

Interval Type-2 Fuzzy Logic Control of Mobile Robots

Nesrine Baklouti¹, Robert John², Adel M. Alimi¹

¹REGIM: Research Group on Intelligent Machines, National Engineering School of Sfax (ENIS), Sfax, Tunisia; ²Centre for Computational Intelligence, De Montfort University, Leicester, UK.
Email: nesrine.baklouti@gmail.com, rij@dmu.ac.uk, adel.alimi@enis.rnu.tn

Received May 9th, 2012; revised July 31st, 2012; accepted August 7th, 2012

ABSTRACT

Navigation of autonomous mobile robots in dynamic and unknown environments needs to take into account different kinds of uncertainties. Type-1 fuzzy logic research has been largely used in the control of mobile robots. However, type-1 fuzzy control presents limitations in handling those uncertainties as it uses precise fuzzy sets. Indeed type-1 fuzzy sets cannot deal with linguistic and numerical uncertainties associated with either the mechanical aspect of robots, or with dynamic changing environment or with knowledge used in the phase of conception of a fuzzy system. Recently many researchers have applied type-2 fuzzy logic to improve performance. As control using type-2 fuzzy sets represents a new generation of fuzzy controllers in mobile robotic issue, it is interesting to present the performances that can offer type-2 fuzzy sets by regards to type-1 fuzzy sets. The paper presented deep and new comparisons between the two sides of fuzzy logic and demonstrated the great interest in controlling mobile robot using type-2 fuzzy logic. We deal with the design of new controllers for mobile robots using type-2 fuzzy logic in the navigation process in unknown and dynamic environments. The dynamicity of the environment is depicted by the presence of other dynamic robots. The performances of the proposed controllers are represented by both simulations and experimental results, and discussed over graphical paths and numerical analysis.

Keywords: Type-2 Fuzzy Logic; Control; Motion Planning; Mobile Robots

1. Introduction

Mobile robotic navigation in a dynamic and unknown environment is a local path planning problem based on sensory information with no knowledge about the form and location of obstacles. Precise end-effector and positioning accuracy are required in the navigation process. Due to their mechanic conception, Robots, by their very nature, have significant uncertainties in their readings and movements. These errors can be summarized as follows:

- Sensor measurements are usually noisy due to the instruments. Generally a robot is equipped with internal sensors like encoder sensors and coupling sensors and external sensors as laser sensors, infrared sensors and cameras. Readings extracted from those sensors are generally noisy [1],
- There are assembly errors that include linear and angular errors produced during the assembly of the various robot mechanical components,
- Indeed, there are uncertainties provoked from a variation in the coupling of actuators characteristics with environmental conditions. Measurement and actuator create end-effector positioning errors. The resolution of encoders and stepper motors are examples of such

errors. Although these errors are in the most cases very small, they can be amplified to cause many errors that can affect the accuracy of the system [1],

- Real environments are not ideal, so they can produce random errors that change unpredictably. Those errors are called non-systematic errors and are generally caused by irregularities or roughness of the floor.

In order to overcome those uncertainties and to develop a robust, flexible and on-line planner, type-1 fuzzy logic has been used [2-5]. However type-1 fuzzy logic cannot fully handle the stated uncertainties because it uses precise type-1 fuzzy sets which don't necessarily cope well with all sources of vagueness and uncertainty. There are at least three sources of uncertainty in type-1 fuzzy logic control:

- According to Mendel [6] "words mean different things to different people", so while designing a fuzzy controller experts are unlikely to agree on the membership functions,
- Consequents in the rules can be uncertain,
- And the database used in the conception of a fuzzy controller is generally not ideal and contains errors, even if the database has been constructed by an expert.

Then in mobile robot control by fuzzy logic, all the cited forms of uncertainties will be multiplied over fuzzification, inference and defuzzification. Those errors can degrade the performance of the whole robot controller. Type-2 fuzzy logic has been used by researchers to try and overcome some of these difficulties [6-8]. And since many researchers have explored the use of type-2 fuzzy logic controllers in various applications [9-16]. In fact, type-2 fuzzy sets were initially introduced by Zadeh [17]. Firstly a general type-2 fuzzy set was defined, where it represents a 3D set in which each membership grade is a type-1 fuzzy set bounded in $[0,1]$. Due to the complexity of the join (OR) and meet (AND) operations performing the inference part [18] and type-reduction in the defuzzification part [19], the application of general type-2 fuzzy sets has been limited. Hence, a simplified version of general type-2 set called interval type-2 fuzzy set is used more widely [6,20]. This kind of set has membership grades that are crisp interval sets bounded in $[0,1]$. The uncertainty here is represented as a 2D bounded region that is called the Footprint of Uncertainty. Various researchers have explored the advantages of interval type-2 fuzzy sets [21,22]. Moreover, a geometric approach has been introduced by Coupland and John [23] distinguishing between fuzzy logic over discrete and continuous domain. But this approach is not fast enough in control applications [24]. In the rest of this paper we treat only interval type-2 fuzzy sets over discrete domains.

In mobile robotics, some researchers have explored the control of mobile robots using interval type-2 fuzzy logic [21,24-29]. As Hagraas states in [21] control using type-2 fuzzy sets represents a new generation of fuzzy controllers. In [25] Hagraas presented an interval type-2 fuzzy logic controller to command a robot in indoor and outdoor unstructured environment. A robot was tested under different sources of non-systematic errors. The results showed that type-2 fuzzy logic outperforms its type-1 counterpart. This was shown through robot paths and control surfaces. In [27], an interval type-2 fuzzy logic was proposed for the control of a robot tracking a mobile object in the context of robot soccer games. In this game the robot has to track a ball. To evaluate the performance of the type-2 fuzzy logic against its type-1 counterpart, graphical paths analysis were presented showing the way the player reaches the position of the ball. Also, an additional test was made presenting the ability of type-2 controller to track the ball with less standard deviation error than its type-1 counterpart.

In this paper we propose specific aspects of control of mobile robots in unknown and dynamic environments using type-2 fuzzy logic. The dynamicity of the environment is depicted by the presence of other dynamic robots. The performances of the proposed controllers are represented by both simulations and experimental results,

and discussed over graphical paths and numerical analysis. This paper has essentially two parts: In the first part we designed an IT2TSK fuzzy logic controller for avoiding obstacles using simulations. In the second part we designed a Mamdani interval type-2 (IT2) fuzzy logic controller for wall following behavior using the robot Khepera II from LAMI [30]. The remainder of the paper is organized as follows: Section II introduces type-2 fuzzy logic. Sections III presents the proposed Interval type-2 fuzzy logic controller for obstacle avoidance behavior. Next section presents the conception of an Interval type-2 fuzzy logic for a wall following behavior. Simulations and experimental results in both applications are presented and are discussed. Finally, some conclusions are pointed out in Section VI.

2. Type-2 Fuzzy Logic

2.1. Overview on Type-2 Fuzzy Sets

Type-1 fuzzy sets are certain and crisp, whereas type-2 fuzzy sets are themselves fuzzy. Type-2 fuzzy sets were first introduced by Zadeh [17]. An Interval T2 fuzzy set \tilde{A} is described as in the following definition, where in (1) all $\mu_{\tilde{A}}(x,u) = 1, 0 \leq \mu_{\tilde{A}}(x,u) \leq 1$.

$$\tilde{A} = \int_{x \in X} \int_{u \in J_X} 1/(x,u) J_X \subseteq [0,1] \quad (1)$$

where \int denotes the union of all admissible x and u . An IT2 FS is represented by a bounded region limited by two membership functions, where corresponding to each primary MF (which is in $[0,1]$), a secondary MF is used to the primary one. The Uncertainty in the primary membership function consists of the union of all membership functions. This Uncertainty represents a bounded region that we call the Footprint of Uncertainty (FOU), *i.e.*,

$$FOU(\tilde{A}) = \bigcup_{x \in X} u \in J_X \quad (2)$$

The (FOU) represents a complete description of an IT2 FS. It uses an (UMF) Upper Membership Function, and (LMF) Lower Membership Function; The (UMF) and the (LMF) of \tilde{A} are two T1 MFs that bound the (FOU). (UMF) represents the upper bound of $FOU(\tilde{A})$ and is denoted $\bar{\mu}_{\tilde{A}}(x)$, and the LMF represent the lower bound of $FOU(\tilde{A})$ and is denoted $\underline{\mu}_{\tilde{A}}(x)$:

$$\begin{aligned} \bar{\mu}_{\tilde{A}}(x) &\equiv \overline{FOU(\tilde{A})} \forall x \in X \\ \underline{\mu}_{\tilde{A}}(x) &\equiv \underline{FOU(\tilde{A})} \forall x \in X \end{aligned} \quad (3)$$

For an IT2 FS, $J_X = [\bar{\mu}_{\tilde{A}}(x), \underline{\mu}_{\tilde{A}}(x)]$, $\forall x \in X$. In the rest of this paper, we will use only the IT2 fuzzy sets in the design of our work.

2.2. Interval Type-2 Fuzzy Logic Controller

In fact, a type-2 fuzzy logic system or controller uses the same familiar notions as used in a type-1 fuzzy logic controller as membership functions, rules, t-norms operations, fuzzification, inference, defuzzification. A type-1 fuzzy logic system consists basically of three blocs; fuzzification, inference and defuzzification as presented in **Figure 1**. A type-2 fuzzy logic system is very similar to type-1, where it follows the same methodology, but the only difference is in the third block where we no longer speak of only defuzzification but we speak about a type reducer and defuzzification parts that constitute both the output processing block. This difference is mainly associated with the nature of the membership functions, where type-reducer is needed due to the added degree in the kind of fuzzy sets. **Figure 2** presents a type-2 fuzzy logic system.

Today, the two most popular fuzzy logic systems used by engineers in control are the Mamdani and TSK systems.

2.2.1. Mamdani Type-2 Fuzzy Logic Controller

2.2.1.1. Fuzzification

In this part, we must first define the fuzzy sets of all inputs' system. Those memberships can contain one or several type-2 fuzzy sets. Second, the fuzzifier maps inputs into the associated fuzzy sets to determine the degree of membership of each input variable. We consider only singleton fuzzification for which the inputs are crisp values.

2.2.1.2. Inference

This block expresses the relationship that exists between the input variables (expressed as linguistic variables) and the output variables (also expressed as linguistic variables). As in type-1 fuzzy logic, in the design of a type-2 fuzzy logic we generally have IF-THEN rules. The for-

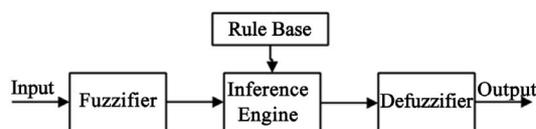


Figure 1. Type-1 fuzzy logic.

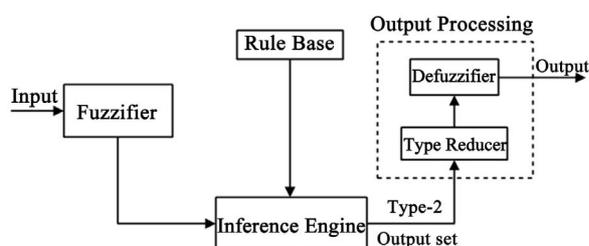


Figure 2. Type-2 fuzzy logic.

mulation of rules is the same. The only distinction between type-1 and type-2 is associated with the nature of the membership functions. The inference engine combines rules and makes a combination between input type-2 fuzzy sets and output type-2 fuzzy sets. This is ensured by searching unions and intersections of type-2 sets, as well as compositions of type-2 relations.

For a type-2 fuzzy logic with p inputs and $x_1 \in X_1, \dots, x_p \in X_p$ and one output $y \in Y$, and with M rules. The ℓ th rule has the following form:

$$R^\ell : \text{IF } x_1 \text{ is } \tilde{F}_1^\ell \text{ and } \dots \text{ and } x_p \text{ is } \tilde{F}_p^\ell \\ \text{THEN } y \text{ is } \tilde{G}^\ell \quad l=1 \dots M$$

The firing strength of the i th rule is as in (4). The result of the input and antecedent operations is an interval type-1 set.

$$F^i(x') = [\underline{f}^i(x'), \bar{f}^i(x')] \equiv [\underline{f}^i, \bar{f}^i] \\ = \left[\underline{\mu}_{\tilde{F}_1^i}(x'_1) * \dots * \underline{\mu}_{\tilde{F}_p^i}(x'_p), \bar{\mu}_{\tilde{F}_1^i}(x'_1) * \dots * \bar{\mu}_{\tilde{F}_p^i}(x'_p) \right] \quad (4)$$

where $\bar{\mu}_{\tilde{F}_\ell^i}(x)$ and $\underline{\mu}_{\tilde{F}_\ell^i}(x)$ designed respectively upper and lower membership grades of $\mu_{\tilde{F}_\ell^i}(x)$ and

$$\underline{f}^i = \underline{\mu}_{\tilde{F}_1^i}(x_1) * \dots * \underline{\mu}_{\tilde{F}_p^i}(x_p) \\ \bar{f}^i = \bar{\mu}_{\tilde{F}_1^i}(x_1) * \dots * \bar{\mu}_{\tilde{F}_p^i}(x_p) \quad (5)$$

Since generally we use the meet operation under product or minimum t -norm. So, at each value of x the intersection and union operations are referred to as the meet and join operations, respectively.

2.2.1.3. Type Reducer and Defuzzification

In a type-1 fuzzy logic system the output of the inference engine corresponding to each fired rule is a type-1 set. The defuzzifier combines those output sets to obtain a single output set. Using one of the existing methods of defuzzification, for example the centroid of sets, the defuzzifier searches the centroid of the obtained set to obtain finally a crisp output. In a type-2 fuzzy logic, since this kind of system deal with type-2 sets, then it is necessary to have a type reducer block to map a T2 FS into a T1 FS, and then defuzzification, as usual, maps that T1 FS into a crisp number. We can consider that the defuzzification block of a T1 fuzzy logic is replaced by the output processing block in a T2 fuzzy logic. That block consists of type-reducer followed by defuzzification. In fact, type Reducer was proposed by Karnik and Mendel [7]. For now, there are five different type-reduction methods. Karnik and Mendel [19] defined the centroid of an IT2 FS which is an IT1 FS that is ensured using the Extension Principle. This IT1 FS is characterized by its left and right end points y_l and y_r , which can be written

in the following equation:

$$[y_l, y_r] = \int_{y^l \in [w_\ell^M, w_r^M]} \dots \int_{v^M \in [w_\ell^M, w_r^M]} \int_{f^M \in [f^M, \bar{f}^M]} 1 / \left(\frac{\sum_{i=1}^M f^i w^i}{\sum_{i=1}^M f^i} \right) \quad (6)$$

This equation is derived from the consequent centroid sets $[w_\ell^M, w_r^M]$.

2.2.2. Interval Type-2 TSK Fuzzy Logic Controller

The differences between a type-1 TSK fuzzy logic controller [31] and a Mamdani T1 fuzzy logic consist essentially of the definition of outputs and then on the consequent part of rules. Consider we have a first order type-1 TSK fuzzy logic with p inputs $x_1 \in X_1, \dots, x_p \in X_p$, one output $y \in Y$, and with M rules. The ℓ th rule can be expressed as:

$$R^\ell : \text{IF } x_1 \text{ is } \tilde{F}_1^\ell \text{ and } \dots \text{ and } x_p \text{ is } \tilde{F}_p^\ell \\ \text{THEN } y^\ell = c_0^\ell + c_1^\ell x_1 + \dots + c_p^\ell x_p$$

A type-2 TSK fuzzy logic controller or system (T2 TSK fuzzy logic) was firstly introduced by Liang and Mendel [32]. Although TSK type-1 fuzzy systems have received a lot attention, the literature on TSK type 2 fuzzy systems is few. Liang and Mendel applied type-2 TSK systems in channel equalization of channels [12]. Where, according to them, there are three models of T2 TSK fuzzy logics depending on the kind of the antecedent and consequent part of rules, to have: T2 TSK-Model I, T2 TSK-Model II and T2 TSK-Model III. We can see in **Tables 1** and **2** the difference between those models. where c_i^ℓ, C_i^ℓ are the consequent parameters, y^ℓ, Y^ℓ are the outputs of the ℓ th rule, $\tilde{F}_j^\ell (j=1 \dots p)$ are type-2 fuzzy sets and $\tilde{F}_j^\ell (j=1 \dots p)$ are type-1 fuzzy sets. The firing strength of the i th rule $W^i(x)$ with meet operation under product or minimum t-norm is an interval type-1 set expressed as follows :

$$W^i(x) = [\underline{w}^i(x), \bar{w}^i(x)] \\ \underline{w}^i(x) = \underline{\mu}_{\tilde{F}_1^i}(x_1) * \dots * \underline{\mu}_{\tilde{F}_p^i}(x_p) \quad (7) \\ \bar{w}^i = \bar{\mu}_{\tilde{F}_1^i}(x_1) * \dots * \bar{\mu}_{\tilde{F}_p^i}(x_p)$$

The final output is also an interval type-1 set and is calculated as follows:

Table 1. Models of T2 TSK FLS.

Model	Antecedents	Consequents
Model I	Type-2 fuzzy sets	type-1 fuzzy sets
Model II	Type-2 fuzzy sets	crisp numbers
Model III	Type-1 fuzzy sets	type-1 fuzzy sets

Table 2. Rules of T2 TSK FLS.

TSK FLS	Rules R^ℓ
Type-1	IF x_1 is F_1^ℓ and \dots and x_p is F_p^ℓ THEN $y^\ell = c_0^\ell + c_1^\ell x_1 + \dots + c_p^\ell x_p$
T2 Model I	IF x_1 is \tilde{F}_1^ℓ and \dots and x_p is \tilde{F}_p^ℓ THEN $Y^\ell = C_0^\ell + C_1^\ell x_1 + \dots + C_p^\ell x_p$
T2 Model II	IF x_1 is \tilde{F}_1^ℓ and \dots and x_p is \tilde{F}_p^ℓ THEN $y^\ell = c_0^\ell + c_1^\ell x_1 + \dots + c_p^\ell x_p$
T2 Model III	IF x_1 is F_1^ℓ and \dots and x_p is F_p^ℓ THEN $Y^\ell = C_0^\ell + C_1^\ell x_1 + \dots + C_p^\ell x_p$

$$Y(Y^1, \dots, Y^M, W^1, \dots, W^M) = [y_l, y_r] \\ = \int_{y^1} \dots \int_{y^M} \int_{w^1} \dots \int_{w^M} 1 / \left(\frac{\sum_{i=1}^M w^i y^i}{\sum_{i=1}^M w^i} \right) \quad (8)$$

where $y_i \in Y^i$, and $Y^i = [y_l^i, y_r^i], (i=1 \dots M)$, thus for each rule we will obtain y_l and y_r . Since all sets are crisp, the Equation (9) results to:

$$y_l = \frac{\sum_{i=1}^M w_l^i y_l^i}{\sum_{i=1}^M w_l^i}; y_r = \frac{\sum_{i=1}^M w_r^i y_r^i}{\sum_{i=1}^M w_r^i} \quad (9)$$

And the defuzzified output is:

$$y = (y_l + y_r) / 2 \quad (10)$$

3. Interval Type-2 TSK Fuzzy Logic for an Obstacle Avoidance Behavior

3.1. Conception of Type-1 Behavior

We designed a zero-order TSK type-1 fuzzy logic controller for the navigation of a mobile robot in dynamic and unknown environment for obstacle avoidance behavior. The purpose of the controller is to perform the navigation in unknown and dynamic environments for polygonal mobile robot. In this behavior we defined six sensor measurements inputs $d_c, d_l, d_r, d_{ll}, d_{rl}$ and d_a , as are shown in **Figure 3** where those distances represent respectively the zones: central, left, right, lateral left, lateral right and back. Each input expressed the distance to the nearest obstacle in the zone of vision of its correspondent sensor and is defined by two MFs Near and Far, which are represented by trapezoidal membership functions as in **Figure 4**.

The outputs of the behavior are the left and right wheels' rotational velocities noted respectively V_l and V_r . We defined 64 TSK rules deduced by authors' human expertise. For obstacles near the robot, we have widely used the rotation without translation to avoid obstacles instantly. An example of rules is expressed in Equation (11), where we defined the case of robot finding obstacles in his whole left side, so to guarantee avoiding static or dynamic obstacles the robot turn little to his right.

$$\begin{aligned} &\text{IF } d_c \text{ is Far and } d_l \text{ is Near and } d_r \text{ is Far} \\ &\text{and } d_{ll} \text{ is Near and } d_{lr} \text{ is Far and } d_a \text{ is Far} \quad (11) \\ &\text{THEN } [V_l, V_r] = [0.5, 0] \end{aligned}$$

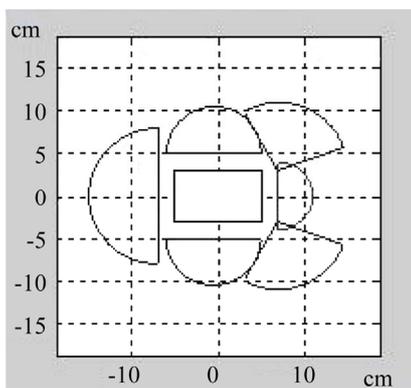


Figure 3. Representation of robot sensor positions.

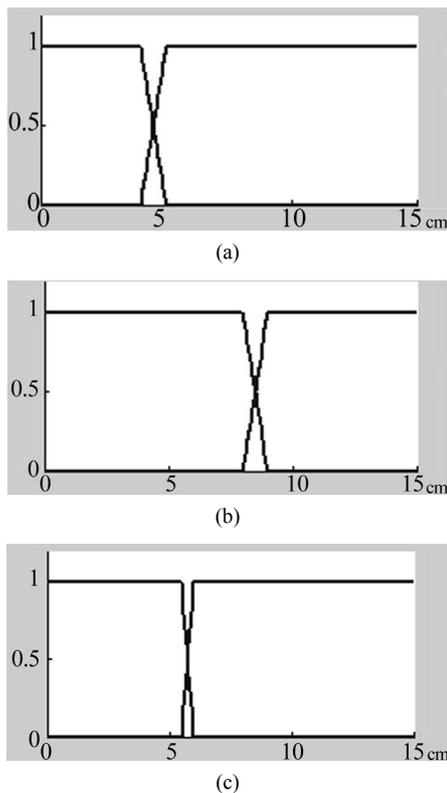


Figure 4. Membership functions of type-1 fuzzy logic inputs.

3.2. Conception of Type-2 Behavior

In the conception of type-2 fuzzy logic for obstacle avoidance behavior, we extended the proposed type-1 controller to a type-2 one by adding uncertainties in both the antecedent and the consequent parts of each rule. The main idea here consists in spreading the membership functions' values of the antecedent part by $\pm A\%$, and the consequent part by $\pm C\%$. Therefore by this, our type-2 fuzzy logic is an IT2 TSK fuzzy logic Model I. So, the six inputs of the behavior were extended by $\pm A\%$. The following figures, **Figures 5-7**, show those inputs for $\pm A\% = \pm 5\%$.

The outputs of the controller are the same as in type-1 fuzzy logic V_r and V_l . We have chosen to fix the spread of the consequent parameter "C" equal to 1 rad/s. We have defined thus 64 TSK T2 rules that are represented in **Table 3** and are as the following form:

$$\begin{aligned} R^\ell : &\text{ IF } d_c \text{ is } \tilde{F}_1^\ell \text{ and } d_l \text{ is } \tilde{F}_2^\ell \text{ and } d_r \text{ is } \tilde{F}_3^\ell \\ &\text{and } d_{ll} \text{ is } \tilde{F}_4^\ell \text{ and } d_{lr} \text{ is } \tilde{F}_5^\ell \text{ and } d_a \text{ is } \tilde{F}_6^\ell \\ &\text{THEN } [V_l^\ell, V_r^\ell] = [C_l^\ell, C_r^\ell] \end{aligned}$$

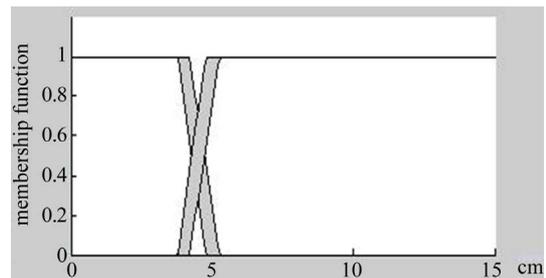


Figure 5. MFs of the frontal input with $\pm 5\%$.

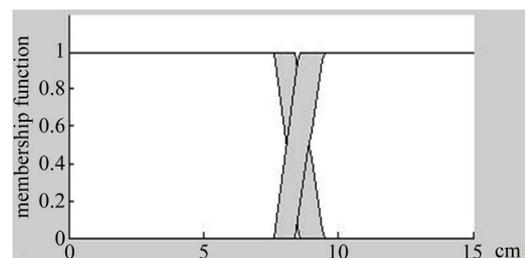


Figure 6. MFs of the lateral input with $\pm 5\%$.

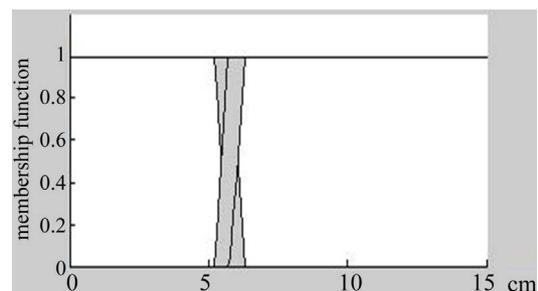


Figure 7. MFs of the back input with $\pm 5\%$.

where \tilde{F}_j^ℓ are type-1 fuzzy sets, and $[C_l^\ell, C_r^\ell]$ are the consequent parameters for $(j=1, \dots, 6)$ and $(\ell=1, \dots, 64)$.

In the inference engine, we use the meet operation under the product t -norm.

3.3. Simulation Results

Several simulations were tested while varying the spread of the antecedent parameters $\pm A\%$. The results were implemented in Matlab 6.5 under the operating system Windows XP. We used an environment with dimensions of 189×190 containing arbitrary complex obstacles, using the SIMROBOT toolbox software [33]. A wheeled mobile robot is considered under the assumptions of non-slipping and pure rolling. Its kinematics can be derived using **Figure 8**.

This robot has two degrees of freedom: y -translation and either x -translation or z -rotation. Its dimensions are taken into account when it navigates in an arbitrary complex environment. The actuated inverse velocity solution of this robot is as the following equation

$$\begin{bmatrix} \omega_{W_1} \\ \omega_{W_2} \end{bmatrix} = \frac{1}{R(l_b^2 + 1)} \begin{bmatrix} -l_a l_b & -l_b^2 - 1 & -l_a \\ l_a l_b & -l_b^2 - 1 & l_a \end{bmatrix} \begin{bmatrix} \mathcal{G}_{B_x} \\ \mathcal{G}_{B_y} \\ \omega_{B_z} \end{bmatrix} \quad (12)$$

Table 3. Rules of the T2 FLC.

Rules	Inputs						Outputs
	d_c	d_l	d_r	d_{ll}	d_{rr}	d_a	$[V_l, V_r]$
1	L	L	L	L	L	L	$[1.85, 1.85] \pm C\%$
2	P	P	P	P	P	P	$[0.95, -0.95] \pm C\%$
...
64	P	P	P	P	P	P	$[0.71, -0.71] \pm C\%$

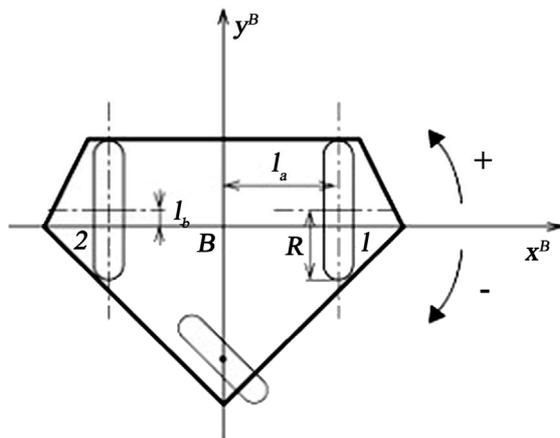


Figure 8. Kinematic of the Simrobot robot.

where: $\mathcal{G}_{B_x}, \mathcal{G}_{B_y}$: are translational velocities of the robot body in m/s , ω_{B_z} : is the robot z -rotational velocity in rad/s , $\omega_{W_1}, \omega_{W_2}$: are wheel rotational velocities in rad/s , R : is actuated wheel radius in m and l_a, l_b : are distances of wheels from robot's axes in m . More details about the robot used kinematics can be found in [33].

To prove the efficiency of T2 fuzzy sets, we firstly choose to test the robot navigation in a complex place, like a narrow passageway. And, as we have not made any learning study on the adequate footprint of uncertainty (FOU) to our T2 fuzzy logic, we make simulations as follows: We tested T1 fuzzy logic and T2 fuzzy logic for different spread or (FOU), from the same initial position and for the same number of steps, which were respectively chosen to “[105 179]” and “110 steps”. Results are presented in the following part; **Figure 9** represents the initial robot position. **Figure 10** shows the trajectory generated by the robot using T1 fuzzy logic controller. From this figure, we notice that while passing through the narrow passage, the robot makes several oscillations to attempt finally, at the end of 110 steps, the position [140 178]. Sure our T1 fuzzy logic controller is not perfect and may contains some issues of errors. This explains well the obtained oscillations in the resulted trajectory where several rules give opposite and acute outputs reflecting an unstable navigation.

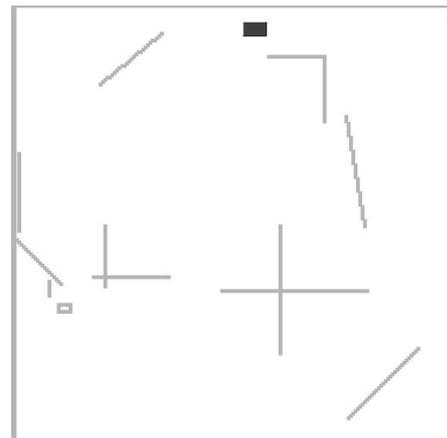


Figure 9. Initial chosen robot position (position = [105 179]).

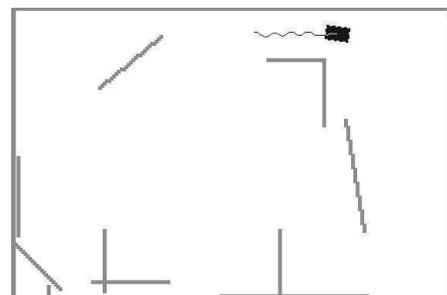


Figure 10. TSK T1 fuzzy logic path for 110 steps.

Figures 11(a)-(f) represent the simulation results of the T2 TSK fuzzy logic for different deviations of the antecedent part $\pm A\%$, which correspond respectively to 5, 10, 15, 20, 30 and 40. It is clear from those figures the ability of T2 fuzzy logics in avoiding obstacles or walls as T1 fuzzy logic, but the most important things to highlight are that all the generated trajectories by T2 fuzzy logics are clearly smoother and contain less oscillations and deviations than the one generated by type-1 fuzzy logic in **Figure 10**. Thus T2 fuzzy sets can reduce uncertainties coming from not perfect tuning or noisy database. Besides, we remark that for the same number of steps, the robot in all T2 fuzzy logics browsed more land than in T1 fuzzy logic. So T2 controllers are more rapid and allow the robot to arrive faster to a given destination point.

To have a quantitative comparison, and to demonstrate well the smoothness of the obtained T2 trajectories, we choose to concentrate only on the narrow passageway; we propose to visualize the angular velocities generated by all T1 and T2 controllers, all from the same initial position [105 179] to almost the same final one [140 178] which represents the end of the passage way as shown in **Figure 12**, while the number of steps is varying depending on the adopted controller to arrive quickly or slowly to the final position. Thus, we obtain those results; **Figure 13** represents the generated angular velocity by T1 fuzzy logic, where the positive values represent the left robot turnings and negative ones represent the right turn-

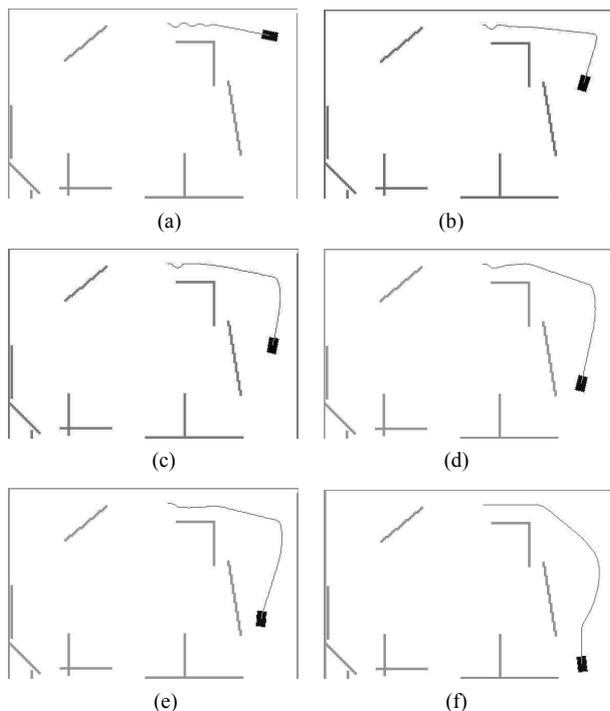


Figure 11. TSK T2 FL paths for 110 steps. (a) $A = 5$; (b) $A = 10$; (c) $A = 15$; (d) $A = 20$; (e) $A = 30$; (f) $A = 40$.

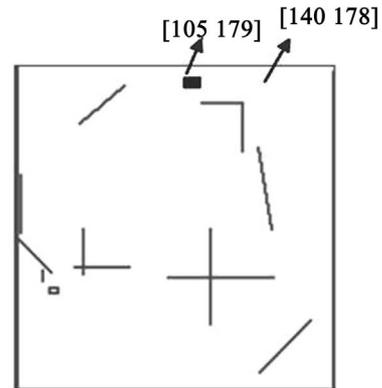


Figure 12. Initial and final chosen robot positions.

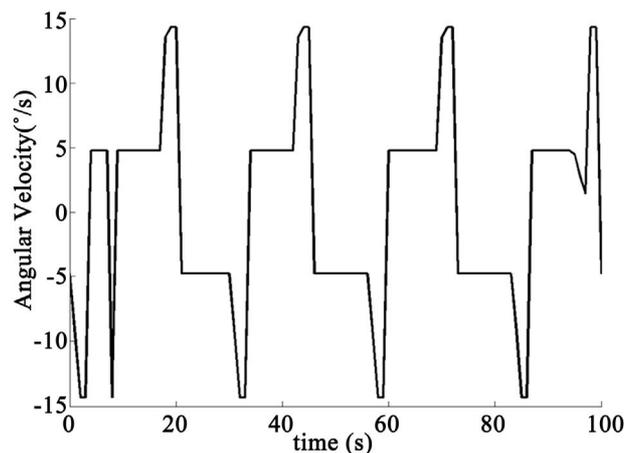


Figure 13. Simulation generated by TSK T1 FLC.

ings. We notice from the figure that the robot continuously oscillates right and left along the whole passage, reflecting by thus the lack of stability and smoothness of T1 fuzzy logic. Whereas **Figures 14(a)-(f)** denote the generated angular velocity by T2 fuzzy logics for different deviations A , which correspond respectively to 5, 10, 15, 20, 30 and 40. We remark clearly from those **Figures 13** and **14**, that all T2 generated trajectories contain less peaks and oscillations than the T1 generated trajectory.

From those figures we extracted in **Table 4** some results; we calculated first number of steps taken by each simulation. We can remark that navigating from the same initial position to the same final one takes from T1 fuzzy logic the biggest number of steps which is 110 steps than all the other T2 fuzzy logics that takes 93 steps for $A = 5$ to only 19 steps for $A = 40$. So, T2 fuzzy logics are faster than T1 fuzzy logic.

Besides, in the table we extracted a smoothness measurement: the angular velocity smoothness Index (AVSI). AVSI represents the average accumulative angular velocities made by the robot simulation during k steps. Moreover, we calculate the Mean Square Error (MSE) expressing the error between the actual generated output

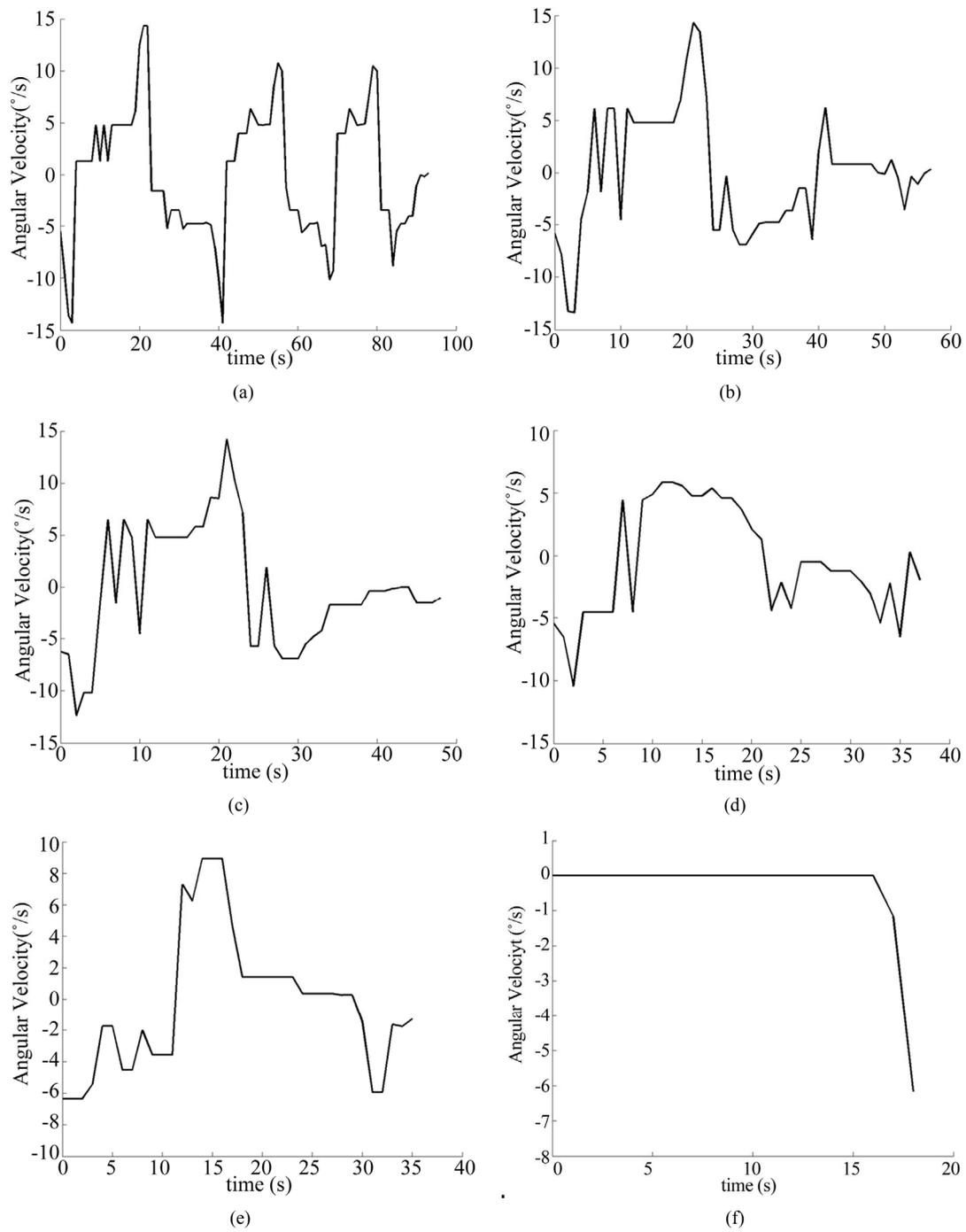


Figure 14. Simulations generated by TSK T2 FLCs for different A. (a) A = 5; (b) A = 10; (c) A = 15; (d) A = 20; (e) A = 30; (f) A = 40.

Table 4. Simulation results given by T1 and T2 controllers.

TSK FLC	T1	T2 A = 5	T2 A = 10	T2 A = 15	T2 A = 20	T2 A = 30	T2 A = 40
Steps	110	93	58	49	38	36	19
AVSI (°/s)	2.6437	2.5582	2.5522	2.4100	2.1224	1.2763	0.3242
MSE	57.08	37.895	33.541	32.075	19.073	19.061	2.0678

from the controller and the expected desired output in the passage that normally might be a straight line path. The equations of these measurements are given as follows:

$$\begin{aligned} \text{AVSI} &= \sum_{i=1}^k \left| \frac{\Delta \overline{AV}_i}{k} \right| \\ \text{MSE} &= \frac{1}{k} \sum_{i=1}^k \left(y_i - y_{d_i} \right)^2 \end{aligned} \quad (13)$$

It can be seen also from **Table 4** that: while the parameter “A” is increasing, the navigator becomes smoother and with less oscillations. This is expressed mainly with a degradation in AVSI and MSE measurements (A = 40, AVSI = 0.3242°/s; MSE = 2.0678). Sure there is a limit of this decrease, but this point is not discussed in this paper. Nevertheless, this remark is very interesting as an optimization point in future work. The most important thing to highlight from the table is that all the T2 generated trajectories represent better results than the T1 one in terms of AVSI or MSE.

Secondly, to prove the efficiency of type-2 fuzzy sets in the setting of local avoiding obstacles, we tested the controllers towards moving obstacles in dynamic and unknown environment. For T2 controller, we choose to test only an example of a T2 controller with a spread equal to A = 20. The dynamicity of the environment is manifested by the presence of several robots. We can see in **Figures 15(a)** and **(b)** the ability of robots in both T1 and T2 controllers in avoiding obstacles and also in avoiding themselves. Although we tested in both figures the robots from same initial positions and for the same number of steps, we can see that we have not the same robots trajectories and this since the controllers did not give the same instant outputs. Also, it is clear from both figures the smoothness, the rapidity of T2 controller in relation to T1 one, where robots in T2 figure browse more land with smooth turnings.

To enhance the found result, we presented another simulation in a different restricted complex place, where the robot is supposed to be front with a corner as in **Figures 16(a)** and **(b)**. We can see from the different generated trajectories that the T2 fuzzy logic with A = 20 presents the smoothest path essentially in the corner part. Where T2 robot turns slowly towards the corner in a way it seems following the wall. Whereas T1 robot presents sharp turns in a way it turns back in other direction.

4. Interval Type-2 Fuzzy Logic for a Wall Following Behavior

In this section we presented a type-2 mamdani fuzzy logic for a wall following behavior for the miniature mobile robot kheperaII from K-Team [30]. In reality, this robot presents a good example of existing uncertainties,

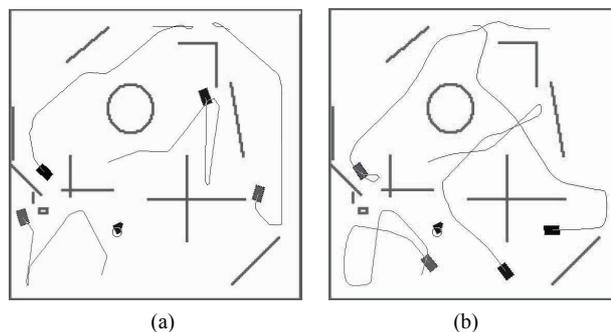


Figure 15. TSK T1 T2 paths (A = 20) from the same initial robots positions.

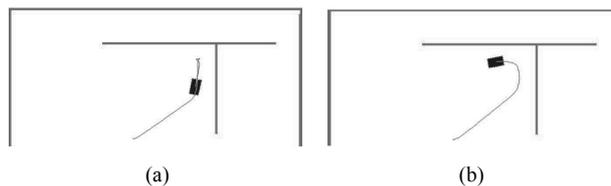


Figure 16. Simulations generated by T1 and T2 FLCs against a corner.

where it is faced to a large amount of vagueness. The kheperaII robot has 8 infrared proximity sensors. Each of them has a maximum range of measurement of about 5cm with accuracy and resolution depending on the measured distance. Those sensors are imprecise and present several kinds of errors like ambient light, color, shape and intensity of the detected obstacle. It was proved in [34] that the total error in the distance estimation by an infrared sensor depends on the uncertainty in the readings and the uncertainty in the angle of incidence of the sensor. In our type-2 fuzzy controller we have three inputs, two outputs and eight rules, that is deduced from its corresponding type-1 fuzzy logic by spreading the antecedent and consequence parts by a footprint of uncertainty. The robot have to navigate forward in the environment until it detects an unexpected obstacle or a wall, follows its contour on the right side with little turns. To detect the wall, the robot used its six frontal proximity sensors which are grouped two by two to constitute the three inputs. Each of them is defined by two fuzzy sets that are presented in **Figure 17**. The outputs are the left and right wheels’ rotational velocities of khepera V_r and V_l . The both are defined with two fuzzy sets and are presented in **Figure 18**.

Whereas, in the inference part we conceived eight rules which are deduced from human expertise. Concerning the implementation part, we used the kMatlab routines [30] to interact with Khepera robot over a serial connection.

The two controllers are compared in same conditions, where robots are fixed approximately at the same initial position and then tested during the same number of steps

chosen equal to 130 steps. **Figure 19** shows the robot in the worked environment. As presented in previous simulations, we extracted the angular velocities generated by T1 fuzzy logic and T2 fuzzy logic illustrated respectively in **Figures 20** and **21**.

We remark from these trajectories that type-2 path presents less deviation with a smoothness measurement index AVSI equal to 1.0934 than type-1 path that presents an AVSI equal to 0.5554. **Figure 22** presents paths generated by both controllers. We noted that in a part of T1 trajectory, the robot touched the wall and did not keep enough distance to it. Whereas T2 robot has kept approximately 8 cm to the wall. Besides, the T2 controller has traveled 22.8 cm more in land than its equivalent T1 controller. Thus, type-1 fuzzy sets may not be robust enough to handle uncertainties caused by infrared measures.

5. Conclusion

In this paper we presented T2 controllers for mobile robot navigation. We can highlight from above results that type-2 fuzzy sets are very interesting in control of mobile



Figure 19. The robot in the worked environment.

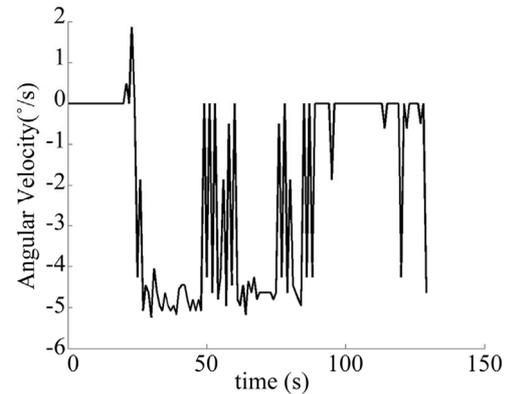


Figure 20. Angular velocities generated by T1 FLC.

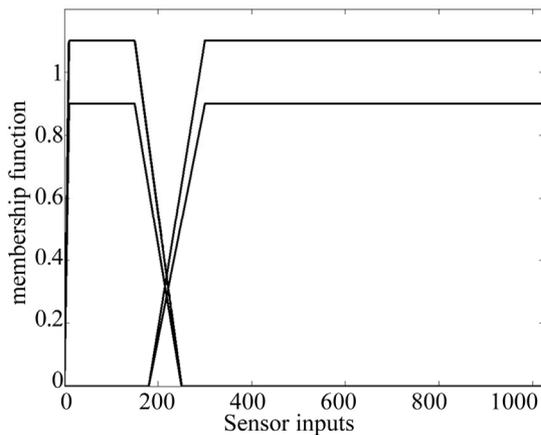


Figure 17. Inputs MFs of the wall following behavior.

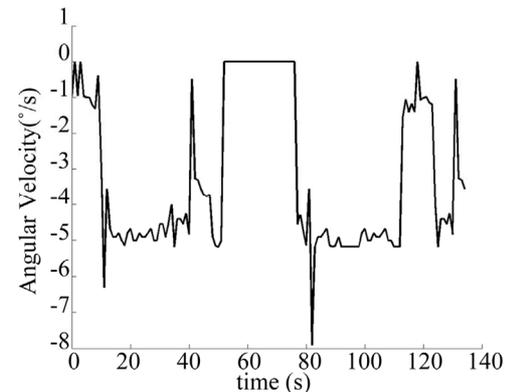


Figure 21. Angular velocities generated by IT2 FLC.

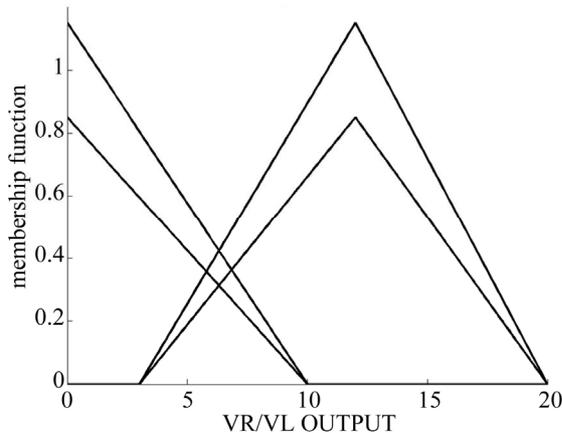


Figure 18. Outputs' MF of the wall following behavior.

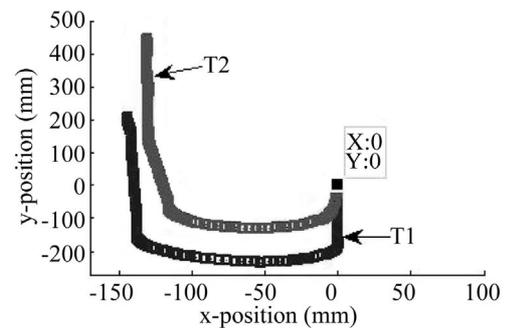


Figure 22. Paths generated by T1 and T2 FLCs.

robots. We have shown deeply that the proposed T2 controllers are more efficient in terms of saving time, smooth trajectories and optimal distance than their counterpart T1s. This was demonstrated through several paths of robots and smoothness and error measures. So, Interval type-2 fuzzy sets help to overcome uncertainties that can exist in real environments.

6. Acknowledgements

The authors would like to acknowledge the financial support of this work by grants from the General Direction of Scientific Research and Technological Renovation (DG-RSRT), Tunisia, under the ARUB program 01/UR/11-02.

REFERENCES

- [1] J. Borenstein, "Experimental Results from Internal Odometry Error Correction with the Omnimate Mobile Robot," *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 6, 1998, pp. 963-969. [doi:10.1109/70.736779](https://doi.org/10.1109/70.736779)
- [2] B.-Y. Chee, S. Lang and P. Tse, "Fuzzy Mobile Robot Navigation and Sensor Integration," *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, Vol. 1, 1996, pp. 7-12. [doi:10.1109/FUZZY.1996.551711](https://doi.org/10.1109/FUZZY.1996.551711)
- [3] A. Ramirez-Serrano and M. Boumedine, "Real-Time Navigation in Unknown Environments Using Fuzzy Logic and Ultrasonic Sensing," Dearborn, 15-18 September 1996, pp. 26-30.
- [4] D. Springer, J. Zhang and A. Knoll, "Integrating Deliberative and Reactive Strategies via Fuzzy Modular Control," 1999. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.5267>
- [5] J. Zhang, F. Wille and A. Knoll, "Fuzzy Logic Rules for Mapping Sensor Data to Robot Control," *Proceedings of the First Euromicro Workshop on Advanced Mobile Robot*, Vol. 10, No. 2, 1996, pp. 29-38. [doi:10.1109/EURBOT.1996.551878](https://doi.org/10.1109/EURBOT.1996.551878)
- [6] J. Mendel and R. John, "Type-2 Fuzzy Sets Made Simple," *IEEE Transactions on Fuzzy Systems*, Vol. 10, No. 2, 2002, pp. 117-127. [doi:10.1109/91.995115](https://doi.org/10.1109/91.995115)
- [7] N. Karnik, J. Mendel and Q. Liang, "Type-2 Fuzzy Logic Systems," *IEEE Transactions on Fuzzy Systems*, Vol. 7, No. 6, 1999, pp. 643-658. [doi:10.1109/91.811231](https://doi.org/10.1109/91.811231)
- [8] J. Mendel and R. John, "A Fundamental Decomposition of Type-2 Fuzzy Sets," *IFSA World Congress and 20th NAFIPS International Conference*, Vol. 4, 2001, pp. 1896-1901.
- [9] R. I. John, P. R. Innocent and M. R. Barnes, "Neuro-Fuzzy Clustering of Radiographic Tibia Image Data Using Type 2 Fuzzy Sets," *Information Sciences*, Vol. 125, No. 1-4, 2000, pp. 65-82. [doi:10.1016/S0020-0255\(00\)00009-8](https://doi.org/10.1016/S0020-0255(00)00009-8)
- [10] R. John and S. Lake, "Type-2 Fuzzy Sets for Modelling Nursing Intuition," *IFSA World Congress and 20th NAFIPS International Conference*, Vol. 4, 2001, pp. 1920-1925.
- [11] N. N. Karnik and J. M. Mendel, "Applications of Type-2 Fuzzy Logic Systems to Forecasting of Time-Series," *Information Sciences*, Vol. 120, No. 1-4, 1999, pp. 89-111. [doi:10.1016/S0020-0255\(99\)00067-5](https://doi.org/10.1016/S0020-0255(99)00067-5)
- [12] Q. Liang and J. Mendel, "Equalization of Nonlinear Time-Varying Channels Using Type-2 Fuzzy Adaptive Filters," *IEEE Transactions on Fuzzy Systems*, Vol. 8, No. 5, 2000, pp. 551-563. [doi:10.1109/91.873578](https://doi.org/10.1109/91.873578)
- [13] Q. Liang and J. M. Mendel, "Modeling mpeg vbr Video Traffic Using Type-2 Fuzzy Logic Systems," In: *Granular Computing: An Emerging Paradigm*, Springer-Verlag, Berlin, 2001, pp. 367-383.
- [14] C. Lynch, H. Hagrais and V. Callaghan, "Embedded Type-2 FLC for Real-Time Speed Control of Marine and Traction Diesel Engines," *The 14th IEEE International Conference on Fuzzy Systems*, Reno, 22-25 May 2005, pp. 347-352
- [15] D. Wu and W. Tan, "A Type-2 Fuzzy Logic Controller for the Liquid-Level Process," *IEEE Proceedings of International Conference on Fuzzy Systems*, Vol. 2, 2004, pp. 953-958.
- [16] T. Dereli, A. Baykasoglu, K. Altun, A. Durmusoglu and I. Türksen, "Review: Industrial Applications of Type-2 Fuzzy Sets and Systems: A Concise Review," *Computers in Industry*, Vol. 62, No. 2, 2011, pp 125-137. [doi:10.1016/j.compind.2010.10.006](https://doi.org/10.1016/j.compind.2010.10.006)
- [17] L. A. Zadeh, "The Concept of a Linguistic Variable and Its Application to Approximate Reasoning-i," *Information Sciences*, Vol. 8, No. 3, 1975, pp. 199-249. [doi:10.1016/0020-0255\(75\)90036-5](https://doi.org/10.1016/0020-0255(75)90036-5)
- [18] N. N. Karnik and J. M. Mendel, "Operations on Type-2 Fuzzy Sets," *Fuzzy Sets and Systems*, Vol. 122, No. 2, 2001, pp. 327-348. [doi:10.1016/S0165-0114\(00\)00079-8](https://doi.org/10.1016/S0165-0114(00)00079-8)
- [19] N. N. Karnik and J. M. Mendel, "Centroid of a Type-2 Fuzzy Set," *Information Sciences*, Vol. 132, No. 1-4, 2001, pp. 195-220. [doi:10.1016/S0020-0255\(01\)00069-X](https://doi.org/10.1016/S0020-0255(01)00069-X)
- [20] J. M. Mendel, R. I. John and F. Liu, "Interval Type-2 Fuzzy Logic Systems Made Simple," *IEEE Transactions on Fuzzy Systems*, Vol. 14, No. 6, 2006, pp. 808-821. [doi:10.1109/TFUZZ.2006.879986](https://doi.org/10.1109/TFUZZ.2006.879986)
- [21] H. Hagrais, "Type-2 FLCs: A New Generation of Fuzzy Controllers," *IEEE Computational Intelligence Magazine*, Vol. 2, No. 1, 2007, pp. 30-43. [doi:10.1109/MCI.2007.357192](https://doi.org/10.1109/MCI.2007.357192)
- [22] R. John and S. Coupland, "Type-2 Fuzzy Logic: A Historical View," *IEEE Computational Intelligence Magazine*, Vol. 2, No. 1, 2007, pp. 57-62. [doi:10.1109/MCI.2007.357194](https://doi.org/10.1109/MCI.2007.357194)
- [23] S. Coupland and R. John, "New Geometric Inference Techniques for Type-2 Fuzzy Sets," *International Journal of Approximate Reasoning*, Vol. 49, No. 1, 2008, pp. 198-211. [doi:10.1016/j.ijar.2008.03.001](https://doi.org/10.1016/j.ijar.2008.03.001)
- [24] N. Baklouti and A. M. Alimi, "The Geometric Interval Type-2 Fuzzy Logic Approach in Robotic Mobile Issue," *IEEE International Conference on Fuzzy Systems*, Jeju Island, 20-24 August 2009, pp. 1971-1976.

- [doi:10.1109/FUZZY.2009.5277307](https://doi.org/10.1109/FUZZY.2009.5277307)
- [25] H. Hagrass, "A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots," *IEEE Transactions on Fuzzy Systems*, Vol. 12, No. 4, 2004, pp. 524-539. [doi:10.1109/TFUZZ.2004.832538](https://doi.org/10.1109/TFUZZ.2004.832538)
- [26] N. Baklouti and A. M. Alimi, "Motion Planning in Dynamic and Unknown Environment Using an Interval Type-2 Fuzzy Logic Controller," *IEEE International Conference on Fuzzy Systems*, London, 23-26 July 2007, pp. 1-6. [doi:10.1109/FUZZY.2007.4295647](https://doi.org/10.1109/FUZZY.2007.4295647)
- [27] J. Figueroa, J. Posada, J. Soriano, M. Melgarejo and S. Rojas, "A Type-2 Fuzzy Controller for Tracking Mobile Objects in the Context of Robotic Soccer Games," *The 14th IEEE International Conference on Fuzzy Systems*, Reno, 22-25 May 2005, pp. 359-364.
- [28] P. Phokharatkul and S. Phaiboon, "Mobile Robot Control Using Type-2 Fuzzy Logic System," *2004 IEEE Conference on Robotics, Automation and Mechatronics*, Vol. 1, 2004, pp. 296-299.
- [29] K. C. Wu, "Fuzzy Interval Control of Mobile Robots," *Computers and Electrical Engineering*, Vol. 22, No. 3, 1996, pp. 211-229. [doi:10.1016/0045-7906\(95\)00038-0](https://doi.org/10.1016/0045-7906(95)00038-0)
- [30] KTEAM, 2007. <http://www.k-team.com>
- [31] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 15, No. 1, 1985, pp. 116-132. [doi:10.1109/TSMC.1985.6313399](https://doi.org/10.1109/TSMC.1985.6313399)
- [32] Q. Liang and J. Mendel, "An Introduction to Type-2 Fuzzy Logic Systems," *1999 IEEE International Conference Fuzzy Systems*, Vol. 3, 1999, pp. 1534-1539.
- [33] Autonomous Mobile Robotics Toolbox SIMROBOT. <http://www.uamt.feec.vutbr.cz/robotics/simulations/amrt/simrobot.zip>
- [34] G. Benet, F. Blanes, J. E. Simy and P. Párez, "Using Infrared Sensors for Distance Measurement in Mobile Robots," *Robotics and Autonomous Systems*, Vol. 40, No. 4, 2002, pp. 255-266.