Applied Soft Computing 11 (2011) 4677–4691

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Enhanced WiFi localization system based on Soft Computing techniques to deal with small-scale variations in wireless sensors

Jose M. Alonso[a,*], Manuel Ocaña[b], Noelia Hernandez[b], Fernando Herranz[b], Angel Llamazares[b], Miguel A. Sotelo[c], Luis M. Bergasa[b], Luis Magdalena[a]

[a] *European Centre for Soft Computing, Mieres, Asturias, Spain*
[b] *Department of Electronics, University of Alcalá, Madrid, Spain*
[c] *Department of Automation, University of Alcalá, Madrid, Spain*

ARTICLE INFO

Article history:
Received 31 May 2010
Received in revised form 20 May 2011
Accepted 7 July 2011
Available online 27 July 2011

Keywords:
Wireless localization
WiFi signal strength sensor
Fuzzy logic
Fuzzy modeling

ABSTRACT

The framework of this paper is robot localization inside buildings by means of wireless localization systems. Such kind of systems make use of the Wireless Fidelity (WiFi) signal strength sensors which are becoming more and more useful in the localization stage of several robotic platforms. Robot localization is usually made up of two phases: training and estimation stages. In the former, WiFi signal strength of all visible Access Points (APs) are collected and stored in a database or WiFi map. In the latter, the signal strengths received from all APs at a certain position are compared with the WiFi map to estimate the robot location. Hence, WiFi localization systems exploit the well-known path loss propagation model due to large-scale variations of WiFi signal to determine how closer the robot is to a certain AP. Unfortunately, there is another kind of signal variations called small-scale variations that have to be considered. They appear when robots move under the wavelength λ. In consequence, a chaotic noise is added to the signal strength measure yielding a lot of uncertainty that should be handled by the localization model. While lateral and orientation errors in the robot positioning stage are well studied and they remain under control thanks to the use of robust low-level controllers, more studies are needed when dealing with small-scale variations. Moreover, if the robot can not use a robust low-level controller because, for example, the environment is not organized in perpendicular corridors, then lateral and orientation errors can be significantly increased yielding a bad global localization and navigation performance. The main goal of this work is to strengthen the localization stage of our previous WiFi Partially Observable Markov Decision Process (POMDP) Navigation System with the aim of dealing effectively with small-scale variations. In addition, looking for the applicability of our system to a wider variety of environments, we relax the necessity of having a robust low-level controller. To do that, this paper proposes the use of a Soft Computing based system to tackle with the uncertainty related to both the small-scale variations and the lack of a robust low-level controller. The proposed system is actually implemented in the form of a Fuzzy Rule-based System and it has been evaluated in two real test-beds and robotic platforms. Experimental results show how our system is easily adaptable to new environments where classical localization techniques can not be applied since the AP physical location is unknown.

1. Introduction

Several applications like surveillance tasks require a priori knowledge of the user location. This position can be determined by the user's device or by the environment itself. By knowing the user position it is possible to interact with him, guiding it through the environment and implementing some tasks depending on the area of interest.

Localization is currently applied at several areas. For instance, there are projects that use localization systems in hospitals which can locate doctors and equipment. Other systems are used for medical assistance [21], inventory control at warehouses, robotics [40], etc.

In the last years, applications of localization systems are growing by means of using different technologies [27]. A great example is GPS (Global Positioning System) [12], which is the most extended technology for devices localization. As an example of the localization importance car drivers usually use GPS to be guided through

* Corresponding author. Tel.: +34 985 45 65 45; fax: +34 985 45 66 99.
E-mail addresses: jose.alonso@softcomputing.es (J.M. Alonso), mocana@depeca.uah.es (M. Ocaña), nhernandez@depeca.uah.es (N. Hernandez), fherranz@depeca.uah.es (F. Herranz), allamazares@depeca.uah.es (A. Llamazares), sotelo@aut.uah.es (M.A. Sotelo), bergasa@depeca.uah.es (L.M. Bergasa), luis.magdalena@softcomputing.es (L. Magdalena).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.07.015

cities. This technology can locate devices with an error that varies from centimeters to one hundred meters, but it does not work properly in indoor environments or even in cities with high buildings.

Thus, it is necessary to find a complementary system for such environments. There are some proposals for indoor localization using infrared [42], computer vision [19], ultrasound [34], laser [7], radio frequency (RF) [6], or even cellular communication [33] based systems. Moreover, there is an increasing interest in WiFi localization for these environments using different algorithms, even looking for complementary characteristics of both GPS (outdoor environments) and WiFi (indoor environments) [13].

One of the main advantages of WiFi technology is its quickly growing degree of coverage. There are WiFi Access Points (APs) in most public buildings like hospitals, libraries, universities, museums, etc. In addition, measuring the WiFi signal level (without transmitting-receiving data) is free even for private WiFi networks. Consequently, WiFi technology is a good choice for global indoor localization systems.

WiFi localization systems use 802.11b/g network infrastructure to estimate a device position. This fact makes WiFi localization systems appropriate to be used in indoor environments where traditional techniques do not work properly. With the aim of estimating a device position, a WiFi localization system measures and processes the received signal level (SL) from each AP by means of a WiFi interface. Notice that, SL depends on the distance and the obstacles between APs and the receiver. Looking for indoor localization, the so-called signal strength approaches are very attractive because they can be applied to wireless networks without needing additional specific hardware [11].

There are two main techniques to estimate an unknown position: deterministic and probabilistic. In the first one, the environment is usually divided into cells and the position is obtained in the estimation stage comparing the measures with the stored pattern [6,44]. On the other hand, probabilistic techniques keep a probabilistic distribution over all positions [15,20]. The last technique gets a better accuracy but with a higher computational cost.

In work [24], the authors estimate the distance to each AP using only odometric calculus and the received SL. They consider trilateration with a propagation model and also a probabilistic approach that applies the Bayes rule to accumulate localization probability. Unfortunately, RF signal is affected by reflection, refraction and diffraction in indoor environments. This effect, known as multipath effect, turns the SL into a complex function of the distance [11]. In addition, classical trilateration algorithms can not be applied when the exact AP physical location is unknown.

Looking for a solution to this problem, authors of [6] proposed a WiFi localization system based on a priori radio map, which stored the received SL of each AP belonging to interest locations. This system has two stages: training and estimation stages. In the first one, a manual radio map is built. While in the estimation stage a vector with received SL of each AP is created and compared with the radio map to obtain the estimated position.

Notice that WiFi technology works at a 2.4 GHz frequency, which is closer to the water resonant frequency, therefore SL is affected by so many variations. One of these variations, studied by the authors in a previous work [40], is the small-scale one and it occurs when the robot moves in a small distance under the wavelength $\lambda = 12.5$ cm. As a result, there are significant changes in the average SL and make difficult to estimate the correct location because they can be up to 10 dBm around the same position. To deal with this, authors proposed the use of a robust low-level controller which integrates WiFi and ultrasound measures in a global navigation system. It is able to handle small-scale problems but only when the environment is organized in perpendicular corridors. Otherwise, the uncertainty level with respect to the measures is so huge that many localization errors appear yielding a bad global navigation performance.

Since we would like to apply our localization system to a wider variety of indoor environments, we should relax the necessity of having a robust low-level controller. In consequence, we have to look for another way of tackling with the intrinsic uncertainty attached to the system. To do so, this work proposes the use of a Fuzzy Rule-based System (FRBS) able to improve the localization stage of our previous navigation system [30] when the robust low-level controller can not be used.

The rest of the paper is organized as follows. The next section introduces the localization system we want to enhance, highlighting its main advantages and drawbacks. In addition, Section 2 presents several related works regarding the applicability of Soft Computing approaches in the context of WiFi localization systems. Then, Section 3 describes our proposal of fuzzy system for dealing with small-scale variations during the localization stage. Section 4 presents the results obtained in experiments carried out with two real prototypes in two different test-bed environments. Finally, Section 5 draws some conclusions and future works.

2. Related works

The main goal of this work is to strengthen the localization stage of our previous WiFi Partially Observable Markov Decision Process (POMDP) Navigation System with the aim of dealing effectively with small-scale variations even in challenging environments where the use of our robust low-level controller is not feasible or it yields really bad performance. Let us summarize our previous proposal which represents the starting point for this work.

2.1. Advantages and drawbacks of our previous work

For a global navigation system, in which the objective is to guide a robot to a goal room in a semi-structured environment, a topological discretization is appropriate to facilitate the planning and learning tasks. It is especially indicated when the environment is very large because it uses a discretization of the environment and divides it in a priori known nodes. With this kind of representation, POMDP models provide solutions to localization, planning, and learning in the global robotics navigation context. These models use probabilistic reasoning to deal with sensor and action uncertainties. It is important to highlight, that robot needs a low-level controller to move across the nodes and perform local navigation actions commanded by the POMDP planner. In this context, using sensors with high uncertainty, like WiFi signal strength sensors, Markov models become the most extended models in order to build a robust global navigation system.

When a robot moves across an environment executing several actions (a_t), in execution step t, and the environment observation is free of uncertainty, the system can be modelized as a Markov Decision Process (MDP). The MDP is a mathematical model that allows the characterization of robotic systems without noise in the environment observation. The MDP considers that only the effect of actions has uncertainty. In addition, when a MDP achieves some execution steps and it goes along different states ($s_0, s_1, \ldots, s_n$) executing some actions ($a_0, a_1, \ldots, a_n$), the probability of being in a state (s_{t+1}) in the execution step $t+1$ is computed by Eq. (1).

$$p(s_{t+1}|s_0, a_0, s_1, a_1, \ldots, s_t, a_t) = p(s_{t+1}|s_t, a_t) \quad (1)$$

The action uncertainty model represents the real errors or failures in the execution of the actions. The transition function T incorporates this information to the MDP. In the discrete case, T is a matrix that represents the probability of reaching the state s_{t+1} when the robot is in the state s_t and it has executed the action a_t. There is a reward function R for each state s and action a. The robot gets the maximum value of the reward function when it reaches the target state travelling through the ideal trajectory and executing the ideal actions.

Although MDP considers that the environment observation is free of uncertainty, in real robotic systems, there are some uncertainties associated to their sensors observations. They are more significant when the observations are provided by the noisy WiFi sensor and when robots move under wavelength. An MDP able to characterize systems with noisy sensors or partial observability is called POMDP. In our previous work [30], we used two different kinds of partial observations: WiFi signal strength and ultrasound sensor. A POMDP is a mathematical model defined by the following elements:

- The same elements included in an MDP: S (states set), A (actions set), T (transition function), and R (reward function).
- The following additional elements: O (observations set ($o \in O$)) and ν (observation function).

Due to the uncertainty of the observations, a POMDP does not know what its real state is. It maintains a Belief Distribution (*Bel*) to solve it. The distribution $Bel(s)$ assigns to each state s a probability which reports the possibility of being the real state. This is the main reason to divide the control stage of a POMDP in two stages:

- State estimator: The input of this block is the current *Observation* and its output is the *Bel* distribution. This block calculates the probability over all possible states.
- Policy: The input of this block is the current *Bel* and its output is the *Action* to perform. This block obtains the optimal action to perform in the next execution step to maximize the reward (R).

The state estimator block corresponds to the global localization system. It updates the *Bel* distribution when a new action or observation is carried out. In the robotics context, these conditions usually are simultaneous. When an action a is executed then a new observation o is taken. Thus, the new probabilities are computed by Eq. (2).

$$Bel_t(s\prime) = \eta \cdot p(o|s\prime) \cdot \sum_{s \in S} p(s\prime|s, a) \cdot Bel_{t-1}(s), \quad \forall s\prime \in S \tag{2}$$

To achieve a high performance of the global navigation system we developed a robust low-level controller [40] with three main goals: (1) Execute the action commanded from the POMDP; (2) inform it when a state transition is detected; and (3) place the robot in the optimal location to measure the WiFi signal.

As a result of using the robust low-level controller, when the robot is placed under a half wavelength (small-scale range), the uncertainty of the signal measure is significantly reduced. However, the robust low-level controller is effective only when there is a mathematical model that robustly reconstructs the geometry of the environment. Unfortunately, such model is only available for environments which are organized in the form of perpendicular corridors. In that case, an H-shape model is adequate to represent the real geometry of each corridor. For that purpose, the width of the corridor W needs to be a priori known (based on the map) or on-line estimated (see Fig. 1).

When the environment is organized in perpendicular corridors of known width, the robust low-level controller is able to obtain a positioning error under a half of the wavelength in almost 70% of the cases. But sometimes, there are environments, like the European Centre for Soft Computing (see Section 4.1), where it is not possible to use this robust low-level controller. In this kind of environments, it is needed to improve the global navigation system by mean of strengthening the global localization system.

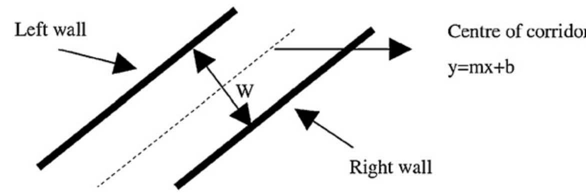


Fig. 1. H-shape corridor model.

In the next section, we explore some Soft Computing approaches for WiFi localization with the aim of finding out a method able to get good results in all kind of environments, so relaxing the need for running a sophisticated low-level controller.

2.2. Soft Computing approaches for wireless localization

Soft Computing (SC) is usually defined by its essential properties, as a family of techniques (Fuzzy Logic, Neuro-computing, Probabilistic Reasoning, Evolutionary Computation, and their hybridizations), as a complement of hard computing, and/or as a tool for coping with imprecision and uncertainty [22]. One of the main issues regarding SC techniques is their complementary and cooperative nature. Each individual technique, even each individual algorithm, has its own advantages and drawbacks. Therefore, designing hybrid systems made up of different techniques working together let us achieving more powerful systems, overcoming the problems which turn up when dealing with the component techniques alone. That is why hybrid systems like for instance neuro fuzzy systems (NFS) [28] and genetic fuzzy systems (GFS) [9] are becoming more and more popular.

Over the last decade, an extensive research has been done on wireless localization based on Soft Computing techniques. We will give a short overview by enumerating some of the most sounded contributions.

Nerguizian et al. proposed the use of neural networks and fingerprinting to deal with the well-known multipath effect in indoor environments [29]. They introduced a method for mobile robot location following a distance-based approach, i.e., their system output provides X–Y coordinates in a two dimensional map. The network learning is done off-line but it may become computational costly (and even unfeasible) for very large environments. More recently, Outemzabet et al. presented a location system also based on neural networks and fingerprinting. The main novelty arises from the fact that the estimated X–Y position is enhanced first with Kalman filtering [31] and later with particle filtering and a low-cost sensor [32].

On the other hand, Dharne et al. advocated for the use of fuzzy logic [10]. They proposed a FRBS able to get good results while reducing the computation time thanks to the use of a grid-based map describing the environment under consideration. Moreover, they reduce the computational cost by taking into account only significant grid-points. Hence, they follow a topology-based approach instead of a distance-based one. The goal is not finding out the exact X–Y coordinates but giving an approximate position (related to the most likely grid-point) with high confidence. Notice that, fuzzy logic is especially useful to handle problems where the available information is vague, which is the typical situation when working with WiFi signal strength sensors. Fuzzy logic lets us to deal with the uncertainty in the environment and makes possible estimate the device position without a high number of samples [4,5,8].

Finally, Yun et al. [45] proposed a Soft Computing based localization system for outdoor environments. To start with, they generated a genetic fuzzy system for individual localization where the edge weights of each anchor node is first modeled by a fuzzy

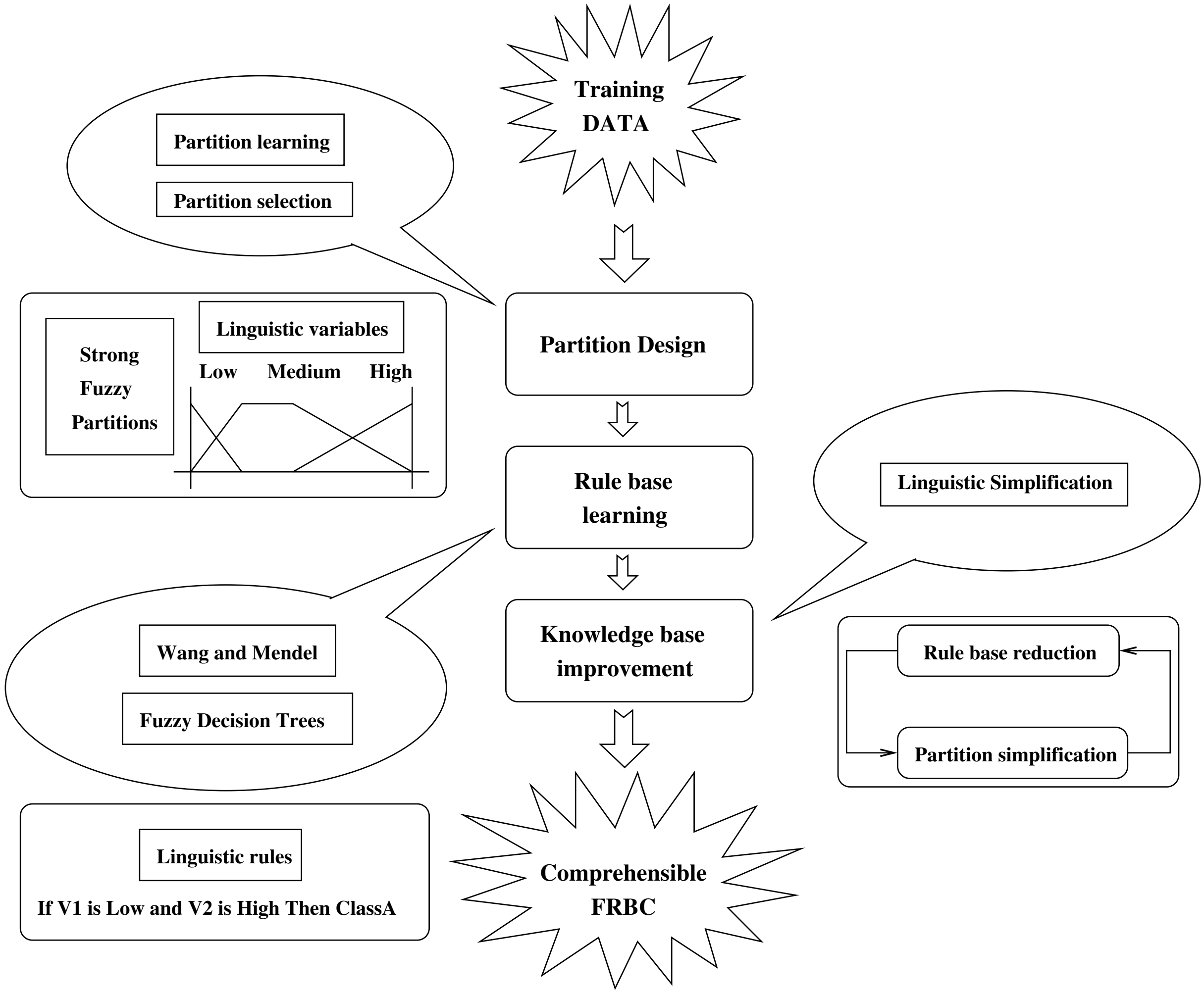


Fig. 2. Scheme of HILK fuzzy modeling methodology.

system and then optimized by a genetic algorithm. Later, they generated a neural network for overall localization. Achieved results were promising and they proved the suitability of considering Soft Computing approaches to deal with wireless localization.

3. Enhancing our localization stage with a fuzzy classifier

This section provides a brief description of the new proposed localization system. As explained in Section 2.1, our previous proposal followed a topology-based approach. We have selected fuzzy logic among all the available Soft Computing techniques because it has been proved to be a good choice (on the light of our previous literature survey) when considering a topological approach.

In classical logic only two crisp values are admissible (false/true, negative/positive, etc), what is a strong limitation when dealing with real-world complex problems where there are many important details which are usually vague. On the contrary, fuzzy logic is a useful tool to handle these problems. In addition, the semantic expressivity of fuzzy logic is well-known to be close to expert natural language yielding powerful tools for linguistic concept modeling. The use of linguistic variables [47] and linguistic rules [23,46] favors the interpretability of fuzzy models, at least from the readability or structural transparency point of view. As a result, fuzzy modeling [14], i.e., system modeling with FRBSs, represents a fruitful research line. Unfortunately, using fuzzy logic is not enough for building interpretable models. The whole modeling process must be carried out carefully, paying special attention to interpretability from the beginning to the end and imposing several constraints [25]. This is the only way for yielding comprehensible models that may be seen as gray boxes where every element of the whole system can be checked and understood by a human being.

We are going to design a topology-based fuzzy system able to handle the SL attenuation due to large-scale variations of WiFi signal with the aim of estimating the robot location among a predefined set of significant positions. Our system is actually implemented in the form of a Fuzzy Rule-based Classification (FRBC) system applied to WiFi localization that is also able to deal with the huge uncertainty derived from small-scale variations when robots work under the wavelength range. Hence, thanks to the use of a fuzzy modeling methodology the designed system will be ready to tackle with the uncertainty related to both the small-scale variations and the lack of a robust low-level controller.

The proposed FRBC has been designed and built using GUAJE [1] a free software tool for generating understandable and accurate fuzzy models. It implements the Highly Interpretable Linguistic Knowledge (HILK) methodology [2,3]. This fuzzy modeling methodology focuses on building comprehensible fuzzy classifiers, i.e., classifiers easily understandable by human beings. As illustrated in Fig. 2, applying fuzzy machine learning techniques HILK is able to automatically extract useful pieces of knowledge from experimental data. Such knowledge is represented by means of linguistic variables and rules under the fuzzy logic formalism.

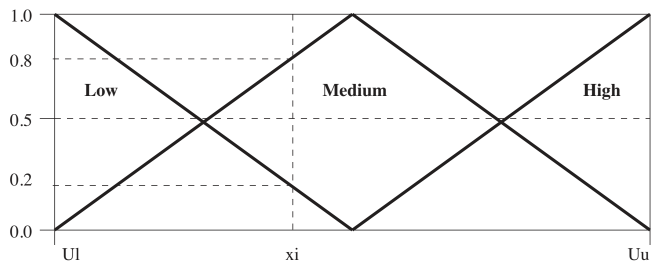


Fig. 3. A strong fuzzy partition with three linguistic terms.

In short, the whole modeling process is made up of three steps:

- **Partition design**. The readability of fuzzy partitioning is a pre-requisite to build interpretable FRBCs. It includes automatic generation of fuzzy partitions from data and partition selection.
- **Rule base learning**. Linguistic rules are automatically extracted from data.
- **Knowledge base improvement**. Iterative refinement process of both partitions and rules.

The use of linguistic variables favors the readability of the model and it becomes a pre-requisite condition to build comprehensible rules. Each system variable is described by a set of linguistic terms, modeled as fuzzy sets represented by membership functions like the ones in Fig. 3. As it can be seen the same value *xi* is partially *Low* (0.2) and *Medium* (0.8), but the addition of both membership degrees equals one. This kind of partition is called Strong Fuzzy Partitions (SFPs) [38] and they are the best ones from the comprehensibility point of view because they satisfy all the semantic constraints [25] (distinguishability, normalization, coverage, overlapping, etc.) demanded to be comprehensible. By default we use SFPs uniformly distributed over the universe of discourse of the variables. Moreover, psychologists [26,39] recommend to work with an odd number of terms (it is easier to make reasoning around a central term) and a small (justifiable) number of terms (7 ± 2 is a limit of human information processing capability).

Once all linguistic variables have been defined with a set of linguistic terms and their associated semantics, they can be used to express linguistic propositions like *Signal received from APi is High*. Then, several propositions are combined to form fuzzy rules describing the system behavior:

$$r: \text{If} \quad \underbrace{\underbrace{X_1 \text{ is } A_1^i}_{Partial\ \ Premise\ \ P_1} \quad \text{AND} \ \ldots \ \text{AND} \quad \underbrace{X_I \text{ is } A_I^j}_{Partial\ \ Premise\ \ P_I}}_{Premise} \quad \text{Then} \ \underbrace{Y_r \text{ is } C^n}_{Conclusion}$$

where given rule r, rule premises are made up of tuples (*input variable*, *linguistic term*) where X_a is the name of the input variable a, while A_a^i represents the label i of such variable, with a belonging to $\{1, \ldots, I\}$ and being I the number of inputs. In the conclusion part, C^n represents one of the possible output classes, i.e., one position in the case of WiFi localization.

For instance, **If** *Signal received from APi is High* **and** *Signal received from APj is Low* **Then** *The robot is close to Position k*. The semantic expressivity of fuzzy logic makes easier the knowledge extraction and representation phase. In addition, it lets us combine under the same formalism knowledge extracted from data and knowledge described by an expert in natural language.

Regarding the rule generation from data, there are lots of methods in the fuzzy literature [17]. However, keeping in mind the comprehensibility goal we have chosen the two following ones to generate rules from data with the previously defined fuzzy partitions:

- **Wang and Mendel** (**WM**) [41]. It starts by generating one rule for each data pair of the training set but new rules will compete with existing ones. As a result, WM generates complete rules (considering all the available variables) which are quite specific.
- **Fuzzy Decision Tree** (**FDT**) [18]. It generates a neuro-fuzzy decision tree (directly from data) which is translated into quite general incomplete rules (only a subset of input variables is considered). In addition, inputs are sorted according to their importance (minimizing the entropy). FDT is a fuzzy version of the popular decision trees defined by Quinlan in [35] and improved in [37].

After defining linguistic variables and rules, HILK offers a powerful and flexible simplification procedure which affects to the whole knowledge base including both rule base simplification and partition reduction. The goal is getting a more compact and general FRBC, keeping high accuracy while increasing even more comprehensibility. It starts looking for redundant elements (terms, variables, rules, etc.) that can be removed without altering the system accuracy. Then, it tries to merge elements always used together. Finally, it forces removing elements apparently needed but not contributing too much to the final accuracy.

The output of the FRBC will be one position along with an activation degree computed as the result of a fuzzy inference that takes into account all defined variables and rules. We have selected the usual fuzzy classification structure with the Max-Min inference scheme, and the winner rule fuzzy reasoning mechanism.

First, given an input vector $x^p = \{x_1^p, \ldots, x_I^p\}$, the firing degree (for each rule r) is computed as the minimum membership degree of x^p to all the attached A_i^j fuzzy sets, for all the I inputs:

$$\mu_r(x^p) = \min_{i=1,\ldots,I} \mu_{A_i^j}(x_i^p) \tag{3}$$

Then, the output class C^i is derived from the highest $\mu_{C^i}(x^p)$ (look at Eq. (4)) which is the membership degree of x^p to the class C^i. It is computed as the maximum firing degree of all rules yielding C^i as output class (look at Eq. (5)). Notice that, several output classes can be activated since several fuzzy rules can be fired at the same time by the same input vector.

$$y_{FRBC}(x^p) = C^i \Leftrightarrow \mu_{C^i}(x^p) = \max_{n=1,\ldots,C} \mu_{C^n}(x^p) \tag{4}$$

$$\mu_{C^n}(x^p) = \max_{r=1,\ldots,R} \mu_r(x^p) \Leftrightarrow Y_r \text{ is } C^n \tag{5}$$

As an illustrative example, if the system output says that the robot is in position A with degree 0.2 and in position B with degree 0.8, it can be concluded that it is located somewhere between A and B but closer to B. However, we do not know the exact location because we follow a topological approach but not a distance-based approach.

Thanks to its flexibility and adaptability the designed FRBC can be used in whatever environment disregarding its specific geometry. Moreover, the FRBC achieves good results whenever the environment does not suffer a great modification, i.e., when some Access Points are switched off. In such case, the system should be re-adjusted, but usually these things do not happen and the fuzzy system is able to deal with slight modifications like people moving in the environment or changes in the state of the doors (open/close).

4. Experimental analysis

Exhaustive experiments have been carried out with two different robotic platforms in two different test-bed environments. The interested reader is referred to Appendix A for more technical details about the robots. This section presents the main set-up

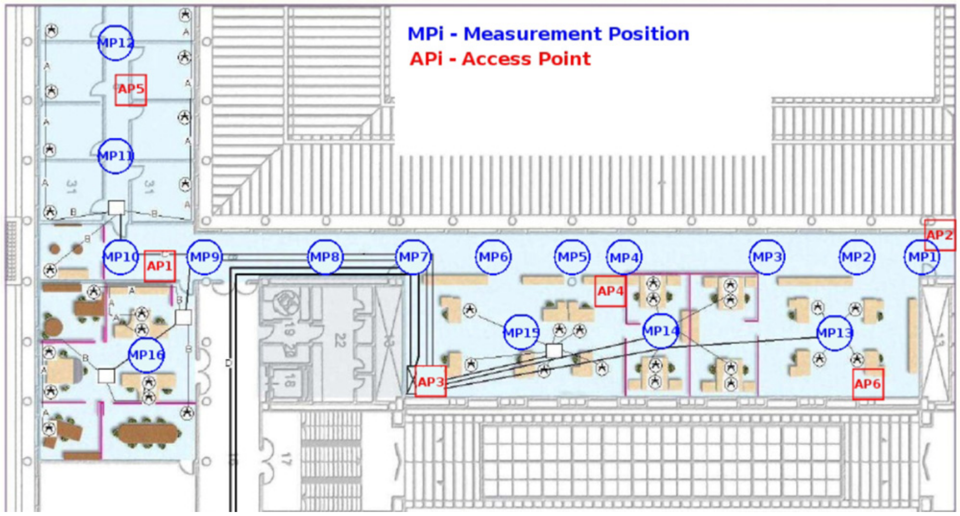


Fig. 4. ECSC environment.

of the two test-beds along with the description of the experimental process and a critical discussion regarding the main obtained results.

4.1. Test-beds

Two different test environments were established at the premises of the European Centre for Soft Computing (ECSC) and the Polytechnic School of the University of Alcalá (UAH):

- The layout of ECSC test-bed is shown in Fig. 4. It has a surface of about 500 m^2 with 15 offices, one main long corridor, and two large open working areas. There are 6 APs distributed over the whole environment which is discretized into 16 significant topological positions (MPi) to be distinguished by the designed fuzzy localization system.
- The layout of UAH environment is illustrated in Fig. 5. It has a surface of 3600 m^2 with four laboratories and 32 offices. There are 54 APs distributed over the whole environment. For simplicity, since the layout is symmetrical from the main diagonal as depicted in the figure, the developed system will be tested in this environment only in three corridors: main, third and fourth corridor. We have considered only 6 APs visible over the analyzed scenario which is discretized into 9 significant topological positions (MPi) to be distinguished by the designed fuzzy localization system.

4.2. Experimental process

First of all, we have captured several SL measures (300 samples) corresponding to each fix measurement position MPi in ECSC (look at Fig. 4). Such measures were collected in the center of each topological position with the aim of building a training data set. Then, the methodology presented in Section 3 has been applied to build the proposed fuzzy localization system. As it will be further explained in the next section, we built several systems with different configuration parameters in order to find the best FRBC for the analyzed environment. Of course, the goodness of the generated systems was checked with an independent test set. Moreover, with the aim of evaluating the robustness of the final fuzzy localization system against small-scale variations, the test set is made up of samples collected according to the pattern shown in Fig. 6 which consists of a 12.5 cm × 12.5 cm testing grid divided by 1 cm side squares.

This small-scale testing grid has been used in the experiments to test the robustness of our localization system. To do so, the grid was sequentially placed on top of each MPi and the same process was repeated. Namely, SL test samples were acquired around each MPi with distances in the range of λ. The aim was to prove

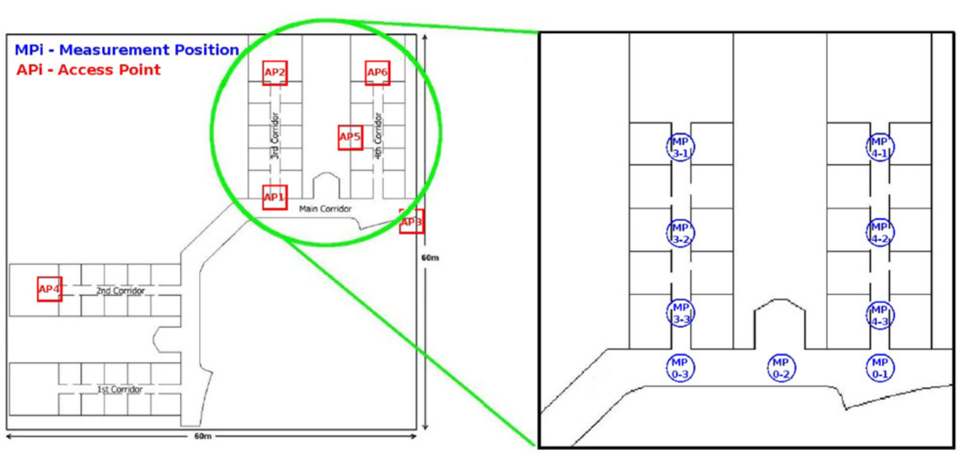


Fig. 5. UAH environment.

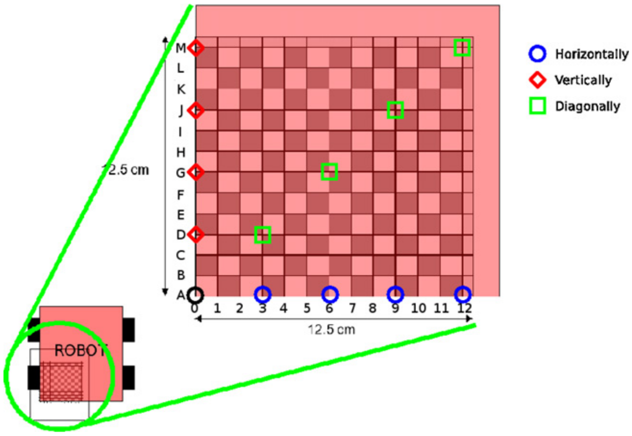


Fig. 6. Small-scale testing grid (robot at position A0).

that our system is able to identify as belonging to the topological location MPi all samples taken around (in the range of small-scale distances) even though training and test samples were not captured exactly in the same *X*–*Y* coordinates. As explained before and it can be appreciated in the illustrative example plotted in Fig. 7, small-scale variations may yield differences up to 10 dBm for very close positions (under the wavelength λ, in the range of cm) around the same MPi. Notice that, a difference about 10 dBm is large enough to induce a misclassification between two distinct locations MPi and MPj separated in the range of 13 meters. Such situation is not desirable and it should be avoided with a robust localization system. In short, the collection of test samples has been done as detailed below:

1. Initially, the device was placed at position A0 (Fig. 6) and 300 samples were collected. This position is taken as the reference position (R).
2. New samples were carried out in three different directions:
   (a) Horizontal: SL is measured on R (A0), R+3 cm (A3), R+6 cm (A6), R+9 cm (A9), R+12 cm (A12) positions. These positions are shown with circles.
   (b) Vertical: SL is measured on R (A0), R+3 cm (D0), R+6 cm (G0), R+9 cm (J0), R+12 cm (M0) positions. These positions are shown with diamonds.

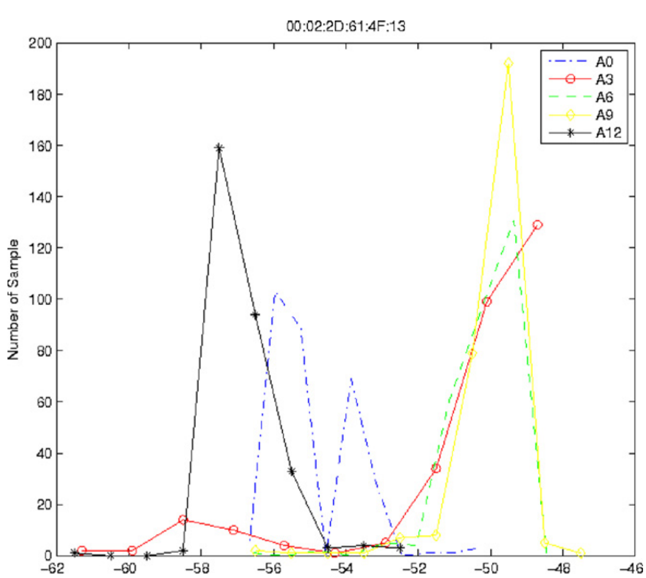


Fig. 7. Example of SL histograms for small-scale variations in horizontal direction.

   (c) Diagonally: SL is measured on R (A0), $R+3\sqrt{2}$ cm (D3), $R+6\sqrt{2}$ cm (G6), $R+9\sqrt{2}$ cm (J9), $R+12\sqrt{2}$ cm (M12) positions. These positions are shown with squares.

4.3. *Discussion about experimental results*

This section describes the experimental results obtained on the designed trials. First, we have focused on the ECSC environment where we have tried several FRBCs with different configuration parameters in order to find out the FRBC yielding the best results. Then, conclusions derived from that preliminary experimentation have been applied to build a FRBC for the UAH premises. In this manner, it is possible to check the extensibility of the proposed system. It is important to highlight that, in our topological approach, it is not necessary to know where APs are exactly located to develop the system. This aspect is especially powerful regarding the deployment of our localization system for a new unknown environment. Finally, a comparison of the best results for ECSC and UAH environments is discussed.

Looking for the best FRBC, the influence of some parameters has been analyzed in the context of ECSC. These parameters are:

1. **Number of input variables**. We have taken measures from six APs regarding both signal (SL) and noise (NL) levels. Thus, we have considered two different situations: 6 inputs (only SL), and 12 inputs (SL and NL).
2. **Number of linguistic terms defined per input**. Following the advices of most psychologists the number of terms should be an odd number smaller or equal than nine. Therefore, with the aim of looking for the most suitable number, four cases have been analyzed: three, five, seven, and nine linguistic terms defined for each input variable.
3. **Rule induction technique**. We have considered the two rule induction algorithms, WM and FDT, introduced in Section 3. The first algorithm (WM) has not any configuration parameters, but for the second one (FDT) two cases are evaluated, the whole tree and the pruned tree (FDTP) with a loss tolerance threshold equal to 0.1. Then, the HILK simplification algorithm (S) has been run for the three analyzed cases (WM, FDT, and FDTP). Hence, a total of six different methods have been tested: Three different rule induction techniques before and after simplification.
4. **Number of averaged samples**. Taking into account that the maximum acquisition frequency of our WiFi interface is 4Hz, the robots are able to capture up to four samples per second. Six cases are evaluated: 1 (raw data without averaging), 4, 12, 28, 40, and 60 averaged samples. The related pre-processing time is not a problem during the training stage of the system because it is made offline. However, it becomes a critical requirement when thinking on the estimation stage which is run online (in real-time).

In summary, we have built 288 ($2 \times 4 \times 6 \times 6$) FRBCs covering all situations described above, and each of them is evaluated with six test data sets yielding a total of 1728 experiments. Following subsections analyze the best options for all the four parameters.

4.3.1. *Number of inputs*

Fig. 8 shows the achieved results during the training stage regarding only SL (six inputs) or considering both SL and NL (12 inputs). All the eight pictures contained in the figure share the same format. On the one hand, vertical axes include the accuracy of the analyzed FRBCs computed as the percentage of correctly classified samples. On the other hand, horizontal axes show the number of averaged samples.

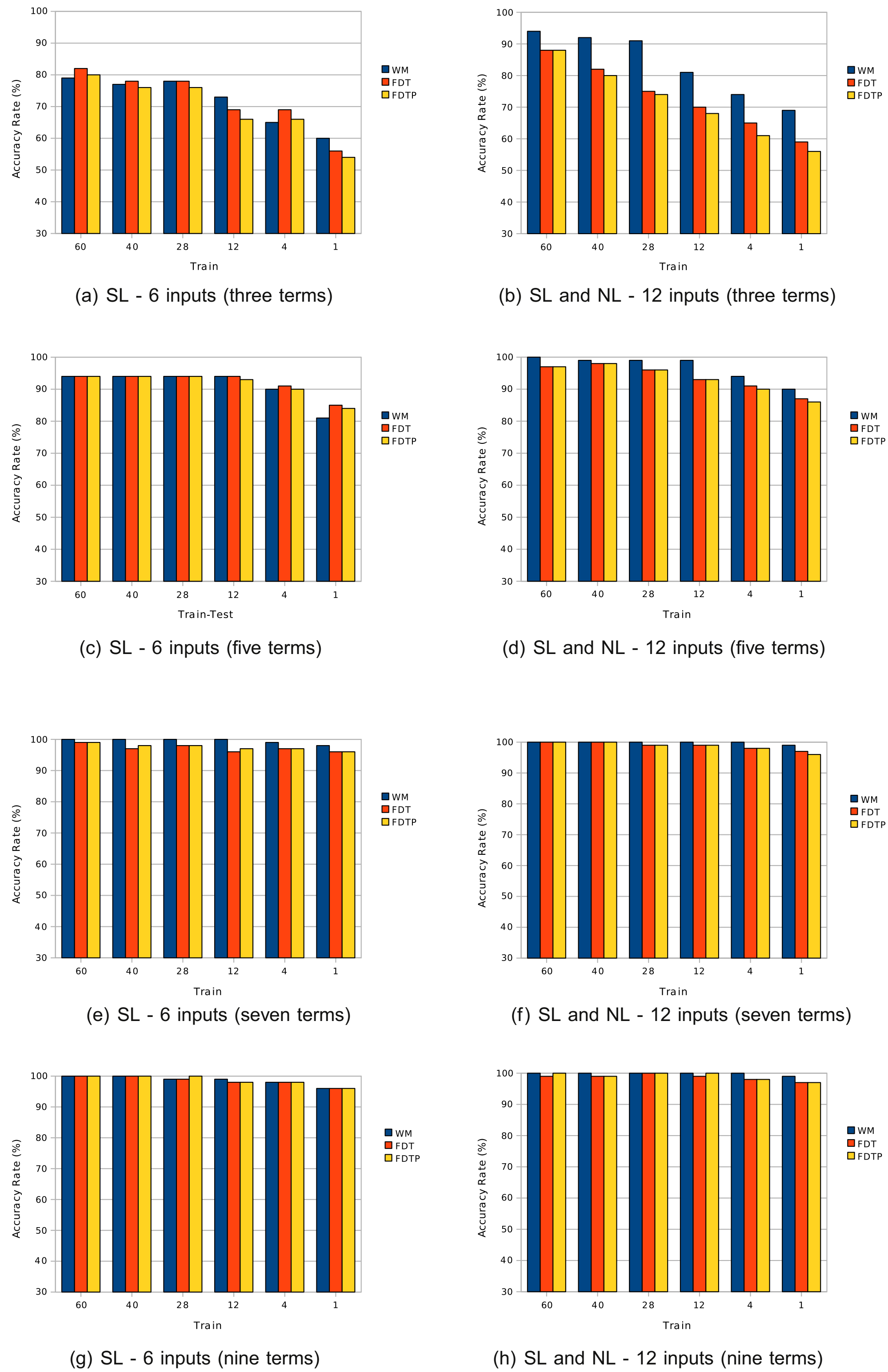


Fig. 8. Comparison with respect to the number of inputs during the training stage. (a) SL – 6 inputs (three terms), (b) SL and NL – 12 inputs (three terms), (c) SL – 6 inputs (five terms), (d) SL and NL – 12 inputs (five terms), (e) SL – 6 inputs (seven terms), (f) SL and NL – 12 inputs (seven terms), (g) SL – 6 inputs (nine terms), and (h) SL and NL – 12 inputs (nine terms).

From those pictures, comparing left and right columns, it is easy to appreciate that computed accuracy is slightly worse when working only with SL. However, results are better using only SL at the test stage what suggests that adding NL there is a kind of overfitting effect. This is deduced from pictures in Fig. 9 where the horizontal axes are slightly different since they include couples of train-test samples. For instance, 12–4 means that the classifier is built considering blocks of twelve averaged samples for training while the number of averaged samples are four for testing.

The decrease in accuracy is produced because NL measures do not follow a particular pattern. That is why the generalization

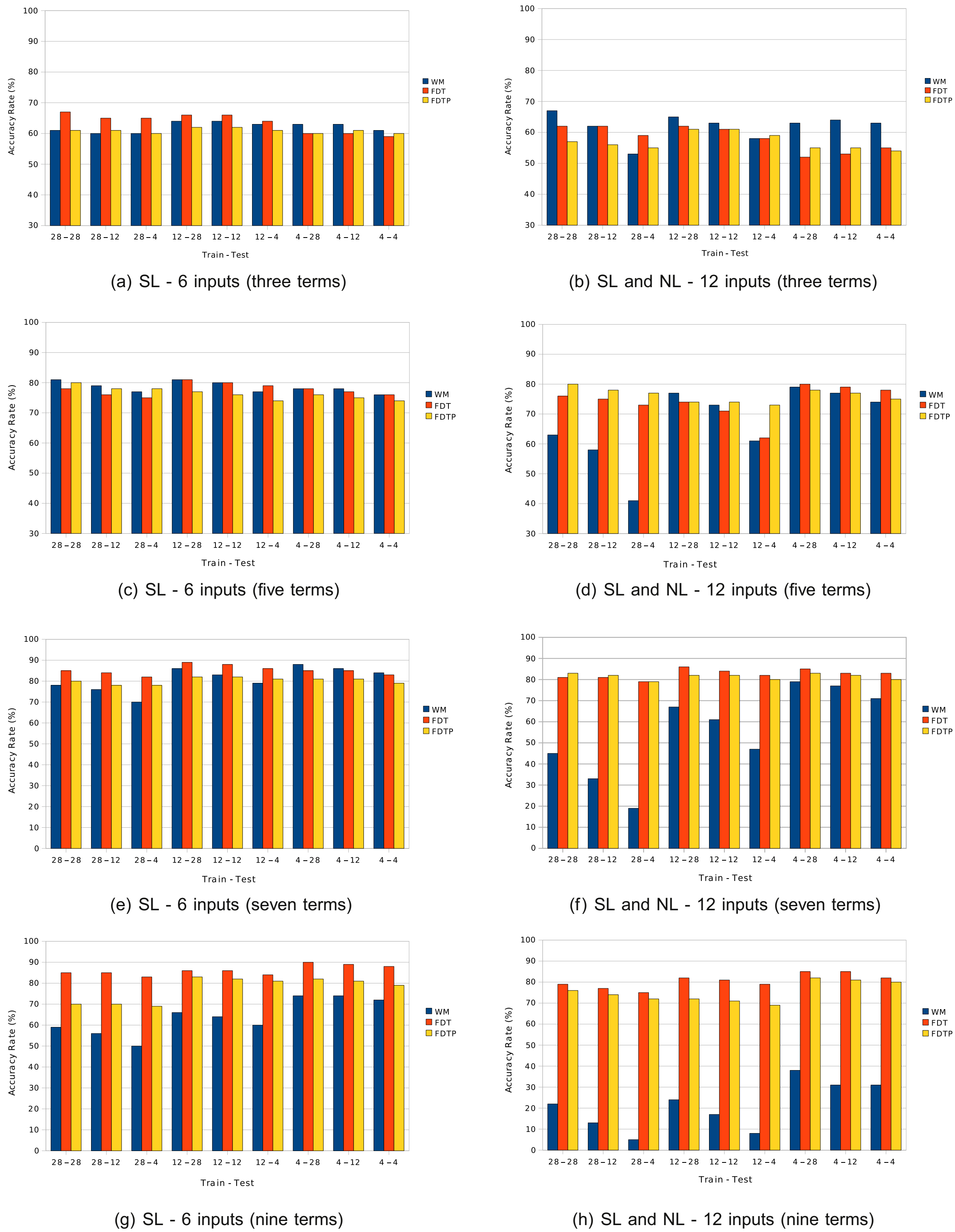


Fig. 9. Comparison with respect to the number of inputs during the test stage. (a) SL – 6 inputs (three terms), (b) SL and NL – 12 inputs (three terms), (c) SL – 6 inputs (five terms), (d) SL and NL – 12 inputs (five terms), (e) SL – 6 inputs (seven terms), (f) SL and NL – 12 inputs (seven terms), (g) SL – 6 inputs (nine terms), and (h) SL and NL – 12 inputs (nine terms).

ability of the classifiers is strongly penalized and accuracy is reduced regarding test data. Moreover, NL varies randomly. In consequence, we observe that NL inputs usually disappear of the rules after simplification for most of the designed classifiers. Therefore, we can conclude that NL does not give any reliable information to design a localization system. Furthermore, increasing the number of inputs makes the system more complex without any advantages, so we can discard NL and build the FRBC regarding only the six inputs for SL.

4.3.2. Number of linguistic terms and rule induction techniques

Figs. 8 and 9 contain much more information than only the number of inputs. The number of linguistic terms varies from three (pictures at the top of the figures) to nine (pictures at the bottom

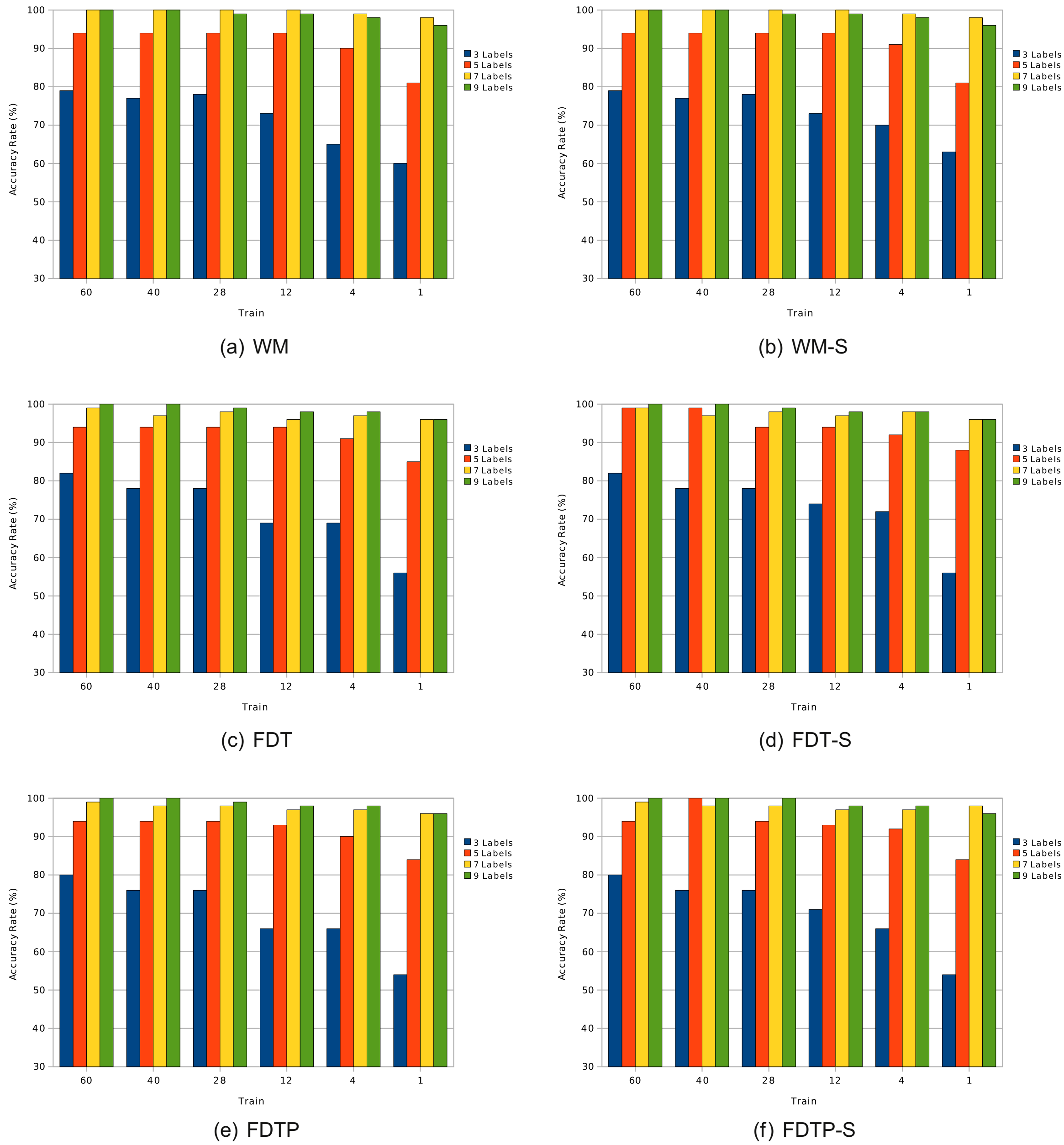


Fig. 10. Comparison of the selected rule induction methods for the training stage. (a) WM, (b) WM-S, (c) FDT, (d) FDT-S, (e) FDTP, and (f) FDTP-S.

of the figures). In addition, we have plotted the accuracy for all the three selected induction methods (WM, FDT, and FDTP).

It is possible to outline some preliminary conclusions. During the training stage, the accuracy is increased when adding more linguistic terms per input what is due to the fact that input space is split into smaller cells thanks to the larger granularity yielding a finer analysis. As a side effect the number of rules is increased. Furthermore rules are more specific and the generalization ability of the FRBCs is reduced depending on the available data as well as on the selected rule induction technique (fuzzy trees provide more general rules than WM for example). That is why accuracy with respect to test data is clearly smaller than regarding training data.

For the sake of clarity and taking advantage of the conclusions derived from the analysis made in the previous section, a more detailed analysis for both number of terms and rule induction algorithms can be made by focusing only on FRBCs with six inputs. The goal is to find out the best combination of both number of terms (3, 5, 7, or 9) and rule induction method with or without simplification (WM, WM-S, FDT, FDT-S, FDTP, and FDTP-S). Figs. 10 and 11 show the comparison of all generated FRBCs regarding both training and test data, for all analyzed combinations. As expected, during the training stage accuracy increases when the number of terms grows up, but results are almost steady from seven labels on. This behavior is not always held on test data where an overfitting effect sometimes appears when passing from seven to nine terms. Such effect is due to the excessive specification of rules when working with a large number of linguistic terms.

After comparing left and right pictures in Fig. 10 it can be deduced that the simplification procedure gets more compact FRBCs keeping (and sometimes increasing) the achieved accuracy during the training stage. Nevertheless, this statement is not always true when looking at test results plotted in Fig. 11. Simplification does not alter accuracy when dealing with WM, but it slightly gets worse accuracy for the fuzzy trees which usually exhibit good

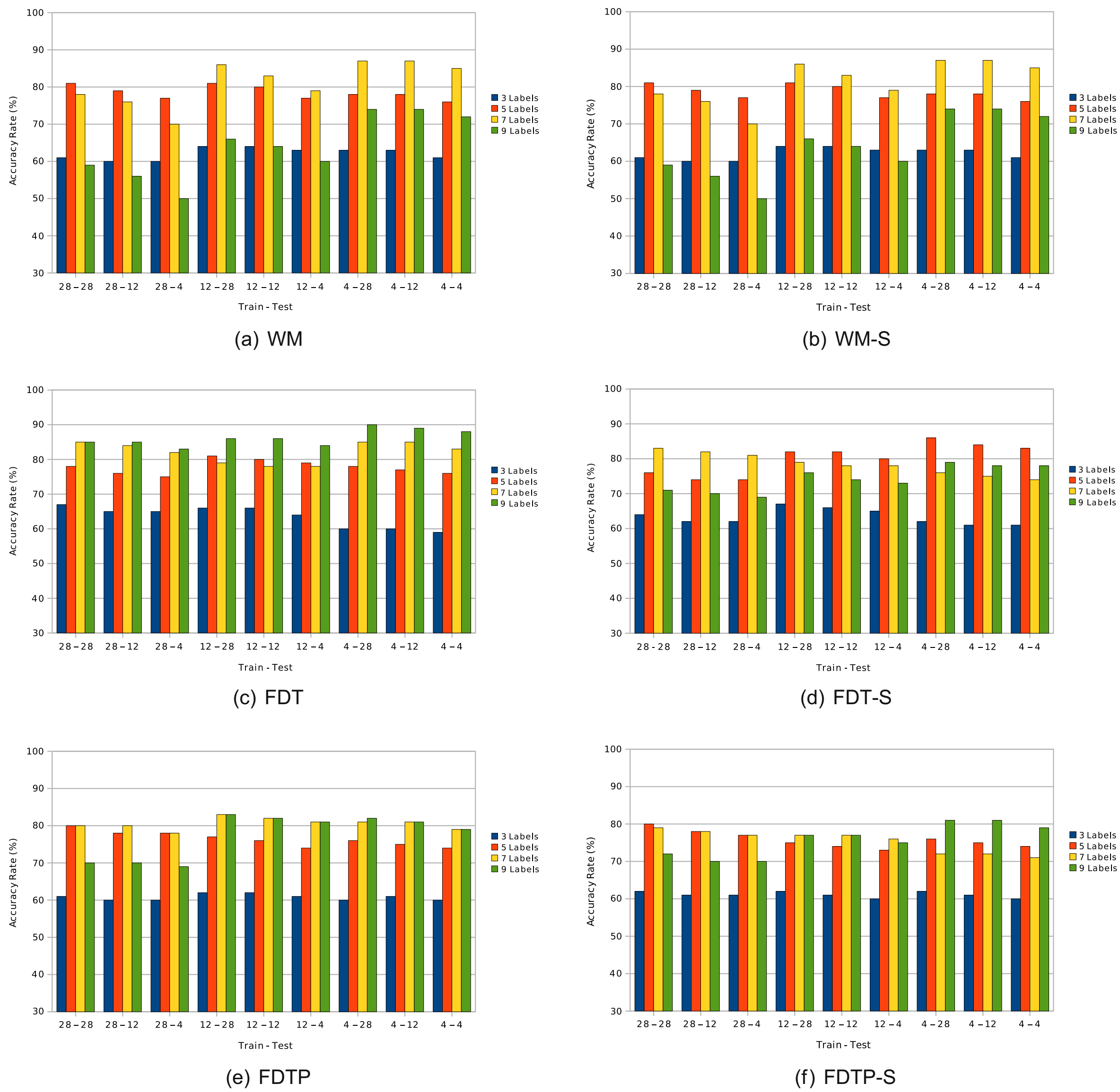


Fig. 11. Comparison of the selected rule induction methods regarding test data. (a) WM, (b) WM-S, (c) FDT, (d) FDT-S, (e) FDTP, and (f) FDTP-S.

generalization ability. Notice that, accuracy and comprehensibility are contradictory goals. The simplification procedure enhances the comprehensibility of the final model but at the cost of losing some accuracy. Of course, depending on the application requirements some accuracy reduction can be admissible in exchange for comprehensibility.

Assuming that all generated FRBCs are comprehensible by superimposing several constraints (SFPs, global semantics, linguistic rules, etc.), the selection of the best FRBC is going to be made giving priority to the accuracy results. Thus, we observe that FDT provides the most accurate FRBCs for both training and test. Moreover, the best results are obtained with nine terms. There are two main reasons for not using more than nine linguistic terms per input variable. First, according to psychologists nine is the largest number of concepts that a human is able to manage at the same time, representing a limit of the human processing capability. Second, a large number of terms leads to overtraining and it may decrease the generalization ability of the model, making more difficult its application to new unknown environments, but it also would be less tolerant to slight modifications (for instance people moving) in the environment where the FRBC was trained.

4.3.3. Number of averaged samples for both training and test

This section is devoted to analyze the influence of the number of averaged samples for both the off-line training stage and the subsequent on-line estimation stage when the localization system is used in a real-time application.

Keeping in mind the previously drawn conclusions, we focus on the best FRBC, i.e., the one built considering six input variables, nine linguistic terms with their associated uniformly distributed strong fuzzy partitions, and linguistic rules automatically generated from training data by means of FDT. Fig. 12 shows how obtained results vary depending on the number of averaged samples at training stage. As expected, the larger the number of averaged samples, the higher accuracy is achieved. The accuracy gets 100% for a number of samples greater or equal to 40. As an effect of averaging, the measured variations are smoother and accuracy is higher but at the cost of a longer acquisition time. Fifteen seconds is the time needed for acquiring 60 samples since the acquisition frequency is equal to 4 Hz. This time can be acceptable for training but it would be too much for testing.

Results on testing are illustrated in Fig. 13. We have compared results achieved by the FRBC trained with 4 averaged samples, and

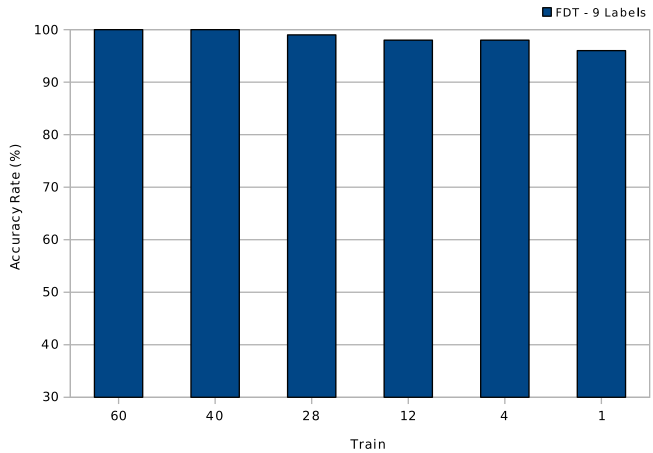


Fig. 12. Effect of the number of averaged samples during the training stage.

then tested against six data sets made up of averaged samples varying from 60 to 1. In this case, it is easy to appreciate that accuracy on testing gets worse when decreasing the number of averaged samples. The most accurate solution is the one obtained when testing with 40 averaged samples, what means 10 s for acquisition time. From a practical point of view, it is desirable an acquisition time as small as possible. Looking at Fig. 13 it seems reasonable to consider only 4 averaged samples, what decreases the acquisition time to 1 second while still keeping a high accuracy around 90 %. It yields a really good trade-off between accuracy and acquisition time.

4.3.4. Evaluation of the final system

In the light of the previous results we have selected the following configuration of parameters as the best one for building the FRBC to be used as part of our localization system: Training with four averaged samples, only considering SL as inputs, with nine terms per input, and fuzzy trees without pruning (FDT) for rule induction.

Table 1 shows results achieved by our localization system (FRBC) after testing it on both ECSC and UAH environments, considering blocks of four averaged test samples. In order to show how simplification jeopardizes accuracy with the aim of improving interpretability, we have also included achieved results by the simplified system (FRBC + S). After simplifying accuracy on training remains almost the same while interpretability is clearly improved since total rule length (computed as the total number of premises in all the rules) drops dramatically. However, the improvement of interpretability is obtained at the cost of a huge reduction of accuracy regarding the test set what is not admissible in the context of our application.

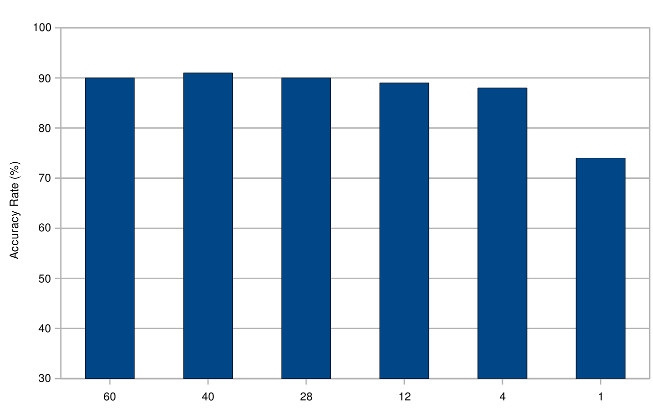


Fig. 13. Effect of the number of averaged samples during the estimation stage.

Table 1
Comparative results.

	FRBC	FRBC + S	FURIA	MP	C4.5
ECSC					
Accuracy-test (%)	87.89	76.82	78.19	83.61	68.69
Accuracy-training (%)	97.67	98.19	100	99.83	99.83
Total rule length	837	60	39	–	72
UAH					
Accuracy-test (%)	89.36	79.15	72.23	84.19	74.73
Accuracy-training (%)	99.11	99.11	99.82	100	99.64
Total rule length	251	43	19	–	30

In addition, for comparison purpose, we have added to Table 1 results provided by other methods implemented in Weka[1] [43]: FURIA (Fuzzy Unordered Rule Induction Algorithm developed by Hühn and Hullermeier [16]), MP (Multilayer Perceptron), and C4.5 algorithm (Qunilan's crisp decision trees [36]). FURIA and MP represent alternative Soft Computing techniques while C4.5 is a well-known classical algorithm recognized because it provides good interpretability-accuracy trade-offs and it is usually taken as baseline for comparisons.

FURIA is a method for building Fuzzy Rule-based Classifiers without taking care of interpretability constraints. It generates compact unordered set of fuzzy rules difficult to interpret in comparison with the linguistic rules provided by HILK, the fuzzy modeling methodology considered in this paper. Anyway, if we compare the reported total rule length then it is obvious that FURIA provides the most compact set of rules. Notice that, we are aware that assessing interpretability is a controversial issue that strongly depends on the subjective opinion of the end-user.

On the other hand, MP consists in a black-box classifier designed as a feed-forward artificial neural network. Notice that, we took the default parameters (learning rate equals 0.3, momentum equals 0.2) suggested by Weka. In consequence, we build neural networks made up of six neurons in the input layer corresponding to the six input variables (one per AP), one hidden layer with seven neurons for UAH and eleven neurons for ECSC (the number of hidden neurons is computed dividing by two the total number of inputs and output classes), and nine (UAH) or sixteen (ECSC) output neurons in the output layer (one for each output class). Since our output is categorical, all the nodes in the generated neural networks are sigmoid.

Looking at results reported in Table 1 we observe that MP is affected by overfitting since it provides the highest accuracy on training set but not regarding test set. On the contrary, our system (FRBC) exhibits the best generalization ability regarding accuracy on test set. Thus, we can say the proposed system is the most robust to small-scale variations since the test set is made up of thousand of samples taken in the small-scale range.

Regarding C4.5, it provides quite compact systems but they are not able to cope properly with small-scale variations. C4.5 gives very low accuracy with respect to the test sets because it is not well fitted to deal with the huge uncertainty attached to test samples.

Although our localization system is based on a topological approach where the system output is aimed to find out the closest MPi reference position instead of giving absolute *X*–*Y* coordinates like other distance-based approaches usually do, we have also reported (just for comparison purpose) the achieved results in terms of cumulative distribution functions (CDF) what is very popular in the specialized literature. Figs. 14 and 15 illustrate the computed CDF for both ECSC and UAH test environments. Notice that, Appendix B gives all details about physical Euclidean dis-

[1] A free software tool for data mining available at: http://www.cs.waikato.ac.nz/ml/weka.

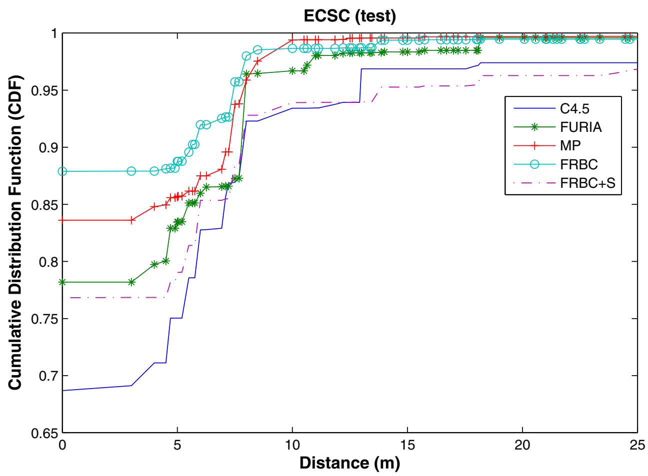


Fig. 14. Cumulative distribution function (ECSC).

tances among MPi for the two test-beds. The pictures should be interpreted as follows. Given a distance of L meters, only misclassifications involving positions MPi and MPj separated by a distance greater than L meters are taken into account when evaluating the error rate. Please, it is worthy to mention that only the trends are important in the figures and not the absolute values because the error strongly depends on the granularity of the nodes in the environment. The global behavior is similar in both scenarios. As expected, most misclassifications correspond to the closest positions. Thus, accuracy arises as the admissible distance for counting a position error is increased. Our system (FRBC) overwhelms all the other methods, especially regarding distances smaller than 5 m. Of course, as the distance increases MP gets closer to FRBC becoming even to be the most accurate method for larger distances (above 10m in the case of ECSC and for more than 16m in the case of UAH). Anyway, the difference between FRBC and MP is almost negligible for large distances. In fact, all methods report high accuracy when considering large distances, but remind that the main goal of this experimentation was checking the robustness of the localization method with respect to small distances.

Finally, we can discuss the goodness of the proposed system in comparison with results reported in [30] where a WiFi + ultrasound POMDP navigation system equipped with a highly accurate and robust low-level controller (WUPOMDPLLC), like the one introduced in Section 2.1, was able to achieve a localization error of only 8% in the UAH environment. From Table 1, it is easy to appreciate how such result clearly outperforms the lowest error rate (10.64%) obtained by the new proposed system for the same environment. Anyway, taking into account the simplicity and generality of the designed FRBC, a lost smaller than 3% may be acceptable. Notice that, the new system is able to estimate the robot location by means of taking only four averaged samples while the old one required sixty averaged samples for the test stage. As a result, we have reduced dramatically the acquisition time (from 15 to 1 s) what makes feasible the use of the new proposed system in real-time applications. Moreover, remind that the new proposed system can be apply no matter the geometry of the environment under study, yielding similar error rates (around 10%) for both ECSC and UAH. However, it is important to remark that our previous WUPOMDPLLC can not be applied to the ECSC environment where the highly accurate and robust low-level controller introduced in [30] was not able to work properly. ECSC contains open areas which are really difficult to characterize. As a result, finding out a mathematical model that reproduces the geometry of the whole environment is not feasible.

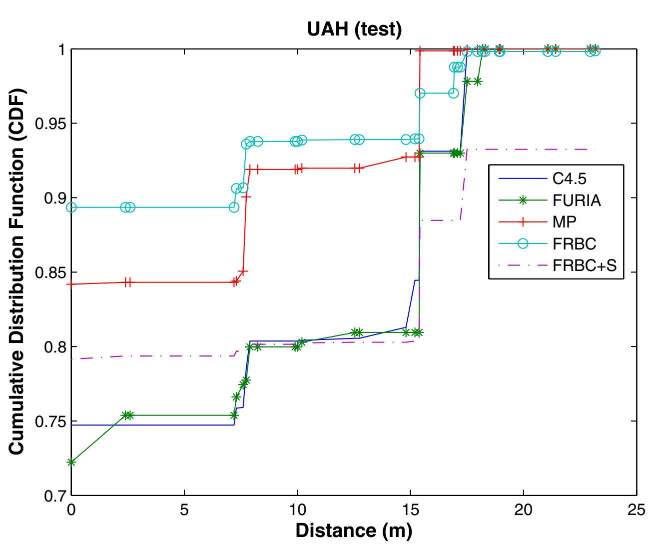


Fig. 15. Cumulative distribution function (UAH).

5. Conclusions

In this paper, we have presented a new WiFi localization system based on an interpretable Fuzzy Rule-based Classifier (FRBC) with the aim of enhancing a previous localization system which was taking as starting point. Our work demonstrates that the designed FRBC is a very useful tool and it is especially indicated to solve the traditional WiFi localization problems. Moreover, the proposed FRBC is able to deal successfully with the small-scale variations that characterize the WiFi signal. These variations introduce a large uncertainty in the received WiFi signal level and they represent the main noise problem in WiFi localization systems. To conclude, the next key points should be remarked:

- The proposed system has been developed with a real robotic platform and test-bed environment and then the conclusions extracted from it have been used for successfully adapting the system to a different test-bed and robotic platform. The results obtained in both experiments have been similar and we have demonstrated the generalization of the system for unknown environments.
- The uncertainty generated by small-scale variations has been overcome with the proposed system and we have obtained accuracy close to 90% with only 4 averaged samples in the test stage. This yields one of the main advantages of this new system, because taking only 4 samples in the test stage, real-time applications could be available since it is not too much time consuming. It is important to remind that four samples imply spending only one second of acquisition time while the fuzzy inference is made in milliseconds.
- The designed system has been compared with the classical and well-known C4.5 algorithm in two environments. Results got for FRBC in both test-beds are better than C4.5 for all the cases. Moreover, the proposed FRBC has turned up as the best one in comparison against other Soft Computing techniques, namely FURIA and MP.
- The proposed system is quite general and easy to adapt to unknown environments. First, it works properly without requiring knowing the exact location of APs. Second, it achieves very close error rates (around 10%) in two very different environments.
- Although the new system is slightly less accurate than the one published in our previous work [30] when dealing with UAH, an

environment that is fully organized in the form of perpendicular corridors, it also gets good results in other more open environments like ECSC where our previous work is not applicable.

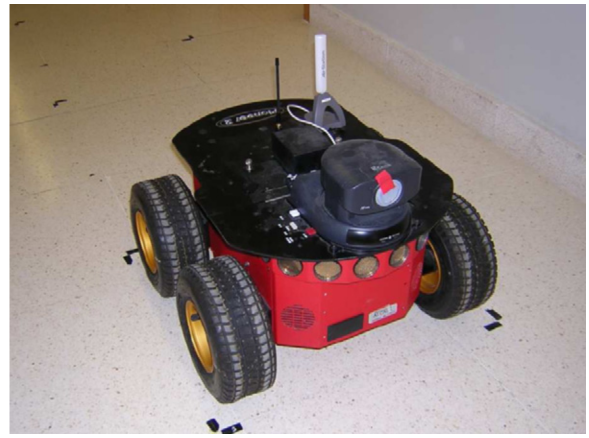
Acknowledgements

This work has been funded by grant CCG08-UAH/DPI-3919 (SISLOPEWI) from CAM/UAH and S2009/DPI-1559 (Robocity2030-II Project) from the Science Department of Community of Madrid. In addition, it has been partially funded by the Foundation for the Advancement of Soft Computing (Mieres, Asturias, Spain) and Spanish government (CICYT) under project TIN2008-06890-C02-01.

Appendix A. Details on robotic platforms

In each test-bed environment, experiments have been carried out with a different robot:

A.16. Real prototype used in the ECSC environment.



A.17. Real prototype used in the UAH environment.

- The robot used in the ECSC environment is called Sancho3. It is shown in Fig. A.16 and it was developed in the ECSC. This robot is based on a modular architecture whose first version was designed in the Technical University of Madrid (UPM). It has the following configuration: Linux Debian 5.0 Lenny operating system, Orinoco PCMCIA Silver wireless card, wireless tools v.28, two ultrasound sensor mounted over servos and one AXIS 213 pan-tilt-zoom camera.
- The robotic platform used in the UAH is called BART (Basic Agent for Robotics Tasks). This robot is based on the 2AT platform of Activmedia Robotics. It is illustrated in Fig. A.17 and it has the following configuration: Orinoco PCMCIA Gold wireless card, Linux Red Hat 9.0 operating system, wireless tools v.28, a 16 ultrasound sensor ring and a SONY pan-tilt-zoom camera.

Appendix B. Distance among topological positions

Tables B.2 and B.3 show the distances (in meters) among the reference MPi positions that are displayed in Figs. 4 and 5. Remind that such distances where used when computing the cumulative distribution functions (CDF) plotted in Figs. 14 and 15. Please, notice that our localization system is based on a topological approach so the system output yields the closer MPi reference point (along with a confidence degree) but not absolute X–Y coordinates like in other distance-based approaches that can be found in the literature. Anyway, we include here the real physical distance to make feasible a quick comparison against other methods that usually report position errors in the form of CDF.

Table B.2
Physical distances (measured in meters) among reference measurement points (ECSC).

	MP1	MP2	MP3	MP4	MP5	MP6	MP7	MP8	MP9	MP10	MP11	MP12	MP13	MP14	MP15	MP16
MP1	0	4	10	18	21	25.5	31	36	43.5	48.2	48.5	49.82	6.93	16.7	24.47	46.82
MP2	4	0	6	14	17	21.5	27	32	39.5	44.2	44.54	45.96	4.9	12.92	20.57	42.85
MP3	10	6	0	8	11	15.5	21	26	33.5	38.2	38.59	40.22	6.93	7.68	14.8	36.91
MP4	18	14	8	0	3	7.5	13	18	25.5	30.2	30.7	32.72	13.86	5.2	7.68	29.02
MP5	21	17	11	3	0	4.5	10	15	22.5	27.2	27.75	29.98	16.7	6.93	5.66	26.09
MP6	25.5	21.5	15.5	7.5	4.5	0	5.5	10.5	18	22.7	23.36	25.96	21.05	10.64	5.03	21.71
MP7	31	27	21	13	10	5.5	0	5	12.5	17.2	18.06	21.32	26.44	15.75	8.49	16.45
MP8	36	32	26	18	15	10.5	5	0	7.5	12.2	13.38	17.54	31.37	20.57	12.92	11.85
MP9	43.5	39.5	33.5	25.5	22.5	18	12.5	7.5	0	4.7	7.23	13.45	38.8	27.92	20.08	6.26
MP10	48.2	44.2	38.2	30.2	27.2	22.7	17.2	12.2	4.7	0	5.5	12.6	43.46	32.56	24.67	5.76
MP11	48.5	44.54	38.59	30.7	27.75	23.36	18.06	13.38	7.23	5.5	0	7.1	44.41	33.81	26.3	11.13
MP12	49.82	45.96	40.22	32.72	29.98	25.96	21.32	17.54	13.45	12.6	7.1	0	46.57	36.6	29.81	18.18
MP13	6.93	4.9	6.93	13.86	16.7	21.05	26.44	31.37	38.8	43.46	44.41	46.57	0	11	19	41.51
MP14	16.7	12.92	7.68	5.2	6.93	10.64	15.75	20.57	27.92	32.56	33.81	36.6	11	0	8	30.51
MP15	24.47	20.57	14.8	7.68	5.66	5.03	8.49	12.92	20.08	24.67	26.3	29.81	19	8	0	22.51
MP16	46.82	42.85	36.91	29.02	26.09	21.71	16.45	11.85	6.26	5.76	11.13	18.18	41.51	30.51	22.51	0

Table B.3
Physical distances (measured in meters) among reference measurement points (UAH).

	MP1	MP2	MP3	MP4	MP5	MP6	MP7	MP8	MP9
MP1	0	7.2	14.8	17.2	18.92	22.95	15.2	17.97	21.07
MP2	7.2	0	7.6	10	12.74	18.19	16.94	15.2	16.9
MP3	14.8	7.6	0	2.4	8.25	15.38	21.42	17.08	15.2
MP4	17.2	10	2.4	0	7.9	15.2	23.17	18.3	15.42
MP5	18.92	12.74	8.25	7.9	0	7.3	18.96	12.54	7.74
MP6	22.95	18.19	15.38	15.2	7.3	0	17.5	10.2	2.6
MP7	15.2	16.94	21.42	23.17	18.96	17.5	0	7.3	9.9
MP8	17.97	15.2	17.08	18.3	12.54	10.2	7.3	0	7.6
MP9	21.07	16.9	15.2	15.42	7.74	2.6	9.9	7.6	0

References

[1] J.M. Alonso, GUAJE: Generating Understandable and Accurate Fuzzy Models in a Java Environment, in: Free Software Under GPL License, 2010, available at http://www.softcomputing.es/guaje.

[2] J.M. Alonso, L. Magdalena, HILK++: an interpretability-guided fuzzy modeling methodology for learning readable and comprehensible fuzzy rule-based classifiers, Soft Computing (2010), doi:10.1007/s00500-010-0628-5.

[3] J.M. Alonso, L. Magdalena, S. Guillaume, HILK: A new methodology for designing highly interpretable linguistic knowledge bases using the fuzzy logic formalism, International Journal of Intelligent Systems 23 (7) (2008) 761–794.

[4] J.M. Alonso, M. Ocaña, M.A. Sotelo, L.M. Bergasa, L. Magdalena, WiFi localization system using fuzzy rule-based classification, in: Lecture Notes in Computer Science, Computer Aided System Theory – EUROCAST09 5717, 2009, pp. 383–390.

[5] J.J. Astrain, J. Villadangos, J.R. Garitagoitia, J.R.G. de Mendívil, V. Cholvi, Fuzzy location and tracking on wireless networks, in: Proceedings of the 4th ACM International Workshop on Mobility Management and Wireless Access (MobiWac06), ACM, New York, NY, USA, 2006, pp. 84–91.

[6] P. Bahl, V. Padmanabhan, RADAR: an in-building RF-based user location and tracking system, in: Proceedings of the 2000 IEEE Infocom, pp. 775–784.

[7] R. Barber, M. Mata, M.J.L. Boada, J.M. Armingol, M.A. Salichs, A perception system based on laser information for mobile robot topologic navigation, in: Proceedings of 28th Annual Conference of the IEEE Industrial Electronics Society, 2002, pp. 2779–2784.

[8] A. Carlotto, M. Parodi, C. Bonamico, F. Lavagetto, M. Valla, Proximity classification for mobile devices using wi-fi environment similarity, in: Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT08), ACM, New York, NY, USA, 2008, pp. 43–48.

[9] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, in: Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, volume 19, Advances in Fuzzy Systems—Applications and Theory, World Scientific Publishing Co. Pte. Ltd., 2001.

[10] A. Dharne, J. Lee, S. Jayasuriya, Using fuzzy logic for localization in mobile sensor networks: simulations and experiments, in: Proceedings of the American Control Conference (ACC06), 2006, pp. 2066–2071.

[11] E. Elnahrawy, X. Li, R. Martin, The limits of localization using signal strength: a comparative study, in: First Annual IEEE Communications Society Conference on Sensor Ad Hoc Communications and Networks (IEEE SECON04), 2004, pp. 406–414.

[12] P. Enge, P. Misra, Special issue on GPS: The Global Positioning System, in: Proceedings of the IEEE99, vol. 87(1), 1999, pp. 3–172.

[13] T. Gallagher, B. Li, A. Dempster, C. Rizos, A sector-based campus-wide indoor positioning system, in: IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN10), 2010, pp. 1–8.

[14] H. Hellendoorn, D. Driankov, in: Fuzzy Model Identification, Springer-Verlag, London, UK, 1997.

[15] R. Huang, G.V. Záruba, Monte carlo localization of wireless sensor networks with a single mobile beacon, Wireless Networks 15 (8) (2009) 978–990.

[16] J.C. Hühn, E. Hüllermeier, FURIA: an algorithm for unordered fuzzy rule induction, Data Mining and Knowledge Discovery 19 (3) (2009) 293–319.

[17] E. Hüllermeier, Fuzzy methods in machine learning and data mining: status and prospects, Fuzzy Sets and Systems 156 (2005) 387–406.

[18] H. Ichihashi, T. Shirai, K. Nagasaka, T. Miyoshi, Neuro-fuzzy ID3: a method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning, Fuzzy Sets and Systems 81 (1996) 157–167.

[19] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, S. Shafer, Multi-camera multi-person tracking for easy living, in: Proceedings of the 3rd IEEE International Workshop on Visual Surveillance, 2000, pp. 3–10.

[20] A. Ladd, K. Bekris, A. Rudys, L. Kavraki, D. Wallach, Robotics-based location sensing using wireless ethernet, Wireless Networks 11 (1–2) (2005) 189–204.

[21] E. López, R. Barea, L. Bergasa, M. Escudero, A human–robot cooperative learning system for easy installation of assistant robots in new working environments, Journal of Intelligent and Robotic Systems 40 (3) (2004) 233–265.

[22] L. Magdalena, What is Soft Computing? Revisiting possible answers, in: Proceedings of the 8th International FLINS Conference on Computational Intelligence in Decision and Control, 2008, pp. 3–10.

[23] E.H. Mamdani, Application of fuzzy logic to approximate reasoning using linguistic systems, IEEE Transactions on Computers 26 (12) (1977) 1182–1191.

[24] V. Matellán, J.M. Cañas, O. Serrano, WiFi localization methods for autonomous robots, Robotica 24 (4) (2006) 455–461.

[25] C. Mencar, A.M. Fanelli, Interpretability constraints for fuzzy information granulation, Information Sciences 178 (2008) 4585–4618.

[26] G.A. Miller, The magical number seven, plus or minus two: some limits on our capacity for processing information, The Psychological Review 63 (2) (1956) 81–97.

[27] G. Molina, E. Alba, Location discovery in wireless sensor networks using metaheuristics, Applied Soft Computing 11 (1) (2011) 1223–1240.

[28] D. Nauck, F. Klawonn, R. Kruse, in: Foundations of Neuro-fuzzy Systems, J. Wiley & Sons, Chichester, UK, 1997.

[29] C. Nerguizian, S. Belkhous, A. Azzouz, V. Nerguizian, M. Saad, Mobile robot geolocation with received signal strength (RSS) fingerprinting technique and neural networks, in: IEEE International Conference on Industrial Technology (IEEE ICIT04), vol. 3, 2004, pp. 1183–1185.

[30] M. Ocaña, L.M. Bergasa, M.A. Sotelo, R. Flores, D.F. Llorca, D. Schleicher, Automatic training method applied to a WiFi + ultrasound POMDP navigation system, Robotica 27 (2009) 1049–1061.

[31] S. Outemzabet, C. Nerguizian, Accuracy enhancement of an indoor ANN-based fingerprinting location system using Kalman filtering, in: Proceedings of the 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC08), 2008, pp. 1–5.

[32] S. Outemzabet, C. Nerguizian, Accuracy enhancement of an indoor ANN-based fingerprinting location system using particle filtering and a low-cost sensor, in: Proceedings of the 2008 IEEE International Conference on Vehicular Technology, 2008, pp. 2750–2754.

[33] B. Parodi, H. Lenz, A. Szabo, H. Wang, J. Horn, J. Bamberger, D. Obradovic, Initialization and online-learning of RSS maps for indoor / campus localization, in: Symposium on Position, Location, and Navigation, 2006, pp. 164–172.

[34] N. Priyantha, A. Chakraborthy, H. Balakrishnan, The cricket location support system, in: Proceedings of the 6th ACM MobiCom, 2002, pp. 155–164.

[35] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.

[36] J.R. Quinlan, in: C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[37] J.R. Quinlan, Improved use of continuous attributes in C4.5, Journal of Artificial Intelligence Research 4 (1996) 77–90.

[38] E.H. Ruspini, A new approach to clustering, Information and Control 15 (1) (1969) 22–32.

[39] T.L. Saaty, M.S. Ozdemir, Why the magic number seven plus or minus two, Mathematical and Computing Modelling 38 (3) (2003) 233–244.

[40] M.A. Sotelo, M. Ocaña, L.M. Bergasa, R. Flores, M. Marrón, M.A. García, Low level controller for a POMDP based on WiFi observations, Robotics and Autonomous Systems 55 (2) (2007) 132–145.

[41] L.X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, IEEE Transactions on Systems, Man and Cybernetics 22 (6) (1992) 1414–1427.

[42] R. Want, A. Hopper, V. Falco, J. Gibbons, The active badge location system, ACM Transactions on Information Systems 10 (1992) 91–102.

[43] I.H. Witten, E. Frank, in: Data Mining Practical Machine Learning Tools and Techniques, 2nd ed., Morgan Kaufmann, San Francisco, 2005.

[44] M. Youssef, A. Agrawala, A. Shankar, WLAN location determination via clustering and probability distributions, in: Proceedings of the 2003 IEEE PerCom, 2003, pp. 143–150.

[45] S. Yun, J. Lee, W. Chung, E. Kim, S. Kim, A soft computing approach to localization in wireless sensor networks, Expert Systems with Applications 36 (4) (2009) 7552–7561.

[46] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Transactions on SMC 3 (1973) 28–44.

[47] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning. Parts I, II, and III. Information Sciences 8, 8, 9 (1975) 199–249, 301–357, 43–80.