

Real-Time joint Landmark Recognition and Classifier Generation by an Evolving Fuzzy System

Xiaowei Zhou, *Student Member, IEEE*, and Plamen Angelov, *Senior Member, IEEE*

Abstract—A new approach to real-time joint classification and classifier design is proposed in this paper. It is based on the recently developed evolving fuzzy systems (EFS) method and is applied to mobile robotics. The approach stems from subtractive clustering method and its on-line evolving extension called eClustering. A new formula for data potential (spatial density) determination based on the participatory learning and data scatter concepts is introduced in the paper that is computationally simpler and more intuitive. An EFS-based self-organizing classifier (eClass) is designed by automatic labeling the landmarks that are detected in real-time. The proposed approach makes possible fully autonomous and unsupervised joint landmark detection and recognition without the use of absolute coordinates, any communication link or any pre-training. The proposed algorithm is recursive, non-iterative, one pass and thus computationally inexpensive and suitable for real-time applications. Extensive simulations as well as real-life tests has been carried out in an indoor environment (an office located at InfoLab21, Lancaster University) using Pioneer3 DX mobile robotic platform equipped with sonar and motion sensors and on-board PC. The results indicate superior rates of recognition, flexibility, and computational demands of the proposed approach comparing with the previously published similar methods. Further investigations will be directed towards development of a cooperative scheme, tests in a realistic outdoor environment, and the presence of moving obstacles.

I. INTRODUCTION

CLASSIFICATION has been around for quite some time [13]. Most of the approaches assume off-line (batch) treatment of the data. Once generated and trained the classifier can then be applied in on-line mode to new data. Validation on different data set however is normally done in off-line mode and on the assumption that statistical characteristics of the validation data set are similar to that of the training data set. Thus, a change in the data pattern or unexpected data can not, by definition, be taken into account by an off-line trained classifier [13].

In some application domains [19,24,26,27,29], however, it is vitally important to have the ability to adapt the classifier to new data patterns (non-stationary data streams, unpredictable and possibly hostile environment etc.). Adaptive classifiers have been reported that are based on Bayesian estimation [17] or the use of genetic algorithms [9]. The adaptation in the

former [17], however, concern the statistical properties of the data and not the structure of the classifier; the latter [9], called also ‘evolving’ (from evolutionary algorithms [8]), are computationally expensive (they operate over a population of candidate solutions) and are thus prohibitive for real-time applications. Additionally, both groups of approaches include supervision. It should be noted that classification, by definition [13], assumes supervision/labeling which is a serious obstacle in designing autonomous and flexible adaptive systems.

An alternative is presented by the self-organizing maps (SOM), introduced originally by T. Kohonen in 1987 for unsupervised learning (clustering) [22]. They are computationally less expensive and has been developed further into eSOM (evolving SOM) for the case when the cluster centers ‘evolve’ [20]. However, they, as well as a number of other evolving and self-organizing neural networks such as growing cell structures [14], adaptive resonance theory mapping [11], generalized growing and pruning radial-basis function networks [16], evolving fuzzy neural networks [20], dynamic evolving neuron-fuzzy inference systems [21], resource allocation networks [30] and their applications [18-19,23-29,32] do not take into account data density and thus are prone to generating too many clusters. Thus, pruning is needed which reduces the quantity of the clusters that are formed [16]. All these approaches are not prototype-based in the sense that the centre of the clusters is not necessarily and is often not located at a feasible point in the data space. It is usually located at the mean or its location is a result of the adaptation. Thus, its location in the data space is an abstraction and may not be a feasible data point. Additional disadvantage of these approaches is that new data point is compared to the cluster centers only because the real-time nature precludes memorizing the data history.

Recently, a novel approach to real-time data clustering was proposed [5] that stems from the well known subtractive clustering [12] and Mountain clustering approaches [34]. This method is fully unsupervised (according to the definition given in [15]) in the sense that number of clusters are also not pre-defined but determined based on the data density. This approach [5], called eClustering (from evolving Clustering) has been used for real-time data partitioning and was

Both authors are with the Intelligent Systems Research Laboratory, Digital Signal Processing Group, Department of Communication Systems, InfoLab21, South Drive, Lancaster University, Lancaster, LA1 4WA, UK,

phone +44 (1524) 510391; fax: +44 (1524) 510489, e-mail: angelov@lancaster.ac.uk

combined with an extended version of the recursive least squares estimation for real-time generation of Takagi-Sugeno type fuzzy rules from data [2]-[4].

In this paper, the concept of spatial density measured by the potential used in subtractive and eClustering is used as a basis for landmark recognition in mobile robotics. An alternative formula for potential (data spatial proximity) calculation is introduced in the paper that is using the concept of participatory learning [31,33] combined with the concept of data scatter [3]. In an inverse (to eClustering) fashion, the data points that have *very low* potential are treated as 'landmarks' in exploring a new environment by a mobile robot using only its on-board sensors and computational device and having no any pre-training, pre-installed knowledge, and no communication link or device (such as GPS, for example). Such situation may occur if the communication link is lost or deliberately ceased.

Landmarks are automatically labeled and further data are classified in real-time into classes associated to the landmarks detected so far or to a default class corresponding to the normal/routine behavior. The number of classes (respectively landmarks) is not pre-specified. Instead it evolves starting 'from scratch' with the very first landmark detected while exploring previously unseen environment.

The EFS-based classifier (eClass) proposed in this paper is formed by real-time detection of landmarks and labeling them. eClass is then used to classify in real-time the data produced by the sonar and motion sensors mounted on a mobile robotic platform Pioneer3-DX. In the experiment carried out in an indoor office environment (office B-69, InfoLab21, South Drive, Lancaster, UK) a robot performed simple 'wall following' behavior [24,26,27] exploring the unknown environment. The landmarks in an empty office are associated with the corners of that office. Each corner differs by its type (convex or concave) and relative position. The result of this experiment was compared with the result of a similar experiment reported in [28] where a SOM classifier pre-trained in off-line (batch) mode having a fixed structure was used. The results illustrate the superiority of the proposed approach in terms of computational efficiency, precision and flexibility.

Future investigations will be directed towards development of a co-operative scheme, tests in a realistic outdoor environment, and in the presence of (moving) obstacles.

I. LANDMARK RECOGNITION IN MOBILE ROBOTICS

As the robot travels in previously unseen environment it generates a stream of data from its sensors (sonar, motion, etc.). The ability to differentiate between common sensory readings and patterns never met before which is called 'novelty detection', is a very useful competence for a mobile robot operating in a real dynamic unexplored environment [26,29]. Using such ability the robot can select which aspects of the environment are unusual, differ from the contextual background and use them as 'landmarks'. Landmarks are

vitaly important for self-localization and navigation of mobile robots (autonomous agents) when operating in previously unseen (possibly hostile) environment [10,26,27]. By differ from 'dead reckoning', which is prone to drifting errors [24,28-29], landmark-based navigation does not suffer from this disadvantage [27]. Thus it can effectively be used for adaptive navigation and route planning.

At the same time the limited computational resources available to an autonomous mobile robot often present challenge for applications that demand real-time processing of large amounts of sensory data, therefore, a recursive algorithm is highly desirable to cope with the memory and time limitations. This is especially important for designing agile compact autonomous devices [10] where the computational and energy requirements are usually very restrictive.

The proposed EFS-based classifier, eClass is an efficient solution that is addressing the problem of real-time joint landmark recognition and classifier generation. It is fully autonomous in the sense that it does not require any pre-training, any pre-installed knowledge or any human intervention can start 'from scratch'. It is also computationally very efficient (it is recursive, one pass, non-iterative, and has very low memory requirements) which makes possible real-time applications [10].

II. EVOLVING FUZZY RULE-BASED CLASSIFIER (ECLASS)

A. eClass generation from data

The eClass method (as the name suggests) assumes an unspecified number of classes that gradually evolve. Their labels are automatically assigned using integer numbers ('1', '2', ...). The approach also assumes joint classification and classifier generation similarly to the joint adaptation and prediction used in conventional (linear) adaptive systems [7].

The eClass starts with a single class labeled '0' which corresponds to the normal/routine behavior and empty rule-base (all data are assumed to be described by the routine/normal behavior until a landmark is detected). The data is read incrementally sample by sample. Each data sample is described by a vector that can be represented as a data point in the data space, $\mathbf{x}_k = [x_k^1, x_k^2, \dots, x_k^n]^T$ where k is the current time instant (in a real-time application the time is open-ended and stops when a stop condition that is external to this algorithm is reached, thus $k=1, 2, \dots$). For the mobile robotic application considered in this paper the n -dimensional data vector includes the sensor readings that are available at given time instant k . For example, rotation, ϕ and the distance to the nearest obstacle, d etc.

eClass, similarly to eClustering, is based on the concept of data spatial density measured by so called potential [1-5]. This concept stems from the mountain function [35] modified later into potential in the subtractive clustering method [12], where a Gaussian exponential is used to describe the potential. A computationally simpler Cauchy function is used in [2,4,5]

and the so-called *scatter* is used in [3]. An alternative formula that combine the concept of participatory learning, proposed by Yager [31,33], and the concept of *scatter* can be given by:

$$P_k = P(x_k) = 1 - \frac{\sum_{i=1}^{k-1} \|x_i - x_k\|^2}{n(k-1)} \quad (1)$$

where P_k denotes the *potential* of the k^{th} data point x_k ; $\|\cdot\|$ denotes the Euclidean distance.

The potential formulated in this way has the meaning of compatibility of the new information compared with the information that already exists [31,33]. The potential is a monotonic, normal function ($0 \leq P \leq 1$), that is inversely proportional to the sum of the (Euclidean) distances between a point in the data space and **all other** data points. This is the main difference between eClustering and eClass from one side, and other clustering and classification approaches used for data space partitioning in self-organized neuro-fuzzy models [11,14,16,18-21,23-30,32,34] all of which ignore the data spatial density-related information.

Note that the expression for the potential (1) is suitable for off-line (batch) calculation since the summation over **all** previous data points is needed to determine the data density. To use potential in a real-time algorithm where memorizing the previous history is prohibitive, a *recursive* version of (1) is needed. One can derive such expressing the projections of the squared distances on axes $[0; x^j]$ from (1) in an explicit form:

$$P_k = 1 - \frac{\sum_{i=1}^{k-1} \sum_{j=1}^n \{ (x_k^j)^2 - 2x_k^j x_i^j + (x_i^j)^2 \}}{n(k-1)} \quad (1a)$$

By reorganizing (1a) one gets:

$$P_k = 1 - \frac{\sum_{j=1}^n (x_k^j)^2 - \frac{2 \sum_{j=1}^n x_k^j \sum_{i=1}^{k-1} x_i^j + \sum_{i=1}^{k-1} \sum_{j=1}^n (x_i^j)^2}{n(k-1)}}{n(k-1)} \quad (1b)$$

$$= 1 - \frac{(k-1) \sum_{j=1}^n (x_k^j)^2 - 2 \sum_{j=1}^n x_k^j \sum_{i=1}^{k-1} x_i^j + \sum_{i=1}^{k-1} \sum_{j=1}^n (x_i^j)^2}{n(k-1)}$$

If use the following notations:

$$a_k = \sum_{j=1}^n (x_k^j)^2; \quad b_k = \sum_{i=1}^{k-1} \sum_{j=1}^n (x_i^j)^2; \quad c_k = \sum_{j=1}^n x_k^j f_k^j; \quad f_k^j = \sum_{i=1}^{k-1} x_i^j \quad (1c)$$

the recursive formula for potential calculation is transformed into:

$$P_k = 1 - \frac{a_k(k-1) - 2c_k + b_k}{n(k-1)} \quad (1d)$$

Values a_k and c_k represent accumulated projections on each dimension of the data space, $[0; x^j]$ and can be calculated based on the availability of the current data point (set of sensor readings), x_k only. The values b_k and f_k^j require accumulation of past information. This can, however, be stored in two

auxiliary variables only (the scalar, b_{k-1} and the n -dimensional vector-column $f_k = (f_k^1, f_k^2, \dots, f_k^n)^T$). One can recursively update these n values by:

$$b_k = b_{k-1} + a_{k-1}; \quad b_1 = 0 \quad (1e)$$

$$f_k^j = f_{k-1}^j + x_{k-1}^j; \quad f_1^j = 0 \quad (1f)$$

In this way, using (1c)-(1f) the spatial density of each new data point, P_k in respect to **all** previous data points can be *recursively* calculated using n accumulated values in the two auxiliary variables (b_k and f_k^j) only. This makes possible real-time applications of the algorithm while keeping the information of spatial data density regarding the whole previous history which is the distinctive feature of the proposed algorithm.

B. Landmark classifier generation

Based on the sensory readings, a data point with low potential is a distinctive (specific) point in the data space that can be used as a landmark. Thus, low value of the potential will indicate the need to introduce a new landmark:

$$P_k < \underline{P} \quad (2)$$

where \underline{P} is a positive threshold (if the value of \underline{P} is too high too many landmarks are generated; if the value of \underline{P} is close to zero some landmark may be missed)

The very first data point that satisfies equation (2) is assumed to be the first landmark that is automatically assigned label '1':

$$x_{s1} \leftarrow x_k \text{ when } P < \underline{P} \quad (3)$$

Once the *Class 1* is formed a fuzzy rule (R_1) is also generated. The premise (antecedent) part of this rule is formed around the cluster centre, while the consequence is considered to be crisp (non-fuzzy):

$$R_1: \text{IF } (x_k^1 \text{ is close to } x_{s1}^1) \text{ AND} \dots \quad (4)$$

$$\text{AND } (x_k^2 \text{ is close to } x_{s1}^2) \text{ AND} \dots$$

$$\dots \text{AND } (x_k^n \text{ is close to } x_{s1}^n) \text{ THEN } (\text{Class is } 1)$$

The algorithm continues with reading the next data point and calculating its potential. By definition, the potential is an accumulated quantity that represents the density of the data and thus it gives information 'on average'. That is, single points may still be close to each other and their potential can be *low* if for the majority of the data the distances to these data points are large. To avoid ambiguity a landmark should be distinct 'on average' and 'in particular'. That means, there should be no indistinguishable, ambiguous landmarks. Therefore, the additional condition that the new candidate landmarks should not be in the vicinity of already existing ones is also introduced. The parameter r used as a threshold determines the zone of influence of a particular landmark (its uniqueness):

$$\|x_i - x_k\| > r/2; i=1,2,\dots,N; k=1,2,\dots \quad (5)$$

where x_{i_j} denotes i^{th} class centre (landmark);

N is the total number of classes (landmarks)

If condition (5) is violated while condition (2) is satisfied then the new data point replaces the nearest landmark:

$$x_{i_j} \leftarrow x_k \text{ when } \|x_i - x_k\| \leq r/2 \text{ and } P < P \quad (5a)$$

Remember that if condition (2) is satisfied a new landmark is detected regardless of condition (5). The new class is labeled with the next integer numeric symbol and a new fuzzy rule is added to the rule-base of eClass.

$$R_2: \text{IF } (x_k^1 \text{ is close to } x_{i_2}^1) \text{ AND...} \quad (4a)$$

AND $(x_k^2 \text{ is close to } x_{i_2}^2) \text{ AND ...}$

..AND $(x_k^n \text{ is close to } x_{i_2}^n) \text{ THEN (Class is 2)}$

In this way, an EFS-based classifier is generated in real-time with a recursive, non-iterative, incremental (one pass) algorithm that is based on the data spatial density (potential).

C. Landmark recognition (real-time classification)

At each time instant a classification (landmark recognition) takes place simultaneously with generating and evolution of the classifier. For the particular application (mobile robot exploring an unknown environment) two general states can be defined:

- normal routine operation (exploring the environment through 'wall following', 'random walk' or following certain navigation goal); this correspond to the default *Class 0*;
- novelty detection and landmark recognition (classify the data into one of the existing classes: *Class₁*, ..., *Class_N* or create a new *Class_{N+1}*). Note the new class may replace one of the existing classes.

During the normal, routine operation (further we assume without limiting the concept 'wall following' to be the normal/routine behavior) certain variation in the data takes place. The data pattern, however, is largely the same. Once a distinct data point is detected (based on condition (2)) a landmark is generated. Following that instant the further data will fall into one of the two cases described above (A or B). Therefore, the task is in real-time to classify the data into A or B. Note that in case of B different landmarks may exist and the task is to classify the data into the particular class or to generate a new class (detect a new landmark). Note, that the new landmark can also replace a previously existing landmark if condition (5a) is satisfied. The default case (normal/routine behavior) is given with an ELSE construct:

$$R_1: \text{IF } (x_k^1 \text{ is close to } x_{i_1}^1) \text{ AND...} \quad (4b)$$

..AND $(x_k^2 \text{ is close to } x_{i_1}^2)$

THEN (Class is 1) ELSE (Class is 0)

Obviously, this process starts once the first class and the respective fuzzy rule has been generated. Before that all data points are classified to be of *Class 0*.

The overall classification in eClass is performed based on the so called 'winner take all' principle [33], which corresponds to the MAX t-co-norm used to produce a de-fuzzification. It is interesting to note that the same de-fuzzification operator is also used in Mamdani type fuzzy models and therefore the fuzzy rule-based classifier eClass can be considered either as zero order Takagi-Sugeno type or as simplified Mamdani type (because the consequent part is crisp, consisting of non-fuzzy singletons):

$$y = y_{win}; \text{win} = \arg \max_{j=1}^N (\lambda_j) \quad (6)$$

where y_{win} represents the *winner* class (landmark)

$$\lambda_j = \begin{cases} 1 & \sum_{i=1}^N \tau_i = 0 \\ \frac{\tau_j}{\sum_{i=1}^N \tau_i} & \text{otherwise} \end{cases} \quad \text{is the normalized firing level}$$

of the j^{th} fuzzy rule, $j=1,2,\dots,N$.

The activation level, τ_i can be defined as a Cartesian product or conjunction (t-norm) of respective fuzzy sets for this rule [33]:

$$\tau_i = \mu_i^1(x^1) \times \mu_i^2(x^2) \times \dots \times \mu_i^n(x^n) = \prod_{j=1}^n \mu_i^j(x^j) \quad (7)$$

where μ_i^j is the membership value of the j^{th} input, x^j ($j=1,2,\dots,n$) to the respective fuzzy set for the i^{th} fuzzy rule ($i=1,2,\dots,N$)

The membership function value represents the degree to which the fuzzy set 'is close to' is satisfied in respect to the closeness to the landmark that is used to form the respective fuzzy set. In this study we use triangular membership functions without limiting the concept:

$$\mu_i^j(x_k) = \begin{cases} 1 - \frac{|x_k^j - x_{i_j}^j|}{r_i^j} & x_k^j - x_{i_j}^j < r_i^j \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where r_i^j is the radius of the zone of influence of the i^{th} landmark in its j^{th} dimension.

Note that one can define different radius for different input variables, x^1, x^2, \dots, x^n (sensory data streams) and different landmarks, $x_{i_j}; i=1,2,\dots,N$ emphasizing their relative importance/weight. Also note that for the routine, behavior all of the membership functions describing closeness to a landmark will have value zero (that is $\mu_i^j(x_k) = 0$ thus

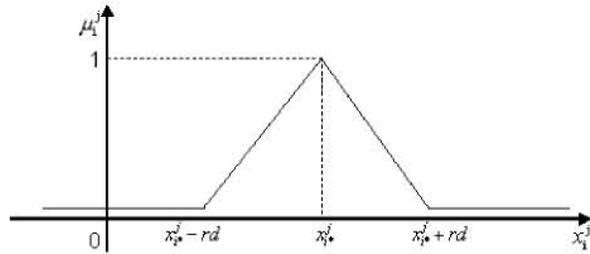


Fig. 1. Membership function of the j th fuzzy set 'is close to' of the i th fuzzy rule formed around i th landmark.

$$\sum_{i=1}^N \tau_i = 0 \text{ and } \lambda = 1 \text{ for } \forall j=1,2,\dots,n \text{ and } i=1,2,\dots,N.$$

By using eClass, a transparent, compact and accurate fuzzy rule-based classifier can be evolved in real-time in parallel with its use based on experimental data only. It is interesting to note that the rate of generating new classes and fuzzy rules representing a distinct landmark does not lead to an excessively large rule base. The reason for this is that the condition (2) is practically very strong for it concerns all previously seen data. Additionally, the possible proximity of a candidate center to the already existing landmarks (condition (5)) leads to just a replacement of the existing landmark, i.e. modification of its coordinates without enlarging the rule-base size.

D. Procedure of eClass

To summarize, the algorithm starts with an empty fuzzy rule-based classifier. The first data point (a n -dimensional vector of sensor readings at an instant of time) is read in real-time. Its *potential* (data spatial density) is assumed to be equal to one (the worst possible value if assume normalized data), $P(x_1) \leftarrow 1$ and the data point is assigned to *Class 0* (routine behavior). Starting with the next data point its potential is calculated recursively using (1c)-(1f). Two auxiliary variables (the one dimensional scalar b and the n -dimensional vector f) are accumulated according to (1e)-(1f). Once a data point has been used it is **discarded** and not stored in the memory (the algorithm is one-pass) which allows computational efficiency and real-time application. During the routine, normal behavior (we consider simple 'wall following' realized by a fuzzy logic controller) no landmarks were detected and no classes were formed, while first corner is met. Thus the winning class is the default (*Class 0*).

The first data point that satisfies equation (2) is used to form the first class:

$$x_{i1} \leftarrow x_i; N \leftarrow 1 \quad (9)$$

A label '1' is assigned to this class and a fuzzy rule of type (4b) is generated. For each new data point:

- 1) its potential is calculated recursively using (1c)-(1f);
- 2) conditions (2) and (5) are checked;
- 3) based on the comparison made at the previous step one of the following actions are taken:

- a) **IF** ((5a) holds) **THEN** (replace a cluster centre that is closer to the new data point $x_i \leftarrow x_k$);
- b) **ELSEIF** ((2) holds) **THEN** (form a new cluster around the new data point ($x_{N+1} \leftarrow x_k$); assign a new label $N \leftarrow N+1$; form a new fuzzy rule of type (4a)-(4b));
- c) **ELSEIF** ($\delta_y < r$; $\delta_i = \|x_i - x_k\|$; $\gamma = \underset{i=1}{\operatorname{argmin}} (\|x_i - x_k\|)$) **THEN** (assign the new data point to the *Class* γ);
- d) **ELSE** (classify the behavior as routine (assign the data point to *Class 0*) and do not change the classifier structure).

These basic steps are repeated for the next data point (set of sensor readings) ($k \leftarrow k+1$) until there is no more available data or until a requirement to stop the process is received.

The formal Procedure can be summarized with the following pseudo-code:

```

Begin eClass
Initialize:
Read  $x_1$ ;
Set  $k \leftarrow 1; N \leftarrow 1; P(x_1) \leftarrow 1; b_1 \leftarrow 0; f_1^j \leftarrow 0$ ;
DO for  $k \leftarrow k+1$ 
  Read  $x_{k+i}$ ;
  Calculate  $P_k$  recursively using (1c)-(1f);
  Accumulate values  $b$  and  $f$  using (1e)-(1f);
  Compare  $P_k$  with  $\underline{P}$  and condition (5);
  IF ((5a) holds)
  THEN (replace a cluster centre that is closer
        to the new data point  $x_i \leftarrow x_k$ );
  ELSEIF ((2) holds)
  THEN (form a new cluster around the new data
        point ( $x_{N+1} \leftarrow x_k$ ); assign a new label  $N \leftarrow N+1$ ;
        form a new fuzzy rule of type (4a)-(4b))
  ELSEIF ( $\delta_y < r$ ;  $\delta_i = \|x_i - x_k\|$ ;  $\gamma = \underset{i=1}{\operatorname{argmin}} (\|x_i - x_k\|)$ )
  THEN (assign  $x_k$  to the Class  $\gamma$ );
  ELSE (classify the behavior as routine (Class 0)
        and do not change the classifier structure);
  END
  /*assign the data sample to the nearest class
  using 'winner takes all' strategy (6) */
END DO
End (eClass)

```

III. EXPERIMENTAL RESULTS

This experimental test illustrates the application of the proposed algorithm eClass for landmark recognition by a mobile robot exploring a completely unknown environment without using GPS, global coordinates, any communication link and any pre-training. Operational data from back sonar sensor and rotation data are processed in real-time by the on-board computer mounted on the mobile robotic platform.

A. The mobile robotic platform used

The autonomous mobile robot Pioneer-3DX [36], produced by Activmedia, USA, (Fig. 2) has on-board computer

(Pentium III CPU, 256 MB RAM), camera, digital compass, sonar and bumper sensors, wireless connection for transmission of data to a desktop or laptop in real time. The robot can be controlled from the on-board computer in a client-server mode using embedded microprocessor ARCOS [6] and pre-programmed behaviors, such as ‘random wandering’, ‘obstacle avoidance’, etc. The fully autonomous behaviour of ‘detection novelties and landmark recognition’ was realized as an ARIA subroutine that runs on the ARCOS [6]. ARIA is a set of C-based foundation classes for control of the Active Media robots [36].



Fig. 2 Pioneer3 DX mobile robot performing autonomous joint landmark detection and recognition

B. Experimental settings

Extensive experiments were conducted in a real indoor environment (an empty office, B-69 located at InfoLab21, Lancaster, UK). As a first step, the same environment as the one used in [28] was re-created to make possible correct comparison of the results. It comprise a rectangular shaped empty office room with 8 corners (6 *concave* and 2 *convex*), as sketched in Fig.3.

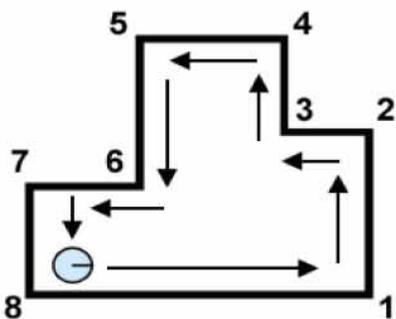


Fig. 3 Experimental enclosure 1 (Test office, B-69, InfoLab21, Lancaster) same as the one used in [28]

The basic objective of the experiment is to enable the robot to autonomously identify all landmarks in that office (that is to identify each corner) in real-time while exploring this completely unknown environment using a routine ‘wall following’ behaviour starting with *no prior* knowledge about the office shape, specifics, corner description, any rules or model nor any global coordinates. It should be noted that in [28] a similar experiment was conducted but pre-training of a SOM neural network was assumed and the network had a

fixed structure consisting of 50 neurons (that is a maximum number of 50 novelties was imposed). The pre-training reported in [28] was done off-line by a supervisor (teacher) walking the robot and showing the landmarks.

The inputs (Fig. 4) include rotation, ϕ (measured in degrees normalized by 360° counterclockwise from the direction opposite to the heading) and the distance to the nearest obstacle detected by the back sonar of the robot, d (measured in m and normalized by M), where M is the range of the sonar (in this experiment $M=10m$). In this way, heading straight corresponds to value $\phi=1/2$; turning 90° (*left*) in respect to the heading corresponds to $\phi=3/4$; turning -90° (*right*) in respect to the heading corresponds to $1/4$ and turn *back* corresponds to $\phi=1$. Note that the inputs are normalized into the range $[0;1]$.

Thus the input vector can be given as:

$$x = [\phi; d]. \quad (10)$$

These readings are taken in real-time while performing a routine ‘wall following’ behavior [35]. It should be mentioned that the features that are selected for the input vector are critical to the result. Experiments were conducted with between 2 and 5 features [35]. Generally, including more features leads to a more refined detection and eventually to a better result, but requires more computing power.

Finally, each landmark can easily be labeled to represent a distinctive class using for example integer numbers (‘1’, ‘2’, ‘3’,...). Thus, the generated classes are closely related to the corners. For example, if we have 8 corners in the office, ideally, there should be 8 landmarks and, respectively, 8 classes corresponding to each corner, (apart from the default Class 0).

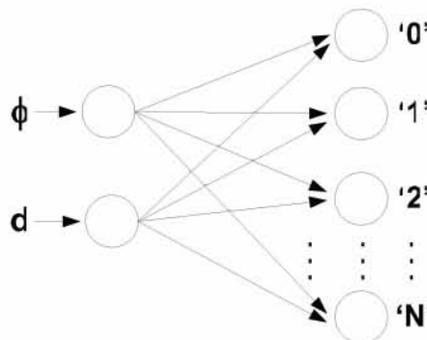


Fig. 4 eClass as a neuro-fuzzy system

C. Results and analysis

Once the input vector, x_k is read by the *eClass* the algorithm returns as an output the class label (‘0’, ‘1’, ‘2’, ...) and updates the rule-base (**adds** a new landmark and a new class respective to a new corner, **replaces** an existing landmark (class) or makes no change to the fuzzy rule-base structure). In this way, an EFS-based classifier has evolved in real-time to the following fuzzy rule-base:

- R₁: **IF** (ϕ is close to $3/4$) **AND** (d is close to 0.3000) (11) **THEN** (Corner is 1)
- R₂: **IF** (ϕ is close to $3/4$) **AND** (d is close to 0.1268)

THEN (Corner is 2)
 R_3 : IF (ϕ is close to $\frac{1}{4}$) AND (d is close to 0.0648)
 THEN (Corner is 3)
 R_4 : IF (ϕ is close to $\frac{3}{4}$) AND (d is close to 0.2357)
 THEN (Corner is 4)
 R_5 : IF (ϕ is close to $\frac{3}{4}$) AND (d is close to 0.0792)
 THEN (Corner is 5)
 R_6 : IF (ϕ is close to $\frac{1}{4}$) AND (d is close to 0.1744)
 THEN (Corner is 6)
 R_7 : IF (ϕ is close to $\frac{3}{4}$) AND (d is close to 0.0371)
 THEN (Corner is 8)

After the robot makes one full run in an anti-clockwise direction it was able to recognize successfully 7 out of 8 corners [35] with the remaining corner ('7') incorrectly classified as ('1') due to the close similarity between description of corners '1' and '7'. This result is better compared with the result reported earlier in [28] where in the similar experiment Only 5 out of 8 corners were recognized correctly with 5 features selected as inputs.

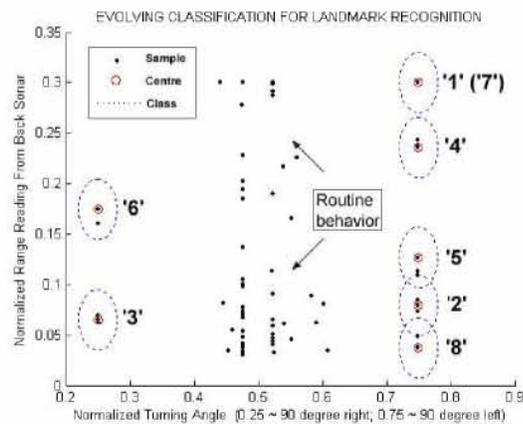


Fig. 5 Landmarks detected after three rounds in the office represent 7 out of the 8 existing corners (zones of influence are marked by a dashed line)

Using eClass algorithm described in the previous section fuzzy rule-based classifier is generated in real-time that classify the current sensor readings as:

- Routine behaviour (in this work 'wall following algorithm is used) that is the default Class θ ;
- One of the previously visited landmark (corner of the office) – Class I - Class N ;
- New landmark (corner) that has not been visited so far; then fuzzy rule-based classifier evolves its structure as described in the previous section.

Landmarks labelled automatically as Classes and the corresponding fuzzy rules practically reflect a corner of the empty office (there are no other landmarks available in this simplified experiment). The number of landmarks (respectively classes and fuzzy rules) is not pre-determined; it emerges as a result of the real-time live experiment and is based on the data alone.

Applying eClass, as described above, a fuzzy rule base was

generated in real-time 'on the fly' while performing the routine environment exploration starting 'from scratch' based on the rotation and odometer sensor readings only. Seven classes were formed during the first circle around the empty unexplored previously office that corresponded to seven (out of eight) really existing corners (Fig. 4). When a landmark (corner) was visited for second time the classifier was able to recognize this fact and information was displayed on the screen of the on-board computer of the mobile robot which can optionally be send wirelessly to another robot or to a monitoring desk-top workstation.

The performance of the proposed approach is then compared to the previously published results [28][35] and presented in Table I.

The proposed approach demonstrates superiority in several aspects: higher recognition rate; higher degree of autonomy; higher flexibility (eClass structure is not fixed and can accommodate more classes if the environment has changed)

TABLE I
 PERFORMANCE COMPARASION

	This paper	[35]	[35]	[28]
No. of features	2	5	4	5
correctly detected	7/(8)	7/(8)	6/(8)	5/(8)
over labeled	0	0	1	1
miss labeled	1	1	1	2
Feature used	ϕ and d	$\phi_0, D_0, \phi_{-1}, D_{-1}, \phi_{-2}$	$\phi_0, D_0, \phi_{-1}, D_{-1}$	$\phi_0, D_0, \phi_{-1}, D_{-1}, \phi_{-2}$
routine behaviors need to be filtered?	No	Yes	Yes	Yes

Where D denotes the distance between two landmarks.

"Over labelled" means more than 1 class was formed for 1 corner.

"Miss labeled" means no specific class was formed for the corner.

and higher computational efficiency.

Additional important advantage of the proposed approach is that the information extracted from the raw sensor readings and stored in the rule-base is fully linguistically transparent and interpretable.

IV. CONCLUSION

A new approach to real-time joint classification and classifier design is proposed in this paper. It is based on the recently developed evolving fuzzy systems (EFS) method and is applied to mobile robotics. The approach stems from subtractive clustering method and its on-line evolving extension called eClustering. A new formula for data potential (spatial density) determination based on the participatory learning and data scatter concepts is introduced in the paper that is computationally simpler and more intuitive. An EFS-based self-organized classifier (eClass) is designed by automatic labeling the landmarks that are detected in real-time. The true power of the proposed approach is in its role of an

engine that can generate interpretable rule-bases from data in real-time that has been realized in the evolvable classifier (eClass).

The proposed approach makes possible fully autonomous and unsupervised landmark detection and recognition without the use of absolute coordinates, any communication link or any pre-training. The proposed algorithm is computationally inexpensive (it is recursive, non-iterative, one pass) and runs in real-time on the on-board computer of the mobile robot. Extensive simulations as well as real-life tests has been carried out in an indoor environment (an office located at InfoLab21, Lancaster University) using Pioneer-3 DX mobile robotic platform equipped with sonar and motion sensors and on-board PC. The results indicate superior rates of recognition, flexibility, and computational demands of the proposed approach comparing with the previously published similar methods. Further investigations will be directed towards development of a cooperative scheme, tests in a realistic outdoor environment, and complicating the problem with moving obstacles.

V. ACKNOWLEDGEMENTS

The authors would like to thank H. V. Neto for his comments regarding an earlier version of the manuscript, G. Morris for his assistance in setting up the experiment, and C. Xydeas for arranging the access to B-69.

REFERENCES

- [1] Angelov P, *Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems*. Heidelberg, Germany: Springer-Verlag, 2002.
- [2] Angelov, P., D. Filev, "An approach to on-line identification of evolving Takagi-Sugeno models", *IEEE Trans. on Systems, Man and Cybernetics, part B*, vol.34, No1, pp. 484-498, 2004.
- [3] Angelov, P., D. Filev, "Simpl_eTS: A Simplified Method for Learning Evolving Takagi-Sugeno Fuzzy Models," *the 2005 IEEE International Conference on Fuzzy Systems FUZZ-IEEE*, Reno, Las Vegas, USA, 22-25 May 2005, pp.1068-1073
- [4] Angelov P., J. Victor, A. Dourado, D. Filev, "On-line evolution of Takagi-Sugeno Fuzzy Models," *2nd IFAC Workshop on Advanced Fuzzy/Neural Control*, Oulu, Finland, 16-17 Sept. 2004, pp.67-72.
- [5] Angelov P., "An Approach for Fuzzy Rule-base Adaptation using On-line Clustering," *Intern. Journal of Approx. Reasoning*, Vol. 35, No3, pp. 275-289, March 2004.
- [6] ARCOS (2005) ActivMedia's Robot Control & Operations Software, ActivMedia Robotics, LLC.
- [7] Astrom K., B. Wittenmark *Computer Controlled Systems: Theory and Design*, NJ USA: Prentice Hall, 1984.
- [8] Bentley P.J. (2000) "'Evolutionary, my Dear Watson': Investigating Committee-based Evolution of Fuzzy Rules for the Detection of Suspicious Insurance Claims," in *Proc. 2nd Genetic and Evolutionary Comp. Conf. (GECCO)*, Las Vegas, Nevada, USA, July 8-12, 2000
- [9] Bull, L., Kovacs, T., *Foundations of Learning Classifier Systems*. Heidelberg, Germany: Springer Verlagl 2005.
- [10] Carline D., P. Angelov, R. Clifford, *Agile Collaborative Autonomous Agents for Robust Underwater Classification Scenarios, Underwater Defense Technology*, Amsterdam, June 2005
- [11] Carpenter G. A., S. Grossberg, J. H. Reynolds, "ARTMAP: Su-pervised real-time learning and classification of non-stationary data by a self-organizing neural network," *Neural Networks*, vol4, pp. 565-588, 1991,
- [12] Chiu, S. L., "Fuzzy model identification based on cluster estimation," *Journal of Intelligent and Fuzzy Syst.* vol.2, pp. 267-278, 1994.
- [13] Duda R.O., P.E. Hart, D. G. Stork, *Pattern Classification, 2nd edition*, NY, USA: John Wiley and Sons Inc., 2001
- [14] Fritzke B. "Growing cell structures – a self-organizing network for unsupervised and supervised learning," *Neural Networks*, vol.7 (9) pp. 1441-1460,1994.
- [15] Gath I., A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Trans. on Pattern Analysis and Machine Intel.*, vol.7, 773-781, 1989.
- [16] Huang G.-B., P. Saratchandran, N. Sundarajan, "A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation," *IEEE Trans. on Neural Networks*, vol.16 (1), 57-67, 2005.
- [17] Jackson, Q., Landgrebe, D.A., "An adaptive classifier design for high-dimensional data analysis with a limited training data set," *Geosciences and Remote Sensing*, Vol 39 (12), 2664-2679, Dec. 2001.
- [18] Juang C.-F., X.-T. Lin, "A recurrent self-organizing neural fuzzy inference network," *IEEE Trans. on Neural Networks*, Vol.10, pp. 828-845, 1999.
- [19] Jun L., T. Duckett, "Robot Behavior Learning with a Dynamically Adaptive RBF Network: Experiments in Offline and Online Learning," *Proc. 2nd Intern. Conf. on Comput. Intelligence, Robotics and Autonomous System, CIRAS*, Singapore, Dec. 15-18, 2003.
- [20] Kasabov N "Evolving fuzzy neural networks for on-line supervised/unsupervised, knowledge-based learning," *IEEE Trans. SMC - part B, Cybernetics* Vol. 31, pp.902-918, 2001.
- [21] Kasabov N., Q. Song "DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction," *IEEE Trans. on Fuzzy Systems*, Vol.10 (2), pp. 144-154, 2002.
- [22] Kohonen, T., *Self-Organization and Associative memory*. New York: Springer Verlag, 1984.
- [23] Leng G., T.M. McGuinty, G. Prasad, "An approach for on-line extraction of fuzzy rules using a self-organizing fuzzy neural network," *Fuzzy Sets and Systems*, Vol.150 (2) pp. 211-243, 2005.
- [24] Li J., T. Duckett, "Growing RBF networks for learning reactive behaviours in mobile robotics, Intern. Journal of Vehicle Autonomous Systems," *Special issue on Comp. Intelligence and its Application to Mobile Robots and Autonomous Systems*, 2005
- [25] Lin, F.-J., C.-H. Lin, P.-H. Shen, "Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive," *IEEE Trans. on Fuzzy Systems*, Vol.9 (5), pp. 751-759, 2001.
- [26] Marsland, S., U. Nehmzow, J. Shapiro, "On-line Novelty Detection for Autonomous Mobile Robots," *J. Robotics and Autonomous Systems*, Vol 51, pp. 191-206, 2005.
- [27] Marsland, S., U. Nehmzow, T. Duckett "Learning to select distinctive landmarks for mobile robot navigation," *Robotics and Autonomous Systems*, Vol. 37, pp.241-260, 2001
- [28] Nehmzow, U., T. Smithers and J. Hallam, *Location Recognition in a Mobile Robot Using Self-Organising Feature Maps*, in *G.Schmidt (ed.), Information Processing in Autonomous Mobile Robots*, Springer Verlag, 1991.
- [29] Neto, H. V., U. Nehmzow "Visual Novelty Detection for Inspection Tasks using Mobile Robots," in *Proc. of 8th Brazilian Symposium on Neural Networks, IEEE Press*, São Luis, Brazil, 2004.
- [30] Plat J., (1991) A resource allocation network for function interpolation, *Neural Computation*, vol. 3 (2) 213-225.
- [31] Silva L., F. Gomide, R. Yager, "Participatory Learning in Fuzzy Clustering," *The 2005 IEEE International Conference on Fuzzy Systems FUZZ-IEEE*, Reno, Las Vegas, USA, 22-25 May 2005
- [32] Wu S., Er M. J., "Dynamic fuzzy neural networks – a novel approach to function approximation," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol 30 (2) pp. 358-364, 2000.
- [33] Yager R., "A model of participatory learning," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.20 (5) pp.1229-1234, 1990.
- [34] Yager R.R., D.P. Filev, "Learning of Fuzzy Rules by Mountain Clustering," *Proc. of SPIE Conf. on Application of Fuzzy Logic Technology*, Boston, MA, USA, pp.246-254, 1993.
- [35] Zhou X., P. Angelov, G. Morris, "Novelty Detection and Landmark Recognition by Real-time Evolving Clustering," *UK Workshop on Computational Intelligence*, London, 5-7 Sept., 155-161.
- [36] *Pioneer 3^{nm} Operations Manual*, Activmedia Robotics, LLC Amherst, NH, 2004, chap 5.