Mobile Robot Feature-Based SLAM Behavior Learning, and Navigation in Complex Spaces

Ebrahim A. Mattar

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.81195

Abstract

Learning mobile robot space and navigation behavior, are essential requirements for improved navigation, in addition to gain much understanding about the navigation maps. This chapter presents mobile robots feature-based SLAM behavior learning, and navigation in complex spaces. Mobile intelligence has been based on blending a number of functionaries related to navigation, including learning SLAM map main features. To achieve this, the mobile system was built on diverse levels of intelligence, this includes principle component analysis (PCA), neuro-fuzzy (NF) learning system as a classifier, and fuzzy rule based decision system (FRD).

Keywords: SLAM, PAC, NF classification, fuzzy rule based decision, navigation

1. Introduction

1.1. Study background

Interactive mobile robotics systems have been introduced by researcher's worldwide. The main focus of such research directions, are how to let a mobile robotic system to navigate in an unstructured environment, while learning its features. To meet these objectives, mobile robots platforms are to be equipped with AI tools. In particular to achieve this, Janglová in [1], describes an approach for solving the motion-planning problem in mobile robot control using neural networks-based technique. The proposed system consists of a head artificial neural network, which was used to determine the free space using ultrasound range finder data. In terms of maps building with visual mobile robot capabilities, a remote controlled vision guided mobile robot system was introduced by Raymond et al. [2]. The drive of the work, was to describe exploratory research on designing remote controlled emergency stop and vision systems for an autonomous mobile robot. Camera modeling and distortion calibration for mobile robot vision was also introduced by Gang et al. [3]. In their paper they presented an essential camera calibration technique for mobile robot, which is based on PIONEER II experiment platform. Bonin-Font et al. [4], presented a map-based navigation and mapless navigation, as they subdivided in metric map-based navigation and topological map based navigation. Abdul et al. [5] have introduced a hybrid approach for vision based self-localization of autonomous mobile robots. They presented a hybrid approach towards self-localization of tiny autonomous mobile robots in a known but highly dynamic environment. Kalman filter was used for tracking of the globally estimated position. In [8], Filliata and Meyer, presented a 3-level hierarchy of localization strategies, and a direct position inference, single-hypothesis tracking, and multiple-hypothesis tracking.

They stated the advantages and drawbacks of these strategies. In [6], Andreja et al. have presented a fuzzy ART neural architecture for robot map learning and navigation. Araujo proposed methods that are integrated into a navigation architecture. Further, intelligence based navigation was further discussed by [9–11]. In [12], Vlassis et al., motioned, "method for building robot maps by using a Kohonen's self-organizing artificial neural network, and describe how path planning can be subsequently performed on such a map". The built ANN related SOM is shown in **Figure 1**. Stereo vision-based autonomous mobile robot was also given by Changhan et al. [13]. In their research, they proposed a technique to give more autonomy to a mobile robot by providing vision sensors. In [14], Thrun reported an approach that integrates two paradigms: grid-based and topological. The intelligent control of the mobile robot, was based on image processing was also given by Nima et al. [15]. In terms of leaning intelligent navigation, intelligent robot control using an adaptive critic with a task control center and dynamic database was also introduced by Hall et al. [16]. This involves development and simulation of a real time controller for an intelligent, vision guided robot. Such models are also necessary for sizing the actuators, tuning the controller, and achieving superior performance. A novel feature of the proposed approach is that the method is applicable to both robot arm manipulators and robot bases such as wheeled mobile robots. Stereo vision based self-localization of autonomous mobile robots was furthermore introduced by Abdul et al. [17]. In reference to the work presented, a vision based self-localization of tiny autonomous mobile robots in a known but highly dynamic environment. A learning mobile robots was shown by Hall et al. [18]. They presented a discussion of recent technical advances in learning for intelligent mobile robots. Novel application of a laser range finder with vision system for wheeled mobile robot was presented by Chun et al. [19], where their research presents a trajectory planning strategy of a wheeled mobile robot in an obstructed environment. A vision-based intelligent path following control of a four-wheel differentially driven skid steer mobile robot was given by Nazari and Naraghi [20]. In this work, a Fuzzy Logic Controller (FLC) for path following of a four-wheel differentially skid steer mobile robot is presented. Color learning and illumination invariance on mobile robots survey was given by Mohan et al. [21]. A major challenge to the widespread deployment of mobile robots is the ability to function autonomously. Two arms and two legs like an ape, was aimed to study a variety of vision-based behaviors. In addition, robot based on the remote-brained approach

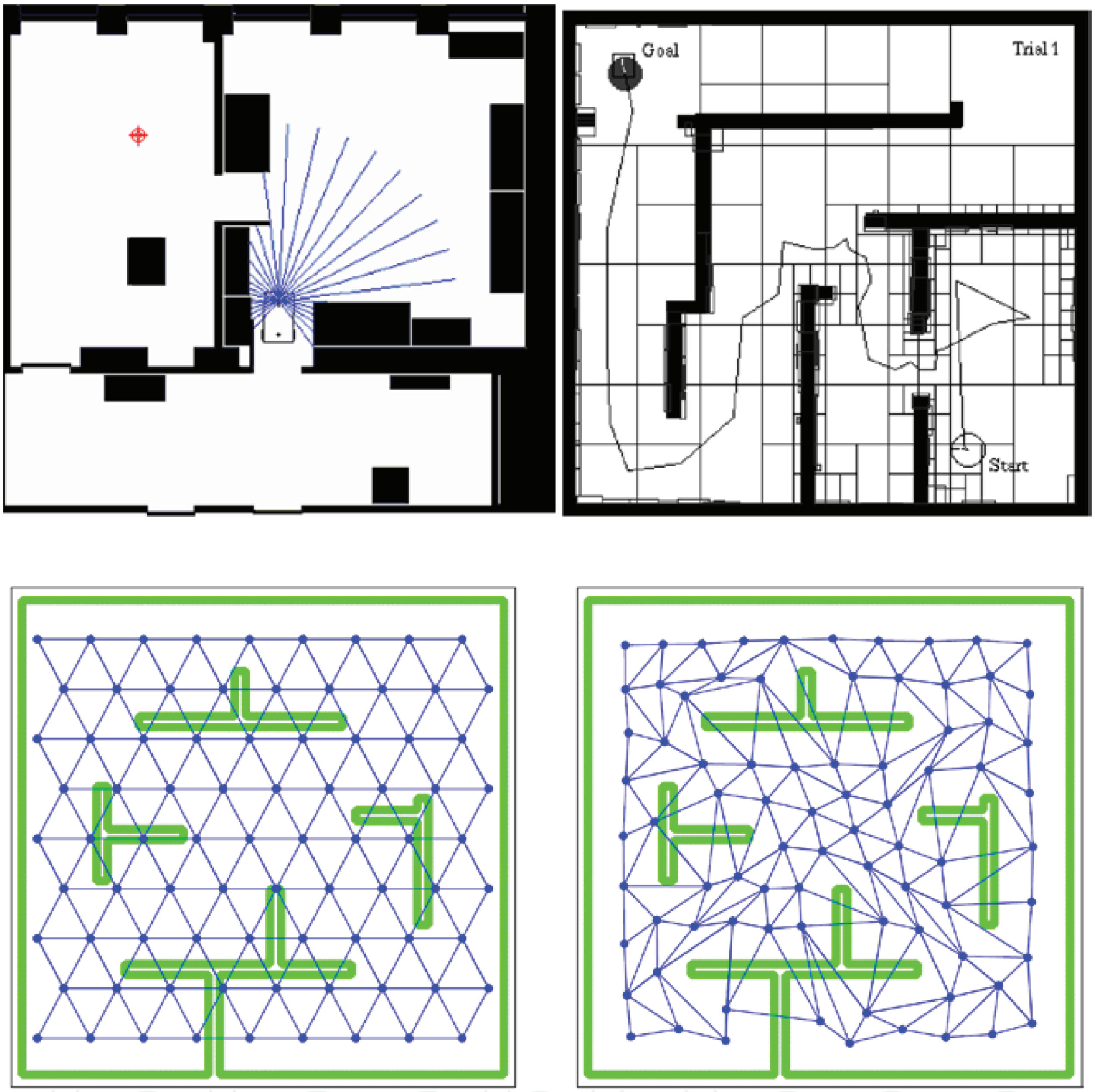
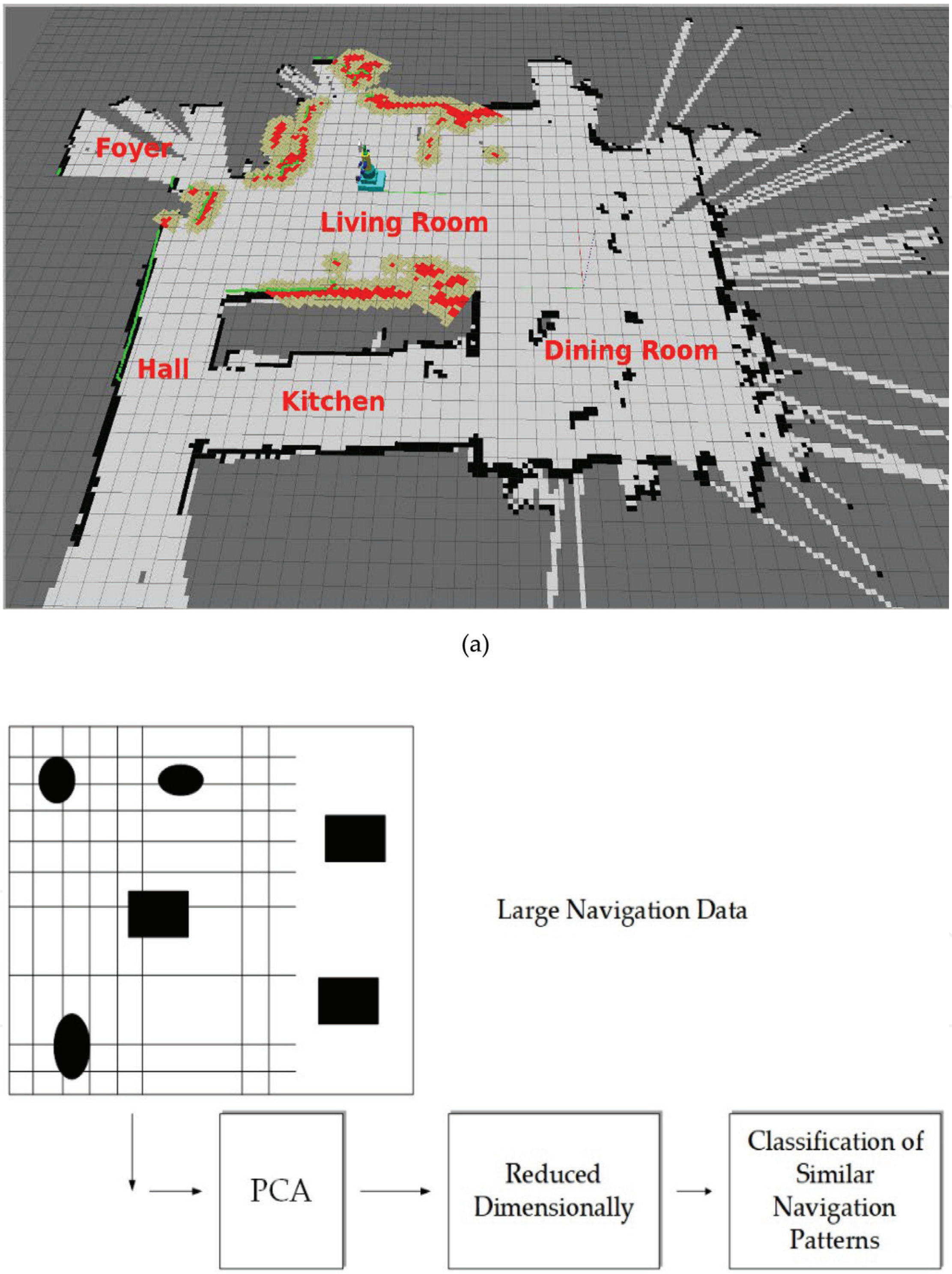


Figure 1. ANN-self organizing maps, for learning mobile robot navigation [1, 6–8].

was given by Masayuki et al. [22]. Localization algorithm of mobile robot based on single vision and laser radar have been presented by Xiaoning [23]. In order to increase the localization precision of mobile robot, a self-localization algorithm based on odometry, single vision and laser radar is proposed. The data provided by odometry, single vision, and laser radar were fused together by means of an Extended Kalman filter (EKF) technique. Mobile robot self-localization in complex indoor environments using monocular vision and 3D model, was moreover presented by Andreja et al. [24], they considered the problem of mobile robot pose estimation using only visual information from a single camera and odometry readings. Human observation based mobile robot navigation in intelligent space was also given by

Takeshi and Hashimoto [25], they investigated a mobile robot navigation system which can localize the mobile robot correctly and navigate based on observation of human walking. Similar work was also given by Manoj and Ernest [26].



(a)

(b)

Figure 2. (a) A sample of SLAM details for learning, picture source robot cartography [27]. (b) Mobile robot training patterns generation. A space is represented by maps space's basis.

1.2. Research objectives

Given the previous background, this current presented work is focusing on learning navigation maps with intelligent capabilities. The system is based on PCA representation of large navigation maps, Neuro-fuzzy classifier, and a fuzzy decision based system. For bulky amount of visual and non-visual mobile data measurements (odometry, and the observations), the approach followed, is to reduce the mobile robot observation dimensionality using principle component analysis (PCA), thus to generate a reduced representation of the navigation map (SLAM), refer to **Figure 2** for details. A learning system was used to learn navigation maps details, hence to classify the representations, (in terms of observation features). The learned system was employed for navigating maps, and other mobile robot routing applications.

2. Building navigation maps

2.1. Simultaneous localization and mapping (SLAM)

SLAM, is a routine that estimates a pose of a mobile robot, while mapping the environment at the same time. SLAM is computationally intensive, since maps represent localization, hence accurate pose estimate is needed for mapping. For creating navigation intelligence capabilities during path planning, this is achieved by learning spaces, once robot was in motion. This is based on learning path and navigation behavior. There are four important stages for building SLAM, this localization, map building and updates, searching for optimal path and planning. Optimal path search is done by A^*, occupancy grids mapping. Given a mobile robot control inputs U as set of controls, $U_{1:k} = (u_1, u_2, \ldots, u_k)$, with mobile parameters measurements (mobile observations) as $Z_{1:k} = (z_1, z_2, \ldots, z_k)$. For odometry, refer to **Figure 3**.

$$x_k = \begin{pmatrix} x_r \\ m_1 \\ m_2 \\ . \\ . \\ m_n \end{pmatrix} \text{ and } C_k = \left(\begin{pmatrix} C_r & \cdots & C_{RM_n} \\ \vdots & \ddots & \vdots \\ C_{M_rR} & \cdots & C_{M_n} \end{pmatrix} \right) \tag{1}$$

In reference to **Figure 3**, the mobile robot starts to move in the space, with a target to research into a predefined final position. While the robot in movement, a SLAM is built, and measurements are recorded from the mobile observations, as $Z_{1:k} = (z_1, z_2, \ldots, z_k)$. All recorded observations are considered as inputs to the PAC, hence they are tabulated into predefined format, for later processing using the PCA algorithm.

2.2. Monte-Carlo (MC) localization

Monte-Carlo localization, is a well-known technique in literature, and still being used for localization parameters estimation. In sampling-based methods, one represents the density by

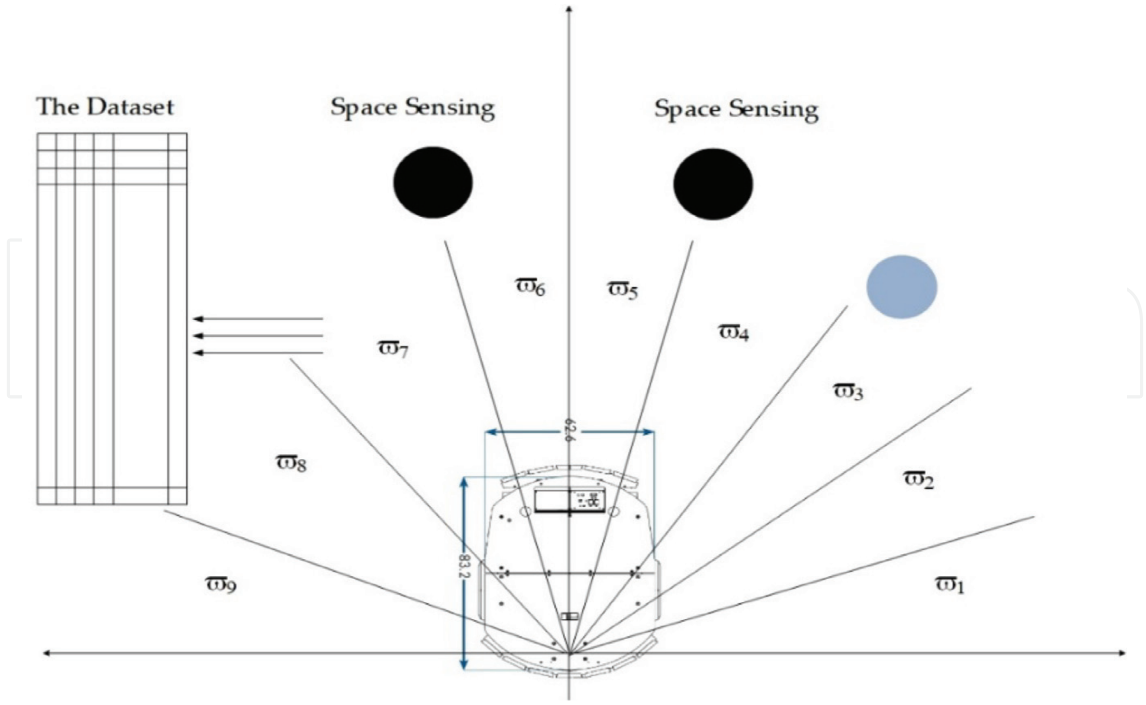


Figure 3. Odometry, generation of navigation patterns for spaces, the (maps).

a set of η random samples of particles. The goal is then to recursively compute at each time step k set of samples of Z_k that is drawn from density $p\left(x_k : Z^k\right)$. A particularly elegant algorithm to accomplish this has recently been suggested independently by various authors. In analogy with the formal filtering problem, the algorithm proceeds in two phases. In the first phase we start from a set of particles $S_{(k-1)}$ computed in the previous iteration, and apply the motion model to each particle $S^i_{(k-1)}$ by sampling from the density $p(x_k \colon S^i_{(k-1)}, u_{(k-1)})$ for each particle $S^i_{(k-1)}$: draw one sample $S^i_{(k)}$ from $(x_k \colon S^i_{(k-1)}, u_{(k-1)})$. We have used a motion model and set of particles $S^i_{(k-1)}$ to build an empirical predictive density function of:

$$(p)' = f(x, y, \theta, \Delta s_r, \Delta s_l),\ (p)' = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{pmatrix} \tag{2}$$

In Eq. (2), we describe a blended density approximation to $p\left(x_k \colon Z^{(k-1)}\right)$. The environment, or the mobile robot space is highly redundant, once used to describe maps.

3. Principle component analysis (PCA)

3.1. PCA based statistically and dimensionality reduction

While in navigation, each traveled path, region, zone, etc. are characterized by diverse behavior (i.e. features), **Figure 4**. If x is matrix of representation for distances and measurements at

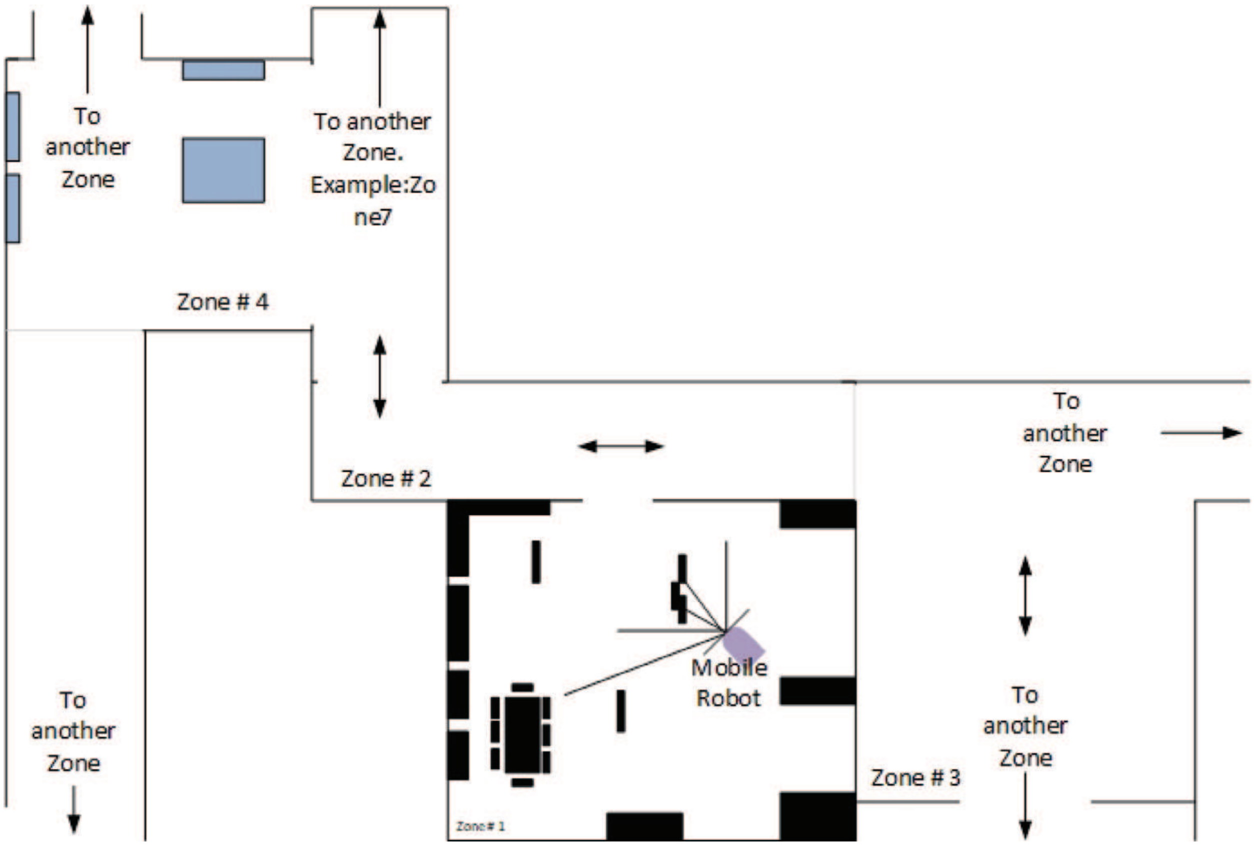


Figure 4. Robot navigation spaces. A representation of navigation segments, zones, areas, ….. (S_1, S_2, S_3, …. S_n, Z_1, Z_2, Z_3, …. Z_m, A_1, A_2, A_3, …. A_n).

each location during navigation, a covariance matrix for the set of maps is considered highly non-diagonal. Mathematically, the previous notation is:

$$\rho = XX^t\left(\begin{pmatrix} \sigma_{11}^X & \cdots & \sigma_{1,w\times h}^X \\ \vdots & \ddots & \vdots \\ \sigma_{w\times h,1}^X & \cdots & \sigma_{w\times h,w\times h}^X \end{pmatrix}\right) \tag{3}$$

σ_{ij}^X represents covariance between distances for location (w) and location (h). There is a relation between covariance coefficients and correlation coefficients. The covariance matrix ρ is expressed as:

$$\rho = \frac{1}{j}\sum_{n=1}^{j}\left(\lambda_n \times \lambda_n^T\right) \tag{4}$$

Since principal components are calculated linearly, let P be a transformation matrix:

$$Y = P^T \times X \quad \textit{and} \quad X = P \times Y$$

In fact, $P = P^{-1}$, since the $\boldsymbol{P}$'s columns are orthonormal to each other, $P^T \times P = I$. Now, the question is, what is the value of P given the condition that S_y must be a diagonal matrix, i.e.

$$S_y = Y \times Y^T = P^T \times \mathrm{X} \times \mathrm{X}^T \times P$$
$$S_y = P^T \times S_y \times P$$

in such away S_y is a rotation of S_x by P. Choosing P as being a matrix containing eigenvectors of S_x:

$$S_x \times P = \Lambda \times P$$

where Λ is a diagonal matrix containing eigenvalues of S_x. In this regard,

$$S_y = P^T \times \Lambda \times P = \Lambda \times P^T \times P = \Lambda$$

and S_y is a diagonal matrix containing eigenvalues of S_x. Since the diagonal elements of S_y are the variance of components of training patterns in the (in navigation) space, the eigenvalues of S_x are those variances.

This is further expanded into:

$$\rho = \begin{pmatrix} \beta_{1,1-}\overline{\beta}_1 & \cdots & \beta_{1,k-}\overline{\beta}_k \\ \vdots & \ddots & \vdots \\ \beta_{j,1-}\overline{\beta}_1 & \cdots & \beta_{j,k-}\overline{x}_k \end{pmatrix} * \begin{pmatrix} \beta_{1,1-}\overline{\beta}_1 & \cdots & \beta_{j,1-}\overline{\beta}_1 \\ \vdots & \ddots & \vdots \\ \beta_{1,k-}\overline{\beta}_k & \cdots & \beta_{j,k-}\overline{\beta}_k \end{pmatrix} \tag{5}$$

Finally, covariance matrix ρ is further expressed by:

$$\rho = \begin{pmatrix} cov(\beta_1, \beta_1) & \cdots & cov(\beta_1, \beta_k) \\ \vdots & \ddots & \vdots \\ cov(\beta_k, \beta_1) & \cdots & cov(\beta_k, \beta_k) \end{pmatrix} \tag{6}$$

The p, is a symmetric (around the main diagonal), and a $\mathfrak{R}^{n \times n}$ matrix. The diagonal of, represents the covariance between the matrix and it self. To recognize normalized dataset for patterns, the covariance ρ of Eq. (6) plays an important role. This can be achieved by getting the eigenvectors of covariance ρ matrix of Eq. (6). Given this background, therefore we need to compute eigenvalues and eigenvectors using numerical approach. For a $\mathfrak{R}^{k \times k}$ matrix ρ, if we search for a row vector $\mathfrak{R}^{k \times 1}$ X that could be multiplied by ρ and get the same vector X multiplied by eigenvalues λ and eigenvector. Matrix ρ transforms the vector X to scale positions by an amount equal to λ, gives a transformation matrix:

$$(\rho X) = (\lambda X) \tag{7}$$

In reference to Eq. (7), and for a $\mathfrak{R}^{k \times k}$ matrix ρ, we shall compute for (k) eigenvalues. The (k) eigenvalue (λ), are hence used for scaling every (k) eigenvectors. Individual eigenvalues *(λ)*, are also found by solving the below defined identity as expressed by Eq. (8):

$$[(\rho - I \times \lambda)\mathrm{X} = 0] \tag{8}$$

where I is an identity matrix. We shall compute for the determinant of Eq. (8), i.e., $|(\rho - I\lambda)| = 0$, while solving for the eigenvalues, λ. While substituting for the (λ) in Eq. (8), and solving for (X), this will result in finding the eigenvector (X), once λ are satisfying the following:

$$|(\rho - I \times \lambda)| = 0 \tag{9}$$

Following Eq. (5) to Eq. (9), and computing for eigenvalues, hence reordering the eigenvalues according to a descending order, this represents a major step in building a PAC based recognition system for the mobile robot dataset generated by the navigation system.

4. Learning system: learning mobile robot navigation maps

4.1. Feature-based SLAM learning

While stating in Section 3 steps for PAC computations, in this section we shall make a focus on building a learning system for the mobile navigation. In reference to **Figure 4**, the mobile robot will be generating navigation dataset. Dataset is generated at different locations, during the mobile robot motion. Navigation dataset involves sensory measurements, odometry, and locality information (i.e. zones, areas, segments …), refer to **Figure 4**. An important part of the dataset, is also the part generated by the SLAM, as already described in Section 2. It is not achievable to encompass all dataset, as this is massive dataset. However, we shall rely on features of the mobile dataset, i.e. the PCA based features of navigation dataset, (the SLAM features). Robot navigation spaces. A representation of navigation segments, zones, areas, ….. are designated as (S_1, S_2, S_3, …. S_n, Z_1, Z_2, Z_3, …. Z_m, A_1, A_2, A_3, …. A_n). For each of such different segments, zones, and areas of navigation, there will be features associated with it. Features during navigation will be used for further processing. This includes a five layers feature learning NF architecture (classifier), and a fuzzy decision system (**Figure 5**).

4.2. Neuro-fuzzy features classifier (NFC) architecture

For the case of fuzzy decision making system, it is essential to incorporate a priori knowledge for the mobile movements in the space. In this respect, many conventional approaches rely on depth physical knowledge describing the system. An issue with fuzzy decision is that knowledge are mathematically impervious. This results in and there is no formal mathematical representation of the system's behavior. This prevents the application of conventional empirical modeling techniques to fuzzy systems, making knowledge validation and comparison hard to perform. Creating a benchmark measure of performance by a minimum distance classifier.

Decision rule adopted by such system, is to assign X to a class whose mean feature vector is closest (Euclidean Distance) to X. A decision is given by:

$$\begin{cases} \|d-\overline{d}_1\| & \leq \ \|d-\overline{d}_2 \Longrightarrow d\in\ h_2 \\ \quad else & \qquad\qquad d\in h_2 \end{cases} \tag{10}$$

Rule-based structure of fuzzy knowledge allows for integrating heuristic knowledge with information obtained from process measurements. The global operation of a system is divided into several local operating conditions. Within each region R_i, a representation:

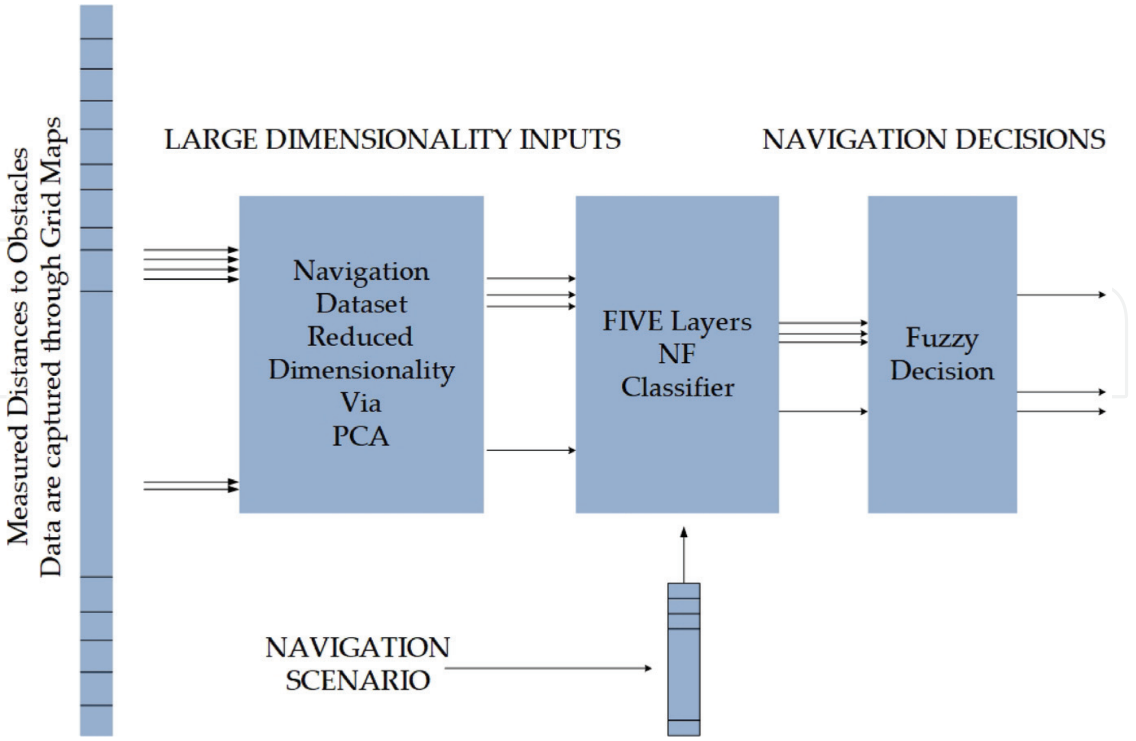

Measured Distances to Obstacles
Data are captured through Grid Maps
LARGE DIMENSIONALITY INPUTS
NAVIGATION DECISIONS
Navigation Dataset Reduced Dimensionality Via PCA
FIVE Layers NF Classifier
Fuzzy Decision
NAVIGATION SCENARIO

Figure 5. The recognition system. PAC for SLAM features computations, a five layers neuro-fuzzy classifier, and last a fuzzy decision based system.

$$R_i\hat{y}_i(k) = \sum_{ij}^{o} \chi_{ij} y(k) + \sum_{ij}^{h} \psi_{ij} u(k) \text{ for h} = 1, 2, \ldots \text{r} \tag{11}$$

In reference to **Figure 6**, for Eq. (11), $\hat{y}_i$ is the computed fuzzy output, u is the system input, in the i^{th} operating region, (h) is the number of fuzzy operating regions. In addition, both (i) and (o) do represent the time lags in the input and the output, respectively, μ_i is the membership function. Finally, χ_{ij} and ψ_{ij} are the few parameters. The membership function for inputs, is constructed in a number of ways.

The fuzzy knowledge system (Neuro-fuzzy) illustrated in **Figure 6**, is an exceptional architecture of network topology. This architecture combines advantages of fuzzy reasoning and the classical neural networks. In its broader sense, the architecture rule (r_i), demonstrates a relation between the input map feature space, and named classes. This is further expressed as follows:

$$Rule\ r_i \ \Longrightarrow if \quad \chi_{si} \ and \ \chi_{sj} \ is \ A_{ij} \ldots\ldots \ and \ \chi_{sn} \ is \ A_{in}, \ \Longrightarrow class\ name\ is\ C_k \tag{12}$$

In Eq. (12), the Gaussian membership function is defined as:

$$\mu_{ij}\left(\chi_{sj}\right) = exp\left(-\left(\chi_{sj} - c_{ij}\right)^2 | 2\sigma_{ij}^2\right)$$

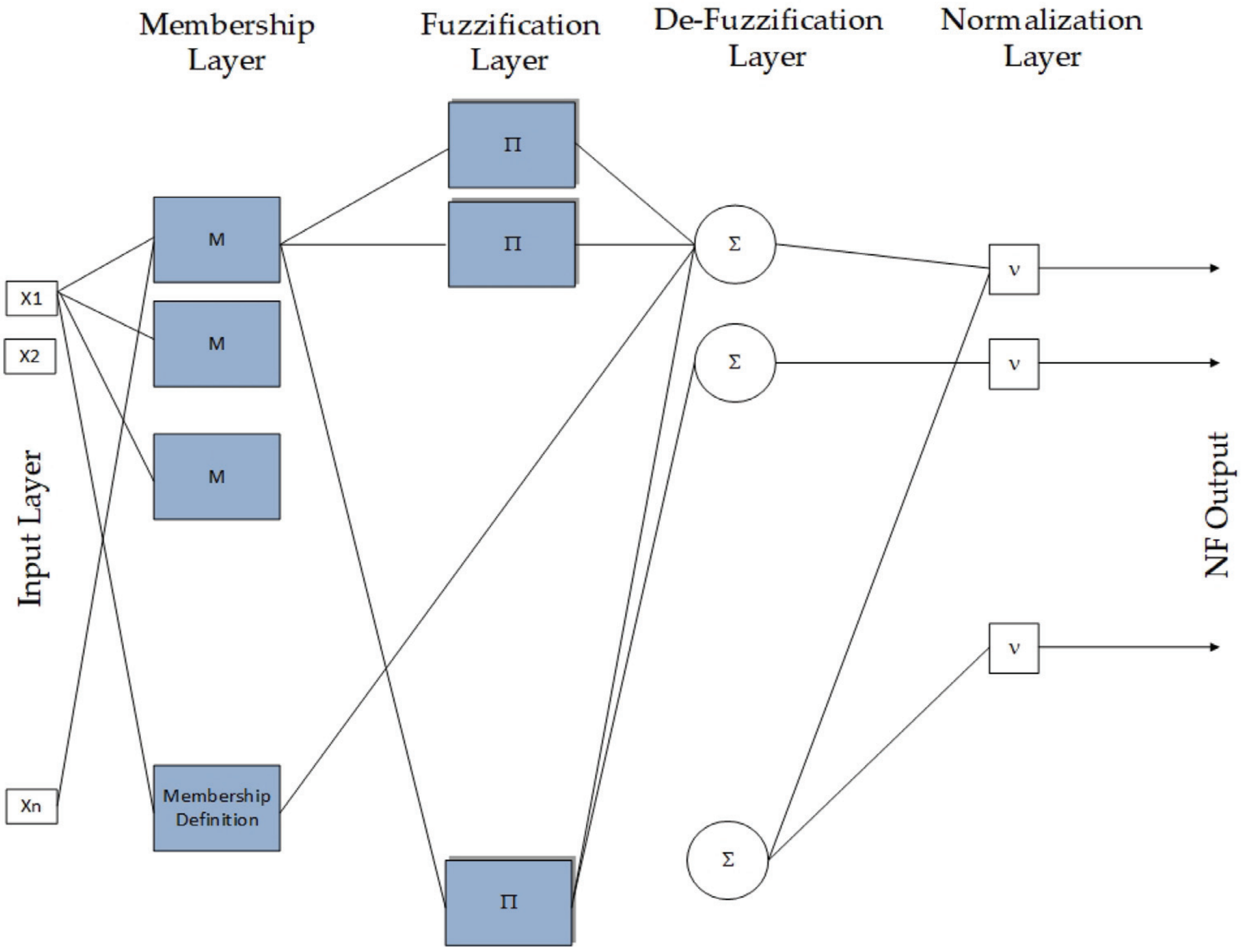


Figure 6. NF classifier architecture. The architecture is used to classify features of navigation maps.

$\mu_{ij}(\chi_{sj})$ is the membership grade of i^{th} rule and j^{th}. That is, the (*if*) parts of the rules are same as in the ordinary fuzzy (*if-then*) rules, (*then*) parts are some combinations of the input variables. For each i^{th} node in this layer is a square node with a node function.

$$\alpha_{is} = \prod_{j=1}^{n} \mu_{ij}(\nu_{sj}) \tag{13}$$

where ν_{sj} is the input to i^{th} node given as the linguistic label (*small, large, .. etc.*) associated with this node function, n is the number of features. The membership is a bell-shape type, and ranged between (1 and 0).

$$O_{sk} = \frac{\beta_{sk}}{\sum_{l=1}^{k} \beta_{sl}} \tag{14}$$

As values of these parameters change, membership shaped functions vary accordingly, thus exhibiting various forms of membership functions on the linguistic label A_i. For every node in this layer, there is a circle node which multiplies incoming signals and sends their product out. Stages of the adopted Neuro-fuzzy classifier, is shown in **Figure 6**.

$$\chi_i = \mu A_i x(k_1) \times \mu B_i y(k_2) \;\; \mathit{for} \;\; i = 1, 2 \tag{15}$$

The output node computes the system output as summation of incoming signals, i.e.:

$$X_i^o = \sum_i \overline{Y}_i f_i \rightarrow X_i^o = \frac{\sum_i \overline{Y}_i f_i}{\sum_i \overline{Y}_i} \tag{16}$$

More precisely, the class label for the s^{th} sample is obtained by the maximum $\{O_{sk}\}$ value as follows:

$$C_s = max_{k=1,2,\dots.K}\{O_{sk}\} \tag{17}$$

The consequent parameters thus identified are optimal (in the consequent parameter space) under the condition that the premise parameters are fixed. The knowledge system's weights are conventionally identified by performing maximum likelihood estimation. Given a training data set $Z^n = (y(k), x(k))_{k=l}^n$, the task is to find a weight vector which minimizes the following cost function:

$$J_n(w) = \frac{1}{n}\sum_{k=1}^{n}(y(k) - \hat{y}(x(k), w))^2 \tag{18}$$

As the knowledge based system, $\hat{y}(x(k), w)$ is much interrelated with respect to the weights, linear optimization techniques cannot be applied. The adopted Neuro-fuzzy system has number of inputs (n) (representing the features) and (m) outputs (representing classes of features). In reference to **Figure 7**, there are dataset about the mobile area and zone of navigation. This is due to the large amount of information coming from the visual system. Here comes the potential of employing the PCA to reduce the dimensionality of the input spaces.

4.3. Fuzzy decision based system

The last stage of the mobile robot maps learning system, is the fuzzy decision system. Within this stage, the hard decisions are undertaken by the mobile robot system during a course of navigation. A fuzzy system is constructed typically from the following rules:

$$\begin{gathered} D = G \cap C \Leftrightarrow \mu_D(a) = \mu_G(a) \bigwedge \mu_{GC}(a), \;\; \mathit{for}\; a \in A \\ a^* = \mathrm{ARG}max\left(\mu_G(a) \wedge \mu_c(a)\right), \dots.. \;\; a \in A \end{gathered} \tag{19}$$

The rules (*if*) parts, are identical to an ordinary fuzzy *IF-THEN* rules. Given an fuzzy inputs $u = (u_1, u_2, \dots., \dots.u_n)^T$ the output $\hat{y}(k)$ of a fuzzy system is computed as the weighted average of the y_s^l, that is:

$$\hat{y}(k) = \frac{\sum_{i=l}^{m} y^l w^l}{\sum_{i=l}^{m} w^l} \tag{20}$$

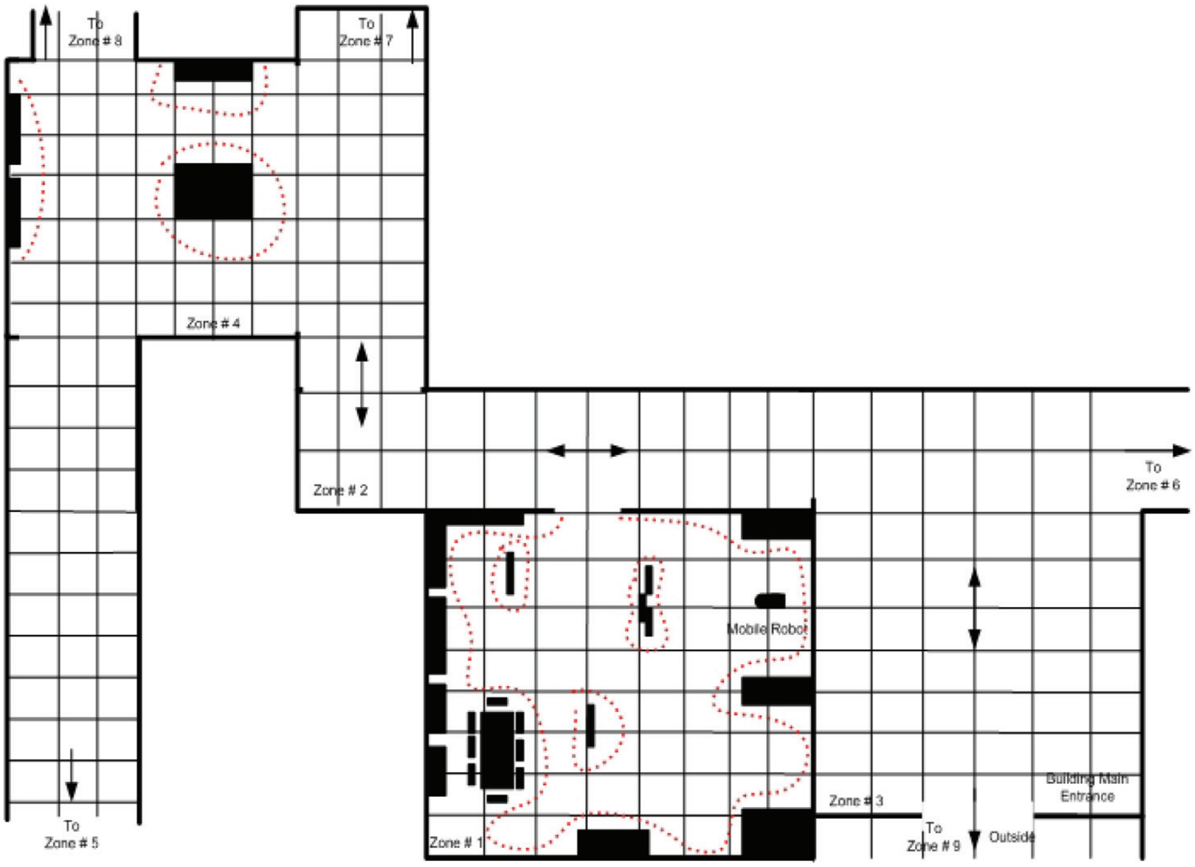


Figure 7. Data from navigation spaces in different zones and areas, (z_1, z_2, z_3, …. z_m, a_1, a_2, a_3, …. A_n), they represent inputs to the PCA.

weights w^i are computed as

$$w^i = \prod_{i=1}^{n} \mu c_i^l(x_i) \tag{21}$$

A dynamic TSK fuzzy system is constructed from the following rules:

$$\begin{gathered} if\ (y(k))\ is\ \left(A_l^P\right)\ \delta\ (y(k_{k-n+l})\ is\ \left(A_n^P\right)\ \delta\ (u(k))\ is\ \left(B^P\right) \\ \Rightarrow \left(y_{n+1} = a_l^p y_k + \dots a_n^P y_{k-n+l} + b^P u(k)\right) \end{gathered} \tag{22}$$

where $\left(A_n^P\right)$ and $\left(B^P\right)$ are fuzzy sets, $\left(a_l^p\right)$ and $\left(b^P\right)$ are constants, $p = (1, 2, \dots., n, u(k))$ are the inputs to the system, and $u(k) = (x_1(k),\ x_2(k), \dots, x_{n+1}(k))$ is the fuzzy system knowledge vector. Typically, the output of the fuzzy decision based system is computed as:

$$x(k) = \frac{\sum_{p=1}^{n} x^p(\beta) v^p}{\sum_{p=1}^{n} v^p} \tag{23}$$

	Inputs		Outputs	
1	x	*Robot Zone*	O_1	*Behavior1*
2	y	*Robot Area*	O_2	*Behavior2*
3	z	*Robot Segment*	O_3	*Behavior3*
4	w	*Delicate Observations*	O_4	*Behavior4*

Table 1. Fuzzy decision based system input-outputs representation.

	Inputs			Outputs	
1	First mobile stimuli	Identification of zone of navigation	Zones identification, and obstacles in zones, z_1, z_2, z_3, z_4, z_5, ……. z_m	First mobile behavior. Behavior1	Rotate around, move robot forward, move robot backward, rotate right, rotate left, …
2	Second mobile stimuli	Identification of area of locality	Areas identification, and obstacles in areas. Obstacles in a1: area floor, obstacles in a2: Out_Corridor, Obstacles in a3: building. Entry, Obstacles in a4:	Second mobile behaviour. Behavior2	Image focus, image capture, … image processing of a scene.
3	Third mobile stimuli	Identification of segment of navigation	Obstacles at different segments within an area, ..	Third mobile behaviour. Behavior3	Video recording zooming with video capture, ..
4	Fourth mobile stimuli	Mobile robot delicate sensory observation	Rotate around, move robot forward, move robot backward, rotate right, rotate left, …	Fourth mobile behaviour. Behavior4	Delicate mobile action.

Table 2 Neuro-fuzzy classifier, input-outputs representation.

where $x^p(k_1)$ is given in Eq. (23) and:

$$v^p = \prod_{i=1}^{n} \mu A_i^p(x_k)\mu B^p(u(k)) \tag{24}$$

The mobile robot training datasets, do consist of four inputs. There are also four output parameters. This is summarized in **Tables 1** and **2**.

5. The experimentation

Within this section, we shall discuss few experimentations results. In order to implement the proposed navigation methodology, the (914 PC BOT) has been reengineered in such a way to allow more control and observations to be communicated through. The main high-level coding was achieved using Matlab. Matlab toolboxes have been integrated in such a way to allow PCA computation, Neuro-fuzzy learning capabilities, and fuzzy decision making routines. This is further indicated to in **Figure 8**.

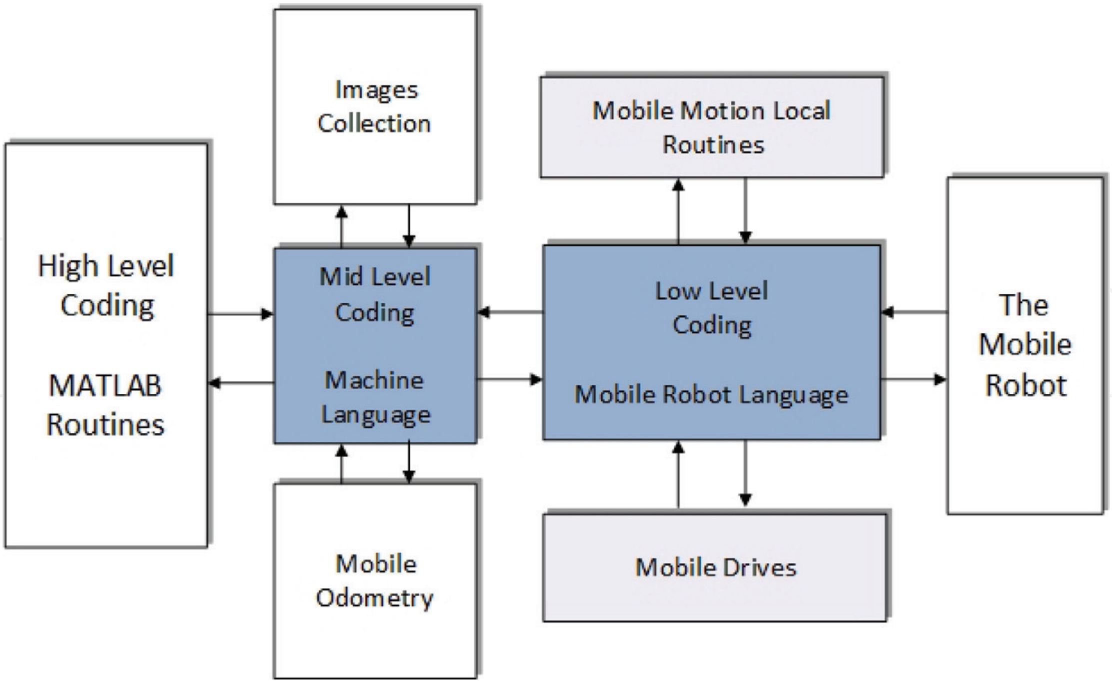


Figure 8. Implementation system hierarchy.

5.1. Behaviour knowledge building

For building the mobile robot behaviour at localities, the mobile system was maneuvered over a space in the laboratory for several trails. Typical physical readings from the robot odometry, and sensory observations were recorded. Typical readings are shown in **Figure 9**. Different mobile behaviors (for learning) were also recorded, beside the odometry, and sensory observations.

5.2. Navigation intelligence

Building the mobile robot navigation intelligence is the next phase. This phase requires blinding all the previous inputs (readings, situations, and behaviors). This will help to take the most appropriate actions. The designated learning and decision making architecture is a Neuro-fuzzy. Typical information, that constitute the Neuro-fuzzy classifier inputs are:

The classifier inputs:

ZONES of navigation. This represents typical zones where the mobile robot is located.

AREAS of navigation. This represents typical areas where the mobile robot is moving.

SEGMENT of navigation. This represents typical segment where the mobile robot is moving.

OBSERVATIONS. This represents typical Observations, the mobile is experiencing at a locality.

The classifier outputs:

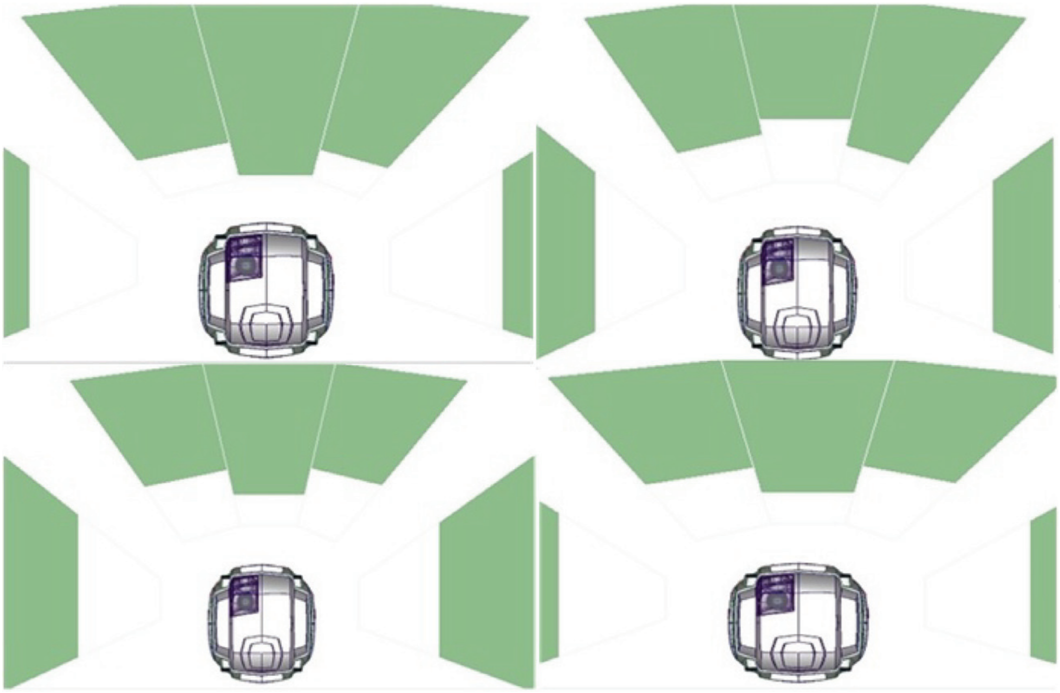


Figure 9. Mobile robot real sensory observations, by experimentation. Dataset have been collected through a number of runs for the PCA.

First mobile behavior. Behavior1, Rotate Around, Move Robot forward, Move Robot Backward, Rotate Right, Rotate Left, ….

Second mobile behavior. Behavior2, Image Processing of a Scene, … .

Third mobile behavior. Behavior3, Video Recording.

Fourth mobile behavior. Behavior4, Delicate Mobile Action.

With such inputs and system outputs, a good degree of a mixture of mobile behaviors can therefore be created. This is further listed below:

The implementation system hierarchy, is shown in **Figure 8**. An adequate of mobile intelligence was created for a mobile navigation within hazardous environments. Inputs to the Neuro-fuzzy decision based system are coming from the PCA network.

5.3. Fuzzy if-then decision system

In addition, the four system inputs-outputs, do represent the system outputs the mobile robot should undertake also at any particular situation. While relying on the fuzzy (if-then) statements, we are able to make further final decision to be undertaken by the mobile robot. Within this sense, we are able to build an (if then statement), as follows:

Typical Fuzzy Rules are:

If (Input_#1 is ….. and Input_#2 is ….) then (Output_#1 is …. and Output_2 is ….) .. ...

If (Input_#3 is ….. and Input_#2 is ….) then (Output_#4 is …. and Output_2 is ….) .. ...

If (Mobile is in zone1 ….. and in area1 ….) then (do image FOUCS).

If (Mobile is in zone1, ….. and in segment2 and, Mobile special task) then (set an ALAM and GAZE).

If (Mobile is in zone3 and in area5 and segment 3, and Image Capture), *… then (do image analyze).*

If (Mobile in zone41 and in area2 and, … and Special task, then (move back).

Given the defined navigation strategy, the mobile robot is able to undertake much detailed navigation and behaviors tasks.

6. Conclusions

Learning mobile robot navigation behavior, is an essential feature, for improved navigation. In addition, it helps to gain further understanding about the spaces of navigation. In this study, navigation maps details have been created while relying on dataset collected by SLAM routines. Due to enormous sensory and environmental data observation to be analyzed during navigation, we have reduced the dimensionally and size of environmental and sensory observation information with PCA technique. Reduction of environment information (*i.e. getting features*), are hence used as learning inputs to a neuro-fuzzy classifier. Examples of Neuro-fuzzy feature inputs are, navigation locations, areas, …, and behaviors related to particular localities. The final stage of mobile robot map building is a fuzzy decision based system. Within this stage, mobile robot navigation decisions are undertaken. With multi-levels of mobile robot sensory and navigation observation dataset, we have designed a learning system for mobile robot maps learning with navigating capabilities.

Author details

Ebrahim A. Mattar

Address all correspondence to: ebmattar@uob.edu.bh

College of Engineering, University of Bahrain, Sukhair, Kingdom of Bahrain

References

[1] Janglová D. Neural networks in mobile robot motion. Inernational Journal of Advanced Robotic Systems. 2004;**1**(1):15-23 ISSN 1729-8806

[2] Raymond A, Tayib S, Hall L. Remote controlled, vision guided, mobile robot system. In: Intelligent Robots and Computer Vision XVI: Algorithms, Techniques, Active Vision, and Materials Handling, vol. 3208; Proceedings of SPIE. The International Society for Optical Engineering; 1997. pp. 126-132

[3] Gang P, Xinhan H, Jian G, Cheng L. Camera modeling and distortion calibration for mobile robot vision. In: Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA), WCICA'08. 2008. pp. 1657-1662

[4] Bonin-Font F, Ortiz A, Oliver G. Visual navigation for mobile robots: A survey. Journal of Intell Robot System. 2008;**53**:263-296. DOI: 10.1007/s10846-008-9235-4

[5] Abdul B, Robert S, Yahya K, Gloom M. A hybrid approach towards vision based self-localization of autonomous mobile robots. In: Proceedings of the International Conference on Machine Vision, ICMV-2007. 2007. pp. 1-6

[6] Araujo R. Prune-able fuzzy ART neural architecture for robot map learning and navigation in dynamic environments. IEEE Neural Network Transactions. 2006;**17**(4)

[7] Vlassis N, Papakonstantinou G, Tsanakas P. Robot map building by Kohonen's self-organizing neural networks. In: Proceedings of 1st Mobinet Symposium; Athens, Greece. 1997

[8] Filliata D, Meyer J. Map-based navigation in mobile robots: I. A review of localization strategies. Cognitive Systems Research. 2003;**4**:243-282

[9] Miftahur R, Rahman M, Rizvi H, Haque Abul L, Towhidul IM. Architecture of the vision system of a line following mobile robot operating in static environment. In: Pakistan Section Multitopic Conference (INMIC 2005). 2005

[10] Tomohiro S, Yoshio M, Taichi K, Masayuki I, Hirochika I. Development and integration of generic components for a teachable vision-based mobile robot. IEEE/ASME Transactions on Mechatronics. 1996;**1**(3):230-236

[11] Ryosuke M, Fumiaki T, Fumio M. Skill acquisition of a ball lifting task using a mobile robot with a monocular vision system. In: IEEE International Conference on Intelligent Robots and Systems. 2006. pp. 5141-5146

[12] Vlassis N, Papakonstantinou G, Tsanakas P. Robot map building by Kohonen's self-organizing neural networks. In: Proceedings of the 1st Mobinet Symposium; Greece. 1997

[13] Changhan P, Sooin K, Joonki P. Stereo vision-based autonomous mobile robot. In: Intelligent Robots and Computer Vision XXIII: Algorithms, Techniques, and Active Vision; vol. 6/6. Proceedings of SPIE. The International Society for Optical Engineering; 2005

[14] Thrun S. Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence Journal. 1998;**99**:21-71

[15] Nima F, Mohammad T, Sadaf S. Intelligent real time control of mobile robot based on image processing. IEEE Proceedings Intelligent Vehicles. 2007;**IV**:410-415

[16] Hall EL, Ghaffari M, Liao X, Alhaj ASM. Intelligent robot control using an adaptive critic with a task control center and dynamic database. In: Intelligent Robots and Computer Vision XXIV: Algorithms, Techniques, and Active Vision, vol. 6384. Proceedings of SPIE. The International Society for Optical Engineering; 2006

[17] Abdul B, Robert S, Jason G, Yahya K, Usman M, Muhammad H, et al. Stereo vision based self-localization of autonomous mobile robots. In: Proceedings (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Robot Vision—Second International Workshop, RobVis., vol. 493-1 LNCS. 2008. pp. 367-380

[18] Hall L, Liao X, Alhaj AM. Learning for intelligent mobile robots. In: Proceedings of SPIE, vol. 5267. The International Society for Optical Engineering; 2003. pp. 12-25

[19] Ya-Chun C, Hidemasa K, Yoshio Y. Novel application of a laser range finder with vision system for wheeled mobile robot. In: Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM. 2008. pp. 280-285

[20] Nazari V, Naraghi M. A vision-based intelligent path following control of a four-wheel differentially driven skid steer mobile robot. In: Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision, ICARCV-2008. 2008. pp. 378-383

[21] Mohan S, Peter S. Color learning and illumination invariance on mobile robots: A survey. Robotics and Autonomous Systems. 2009;**57**(6–7):629-644

[22] Masayuki I, Fumio K, Satoshi K, Hirochika I. Vision-equipped apelike robot based on the remote-brained approach. In: Proceedings IEEE International Conference on Robotics and Automation, vol. 2. 1995. pp. 2193-2198

[23] Xiaoning C, Yuqing H, Gang L, Jin G. Localization algorithm of mobile robot based on single vision and laser radar. In: Proceedings of the 7th World Congress on Intelligent Control and Automation, WCICA'08. 2008. pp. 7661-7666

[24] Andreja K, Sanjin B, Ivan P. Mobile robot self-localization in complex indoor environments using monocular vision and 3D model. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics AIM. 2007

[25] Takeshi S, Hashimoto H. Human observation based mobile robot navigation in intelligent space. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2006. 2006. pp. 1044-1049

[26] Manoj K, Ernest L H. Intelligent robot control using omnidirectional vision. In: Proceedings of SPIE, vol. 1825. The International Society for Optical Engineering; 1993. pp. 573-584

[27] Robot Cartograsphy: ROS + SLAM. http://www.pirobot.org/blog/0015/