IEEE Robotics and Automation Magazine, Vol. 8, No.3, pp 53–69, September 2001

Prototyping Design and Learning in Outdoor Mobile Robots operating in unstructured outdoor environments

Hani Hagras, *Member, IEEE,* Victor Callaghan, Martin Colley

Department of Computer Science, University of Essex, Wivenhoe Park, Colchester, CO43SQ, UK
Email: hani@essex.ac.uk

Index Terms— Type-2 fuzzy logic systems, Type-2 fuzzy logic controllers, Hierarchical type-2 fuzzy logic controllers, Mobile robots, Behaviour based robots, Navigation in indoor and outdoor environments,

Abstract

Natural outdoor environments present major navigation challenges to autonomous mobile robots as they involve large amounts of complex sensory data with uncertainties that are inherent to the sensing systems and the unstructured highly dynamic environments involved. The traditional type-1 Fuzzy Logic Controller (FLC) using crisp type-1 fuzzy sets cannot directly model and handle all such uncertainties. A Type-2 Fuzzy Logic Controller (FLC) using type-2 fuzzy sets can model and minimise the effect of uncertainties to produce a better performance. In this paper we present a novel control architecture for autonomous mobile robots based on a hierarchy of Type-2 FLCs to implement the basic navigation behaviours and the coordination between robot control behaviours. We have implemented this 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 linguistic and numerical uncertainties associated with the changing environment and robot conditions and resulted in very good performance that out-performed the type-1 based control system whilst achieving a significant rule reduction compared to the type-1 system.

1. 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. The choice of adequate methods to model and deal with such uncertainties is crucial for mobile robots navigating in changing unstructured environments.

The Fuzzy Logic Controller (FLC) has been credited with being an adequate methodology for designing robust controllers that are able to deliver a satisfactory performance in the face of uncertainty and imprecision [33]. As a result, the FLC has become a popular approach to reactive mobile robot control in recent years [33,34]. The use of fuzzy rules and linguistic variables makes fuzzy control an adequate design tool for non-linear systems for case where a precise mathematical model cannot be easily obtained, but where heuristic control knowledge is available, as is the case of mobile robots navigating in open unstructured environments.

There are many sources of uncertainty facing an mobile robots FLC that is navigating a 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 (ie their characteristics are changed by the environmental conditions such as wind, sunshine, humidity, rain, etc found in outdoor environments). These input uncertainties cause difficulty in determining the antecedents fuzzy membership functions; as if they are determined and tuned in a given environmental circumstances they may need to be changed in different environments.
- Uncertainties in control actions which translates to uncertainties in the output membership functions of the FLC. One cause for such uncertainties is the change of an actuator's characteristic due to, for instance, the inconsistency of the terrain or perhaps an environmental change. For example, a "fast speed" on a sunny day with dry ground may be different to "fast speed" on a rainy day with muddy ground and wheel slip .etc. Another source of uncertainty in the control actions is caused by changes of the mobile robot's physical properties such as changing of the diameter of the wheels, belts loosening due to wear and tear, etc. These output uncertainties cause difficulty in determining the consequents fuzzy membership functions; as if determined and tuned for a specific environmental and robot circumstance, it may have to be changed for different circumstances.
- Linguistic uncertainties, as the meaning of words used in the antecedents and consequents linguistic labels can be uncertain, as words mean different things to different people [23].

All of these uncertainties translate into uncertainties about fuzzy set membership functions [23].

To date all the FLC implementations in robot control are based on the traditional type-1 FLC. Type-1 FLC uses type-1 fuzzy sets which handles the uncertainties associated with the inputs and outputs by using *precise and crisp* membership functions that the user believes capture the uncertainties. Once the type-1 membership functions have been chosen, all the uncertainty disappears, because type-1 membership functions are totally precise [10]. According to Klir and Fogler [15] " *….it may seem problematical, if not paradoxical, that a representation of fuzziness is made using membership grades that are themselves precise real numbers*".

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 [7,20,33]. Even if a human expert can help in the specification of such a complex system, improper solutions maybe produced, since there is a high probability of neglecting some important aspects and overemphasizing others and it will take usually many iterations to determine, and tune a good FLC for specific robot and environment conditions. Also as the number of inputs variables increases (which is the case of mobile robots) the number of rules increase exponentially which creates difficulties for an expert who needs to determine large numbers of rules [9]. Another difficulty with developing a type-1 FLC using human experience is the uncertainty associated with the design process. As the rules and membership functions are based on knowledge from humans who use words to describe their knowledge, the process is error-prone as words mean different things to different people [24]. Also experts don't 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 [24].

Several researchers have also explored the use of learning techniques to learn the type-1 FLC for mobile robot control [3,4,8,22,30]. In our previous work [25,26,27,29] we have developed an online fast converging system based on our patented Fuzzy-Genetic technique (*British patent No 99-10539.7*) which was able to learn online the rules and membership functions for type-1 FLCs of mobile robots navigating in indoor and outdoor unstructured environments [REFS ?].

The type-1 FLCs have the problem that they were designed under specific robot and environment conditions (using human experience or using learning mechanisms). As the type-1 FLC, using precise type-1 fuzzy sets, cannot model or accommodate membership boundary uncertainties, the FLC may fail when used in conditions different from the design conditions. So in order for the type-1 FLC to deal the with the uncertainties caused by the changing environmental and robot conditions and their effects on determining the antecedents and consequents membership functions, people commonly end up wasting time in redesigning or tuning the FLC to work under different conditions.

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] [17]. The membership functions of type-2 fuzzy sets are three-dimensional, it is the new third-dimension of type-2 fuzzy sets that provides additional degrees of freedom that make it possible to directly model membership boundary uncertainties [23] which is a significant uncertainty in the overall system. The type-2 fuzzy sets are useful where it is difficult to determine the exact membership functions (which is the case for mobile robots in unstructured environments); hence, they are useful for incorporating and handling such uncertainties [17]. Although a type-2 membership function will also be totally precise, it includes the footprint of uncertainty that provides new degrees of freedom that let uncertainties be handled in totally new ways [22]. The type-2 FLC presented in this paper uses type-2 fuzzy sets for the input and output membership functions. The type-2 FLC produces a reduction in the rule base as opposed to the type-1 FLC as the linguistic uncertainty provided by the type-2 fuzzy sets allows us to cover the same range with a much smaller number of labels [22]. Also the type-2 FLC will be able to handle and model the uncertainties faced by the mobile robots in changing and dynamic unstructured environments and hence it will have the potential to produce a better performance than the type-1 FLC as will be demonstrated through experiments in indoor and outdoor dynamic unstructured environments.

Using a 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 this problem, a common strategy is to hierarchically decompose the control problem by breaking down the input space for analysis sharing it amongst multiple low level behaviours, each of which responds to specific types of situations, before integrating the recommendations of these behaviours via a high level coordination layer [34, 35]. Previous work had implemented the type-1 Hierarchical Fuzzy Logic Controller (HFLC) which used type-1 fuzzy systems for the implementation of the basic behaviours 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 behaviour 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 shown later.

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 [37] which used fuzzy interval control but they used type-1 fuzzy sets and type-1 FLC. In this the type-1 fuzzy sets were adaptive , using a set of parameters called sensitivity indices which determined the support and domain of membership function. For this they used a performance optimiser which dynamically adjusted the values of the sensitivity indices and the input and output fuzzy gains. However they did not address the use of type-2 fuzzy sets nor the issue of how to implement a real time type-2 FLC for a mobile robots navigating in a complex unstructured dynamic outdoor environment. Their fuzzy interval system was implemented in a very small mobile robot in a simple environment where it used only an edge following behaviour to navigate in a maze.

In Section (2) we will review the type-2 fuzzy sets and their associated terminologies and the type-2 set theoretic operations. In Section (3) we will introduce the type-2 FLC and its various components. In Section (4) we will introduce the type-2 Hierarchical Fuzzy Logic Controller (HFLC). In Section (5) we present the application of the type-2 HFLC to mobile robot control in indoor and outdoor unstructured environments. In Section (6) we will present the experiments and results using different real robots navigating in indoor and outdoor unstructured environments. Finally, conclusions are presented in Section (7).

2. Type-2 Fuzzy Sets

Type-2 fuzzy sets are better able to model the linguistic and numerical uncertainties associated with the robot FLC because their membership functions are themselves fuzzy [23]. According to [23] imagine blurring the type-1 membership function depicted in Figure (1-a) by shifting the points on the triangle either to the left or to the right and not necessarily by the same amounts, as in Figure (1-b). Then, at a specific value of *x*, say *x'*, there is no longer a single value for the membership function (*u'*); instead, the membership function takes on values wherever the vertical line intersects the blured area shaded in grey. Those values need not all be weighted the same; hence, we can assign an amplitude distribution to all those points. Doing this for all, we create a three-dimensional membership function—a type-2 membership function—that characterizes a type-2 fuzzy set. In the next subsection we review the definition of a type-2 fuzzy set and the associated terminologies.

2.1 Definition of type-2 fuzzy set and its associated terminologies

A type-2 fuzzy set, denoted $\tilde{A}$ is characterised by a type-2 membership function $\mu_{\tilde{A}}(x,u)$ [23], where $x \in X$ and $u \in J_x \subseteq [0,1]$, i.e.,

$$\tilde{A} = \{ ((x,u),\ \mu_{\tilde{A}}(x,u)) \mid \forall\ x \in X,\ \forall\ u \in J_x \subseteq [0,1] \} \qquad (1)$$

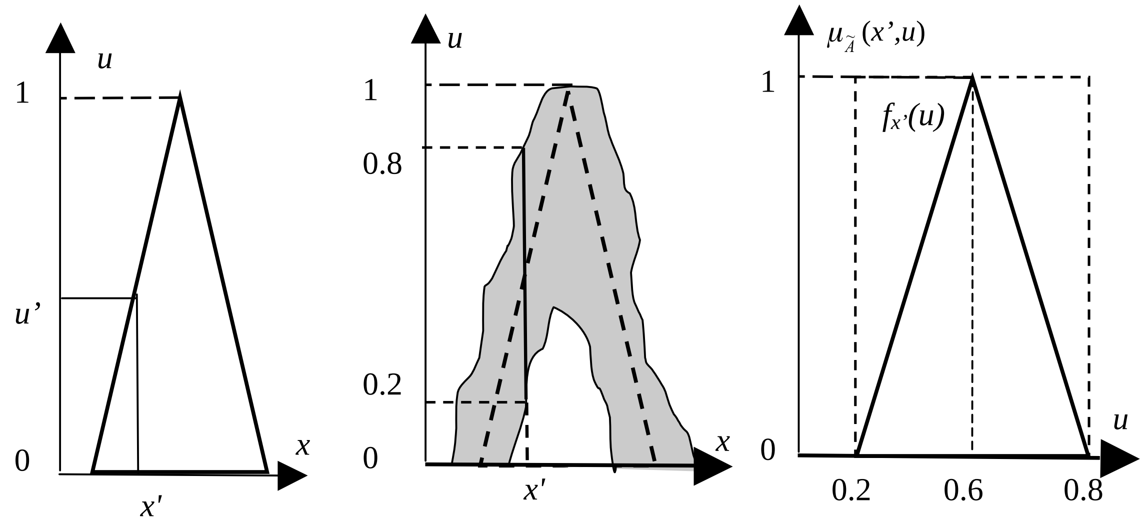
in which $0 \leq \mu_{\tilde{A}}(x,u) \leq 1$. $\tilde{A}$ can be expressed as

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

Where $\int\int$ denotes union over all admissible x and u [23].

If both X and J_x are discrete then $\tilde{A}$ can be expressed as [23]

$$\tilde{A} = \sum_{x \in X} \sum_{u \in J_x} \mu_{\tilde{A}}(x,u)/(x,u) \tag{3}$$



(a) (b) (c)

Figure (1): a) Type-1 membership function. b) Blurred type-1 membership function. c) Triangular secondary membership function plotted in thick line; interval secondary membership function plotted in dashed line

At each value of x say $x = x'$, the 2-D plane whose axes are u and $\mu_{\tilde{A}}(x',u)$ is called a vertical slice of $\mu_{\tilde{A}}(x,u)$ [23]. A *secondary membership function* is a vertical slice of $\mu_{\tilde{A}}(x,u)$. It is $\mu_{\tilde{A}}(x=x',u)$ for $x \in X$ and $\forall\ u \in J_{x'} \subseteq [0,1]$ [23], i.e.

$$\mu_{\tilde{A}}(x=x',u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u)/(u) \qquad J_{x'} \subseteq [0,1] \tag{4}$$

in which $0 \le f_{x'}(u) \le 1$. Because $\forall x' \in X$, according to [23] the prime notation on $\mu_{\tilde{A}}(x')$ is dropped and we refer to $\mu_{\tilde{A}}(x)$ as a secondary membership function; it is a type-1 fuzzy set which is also referred to as a secondary set. Many choices are possible for the secondary membership functions. According to Mendel [22] the name that we use to describe the entire type-2 membership function is associated with the name of the secondary membership functions; so, for example if the *secondary membership function* is triangular *then we refer to* $\mu_{\tilde{A}}(x,u)$ as a triangular type-2 membership function. Figure (1-c) shows triangular *secondary membership function at x'* which is drawn using the thick line.

Based on the concept of secondary sets, type-2 fuzzy sets can be written as the union of all secondary sets [22], so if X and J_x are continuous the type-2 fuzzy set $\tilde{A}$ can be written as follows [22]:

$$\widetilde{A} = \int_{x \in X} \mu_{\widetilde{A}}(x)/(x) = \int_{x \in X} \Big[\int_{u \in J_x} f_x(u)/u \Big] \Big/ x \qquad J_x \subseteq [0,1] \tag{5}$$

if X *and* J_x are both discrete the type-2 fuzzy set $\widetilde{A}$ can be written as follows[22]:

$$\widetilde{A} = \sum_{x \in X} \Big[\sum_{u \in J_x} f_x(u)/u \Big] \Big/ x \tag{6}$$

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* [22]. Interval secondary membership functions reflect a uniform uncertainty at the primary memberships of *x* [22]. Figure (1-c) shows the secondary membership at *x'* in case of interval type-2 fuzzy sets.

The domain of a secondary membership function is called *primary membership* of *x* [23]. In Equation (5), J_x is the primary membership of *x*, where $J_x \subseteq [0,1]$ for $\forall$ x $\in X$ [23].

2.2 Footprint of uncertainty

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) [23]. It is the union of all primary memberships [23], i.e.,

$$\text{FOU}\,(\widetilde{A}) = \bigcup_{x \in X} J_x \tag{7}$$

The shaded region in Figure (1-b) is the FOU. It is very useful, because according to Mendel and John [23] it not only focuses our attention on the uncertainties inherent in a specific type-2 membership function, whose shape is a direct consequence of the nature of these uncertainties, but it also provides a very convenient verbal description of the entire domain of support for all the secondary grades of a type-2 membership function. The shaded FOU implies that there is a distribution that sits on top of it—the new third dimension of type-2 fuzzy sets [23]. What that distribution looks like depends on the specific choice made for the secondary grades [23]. When they all equal one, the resulting type-2 fuzzy sets are called *interval type-2 fuzzy sets*[23]. Establishing an appropriate FOU is analogous to establishing a probability density function (pdf) in a probabilistic uncertainty situation [24]. The larger the FOU the more uncertainty there is. When the FOU collapses to a curve, then its associated type-2 fuzzy set collapses to a type-1 fuzzy set, in much the same way that a pdf collapses to a point when randomness disappears [24]. 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 [24]. According to Mendel and Wu [24], 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-1fuzzy sets, the issue of which type-1 membership function to choose diminishes in importance.

2.3 Embedded fuzzy sets

For continuous universes of discourse *X* and *U* , an embedded type-2 set $\widetilde{A}_e$ is defined as follows

$$\widetilde{A}_e = \int_{x \in X} [f_x(u)/u]/x \quad u \in J_x \subseteq U = [0,1] \tag{8}$$

Set $\widetilde{A}_e$ is embedded in $\widetilde{A}$ and there is an uncountable number of embedded type-2 sets is $\widetilde{A}$ [22].

For discrete universes of discourse X and U *an embedded type-2 set* $\widetilde{A}_e$ has N elements, where $\widetilde{A}_e$ contains exactly one element from J_{x1}, J_{x2},.... J_{xN}, namely $u_1, u_2, \ldots\ldots u_N$, each with its associated secondary grade $f_{x1}(u_1)$, $f_{x2}(u_2)$,….., $f_{xN}(u_N)$ [11,23], i.e.

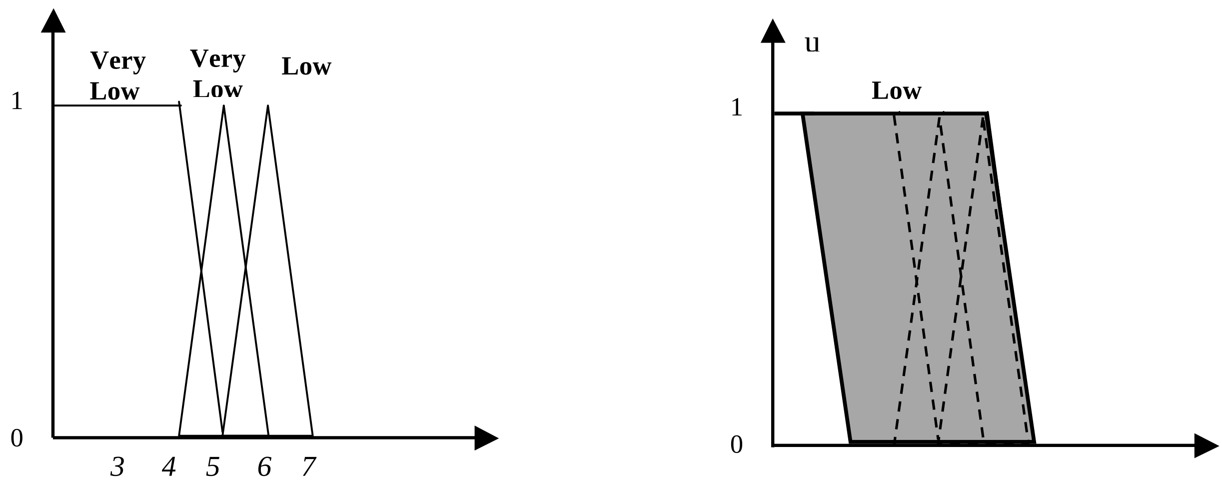
$$\widetilde{A}_e = \sum_{d=1}^{N} [f_{xd}(u_d)/u_d]/x_d \quad u_d \in J_{x_d} \subseteq U = [0,1] \tag{9}$$

Set $\widetilde{A}_e$ is embedded in $\widetilde{A}$ and there is a total of $\prod_{d=1}^{N} M_d$ $\widetilde{A}_e$ [23] where M_d is the total number of points where $\mu_{\widetilde{A}}(x_d,u) \neq 0$. If for example the X universe of discourse is discretised into 5 points and J_{x1}, J_{x2} each has 3 points in which $\mu_{\widetilde{A}}(x_d,u) \neq 0$ where d=1,2, and J_{x3}, J_{x4}, J_{x5} each has 4 points in which $\mu_{\widetilde{A}}(x_d,u) \neq 0$, where d=3,4,5 in this case we will have 3*3*4*4*4= 576 *embedded type-2 sets (* $\widetilde{A}_e$ *).*

For continuous universes of discourse X and U , an embedded type-1 set A_e is defined as follows [22]

$$A_e = \int_{x \in X} u/x \qquad u \in J_x \subseteq U = [0,1] \tag{10}$$

Set A_e is the union of all the primary memberships of set $\widetilde{A}_e$ in Equation (8) and there is an uncountable number of A_e



(a) (b)

Figure(2): a)Three type-1 fuzzy sets representing an input to the FLC. b) The three type-1 fuzzy sets in Figure (2-a) are embedded in the LOW type-2 fuzzy set.

For discrete universes of discourse X and U *an embedded type-1 set* A_e has N elements, one each from J_{x1}, J_{x2},.... J_{xN}, namely $u_1, u_2, \ldots\ldots u_N$, [23],*i.e.*

$$A_e = \sum_{d=1}^{N} u_d / x_d \quad u_d \in J_{xd} \subseteq U = [0,1] \tag{11}$$

Set A_e is the union of the primary memberships of set $\widetilde{A}_e$ in Equation (9) and there is a total of $\prod_{d=1}^{N} M_d$ A_e [11,23].

It has proven by Mendel and John [23] that a type-2 fuzzy set $\widetilde{A}$ can be represented as the union of its type-2 embedded sets, i.e.

$$\widetilde{A} = \sum_{f=1}^{n} \widetilde{A}_e^f \qquad \text{where } n \equiv \prod_{d=1}^{N} M_d \tag{12}$$

Figure (2-a) shows three type-1 fuzzy sets (*Very Very Low, Very Low and Low*) used to express in detail the different fuzzy levels of Low for an input to the FLC. In Figure (2-b) notice that the type-1 fuzzy sets for *Very Very Low, Very Low and Low* are embedded in the type-2 fuzzy set *Low*, not only this but there is a total of $\prod_{d=1}^{N} M_d$ embedded type-1 fuzzy sets.

After reviewing the definition of the type-2 fuzzy sets and their associated terminologies, we can realise that 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:

- The type-2 fuzzy sets through their FOUs 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 FLCs that are based on type-2 fuzzy sets will have the potential to produce a better performance than the type-1 FLCs.
- Type-2 fuzzy sets enables us to handle the uncertainty associated with determining the exact membership functions for the fuzzy sets associated with the inputs and outputs of the FLC [17]. Also regardless of the choice of the primary membership function (triangle, Gaussian, trapezoid), the resulting FOU is about the same as a result the FOU of a type-2 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-1fuzzy sets, the issue of which type-1 membership function to choose diminishes in importance [24].
- Each input and output will be represented by a big number of type-1 fuzzy which are embedded in the type-2 fuzzy sets. 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 give a much smoother control surface and response [12].
- Using type-2 fuzzy sets to represents the FLC inputs and outputs will result in the reduction of the FLC rule base when compared to using type-1 fuzzy sets as type-2 fuzzy sets rely in linguistic uncertainty to cover the same range as type-1 fuzzy sets with much smaller number of labels and the rule reduction will be greater when the number of the FLC inputs increases[22]. In terms of the FLC, uncertainty can fire rules, which is not available in type-1 FLC [22].
- According to Mendel [22] by modelling uncertainties using appropriate FOUs, it should be possible to design for robustness using type-2 FLCs.

So using type-2 fuzzy sets to represent the inputs and outputs of a mobile robot FLC enables us to model and handle uncertainties and can result in a better performance than when using type-1 fuzzy sets while using less number of rules, this will be justified through experiments using real robots navigating in challenging unstructured environments.

2.4 Interval Type-2 Fuzzy sets

In Equation (5) and (6) 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[22]. Interval secondary membership functions reflect a uniform uncertainty at the primary memberships of x [22]. Interval type-2 sets are very useful when we have no other knowledge about secondary memberships [17]. The membership grades of the interval type-2 fuzzy sets are called "interval type-1 fuzzy sets" [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*] [17]. The two end-points are associated with two type-1 membership functions that are referred to as *upper* and *lower* membership functions [17].

The upper and lower membership functions are two type-1 membership functions which are bounds for the footprint of uncertainty *FOU* ($\widetilde{A}$) of a type-2 fuzzy set $\widetilde{A}$ [22]. The upper membership function is associated with the upper bound of *FOU* ($\widetilde{A}$) and is denoted by $\overline{\mu}_{\widetilde{A}}(x), \forall x \in X$ and can be written as follows [22]:

$$\overline{\mu}_{\widetilde{A}}(x) \equiv \overline{FOU(\widetilde{A})} \quad \forall x \in X \tag{13}$$

The lower membership functions is associated with the lower bound of *FOU* ($\widetilde{A}$) and is denoted by $\underline{\mu}_{\widetilde{A}}(x), \forall x \in X$ and can be written as follows [22]:

$$\underline{\mu}_{\widetilde{A}}(x) \equiv \underline{FOU(\widetilde{A})} \quad \forall x \in X \tag{14}$$

According to Mendel [22] we can re-express Equation (5) as follows to represent the type-2 fuzzy set $\widetilde{A}$ in terms of upper and lower membership functions as follows:

$$\widetilde{A} = \mu_{\widetilde{A}}(x,u) = \int_{x \in X} \mu_{\widetilde{A}}(x)/(x) = \int_{x \in X} \left[\int_{u \in J_x} f_x(u)/u \right] / x$$

$$= \int_{x \in X} \left[\int_{u \in [\underline{\mu}_{\widetilde{A}}(x), \overline{\mu}_{\widetilde{A}}(x)]} f_x(u)/u \right] / x \tag{15}$$

The secondary membership $\mu_{\widetilde{A}}(x)$can be expressed in terms of upper and lower membership functions as follows [22]:

$$\mu_{\widetilde{A}}(x) = \int_{u \in [\underline{\mu}_{\widetilde{A}}(x), \underline{\mu}_{\widetilde{A}}(x)]} f_x(u)/(u) \tag{16}$$

In case of the interval type-2 fuzzy sets when the secondary membership functions are interval sets where $f_x(u)$= 1, so we can write the interval type-2 fuzzy sets as follows [22]:

$$\widetilde{A} = \int_{x \in X} \left[\int_{u \in [\underline{\mu}_{\widetilde{A}}(x), \underline{\mu}_{\widetilde{A}}(x)]} 1/u \right] / x \tag{17}$$

2.5 Type-2 Set Theoretic Operations

For type-2 fuzzy sets there are new operators named the meet and join which replaces the intersection and union in type-1 fuzzy sets. Linag and Mendel [18] had derived the expressions for meet and join in interval type-2 fuzzy sets in which we need to compute the join, meet of secondary membership functions which are type-1 interval fuzzy sets [22].

Let $F = \int_{v \in F} 1/v$ be an interval type-1 set with domain v∈ $[l_f, r_f] \subseteq [0,1]$ and $G = \int_{w \in G} 1/w$ is another interval type-1 set with domain $[l_g, r_g] \subseteq [0,1]$. The *meet Q* between *F* and *G* under the product t-norm used in our type-2 FLC is written as follows[18]

$$Q= F \sqcap G = \int_{q \in [l_f l_g, r_f r_g]} 1/q \tag{18}$$

From Equation(18) each term in $F \sqcap G$ is equal to the product of $v.w$ for some $v \in F$ *and* $w \in G$ in which the smallest term being $l_f l_g$ and the largest is $r_f r_g$. Since both *F* and *G* have continuous domains, $F \sqcap G$ has a continuous domain, therefore $F \sqcap G$ *is an interval type-1 set with domain* $[l_f l_g, r_f r_g]$ [17]. In a similar manner the meet under product t-norm of *n* interval type-1 sets $F_1, \ldots\ldots, F_n$ having domains $[l_1, r_1]$, …..$[l_n, r_n]$ respectively, is an interval set with domain with domain $\left[\prod_{s=1}^{n} l_s, \prod_{s=1}^{n} r_s\right]$. The meet under minimum t-norm is calculated in a similar manner [17].

The join between *F and G* is given by

$$Q = F \sqcup G = \int_{q \in [l_f \vee l_g, r_f \vee r_g]} 1/q \tag{19}$$

Where $q = v \vee w$, where ∨ denotes the maximum operation used in out type-2 FLC.

The join of *n* interval type-1 sets $F_1, \ldots\ldots, F_n$ having domains $[l_1, r_1]$, …..$[l_n, r_n]$ respectively is an interval set with domain [$(l_1 \vee l_2 \vee \ldots \vee l_n)$, $(r_1 \vee r_2 \vee \ldots \vee r_n)$], i.e. with domain equal [max($l_1, l_2, \ldots, l_n$), max($r_1, r_2, \ldots, r_n$] [18].

3. Type-2 Fuzzy Logic Controller (FLC)

A type-2 FLC is depicted in Figure (3), it contains five components which are a fuzzifier, rule base, fuzzy inference engine, type-reducer and a defuzzifier. In developing the type-2 FLC for mobile robot control both the inputs and outputs will be represented by type-2 fuzzy sets to account for the uncertainties associated with the input and output variables and the difficulty to precisely determine their memberships. We use interval type-2 fuzzy sets to represent the input and output variables as they are simple [23] and they are very useful when we have no other knowledge about secondary memberships [17] (which the case in robot control) as they distribute the uncertainty evenly among all admissible primary memberships [22]. Also according to Mendel and John [23] at present it is very difficult to justify the use of other kinds of type-2 fuzzy sets, e.g., as there is no best choice for type-1 fuzzy sets, so to compound this non-uniqueness by leaving the choice of the secondary membership functions arbitrarily is hardly justifiable [23]. Furthermore the general type-2 FLC is computationally intensive [18] and the computation simplifies a lot when using interval type-2 FLC (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 activates the inference block and the rule base to produce output type-2 fuzzy sets. The type-2 fuzzy outputs of the inference engine are then processed by the type-reducer which combines the output sets and then performs a centroid calculation, which leads to a type-1 fuzzy sets called the type-reduced sets [22]. The defuzzifier can then defuzzify the type-1 fuzzy outputs to produce crisp outputs to be fed to the actuators. Like the type-1 FLC, the type-2 FLC can be viewed as a mapping from crisp inputs to crisp outputs and can be expressed quantitatively as $y= f(x)$. However a type-2 FLC has more design degrees of freedom than the type-1 FLC, because its type-2 fuzzy sets are described by more parameters than are type-1 fuzzy sets [22].

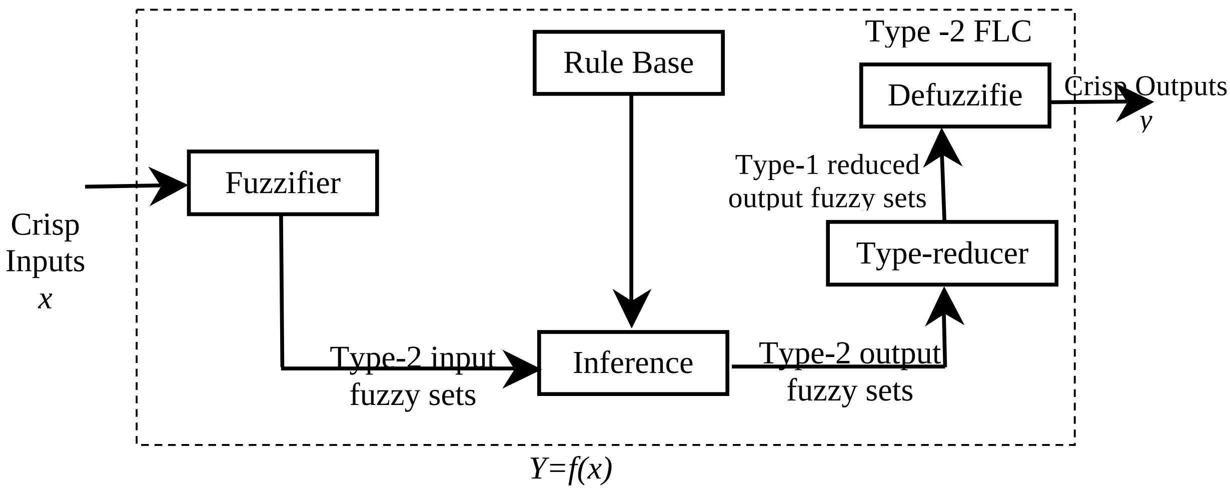


Figure (3): Type-2 Fuzzy Logic Controller (FLC)

According to Karnik *et al.* [14] as the type-reduced set of a type-2 FLC is the centroid of a type-2 fuzzy output set for the FLC; consequently, each element of the type-reduced set is the centroid of some type-1 set embedded in the output set of the type-2 FLC. Each of these embedded sets might be thought of as an output set of some type-1 FLC and, correspondingly, the type-2 FLC can be thought of as a collection of many different type-1 FLCs. Each of these type-1 FLCs is *embedded* in the type-2 FLC, so the type-reduced set is a collection of the outputs of all the type-1 FLCs embedded in the type-2 FLC. So according to Karnik *et al.* [14] If we think of a type-2 FLC as a "perturbed" version of a type-1 FLC, due to uncertainties in the membership functions, the type-reduced set of the type-2 FLC can then be thought of as representing the uncertainty in the crisp output due to uncertainties in the membership functions. The crisp outputs to the robot actuators can be obtained by aggregating the outputs of all the embedded type-1 FLCs [22]. In this way the type-2 FLC has the potential to over perform the type-1 FLC as it is dealing with the uncertainty thorough different embedded type-1 FLCs.

In the following subsection we will introduce each block in the type-2 FLC

3.1 The 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, this fuzzy sets can, in general, be a type-2 fuzzy input set $\widetilde{A}_x$ [22]. However we will use a 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, i.e. [22]:

$\widetilde{A}_x$ is a type-2 fuzzy singleton if $\mu_{\widetilde{A}_x}(\mathbf{x})$=1/1 for **x=x'** and $\mu_{\widetilde{A}_x}(\mathbf{x})$= 1/0 for all other **x≠ x'**

3.2 Rule Base

According to Mendel [22] the rules will remain the same as in type-1 FLC but the antecedents and the consequents will be represented by 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 rule base for this Multi Input Multi Output *(MIMO)* FLC with M rules can be written as follows:

$$R= \{R^1_{MIMO}, R^2_{MIMO}, \ldots\ldots, R^M_{MIMO}\} \tag{20}$$

Where the i^{th} rule has the following format:

$$R^i_{MIMO} : IF\ x_1\ is\ \widetilde{F}_1^i\ and\ \ldots\ldots and\ x_p\ is\ \widetilde{F}_p^i\ THEN\ y_1\ is\ \widetilde{G}_1^i, \ldots\ldots, y_c\ is\ \widetilde{G}_c^i \quad i=1,\ldots.M \tag{21}$$

According to Mendel [22] as the type-2 FLC rule base will have the same structure as the type-1 FLC then we can use the results from [19] which states that a rule base of *Multi Inputs and Multi Outputs(MIMO)* can be considered as a group of *Multi Input Single Output(MISO)* rule bases *as follows:*.

$$R = \{ RB_{1\,MISO}, RB_{2\,MISO}, \ldots\ldots, RB_{c\,MISO} \} \tag{22}$$

Where $RB_{k\,MISO}$ is the rule base for the *Multi p Input* and the k^{th} *Single Output (MISO),* where $k=1,..c$ and c is the total number of outputs of the robot type-2 FLC, this rule base contains M rules.

3.3 Fuzzy Inference Engine

The fuzzy inference engine block in Figure (3) gives a mapping from input type-2 sets to output type-2 sets [22]. 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. Just as the sup-star composition is the backbone computation for type-1 FLC, the extended sup-star composition is the backbone computation for a type-2 FLC [22]. The extended sup-star composition can be obtained simply by extending the type-1 sup-star composition by replacing type-1 membership functions by type-2 membership functions, the sup operation with join operation and the t-norm operation with the meet operation [22].

In subsection (3.2) we mentioned that a *MIMO* rule base can be considered as a group of *MISO* rule bases, so like [22] we will concentrate on *MISO* rule base. 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 [22]:

$$R^i_{k\,MISO}: IF\ x_1\ is\ \widetilde{F}_1^i\ and\ \ldots\ldots and\ x_p\ is\ \widetilde{F}_p^i\ THEN\ y_k\ is\ \widetilde{G}_k^i \quad i=1,\ldots.M \tag{23}$$

Equation (23) can be interpreted as a type-2 fuzzy implication, let $\widetilde{F}_1^i \times \ldots \times \widetilde{F}_p^i = \widetilde{A}^i$ then according to [22] Equation (24) can be rewritten as follows:

$$R^i_{k\,MISO}: \widetilde{F}_1^i \times \ldots \times \widetilde{F}_p^i \rightarrow \widetilde{G}_k^i = \widetilde{A}^i \rightarrow \widetilde{G}_k^i \qquad i=1,\ldots.M \tag{24}$$

$R^i_{k\,MISO}$ 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 [22] $\mu_{R^i}(\mathbf{x},y_k)$ can be written as follows:

$$\mu_{R^i}(\mathbf{x},y_k) = \mu_{\tilde{A}^i \rightarrow \tilde{G}^i{}_k}(\mathbf{x},y_k) = \mu_{\tilde{F}_1^i}(x_1) \sqcap \ldots \sqcap \mu_{\tilde{F}_p^i}(x_p) \sqcap \mu_{\tilde{G}^i{}_k}(y_k)$$

$$= [\sqcap_{j=1}^{p} \mu_{\tilde{F}_j^i}(x_j)] \sqcap \mu_{\tilde{G}^i{}_k}(y_k) \qquad (25)$$

Each rule $R^i_{k\ MISO}$ determines a type-2 output fuzzy set $\tilde{B}^i_k$ for the k^{th} output y_k. $\tilde{B}^i_k$ is obtained using the extended sup-star composition (denoted by o) where $\tilde{B}^i_k = \tilde{A}_x \text{ o } R^i_{k\ MISO}$. As we are using the singleton fuzzification for our type-2 FLC then the type-2 fuzzy input set $\tilde{A}_x$ contains a single element **x'** and each $\mu_{\tilde{X}_j}(x_j)$ is non zero only at one point $x_j = x_j'$, so according to [22] $\mu_{\tilde{B}^i_k}(y_k)$ can be written as follows:

$$\mu_{\tilde{B}^i_k}(y_k) = \mu_{\tilde{G}^i{}_k}(y) \sqcap [\sqcap_{j=1}^{p} \mu_{\tilde{F}_j^i}(x_j')] \qquad y_k \in Y_k,\ i=1, \ldots .M \qquad (26)$$

In our interval type-2 FLC we will use the meet under product t-norm so according to [22] the result of the input and antecedent operations, which are contained in the firing set $\sqcap_{j=1}^{p} \mu_{\tilde{F}_j^i}(x_j') \equiv F^i(\mathbf{x}')$, is an interval type-1 set, as follows

$$F^i(\mathbf{x}') = [\underline{f}^i(\mathbf{x}'), \overline{f}^i(\mathbf{x}')] \equiv [\underline{f}^i, \overline{f}^i] \qquad (27)$$

where

$$\underline{f}^i(\mathbf{x}') = \underline{\mu}_{\tilde{F}_1^i}(x_1') * \ldots\ldots * \underline{\mu}_{\tilde{F}_p^i}(x_p') \qquad (28)$$

and

$$\overline{f}^i(\mathbf{x}') = \overline{\mu}_{\tilde{F}_1^i}(x_1') * \ldots\ldots * \overline{\mu}_{\tilde{F}_p^i}(x_p') \qquad (29)$$

Where * denotes the product operation.

In our interval type-2 FLC the fired consequent fuzzy set $\mu_{\tilde{B}^i_k}(y_k)$ for each rule $R^i_{k\ MISO}$ in Equation (27) can be written as follows according to [22] as follows:

$$\mu_{\tilde{B}^i_k}(y_k) = \int_{b^i \in [\underline{f}^i * \underline{\mu}_{\tilde{G}^i_k}(y_k),\ \overline{f}^i * \overline{\mu}_{\tilde{G}^i_k}(y_k)]} 1/b^i \qquad y_k \in Y_k \qquad (30)$$

Where $\underline{\mu}_{\tilde{G}^i_k}(y_k)$ and $\overline{\mu}_{\tilde{G}^i_k}(y_k)$ are the lower and upper membership grades of $\mu_{\tilde{G}^i_k}(y_k)$ [22].

Suppose than N of the M rules in the FLC fire, where $N \leq M$, then the combined output fuzzy set for the k^{th} *output* is obtained by combining the fired output consequent sets; i.e. $\mu_{\tilde{B}_k}(y_k) = \sqcup_{i=1}^{N} \mu_{\tilde{B}^i_k}(y_k)\ \ y_k \in Y_k$; then according to [22]

$$\mu_{\tilde{B}_k}(y_k) = \int_{b \in \left[[\underline{f}^1 * \underline{\mu}_{\tilde{G}^1_k}(y_k)] \vee \ldots\ldots \vee [\underline{f}^N * \underline{\mu}_{\tilde{G}^N_k}(y_k)],\ [\overline{f}^1 * \overline{\mu}_{\tilde{G}^1_k}(y_k)] \vee \ldots\ldots \vee [\overline{f}^N * \overline{\mu}_{\tilde{G}^N_k}(y_k)] \right]} 1/b \qquad y_k \in Y_k \qquad (31)$$

Where $\vee$ denotes the maximum operation in our type-2 FLC.

In a similar manner all the combined output fuzzy sets for the c different outputs of the robot controller can be calculated by finding for each rule i $\underline{\mu}_{\tilde{G}^i_k}(y_k)$ and $\overline{\mu}_{\tilde{G}^i_k}(y_k)$ for each output $k = (1, \ldots .c)$ and substituting in Equation (31) while the values of $\underline{f}^i, \overline{f}^i$ will be the same for all the different outputs, where $i=(1 \ldots N)$ and N is the number of the firing rules in the type-2 *MIMO* fuzzy rule base of total M rules.

3.4 Type Reduction

Type-reduction was proposed by Karnik and Mendel [13,14,18], 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” [18]. These type-reduced sets are then deuzzified to obtain crisp outputs that are sent to the motors of the mobile robots.

As we are dealing with interval sets, the type-reduced set for the k^{th} output Y_{TR_k} will also be an interval set [22] and has the following structure:

$$Y_{TR_k} = [\, y_{lk},, y_{rk}] \tag{32}$$

where $k= (1,\ldots.c)$and c is the total number of outputs for the FLC.

As in [18] 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-reduction which have problems when only one rule fires [22]. 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 $Y_{\cos}$ using the center of sets type-reduction can be expressed as follows:

$$Y_{\cos}(\mathbf{x})_k = [\, y_{lk}, y_{rk}]= \int_{y_k^1\in[y_{lk}^1, y_{rk}^1]} \cdots \int_{y^M\in[y_{lk}^M, y_{rk}^M]} \int_{f^1\in[\underline{f}^1, \overline{f}^1]} \cdots \int_{f^M\in[\underline{f}^M, \overline{f}^M]} 1 \Big/ \frac{\sum_{i=1}^{M} f^i y_k^i}{\sum_{i=1}^{M} f^i} \tag{33}$$

Where $\mathbf{x}$ is the input vector to the FLC and $Y_{\cos}(\mathbf{x})_k$ for the for k^{th} output is an interval set determined by its left most y_{lk} and its right most point y_{rk}. $i=(1\ldots M)$ where M is the number of rules. y_k^i corresponds to the centroid of the type-2 interval consequent set $\tilde{G}_k^i$ of the i^{th} rule for the for k^{th} output , y_k^i is a type-1 interval fuzzy set determined by its left most point y_{lk}^i and its right most point y_{rk}^i [18]. f^i denotes the firing strength (degree of firing) of the i^{th} rule which is an interval type-1 set determined by its left most $\underline{f}^i$ and right most point $\overline{f}^i$ [18] where $\underline{f}^i$ is calculated using Equation (28) and $\overline{f}^i$ is calculated using Equation (29).

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 the i^{th} rule which is conducted ahead of time and before starting the control cycle of the robot FLC. 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 subsections we describe these two stages.

3.4.1 Calculating the centroids of the rule consequents

The center of sets type-reducer for the k^{th} output replaces each type-2 interval consequent for the i^{th} rule $\tilde{G}_k^i$ by its centroid y_k^i which is a type-1 interval set determined by its left most point y_{lk}^i and its right most point y_{rk}^i. For any output k, the consequent of each rule will be one of the output type-2 fuzzy sets representing this output. If for each output we determined the centroids of all the output type-2 interval fuzzy sets representing this output, then the centroid of the type-2 interval consequent for the i^{th} rule will be one of the centroids of the pre-calculated type-2 output sets which corresponds to the rule consequent. So for any output k the centroid of the i^{th} rule consequent y_k^i will be one of centroids of the output fuzzy sets y_k^t which corresponds to the rule consequent where $t=1,..T$ and T is number of output fuzzy sets for this output. We must calculate these centroids ahead of time, and before starting the control cycle of the type-2 FLC, as they are needed for the computation of $Y_{\cos}(\mathbf{x})_k$ [22].

The centroid of the t^{th} output fuzzy set consequent 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 [22].

$$y_k^t = [y_{lk}^t, y_{rk}^t] = \int_{\vartheta_1 \in J_{y1}} \cdots\cdots \int_{\vartheta_N \in J_{yN}} 1 \Bigg/ \frac{\sum_{g=1}^{G} y_g \theta_g}{\sum_{g=1}^{G} \theta_g} \tag{34}$$

To compute the centroid of each output fuzzy set we will use a procedure which was developed by [22]. In this procedure for the k^{th} output we will discretise each output fuzzy set into G points, $y_1, \ldots.. y_G$ where $g=(1\ldots G)$. Let $J_{y_g} \equiv [L_g, R_g]$ and $h_g = (L_g+R_g)/2$ and $\Delta_g = (R_g - L_g)/2$. Figure (4.b) shows for each y_g how to calculate L_g , R_g , h_g , Δ_g needed by the iterative procedure in Figure (4.a). Note that y as a function of $(\theta_1, \ldots.\theta_G)$ can be written as follows [22]

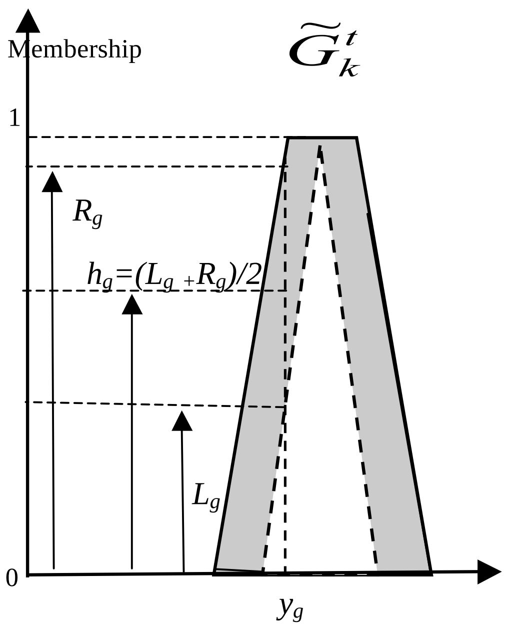
$$y(\theta_1, \ldots.\theta_N) = \frac{\sum_{g=1}^{G} y_g \theta_g}{\sum_{g=1}^{N} y_g} \tag{35}$$

To calculate y_{rk}^t we will apply the iterative procedure proposed by [22] shown in Figure (4-a)

Without loss of generality assume that y_g are arranged in an ascending order; i.e. $y_1 \le y_2 \le \ldots \le y_G$ *.Then,*

1. Initialize θ_g by setting $\theta_g = h_g$ for $g = 1, \ldots., G$ and then compute $y' = y(h_1, \ldots.\ h_N)$ using Equation (35).
2. Find z $(1 \le z \le G-1)$ such that $y_z \le y' \le y_{z+1}$.
3. *Set* $w_g = h_g - \Delta_g$ *, for* $g \le z$ *and* $w_g = h_g + \Delta_g$ *, for* $g \ge z+1$ *and compute* $y'' = y(h_1 - \Delta_1, \ldots.\ h_z - \Delta_k, h_{z+1} + \Delta_{z+1}, \ldots, h_N + \Delta_N)$ *using Equation (35)*
4. *Check if* $y'' \ge y'$ *. If yes stop* y'' *is the maximum value of* $y(\theta_1, \ldots.\theta_G)$ *and it is* y_r^i*, if no go to step 5*
5. *Set y' equal y". Go to step 2.*

(a)



(b)

Figure (4): a)An iterative procedure to calculate y_r^t .b)Parameters needed by each y_g in the procedure in Figure (4-a).

The value of y_{lk}^{t} can be obtained using a similar procedure to the one used in Figure (4-a) but doing one change

- In step 3, set $w_g=h_g+\Delta_g$, *for* $g\leq z$ *and* $w_g=h_g-\Delta_g$, *for* $g\geq z+1$and compute $y''= y(h_1+\Delta_1 , \ldots. h_z+\Delta_z , h_{z+1}-\Delta_{z+1},\ldots, h_N-\Delta_N)$using Equation (35) .

This iterative procedure converges in at most G iterations [22]. Where each iteration consists of one pass through steps 2 to 5 as step 1 is an initialisation step. We have chosen G 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.

3.4.2 Calculating the type-reduced set

For any output k to compute $Y_{\cos}(\mathbf{x})_k$ we need to compute its two end points y_{lk} and y_{rk} . For each rule we need to calculate f^i which is the firing strength (degree of firing) of the i^{th} rule which is an interval type-1 set determined by its left most point $\underline{f}^i$ and its right most point $\overline{f}^i$ [18] where $\underline{f}^i$ is calculated using Equation (28) and $\overline{f}^i$ is calculated using Equation (29) where $i=(1\ldots M)$ and M is the number of rules. Also we need to attach to the firing strength f^i the centroid of the i^{th} rule consquent y_k^i which is a type-1 interval set determined by its left most point y_{lk}^i and its right most point y_{rk}^i . As mentioned in subsection (3.4.1) the i^{th} rule consequent will be one of the output fuzzy sets and thus its centroid will be one of centroids of the output fuzzy sets y_k^t which corresponds to the rule consequent where $t=1,..T$ and T is number of output fuzzy sets for each output.

According to [18] let the values of f^i and y_k^i that are associated with y_{lk} be denoted f_l^i and y_l^i respectively and the values of f^i and y^i that are associated with y_{rk} be denoted f_r^i and y_r^i respectively. From Equation (33) we see that

$$y_{lk}=\frac{\sum_{i=1}^{M} f_l^i y_{lk}^i}{\sum_{i=1}^{M} f_l^i} \tag{36}$$

and

$$y_{rk}=\frac{\sum_{i=1}^{M} f_r^i y_{rk}^i}{\sum_{i=1}^{M} f_r^i} \tag{37}$$

In order to calculate y_l 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_r we need to determine $\{ f_r^i , i=1,\ldots.M\}$ and its associated $\{ y_{rk}^i , , i=1,\ldots.M\}$.This can be done by using the procedure shown in Figure (5) to compute y_r [22]. This procedure was developed by [22] and it is a four step iterative procedure where step 1 is an initialisation 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_r in Equation(37) can be written as follows[22]:

$$y_{rk} = y_{rk}(\underline{f}^1, \ldots, \underline{f}^R, \overline{f}^{R+1}, \ldots \overline{f}^M, y_r^1, \ldots, y_r^M) = \frac{\sum_{u=1}^{R} \underline{f}^u y_{rk}^u + \sum_{v=R+1}^{M} \overline{f}^v y_{rk}^v}{\sum_{u=1}^{R} \underline{f}^u + \sum_{v=R+1}^{M} \overline{f}^v} \tag{38}$$

The procedure for computing y_l is similar to the one given to calculate y_r just replace y_r^i *by* y_l^i and in step 2 find $L (1 \le L \le M-1)$ such that $y_l^L \le y_l' \le y_l^{L+1}$. Additionally in step 3 we compute y_l in Equation (36) with $f_l^i = \overline{f}^i$ for $i \le L$ *and* $f_l^i = \underline{f}^i$ *for* $i>L$. then y_l in Equation (36) can be represented as:

$$y_{lk} = y_{lk}(\overline{f}^1, \ldots, \overline{f}^L, \underline{f}^{L+1}, \ldots \underline{f}^M, y_l^1, \ldots, y_l^M) = \frac{\sum_{u=1}^{L} \overline{f}^u y_{lk}^u + \sum_{v=L+1}^{M} \underline{f}^v y_{lk}^v}{\sum_{u=1}^{L} \underline{f}^e + \sum_{v=L+1}^{M} \overline{f}^v} \tag{39}$$

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

1. Compute y_r *in Equation(37) 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 (28) and Equation(29)respectively and let* $y_r' \equiv y_r$

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

3. Compute y_r *in Equation (37) using* $f_r^i = \underline{f}^i$ *for* $i \le R$ *and* $f_r^i = \overline{f}^i$ *for* $i>R$ *and let* $y_r'' \equiv y_r$

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

5. Set y_r' *equal to* y_r'' *and return to step 2.*

Figure (5): An iterative procedure to calculate y_r

This iterative procedure is proven [18,38] 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 in the control cycle to calculate the type-reduced sets which will be then defuzzified to give the output crisp controls to the robot output. It is noticed that the rule base grows larger as we have more inputs to the robot FLC as the number of rules in the rule base is equal to m^n where m is the number of fuzzy sets and n is the number of inputs. For example our mobile robots (indoor and outdoor) have 7 sonar inputs and one goal detection sensor, i.e. 8 sensors in total, if we represented each input only by three fuzzy sets this will result in $3^8 = 6561$. In this case we might need maximum 6561 iterations to determine y_{rk} and maximum 6561 iterations to determine

y_{lk}, thus we might need 13122 iterations to finish the type-reduction which will not allow the type-2 FLC to operate in real time thus. Wu and Mendel [38] introduced a method to approximate the type-reduced set by the inner and outer bound sets, however this methods needs training data to get a good approximation error and this data is difficult to acquire for a mobile robot which encounter a lot of unforeseen circumstances and will navigate mostly in unknowns environments.

3.5 Defuzzification

From the type-reduction stage we have for each output a type-reduced set $Y_{\cos}(\mathbf{x})_k$ where $k=1,..c$ where c is the total number of outputs. Each type-reduced set is an interval type-1 set determined by its left most point y_l and right most point y_r. 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* [18]:

$$Y_k(\mathbf{x}) = \frac{y_{lk} + y_{rk}}{2} \tag{40}$$

3.6 An illustrative example to summarize operation of the type-2 FLC

In this sub section we summarize the operation of the type-2 FLC through an example of a type-2 FLC that realises the right edge following behaviour for an outdoor robot. The objective of this behaviour is to follow an edge to the right of the robot at a desired distance, such type-2 FLC will have two inputs from two right side sonar sensors, the first input is from the Right Side Front sensor (RSF) and the second input is from the Right Side Back sensor (RSB). The RSF input will be denoted by x_1 and the RSB will be denoted by x_2. The type-2 FLC controls two outputs which are the robot speed denoted by y_1 and the robot steering denoted by y_2. Each input will be represented only by two type-2 fuzzy sets which are *Near* and *Far* as shown in Figure (6). The output robot speed will be represented by three type-2 fuzzy sets which are *Slow, Medium, Fast* while the robot steering will be represented by two type-2 fuzzy sets which are *Left, Right.* In what follows we will follow an input vector through the various components of the type-2 FLC till we get crisp output signals to the robot actuators. The input vector will consist of two inputs, the first one representing the reading of RSF which we term x_1' and the second input represents the reading of RSB which we term x_2', the type-2 FLC outputs corresponding to x_1' and x_2'are y_1'for the robot speed and y_2'for the robot steering.

3.6.1 Fuzzification

In the fuzzification stage each input is matched against its membership functions to calculate the upper and lower membership values for each fuzzy set. The input $x_{1'}$ of the RSF is matched against its membership functions in Figure (6-a) and it was found that the lower membership value for the *Near* type-2 fuzzy set is 0.35 while the upper membership value is 0.85. For the *Far* fuzzy set, the lower membership value is 0.15 while the upper membership value is 0.75. The input $x_{2'}$ of the RSB is matched against its membership functions in Figure (6-b) and it was found that the lower membership value for the *Near* fuzzy set is 0.1 and the upper membership value is 0.7. For the *Far* fuzzy set the lower membership value is 0.3 while upper membership value is 0.8.

3.6.2 The Rule Base

The rule base for this type-2 FLC is shown in Table (1), where *p* (the number of inputs) is 2 and *c* (the number of outputs) is 2. Any MIMO rule from Table (1) can be written according to Equation (21), for example rule (1) can be written as follows

$R^1{}_{MIMO}$: *IF* x_1 *is* $\tilde{F}_1^1$ *and* x_2 *is* $\tilde{F}_2^1$ *THEN* y_1 *is* $\tilde{G}_1^1$, y_2 *is* $\tilde{G}_2^1$

Where $\tilde{F}_1^1$ is the *Near* type-2 fuzzy set for RFS and $\tilde{F}_2^1$ is the *Near* type-2 fuzzy set for RBS and $\tilde{G}_1^1$ is the *Slow* output type-2 fuzzy set for the robot speed and $\tilde{G}_2^1$ is the *Left* output type-2 fuzzy set for the robot steering

Rule No	RSF	RSB	Speed	Steering
1	Near	Near	Slow	Left
2	Near	Far	Slow	Left
3	Far	Near	Medium	Right
4	Far	Far	Fast	Right

Table (1): An example rule base of a right edge following behaviour implemented by a type-2 FLC.

3.6.3 Fuzzy Inference Engine

In the fuzzy inference engine we need to calculate the firing strength of each rule. According to Equation (27) the firing strength of each rule is an interval type-1 set $[\underline{f}^i, \overline{f}^i]$ where $\underline{f}^i$ is calculated according to Equation (28) and $\overline{f}^i$ is calculated according to Equation (29). Note that in our FLC we use the meet under the product t-norm. So for rule (1) we can calculate $\underline{f}^1$ and $\overline{f}^1$ as follows:

$$\underline{f}^1 = \underline{\mu}_{\tilde{F}_1^1}(x_1').\underline{\mu}_{\tilde{F}_2^1}(x_2') = 0.35*0.1=0.035$$

where $\underline{\mu}_{\tilde{F}_1^1}(x_1')$ is the lower membership value for $x_{1'}$ for the *Near* type-2 fuzzy set which is 0.35 and $\underline{\mu}_{\tilde{F}_2^1}(x_2')$ is the lower membership value $x_{2'}$ for the *Near* type-2 fuzzy set which is 0.3, these membership values were calculated before in the fuzzification stage. In a similar manner by using the upper membership values we can calculate $\overline{f}^1$ as follows:

$$\overline{f}^1 = \overline{\mu}_{\tilde{F}_1^1}(x_1').\overline{\mu}_{\tilde{F}_2^1}(x_2') = 0.85*0.7=0.595.$$

In a similar manner we can calculate the rest of the firing strengths for all the rule as follows:

$$\underline{f}^2 = \underline{\mu}_{\tilde{F}_1^2}(x_1').\underline{\mu}_{\tilde{F}_2^2}(x_2') = 0.35*0.3=0.105, \quad \overline{f}^2 = \overline{\mu}_{\tilde{F}_1^2}(x_1').\overline{\mu}_{\tilde{F}_2^2}(x_2') = 0.85*0.8=0.68$$

$$\underline{f}^3 = \underline{\mu}_{\tilde{F}_1^3}(x_1').\underline{\mu}_{\tilde{F}_2^3}(x_2') = 0.15*0.1=0.015, \quad \overline{f}^3 = \overline{\mu}_{\tilde{F}_1^3}(x_1').\overline{\mu}_{\tilde{F}_2^3}(x_2') = 0.75*0.7=0.525$$

$$\underline{f}^4 = \underline{\mu}_{\tilde{F}_1^4}(x_1').\underline{\mu}_{\tilde{F}_2^4}(x_2') = 0.15*0.3 =0.045, \quad \overline{f}^4 = \overline{\mu}_{\tilde{F}_1^4}(x_1').\overline{\mu}_{\tilde{F}_2^4}(x_2') = 0.75*0.8=0.6$$

3.6.4 Type-Reduction

3.6.4.1 Calculating the centroids of the rule consequents

In this stage we need to calculate for each output the centroids of all the output type-2 fuzzy sets, so that we can calculate the centroid of the consequent of each rule which will be one from the output type-2 fuzzy sets as was explained in Section (3.4.1). To calculate these centroids we will use the iterative procedure shown in Figure (4-a) using 100 sampling points as explained in subsection (3.4.1). For each output we calculate the centroids of all the type-2 fuzzy sets $C_{\tilde{G}_k^t}$ $t=1,..T$. For the speed output, the number of output fuzzy sets $T=3$. Assume for illustrative purposes that that the centroid of the *Slow* fuzzy set is [0.43,0.55], the centroid of the *Medium* fuzzy set is [0.63,0.76] and the centroid of the *High* fuzzy set is [1.03,1.58]. Next we can determine the centroids of the rule consquents of the output speed $C_{\tilde{G}_1^i}$ as follows:

$$C_{G_1^1}=[y_{l1}^1, y_{r1}^1]=C_{G_1^2}=[y_{l1}^2, y_{r1}^2]=[0.43,0.55], C_{G_1^3}=[y_{l1}^3, y_{r1}^3]=[0.63,0.76], C_{G_1^4}=[y_{l1}^4, y_{r1}^4]=[1.03,1.58]$$

For the output steering the number of outputs fuzzy sets $T=2$. The right steering values are positive values and the left steering values are negative. Again for illustrative purposes assume that the centroid for the *Left* fuzzy set is [56.8, 85.4] and the centroid for *Right* fuzzy set is [-85.4,-56.8]. Next we can determine the centroids of the rule consquents $C_{\tilde{G}_2^i}$ as follows:

$$C_{G_2^1}=[y_{l2}^1, y_{r2}^1]=C_{G_2^2}=[y_{l2}^2, y_{r2}^2]=[-85.4, -56.8], C_{G_2^3}=[y_{l2}^3, y_{r2}^3]=C_{G_2^4}=[y_{l2}^4, y_{r2}^4]=[56.8,85.4]$$

3.6.4.2 Calculating the type-reduced set

For each output k to compute the type-reduced $Y_{\cos}(\mathbf{x})_k$ we need to compute its two end points y_{lk} and y_{rk} as was explained in subsection (3.4.2). Using the iterative procedure in Figure (5) we can determine L and R. For the speed output we donot need to reorder y_{l1}^i as they are already ordered in ascending order where $y_{l1}^1 \le y_{l1}^2 \le y_{l1}^3 \le y_{l1}^4$, *the* same applies for y_{r1}^i. By using the iterative procedure in Figure(5), it was found that $L=2$ so y_{l1} can be found by substituting in Equation (39) as follows:

$$y_{l1}=\frac{\sum_{u=1}^{2}\overline{f}^u y_{l1}^u+\sum_{v=3}^{4}\underline{f}^v y_{l1}^v}{\sum_{u=1}^{2}\underline{f}^u+\sum_{v31}^{4}\overline{f}^v}=\frac{\overline{f}^1 y_{l1}^1+\overline{f}^2 y_{l1}^2+\underline{f}^3 y_{l1}^3+\underline{f}^4 y_{l1}^4}{\overline{f}^1+\overline{f}^2+\underline{f}^3+\underline{f}^4}=\frac{0.595\times0.43+0.68\times0.43+0.015\times0.63+0.045\times1.03}{0.595+0.68+0.015+0.045}=0.452$$

to calculate y_{r1} we use the iterative procedure in Figure(5), it was found that R=3, so y_{r1} can be found by substituting in Equation (38) as follows:

$$y_{r1}=\frac{\sum_{u=1}^{3}\underline{f}^u y_{r1}^u+\sum_{v=4}^{4}\overline{f}^v y_{r1}^v}{\sum_{u=1}^{3}\underline{f}^u+\sum_{v=4}^{4}\overline{f}^v}=\frac{\underline{f}^1 y_{r1}^1+\underline{f}^2 y_{r1}^2+\underline{f}^3 y_{r1}^3+\overline{f}^4 y_{r1}^4}{\underline{f}^1+\underline{f}^2+\underline{f}^3+\overline{f}^4}=\frac{0.035\times0.55+0.105\times0.55+0.015\times0.76+0.6\times1.58}{0.035+0.105+0.015+0.6}=1.37$$

For the steering output, we donot need to reorder y_{l2}^i as they are already ordered in ascending order where $y_{l2}^1 \le y_{l2}^2 \le y_{l2}^3 \le y_{l2}^4$, *the* same applies for y_{r2}^i. By using the iterative procedure in Figure(5) it was found that $L=2$ so y_{l2} can be found by substituting in Equation (39) as follows:

$$y_{l2}=\frac{\sum_{u=1}^{2}\overline{f}^{u}y_{l2}^{u}+\sum_{v=3}^{4}\underline{f}^{v}y_{l2}^{v}}{\sum_{u=1}^{2}\underline{f}^{e}+\sum_{v31}^{4}\overline{f}^{v}}=\frac{\overline{f}^{1}y_{l2}^{1}+\overline{f}^{2}y_{l2}^{2}+\underline{f}^{3}y_{l2}^{3}+\underline{f}^{4}y_{l2}^{4}}{\overline{f}^{1}+\overline{f}^{2}+\underline{f}^{3}+\underline{f}^{4}}=\frac{0.595\times-85.4+0.68\times-85.4+0.015\times56.8+0.045\times56.8}{0.595+0.68+0.015+0.045}=-79.01$$

to calculate y_{r2} we use the iterative procedure in Figure(5), it was found that *R=2,* so y_{r2} can be found by substituting in Equation (38) as follows:

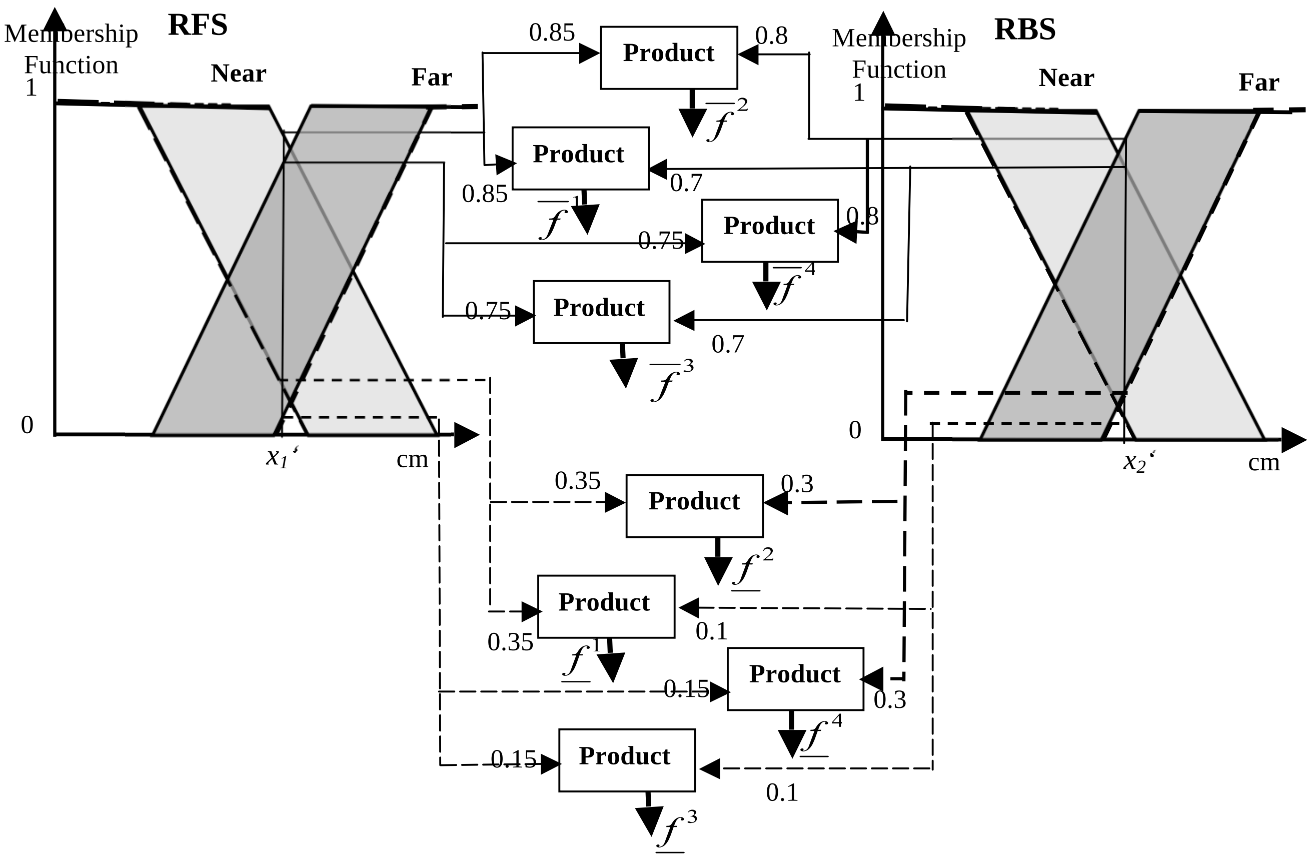
$$y_{r2}=\frac{\sum_{u=1}^{2}\underline{f}^{u}y_{r2}^{u}+\sum_{v=3}^{4}\overline{f}^{v}y_{r2}^{v}}{\sum_{u=1}^{2}\underline{f}^{u}+\sum_{v=3}^{4}\overline{f}^{v}}=\frac{\underline{f}^{1}y_{r2}^{1}+\underline{f}^{2}y_{r2}^{2}+\overline{f}^{3}y_{r2}^{3}+\overline{f}^{4}y_{r2}^{4}}{\underline{f}^{1}+\underline{f}^{2}+\overline{f}^{3}+\overline{f}^{4}}=\frac{0.035\times-56.8+0.105\times-56.8+0.525\times85.4+0.6\times85.4}{0.035+0.045+0.595+0.68}=69.66$$

3.6.5 Defuzzification

From the type-reduction stage we have for each output *k* a type-reduced set, we defuzzify the interval set by calculating the average of y_{lk} *and* y_{rk} using Equation (40) for both outputs as follows:

the speed output $y_1' = \frac{y_{l1}+y_{r1}}{2} = \frac{0.452+1.37}{2}$ =0.911 m/s

And the steering output $y_2' = \frac{y_{l2}+y_{r2}}{2} = \frac{-79.01+69.66}{2}$ = -4.6751 %



Figure(6): Pictorial description of fuzzification and inference operations for a robot type-2 FLC.

4. Type-2 Hierarchical Fuzzy Logic Controllers

4.1 Problems with Single Rule Base FLC

Reactive autonomous mobile robots must be capable of achieving multiple goals, whose priorities may change with time, and must react to dynamic events in unstructured environments using a large number of sensor inputs and actuator outputs [9]. It is also important for a reactive robot controller to map sensors inputs to control signals quickly to allow for a good real time performance [9], where real time is defined as producing a result (control action) in time enough to be useful [6] (50 ms in our case using our robots).

It is difficult if not impossible to achieve this reactive performance for mobile robots using a single rule base FLC without sacrificing the real time performance as FLCs suffer from the serious limitation that the number of rules increases *exponentially* with the number of variables involved [9]. This limitation is termed *the rule explosion problem* where for n input variables and m fuzzy sets (type-1 or type-2) defined for each variable, we need m^n rules to construct a complete fuzzy rule base [31,36]. As n and m increase, the rule base gets larger which will result in the difficulty in designing the FLC as it will be very difficult to determine and manage a large number of rules, also the big rule base will quickly overload the memory of the robot [36]. The increase in the rule base size has a direct influence on the real time performance of the robot, as during any given control cycle all the rules in the fuzzy rule base are sequentially processed, so the computation for a large rule base will be intense thus delaying the control response for a given input vector. For our robots (indoor and outdoor) we have eight input sensors (7 sonar sensors and one goal detection sensor), if we represented each input by only three fuzzy sets (linguistic labels) then for a single rule base we need to determine $3^8=6561$ rules, which is very difficult to design or learn, also this huge rule base translates directly to slower controller response.

Interval type-2 FLCs have another problem related to the increase of the rule base size, as computing the type-reduced fuzzy set is directly proportional to the number of rules and for a large rule base it will take a long time for each control cycle to converge to the type-reduced set thus limiting the real time application of the type-2 FLC [38].

From the above discussion we conclude that there is a need develop an architecture to deal with the problems associated with the *rule explosion problem* in single rule base type-2 fuzzy controllers.

4.2 Hierarchical Fuzzy Systems

To cope with the problem of rule explosion in FLC 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 behaviours, each of which responds to specific types of situation, and then integrating the recommendations of these behaviours via a high level coordination layer [34,35]. The hierarchical structure of a fuzzy controller results from the desire to achieve a system goal for a complex process using a "divide and conquer" strategy, where the control design goal(s) of a robot are decomposed into several sub-goals handled by FLC behaviours. This is possible by partitioning the input space into a finite number of regions where appropriate control actions in each region is executed by a FLC behaviour to achieve a sub-goal. Each behaviour is an independent and self contained FLC with a small number of inputs and a small number of outputs and it serves a single purpose (e.g. edge-following or obstacle-avoidance) while operating in a reactive fashion. The behaviours will typically (but not necessarily) map different inputs sensors to common output actuators [27]. Such primitive behaviours are building blocks for more intelligent composite behaviours, i.e. their capabilities can be combined through synergistic coordination by a high level coordination layer to produce composite behaviours [4,27,29,32,34,35]. Behaviour based approaches to mobile robot control have gained increasing popularity being supported by considerations arising from the study of animal behaviour [1].

In a two level fuzzy hierarchical system, there are low-level behaviours and a high-level coordination layer. For each low level behaviour b with n_b number of inputs, if each input is represented by m_b linguistic labels (fuzzy sets) then the total number of rules in the rule base of this behaviour is $rules_b = {m_b}^{n_b}$ which is a small number as the number of inputs for each behaviour is a small number.

The high-level coordination layer will have a coordination rule for each behaviour expressing when the behaviour should be activated. As the number of inputs increases we increase the number of behaviours where each behaviour will handle a small number of inputs. For H number of low level behaviours used in the hierarchical system, we will have H coordination rules in the high level coordination layer and the total number of rules in the low level behaviours denoted by $rules_H$ can be calculated as follows:

$$rules_H = \sum_{b=1}^{H} {m_b}^{n_b} \tag{42}$$

From this we see that the hierarchical fuzzy systems have a beneficial property that the total number of rules increases linearly rather than exponentially as in the single rule base FLC [31,36]. For example the robot controller can be divided into four co-operating behaviours, obstacle avoidance, left and right edge following and goal seeking. These behaviours will enable the robot to navigate safely to its goals in unstructured environments [27,28,29]. Each individual behaviour will only require a subset of the total number of available inputs. If ,as in the case of a single rule base FLC, we represent each input using three fuzzy sets then the obstacle avoidance behaviour, using three forward facing sonar sensors, will have a rule base of 3^3 = 27 rules. The left edge following behaviour, using two left side facing sonar sensors, will have a rule base of 3^2 = 9 rules, the right edge following behaviour will have the same number of rules. The goal seeking behaviour, using a single goal detection sensor (more accurately represented by seven fuzzy sets) will have rule base of 7 rules. Thus the total number of rules in the low behaviours is 27 + 9 + 9 + 7 = 52 rules and the total number of rules in the coordination layer is 4 (number of behaviours) 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. Also having a small number of rules in each FLC behaviour will result in fast real time response.

To use such a behaviour based hierarchical system, a co-ordination scheme is required to combine the several independent behaviours to obtain the overall coherent behaviour that achieves the intended task(s). There are two problems associated with behaviour coordination (i) How to decide which behaviour(s) should be activated at each moment, (ii) How to combine the results from different behaviours into one command to be sent to the robot's effectors, enabling the controller to consider multiple concurrent requirements [33].

In behaviour combination we need to decide which behaviour (s) should be activated in each situation, so there is a need for an arbitration policy that determine which behaviour(s) should influence the operation of the robot at each moment. Early solutions such as the subsumption architecture proposed by Brooks [5] relied on a fixed arbitration policy, hardwired into a network of suppression and inhibition links. This rigid organisation contrasts with the requirement that an autonomous robot should be capable of being programmed to perform a variety of different tasks in a variety of different environments. Other architectures used dynamic arbitration schemas where the decision of which behaviour to activate relied on both the current plan and the environment contingencies. Many of these proposals do not allow for the concurrent execution of behaviours [2,34].

Both fixed and dynamic arbitration policies can be implemented using fuzzy logic as it allows the ability to express partial and concurrent activations of behaviours; and the smooth transition between behaviours. Fuzzy Logic provides useful tools to how to coordinate the concurrent execution of several behaviours aimed at the achievement of different, possibly conflicting objectives. There are many papers reporting implementations of type-1 Hierarchical Fuzzy Logic Controllers (HFLCs) which produced good results [4,26,27,29,34,35]. However as type-1 HFLC are composed from type-1 FLC behaviours then the total coordinated system suffers from the uncertainty problems associated with type-1 fuzzy systems operating in changing unstructured environments. The type-1 fuzzy controllers can deal with a degree of imprecision present in the robot environment, however substantial environmental and robot changes cannot be accommodated (as type-1 controllers use precise fuzzy sets) which will prevent the controller from operating correctly. One solution to this problem is to provide a new controller for each of the different environments, although prior knowledge of the different environments would be needed, which is difficult if not impossible for dynamic environments and especially the outdoor environments which are a severe test for the robots. An alternative solution which we tried before in our previous research is to learn online a type-1 HFLC [25,26,27,28,29] which is sub-optimal under certain environmental and robot kinematics conditions, but if the environment or the robot kinematics changed, which can happen easily in outdoor environments, the type-1 HFLC would not be sub-optimal. So we need to allow the HFLC to be adapted to compensate for the robot and environmental differences [27]. This means that the system must be in a continuous learning mode, to compensate for the various uncertainties and imprecision available in changing and dynamic unstructured environments, which will consume much time in the learning and adaptation cycles and might disturb the real time operation of the robot. Another solution which we introduce in this paper is to use type-2 FLC to implement the basic behaviours and coordination between these behaviours to produce a type-2 HFLC. This type-2 HFLC will use type-2 fuzzy sets which can model and handle the uncertainties in the input signals and the control outputs which are caused by the environmental or robot changes. Of course the type-2 system will handle uncertainties only defined within its footprint of uncertainty of the various type-2 fuzzy sets.

4.3 The Architecture of the Type-2 Hierarchical Fuzzy Logic Controller

4.3.1 The Low Level Type-2 FLC Based Behaviours

In our proposed type-2 Hierarchical Fuzzy Logic Controller (HFLC), each low level reactive behaviour will be an interval type-2 FLC using type-2 fuzzy sets to represent the input and output variables of each behaviour as explained in Section (3). Each low level behaviour will receive a subset x_h of the total inputs x available to the type-2 HFLC, all behaviours will produce *preferences* to the same common outputs, which are the outputs of the HFLC, so each behaviour will map different inputs sensors to common outputs. The low level type-2 FLCs will be the same as Section (3) except that there is no defuzzification block and the outputs from each low level behaviour will be a type-reduced sets to represent the *preferences* from the perspective of the behaviour goals, this will be explained in subsection (4.3.2).

In type-1 FLC the use of 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 give a much smoother control surface and a better response [12]. However as mentioned in subsection (4.1) an increased number of type-1 fuzzy sets will increase the size of the rule base which will cause real time performance problems for the controller besides the problems associated with the design of such controllers. Using interval type-2 fuzzy sets will lead to using less number of linguistic labels than using type-1 fuzzy sets as linguistic uncertainty represented in the footprint of uncertainty lets us cover the same domain with a smaller number of linguistic labels [22]. Besides using smaller number of linguistic variables, each input and output will be represented by a big number of type-1 fuzzy sets (uncountable for continuous universes of discourse) which are embedded in the type-2 fuzzy sets [22,23]. An embedded type-1 FLC for an interval type-2 FLC is associated with the embedded type-1 fuzzy sets of the antecedents and consequents, therefore a type-2 FLC can be interpreted as a collection of embedded type-1 FLCs [38].

From the above discussion we can see that using a type-2 FLC to represent the low level behaviours for a mobile robot will be better than using type-1 FLC, as a type-2 FLC will have a smaller rule base as it uses type-2 fuzzy sets which use a smaller number of linguistic labels to represent the input and output variables. Moreover type-2 FLC will have the chance to over perform the type-1 FLC as each input and output variable will be represented by a huge number of embedded type-1 fuzzy sets which will allow for the detailed description of the analytical control and thus a much smoother control surface and better response.

However type-2 FLC will have the problem that as the rule base and the number of inputs to the FLC gets larger, the computation for the type-reduction represents a major bottleneck to the use of interval type-2 FLC in real time applications [38]. We will solve this problem by using simple type-2 FLCs to implement the basic low level behaviours, these FLCs will have a small number of inputs and outputs and small rule bases. We will use a high-level coordination layer to coordinate the actions of these low level type-2 behaviours to achieve the different goals of the mobile robot.

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

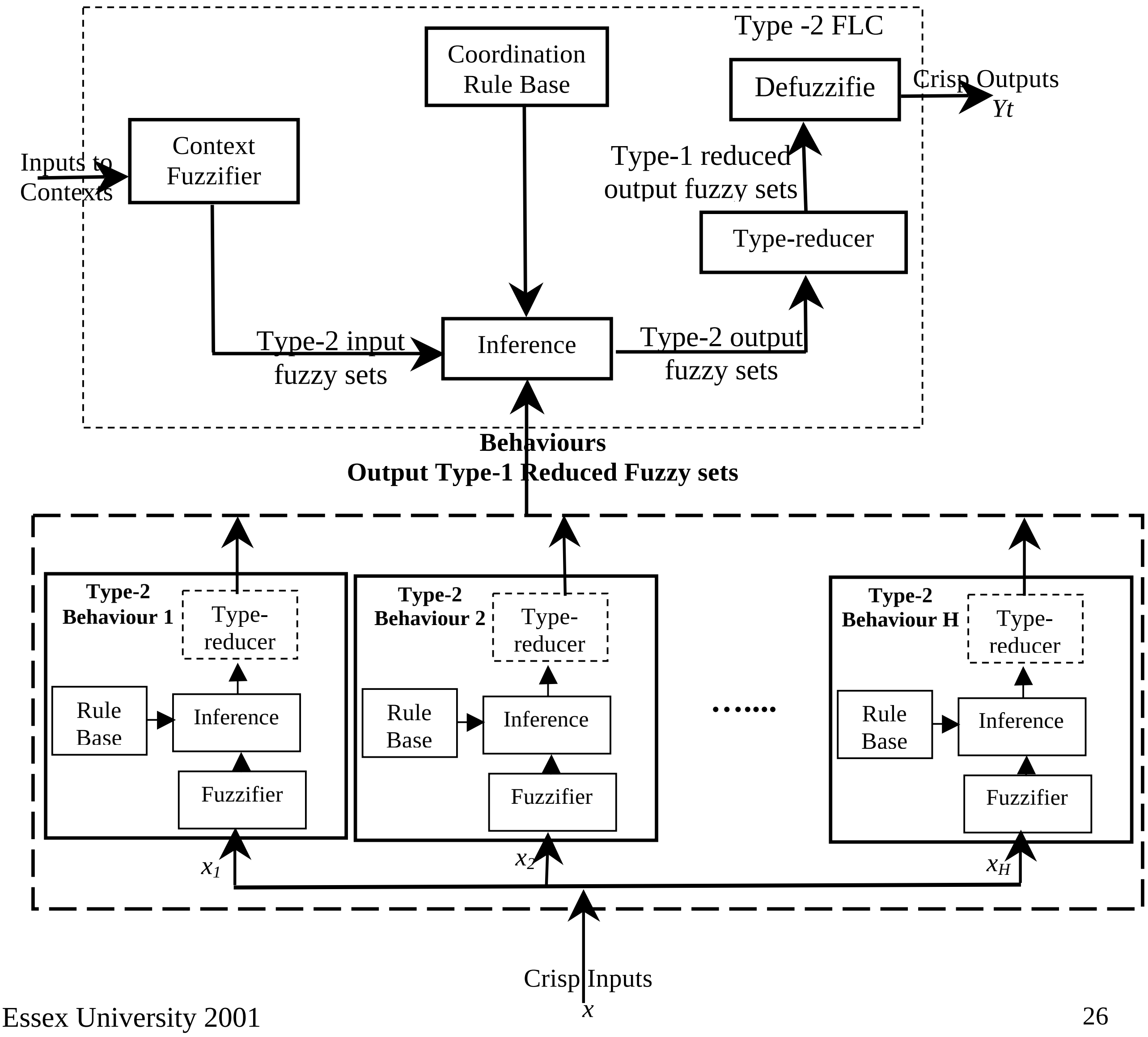


Figure (7): The type-2 Hierarchical Fuzzy Logic Controller architecture.

4.3.2 The High -evel type-2 Coordination Layer

Figure (7) shows the type-2 Hierarchical Fuzzy Controller (HFLC) architecture, at the high level there is a type-2 FLC which is responsible for the coordination of the low level type-2 FLC based behaviours. Each low level behaviour $\widetilde{B}_j$ has a *context* of activation $\widetilde{C}_j$ representing when it should be activated. The *context* is represented by an interval type-2 membership as type-2 membership function can handle the linguistic and numerical uncertainties associated with these *contexts*. Again interval type-2 fuzzy sets were chosen as using interval secondary membership functions reflect a uniform uncertainty at the primary membership [22] and they are very useful when we have no other knowledge about secondary memberships [17], which is our case. Also using interval sets will simplify the calculations in type-2 systems thus allowing real time operation [18]. Each behaviour 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,..H)*, where *H* is the total number of behaviours. These crisp inputs are then fuzzified by matching each input to its *context* interval type-2 membership function. As in the case of type-2 FLC we have chosen singleton fuzzification to minimise the computations and to allow for a real time performance. When the crisp inputs are fuzzified using the interval type-2 *context* membership functions we obtain for each crisp input d_j a lower membership $\underline{\mu}_{\widetilde{C}_j}(d_j)$ and an upper membership $\overline{\mu}_{\widetilde{C}_j}(d_j)$ of 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 the arbitration policy, i.e. how each behaviour can influence the operation of the robot at each moment, the coordination rules take the following format

$$\textit{IF Context is } \widetilde{C}_j \textit{ THEN Behaviour is } \widetilde{B}_j \qquad j=1,..H \qquad (43)$$

Where *H* is the total number of behaviours. Note that we have a coordination rule for each behaviour, thus we have a total of *H* coordination rules

From Equation (44) we see that that each behaviour $\widetilde{B}_j$ 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 product t-norm for the meet operation and the maximum for the join operation and extended sup-star composition. 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-reduction which have problems when only one rule fires [22]. The center of sets type-reduction computation 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 behaviours *H* which is a small number maximum 15-20 behaviours. However to use the center of sets type- reduction we need to compute the centroid of the type-2 fuzzy output of each low level behaviour $\widetilde{B}_j$ which must be pre-computed before the computation of the type-reduced set of the HFLC. However if for each behaviour we calculate the centroid of the output type-2 fuzzy set which is proportional to the number of discrete levels *N* (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 type-2 fuzzy output of each behaviour is to calculate the center of sets type-reduced set for each behaviour. As the center of sets type-reduction like any type-reducer combines all type-2 rules outputs sets and then performs a centroid calculation which leads to a type-1 fuzzy set [22]. The computation of the center of sets type-reduced set for each low level behaviour is proportional to the number of rules in each behaviour which is a small number as mentioned in Section(4.3.1). So approximating the centroid of the type-2 output fuzzy set of each behaviour by the center of sets type-reduced set will result in faster computation which will result in real time performance.

Each low level type-2 FLC based behaviour will receive a subset x_h of the total inputs x 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 type-2 output of each behaviour. Where $j=1,..H$, and H is the total number of behaviours. y_{lk}^j and y_{rk}^j represent the left and right end points for the type-reduced set for each behaviour j and for each common output k where $k=1, \ldots c$ and c is the total number of outputs of the HFLC. y_{lk}^j and y_{rk}^j can be obtained from Equations(39) and (40) by using the iterative procedure in Figure (5) as was explained in subsection (3.4.2).

The final output k $(k=1, \ldots c)$ of the HFLC Yt_k 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 Figure (5) and using the same manner for calculating the type-reduced set as in Section (3.4.2)

$$yt_{lk} = \frac{\sum_{j=1}^{L} y_{lk}^j \overline{\mu}_{\widetilde{C}_j} + \sum_{j=L+1}^{H} y_{lk}^j \underline{\mu}_{\widetilde{C}_j}}{\sum_{j=1}^{L} \overline{\mu}_{\widetilde{C}_j} + \sum_{j=L+1}^{H} \underline{\mu}_{\widetilde{C}_j}} \tag{44}$$

$$yt_{rk} = \frac{\sum_{j=1}^{R} y_{rk}^j \underline{\mu}_{\widetilde{C}_j} + \sum_{j=R+1}^{H} y_{rk}^j \overline{\mu}_{\widetilde{C}_j}}{\sum_{j=1}^{L} \underline{\mu}_{\widetilde{C}_j} + \sum_{j=L+1}^{H} \overline{\mu}_{\widetilde{C}_j}} \tag{45}$$

Where H is the total number of behaviours and L and R can be calculated using the procedure in Figure (5) in subsection (3.4.2). Note that the iterative procedure will converge in no more than H iterations to find R and H iterations to find L. The final crisp output Yt_k for output k can be found by finding the average of yt_{lk} , yt_{rk} as follows:

$$Yt_k = \frac{yt_{lk} + yt_{rk}}{2} \tag{46}$$

The same procedure can be repeated for all the c outputs.

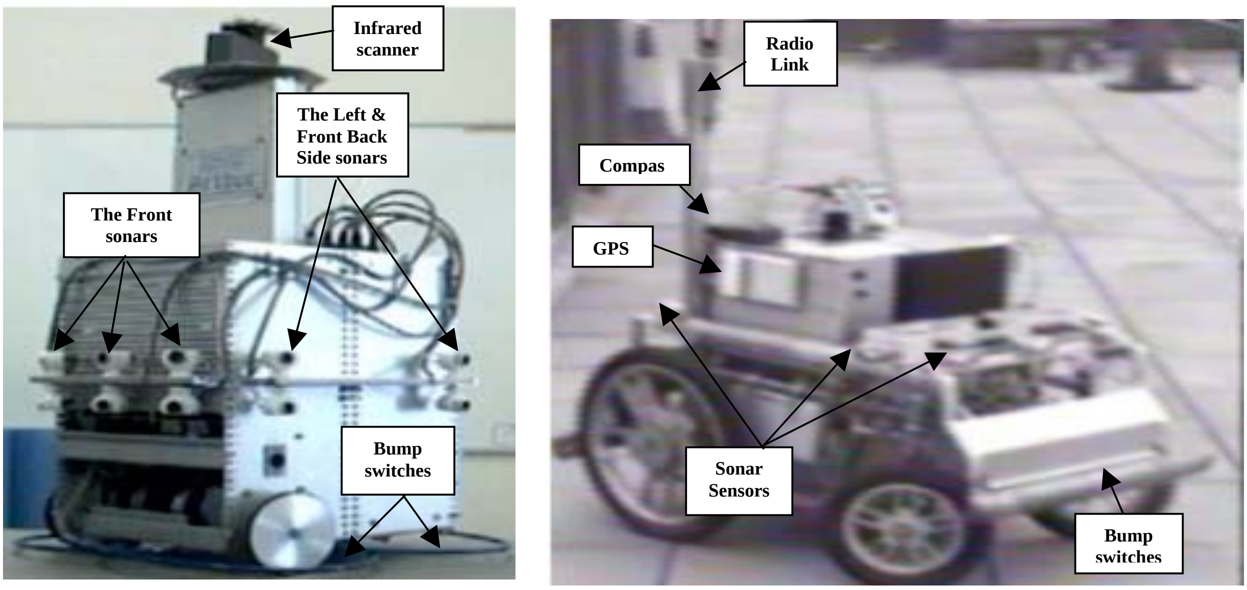
This hierarchical structure can be iterated at different levels when the number of behaviours grows larger as there can be another high level type-2 FLC with coordination rules to coordinate many of the type-2 HFLC in Figure (7). The output of each type-2 HFLC will be in the form of a type-reduced sets, always leaving the defuzzification to the highest level FLC. This hierarchical organisation level can be iterated through many other levels, it can therefore address the scalability issue in fuzzy system [33]

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

In this section we introduce the application of the proposed interval type-2 hierarchical fuzzy logic control architecture to robot navigation in indoor and outdoor unstructured environments. We have used indoor and outdoor robots with different sensors, actuators, geometrical shapes, dynamics and kinematics. We used different robots to verify that our type-2 HFLC is general and robot independent. In the next subsection we will introduce the robots used in our experiments.

5.1 Mobile Robots Descriptions

The indoor robots shown in Figure (8-a) have a ring of 7 ultrasonic proximity detectors (with a covering cone of 30°), an 8-axis vectored bump switch, an infra-red scanner sensor and two independent stepper motors for driving plus differential steering. The hardware was based on embedded Motorola 68040 (40 MHz speed) processors on a VME bus running the VxWorks Real Time Operating System (RTOS). In the initial experiments infrared beacons were used to simulate the goals in indoor environments. Control programs were developed using the Tornado environment and then downloaded via an ethernet cable to the robots, after which the cable was disconnected and the navigation was autonomous.



(a) (b)

Figure (8): a) The indoor robot and its sensor configuration. b) The outdoor robot and its sensor configuration.

The outdoor robots shown in Figure (8-b) are based around a distributed field bus control system, in particular, we use the CANbus (Controller Area Network) developed for automotive industry. The outdoor robots have a ring of 7 ultrasonic proximity detectors (with a covering cone of 30°). The robots have two bump switches, one is at the front and other is at the back of the robot. The robots are also supplied with GPS and a compass for goal determination in outdoor environments. The outdoor robots have two different motors one for controlling the speed of the front wheels and the other for controlling the steering of the front wheel. For our outdoor robots, positive steering values means steering to the right while negative steering values means steering to the left, the steering values are in percentage where 100 % indicates turning fully to the right and -100 % indicates turning fully to the left. We tried to give all our robots a similar architecture (to simplify development work) so the outdoor robots hardware is also based on embedded Motorola 68040 processors running VxWorks RTOS. 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.

As explained in Section (4.1.1) both the indoor and outdoor mobile robots will have four basic low level type-2 FLC based behaviours which are obstacle avoidance, left edge following, right edge following and goal seeking, these basic behaviours will allow the robots to navigate safely in indoor and outdoor unstructured environments [27,28]. The total available inputs to the interval type-2 HFLC is 8 inputs which are 7 sonar sensors plus a goal detection sensor (infra red scanner in case of indoor robots and GPS/compass in case of outdoor robots). The HFLC and all the behaviours have common two outputs (the left and right wheel speeds for the indoor robots and the wheel speeds and steering for the outdoor robots).

Each low level behaviour type-2 FLC will receive a subset of the total inputs available to the HFLC and will produce its output *preferences* (in the form of a type-reduced set) for the two outputs, these *preferences* will be fed to the high level coordination type-2 FLC. In the following subsection we will introduce the various layers of the type-2 HFLC, where in subsection (5.2) we will present in detail the low level type-2 behaviours and in subsection (5.3) we will present high level coordination type-2 FLC.

5.2 The Low-Level Behaviours

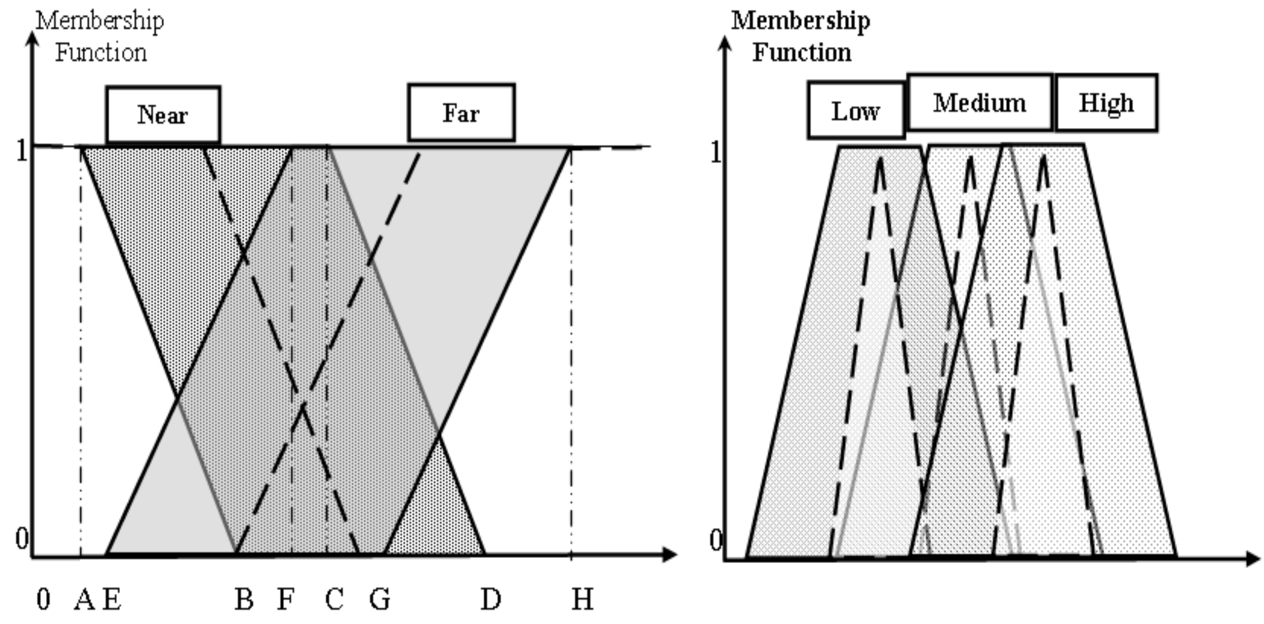
5.2.1 Right and Left Edge Following Type-2 FLC Based Low Level Behaviours

Edge following behaviours are used to follow edges at a desired distance. In our system we will use two edge following behaviours which are left edge following and right edge following behaviours to follow edges on the left and right side of the robot.

The right edge following behaviour type-2 FLC in both the indoor and outdoor robots receives only two inputs from the two right side sonar sensors defined as Right Side Front (RSF) and Right Side Back (RSB). The left edge following behaviour type-2 FLC in both the indoor and outdoor robots receives only two 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 behaviours 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 Figure

(9-a). The membership function in thick dashed line shown in Figure (9-a) represent type-1 fuzzy set that might be used by the robot to follow the edge at the desired distance. As explained in Section (1) type-1 fuzzy sets can not represent and deal with the linguistic and numerical uncertainties associated with the changing of the environment or robot characteristics. Type-2 fuzzy sets uses a footprint of uncertainty which is shaded in grey in Figure (9-a) to deal with the uncertainties associated with type-1 fuzzy sets, the footprint of uncertainty covers the ranges of the possible values of the type-1 fuzzy sets for various environments and robot characteristics. In interval type-2 membership functions, the footprint of uncertainty are obtained by specifying the upper and lower membership functions for each type-2 fuzzy set [17]. We have determined the lower and upper membership functions for the fuzzy sets *Near* and *Far* by finding the lower and upper type-1 membership functions that the robot needed to follow the edge at the desired distance in different environmental and robot configurations, this was aided by using our online learning method developed in our previous work to learn the type-1 membership functions [25,27]. For the *Near* interval type-2 fuzzy set the points for the lower membership function are A, B and the points for the upper membership function are C, D. For the *FAR* interval type-2 fuzzy set the points for the lower membership function are E, F and the points for the upper membership function are G, H.

For the indoor robots the right and left edge following behaviours like all the other behaviours 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 type-2 fuzzy sets which are *Low*, *Medium and High.* which are shown in Figure (9-b). The triangular membership functions with thick dashed line in Figure (9-b) are possible type-1 membership functions, we use type-2 membership function to handle and model the uncertainties associated with determining the output membership functions.

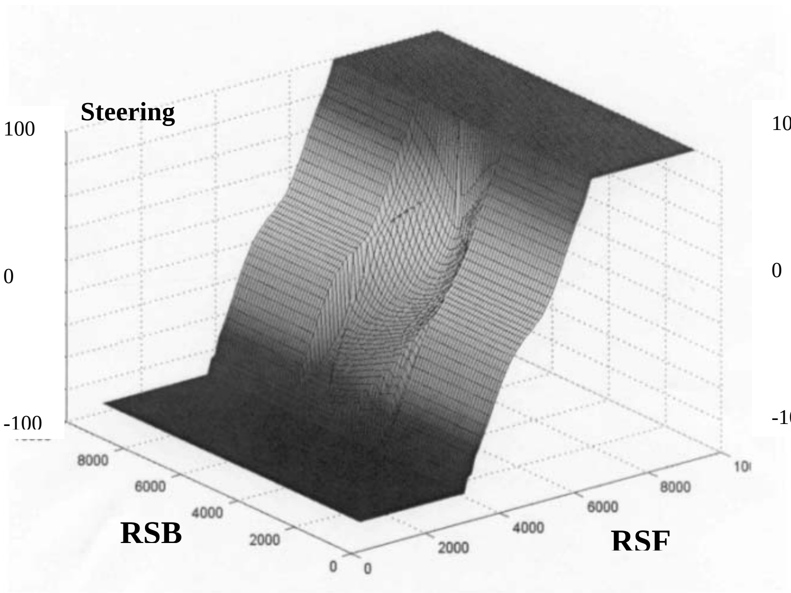


(a) (b)

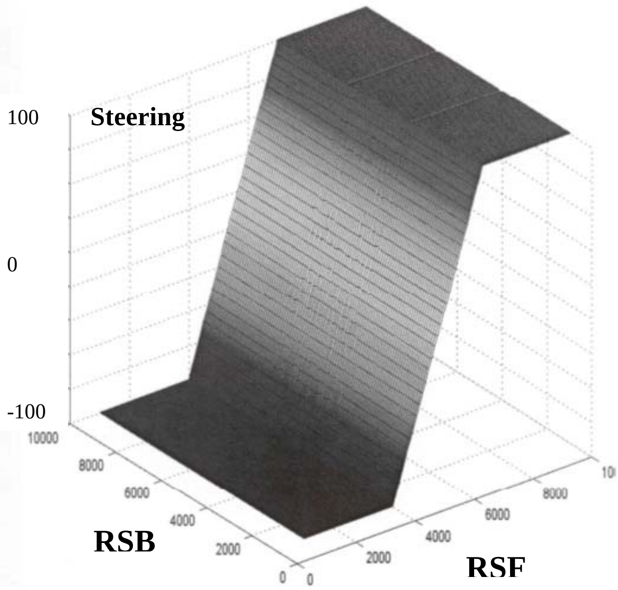
Figure (9): a) The input type-2 membership function for the input sensors for the edge following behaviours and the obstacle avoidance behaviour in indoor and outdoor robots. b) Output membership function for the velocities in the indoor robots.

For the outdoor robots, the right and left edge following behaviours like all the other behaviours will have two outputs *preferences*, one for the front wheels speed and one for the steering. The speed will be represented by three interval type-2 output fuzzy sets which *Slow, Medium, Fast*. The steering will be represented by two interval type-2 fuzzy sets which are *Right and Left.*

For the indoor and outdoor robots as we have used two inputs for each edge following behaviour and we represented each input by two fuzzy sets therefore we have a rule base of 2^2= 4 rules. Table (1) had shown the rule base for the right edge following behaviour for the outdoor robots, its complement is the rule base for the left edge following behaviour. 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 out-perform a type-1 FLC using a larger rule base. This is because a type-2 FLC can be interpreted as a collection of embedded type-1 FLCs [38], this can be shown in the control surface which graphically represent the unknown function articulated by the rules [16]. Figure (10-a) shows the control surface for the type-2 FLC steering output of the outdoor robot implementing the right edge following behaviour using only two fuzzy sets to represent each input and thus a rule base of 4 rules. This control surface represents the experiments shown in Figure (17-a) 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 transfers to a smooth control response that can deal with noise and imprecision. Note that the values for the RSF and RSB are the raw sonar values which are then transformed to the physical distance from the edge.

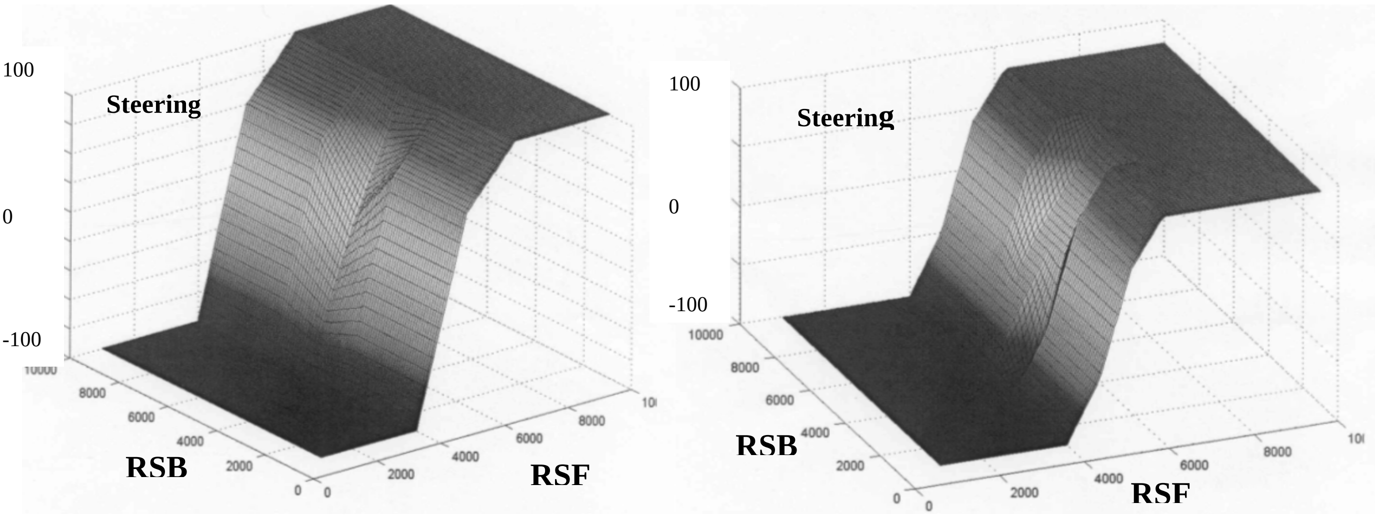


(a)



(b)

Figure (10): a) The Control Surface of a type-2 FLC for the right edge following behaviour. b) The Control Surface of a type-1 FLC using 2 fuzzy sets to represent each input.



(a) (b)

Figure (11): a) The Control Surface of a type-1 FLC using 3 fuzzy sets to represent each input.
b) The Control Surface of a type-1 FLC using 5 fuzzy sets to represent each input.

Figure (10-b) shows the control surface for a type-1 FLC for the right edge following behaviour using also two type-1 fuzzy sets to represent each input, Figure (11-a) shows the control surface for a type-1 FLC using three type-1 fuzzy sets to represent each input, Figure (11-b) shows the control surface for a type-1 FLC using five type-1 fuzzy sets to represent each input. Note that the more fuzzy sets used in type-1 FLC the more its response approaches that of the type-2 FLC. The same results were obtained for all the other behaviours in indoor and outdoor robots. This justifies the point that type-2 has the potential to over perform the type-1 FLC as its type-2 fuzzy sets conations a big number of type-1 embedded sets which enables the type-2 FLC to allow for the detailed description of the analytical control surface as the addition of the extra levels of classification give a much smoother control surface and response, this will be further demonstrated in the experiments section.

5.2.2 Obstacle Avoidance Type-2 FLC Based Low Level Behaviour

The obstacle avoidance behaviour is needed for safe navigation in unstructured environments and is needed to avoid static and dynamic objects at a safe avoiding distance. In this behaviour in both the indoor and outdoor robots we will use 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 Figure (9-a).

For the indoor robots each output velocity will be represented by three type-2 fuzzy sets which are *Low, Medium and High.* which are shown in Figure (9-b).

For the outdoor robots, the speed will be represented by three interval type-2 output fuzzy sets which are *Slow, Medium, Fast*. The steering will be represented by three interval type-2 fuzzy sets which are *Right, Zero and Left*

For the indoor and outdoor robots as we have used three inputs for the obstacle avoidance behaviour and we represented each input by two fuzzy sets therefore we have a rule base of 2^3= 8 rules.

5.2.3 Goal seeking Type-2 FLC Based Low Level Behaviour

The goal seeking behaviour 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 infra red beacon and the bearing from it is measured by an infra red 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 fuzzy sets which are *Negative, Zero and Positive* as shown in Figure (12-a).

For the indoor robots each output velocity will be represented by three type-2 fuzzy sets which are *Low, Medium and High.* which are shown in Figure (9-b). For the outdoor robots, the speed will be represented by three interval type-2 output fuzzy sets which *Slow, Medium, Fast*. The steering will be represented by three interval type-2 fuzzy sets which are *Right, Zero and Left*

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.

5.3 The High Level Type-2 FLC coordination layer

As mentioned in Section (4.3.2) the high level type-2 FLC receives the crisp inputs to the *contexts* d_j $(j=1,..H)$, where H *is* the total number of behaviours, in our case $H=4$. Each *context* is attached to a behaviour defining when the behaviour should be activated. Each *context* $\widetilde{C}_j$ is a type-2 fuzzy *context* represented by an interval type-2 fuzzy set to deal with the uncertainties associated with the changing of the environment or robot characteristics.

The crisp input d_1 to the obstacle avoidance behaviour context $\widetilde{C}_1$ in both the indoor and outdoor robots is the minimum distance of the front sensors. The context $\widetilde{C}_1$ for the obstacle avoidance behaviour is in the form of a type-2 interval fuzzy set as shown in Figure (12-b) which defines that the obstacle avoidance behaviour 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 behaviour. d_1 was chosen to be the minimum distance of the front sensors so that the obstacle avoidance behaviour 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 $\overline{\mu}_{\widetilde{C}_1}(d_1)$.

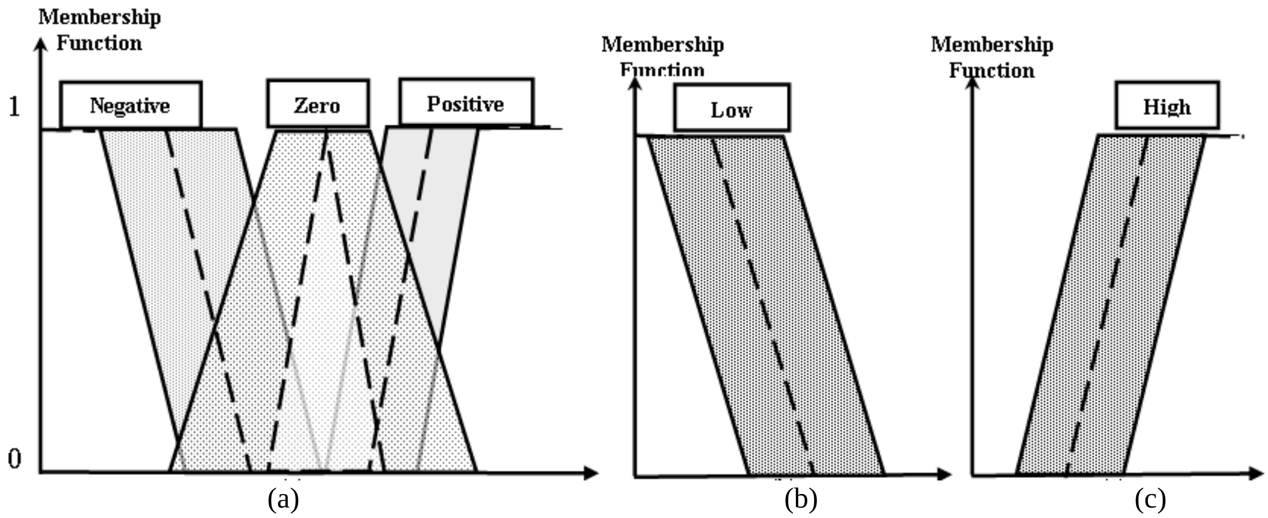


Figure (12): a) The type-2 input membership function for the goal seeking behaviour in the indoor and outdoor robots. b) The type-2 context fuzzy set for the obstacle avoidance behaviour and the right and left edge following behaviours. c) The type-2 context fuzzy set for the goal seeking behaviour.

The crisp input d_2 to the left edge following behaviour context $\widetilde{C}_2$ in both the indoor and outdoor robots is the minimum distance of the left side sensors. The context $\widetilde{C}_2$ for the obstacle avoidance behaviour is in the form of a type-2 interval fuzzy set as shown in Figure (12-b) which defines that the left edge following behaviour 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 behaviour. 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 behaviour context $\widetilde{C}_3$ in both the indoor and outdoor robots is the minimum distance of the right side sensors. The context $\widetilde{C}_3$ for the obstacle avoidance behaviour is in the form of a type-2 interval fuzzy set as shown in Figure (12-b) which defines that the right edge following behaviour should be active when an edge is detected to the right side of the robot, the closer the robot gets to the edge on its right, the higher will be the activation of the right edge following behaviour. d_3 is fuzzified using the interval type-2 *context* membership functions of $\widetilde{C}_3$ to a lower membership $\underline{\mu}_{\widetilde{C}_3}(d_3)$ and an upper membership $\overline{\mu}_{\widetilde{C}_3}(d_3)$.

The crisp input d_4 to the goal seeking behaviour 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 behaviour is in the form of a type-2 interval fuzzy set as shown in Figure (10-c) which defines that the goal seeking behaviour 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 behaviour. 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,..4) are* matched and fuzzified against their type-2 fuzzy contexts $\widetilde{C}_j$, the fuzzified values are then fed to the inference engine which determines which rules are fired from the coordination rule base.

The coordination rule base will contain a coordination rule for each behaviour which relates the *contexts* to the behaviours to decide which behaviour(s) should be activated at each moment. As we have four behaviours 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

After inference engine, the type-reduction block receives the output type-reduced sets from the low level behaviours which represent the *preference* from the perspective of the goal of the behaviours. For indoor and outdoor robots the total number of outputs $c = 2$ which are the left or right wheel velocities for indoor robots and the speed and steering for the outdoor robots. For each output $k = 1,2$ we have the type reduced set $[y_{lk}^j, y_{rk}^j]$ which approximates the centroid of the behaviour type-2 output set. To obtain the type-reduced set for each output *k* we then apply the iterative procedure in Figure (5) to obtain *L* and *R* and then we substitute in Equation (44) and (45) to find the left most point yt_{lk} and the right most point yt_{rk}. To obtain the total crisp output to be sent the robot actuators we find the average of the type-reduced set by substituting in Equation (46) to obtain the final crisp output Yt_k.

The computation time for this hierarchical structure is small as for the high level type-2 FLC we will need maximum 4 iterations (number of coordination rules) to find *R* and 4 iterations to find *L*, thus we need maximum 8 iterations to find the high level type-reduced set. Also we need to perform the type-reduction for the low level behaviours 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 behaviour contain 8 rules, therefore we will need maximum 16 iterations to find *L* and *R* for the low level behaviours. Therefore the maximum number of iterations to find a type-reduced set and thus an output at the end of the control cycle is 8+16 = 24 iterations which is a small number of iterations that will enable real time operation even with our robots using slow processors.

From the above discussion we can realise that our hierarchical architecture has the following benefits when applied to mobile robot control:

- It 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.
- It harnesses the benefits of type-2 fuzzy logic to deal with the large amounts of imprecision and uncertainty present in and the highly dynamic unstructured environments.
- The robots can achieve multiple goals, whose priorities may change with time, as the behaviour *preferences* outputs and the *context* truth values (weight) are dynamic, taking into account the situation of the mobile robot. This characteristic is required for reactive navigation as mentioned in subsection (4.1).

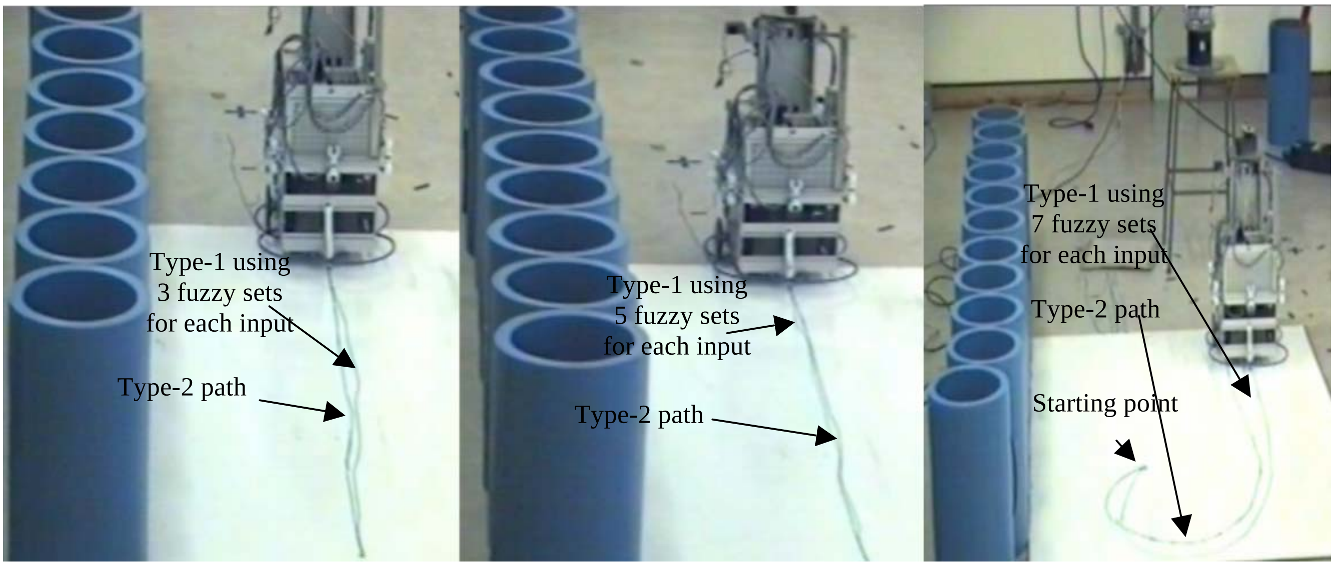
- Our proposed system uses type-2 fuzzy logic for the co-ordination between the different behaviours, which deals with the uncertainty in the coordination level. The type-2 fuzzy coordination provides a smooth transition between behaviours with a consequent smooth output response, which allows more than one behaviour to be active to differing degrees thereby avoiding the drawbacks of on-off switching schema (i.e. dealing with situations where several criteria need to be taken into account). In addition using fuzzy co-ordination provides a smooth transition between behaviours with a consequent smooth output response as will be shown in the experiments section.
- This hierarchical structure offers a flexible structure where new behaviours can be added or modified easily. The system is capable of performing very different tasks using identical behaviours by changing only the context rules and co-ordination parameters to satisfy a different high-level objective. We can eliminate the unneeded behaviours from the context rules according to the robot's mission. For example, in a situation involving reaching a goal by left edge following, there is no need for right edge following and this behaviour together with its context and context rule can be deleted.

In the next section we will introduce our 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.

6. 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 8 experiments, where we used different geometrical structures and started the robot from different random locations. We also experimented with different robots having the same shape, sensor configurations and computational power but with different kinematics as some robots are using stepper motors and others are using DC motors. We will start by experiments aimed at testing the individual type-2 FLC based behaviours where the robot will have one objective related to the behaviour goal and the outputs from the type-reduction will be defuzzified and fed to the actuators

Figure (13-a) shows the indoor robot implementing the type-2 FLC based left edge following behaviour 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 4 rules as explained in subsection (5.2.1). 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. Figure (13-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)

Figure (13): a) Indoor robot path using type-2 FLC to implement the left edge following behaviour compared against type-1 FLC path using 3 fuzzy sets to represent each input. b) The type-2 FLC path compared against type-1 FLC using 5 fuzzy sets to represent each input. c) The robot path using type-2 FLC to implement the goal seeking behaviours compared against a type-1 FLC using 7 fuzzy sets to represent the input.

We have compared the type-2 FLC response with type-1 FLC using bigger number of fuzzy sets and thus a larger rule base. All the type-1 FLC parameters were learnt online using our online hierarchical fuzzy genetic system [27,29] which learns the best type-1 FLC under certain environmental and robot kinematics conditions. If the environmental or the robot kinematics conditions changed these type-1 parameters would not be sub-optimal and the learning cycle would have to be repeated from the beginning as type-1 fuzzy sets use precise membership function which cannot model and handle uncertainties.

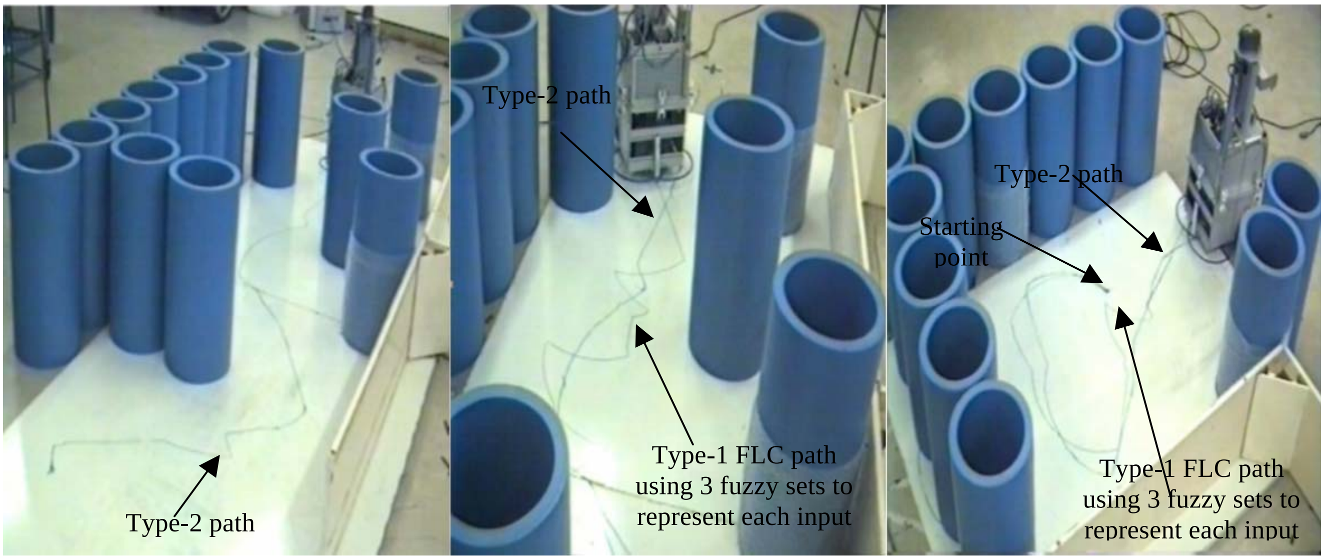
Figure (13-a) shows a type-1 FLC which uses three fuzzy sets to represent each input thus it has rule base of 9 rules, the type-1 FLC had produced a path of average deviation from the desired distance of 3.1 cm and a standard deviation of 1.4. We also experimented with another type-1 FLC with five fuzzy sets to represent each input and thus it has a rule base of 25 rules. As shown in Figure (13-b), 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 Figure(13-a), Figure(13-b) it is obvious that the type-2 FLC had over performed the performance of the type-1 FLC while using a smaller number fuzzy sets to describe each input and thus a smaller rule base. Also note that as the number of fuzzy sets in type-1 FLC increases the type-1 performance approaches the type-2 performance which agrees with the control surface results in Figure (10),(11) in section (5.2.1).

Figure (13-c) shows the indoor robot implementing the type-2 FLC based goal seeking behaviour, where the goal is represented by an infra red beacon and the main objective of this behaviour is to reach the goal and align with it with zero deviation. The type-2 FLC for the goal seeking behaviour has only one input represented by three type-2 fuzzy sets as explained in subsection (5.2.3) thus the type-2 FLC has a rule base of only three rules. Figure (13-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^o and a standard deviation of 0.7. We have tried type-1 FLCs using more number of fuzzy sets to represent each input and thus a bigger rule base. We tried a type-1 FLC using 5 fuzzy sets to represent the input and thus having a rule base of 5 rules, this type-1 FLC has given an average deviation of 12^o and a standard deviation of 2.4. Figure (13-c) shows a type-1 FLC using 7 fuzzy sets to represent the input and thus having a rule base of 7 rules, this type-1 FLC has given an average deviation of 8^o and a standard deviation of 1.6. Again note that the type-2 FLC had over performed the performance of the type-1 FLC while using a smaller number fuzzy sets to describe each input and thus a smaller rule base. Again as the number of fuzzy sets in type-1 FLC increases the type-1 performance approaches the type-2 performance.

Figure (14) shows the robot using a type-2 FLC to perform the obstacle avoidance behaviour, the robot goal is maintain a minimum safe distance of 45 cm from any front obstacle. The type-2 FLC for the obstacle behaviour has three inputs, each is represented by two type-2 fuzzy sets as explained in subsection (5.2.2) thus the FLC has a rule base of only eight rules. Figure (14-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 and thus a bigger rule base. Figure (14-b) shows a type-1 FLC using 3 fuzzy sets to represent the 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. Figure (14-c) shows another experiment where the robot is started facing a dead end and it has to find its way out, again we compared the responses of the type-2 and type-1 FLCs and from the Figure (14-c) its is obvious that the type-2 FLC had produced a better smooth response with less average and standard deviation. Again note that the type-2 FLC had over performed the performance of the type-1 FLC while using a small number of fuzzy sets to describe each input and thus a smaller rule base. Again as the number of fuzzy sets in type-1 FLC increases the type-1 performance approaches the type-2 performance.

In all the previous experiments the type-2 FLC response was repeatable when started from different starting positions or when tested with changing the geometrical setting such as trying different edges for the edge following behaviours .etc. The type-2 FLC had dealt in real time with dynamic changes in the environment. The FLC has also dealt with changes in the robots kinematics such as using similar robots but with different kinematics (using stepper motors rather than DC motors). The type-2 FLC was always able to give almost the same very good response as it can model and handle the uncertainty and impression.

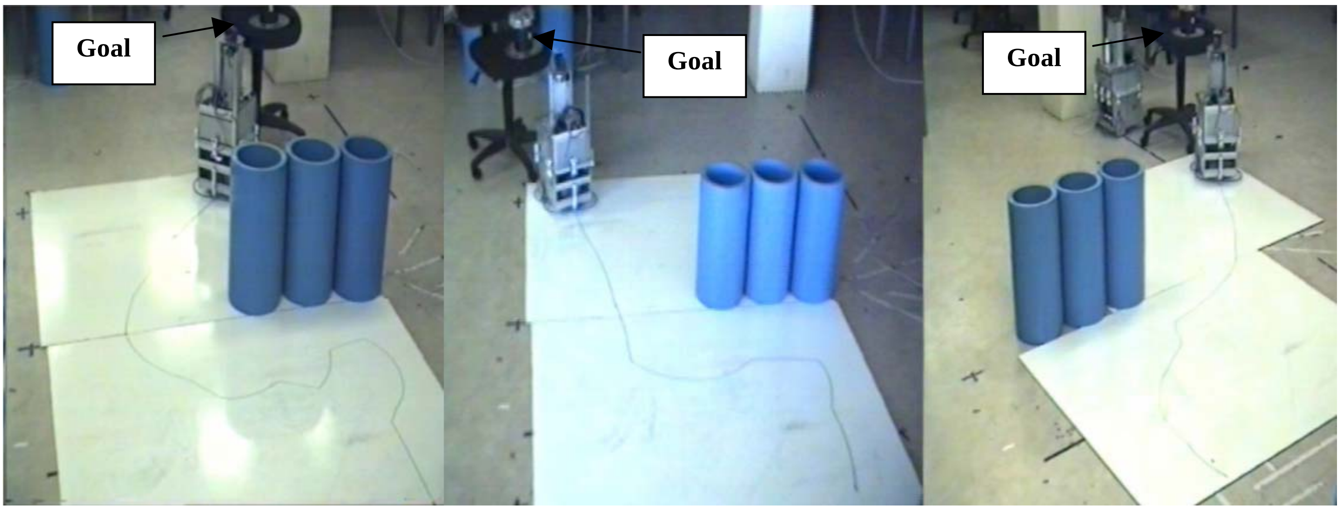
In the previous experiments we have introduced the basic type-2 FLC based behaviours and how they had over performed the type-1 FLC based behaviours, while the type-2 FLC based behaviours have used a smaller number of fuzzy sets to represent each input and thus a smaller rule base. In the following experiments we are going to present the results obtained from type-2 HFLC coordinating more than one behaviour to satisfy more than one objective.



(a) (b) (c)

Figure (14): a) The indoor robot path using type-2 FLC to implement the obstacle avoidance behaviour. b)The type-2 FLC path compared against type-1 FLC path using 3 fuzzy sets to represent each input. c)Another experiments where the type-2 FLC path compared against type-1 FLC path using 3 fuzzy sets to represent each input.

Figure (15) shows the indoor robot coordinating the goal seeking and the obstacle avoidance type-2 based behaviours to reach a goal while avoiding obstacles. As we mentioned in subsection (5.3) as we do not need the left and right edge following behaviours they were removed with their *contexts* and *contexts rules* from the HFLC. We have tried 8 experiments with various obstacles sizes and shapes and with placing the goals at different positions. Figure (15-a, 15-b, 15-c) show the smooth response of the coordinated type-2 HFLC where the goal is placed in different places. The robot had produced a smooth response and in all cases the robot was able to reach its goal with an average deviation from the goal of 4^o and a standard deviation of 0.8. Also it was always able to arrive safely to its goal while keeping a safe distance of 45 cm from any obstacle with an average deviation of 2.7 cm and a standard deviation of 1.1.



(a) (b) (c)

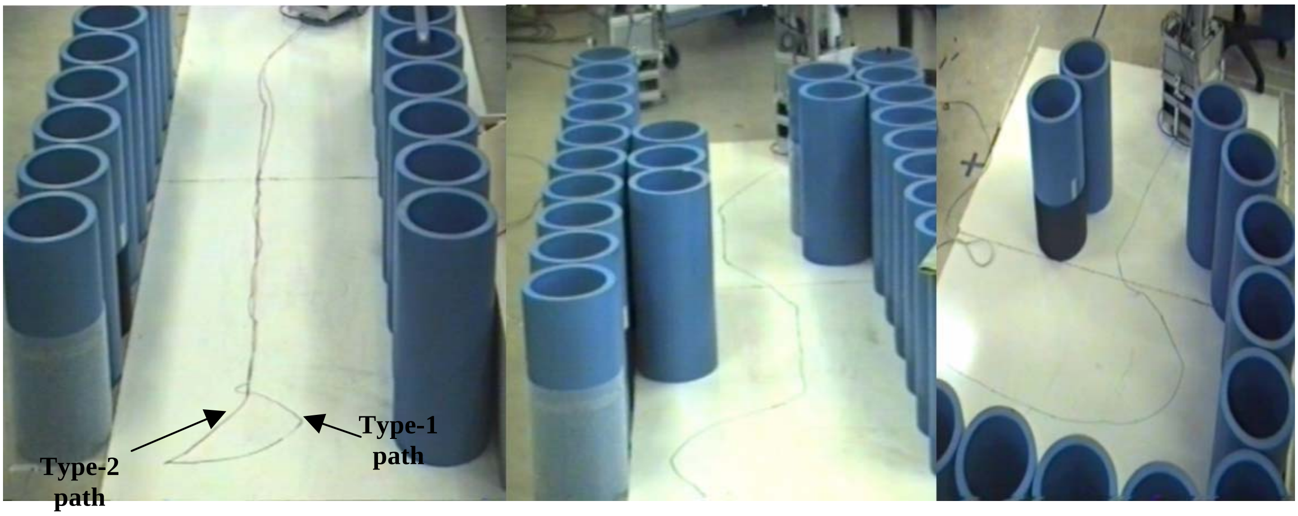
Figure (15):a) The indoor robot path using type-2 HFLC to reach a goal aligned with the robot. b) The robot path when the goal is to the left of the robot. c) The robot path when the goal is to the right of the robot.

Figure (16-a) shows the robot coordinating all its behaviours 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 behaviours the corridor following behaviour had emerged. We have performed 8 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 the goal at the end of the corridor. Again the type-2 HFLC had over performed the type-1 HFLC as shown in Figure(14-a) which produced an average deviation of 3.4 cm and a standard deviation of 1.8 from the centre line of the corridor. The type-1 HFLC used an obstacle avoidance behaviour of rule base of 27 rules and left and right edge following behaviour, each having 9 rules and the goal seeking behaviour had a rule base of 7 rules, thus the total number of rules in this type-1 HFLC is 27+9+9+7= 52 rules while the type-2 HFLC had only a total number of 8+4+4+3= 19 rules so the type-2 HFLC produced a rule reduction of about 64 % and over performed the performance of type-1 HFLC.

Figure (16-b) shows the robot following the centre line of the corridor while avoiding obstacles and reaching a goal at the end of the corridor. The robot had followed the centre line of the corridor with almost the same average and standard deviation as Figure (16-a), while avoiding the obstacles at the safe distance with an average deviation of 2.7 cm and a standard deviation of 1.1 and it had always reached its goal.

Figure (16-c) shows the robot coordinating three type-2 behaviours 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. 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.

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 amount of uncertainty and imprecision associated with the robot sensors and actuators in such changing and dynamic environments.



(a) (b) (c)

Figure (16): a) The indoor robot path using type-2 HFLC to follow a corridor and reaching a goal at the end the corridor. b) The path of the type-2 HFLC to follow a corridor while avoiding obstacles and reaching a goal at the end of the corridor. c) The path of the type-2 HFLC to follow an edge while avoiding obstacles and reaching a goal at the end of the edge.

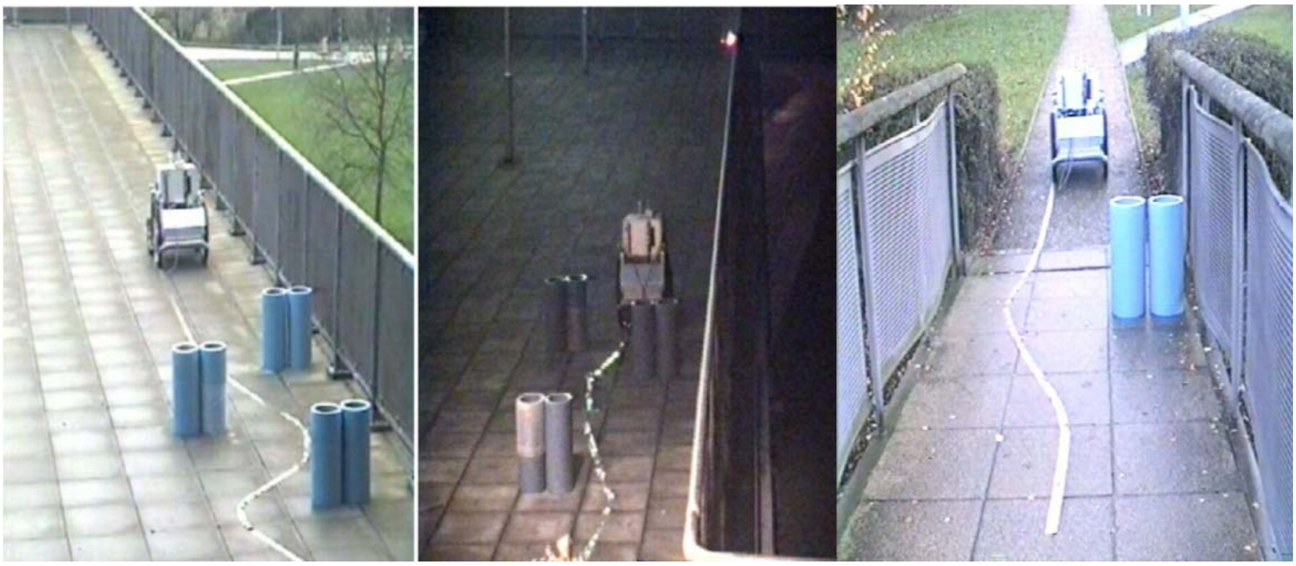
We had performed many experiments using our outdoor robots in outdoor changing and dynamic environments. The robot path was recorded using a tape fixed under the left back wheel and we calculated the average and standard deviations from the desired values. In all the following experiments, all the average and standard deviations from the desired values were calculated over 8 experiments where the robot was started from different random locations and under different environmental conditions from rain to sunshine, .etc and different ground conditions such as slippery, dry grounds and even different times of the day. We also tried different challenging environmental features like metallic and plant edges which offer bad sonar response.

Figure (17-a) shows the outdoor robot implementing the type-2 based right edge following behaviour which used two type-2 fuzzy sets to represent each input (and thus having 4 rules in the rule base) to follow an irregular metallic edge at a desired distance of 1.2 m. The robot had followed the irregular edge with an average deviation of 2.9 cm and a standard deviation of 0.9. The type-1 FLC can give a good response at a specific weather, ground and robot condition but if any of these conditions change like operating in a windy weather condition then the type-1 FLC with 9 rules shown in Figure (17-b) will fail and give a bad response as it is using crisp membership function that cannot handle uncertainty and imprecision. Again the type-2 FLC produces smoother and better response than the type-1 FLC which uses more rules.

We have tried also many experiments to test the type-2 HFLC in outdoor environments, Figure (18-a) shows the robot coordinating the right edge following behaviour and the obstacle avoidance behaviour to follow an irregular edge at a desired distance of 1.2 m and avoid obstacles at a safe distance of 1.4 m. We have performed many experiments starting the robot from different random positions and with different obstacles configuration and different weather conditions, and different time of the day as shown in Figure (18-b) where we performed some experiments during the night. The type-2 HFLC had always given a very good response as shown in Figure (18-a, 18-b) where the robot had followed the edge with a smooth response and with an average deviation of 2.9 cm and standard deviation of 0.9 from the desired distance, it also avoided the obstacles within the safe distance with an average deviation of 3.2 cm and a standard deviation of 1.2.

(a) (b)

Figure (17): a) The outdoor robot path using type-2 FLC to implement the right edge following behaviour to follow an irregular edge. b) The robot path using a type-1 FLC which gave a bad response when the environment changed (windy weather).



(a) (b) (c)

Figure (18): a) The outdoor robot path using the HFLC to follow an edge while avoiding obstacles. b) The same experiment as in Figure (18-a) carried during night using different obstacles configurations. c) The outdoor robot path using the HFLC to follow an irregular corridor while avoiding obstacles

Figure (18-c) shows the outdoor robot coordinating its left and right edge following and obstacle avoidance behaviours 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 centre line of the corridor with an average deviation of 2.9 cm and standard deviation of 1.1 while avoiding obstacles at the desired safe distance.

In all the previous experiments the type-2 FLC and HFLC response was repeatable when started from different starting positions and under different environmental conditions from rain to sunshine, .etc and different ground conditions such as slippery, dry grounds and even different times of the day. The type-2 FLC can also deal in real time with any dynamic changes. Thus we found that type-2 was particularly suitable for outdoor environments which is intrinsically more changeable than an indoor environment.

7. Conclusions and Future Work

In this paper we presented a novel type-2 fuzzy architecture for the real time control of mobile robots navigating in highly dynamic unstructured indoor and outdoor environments. This architecture was based on type-2 FLC to implement the basic navigation behaviours and the coordination between these behaviours. 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 [22].

We have shown that a type-2 FLC can over perform the type-1 FLC for the following reasons:

- The type-2 FLC uses type-2 fuzzy sets, which can handle and minimize the effects of uncertainties associated with such unstructured environments.
- The type-2 FLC has a better performance and smother control surface than the type-1 FLC as each type-2 fuzzy set is a collection of many embedded type-1 fuzzy sets to describe the input and output variables which allows for a more detailed description of the analytical control surface as the addition of the extra levels of classification, giving a much smoother control surface and response [12].
- The use of type-2 FLC results in a significant rule reduction when compared with the type-1 FLC as linguistic uncertainty provided by the type-2 fuzzy sets allows us to cover the same range with a much smaller number of labels [22].

However the type-2 FLC for a mobile robot with many inputs (and hence a large rule base) has a major bottleneck in the type-reduction process where the time needed to compute the type-reduced sets increases with the number of rules, which might prohibit the real-time operation of type-2 FLC [22]. In this paper we have introduced a mechanism that solves this problem by using a two-level type-2 hierarchical fuzzy system which has the following advantages:

- It 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.
- It harnesses the benefits of type-2 fuzzy logic to deal with the large amount imprecision and uncertainty present in such a highly dynamic unstructured environments
- It uses type-2 fuzzy logic for the co-ordination, which deals with the uncertainty in the coordination level and provides a smooth transition between behaviours with a consequent smooth output response.
- The robots can achieve multiple goals, whose priorities may change with time.
- It offers a flexible structure where new behaviours can be added or modified easily.

We have presented numerous experiments using different robots, navigating in varied andchallenging indoor and outdoor unstructured environments. We have shown how the type-2 FLCs and HFLCs has produced very good real-time responses that had out-performed the type-1 FLCs and HLFCs while resulting in a significant rule reduction of about 64 %.

We are currently designing type-2 FLCs for the control of large marine diesel engines. In addition we have aproject that is applying these techniques to the control of intelligent buildings and the creation of ambient-intelligence in pervasive and ubiquitous computing environments.

Although type-2 fuzzy controllers can handle some uncertainties and deal with changing environment and robot conditions, it cannot deal with any large changes not accounted for in the fuzzy sets footprint of uncertainties. Thus there is a need to develop an system that can adapt the type-2 controllers online, if the changes can no longer be handled with the type-2 fuzzy system (similar to the system designed by us for type-1 systems [26,27, 28]). Such online adaptive system is a primary aim for our future work.

8. References

[1] M. Arbib, “Perceptual structures and distributed motor control”, *Handbook of Physiology- The Nervous System II*, Editor, V. Brooks, pp. 1449-1465, American Physiological Society, Bethesda ,MD, 1981.

[2] R. Arkin, *Behaviour-based Robotics*, MIT Press, 1998.

[3] H. Beom and H. Cho, “ A sensor-based navigation for a mobile robot using fuzzy Logic and reinforcement learning”, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 25, No. 3, pp. 464- 477, 1995.

[4] A. Bonarini, “Anytime learning and adaptation of hierarchical fuzzy logic behaviours,” *Adaptive Behavior Journal*, Vol. 5, No. 3-4, pp. 281-315, 1997.

[5] R. Brooks, “A robust layered control system for a mobile robot,” *IEEE Journal of Robotics and Automation*, Vol.2, No.1, pp.14-34, 1986.

[6] R. Brooks, *Artificial Life and Real Robots*, MIT press 1992.

[7] A. Cerezo, A. Mandow and M. Lopez, “Fuzzy modelling operator navigation behaviours, ” *Proceedings of the 6th IEEE International Conference on Fuzzy Systems*, pp. 1339-1345, Barcelona, Spain, 1997.

[8] T. Fukuda and N. Kubota, "An intelligent robotic system based on fuzzy approach," *Proceedings of the IEEE*, Vol. 87, No. 9, pp. 1448-1470. 1999.

[9] S. Goodridge, M. Kay and R. Luo, "Multi-layered fuzzy behaviour fusion for reactive control of an autonomous mobile robot, " *Proceedings of the 6th IEEE International Conference on Fuzzy Systems*, pp. 579-584, Barcelona, Spain, 1997.

[10] R. John, "Type 2 fuzzy sets: An appraisal of theory and applications, " *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*, Vol. 6, No6, pp. 563-576, 1998.

[11] R. John, “Embedded interval valued fuzzy sets,” *Proceedings of IEEE International Conference on Fuzzy Systems*, pp. 1316-1321, Honolulu, USA, 2002.

[12] A. Kandel and G. Langholz**,** *Fuzzy Control Systems***,** Boca Raton : CRC Press, 1994.

[13] N. Karnik and J. Mendel, “An introduction to type-2 fuzzy logic systems,” USC report, http://sipi.usc.edu/~mendel/report, October 1998.

[14] N. Karnik, J. Mendel and Q. Liang, "Type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems*, Vol. 7, pp. 643-658, December 1999.

[15] G. Klir and T. Fogler, *Fuzzy Sets, Uncertainty and Information.* Prentice Hall, 1988

[16] B. Kosko, *Neural Networks and Fuzzy Systems : a Dynamical Systems Approach to Machine Intelligence*, Englewood Cliffs, NJ : Prentice Hall, 1992

[17] Q. Liang, N. Karnik and J. Mendel, "Connection admission control in ATM networks using survey-Based type-2 fuzzy logic systems," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, Vol. 30, pp. 329-339, August 2000.

[18] Q. Liang, and J. Mendel, "Interval type-2 fuzzy logic systems: theory and design," *IEEE Transactions on Fuzzy Systems*, Vol. 8, pp. 535-550, October 2000.

[19] C. Lee, “Fuzzy logic in control systems: Fuzzy logic controller, Part II“, *IEEE Transactions on Systems, Man and Cybernetics*, Vol.20, pp. 419-434, March 1990.

[20] M. Maeda, Y. Maeda and S. Murakami, “Fuzzy drive control of an autonomous mobile robot,” *Journal of Fuzzy sets and Systems*, Vol. 39, pp. 195-204, 1991.

[21] V. Matellan, C. Fernandez and J. Molina, "Genetic learning for fuzzy reactive controllers", *Journal of Robotics and Autonomous Systems*", Vol. 25, pp.33-41, 1998.
[22] J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Upper Saddle River, NJ: Prentice-Hall, 2001.
[23] J. Mendel and R. John, "Type-2 fuzzy sets made simple," *IEEE Transactions on Fuzzy Systems*, Vol. 10, pp. 117-127, April 2002.
[24] J. Mendel and H. Wu, "Uncertainty versus choice in rule-based fuzzy logic systems," *Proceedings of IEEE International Conference on Fuzzy Systems*, pp. 1336-1342, Honolulu, USA, 2002.
[25] H.Hagras, V.Callaghan and M.Colley, "On line calibration of the sensors fuzzy membership functions in autonomous mobile robots" *Proceedings of the 2000 IEEE international conference on Robotics and Automation*, pp. 3233-3239, San Francisco, USA, April 2000.
[26] H.Hagras, V.Callaghan and M.Colley, "Learning fuzzy behaviour co-ordination for autonomous multi-agents online using Genetic Algorithms & real-time interaction with the environment," *Proceedings of the 2000 IEEE international conference on Fuzzy Systems*, pp.853-859, San Antonio, USA, May 2000.
[27] H. Hagras, V. Callaghan and M. Colley, "Prototyping design and learning in outdoor mobile robots operating in unstructured outdoor environments," *IEEE International Robotics and Automation Magazine*, Vol. 8, No.3, pp.53-69, September 2001.
[28] H. Hagras, M. Colley, V. Callaghan and M. Carr-West, "Online learning and adaptation of autonomous mobile robots for sustainable agriculture," *International Journal of Autonomous Robots*, Vol. 13 pp. 37-52, July 2002.
[29] 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," *To appear in the Journal of Fuzzy Sets and Systems*, 2003.
[30] F. Hoffmann and G. Pfister, "Evolutionary design of a fuzzy knowledge base for a mobile robot," *International Journal of Approximate Reasoning,* Vol. 17, No. 4, pp. 447-469, 1997.
[31] G. Raju and J. Zhou, "Adaptive hierarchical fuzzy controller," *IEEE Transactions on Systems, Man and Cybernetics*, Vol.23, No. 4, pp.973-980, 1993.
[32] E. Ruspini, "Truth as utility a conceptual synthesis", *In Proceedings of the 7th Conference on uncertainty in Artificial Intelligence,* pp. 458-464, Los Angeles, USA 1991.
[33] A. Saffiotti, " The uses of fuzzy logic in autonomous robot navigation, " *Journal of Soft Computing*, Vol. 1, No. 4, pp. 180-197, 1997.
[34] A. Saffiotti, "Fuzzy logic in autonomous robotics: behaviour coordination," *Proceedings of the 6th IEEE International Conference on Fuzzy Systems*, pp. 573-578, Barcelona, Spain, 1997.
[35] E. Tunstel, T. Lippincott and M. Jamshidi, "Behaviour hierarchy for autonomous mobile robots: Fuzzy behaviour modulation and evolution," *International Journal of Intelligent Automation and soft computing*, Vol.3, No.1, pp. 37-49, 1997.
[36] L. Wang, "Analysis and design of hierarchical fuzzy systems," *IEEE Transaction on Fuzzy Systems*, Vol.7, No.5, pp. 617-624, October 1999.
[37] K. Wu, "Fuzzy interval control of mobile robots," *Journal of Computers Electrical Engineering*, Vol. 22, No.3, pp. 211-229, 1996.
[38] H. Wu and J. Mendel, "Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems", *IEEE Transaction on Fuzzy Systems,* Vol. 10, No. 5, pp. 622-639, October 2002.

