

Prune-Able Fuzzy ART Neural Architecture for Robot Map Learning and Navigation in Dynamic Environments

Rui Araújo, *Member, IEEE*

Abstract—Mobile robots must be able to build their own maps to navigate in unknown worlds. Expanding a previously proposed method based on the fuzzy ART neural architecture (FARTNA), this paper introduces a new online method for learning maps of unknown dynamic worlds. For this purpose the new Prune-able fuzzy adaptive resonance theory neural architecture (PAFARTNA) is introduced. It extends the FARTNA self-organizing neural network with novel mechanisms that provide important dynamic adaptation capabilities. Relevant PAFARTNA properties are formulated and demonstrated. A method is proposed for the perception of object removals, and then integrated with PAFARTNA. The proposed methods are integrated into a navigation architecture. With the new navigation architecture the mobile robot is able to navigate in changing worlds, and a degree of optimality is maintained, associated to a shortest path planning approach implemented in real-time over the underlying global world model. Experimental results obtained with a Nomad 200 robot are presented demonstrating the feasibility and effectiveness of the proposed methods.

Index Terms—Dynamic worlds, mobile robot navigation, neural architecture, self-organizing maps learning.

NOMENCLATURE

ART	Adaptive resonance theory.
CD	Category direct.
CDC	Category direct creation.
CDU	Category direct update.
CP	Category pruning.
FAR	Fuzzy ART rectangle.
FART	Fuzzy ART.
FARTNA	Fuzzy ART neural architecture.
MR	Multiresolution.
PAFARTNA	Prune-able fuzzy ART neural architecture.
POTF	Predictive online trajectory filtering.
PR	Perceptual range.

I. INTRODUCTION

ROBOTS must be able to build their own maps of the world, in order to navigate in complex, unknown, and changing environments. Maps are needed for path-planning,

Manuscript received December 16, 2004; revised February 7, 2006. This work was supported in part by Portuguese Science and Technology Foundation (FCT) under Project POSI/SRI/42043/2001.

The author is with the Institute for Systems and Robotics (ISR) and the Department of Electrical and Computer Engineering, University of Coimbra, P-3030-290 Coimbra, Portugal (e-mail: rui@isr.uc.pt).

Digital Object Identifier 10.1109/TNN.2006.877534

self-localization, and human-robot interaction. Grid-based certainty maps are a widely used representation tool in robotics research to store and maintain occupancy information [1]–[3]. Certainty grids have a simple geometric structure with a constant resolution that must be high enough to model all the different levels of local clutter and complexity in all areas of the world. This implies high costs and complexity in terms of data space and computation. Representations based on geometric features [4], on the other hand, have been difficult to build, but are significantly more compact, less complex, and fully applicable to high- and low-level motion planning (e.g., Section VI) and localization approaches [4]. Another alternative for overcoming the space and time complexities of grid-based methods is to use an MR state-space partition [5], [6]. In topological maps [7], [8], a graph-based world representation is maintained, with nodes representing sensory identifiable places, and links corresponding to navigation paths between places. The resolution of topological maps tends to be determined by the complexity of the environment, thus they usually have the advantage of being compact, and permit fast planning. Additionally, they are usually more tolerant to errors in the exact determination of the robot location.

A. Specific Context

This paper focuses on learning geometric feature-based global maps in unknown dynamic environments. Previous work [5] introduced and demonstrated the effectiveness of a new approach for sensor-based online learning of a world map. The approach is based on the FARTNA [9], [10], and incrementally constructs a map composed of rectangular geometric primitives (FARs), whose union represents occupied space, where sensor data points associated with objects have been perceived—a kind of unsupervised clustering [11] with features extraction. It has several desirable characteristics [5]: Self-organizing from sensor data; multifunctionality for map-building, motion planning, localization; small data requirements and low computational complexity; possible application to higher dimensional spaces without adversely impacting on its small data size and computation requirements; and updatability—incremental, online update by separately learning each sensor data point as it is received, with the same result as if the update were made in conjunction with a set of other data points not requiring the simultaneous consideration of a, possibly large, set of data points, making the model available as soon as possible to other system components such as path planning and localization. To combine advantages from different approaches, this method has been integrated into an architecture for navigation in dynamic

worlds [5], [12] that also incorporates ideas from MR grid methods and, to a lower extent, graph-based methods.

1) *Dynamic Worlds*: Beyond the aforementioned six desirable characteristics, an important general aspect of a map building method is its ability to cope with dynamic worlds. A changing robot world can be seen as exhibiting a union of one or more changes, each belonging to one out of two possible classes [5]. On Class 1, a new object is created on a previous free-space location. Changes of Class 2 correspond to the opposite, i.e., an object is removed creating a free area in the world. Moving objects correspond to combinations of changes of Classes 1 and 2. The FART-based map building method is able to cope with changes of Class 1. In fact, a new object will lead to new sensor perception points, which will generate new, or update existing, FART categories and corresponding rectangular geometric primitives in the map. However, the method is not able to appropriately cope with changes of Class 2. Thus, it should be complemented with the ability to remove or update geometric primitives, in response to the possible removal of objects in the world. For this purpose, this paper introduces the new PAFARTNA that extends FARTNA with structural and parametric adaptation mechanisms.

B. Related Work

1) *Feature-Based Maps*: In [4], a map composed of line and circle geometric features is built using a laser range sensor. A restricted use is made of the available sensor data to test for changes of Class 2. First, a test for feature(s) deletion, shrinking, or splitting is triggered *only* when a new feature is added to the model, or a new estimate of an existing feature is made. Second, the perceptual area used to test for changes of Class 2 is only a fraction of the total PR: A “wipe-triangle” region that spans *only* each newly created (or updated) line primitive when it is created (or updated), is used.

In [13], an extended Kalman filter, and feature track management operations are used to build a map of linear segments from laser range data. Features in the current scan direction that do not become matched by the range value (maximum or not) become eroded. If the measurement falls beyond the feature, this assumption is natural; otherwise, this implies an underlying assumption that “what is occluded (or invisible due to PR limitation) is empty” so that corresponding features should be removed—a change of Class 2 is being assumed, but this induces excessive forgetting of features, without sensory evidence support. Thus, [13] builds only a local map of the closest features inside the robot PR; however, the absence of a global model of the dynamic world invalidates the possibility to use global path planning to attain more accurate decisions.

In [14], a hyperellipsoidal clustering (HEC) Kohonen neural network [15] is proposed to build a map where occupied space is represented by ellipsoidal features. The map is built by simultaneously processing a large set of sensor data points gathered from a previous exploration phase performed over a static environment.

In [16], an enhanced adaptive fuzzy clustering algorithm along with noise clustering are proposed to build a map of

linear segments from range data in static environments. The algorithm works by iteratively learning a set of data points (in [16], points come from one complete scan).

In [17], a method is proposed to build a map that represents obstacles with a number of stochastic obstacle regions characterized by stochastic parameters such as mean and covariance. To operate, the method needs to simultaneously process a large set of sensor data points. Additionally, the method does not take advantage of all available sensor information because it has no capability for deleting subregions of regions that partially span outside the field of view of the laser scanner.

2) *Topological Maps*: In [18], vision and odometry sensor data are used to encode place information in an unsupervised growing neural network model of the environment that then forms the basis for action learning for goal-directed navigation. There is no provision to tackle nonstatic worlds with the model or actions.

In the topological model of [8], a known map of static aspects is assumed to be available *a priori*. Besides lacking autonomy, this forces a fixed overall structure of the environment. No provision exists to take into account changes of Class 2 in the static map. Predefined deterministic link costs model the known part. Statistical costs quantify the unknown part: *a priori* distributions for encountering unexpected obstacles are dynamically updated on the basis of sensor data and are subject to forgetting without sensory evidence.

Both [7] and [19] build topological maps. Places are categorized by sensory signatures of natural landmarks [7], or by odometry-based location information [19]. While in [7], there are two sequential operation phases: Initial map building and map application phase. In [19], there is the possibility to add links and places during navigation. To take into account changes of Class 1, in [7] and [19], a link confidence, established at link creation, is increased on successful traverses and reduced on fails: Normally repeated attempts are required before navigation discards a recently blocked link. Longer but more attempted paths may be undesirably preferred, probably never increasing confidence of better free paths later observed but reattempted only if other paths get even lower confidence. Also, in [7], the possibility is not considered that world changes during the map application phase may require insertion, deletion, or splitting of landmarks or links. Navigation attempts over changed areas are required to collect outcomes to update link confidence. This discards a wealth of other sensor information, continuously observed inside the PR, and that could be used to update the model.

3) *Approaches Based on Grid Maps*: To perform localization, sometimes it may be appropriate not to include dynamic objects into the map. In [2] and [3], constant resolution certainty grid maps of the static part of the world are built. The dynamic aspects of the world are filtered using entropy or distance filters on sensor data [2], or the EM algorithm [3], [20]. To perform real-time global dynamic path planning through free space in changing worlds, it is necessary to maintain a map that contains both the static and the dynamic objects that are being perceived from sensor data.

In [21], a global navigation system is presented that uses either case-based reasoning (CBR) or map-based path planning if CBR fails. There is not a unified global map to be continuously

built and updated in real-time and in response to dynamic environment changes. Two constant-resolution grids are used: One unchanging prespecified map with known static objects and a temporary map of perceived objects that is built from scratch every time a path is traversed. Other information is indirectly stored in a case-base of *global navigation path experiences*. The present work is somewhat related to [21] in that the approach of Section VI and [5] uses a database of *local navigation experiences* for global planning.

In [22], a neural dynamics-based approach is presented for real-time motion planning of a mobile robot. The current complete environment occupancy map is assumed as an externally available input. No capacity exists to learn the map from perceived sensor data.

In [23], constant resolution grid models are used. The map is subject to time-decaying changes without sensor evidence support. This may lead to the presentation of false-free areas to the planning subsystem. For planning, a hybrid method is proposed where between replannings, if the plan cannot be followed in the real world, the reactive control is used. Various time- and event-based methods for deciding when to replan are compared. The present method performs POTF replanning at the beginning of every cell-aim (Section VI).

4) *Neural Network Structure Management*: The use of mechanisms to influence or control the structure of a neural network may significantly contribute to attain the network's operational goals. The structure involves aspects such as the number of neurons and their topological interconnection patterns. In [24], a modified ART 2A growing network capable of generating a fixed maximal number of recognition categories is proposed. In [25], nodes are automatically created and pruned in a competitive learning network that can be used for clustering and quantization of a data set. The network can approximately estimate the cluster number and is adaptive to nonstationary input data. In [26], nodes are dynamically created and pruned with a sequential learning algorithm in an RBF neural network for function approximation. In [27], approaches are presented for a multi-experts network to learn nonlinear mappings. During training, there is addition or deletion of hidden nodes and adaptation of network parameters.

C. Contributions and Organization

This paper introduces the new PAFARTNA that is an extension of the FARTNA. The underlying motivation for identifying the need to develop the new PAFARTNA was the research and experimental work using FARTNA on a robot navigation architecture for dynamic environments. PAFARTNA incrementally constructs the map online by performing an update for each sensor data point. The PAFARTNA introduces the following three new mechanisms: Category removal, CDU, and CDC. The paper formulates and demonstrates important PAFARTNA properties. PAFARTNA map learning handles dynamic worlds. The new PAFARTNA mechanisms will directly support the capabilities of removing, updating, or splitting geometric features of the map, and will enable the map building method to also cope with changes of Class 2. For this purpose, it is necessary to detect the removal of objects in order to trigger the corresponding removals of complete or partial rectangular

categories existing in the map. To this aim, the paper also introduces a perception method that is integrated with PAFARTNA. All the static and dynamic aspects of the world are subject to learning into the model. Efficient use of sensor information is performed: All map features that have a nonempty intersection with the full current PR are subject to learning.

The proposed map learning methods contribute to a navigation architecture that has optimality characteristics due to its capacity of online global path planning using a minimax shortest path approach [5]. The underlying global world model is dynamically learned and updated according to all sensor data that is being captured inside the PR of the robot. This contrasts to local map building methods (e.g., [13]) and reactive navigation approaches (e.g., [28] and [29]) that treat the problem locally and adapt as a function of dynamic changes seen only inside the present PR. Decisions taken only from an instantaneous local PR may not be as accurate as those that are based on a global model of the scene. The navigation architecture also integrates an MR model which significantly improves the global system. All PAFARTNA and MR map learning activities start from empty world models. No *a priori* known map is required. Additionally, the PAFARTNA permits the introduction of innovative benefits on the behavior of the MR component of the system in dynamic worlds (Section VI).

The present work advocates (and uses) the assumption that the world is unchanged in the absence of contrary perceptual evidence. All map learning operations are based (have support) on observed spatial sensory information. If necessary, an additional exploration module should be included to explicitly control exploration beyond the current PR by properly choosing sequences of goal locations over target areas.

Also, compared to other methods in the field of dynamic map building/update, the present navigation architecture, based on the PAFARTNA features and the MR partition, inherits the benefits of feature-based and MR methods that were discussed at the beginning of Section I, in the context of worlds not necessarily dynamic. It also inherits the desirable characteristics of FARTNA map learning (Section I-A).

Section II overviews the FARTNA, both to describe its operation and to provide notations and background for its extension to PAFARTNA. Section III introduces PAFARTNA and in Section IV relevant PAFARTNA properties are formulated and demonstrated. Section V describes mechanisms for perception of world object removals in order to update the map. Section VI overviews a navigation architecture where the new map learning method has been integrated. In Section VII, experimental results obtained with a Nomad 200 robot are presented demonstrating the feasibility and effectiveness of the proposed methods. Section VIII discusses future research directions. Section IX makes concluding remarks.

II. FARTNA

This section briefly overviews the FARTNA [9], [10] and discusses its application for building maps [5] composed of geometric primitives. This network permits the extraction of a set of FART (hyper-) rectangles (FARs), whose union represents occupied space, where sensor data points associated with objects have been perceived.

A FART system includes a field F_0 of nodes representing a current input vector; a field F_1 that receives both bottom-up input from F_0 and top-down input from a field F_2 that represents the active code or category. The F_0 activity vector is denoted by $\mathbf{I} = (I_1, \dots, I_M)$ with $I_i \in [0, 1], i = 1, \dots, M$. The F_1 and F_2 activity vectors are, respectively, denoted by $\mathbf{y}_1 = (y_{11}, \dots, y_{1M})$ and $\mathbf{y}_2 = (y_{21}, \dots, y_{2N})$. The number of nodes in each field is arbitrary. Associated with each F_2 category node j ($j = 1, \dots, N$) is a vector $\mathbf{w}_j = (w_{j1}, \dots, w_{jM})$ of adaptive weights. Initially, all weights are set to $w_{j1}(0) = \dots = w_{jM}(0) = 1$ and all categories are said to be *uncommitted*. As explained in this section, when a category is selected for coding it becomes *committed*. The operation of the FART method is controlled by a choice parameter $\alpha > 0$, a learning rate parameter $\beta \in [0, 1]$, and a vigilance parameter $\rho \in [0, 1]$.

When an input data vector is presented to the system, a search is performed for a best but sufficiently matching category already in use. If this search fails, then an uncommitted category is recruited, thus becoming committed. In this context, for each presentation of input \mathbf{I} and F_2 node j , a *choice function* is defined by $T_j(\mathbf{I}) = |\mathbf{I} \wedge \mathbf{w}_j| / (\alpha + |\mathbf{w}_j|)$ where for any M -dimensional vectors \mathbf{p} and \mathbf{q} , “ \wedge ” denotes the min version of the fuzzy AND operator defined by $(\mathbf{p} \wedge \mathbf{q})_i = \min(p_i, q_i)$ and $|\cdot|$ denotes the Manhattan norm defined by $|\mathbf{p}| = \sum_{i=1}^M |p_i|$. For notational simplicity, $T_j(\mathbf{I})$ is often written as T_j when input vector \mathbf{I} is fixed.

The system is said to make a *category choice* when at most one F_2 node can become active at a given time. The category choice is indexed by J , where

$$T_J = \max\{T_j : j \in \{1, \dots, N\}\}. \quad (1)$$

If more than one T_j is maximal, the category j with the smallest index is chosen. In particular, nodes become committed in order $j = 1, 2, 3, \dots$. When the J th F_2 category is chosen, $y_{2J} = 1, y_{2j} = 0$ for $j \neq J$ and the F_1 activity vector is given by $\mathbf{y}_1 = \mathbf{I} \wedge \mathbf{w}_J$. By definition, *resonance* is said to occur if the *match function* between the presented input vector \mathbf{I} and the chosen category J , $C_J(\mathbf{I}) = |\mathbf{I} \wedge \mathbf{w}_J| / |\mathbf{I}|$ meets the following vigilance criterion:

$$|\mathbf{y}_1| = |\mathbf{I} \wedge \mathbf{w}_J| \geq \rho |\mathbf{I}|. \quad (2)$$

If so, then learning takes place as defined in (3). *Mismatch reset* occurs if $|\mathbf{y}_1| = |\mathbf{I} \wedge \mathbf{w}_J| < \rho |\mathbf{I}|$. In this situation, category J is *inhibited*, which means that the choice function T_J is set to 0 for the duration of the current input presentation to avoid the persistent selection of the same category during search. A new index J maximizing the choice function is chosen, and the search process continues until either the chosen J leads to resonance or no more noninhibited committed nodes exist. The later case means that the input vector does not match “sufficiently well” none of the committed categories. In this case, search ends with the recruitment and commitment of a previously uncommitted category J . Once search ends, *learning* takes place by updating weight vector J according to the following rule:

$$\mathbf{w}_J^{(\text{new})} = \beta (\mathbf{I} \wedge \mathbf{w}_J^{(\text{old})}) + (1 - \beta) \mathbf{w}_J^{(\text{old})}. \quad (3)$$

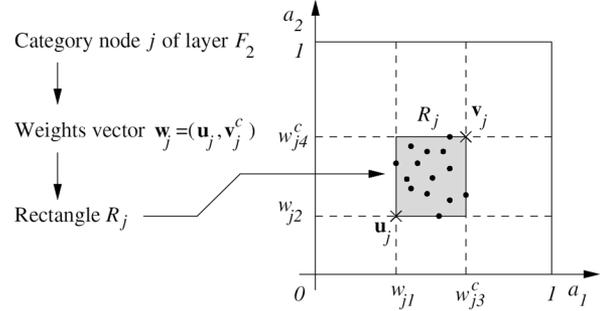


Fig. 1. FAR associated to category j .

By definition, *fast learning* corresponds to setting $\beta = 1$.

To avoid proliferation of F_2 categories, a *complement coding* input normalization rule is used [9], [10]. With complement coding, if the input is an M -dimensional vector $\mathbf{x}, \mathbf{x} = (x_1, \dots, x_M)$, then field F_0 receives the $2M$ -dimensional vector $\mathbf{I} = (\mathbf{x}, \mathbf{x}^c) = (x_1, \dots, x_M, x_1^c, \dots, x_M^c)$, where the complement of \mathbf{x} is denoted by $\mathbf{x}^c = (x_1^c, \dots, x_M^c)$ with $x_i^c = 1 - x_i$. In this work, when FARTNA (or PAFARTNA) is applied to map learning, \mathbf{x} represents a sensor data point perceived to belong to occupied space in the robot environment.

A. FARTNA Properties and Map Learning

Similarly to input vector \mathbf{x} , the weight vector \mathbf{w}_j can also be written in complement coding form: $\mathbf{w}_j = (\mathbf{u}_j, \mathbf{v}_j^c)$ where $\mathbf{u}_j = (u_{j1}, \dots, u_{jM})$ and $\mathbf{v}_j = (v_{j1}, \dots, v_{jM})$ are M -dimensional vectors. Let a FAR R_j be defined by using vectors \mathbf{u}_j and \mathbf{v}_j to define two of its vertices which are the borders of one of the diagonals of R_j and are the vertices which are, respectively, the closest and the farthest to the origin of the reference frame, as illustrated in Fig. 1. Additionally, rectangle R_j has its sides perpendicular to the axis of the reference frame. The size of R_j is defined as $|R_j| = |\mathbf{v}_j - \mathbf{u}_j| = \sum_{i=1}^M |v_{ji} - u_{ji}|$ which in the two-dimensional (2-D) case is equal to the sum of the height and width of the rectangle.

Theorem 1 (FART Stable Category Learning): If a FART system uses complement coding, fast learning ($\beta = 1$), and constant vigilance ρ , then the following conclusions **C1)–C5)** hold:

C1): The system forms hyperrectangular categories R_j that grow monotonically in all dimensions and converge to limits in response to an arbitrary sequence of input vectors. R_j grows to the smallest axis-aligned hyperrectangle that includes (thus represents) the set of all input data points, which have activated category j without reset (thus, either resonated or triggered the initial commitment of category j).

C2): Let \mathbf{w}_j be the weights vector of category j . Then

$$0 \leq w_{jk} \leq 1, \quad \text{for } k = 1, \dots, 2M \quad (4)$$

The size of the corresponding hyperrectangle equals $|R_j| = M - |\mathbf{w}_j|$ and is bounded as follows:

$$|R_j| \leq (1 - \rho)M. \quad (5)$$

C3): Let \mathbf{x} be an input data vector. When, after presenting \mathbf{x} the search process associated to learning ends with the initial commitment of a new category J , then the application of learning rule (3) leads to $R_J^{(new)} = (\mathbf{x}, \mathbf{x}^c)$, which represents a FAR degenerated to a point. Otherwise, if search ends with resonance with category J , then the application of learning rule (3) leads to $R_J^{(old)}$ being expanded to $R_J^{(new)} = R_J^{(old)} \oplus \mathbf{x}$, the smallest hyperrectangle that includes R_J , and the input vector \mathbf{x} , provided that $|R_J^{(old)} \oplus \mathbf{x}| \leq (1 - \rho)M$.

C4): In the conservative limit ($\alpha \rightarrow 0$) one-pass learning occurs such that no weight change or search occurs after each input data vector \mathbf{x} is presented just once, although some inputs may select different categories in future trials.

C5): Moreover, if $0 \leq \rho < 1$, the number of created (re-recruited or committed) categories is bounded, even if the number of exemplars in the training set is unbounded.

Proof: Equation (4) is trivial since $\beta = 1$, $\mathbf{w}_J^{(new)} = \mathbf{I} \wedge \mathbf{w}_J^{(old)}$ initially $w_{Jk}(0) = 1$ and $I_i \in [0, 1]$. For the rest of the proof see [9] and [10].

A FART system is said to operate in fast-commit, slow-recode case when $\beta = 1$ and J is an uncommitted node and $\beta < 1$ after the category is committed. Properties similar to Theorem 1 hold for the fast-commit, slow-recode case, except that repeated presentations of each input may be needed before stabilization occurs, even in the conservative limit. Only the FARTNA case defined by the assumptions of Theorem 1 will be considered in the sequel.

From conclusions **C1)** and **C3)** of Theorem 1, it is seen that the FARs can be used as geometric primitives that represent occupied space where sensor data points associated to the presence of objects have been perceived. Therefore, a FARTNA can be used to learn a map of the robot environment.

Fig. 2(a)–(d) geometrically illustrate the possible learning outcomes after presenting one input data point \mathbf{x} to FARTNA. In Fig. 2(a), the search process ends with the commitment of a new category. In this case, a new FAR degenerated to point \mathbf{x} is created. When the search ends with resonance with category J , then, from conclusion **C3)** of Theorem 1, this either leads to category J remaining unchanged because $\mathbf{x} \in R_J$ [e.g., Fig. 2(b)] or to what is defined as a *category growing* operation [e.g., Fig. 2(c) and (d)]. From conclusion **C2)**, it is seen that the maximum size of the rectangles R_j can be controlled with the vigilance parameter ρ .

To implement the FART-based map learning approach, the following additional mechanisms were introduced into the navigation architecture [5]: Conversion of sensor range points to world coordinates; sensor data filtering to reject noisy exemplars; sensor data scaling to the “FART state-space;” and controlling the size of the rectangles. Besides other things, this establishes for each committed F_2 category $j = 1, \dots, N$ a direct correspondence between rectangle R_j (weights \mathbf{w}_j) defined on the FART state-space and a rectangle R_j^W defined in world state-space of the map. Rectangles R_j^W form what is defined as the *FART (world) model*.

III. PAFARTNA

For map learning, it is important to take into consideration objects which have previously had influence in the construction of

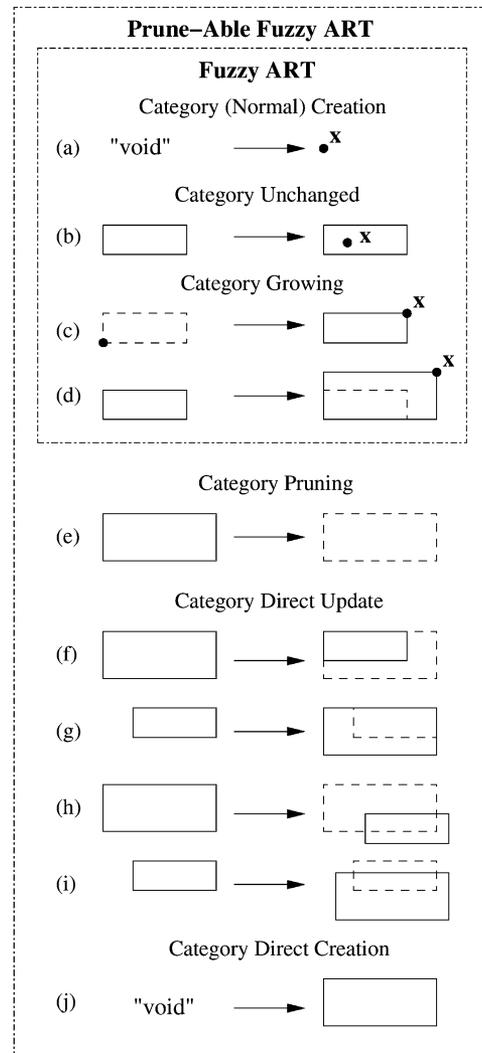


Fig. 2. FART and PAFARTNA: Geometric illustration of possible transformations of FARs through learning. Dashed rectangles correspond to FARs that either do not yet exist (c) or ceased to exist because of growing (d), or update (f)–(i). Points correspond to input data and/or FARs. (a)–(d): Presentations of an input data point to FART. (a) Presentation leads to the initial creation (commitment) of a category—a degenerate point-FAR is created. (b)–(d) Presentations lead to an existing FAR being either unchanged (b) or grown [(c)–(d)]. (e) Removal of a category and corresponding FAR. (f)–(i) Direct update of a category. (f) Category shrinking. (g) CD growing. (h) Category shrinking and displacement. (i) CD growing and displacement. (j) Direct creation of a category/FAR.

the world model but have then been removed (changes of Class 2). Thus, it is important that the map building method be able to remove from the model those parts of geometric primitives that represent objects that are no longer present in the world. These primitives complicate the map, retain a degree of representation of occupied space that has become excessive, and may undesirably prevent a motion planning module from recognizing “better” navigation paths which may have become available in the real world. They can also render the operation of a localization module more difficult. However, the FARTNA does not contemplate the possibility of discarding categories and the corresponding weights or performing a shrinking update of the geometric span of categories. In the FARTNA, from the moment

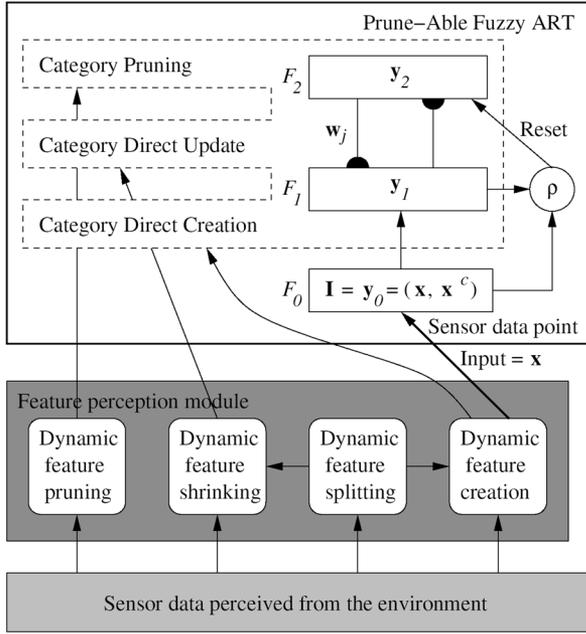


Fig. 3. Architecture for map learning in dynamic worlds, including the PAFARTNA and the feature perception module (Section VI).

an F_2 category node becomes committed, it remains committed forever and it can only grow its geometric span.

This raises the necessity to adjust the FARTNA so that the model is able not only to adapt itself to new sensor data points, but also to remove categories or parts of categories corresponding to removed objects. For this purpose, the PAFARTNA [30] will be introduced. It results from complementing the FARTNA (operating with complement coding, fast learning, and constant vigilance) with the following new mechanisms: Category pruning (CP), CDU (i.e., change the corners of the corresponding rectangle), and CDC. In complement to PAFARTNA, new mechanisms for detecting the removal of obstacles were developed (Section V) and integrated into the experimental navigation architecture of Section VI. Fig. 3 illustrates the map learning architecture—it includes a diagram of the PAFARTNA and its interaction with the feature perception module presented in Section V.

A. CP

Whenever the system detects the removal of an obstacle, a search is performed to identify the FARs that represent the world-obstacle in the model (see Section V). If an entire rectangle is being used in the representation of the object, then the corresponding category must be removed [e.g., Fig. 2(e)]. The problem for the PAFARTNA is the following: Given a rectangle R_j^W to be removed from the map, or equivalently its corresponding FAR, R_j updates the neural system to remove the corresponding recognition category.

Suppose rectangle R_j (i.e., category j) is to be removed. The corresponding weights vector \mathbf{w}_j can be written in complement coding form: $\mathbf{w}_j = (\mathbf{u}_j, \mathbf{v}_j^c)$ where \mathbf{u}_j and \mathbf{v}_j are vertices of the rectangle as discussed in Section II (e.g., Fig. 1). To remove category j would be equivalent to force neuron j of layer F_2

to return to an uncommitted state. However, since the FARTNA assumes that nodes are committed in the order $j = 1, 2, 3, \dots$, this would create a “hole”-neuron in the middle of F_2 violating the assumption. To avoid this, PAFARTNA starts by transferring the last node (N) to neuron j . To accomplish this, the system assigns weights \mathbf{w}_N to \mathbf{w}_j and assigns the values of the corresponding nodes of layers F_1 and F_2

$$\begin{cases} w_{ji} := w_{Ni}, & i = 1, \dots, M \\ y_{1j} := y_{1N}, \\ y_{2j} := y_{2N}. \end{cases} \quad (6)$$

Next, the components of weight vector N are forced to 1, which returns neuron N of layer F_2 to the uncommitted state and, correspondingly, the number of committed nodes of layer F_2 is decremented

$$\begin{cases} w_{N1} = \dots = w_{NM} = 1 \\ N := N - 1. \end{cases} \quad (7)$$

After performing the updates (6) and (7), the PAFARTNA can continue to be trained with new input exemplars and generate new categories if necessary.

Equation (6) has a reordering effect on the FART categories. In the FARTNA, the neuron to be updated in response to a particular input vector is determined by a search process (Section II). The order by which F_2 nodes are searched is determined by the application of (1). Due to the maximization involved in (1) the order and outcome of future searches is not affected by the reordering of F_2 nodes that result from the updates made by (6) and (7), except in cases where more than one T_j is maximal in one iteration of the search process. This raises no problem since it only opens the possibility that a specific input data vector may become represented by a different FART category (or corresponding rectangle primitive in the map). However, the outcome of the search process can be maintained in all presentations (with the obvious exception of the occurrence of the previous category j that is being removed) if the order of all neurons is maintained. This can be achieved by shifting down all nodes from $j+1$ to N . For this purpose, (6) can be changed as follows:

$$\begin{cases} w_{ki} := w_{(k+1)i}, & i = 1, \dots, M, \quad k = j, \dots, N - 1 \\ y_{1k} := y_{1(k+1)}, & k = j, \dots, N - 1 \\ y_{2k} := y_{2(k+1)}, & k = j, \dots, N - 1. \end{cases} \quad (8)$$

However, this alternative involves more computational costs without any significant advantage regarding the specific role of the neural architecture.

B. CDU

Here, the objective is to perform a resizing update of a given neural rectangular category R_j . This involves the update of the corresponding weights vector \mathbf{w}_j . First, among the new rectangle vertices, the closest (\mathbf{u}_j) and farthest (\mathbf{v}_j) to the origin are identified. The new weights vector will then be given by $\mathbf{w}_j = (\mathbf{u}_j, \mathbf{v}_j^c)$ where \mathbf{v}_j^c results from complement coding \mathbf{v}_j . A shrinking resizing can proceed normally and then the category

can resume growing monotonically in all dimensions. An expanding update can be performed only if the new size still meets the rectangle size bound condition (5), and the corresponding w_j meets condition (4).

CDU is a fairly general mechanism permitting the following types of actualizations to the PAFARTNA: Shrinking updates [e.g., Fig. 2(f)], growing updates [e.g., Fig. 2(g)], or shrinking or growing updates with displacement [e.g., Fig. 2(h), and (i)]. However, for the map learning method of this paper, shrinking updates are the only type of relevant CDUs.

C. CDC

In FARTNA, the only way a new category may be created (become committed) is when an input vector does not sufficiently match none of the existing categories. If this is not the case, the best-matching category is updated, which corresponds to maintaining or, if necessary, growing the geometric span of the corresponding FAR R_j to represent the new input vector. However, as will become clear in Section V, there are situations where it is required to force the integration of a specific FAR into the model [e.g., Fig. 2(j)]. Trying to do that through the presentation to FARTNA of a set of artificially designed input vectors is not a solution because the internal result on FARTNA could be just the update of one or more existing categories, i.e., growing the geometric span of the corresponding FAR(s). That could undesirably result on the inclusion of free space into the FART world model of occupied space.

The procedure for CDC can only proceed if the candidate to become a new FAR meets the rectangle size bound condition (5) and the corresponding w_j meets condition (4). If so, then a new category is created in the first place: The number of categories is incremented, and the new weights vector is initialized to zero (FAR encompassing all the FART state-space)

$$\begin{cases} N := N + 1 \\ w_{N1} = \dots = w_{NM} = 0. \end{cases}$$

The second step is to create the corresponding weights vector that is defined by the vertices of the new rectangle. This is performed through a shrinking CDU as defined previously.

IV. PAFARTNA PROPERTIES

This section describes and demonstrates a set of important PAFARTNA theoretical properties. There are the following two classes of operations that can be performed in an PAFARTNA system: 1) The inherited *FARTNA (learning) operations* which are the ones defined in Section II, and 2) *CD operations* which are the three operations defined in Sections III-A–C. An *FARTNA session* is defined as a sequence of (only) FARTNA operations, and a *CD session* is defined as a sequence of (only) CD operations. Thus, the operation of an PAFARTNA system can be decomposed into a sequence of FARTNA sessions and CD sessions. Theorem 2 will describe and establish the main PAFARTNA properties. Without loss of generality it will be assumed that each input vector \mathbf{x}_i is presented just once to a

FARTNA or PAFARTNA. Multiple presentations just mean that $\mathbf{x}_i = \mathbf{x}_{i+1}$ may happen.

Definition 1: At the moment an input vector \mathbf{x}_i is presented to a FARTNA or PAFARTNA, π on a FARTNA operation, it is said to become *associated* to category j if it activates category j without reset (thus, either resonates with, or triggers the commitment of, category j). An input vector \mathbf{x}_i is always associated to one and only one category which is denoted by $\Lambda_\pi(\mathbf{x}_i)$. Subsequently, whenever a CP operation (Section III-A) removes category q , a reordering of categories occurs. This reordering effect is also taken into account into the definition of $\Lambda_\pi(\mathbf{x}_i)$ (for all \mathbf{x}_i) as follows. If (6) was used on the CP, then if $\Lambda_\pi(\mathbf{x}_i)^{(\text{old})} = N$, $\Lambda_\pi(\mathbf{x}_i)^{(\text{new})} = q$. Otherwise, if (8) was used on the CP, then if $\Lambda_\pi(\mathbf{x}_i)^{(\text{old})} = j + 1$, $\Lambda_\pi(\mathbf{x}_i)^{(\text{new})} = j$ ($j = q, \dots, N - 1$).

The next lemma shows that a FARTNA ψ^* composed of a subset of categories of FARTNA ψ can be constructed by presenting to ψ^* the subsequence of input vectors of ψ that were associated to categories belonging to the subset.

Lemma 1: Let ψ be a FARTNA system with N_0 categories. Define $S_\psi = \{1, \dots, N_0\}$ to be the ordered set of all the categories of ψ . Let ψ^* be a FARTNA system composed by subsequence $S_{\psi^*} = \{f_1, \dots, f_{N_0^*}\} \subseteq S_\psi$ of N_0^* ($\leq N_0$) categories of ψ that on ψ^* are renumbered (preserving the order) to $S_{\psi^*} = \{1, \dots, N_0^*\}$. Let $X(\psi) = \{\mathbf{x}_1, \dots, \mathbf{x}_Q\}$ be the sequence of all the Q input data vectors that were presented to construct (i.e., were associated to categories of) ψ and $X(\psi^*) = \{\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_{Q^*}}\} \subseteq X(\psi)$, the subsequence of all the Q^* ($\leq Q$) input data vectors that were associated to categories $j \in S_{\psi^*} \subseteq S_\psi$. Then, ψ^* is a FARTNA system that can be equivalently constructed by separately presenting subsequence $X(\psi^*)$ to ψ^* immediately after initialization of ψ^* . Furthermore, the weights vector of category $f_k \in S_{\psi^*}$ of ψ is equal to the weights vector of category k of ψ^* for $k = 1, \dots, N_0^*$.

Proof: The proof will be done by induction on variable Q and by construction of ψ and ψ^* in parallel. Suppose ψ and ψ^* are both initialized, and then the ordered presentation to ψ of all the elements of sequence $X(\psi)$ is performed. If $Q = 1$, then clearly $S_\psi = \{1\}$, $S_{\psi^*} = S_{\psi^*}$, and S_{ψ^*} is either \emptyset or $\{1\}$, where symbol “ \emptyset ” represents the empty set. If $S_{\psi^*} = \emptyset$, then $X(\psi^*) = \emptyset$ and thus ψ^* is an empty FARTNA. If $S_{\psi^*} = \{1\}$, then $S_\psi = S_{\psi^*} = \{1\}$, $X(\psi) = X(\psi^*) = \{\mathbf{x}_1\}$ and the presentation of \mathbf{x}_1 to ψ and ψ^* results in ψ and ψ^* being completely equal. Thus, in both cases, Lemma 1 is valid for $Q = 1$.

Now assume that the lemma is valid for any integer $Q = i - 1 > 0$. It remains to be proved that it is also valid for $Q = i$. Let $S_\psi^0 = S_\psi \cap \overline{S_{\psi^*}^0}$ where symbols “ \cap ” and “ $\overline{}$ ” represent the intersection and complement set operations, respectively. Since the lemma is valid for $Q = i - 1$, then by presenting sequence $X_1(\psi) = \{\mathbf{x}_1, \dots, \mathbf{x}_{i-1}\}$ to ψ and sequence $X_1(\psi^*) = X(\psi^*) \cap X_1(\psi)$ to ψ^* , the FARTNAs ψ and ψ^* are partially constructed up to intermediate stages, where they jointly meet the conclusions of the lemma. When the final input vector $\mathbf{x}_i \in X(\psi)$ is presented to complete the construction of ψ , if it also happens that $\mathbf{x}_i \in X(\psi^*)$, then the associated category is a $J \in S_{\psi^*}^0$, which is equivalent to $J \notin S_\psi^0$, and in

this case let \mathbf{x}_i be also presented to ψ^* . Let $N(i)$ and $N_0^*(i)$ be, respectively, the number of committed categories in ψ and ψ^* after the presentation of \mathbf{x}_{i-1} . When $J \notin S_\psi^0$ occurs, this happens because either no $j \in S_\psi^0 \subseteq S_\psi$ maximized (1) or if some maximized it (or they) failed to meet vigilance criterion (2). By this reason maximization (1) can be substituted by $T_J = \max\{T_j : j \in S_\psi^* \cap \{1, \dots, N(i)\}\} = \max\{T_j : j \in S_\psi^*\}$, which yields a maximizing $J = f_k \in S_\psi^*$ for some k . When \mathbf{x}_i is presented to ψ^* , then (1) takes the form $T_J = \max\{T_j : j \in \{1, \dots, N_0^*(i)\}\} = \max\{T_j : j \in S_{\psi^*}\}$ which, due to the order-preserving renumbering assumption of this lemma, yields a maximizing category $J^* = k$ of ψ^* , because, by definition, category k ($\in S_{\psi^*}$) of ψ^* is equal to category f_k ($\in S_\psi^*$) of ψ . The subsequent steps of vigilance criterion verification (2) and learning (3) depend only on \mathbf{x}_i and weights vectors \mathbf{w}_J and \mathbf{w}_{J^*} of the corresponding chosen categories of ψ and ψ^* . This implies that: 1) The search operations on ψ and ψ^* end yielding categories $J = \Lambda_\psi(\mathbf{x}_i) = f_k \in S_\psi^*$ of ψ and $J^* = \Lambda_{\psi^*}(\mathbf{x}_i) = k$, of ψ^* , for some k , respectively; and 2) learning has the same effect on categories $J = f_{J^*}$ of ψ and J^* of ψ^* (and all other categories remain unchanged). This implies that the weights vectors of these two categories, not only were equal by hypothesis before the presentation of \mathbf{x}_i (because the lemma is valid for $Q = i - 1$), but also continue to be equal after the presentation of \mathbf{x}_i . Thus, by presenting all input vectors $\mathbf{x}_n \in X(\psi)$, the construction of ψ and ψ^* meeting the conclusions of Lemma 1 has just been completed.

Theorem 2 (Prune-Able FART Stable Learning): If an PAFARTNA system π uses complement coding, fast learning ($\beta = 1$), and constant vigilance ρ , then conclusions **C1**–**C5** hold on π . These conclusions are, respectively, equal to the conclusions with the same label on Theorem 1 except that the following adaptations are performed. **C1**, **C3**, and **C4** hold during all FARTNA sessions of π and **C2** always holds on π . **C5** holds if the number of CDC operations is bounded, and **C5** holds during FARTNA sessions of π .

Proof: During operation, PAFARTNA π is always composed of a sequence of N categories $i = 1, \dots, N$, where N is varying. Define the ordered set $S_\pi = \{1, \dots, N\}$. Note that after a category j undergoes a CDU or a CP operation, all the input vectors \mathbf{x}_i that were associated to category j (i.e., $\Lambda_\pi(\mathbf{x}_i) = j$) before that CDU or CP operation, cease to have any effect on \mathbf{w}_j or on π [however, note that, in the future, all input vectors presented to π after that CDU or CP operation may, independently of their value, have effect on π after (due to) the corresponding presentation]. By this reason, define $X(\pi) = \{\mathbf{x}_1, \dots, \mathbf{x}_Q\}$ to be the sequence of all the Q input vectors \mathbf{x}_i that were presented to π and such that category $\Lambda_\pi(\mathbf{x}_i)$ was not subject to any CDU or CP operations after the presentation of \mathbf{x}_i . Let $S_f = \{f_1, \dots, f_F\} \subseteq S_\pi$ be the subsequence of all the F categories of π that were created (recruited) during FARTNA sessions and were not subsequently updated by any CDU operation. Let $X(0) = \{\mathbf{x}_{i_{01}}, \dots, \mathbf{x}_{i_{0L(0)}}\} \subseteq X(\pi)$ be the subsequence of all the $L(0)$ input vectors $\mathbf{x}_{i_{0k}} \in X(\pi)$ ($k = 1, \dots, L(0) \leq Q$) associated to categories in S_f , i.e., $\Lambda_\pi(\mathbf{x}_{i_{0k}}) \in S_f \subseteq S_\pi$. Let $c_0 = 0$, and $\psi(c_0)$ be the FARTNA system that is separately constructed by separately

presenting sequence $X(0)$ to $\psi(c_0)$ immediately after initialization of $\psi(c_0)$. From Lemma 1 $\psi(c_0)$ has F categories. Let $X(j) = \{\mathbf{x}_{i_{j1}}, \dots, \mathbf{x}_{i_{jL(j)}}\} \subseteq X(\pi)$ ($j = 1, \dots, N$) be the subsequence of all the $L(j)$ input vectors $\mathbf{x}_{i_{jk}} \in X(\pi)$ ($k = 1, \dots, L(j) \leq Q$) associated to category j , i.e., $\Lambda_\pi(\mathbf{x}_{i_{jk}}) = j \in S_\pi$.

When a CDC or a CDU operation is performed, assume that the weights vector of the target category, say j , is set to $\mathbf{w}_j^{(\text{cdo})} = [\mathbf{u}_j^{(\text{cdo})}, (\mathbf{v}_j^{(\text{cdo})})^c]$ where $\mathbf{u}_j^{(\text{cdo})}$ and $\mathbf{v}_j^{(\text{cdo})}$ are the vertices of the corresponding new hyperrectangle that are closest and farthest to the origin, respectively. On a CDC, j is a newly created category and on a CDU, j is an existing category. Let $S_c = \{c_1, \dots, c_C\} \subseteq S_\pi$ be the subsequence of all the C categories of π that were created by CDC operations or updated by CDU operations. Clearly $N = F + C$, $S_f \cup S_c = S_\pi$ and $S_f \cap S_c = \emptyset$, where symbol “ \cup ” represents the union set operation. For $c_p \in S_c \subseteq S_\pi$ ($p = 1, \dots, C$), let $\psi(c_p)$ be a FARTNA system that is separately constructed by separately presenting the following two sequences to $\psi(c_p)$ immediately after initialization of $\psi(c_p)$: First, $\{\mathbf{u}_{c_p}^{(\text{cdo})}, \mathbf{v}_{c_p}^{(\text{cdo})}\}$, and second, $X(c_p)$. (Note: Since by definition $X(c_p) \subseteq X(\pi)$, we have $X(c_p) = \emptyset$ immediately after the CDU or CDC operation). Note that, from the definitions it follows that input vectors can be associated to categories of $\psi(c_0)$ either by resonance, or initial commitment, but can only be associated to the category of $\psi(c_p)$ for any $p = 1, \dots, C$ by resonance.

Next it will be shown that π can be equivalently represented by a set of FARTNA systems φ , defined as follows: $\varphi = \cup_{p=0}^C \{\psi(c_p)\}$. By Lemma 1 and the definitions of $\psi(c_p)$ ($p = 0, 1, \dots, C$) it is clear that: 1) $\psi(c_0)$ is a FARTNA system composed of one copy of each category $j \in S_f \subseteq S_\pi$, and 2) $\psi(c_p)$ ($p = 1, \dots, C$) is a FARTNA system composed only by one copy of category c_p of π . Further, the categories of $\psi(c_0)$ are arranged on the same order as the relative order that their corresponding copies have on π . The total number of categories of φ is $F + C = N$. Let the categories of φ be numbered such that category j of φ is equal to category j of $\psi(c_0)$ for $j = 1, \dots, F$, and equal to the category of $\psi(c_{j-F})$ for $j = F + 1, \dots, N$. A category is said to correspond to another if the weights vectors of both categories are equal. Clearly for every category j of π there is one and only one corresponding category $h(j)$ of φ , where function h is defined as follows: $h^{-1}(j) = f_j$ if $j = 1, \dots, F$, and $h^{-1}(j) = c_{j-F}$ if $j = F + 1, \dots, N$. It has just been shown that an PAFARTNA system π constructed using only FARTNA, CDC, and CDU operations can be equivalently represented by φ . However, this equivalence remains valid in general PAFARTNA systems, i.e., those where CP operations are performed. In fact, assume that category q of π ($1 \leq q \leq N$) is pruned. Then, from the definition of φ and by Lemma 1, φ can be equivalently updated by removing category $h(q)$ of $\psi(c_0) \subseteq \varphi$ if $q \in S_f$ (i.e., $0 \leq h(q) \leq F$), or by removing FARTNA $\psi(c_q)$ of φ if $q \in S_c$ (i.e., $F < h(q) \leq N$).

Since, as aforementioned, π is equivalent to the set of FARTNAs φ , then conclusions **C1**–**C3** of Theorem 1 always hold during FARTNA sessions of π because **C1**–**C3** describe characteristics of categories of π that have been equivalently constructed in the FARTNAs $\psi(c_p)$ ($p = 0, \dots, C$). Addi-

tionally, **C2**) also holds during CD sessions because (4) and (5) are enforced on CDU and CDC operations (Sections III-B and C), and the size of hyperrectangle R_j only depends on the corresponding weights vector \mathbf{w}_j .

To prove that **C4**) holds, suppose a repeated presentation of the same input $\mathbf{x}_{i+1} = \mathbf{x}_i = \mathbf{x}$ to π . Let superscripts “(old)” and “(new)” indicate the values on the first and second presentations, respectively. If all components of some vectors \mathbf{a} and \mathbf{b} are nonnegative, then $|\mathbf{a} \wedge \mathbf{b}| \leq |\mathbf{a}|$. Thus, if $\Lambda_\pi(\mathbf{x}_i) = J$ then from (1) it follows that, for $j = 1, \dots, N$:

$$\begin{aligned} T_J^{(\text{new})}(\mathbf{I}) &= \frac{|\mathbf{I} \wedge \mathbf{w}_J^{(\text{new})}|}{\alpha + |\mathbf{w}_J^{(\text{new})}|} = \frac{|\mathbf{I} \wedge (\mathbf{I} \wedge \mathbf{w}_J^{(\text{old})})|}{\alpha + |\mathbf{I} \wedge \mathbf{w}_J^{(\text{old})}|} \\ &= \frac{|\mathbf{I} \wedge \mathbf{w}_J^{(\text{old})}|}{\alpha + |\mathbf{I} \wedge \mathbf{w}_J^{(\text{old})}|} \geq \frac{|\mathbf{I} \wedge \mathbf{w}_J^{(\text{old})}|}{\alpha + |\mathbf{w}_J^{(\text{old})}|} \\ &= T_J^{(\text{old})}(\mathbf{I}) \geq \frac{|\mathbf{I} \wedge \mathbf{w}_j^{(\text{old})}|}{\alpha + |\mathbf{w}_j^{(\text{old})}|} = T_j^{(\text{old})}(\mathbf{I}). \end{aligned}$$

Since $T_j^{(\text{new})}(\mathbf{I}) = T_j^{(\text{old})}(\mathbf{I})$ for $j \neq J$ (because the learning of \mathbf{x}_i has effect at most on category J), then $T_J^{(\text{new})}(\mathbf{I}) \geq T_j^{(\text{new})}(\mathbf{I})$ for $j = 1, \dots, N$. Thus $\Lambda_\pi(\mathbf{x}_{i+1}) = \Lambda_\pi(\mathbf{x}_i) = J$ which means that both presentations will be associated to the same category of π or equivalently to the same FARTNA $\psi(c_p)$ of φ for some p . It follows that conclusion **C4**) of Theorem 1 holds during FARTNA sessions of π .

Proof of C5): As aforementioned, during FARTNA sessions categories can only be created (initial commitment) on $\psi(c_0)$ and not on $\psi(c_p)$ for $p = 1, \dots, C$. Let $P (\geq Q)$ be the number of *all* input vectors \mathbf{x}_i presented to π and $P \rightarrow \infty$. As input vectors are being presented to π , each input vector is equivalently presented to a FARTNA $\psi(c_p)$, for some $p = 0, 1, \dots, C$, and the subsequences of input vectors presented to $\psi(c_p)$ are $X(c_p)$ ($p = 0, 1, \dots, C$), and are being implicitly updated according to their aforementioned definitions. At any moment, the only exemplars of the input training set that may still have any effect on the current state of π are the Q vectors that belong to the current value of set $X(\pi)$. If it also happens that $Q \rightarrow \infty$, then $L(c_p) \rightarrow \infty$ for one or more $p = 0, 1, \dots, C$. However, by conclusion **C5**) of Theorem 1, the number of categories created on $\psi(c_0)$ (or equivalently on π) during FARTNA sessions is *bounded* even if $L(c_0) \rightarrow \infty$. Each CP or CDU operation makes the number of categories decrease by 1 or 0 units, respectively. Thus, assuming that the number of CDC operations is bounded, then the number of categories created on π is *bounded* even if $P \rightarrow \infty$. Thus **C5**) holds. This completes the proof of Theorem 2.

Theorem 2 shows how the properties of FARTNA learning and formation of hyperrectangular categories are directly extended to be valid in the PAFARTNA. It provides the theoretical support for the application of PAFARTNA to map learning in dynamic worlds. In particular, it shows that the new CD operations, while providing support in PAFARTNA for handling changes of Class 2 in dynamic worlds, do not affect the properties of FARTNA sessions of PAFARTNA, which retain the properties of the FARTNA (Section II) and can be used to learn maps of rectangular features in response to sequences of input

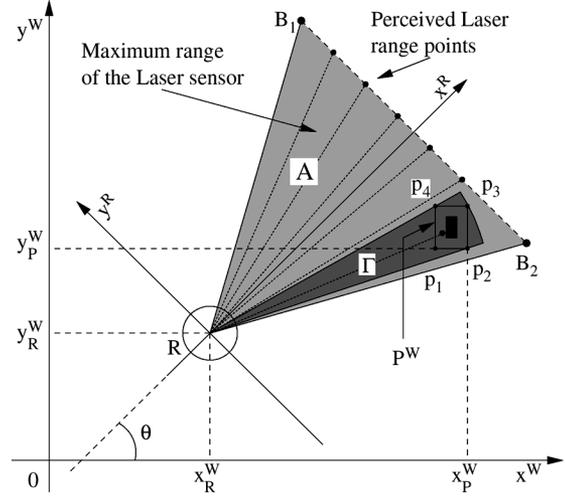


Fig. 4. Illustration of the definition of areas A and Γ . Area A closely approximates the maximum range span of the laser range finder.

sensor data points. Specifically, CP, shrinking CDU, and CDC operations are used (Section V) to provide neural support in PAFARTNA for handling Class 2 changes.

V. PERCEPTION OF OBJECT REMOVALS

In order to apply the PAFARTNA introduced in Section III to dynamic worlds, it is necessary to have a perception method that is able to detect the removal of obstacles from the world. This section describes methods that can be used to analyze if a given FAR, or part of it, still represents an object in the world. The methods assume that a “planar-scanning” laser range finder is used as the sensor to detect object removals.

A. Complete FAR Analysis

1) *Defining the PR of the Laser Sensor*: This is the maximum range spanned/swept by the laser sensor. It is approximated as a triangular region A (Fig. 4), defined by its three vertices, one at the laser sensor center, and the remaining are the two extreme points of the laser light-plan at the maximum angular aperture (points B_1, B_2 in Fig. 4). In this work, the maximum range is 3000 mm and the aperture is 30° .

2) *Position of the Rectangle With Respect to A*: Let P^W be a world FAR and p_1, p_2, p_3, p_4 its corners. It is verified if P^W is completely contained in region A , i.e., if $(p_1 \in A \wedge p_2 \in A \wedge p_3 \in A \wedge p_4 \in A)$. If $P^W \subset A$, the system proceeds with the rectangle analysis; otherwise, the remaining primitives, if any, will be analyzed.

3) *Test the Contents of the FAR*: Let Γ be the smallest (area) circular slice with the same center as the laser sensor, and such that $P^W \subset \Gamma$ (Fig. 4). If there exists some range point p_L perceived by the laser sensor, such that $p_L \in \Gamma$, then it is assumed that the FAR represents an object that still exists in the world (p_L may be a point belonging to the object or “occluding the object”). Otherwise, it is assumed that the obstacle was removed and the FAR P^W should be pruned from the map using the method proposed in Section III. Note that the angular resolution of the laser points determines the smallest object that can be reliably detected.

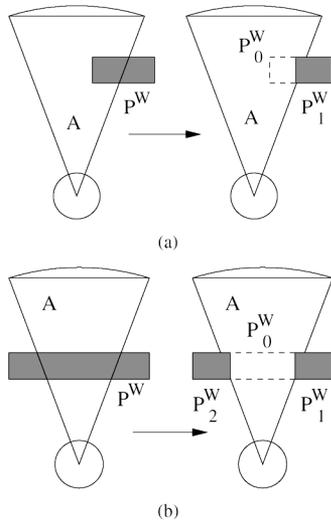


Fig. 5. Perception of object removals: (a) update and (b) splitting of a FAR.

B. Partial FAR Analysis

Rectangle size is an important aspect. Big rectangles and rectangles closer to the robot have a lower probability of being completely included inside the laser area A to be tested. The problem worsens as the maximum range of the laser sensor decreases. In this work, a maximum range of as low as 2 m had to be used to ensure an adequate quality of laser distance readings. These issues may invalidate the operation of the just described algorithm for perception of object removals. The problem is that the system may not be able to properly update the FART world model in response to the removal of an object if the rectangle(s) that represented that object never happen(s) to be completely included inside the laser area A and thus never being a candidate to be analyzed.

Two situations of interest are when two sides of the FAR are intersected by one [Fig. 5(a)] or two [Fig. 5(b)] radial border lines of the laser area A . The general method to deal with these cases is to partition the FAR of interest P^W in the biggest subrectangle completely included in A , $P_0^W \subset A$, and a collection of P_1^W, \dots, P_n^W other rectangles, all with sides parallel to the original rectangle. When one of the situations depicted in Fig. 5(a) or (b) occurs, the subrectangle completely inside the laser area P_0^W is analyzed separately to verify if it still represents an object.

1) *Rectangle Shrinking*: When the situation depicted in Fig. 5(a) occurs, it is processed as follows: If P_0^W does not represent an object anymore, then the original FAR P^W is shrunk to P_1^W by direct updating the corresponding category using the method proposed in Section III.

2) *Rectangle Splitting*: A second case of interest is depicted in Fig. 5(b). If the subrectangle completely inside the laser area P_0^W does not represent an object, it can be removed. For that purpose, two steps are performed using the methods proposed in Section III: First, the original FAR is shrunk to P_1^W and second, a new FAR is *directly created* to represent P_2^W .

VI. NAVIGATION ARCHITECTURE

Core of the Architecture: This section overviews a navigation architecture [5], [12] into which the PAFARTNA map building method was integrated, and that served as a test-bed for the method. The new PAFARTNA extends the navigation architecture in order to permit navigation in general dynamic worlds exhibiting changes of both Classes 1 and 2. The original core from which the architecture was developed is the parti-game learning approach [31], [5]. The system can, simultaneously, learn a model of its environment and learn to navigate to a goal region in an unknown world, having the predefined abilities of doing straight-line motion to a specified position and obstacle detection (not avoidance). In a selective and iterative process, the robot moves/explores and learns an MR partition model of the world P composed of rectangular cells. It begins with a large partition, and then increases resolution by subdividing cells (e.g., Fig. 6) in areas where the learner predicts that a higher resolution is needed. Cells are organized in a kd tree, for fast state-to-cell mapping [5]. For each cell i there is a set $\text{NEIGHS}(i)$ of (cell-) neighbors of i . The robot path is planned to traverse a sequence of cells to reach the goal. The straight-line motion ability is used as a greedy controller for cell-to-cell motion. The request to move to the next (neighboring) cell on the path may fail — usually due to a detected obstacle. A database D of cell outcomes observed when the robot aims at a new cell is memorized and maintained as a collection of $\text{OUTCOMES}(i, j)$ sets, accumulating experience in real-time. $\text{OUTCOMES}(i, j)$ is the set of cells that were previously observed to be attained when the system was on cell i and aimed at cell j . D can be projected to the future as a set of plausible navigation outcome possibilities. Wherever there is absence of observed experience, the optimistic assumption is used in D [5]. The combination of partition P and database D constitutes the *parti-game (world) model*.

Database D is in turn used to plan the sequence of cells to reach the goal cell, using a game-like minimax shortest path approach. The next cell on the path is chosen taking into account a worst case assumption, i.e., we imagine that for each cell we may aim, an imaginary adversary is able to force the worst next cell outcome in D . In this way, we always aim at the neighboring cell with the best worst-outcome. For this purpose, the minimax shortest cell distance from cell i to goal $J_{WC}(i)$ is computed [5] using dynamic programming methods. For choosing spatial resolution, cells are split when the robot is caught on a losing cell—a cell for which the distance to the goal cell is ∞ , i.e., for the current resolution, the game of arriving at the goal cell is lost. In these situations, as explained in [5], cells in the border between losing and nonlosing cells are split. Cells which have just been split must be subject to forgetting of accumulated cell-outcome experience. This induces further and more detailed local exploration in places the system had difficulties to navigate [5].

The global world model is dynamically learned and updated from sensor data. The robot path is globally planned to optimize the minimax shortest path criterion over the underlying current parti-game global world model. Thus, the navigation architecture exhibits global optimality characteristics.

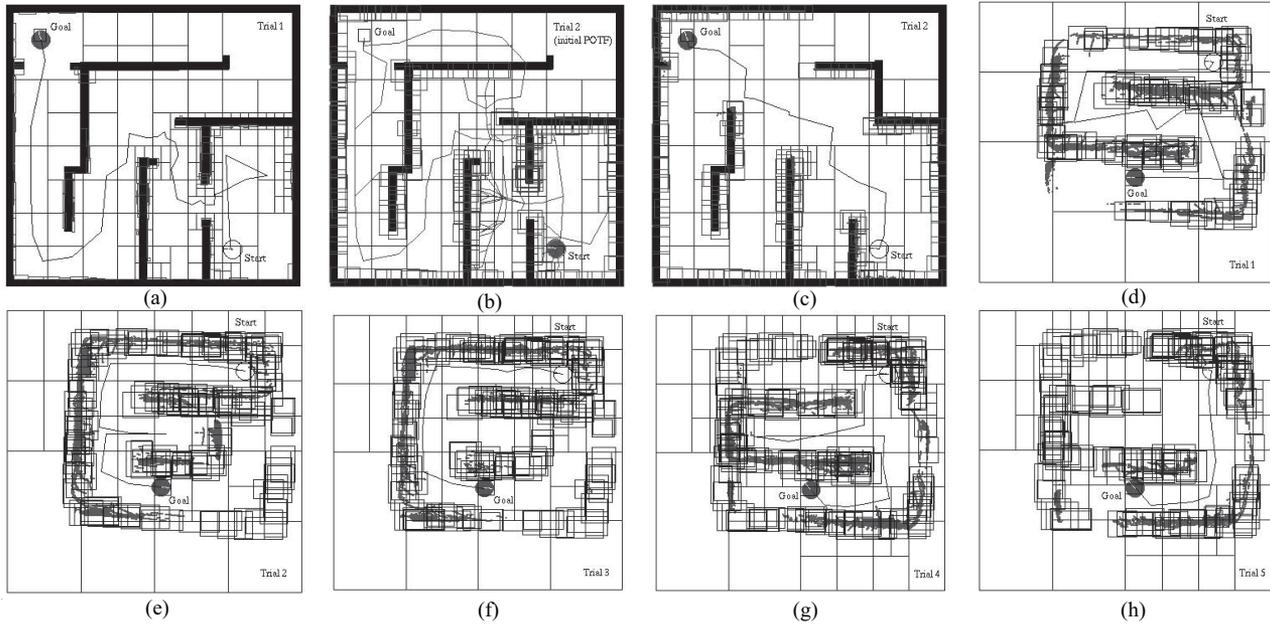


Fig. 6. Experiment 1: (a) Navigation trajectory on Trial 1, (b) initial POTF at the beginning of Trial 2, and (c) trajectory on Trial 2. Experiment 3: Navigation trajectories on (d) Trial 1, (e) Trial 2, (f) Trial 3, (g) Trial 4, and (h) Trial 5. In (b), a predictive trajectory crosses an obstacle because the world model (not the world) is used for predictions, and that part of the obstacle is not yet represented in the FART model because it has not yet passed inside the robot's PR.

Learning A Feature Map of the World: The rectangular features, or equivalently the FARs, extracted and maintained using PAFARTNA and the other methods motivated and presented in Sections I–V, form what is defined as the *FART (world) model*. This model permits to take further advantage of the received sensor information. To provide a safety distance to obstacles, the architecture expands the FARs with a security border gap when performing motion planning. The parti-game and FART models form the (*overall*) *world model*.

POTF: The architecture was extended by introducing POTF. A distinction between a *predictive mode*, i.e., POTF, and a *real mode* is established. Priority is given to predictive mode, where exploration is performed using the learned FART model, instead of using the real robot and world. Only after a successful predictive path (cell sequence) to goal is found (predictive arrival) will the real robot try the corresponding trajectory. This allows a very significant reduction on the time-consuming exploration effort that is associated with searching the world with a real robot. In both modes, path planning is performed using the parti-game approach with the parti-game model being incrementally updated, according to the results of both predictive and real exploration. However, only in real mode is the FART model incrementally updated, because only in this mode is real sensor data available for this purpose. In all experiments of this paper, POTF was performed at the beginning of all cell aims (cell-to-cell motions).

Dynamic Cells Merging: Besides its own contribution, the PAFARTNA also enabled the integration of new developments into the parti-game model in order to have a more suitable application in dynamic worlds. One such development is the local merging of parti-game cells when the world becomes locally “less complicated.” Whenever a complete or partial FAR is removed from the FART model, this is a sign that some obstacle in

the world has disappeared, making the world less cluttered and complex locally. The system clearly identifies this as an opportunity for simplifying the partition model by lowering local resolution through the merging of a selected set of local cells. This method introduces the following advantageous characteristics: Becoming an adaptive MR method—increase or decrease the partition resolution to better adapt the model according to variations on the spatial distribution of local clutter and complexity of the world (comprising changes of Classes 1 and/or 2). This triggers a selective forgetting of cell-outcome information which in turn induces new exploration (mostly done in predictive mode), and which enables the system to take advantage of better navigation paths that may have become available after a removal of obstacles. Thus, improvements in both the PAFARTNA and MR models overcome the prior limitation of the system in response to world changes of Class 2 (Section I). See [12] for a more detailed discussion on the operation, impacts on the system, and benefits of the dynamic cells merging method.

VII. EXPERIMENTAL RESULTS

The methods proposed in this paper were successfully applied to learning maps and navigating in dynamic environments, using simulated and real robot data. The experiments here presented were conducted using a Nomad 200 robot, the PAFARTNA, and the navigation architecture of Section VI. They demonstrate the feasibility and effectiveness of the proposed methods. The infrared and laser sensors of the robot were used to create new FAR features, and the laser was used for the obstacle removal perception mechanism (Section V).

The experiments were organized as a sequence of trials to navigate to goal. Only the first trial starts with an empty world model and then the model is continuously updated during the sequence of trials. The system requires the knowledge of the

current robot location. However, this paper does not deeply address the problem of mobile robot localization. Accumulation of encoder information is used to perform robot localization. Even though this simple approach induces errors, it was sufficient to experimentally validate the feasibility and effectiveness of the proposed methods of map learning and navigation in changing worlds. The results of Fig. 6(a)–(c) show an example of one simulation experiment. Figs. 6(d)–(h), 7(a)–(c), and 8(a)–(c) present three examples of real-robot experiments. In all experiments, the dimensions of the state-space were 7.2×7.0 m, and POTF was performed at the beginning of all cell aims. All the figures include the FARs and the constructed MR partition model. Only in Fig. 6(a) are the FARs represented without the security border gap (SBG). In Experiments 1 and 2–4, the SBG was 100% and 90% of the robot radius, respectively. Figs. 6(b) and 8(c) present the subset of POTF trajectories that were performed at the beginning of the respective trials. Figs. 6(c)–(h), 7(a)–(c), and 8(a)–(c) present the points perceived by the infrared and laser sensors. These points correspond to objects that have passed inside the PR during the respective trials. Points are cleared from the picture at the beginning of each trial. FARs that do not have sensor points depicted inside indicate areas 1) where objects have been perceived into the FART model during previous trials, but 2) have not passed inside the PR during the current trial.

Fig. 6(a)–(c) presents the results of Experiment 1. In Trial 1, the system explores the world and constructs a resulting model. The POTF trajectories at the beginning of Trial 2 [Fig. 6(b)] give a clear example of how the predictive exploration works and significantly decreases the real-robot exploration effort. The parti-game model is updated during POTF exploration. Thus, after the system finds a predictive solution-path to goal, a new POTF exploration starting from the same location may attain a different solution path [e.g., Fig. 6(b)]. The system waits for a stable solution before the real robot tries it. In Trial 2, shortly after the beginning, on the central part of the world, some obstacles were removed and some new obstacles were created [Fig. 6(c)]. After these changes have been observed inside the PR of the robot, the methods introduced in this paper enabled the FART world model to be correctly updated by removing (in this section when “remove FARs” is mentioned, it is meant “remove and/or shrink FARs”) and adding FARs in response to these changes of Classes 2 and 1 in the world. This also enabled the overall navigation system to discover a shorter path to the goal. Clearly, without FARs removals and updates (possible with the proposed PAFARTNA) this better path would not be used because the FARs would block a subset of the POTF trajectories. It can also be observed that, in response to the removal of FARs, the cells merging method proposed in Section VI has triggered the merging of a set of cells (central region), thus lowering the local resolution consistently with the decrease in local complexity and clutter of the world.

Fig. 7(a)–(c) presents three snapshots that illustrate the dynamic building of the environment map in the real-robot Experiment 2. Movies presenting both Trials 1 and 2 of Experiment 2 can be found in [32]. In Trial 2, there is a change of Class 2 in the world: An object is removed in the central left area of the world [Fig. 7(c)]. This change is appropriately handled by the map

learning and navigation system. From Fig. 7, it is understood that the new free-space area and the optimality characteristics of the navigation architecture (Section VI) lead to the attainment of a better (shorter) navigation path in Trial 2, as compared to the path of Trial 1. This induced a total elapsed time (TET) in Trial 2 that is only 59% of the TET of Trial 1.

Fig. 6(d)–(h) presents the results of the real-robot Experiment 3. In Trial 1 [Fig. 6(d)], the robot explored the world and constructed a model in its navigation to goal. In Trials 2 and 4 [Fig. 6(e) and (g)], obstacles were introduced blocking the previous solution-path but other objects were removed opening new paths to goal. These changes in the distributions of objects can be observed by comparing FART models (the FARs) attained at the end of the various trials. The experiment demonstrates that not only the FARTNA perceives new FARs into the model in response to changes of Class 1 in the world, but also the new proposed PAFARTNA now enables FARs representing previously occupied space to be removed from the FART model in response to changes of Class 2. These two types of updates occur as soon as new objects or areas of removed objects fall inside the PR of the robot sensors, respectively. In Trials 2 and 4, the robot enters dead-end situations but then escapes and attains a new path to goal after all relevant new blocking obstacles and new free-space areas have been observed inside the PR, and thus have been perceived and integrated into the FART model. In Trial 3 [Fig. 6(f)], there are no changes in the world and, in Trial 5 [Fig. 6(h)], another small object is removed close to the starting location. This did not significantly change the navigation possibilities of the robot, and the system converged to a shorter navigation path in these trials. Note that if only FARTNA were used, the motion planning process would have become permanently blocked inside a closed chain of FARs as early as in Trial 2.

Fig. 8 presents the results of the real-robot Experiment 4. In Trial 1 [Fig. 8(a)], the robot explored the world and constructed a model in its navigation to goal. In Trial 2 [Fig. 8(b)], an object was removed opening a new path to goal. The methods introduced in this paper enabled the FART world model to be correctly updated by removing FARs in response to this Class 2 world change. This also enabled the overall navigation system to discover a shorter path to goal. Again, without FARs removals that are possible with the proposed PAFARTNA this better path would not have been used because the FARs would block a subset of the corresponding POTF trajectories. In Trial 3 [Fig. 8(c)], an object close to the start location was removed, which led to the removal of some FARs in this area. As a consequence of this removal, the cells merging method proposed in Section VI adaptively triggered the merging of a set of cells close to the start location, thus lowering the local resolution consistently with the decrease in local complexity and clutter of the world. Additionally, this opened a new possible passage to goal that was explored by POTF. However, this POTF trajectory was blocked by a FAR close to the goal, which led to an attained POTF solution and a real robot trajectory [Fig. 8(c)] both closely similar to the path of Trial 2 [Fig. 8(b)]. In comparison to [5], the experiments of this section show an improved capability to operate in changing worlds (notably with Class 2 changes). The system updates the FART and MR maps according to the dy-

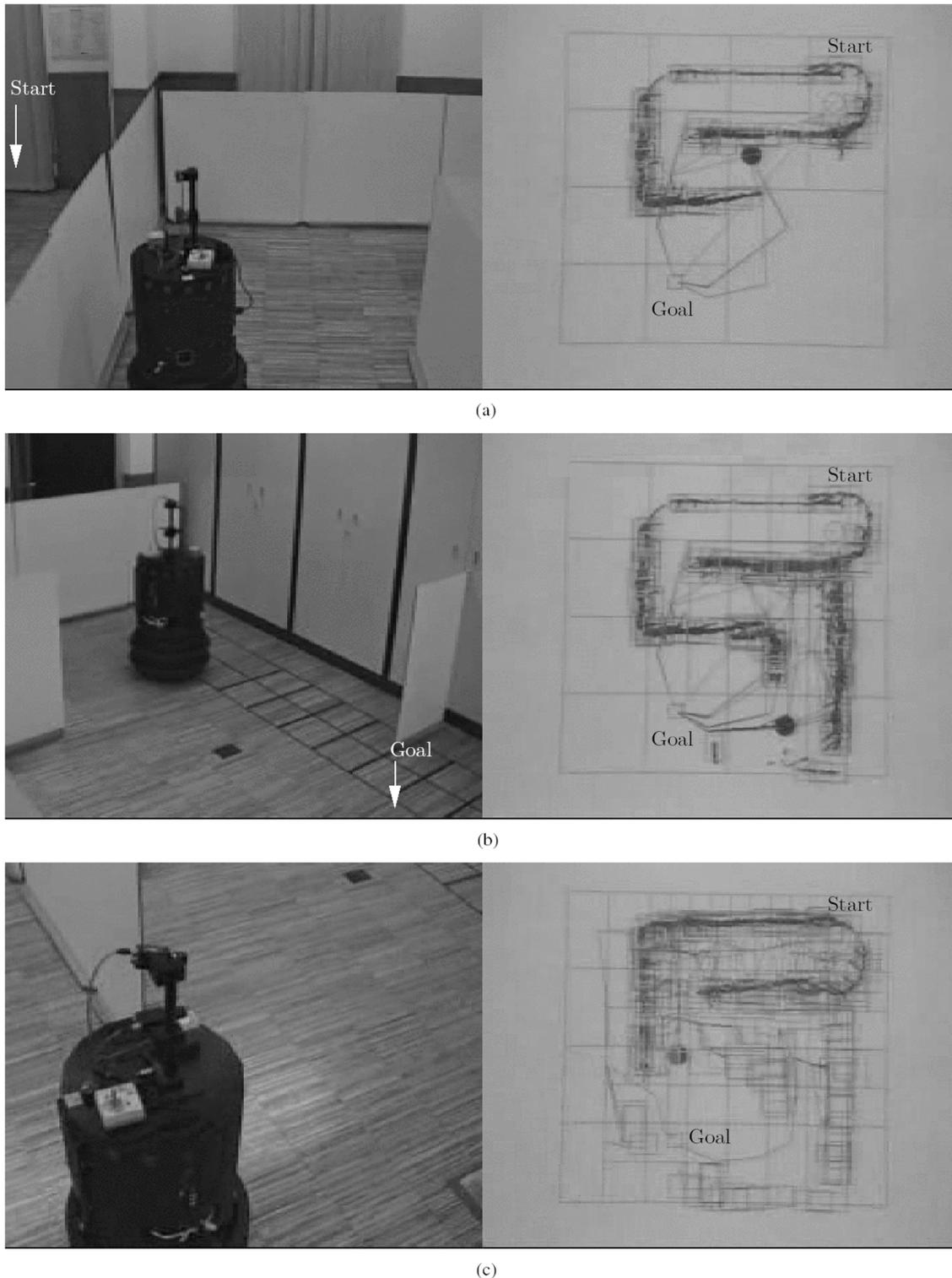


Fig. 7. Experiment 2: Three snapshots of the dynamic building of an environment map. Each snapshot illustrates the real robot navigating in the environment (left) and the graphical interface of the robot control program (right) at the same time instant. (a) Trial 1, Snapshot 1. (b) Trial 1, Snapshot 2. (c) Trial 2, Snapshot 1. (a) and (b) Snapshots taken at 36.2%, and 87.3% of the total elapsed time of Trial 1, respectively. (c) Snapshot taken at 72.5% of the total elapsed time of Trial 2. All the navigation and POTF trajectories, performed from the beginning of the trials until the snapshot instant, are represented.

dynamic changes of both Classes 1 and 2 perceived in the world. In turn, this is used to perform in real-time global dynamic path planning through free space in changing worlds, taking into account the most up to date available information, permitting the attainment of a degree of optimality in navigation.

VIII. DISCUSSION

A relevant current and future research direction on the navigation architecture of this paper and on the navigation in dynamic worlds in general is the development of methods for modeling

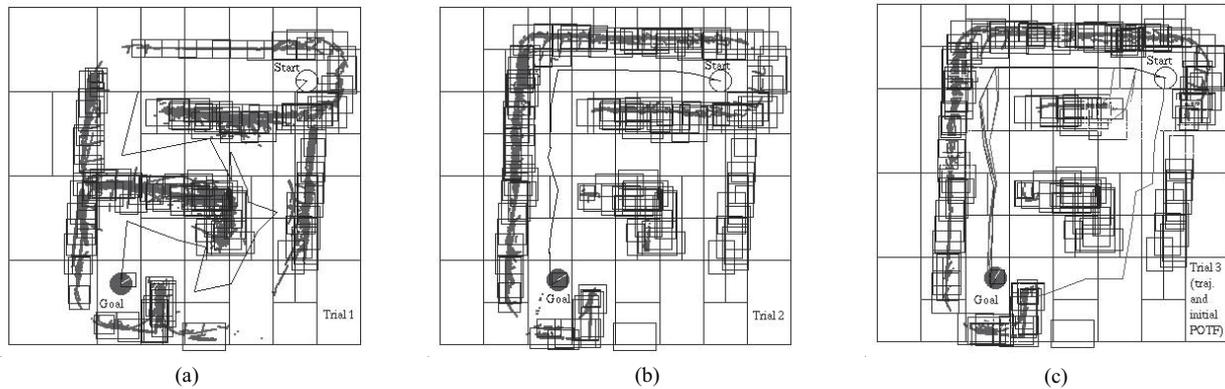


Fig. 8. Experiment 4: (a) Trajectory in Trial 1, (b) trajectory in Trial 2, and (c) initial POTF and navigation trajectory and Trial 3.

and predicting the environment dynamics, such that the degree of anticipatory adaptation to environment evolution may be increased. In [33], it is attempted to predict short-term and long-term motion trajectories of humans and obstacles using polynomial neural networks and probabilistic modeling of points of interest specified in the world. A simulation study is performed where this information is integrated into constant resolution grid models, and a hierarchical partially observable Markov decision process is employed to control the robot motion. In [34], conditional particle filters are used for simultaneous localization and people tracking in an environment previously mapped with a constant resolution grid. Also, the FART world model and ART learning remain open to future research towards its application to localization and to cope with the associated uncertainties. Other relevant topics for future work include developing the self-organizing system in order to permit both 1) unrestricted category alignment directions, 2) to merge “related” categories (e.g., several categories representing one wall) into one “long” category, provided that they meet some “merging” criteria, and 3) more general geometric forms of categories. While with 1) and 2) the complexity of processing and human interpretation of the model would tend to decrease, in 3) these aspects should be considered as development goals.

IX. CONCLUSION

This paper presents the new PAFARTNA that extends the FARTNA in order to make possible the removal, direct update, and direct creation of recognition categories. Relevant PAFARTNA theoretical properties were formulated and demonstrated. Also, a perception mechanism to detect the removal of obstacles from the world was introduced. The two methods were integrated to develop a new map building method that can be applied to dynamic worlds. The introduction of the methods into an overall navigation system was performed which enabled it to appropriately cope with changes of both Classes 1 and 2 in the world. The overall navigation architecture enables the robot to navigate in dynamic environments with optimality characteristics due to its capacity of global path planning using a world model that is dynamically learned and updated online. Experimental results were presented demonstrating the feasibility and effectiveness of the proposed methods for learning maps and navigation in dynamic environments.

REFERENCES

- [1] H. P. Moravec and A. Elfes, “High resolution maps from wide angle sonar,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 1985, pp. 116–121.
- [2] D. Fox, W. Burgard, and S. Thrun, “Markov localization for mobile robots in dynamic environments,” *J. Artif. Intell. Res.*, vol. 11, pp. 391–427, 1999.
- [3] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun, “Map building with mobile robots in dynamic environments,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2003, pp. 1557–1563.
- [4] J. Vandorpe, H. V. Brussel, and H. Xu, “Exact dynamic map building for a mobile robot using geometrical primitives produced by a 2d range finder,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 1996, pp. 901–908.
- [5] R. Araújo and A. T. de Almeida, “Learning sensor-based navigation of a real mobile robot in unknown worlds,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29-B, no. 2, pp. 164–178, Apr. 1999.
- [6] A. Zelinsky, “A mobile robot exploration algorithm,” *IEEE Trans. Robot. Autom.*, vol. 8, no. 6, pp. 707–717, Dec. 1992.
- [7] U. Nehmzow and C. Owen, “Robot navigation in the real world: Experiments with manchester’s fortytwo in unmodified, large environments,” *Robot. Autom. Syst.*, vol. 33, pp. 223–242, 2000.
- [8] H. Hu and M. Brady, “Dynamic global path planning with uncertainty for mobile robots in manufacturing,” *IEEE Trans. Robot. Autom.*, vol. 13, no. 5, pp. 760–767, Oct. 1997.
- [9] G. A. Carpenter, S. Grossberg, and D. B. Rosen, “Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system,” *Neural Netw.*, vol. 4, no. 6, pp. 759–771, 1991.
- [10] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, “Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps,” *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 698–713, Sep. 1992.
- [11] R. Xu and D. Wunsch, II, “Survey of clustering algorithms,” *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.
- [12] R. Araújo and U. Nunes, “Adaptive-resolution map building approach for robot navigation in dynamic environments,” *Mach. Intell. Robot. Control (MIROC)*, vol. 4, no. 2, pp. 61–70, Jun. 2002.
- [13] N. E. Pears, “Feature extraction and tracking for scanning range sensors,” *Robot. Autom. Syst.*, vol. 33, pp. 43–58, 2000.
- [14] J. A. Janét, S. M. Scoggins, M. W. White, I. J. C. Sutton, E. Grant, and W. E. Snyder, “Self-organising geometric certainty maps: A compact and multifunctional approach to map building, place recognition and motion planning,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 1997, pp. 3421–3426.
- [15] J. Mao and A. K. Jain, “A self-organizing network for hyperellipsoidal clustering (hec),” *IEEE Trans. Neural Netw.*, vol. 7, no. 1, pp. 16–29, Jan. 1996.
- [16] Y. L. Ip, A. B. Rad, K. M. Chow, and Y. K. Wong, “Segment-based map building using enhanced adaptive fuzzy clustering algorithm for mobile robot applications,” *J. Intell. Robot. Syst.*, vol. 35, pp. 221–245, 2002.
- [17] Y. D. Kwon and J. S. Lee, “A stochastic map building method for mobile robot using 2-d laser range finder,” *Autonom. Robots*, vol. 7, pp. 187–200, 1999.

- [18] A. Arleo, F. Smeraldi, and W. Gerstner, "Cognitive navigation based on nonuniform gabor space sampling, unsupervised growing networks, and reinforcement learning," *IEEE Trans. Neural Netw.*, vol. 15, no. 3, pp. 639–652, May 2004.
- [19] B. Yamauchi and R. Beer, "Spatial learning for navigation in dynamic environments," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 3, pp. 496–505, Jun. 1996.
- [20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Royal Statist. Soc. Series B*, vol. 39, no. 1, pp. 1–38, 1977.
- [21] M. Kruusmaa, "Global navigation in dynamic environments using case-based reasoning," *Autonom. Robots*, vol. 14, pp. 71–91, 2003.
- [22] S. X. Yang and M. Q.-H. Meng, "Real-time collision-free motion planning of a mobile robot using a neural dynamics-based approach," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1541–1552, Nov. 2003.
- [23] R. R. Murphy, K. Hughes, A. Marzilli, and E. Noll, "Integrating explicit path planning with reactive control of mobile robots using trulla," *Robot. Autonom. Syst.*, vol. 27, pp. 225–245, 1999.
- [24] J. He, A.-H. Tan, and C.-L. Tan, "Modified art 2a growing network capable of generating a fixed number of nodes," *IEEE Trans. Neural Netw.*, vol. 15, no. 3, pp. 728–737, May 2004.
- [25] H. Xiong, M. N. S. Swamy, O. Ahmad, and I. King, "Branching competitive learning network: A novel self-creating model," *IEEE Trans. Neural Netw.*, vol. 15, no. 2, pp. 417–429, Mar. 2004.
- [26] G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A generalized growing and pruning rbf (ggap-rbf) neural network for function approximation," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 57–67, Jan. 2005.
- [27] C. K. Loo, M. Rajeswari, and M. V. C. Rao, "Novel direct and self-regulating approaches to determine optimum growing multi-experts network structure," *IEEE Trans. Neural Netw.*, vol. 15, no. 6, pp. 1378–1395, Nov. 2004.
- [28] J. Borenstein and Y. Koren, "The vector field histogram—Fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 7, no. 4, pp. 278–288, Jun. 1991.
- [29] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.
- [30] R. Araújo, G. Gouveia, and N. Santos, "Learning self-organizing maps for navigation in dynamic worlds," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2003, pp. 1312–1317.
- [31] A. W. Moore and C. G. Atkeson, "The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces," *Mach. Learn.*, vol. 21, no. 3, pp. 199–233, Dec. 1995.
- [32] R. Araújo, Prune-able fuzzy art neural architecture for robot map learning and navigation in dynamic environments 2006 [Online]. Available: <http://www.isr.uc.pt/~rui/pafartna>
- [33] A. Foka and P. E. Trahanias, "Predictive autonomous robot navigation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2002, pp. 490–495.
- [34] M. Montemerlo, S. Thrun, and W. Whittaker, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2002, pp. 695–701.



Rui Araújo (M'06) received the B.Sc. degree (*Licenciatura*) in electrical engineering, the M.Sc. degree in systems and automation, and the Ph.D. degree in electrical engineering from the University of Coimbra, Coimbra, Portugal, in 1991, 1994, and 2000, respectively.

He joined the Department of Electrical and Computer Engineering of the University of Coimbra where he is currently an Assistant Professor. He is a founding member of the Portuguese Institute for Systems and Robotics (ISR-Coimbra), where he is now a Researcher. His research interests include learning systems, sensor-based mobile robot navigation, fuzzy systems, and neural networks, and in general architectures and systems for controlling robot manipulators, and for controlling mobile robots.

Dr. Araújo was the Information Technologies Chair on the *11th International Conference on Advanced Robotics (ICAR-2003)*.