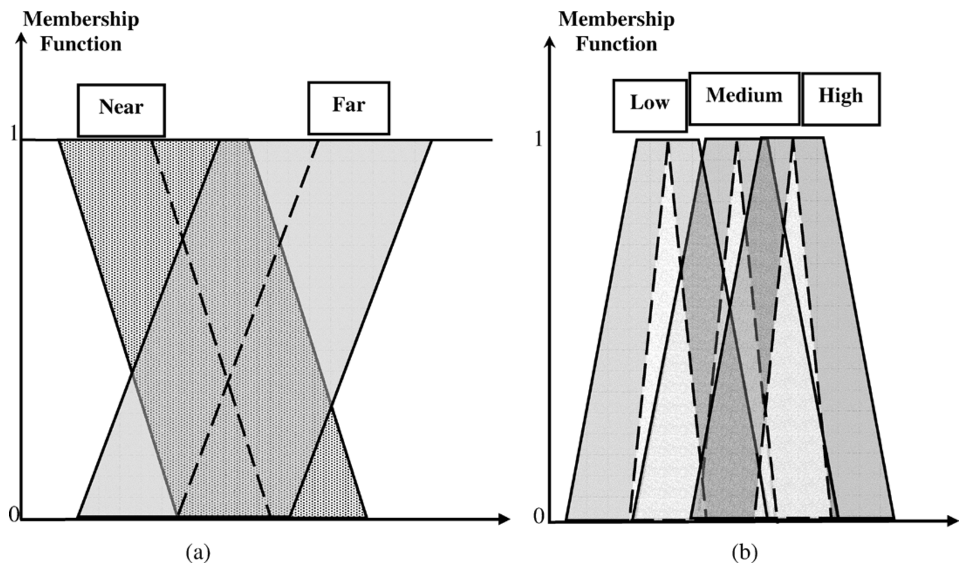


Fig. 5. (a) Input interval type-2 membership function shapes for the input sensors for the edge following behaviors and the obstacle avoidance behavior in indoor and outdoor robots. (b) Output interval type-2 membership function for the velocities in the indoor robots.

right side back (RSB). The left edge following behavior type-2 FLC in both the indoor and outdoor robots receives only two crisp inputs from the two left side sensors defined as left side front (LSF) and left side back (LSB). For both the left edge following and the right edge following behaviors in indoor and outdoor robots, we represented each input by only two interval type-2 fuzzy sets which are *Near* and *Far* as shown in Fig. 5(a). The type-2 fuzzy sets have a footprint of uncertainty which is shaded in grey in Fig. 5(a) to model and handle the uncertainties and to deal with the difficulty associated with determining the exact membership functions [13]. The membership function in thick dashed line shown in Fig. 5(a) represents a type-1 fuzzy set that might be used by the robot to follow the edge at the desired distance.

In interval type-2 membership functions, the footprint of uncertainty is obtained by specifying the bounding upper and lower type-1 membership functions [13], [14]. We have determined the lower and upper type-1 membership functions for the type-2 fuzzy sets *Near* and *Far* by finding the lowest and highest type-1 membership functions that the robot needed to follow the edge at the desired distance in different environmental and robot conditions, this was aided by using our online learning method developed in our previous work to learn the type-1 membership functions [5].

For the indoor robots, the edge following behaviors like all the other behaviors will have two outputs *preferences*, one for the left wheel velocity and the other for the right wheel velocity. Each output velocity will be represented by three interval type-2 fuzzy sets which are *Low*, *Medium*, and *High* as shown in Fig. 5(b).

For the indoor and outdoor robots as we have used two inputs for each edge following behavior and we represented each input by two type-2 fuzzy sets therefore we have a rule base of $2^2 = 4$ rules. Although we have used a small number of fuzzy sets to represent the inputs and outputs and thus a small rule base, the type-2 FLC has the potential to outperform a type-1 FLC using a larger rule base. This can be shown in the control surface which graphically represents the unknown function articulated by the FLC. Fig. 6(a) shows the control surface for the type-2 FLC steering output of the outdoor robot implementing the right edge following behavior using only two type-2 fuzzy sets to represent each input and thus a rule base of four rules. This control surface represents the experiments shown in Fig. 10(b) in which the outdoor robot will follow an irregular edge in an outdoor changing and dynamic environment. Note the smooth shape of the control surface which translates to a smooth control response that can deal with uncertainty and imprecision.

Fig. 6(b) shows the control surface for a type-1 FLC steering output for the right edge following behavior using also two type-1 fuzzy sets to represent each input and thus having a rule base of four rules. Fig. 6(c) shows the control surface for a type-1 FLC using three type-1 fuzzy sets to represent each input and thus having a rule base of eight rules. Fig. 6(d) shows the control surface for a type-1 FLC using five type-1 fuzzy sets to represent each input and thus having a rule base of 25 rules. Note that the more type-1 fuzzy sets used in the type-1 FLC the more its response approaches the smooth response of the type-2 FLC. This is because the type-2 fuzzy sets contain a large number of embedded type-1 fuzzy sets which allow for the detailed description of the analytical control surface as the addition of the extra levels of classification gives a much smoother control surface and response, this will be further demonstrated in our experiments. The same results were obtained for all the other behaviors in indoor and outdoor robots for both of the controller outputs.

2) Obstacle Avoidance Type-2 Behavior: The obstacle avoidance behavior is needed for safe navigation in unstructured environments and is needed to avoid static and dynamic objects at a safe avoiding distance. In this behavior, the type-2 FLC in both the indoor and outdoor robots receives three crisp inputs from the three front sonar sensors, defined as left front sonar sensor (LFS), middle front sonar sensor (MFS), and right front sonar sensor (RFS). We represent each sensor by only two interval type-2 fuzzy sets which are *Near* and *Far* as shown in Fig. 5(a) (of course, the base values are different from

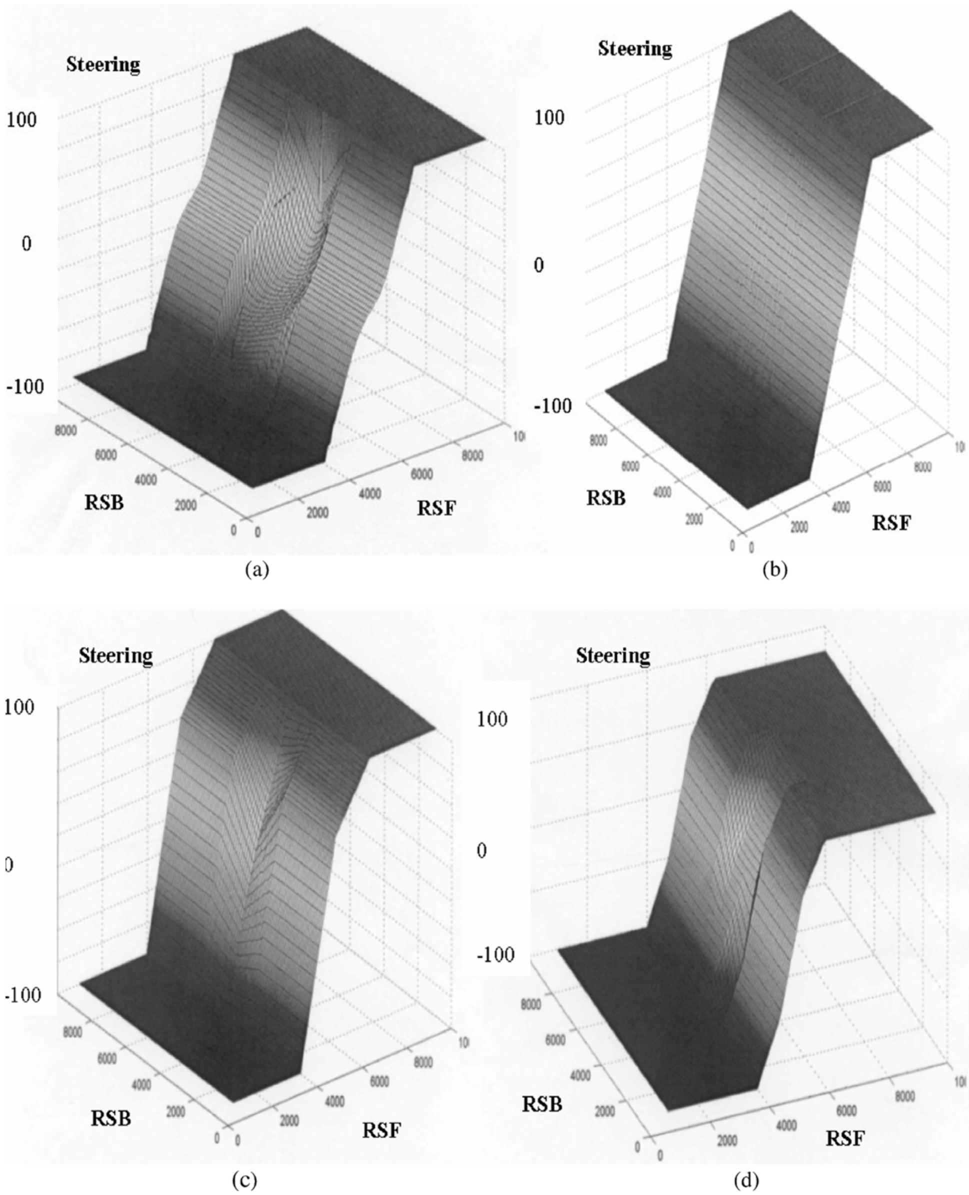


Fig. 6. (a) Control surface of a type-2 FLC for the right edge following behavior. (b) Control surface of a type-1 FLC using two fuzzy sets to represent each input. (c) Control surface of a type-1 FLC using three fuzzy sets to represent each input. (d) Control surface of a type-1 FLC using five fuzzy sets to represent each input.

the edge following behaviors which have the same shape). For the indoor and outdoor robots, we have used three inputs for the obstacle avoidance behavior and we represented each input by two fuzzy sets. Therefore, we have a rule base of $2^3 = 8$ rules.

3) Goal Seeking Type-2 Behavior: The goal seeking behavior is needed to reach a specific goal in unstructured environment. In both the indoor and outdoor robots the input to the goal seeking type-2 FLC takes the form of a bearing from the goal. In indoor environments, this goal can be in the form of an infrared beacon and the bearing from it is measured by an infrared scanner, while in outdoor environments the bearing from the goal can be measured by a GPS or a compass. The robot is supposed to achieve zero deviation from its target and align completely with its target to reach it. In both the indoor and outdoor robots the bearing input to the goal seeking type-2 FLC is represented by three interval type-2 fuzzy sets which are *Negative, Zero, and Positive* as shown in Fig. 7(a). For the indoor and outdoor robots as we have used one input to the goal seeking type-2 FLC and we represented each input by three fuzzy sets therefore we have a rule base of $3^1 = 3$ rules.

C. High-Level Type-2 FLC Coordination Layer

For indoor and outdoor robots the total number of outputs $c = 2$ which are the left and right wheel velocities for indoor robots and the speed and steering of the front wheels for the outdoor robots. As mentioned above the high-level type-2 FLC receives the crisp inputs to the *contexts* d_j ($j = 1, \ldots, H$), $H = 4$. Each type-2 *context* $\widetilde{C}_j$ is attached to the jth behavior defining when the behavior should be activated.

The crisp input d_1 to the obstacle avoidance behavior context $\widetilde{C}_1$ in both the indoor and outdoor robots is the minimum value

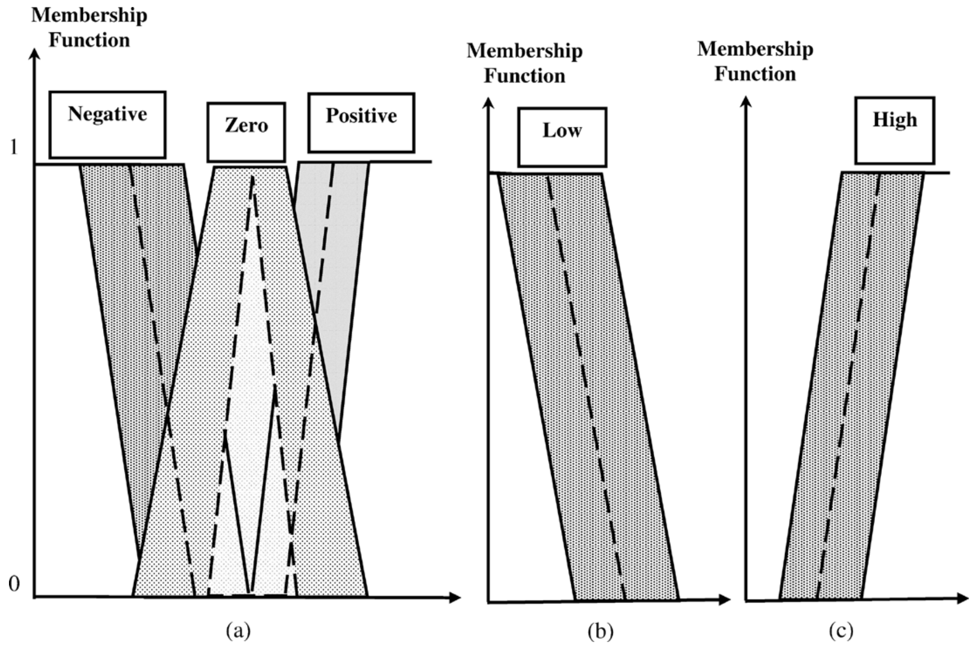


Fig. 7. (a) Interval type-2 input membership function for the goal seeking behavior in the indoor and outdoor robots. (b) Context interval type-2 fuzzy set shape for the obstacle avoidance behavior and the right and left edge following behaviors. (c) Context interval type-2 fuzzy set for the goal seeking behavior.

of the front sonar sensors. The context $\widetilde{C}_1$ for the obstacle avoidance behavior is in the form of an interval type-2 fuzzy set as shown in Fig. 7(b) which defines that the obstacle avoidance behavior should be active when the robot path is obstructed by an obstacle and the closer the robot gets to the obstacle, the higher will be the activation of the obstacle avoidance behavior. d_1 was chosen to be the minimum distance of the front sensors so that the obstacle avoidance behavior is activated according to the nearest obstacle. d_1 is fuzzified using the interval type-2 *context* membership functions of $\widetilde{C}_1$ to a lower membership $\underline{\mu}_{\widetilde{C}_1}(d_1)$ and an upper membership $\overline{\mu}_{\widetilde{C}_1}(d_1)$.

The crisp input d_2 to the left edge following behavior context $\widetilde{C}_2$ in both the indoor and outdoor robots is the minimum value of the left side sonar sensors. The context $\widetilde{C}_2$ for the left edge following behavior is in the form of an interval type-2 fuzzy set as shown in Fig. 7(b) which defines that the left edge following behavior should be active when an edge is detected to the left side of the robot, the closer the robot gets to the edge on its left, the higher will be the activation of the left edge following behavior. d_2 is fuzzified using the interval type-2 *context* membership functions of $\widetilde{C}_2$ to a lower membership $\underline{\mu}_{\widetilde{C}_2}(d_2)$ and an upper membership $\overline{\mu}_{\widetilde{C}_2}(d_2)$.

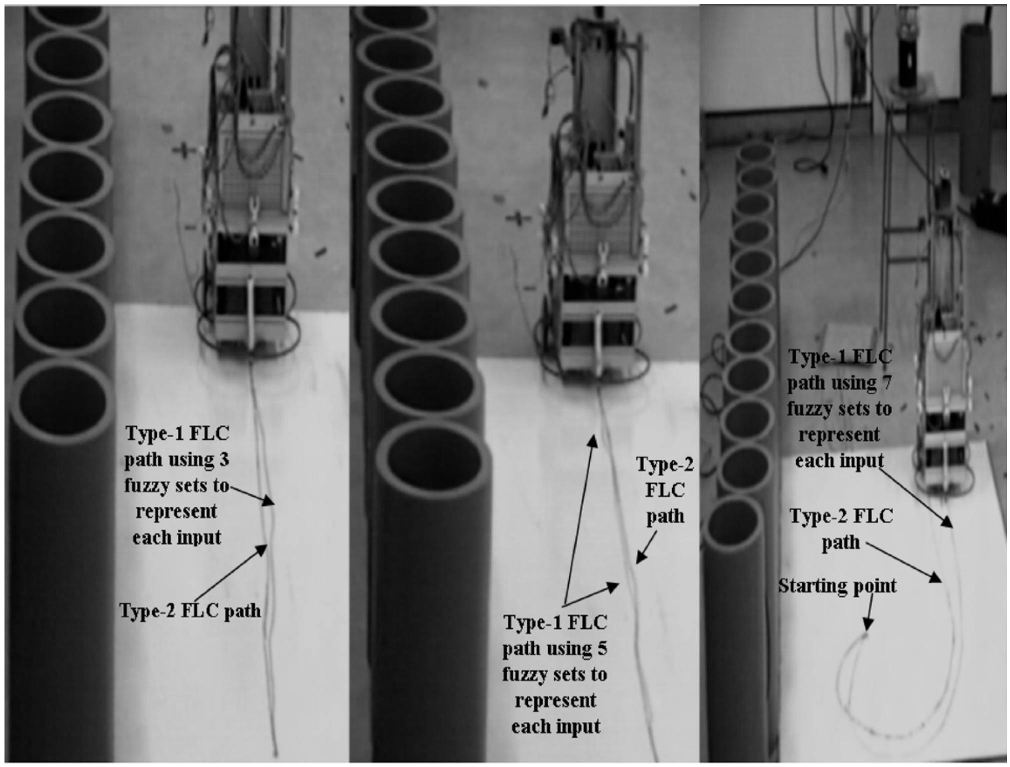
The crisp input d_3 to the right edge following behavior context $\widetilde{C}_3$ in both the indoor and outdoor robots is the minimum value of the right side sonar sensors. d_3 is fuzzified using the interval type-2 *context* membership functions of $\widetilde{C}_3$ shown in Fig. 7(b) to a lower membership $\underline{\mu}_{\widetilde{C}_3}(d_3)$ and an upper membership $\overline{\mu}_{\widetilde{C}_3}(d_3)$. Note that although the shapes of $\widetilde{C}_1$, $\widetilde{C}_2$, $\widetilde{C}_3$ look the same, the base values are different.

The crisp input d_4 to the goal seeking behavior context $\widetilde{C}_4$ in both the indoor and outdoor robots is the minimum value of the of d_1, d_2, *and* d_3. The context $\widetilde{C}_4$ for the goal seeking behavior is in the form of an interval type-2 fuzzy set as shown in Fig. 7(c) which defines that the goal seeking behavior should be active when the robot path is clear from the sides and the front, the clearer the robot path the higher will be the activation of the goal seeking behavior. d_4 is fuzzified using the interval type-2 *context* membership functions of $\widetilde{C}_4$ to a lower membership $\underline{\mu}_{\widetilde{C}_4}(d_4)$ and an upper membership $\overline{\mu}_{\widetilde{C}_4}(d_4)$.

After all the crisp inputs d_j *(*$j = 1, \ldots 4$*) are* matched and fuzzified against their type-2 fuzzy contexts, the fuzzified values are then fed to the inference engine which determines which rules (and hence which behaviors) are fired from the coordination rule base.

The coordination rule base contains a coordination rule for each behavior which relates the *contexts* to the behaviors. As we have four behaviors, then we will have four coordination rules as follows: *IF* d_1 *is LOW THEN Obstacle Avoidance, IF* d_2 *is LOW, THEN Left Edge Following, IF* d_3 *is LOW, THEN Right Edge Following, IF* d_4 *is HIGH, THEN Goal Seeking*. The system is capable of performing very different tasks using identical behaviors by changing only the context rules and co-ordination parameters. We can eliminate the unneeded behaviors from the context rules according to the robot's mission.

$\underline{\mu}_{\widetilde{C}_j}(d_j)$ and $\overline{\mu}_{\widetilde{C}_j}(d_j)$ represent the left and right most points of the firing strength of the jth context rule. For each output $k = 1$ *, 2* the type-reduction block receives the output type-reduced sets from the four low level behaviors $[y_{lk}^j, y_{rk}^j]$, where $j = 1, \ldots, 4$. These type-reduced sets approximate the centroids of the behaviors output type-2 fuzzy sets and they represent the *preferences* from the perspective of the goals of the low level behaviors. To obtain the type-reduced set for each kth HFLC output we apply the iterative procedure in Fig. 3 and we substitute in (19) and (20) to find the left most point yt_{lk} and the right most point yt_{rk}. For each kth HFLC output, to obtain the defuzzified crisp output to be sent the robot actuators at the end of each control cycle we find the average of the type-reduced set by substituting in (21) to obtain the final crisp output Yt_k.



(a) (b) (c)

Fig. 8. (a) Indoor robot path using type-2 FLC to implement the left edge following behavior compared against type-1 FLC path using three fuzzy sets to represent each input. (b) Type-2 FLC path compared against type-1 FLC using five fuzzy sets to represent each input. (c) Robot path using type-2 FLC to implement the goal seeking behaviors compared against a type-1 FLC using seven fuzzy sets.

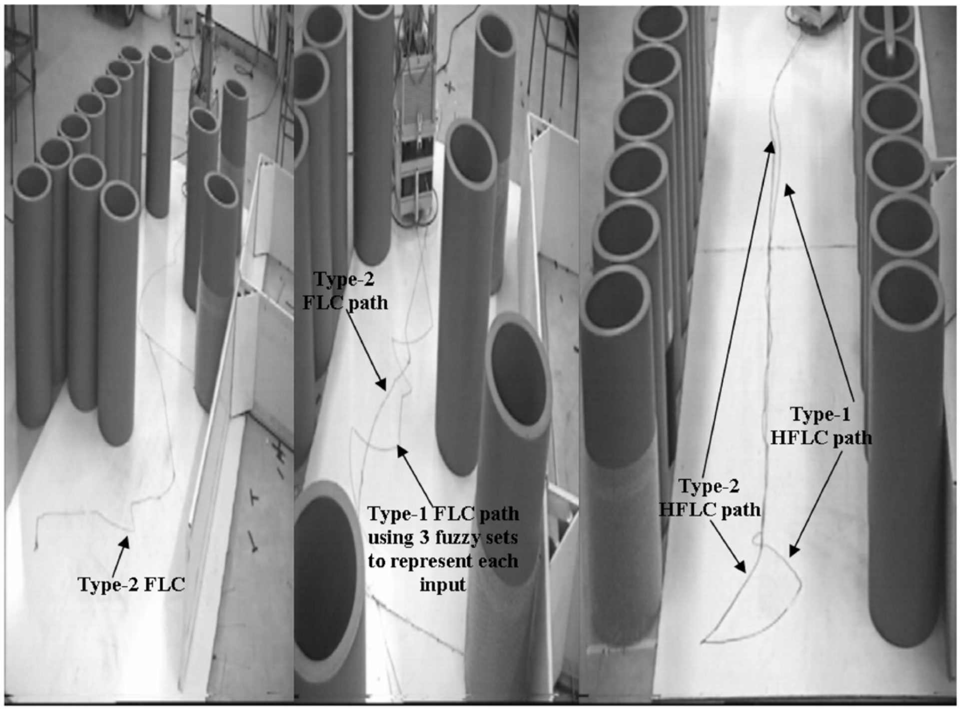
Note that the robots can achieve multiple goals, whose priorities may change with time as the behavior *preferences* outputs and the *context* truth values (weights) are dynamic taking into account the situation of the robot.

The computation time for this hierarchical structure is small as for the high level type-2 FLC to compute the type reduced set for each HFLC output we will need at most four iterations (number of coordination rules) to find yt_{lk} and maximum 4 iterations to find yt_{rk}, thus we need maximum eight iterations to compute the high level type-reduced set for each HFLC output. Also, we need to perform the type-reduction for the low level behaviors that are performed first and then fed to the high level type-2 FLC, however these calculations will be performed in parallel, and as the biggest rule base which is the obstacle avoidance behavior contains eight rules, therefore, we will need maximum 16 iterations to compute the type-reduced sets for the low level behaviors for each output. Therefore, the maximum number of iterations to compute a type-reduced set and thus a HFLC crisp output at the end of the control cycle is $8 + 16 = 24$ iterations for each HFLC output (in both the indoor and outdoor robots we have only 2 HFLC outputs). This small number of iterations will enable real time operation even with our robots using slow processors, note that the 24 iterations is a small number compared to 13122 iterations to find a type-reduced set for each output in case of a single rule base type-2 FLC. In Section VI, we will introduce many experiments using real indoor and outdoor robots operating in unstructured environments which will enable us to evaluate the real time performance of the type-2 FLC and HFLC.

VI. Experiments and Results

In the first part of our experiments, we used indoor robots navigating in indoor unstructured environments. We used noisy sensors and different irregular geometrical structures which present a real challenge to the ultrasound sensors (multiple reflections, sonar diffuse reflection, etc). The robot path was drawn using a pen fixed to the back of the robot to record actual paths. In all the following experiments, all the average and standard deviations from the desired values were calculated over eight experiments, where we used different geometrical structures and started the robot from different random locations. We will start by experiments aimed at testing the individual type-2 FLC-based behaviors where the robot will have one objective related to the behavior goal and the outputs from the type-reduction will be defuzzified and fed to the actuators.

Fig. 8(a) shows the indoor robot implementing the type-2 FLC-based left edge following behavior to follow an irregular edge which offers poor ultrasound reflections. The robot is required to follow the edge at a desired distance of 35 cm. This type-2 FLC only used two type-2 fuzzy sets to represent each input and thus it has a rule base of only four rules. The type-2 FLC had succeeded in following the irregular edge with an average deviation from the desired distance of 1.6 cm and a standard deviation of 0.9. Fig. 8(a) shows the type-2 FLC response which is a smooth response and can deal with the impression and uncertainty available in this real world indoor unstructured environment.



(a) (b) (c)

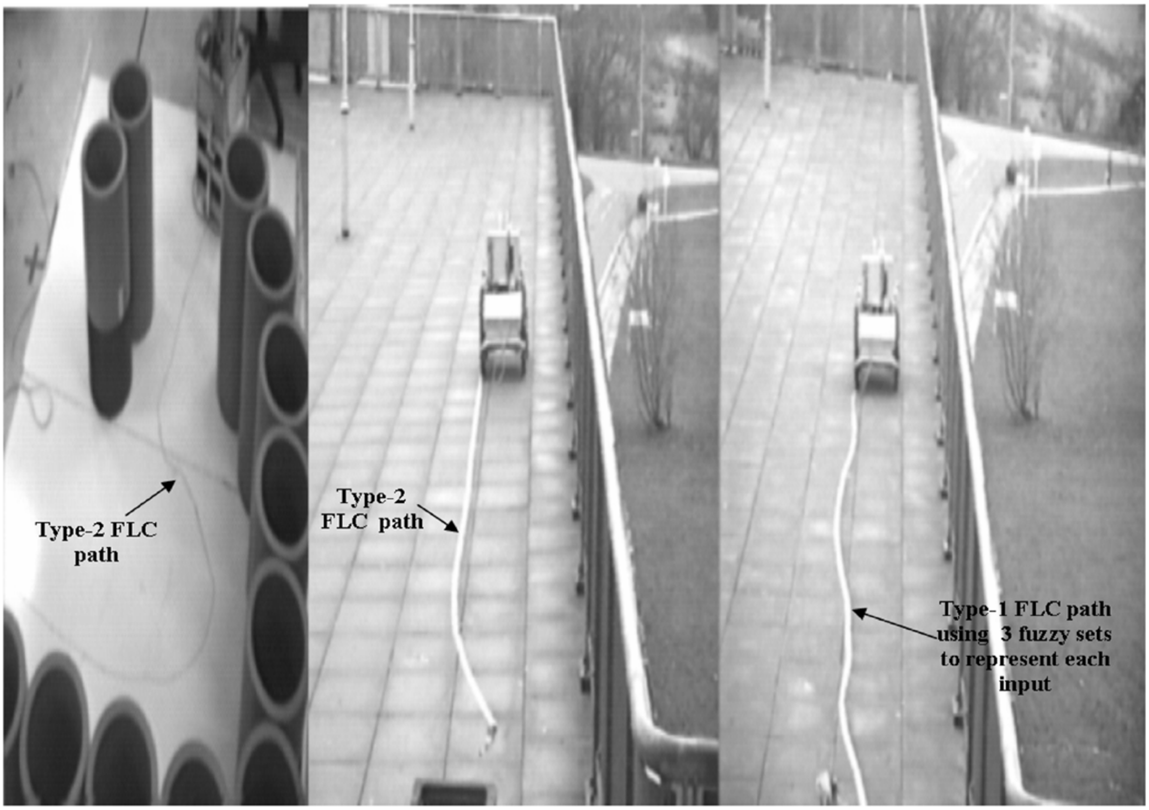
Fig. 9. (a) Indoor robot path using type-2 FLC to implement the obstacle avoidance behavior. (b) Type-2 FLC path compared against type-1 FLC path using three fuzzy sets to represent each input. (c) Indoor robot path using type-2 HFLC to follow a corridor and reaching a goal at the end the corridor.

We have compared the type-2 FLC response with type-1 FLCs using bigger number of fuzzy sets (thus having larger rule bases). All the type-1 FLC parameters were learned online using our online fuzzy genetic system [5], [6] which learns the best type-1 FLC under certain environment and robot conditions. Fig. 8(a) shows a type-1 FLC which uses three type-1 fuzzy sets to represent each input thus it has rule base of nine ules, the type-1 FLC had produced a path of average deviation from the desired distance of 3.1 cm and a standard deviation of 1.4. In Fig. 8(b), we experimented with another type-1 FLC using five type-1 fuzzy sets to represent each input and thus it has a rule base of 25 rules, the type-1 FLC had produced a path of average deviation of 2 cm from the desired distance and a standard deviation of 1.2. From Fig. 8(a)(b) it is obvious that the type-2 FLC had outperformed the performance of the type-1 FLC while using a smaller number of fuzzy sets to describe each input (thus having a smaller rule base). Also, note that as the number of type-1 fuzzy sets in type-1 FLC increases the type-1 FLC performance approaches the type-2 FLC performance which agrees with the control surface results in Fig. 6.

Fig. 8(c) shows the indoor robot implementing the type-2 FLC-based goal seeking behavior, where the goal is represented by an infrared beacon and the main objective of this behavior is to reach the goal and align with it with zero deviation. The type-2 FLC for the goal seeking behavior has only one input represented by three type-2 fuzzy sets, thus the type-2 FLC has a rule base of only three rules. Fig. 8(c) shows the path of the type-2 FLC where the goal is placed behind the robot and the robot turns and reaches the goal, the type-2 FLC had produced a smooth path with an average deviation from the desired zero degree alignment of 3° and a standard deviation of 0.7. We have tried type-1 FLCs using more number of fuzzy sets to represent each input (thus having larger rule bases) as shown in Fig. 8(c). The type-2 FLC had outperformed the performance of the type-1 FLC while using a smaller number fuzzy sets to describe each input (thus, having a smaller rule base). In addition, as the number of type-1 fuzzy sets in type-1 FLC increases the type-1 performance approaches the type-2 performance.

Fig. 9(a) shows the robot using a type-2 FLC to perform the obstacle avoidance behavior, the robot goal is to maintain a minimum safe distance of 45 cm from any front obstacle. The type-2 FLC for the obstacle behavior has three inputs; each is represented by two type-2 fuzzy sets thus the FLC has a rule base of only 8 rules. Fig. 9(a) shows the path of the type-2 FLC where the robot navigates in a tight environment, the type-2 FLC had produced a smooth path with an average deviation from the desired safe distance of 2.4 cm and a standard deviation of 1.1. We have tried type-1 FLCs using more number of fuzzy sets to represent each input (thus having larger rule bases). Fig. 9(b) shows a type-1 FLC using three type-1 fuzzy sets to represent each input and thus having a rule base of 27 rules, this type-1 FLC had given an average deviation of 5.3 cm and a standard deviation of 3.2 from the desired safe distance.

In the previous experiments, we have introduced the basic type-2 FLC-based behaviors and how they dealt in real time with the uncertainties and outperformed the type-1 FLC-based behaviors, while the type-2 FLC-based behaviors have used a smaller number of fuzzy sets to represent each input and thus they used smaller rule bases. In the following experiments, we will present the results obtained from the type-2 HFLC coordinating more than one behavior to satisfy more than one objective.



(a) (b) (c)

Fig. 10. (a) Path of the type-2 HFLC to follow an edge while avoiding obstacles and reaching a goal at the end of the edge. (b) Outdoor robot path using type-2 FLC to implement the right edge following behavior to follow an irregular edge. (c) Robot path using a type-1 FLC when environment changed (windy weather).

Fig. 9(c) shows the robot coordinating all its behaviors to follow a corridor and reaching a goal at the end of the corridor. Although none of the *context rules* dealt with corridor following but because of the balanced *contexts* of the left and right edge following behaviors the corridor following behavior had emerged. We have performed eight experiments using different sizes of corridors and starting the robot from different random positions. The robot had followed the centre line of the corridor with an average deviation of 1.1 cm and a standard deviation of 0.7 and it always reached and aligned with the goal at the end of the corridor. The type-2 HFLC had outperformed the type-1 HFLC as shown in Fig. 9(c) which produced an average deviation of 2.4 cm and a standard deviation of 1.2 from the centre line of the corridor. The type-1 HFLC was used in our previous work [5], [6] and had a total number of 52 rules in the low level behaviors as mentioned in Section IV. The type-2 HFLC had only a total number of $8 + 4 + 4 + 3 = 19$ rules in the low level behaviors, so the type-2 HFLC produced a rule reduction of about 64% and outperformed the type-1 HFLC.

Fig. 10(a) shows the robot coordinating three type-2 behaviors which are right edge following, obstacle avoidance and goal seeking to follow an edge while avoiding obstacles and reaching a goal at the end of the edge. As we do not need the left edge following behavior, it was removed with its *context* and *context rules* from the HFLC. Again, the robot had followed the edge with a small average and standard deviation from the desired distance while avoiding any obstacles at the desired safe distance and aligning with the goal and reaching it with minimum deviation.

In all the previous experiments, the type-2 FLC and HFLC had dealt in real time with the uncertainties available in the unstructured indoor environments. The type-2 FLC and HFLC responses were repeatable when started from different starting positions or when tested with changing the geometrical settings and they dealt in real time with dynamic changes in the environment and the ground surface.

Although we did what we can to introduce as much noise, imprecision and dynamic changes to the indoor environments to evaluate the real time performance of the type-2 FLC and HFLC, there are clearly big differences between the indoor environments and the outdoor changing and dynamic unstructured environments. The outdoor unstructured environments will be a severe test to evaluate the real time performance of the robot type-2 FLC and HFLC and how they handle the large amounts of uncertainty and imprecision facing the mobile robots in such changing and dynamic environments.

We had performed many experiments using our outdoor robots in outdoor changing and dynamic environments. The robot path was recorded using a tape fixed under the left back wheel and we calculated the average and standard deviations from the desired values. In all the following experiments, all the average and standard deviations from the desired values were calculated over eight experiments where the robot was started from different random locations and under different environmental conditions like rain, wind, sunshine, etc., and different ground conditions such as slippery, dry grounds, and even different times of the day. We also tried different chal-

(a) (b) (c)

Fig. 11. (a) Outdoor robot path using the type-2 HFLC to follow an edge while avoiding obstacles. (b) Same experiment as in Fig. 11(a) carried during night using different obstacles configurations. (c) Outdoor robot path using the HFLC to follow an irregular corridor while avoiding obstacles in a slippery ground.

lenging environmental features like metallic and plant edges which offer bad sonar response.

Fig. 10(b) shows the outdoor robot implementing the type-2-based right edge following behavior to follow an irregular metallic edge at a desired distance of 1.2 m. The type-2 FLC used two type-2 fuzzy sets to represent each input and thus it has four rules in the rule base. The robot type-2 FLC had followed the irregular edge with an average deviation of 2.1 cm and a standard deviation of 0.8. The type-1 FLC can give a good response under specific weather, ground and robot conditions but if any of these conditions change like operating in a windy weather condition then the type-1 FLC with nine rules shown in Fig. 10(c) will fail and give a bad response as it is using crisp and precise membership functions that cannot handle the uncertainties associated with changing unstructured outdoor environments, while the type-2 FLC shown in Fig. 10(b) can handle such uncertainties to give a better response while using a smaller rule base.

We have tried also many experiments to test the type-2 HFLC in outdoor environments, Fig. 11(a) shows the robot coordinating the right edge following behavior and the obstacle avoidance behavior to follow an irregular edge at a desired distance of 1.2 m and to avoid obstacles at a safe distance of 1.4 m. We have performed many experiments starting the robot from different random positions using different obstacles configuration under different weather conditions and during different times of the day as shown in Fig. 11(b) where we performed some experiments during the night. The type-2 HFLC had always given a very good response as shown in Fig. 11(a) and (b) where the robot had followed the edge with a smooth response and with a small average and standard deviations while avoiding obstacles at the desired safe distance.

Fig. 11(c) shows the outdoor robot coordinating its obstacle avoidance, left and right edge following behaviors to follow an irregular composite corridor consisting of an irregular metallic corridor and a plant hedge corridor full of gaps, the robot should also avoid obstacles at the desired safe distance on a slippery ground. The robot had followed the center line of the corridor with an average deviation of 1.9 cm and standard deviation of 0.8 while avoiding obstacles at the desired safe distance.

In all the previous experiments, the type-2 FLC and HFLC had dealt in real time with the uncertainties available in the changing and dynamic outdoor unstructured environments. The type-2 FLC and HFLC responses were repeatable when started from different starting positions and under different environmental conditions (such as wind, rain, sunshine, .etc) and under different ground conditions (such as slippery and dry grounds) and during different times of the day. The type-2 FLC and HFLC had always outperformed the response of the type-1 FLC and HFLC while using less number of rules.

We had performed preliminary empirical analysis to compare the computational tradeoffs between the type-2 FLC and HFLC and their type-1 counterparts. It was found that a type-2 FLC compared to a type-1 FLC with the same rule base involves more computation mainly due to the type-reduction calculation. The type-reduction calculation adds a small computation overhead which is hardly observable (even though we use slow processors on our robots) when the type-2 FLC rule base is small (which is our case). As all our type-2 FLCs were using much smaller rule bases than their type-1 counterparts, we found empirically that the type-2 FLCs provided a faster computation for outputs from given inputs besides giving a better performance. This became more evident in the type-2 HFLC which gave a much faster computation than the type-1 HFLC as it has 64% less rules. We hope to provide more detailed analysis in our future work.

VII. Conclusion

In this paper, we presented a novel reactive type-2 fuzzy architecture for the real time control of mobile robots navigating in changing and dynamic unstructured indoor and outdoor environments. This architecture was based on using interval type-2 FLC to implement the basic navigation behaviors and the coordination between these behaviors to produce a type-2 HFLC. We have shown how the type-2 hierarchical fuzzy system simplifies the design of the robotic controller and reduces the number of rules to be determined so we can have a real time operation of the type-2 robot controller. To the author's knowledge, this is the first paper applying type-2 fuzzy systems to real time robot control and to control in general.

We have presented numerous experiments using different mobile robots navigating in challenging indoor and outdoor unstructured environments. We have shown how the type-2 FLCs and HFLCs can deal in real-time with the uncertainties facing mobile robots in changing and dynamic unstructured environments and that they resulted in very good real time control responses that had outperformed the type-1 FLCs and HLFCs while resulting in a significant rule reduction which was about 64% in case of the HFLCs.

We are currently working on producing type-2 FLCs for the control of marine diesel engines. In addition, we have another project that aims to produce intelligent embedded agents that can learn type-2 FLCs to control intelligent buildings and to create ambient intelligence in ubiquitous computing environments.

References

[1] R. Arkin, *Behavior-Based Robotics*. Cambridge, MA: MIT Press, 1998.
[2] A. Bonarini, "Anytime learning and adaptation of hierarchical fuzzy logic behaviors," *Adapt. Behavior J.*, vol. 5, no. 3–4, pp. 281–315, 1997.
[3] A. Cerezo, A. Mandow, and M. Lopez, "Fuzzy modeling of operator navigation behaviors," in *Proc. 6th IEEE Int. Conf. Fuzzy Systems*, Barcelona, Spain, 1997, pp. 1339–1345.
[4] T. Fukuda and N. Kubota, "An intelligent robotic system based on fuzzy approach," *Proc. IEEE*, vol. 87, pp. 1448–1470, Aug. 1999.
[5] H. Hagras, V. Callaghan, and M. Colley, "Prototyping design and learning in outdoor mobile robots operating in unstructured outdoor environments," *IEEE Robot. Automat. Mag.*, vol. 8, no. 3, pp. 53–69, Sept. 2001.
[6] H. Hagras, M. Colley, and V. Callaghan, "Learning and adaptation of an intelligent mobile robot navigator operating in unstructured environments based on a novel online fuzzy-genetic system," *J. Fuzzy Sets Syst.*, vol. 141, no. 1, pp. 107–160, Jan. 2004.
[7] F. Hoffmann and G. Pfister, "Evolutionary design of a fuzzy knowledge base for a mobile robot," *Int. J. Approx. Reason.*, vol. 17, no. 4, pp. 447–469, 1997.
[8] R. John, "Type 2 fuzzy sets: An appraisal of theory and applications," *Int. J. Uncertainty, Fuzziness, Knowledge Based Syst.*, vol. 6, no. 6, pp. 563–576, 1998.
[9] A. Kandel and G. Langholz, *Fuzzy Control Systems*. Boca Raton, FL: CRC Press, 1994.
[10] N. Karnik and J. Mendel. (1998) An introduction to type-2 fuzzy logic systems. Univ. Southern California, Los Angeles, CA. [Online]. Available: http://sipi.usc.edu~mendel/report
[11] N. Karnik, J. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 7, pp. 643–658, Dec. 1999.
[12] C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller, Part II," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, pp. 419–434, Mar. 1990.
[13] Q. Liang, N. Karnik, and J. Mendel, "Connection admission control in ATM networks using survey-based type-2 fuzzy logic systems," *IEEE Trans. Syst., Man, Cybern. C*, vol. 30, pp. 329–339, Aug. 2000.
[14] Q. Liang and J. Mendel, "Interval type-2 fuzzy logic systems: Theory and design," *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 535–550, Oct. 2000.
[15] M. Maeda, Y. Maeda, and S. Murakami, "Fuzzy drive control of an autonomous mobile robot," *J. Fuzzy Sets Syst.*, vol. 39, pp. 195–204, 1991.
[16] V. Matellan, C. Fernandez, and J. Molina, "Genetic learning for fuzzy reactive controllers," *J. Robot. Auton. Syst.*, vol. 25, pp. 33–41, 1998.
[17] J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper Saddle River, NJ: Prentice-Hall, 2001.
[18] J. Mendel and R. John, "Type-2 fuzzy sets made simple," *IEEE Trans. Fuzzy Syst.*, vol. 10, pp. 117–127, Apr. 2002.
[19] J. Mendel and H. Wu, "Uncertainty versus choice in rule-based fuzzy logic systems," in *Proc. IEEE Int. Conf. Fuzzy Systems*, Honolulu, HI, 2002, pp. 1336–1342.
[20] A. Saffiotti, "The uses of fuzzy logic in autonomous robot navigation," *J. Soft Comput.*, vol. 1, no. 4, pp. 180–197, 1997.
[21] ——, "Fuzzy logic in autonomous robotics: Behavior coordination," in *Proc. 6th IEEE Int. Conf. Fuzzy Systems*, Barcelona, Spain, 1997, pp. 573–578.
[22] E. Tunstel, T. Lippincott, and M. Jamshidi, "Behavior hierarchy for autonomous mobile robots: Fuzzy behavior modulation and evolution," *Int. J. Intell. Automat. Soft Comput.*, vol. 3, no. 1, pp. 37–49, 1997.
[23] L. Wang, "Analysis and design of hierarchical fuzzy systems," *IEEE Trans. Fuzzy Syst.*, vol. 7, pp. 617–624, Oct. 1999.
[24] K. Wu, "Fuzzy interval control of mobile robots," *J. Comput., Elect. Eng.*, vol. 22, no. 3, pp. 211–229, 1996.
[25] K. Wu and J. Mendel, "Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 10, pp. 622–639, Oct. 2002.

Hani A. Hagras (M'03) received the B.Sc. and M.Sc. degrees in electrical engineering from Alexandria University, Alexandria, Egypt, and the Ph.D. degree in computer science from the University of Essex, Colchester, U.K.

He is currently a Senior Lecturer in the Department of Computer Science at the University of Essex. His research interests include computational intelligence, robotics, embedded agents, pervasive computing, and ambient intelligence.

Dr. Hagras is a Member of the Executive Team of the IEE's Robotics and Mechatronics Professional Network.