Appl Intell
DOI 10.1007/s10489-011-0320-1

Autonomous navigation system using Event Driven-Fuzzy Cognitive Maps

Márcio Mendonça · Lúcia Valéria Ramos de Arruda · Flávio Neves Jr.

© Springer Science+Business Media, LLC 2011

Abstract This study developed an autonomous navigation system using Fuzzy Cognitive Maps (FCM). Fuzzy Cognitive Map is a tool that can model qualitative knowledge in a structured way through concepts and causal relationships. Its mathematical representation is based on graph theory. A new variant of FCM, named Event Driven-Fuzzy Cognitive Maps (ED-FCM), is proposed to model decision tasks and/or make inferences in autonomous navigation. The FCM's arcs are updated from the occurrence of special events as dynamic obstacle detection. As a result, the developed model is able to represent the robot's dynamic behavior in presence of environment changes. This model skill is achieved by adapting the FCM relationships among concepts. A reinforcement learning algorithm is also used to finely adjust the robot behavior. Some simulation results are discussed highlighting the ability of the autonomous robot to navigate among obstacles (navigation at unknown environment). A fuzzy based navigation system is used as a reference to evaluate the proposed autonomous navigation system performance.

Keywords Mobile robot · Autonomous navigation · Fuzzy Cognitive Maps · Intelligent decision systems

M. Mendonça · L.V.R. de Arruda · F. Neves Jr. (✉)
Graduate School in Electrical Engineering and Applied Computer Sciences, Federal University of Technology—Paraná, Curitiba, PR, Brazil
e-mail: neves@utfpr.edu.br

M. Mendonça
e-mail: mendonca@utfpr.edu.br

L.V.R. de Arruda
e-mail: lvrarruda@utfpr.edu.br

1 Introduction

Currently there is a growing interest in the development of autonomous robots and vehicles, due to the great diversity of tasks that they can accomplish, especially those tasks that normally endanger human health and/or the environment [1]. As an example, Mandow and co-workers [2] describe an autonomous mobile robot designed to replace human labor in agriculture inhospitable activities such as spraying insecticides.

The application of robots in several types of jobs is steadily increasing, as a result of recent research, studies and technological advances. In general, during the development of autonomous robots and/or vehicles two main sub problems must be solved:

(1) Navigation that corresponds to the determination of the robot/vehicle position and its orientation in a given time.
(2) Guiding that refers to the path's control to be followed by the robot/vehicle.

Nowadays, there are many works in literature related to the development of such systems using a range of approaches [3–10]. Most of these describe the development of intelligent systems, based on neural networks, fuzzy systems, evolutionary algorithms and other soft computing techniques, in order to build autonomous robots/vehicles or only solve specific navigation or guiding problems [3–8].

Autonomous navigation systems should be able to define a sequence of actions that allow the mobile robots to attain a goal or implement a set of tasks. These robots are equipped with a limited set of sensors and they can move through an unknown environment. Moreover, the robots must execute a list of objectives previously specified while preserving their integrity. These robots must have skills, from generating efficient paths and decision making to navigation in different

Published online: 13 October 2011

environments and/or scenarios [8]. Such autonomous robots have high potential for practical application in industry.

The task of generating efficient paths is complex [3, 4]. In many cases the autonomous system must be able to learn a navigation strategy through interaction with the environment [9, 10]. If the environment is a dynamic one, several navigational abilities for decision-making and dynamic adaptation are required. For example, the ability to decide and perform actions about direction and speed changes in order to avoid obstacles that can suddenly cross the robot's planned trajectory.

In mobile robotics, basically, there are two approaches to address this problem: the reactive and deliberative paradigms [11]. In the reactive approach, the robot has an immediate reaction to any environmental stimuli. Reactive architecture is typically used at a lower level in navigation systems, i.e. the robot's actions and decisions are based only on reading sensors [12]. In the deliberative approach, a navigation system must make decisions about actions that can be performed using a robot's perception combined with initial strategic planning. The advantage of the deliberative strategy is global planning that considers a priori motion safe and trajectory issues for a mapped environment. On the contrary, the reactive strategy takes into account dynamic aspects of the environment and directly applies sensor data to determine the robot's path. However, deliberative architecture generally requires longer processing time, hindering the performance of the robot in real time. Hybrid solutions that combine reactive and deliberative paradigms are currently overexploited [13].

Following this tendency, this work develops a standalone navigation system that combines both active and reactive strategies in which the behavior of the robot in different situations is heuristically modeled. We propose the use of fuzzy cognitive maps to build such a system. As described by [14, 15], the human experience and knowledge about the operation of complex systems can be easily embedded into FCM. Moreover the FCM modeling approach develops cognitive maps based on human experts that have observed the operation of the system and its behavior under different circumstances [14]. In cognitive map models, beliefs or statements about a well known subject are expressed through words or linguistic expressions, linked by simple cause and effect relationships (question/no-question). In fact, FCM are fuzzy-graph structures for representing causal reasoning [16]. Thus FCM can be easily used to model a reactive behavior of a mobile robot but can fail to model deliberative behaviors. To address these limitations of FCM and to allow the use of the technique in navigation, we developed a new type of FCM with a firing mechanism that determines the modification of concepts and causal relationships among them according to the occurrence of events. Moreover, to fine tune the strength of relationships in the presence of unknown environmental changes, a learning algorithm is also introduced into the FCM.

This paper is organized as follows. Section 2 introduces the Fuzzy Cognitive Maps and provides a brief review of its use in autonomous navigation. The new model of fuzzy cognitive map driven by events is proposed in Section 3 and its application to develop an autonomous navigation system is also described. Section 4 presents the simulation results obtained with the proposed navigation system. The system performance is evaluated having as its reference a fuzzy navigation system. Section 5 concludes the paper and addresses future works.

2 Fuzzy cognitive maps

Cognitive maps were initially proposed by Axelrod [17] to represent words, thoughts, tasks, or other items linked to a central concept and orbiting around this concept. Axelrod also developed a mathematical treatment for these maps, based in graph theory, and operations with matrices. These maps can be considered mathematical models for the belief structure of a person or group. They allow inferring or predicting the consequences of the organization of ideas represented in the universe. This mathematical model was adapted for inclusion of uncertainty modeled with Fuzzy Logic by Kosko, generating fuzzy cognitive maps [16]. Like the original, FCMs are signed and directed graphs, in which the involved variables are fuzzy numbers. The "graph nodes" represent linguistic concepts modeled by fuzzy sets and they are linked with other concepts through fuzzy connections. Each of these connections has a numerical value (weight) taken from a fuzzy set, which models the relationship strength among concepts. An example of a cognitive map is given in Fig. 1 and the corresponding connection matrix (equivalent weights matrix) is given by (1).

The concepts of a cognitive map can be updated through the iteration with other concepts and with its own value. Equation (2) drives this interaction where the strength of the

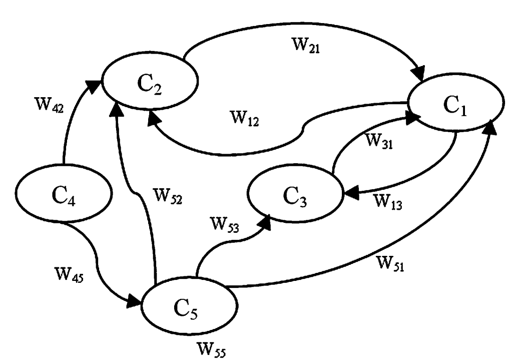


Fig. 1 An example of Cognitive Map

causal relationships is represented by the weight of the summation. After some iterations, the values of the concepts will evolve until they stabilize at a fixed point or a cycle limit.

$$W = \begin{bmatrix} 0 & w_{12} & w_{13} & 0 & 0 \\ w_{21} & 0 & 0 & 0 & 0 \\ w_{31} & 0 & 0 & 0 & 0 \\ 0 & w_{42} & 0 & 0 & w_{45} \\ w_{51} & w_{52} & w_{53} & 0 & 0 \end{bmatrix} \quad (1)$$

$$A_i = f\left(\sum_{\substack{j=1 \\ j \neq i}}^{n} (A_j \times W_{ji}) + A_i^{previous} \right) \quad (2)$$

where k is the counter of iterations, n is the number of nodes in the graph, W_{ij} is the weight of the arc that connects the concept C_j and C_i, A_i ($A_i^{previous}$) is the value of concept (C_i) at the current (previous) iteration and the function $f(\cdot)$ is a sigmoid type function:

$$f(x) = \frac{1}{1 + e^{-\lambda x}} \quad (3)$$

In this context, FCMs are based on a structured knowledge representation through causal relationships that can be computed mathematically by means of matrix operations. This knowledge representation differs from others intelligent systems that are based on if-then rules. However, some core modeling capabilities are missing in the FCM model. FCMs do not describe the dynamics of the relationships. There is no differentiation between long-term events and immediate events. To infer an effect with an FCM model, it is necessary to have simultaneous occurrence of all modeled causes [16]. The lack of temporal concept that is crucial in many applications, associated with non-monotonic inference, has limited the use of FCM models in dynamic application, as robotics and navigation systems.

To circumvent this problem, this paper develops a new type of FCM in which the concepts and causal relations are dynamically changed into a graph from the occurrence of events. Thus, the new model of a fuzzy cognitive map is able to acquire and use heuristic knowledge dynamically. We named this new FCM as event-driven fuzzy cognitive map (ED-FCM) in order to preserve the idea that the causal relationship of an FCM can dynamically change with the occurrence of special events. Moreover, to implement a fine adjustment in the values of the weights to be applied during the occurrence of events, we propose the use of a reinforcing learning algorithm.

These adjustments enable a soft control of the mobile robot during the occurrence of dynamic changes in the environment and/or in the robot (for example, wear and tear of parts). In general, reinforcement algorithms allow an agent to learn directly from their interaction with the environment.

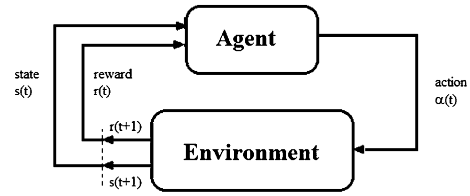


Fig. 2 Reinforcement learning model

The standard model of a system with reinforcement learning is illustrated in Fig. 2. At each time t (called iteration), the robot (agent), through its sensors, establishes its current state s_t, and in accordance with its policy, the robot chooses an action a_t to be performed by its actuators. The action causes a state transition and the agent reaches the next state s_{t+1}. Due to this change, the environment returns with a measure of reinforcement r_{t+1}, that can be a reward if the action was good, or a punishment if the action has been bad. These measures of reinforcement, over time, lead to a correct policy, defining an interactive and dynamic learning process [18].

One difficulty with the reinforcement learning algorithm is finding a policy or strategy that maximizes the return provided by the environment. In fact, the policy is used to map states into actions, i.e., to determine what action a_t should be performed when the robot (agent) is in state s_t, thus defining the agent's behavior over time.

There are different ways to solve the problem of reinforcement learning [18]. The solution adopted in this paper is based on the Q-learning algorithm [18]. The bonus and penalties policy is modeled by simple rules that reflect the dynamic measurements of sensors. A detailed explanation of the reinforcement learning algorithm and the used rule base will be presented in Section 3.

Thus an intelligent system based on ED-FCM with reinforcement learning can be designed to implement several navigation tasks in dynamic environments. With this system, a robot can quickly react to changes, making it suitable for use in a reactive architecture control system. Moreover, the heuristic knowledge, modeled as a cognitive map, can be easily used to plan strategies in a deliberative approach. In the proposed autonomous navigation system, relations are dynamically adapted by FCM rules that are triggered by the occurrence of special events. The resulting inference engine allows non-monotonic reasoning and aims to develop decision making skills in dynamic applications.

2.1 FCM in intelligent obstacle navigation

Some related work using cognitive maps in the robotics area can be found in literature [19–21]. Among them, we can cite the use of a probabilistic FCM in robotic soccer [19]. In this paper, the ability to decide actions taken by players is modeled as an FCM. These actions are related to a player's behavior, such as, kicking the ball in the presence of opponents, among others. This developed FCM was designed from a priori knowledge of experts and applied a probability function to update the concepts of the map. A learning algorithm based on Gradient Descent was also used to adjust weights values of the Probabilistic-FCM, and, a genetic algorithm was used to improve the team tactical strategy.

Yeap and co-workers [20] also applied FCM in mobile robotics. Despite the use of a known trajectory, decisions were necessary due to the errors and uncertainties of robot displacement, such as overspending, sensor reading errors, among others. The paper also presented a brief review of works related to the use of FCM in robotics and intelligent navigation.

Other works related to the use of intelligent navigation in robotics had been cited in Pipe [21]. This paper also presented a cognitive map for implementing a 3-D representation of the environment in which an autonomous robot must navigate.

In all cited works [19–21], the developed FCM models were static ones and applied only to reactive strategy or to model static environment maps. In this paper, the proposed FCM model is dynamic and also implements a deliberative strategy.

Moreover, several studies in the literature described applications of reinforcement learning (RL) in robotics and navigation [22–29]. These works [22–24] used information from computer vision based sensors embedded in RL algorithms to control navigation of mobile robots. The well known Q-Learning algorithm was used in [25] for robotic navigation. The proposed architecture incorporated a mechanism for oversight, allowing exploration and learning based on environmental characteristics and through states mapping. A learning algorithm based on information from the sensors was combined with fuzzy logic and behavior control strategy for autonomous mobile robot mapping [26]. A Real-time robot learning algorithm was presented in [27] and used to allow navigation into a physical environment of mazes. The neural networks and RL were used in [28] to address the problem of structural representation of continuous states to obstacle avoidance in mobile robots. In [29], a supervised learning assisted reinforcement learning (SL-RL) method has been developed to search a movement path with a higher positioning accuracy in the microrobot state space.

Similar to these cited works, we propose a new architecture that uses a fuzzy cognitive map to abstract qualitative knowledge and perform navigational tasks. However the proposed navigation system does not use a priori information about the environment. The FCM model represents ordinary navigation action, such as turning right, left, accelerating and so on. The ability to adapt to environmental changes and make decisions about the presence of random events is achieved through a rule-based system. These rules are triggered in terms of "intensity" from sensor measurements. Moreover, the changes in floor and wheel adhesion, and the provision of barriers and/or targets are environmental variables that are difficult to predict. Thus to minimize the use of uncertain knowledge, the proposed autonomous navigation system implements a reactive architecture, based on the intensity of sensor responses but without any a priori knowledge about the environment. Moreover, a dynamic refinement in the strength of causal relationships which is proportional to sensor measurements is developed. This refinement does not change the FCM modeled actions (turn left, turn right, etc.) and/or the rule base, but it computes the actions shooting time, according to the possible changes of some intrinsic variables. These refinements are reached by applying a Q-learning algorithm [18] in order to tune some related weights in presence of environmental changes.

The combination of heuristic rules with reinforcement learning provides our architecture the following characteristics from subsumption architectures [10]:

- Ability to acquire knowledge through interaction with the environment.
- Ability to adapt its behavior based on sensory information.
- Ability to operate in adverse conditions such as: absence of a complete set of information for pre-planning its behavior, unpredictable interaction with the environment (possibly unknown topology), noise in sensors and actuators.

These properties enable features to successfully implement an intelligent autonomous system. In fact, the subsumption architecture represents a "new" approach to autonomous navigation derived from behavior-based robotics [10, 12]. Navigation systems based on classical artificial intelligence often have the problems of extensibility, robustness, and achieving multiple goals. The subsumption architecture was proposed by Brooks as an incremental and bottom-up approach to deal with these problems [10]. The subsumption architecture reveals a problem in terms of the behaviors exhibited by the robots instead of the stages of information flowing within the controller as in a traditional AI design. In the next sections, the incremental development of a navigation system based on FCM will be detailed, highlighting the abilities cited above (subsumption architecture characteristics). Moreover its application in mobile navigation will also be tested and validated in a simulation environment and with a real robot.

3 ED-FCM navigation system development

From Sect. 2, the FCM knowledge representation is structured through cause-effect relationships that are mathematically computed by means of matrix operations. The FCM is

Table 1 Building FCM models

Step 1	Identification of concepts and their roles (input, output, and selection), their interconnections, and/or selection of relationships determining their causal nature (positive, negative, neutral) and their type (purely causal or time-variant, fuzzy and/or conditional declaration).
Step 2	Initial data acquisition, through expert opinion and/or analysis of a mathematical model, or data analysis.
Step 3	Submission of data concerning the views of various experts to a logical fuzzy system that has the values of FCM weights as output.
Step 4	Construction of the rule base that constitutes the inference engine to predict some FCM weights.
Step 5	Incorporation of a learning procedure to fine adjust the FCM weights.
Step 6	Treatment of information, adaptation and optimization of FCM by adjusting their answers to the desired output.
Step 7	Validation of an FCM model that is tested in the operation conditions of the system modeled.

a graph where the arc weights are estimated based on expert opinion gathering and/or acquired data. Despite the graph representation, some authors argue that FCMs are not able to properly represent real systems, especially the dynamic systems, because FCM models can only express monotonic and symmetric causal relationships. These causal relationships also fail to express the concept of feedback and the temporal aspects associated with real systems [30]. The operation of an FCM model in close cooperation with the real system it describes remains an open issue. In order to operationally extend FCMs to support the close interaction with the real system they describe and consequently become appropriate for robot navigation applications, we propose a new framework where new types of concepts and temporal relationships among concepts are introduced into FCM models.

In accordance with the reccommendation of [30–32], the proposed ED-FCM supports several types of concepts and relationships:

- Level concept: this concept is perfectly represented by an absolute value.
- Variation concept: the concept value represents a variation in a given time.
- Input concept: this type of concept receives input data describing the system state, and can interact with each other to represent system evolution.
- Output or decision concept: these concept values are the result from FCM inference and do not interact with other concepts.
- Causal relations: they represent cause and effect relationships between two concepts, modeled by (1)–(3). The causal relations are calculated through a constant matrix W.
- Conditional declarations: They are cause-effect relations expressed in the form of if-then rules. These relations are naturally time-varying, since the value of the matrix W is resulting from fired rules.

The development of an ED-FCM model follows the steps listed in Table 1. This algorithm is an extension of the algorithm proposed by Kosko [16]. However, the first step of the Kosko algorithm in Table 1 has only causal relations, and step 4 and 5 are lacking. Thus the new algorithm to ED-FCM development in Table 1 is a proposal of this paper in order to mitigate the FCM drawbacks discussed above.

As subsumption architecture, the proposed ED-FCM to robot navigation is developed and structured in two layers. A lower layer is related to direct actions such as turning left, right and acceleration. This layer is modeled as a classical FCM (Kosko's Algorithm [16]), by means of only causal relations and input, output and level concepts. The lower layer models a reactive behavior with decisions based on sensor measurements and are directly sent to low-level controllers. In the upper layer, the occurrence of dynamic changes (events) is represented and the robot reactions and decisions are modeled by means of rules (conditional declaration). This layer models a deliberative strategy where the planned goals are embedded into the rules.

3.1 The reactive layer

To model the lower layer, we identify 3 inputs related to the description of the environment (presence of obstacles) and 3 outputs describing the mobile's movements: turning left, turning right and moving forward. The three inputs take values from the three sensors located at the left, right and front side of the robot. These concepts are connected by arcs representing the actions of acceleration (positive) and braking (negative).

Three decisions are originally modeled. If the left sensor encounters an obstacle on that side, the robot must turn to the right side and if the right sensor encounters an obstacle on the right side, the vehicle turns to the other side. The decision about direction changes implies smooth robot deceleration. The third decision is related to free obstacle environment; in this case the robot follows a straight line accelerating smoothly.

The FCM corresponding to the reactive layer is shown in Fig. 3. The input concepts are LS, RS, and FS that are related respectively to left, right and forward sensors. The concepts LO and RO represent decisions about turning left or turning right if we consider only a reactive strategy. The output concepts are "OUT Left", "OUT Right" and "OUT Front" that

Table 2 Causal relationships in the FCM model at the reactive layer

Causal relation	Description (connection)	Effect	Intensity
w_1	Right Sensor (RS) to Left Output (LO)	Positive	Strong
w_2	Front Sensor (FS) to Left Output (LO)	Positive	Medium
w_3	Front Sensor (FS) to Front Output (FO)	Positive	Strong
w_4	Front Sensor (FS) to Right Output (RO)	Positive	Medium
w_5	Left Sensor (LS) to Right Out (RO)	Positive	Strong
w_6	Left Out (LO) to Right Out (RO)	Negative	Weak
w_7	Right Out (RO) to Left Out (LO)	Negative	Weak
w_8	Right Sensor (RS) to Right Out (RO)	Negative	Weak
w_9	Left Sensor (LS) to Left Out (LO)	Negative	Weak

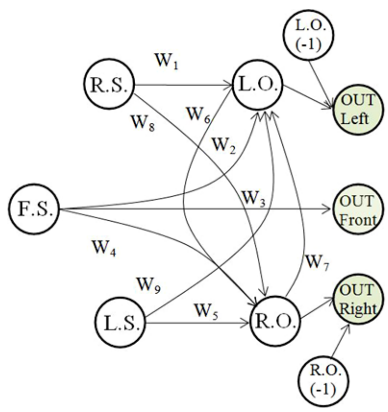


Fig. 3 Low layer of the navigation system based on FCM model of Table 1

represent the next movement of the robot at each direction and the acceleration that will be applied. The acceleration is inversely proportional to the value of output concept.

The values of the input concepts are read from corresponding sensors, which are level concepts. As a fuzzy number, these values are normalized in the interval [0, 1]. The relationships between these concepts are modeled by weights w_1 to w_9 and are calculated by (2). The Table 2 details and describes the causal relationships of the FCM model in Fig. 3.

It is worthy to note from Fig. 3, that there are two other intermediary concepts LO (-1) and RO (-1). These concepts remember the values of decision concepts (LO and RO) in the previous state. Thus, the outputs concepts "OUT Left" and "OUT Right" are computed considering the decision in current time and the actions taken in a past time. As a result, the robot can maintain a trend of movement. A memory of old decisions in the low layer also prevents abrupt direction changes that can result in a zig-zag trajectory.

The reactive system modeled by this FCM (Fig. 3) always has the same behavior independent of variations in the environment and/or in the robot (part wear and tack changes due to floor). To circumvent this drawback, a dynamic adjustment of the causal relationship values (w) can be carried out by means of a learning algorithm. This is not corresponding to a training period as it occurs in neuro-fuzzy applications, but rather a refinement of the dynamic causal relationships (step 5 of the algorithm in Table 1) within a range previously established by experts, according to heuristic rules.

In this paper, the algorithm used to learning step (step 5 in Table 1) is similar to Q-learning. In this algorithm, during each interaction with the environment, the cost $Q(s,a)$ to take an action "a" that drives the system from the actual state "s" to next state "s'" is up-dated by means of the rule [16]:

$$Q(s,a) = Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q\left(s',a'\right) - Q(s,a)\right] \quad (4)$$

where α is a learning factor, γ is a discount factor, r is the value of the reward or punishment and $\max Q(s',a')$ is a prediction of reward in the next state s'. The discount factor γ can be considered as a bonus rate or a probability measure of the agent to go to the next state or a mathematical artifice to prevent an infinite sum in the equation. The learning factor α plays the same role of learning factor in neural network or step factor at gradient descent algorithms.

In the proposed learning algorithm, the reward and punishment policy are modeled by heuristic rules. These rules establish the maximum/minimum values of the causal relationship among concepts in accordance with the intensity of the sensor measurements that represent the environment stimuli.

As an example, if the right sensor (RS concept) measures a very intense stimulus, then the weight (w_1) connecting this sensor to the concept LO, that models the action of turning left, can be increased. This tuning allows a better dynamic behavior of the robot according to changes in the environment.

With this strategy, the values of the causal relationships w_1, w_3 and w_5 at each iteration "k" are computed by the equation:

$$w_i(k) = w_i(k-1) + \alpha \times \left[r + \gamma \times W_{\text{lim}} - w_i(k-1)\right] \quad (5)$$

where the prediction of reward W_{lim} can be a maximum or minimum in accordance with the following rules:

1. **IF** the intensity of the *front sensor* (FS) is greater than a medium threshold **THEN** the W_{lim} applied to compute the relationship w_3 is the maximum value $WF_{\max}$.
2. **IF** the intensity of the *front sensor* (FS) is "smaller" than a minimum threshold **THEN** the W_{lim} applied to compute the relationship w_3 is the minimum value $WF_{\min}$.
3. **IF** the intensity of the *right sensor* is "greater" than a medium threshold **THEN** the W_{lim} applied to compute the relationship w_1 is the maximum value $WR_{\max}$.
4. **IF** the intensity of the *right sensor* is "smaller" than a minimum threshold **THEN** the W_{lim} applied to compute the relationship w_1 is the minimum value $WR_{\min}$.
5. **IF** the intensity of the *left sensor* is "greater" than a medium threshold **THEN** the W_{lim} applied to compute the relationship w_5 is the maximum value $WL_{\max}$.
6. **IF** the intensity of the *left sensor* is "smaller" than a minimum threshold **THEN** the W_{lim} applied to compute the relationship w_5 is the minimum value $WL_{\min}$.

These rules determine the policy to state changes in the ED-FCM weights through reward and punishment. These weights are responsible for making decisions about turning right, left and accelerating the robot. In this way, the current value of these weights depends on the difference between their values in a previous instance and their admissible maximum (minimum) values weighted by a discount factor γ. The increment of the weights also counts the value of the reward or punishment (r) and the learning factor α.

3.2 The deliberative layer

In order to confer adaptive skills to the navigation system, we introduce new concepts associated with the intensity of movement (acceleration and braking) in each direction as the second layer. These concepts are factor type concepts that have their values changed according to the current condition of the robot movement and the occurrence of events that can determine a change in the direction. The role of these factor concepts is to minimize sudden changes in the robot's direction and also to assure that the robot follows the planned goal. For example, when the robot is making a right turn, the decision about "left turn" is softly attenuated. However, this attenuated action must remain only a brief period of time, after this the robot returns to the previous planed goals. The resulting FCM with reactive and deliberative strategy is shown in Fig. 4.

In Fig. 4, the weights connecting the concepts *Right Factor*, *Left Factor* and *Front Factor* are time varying, modeled as a conditional declaration. Their values are obtained by applying the IF-THEN rules described by means of linguistic terms. These rules represent decisions, as if the robot is turning to the right because the left sensor has detected an obstacle in the past, and suddenly the right sensor also detects an obstacle, and the weight (ws_3) that represents the

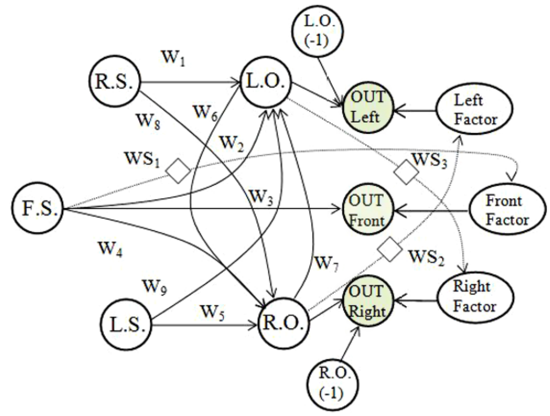


Fig. 4 Proposed ED-FCM

decision about "right turn" becomes small. As a result, the value of the concept "Right Factor" is reduced. If a rule is triggered, the weights will become zero. Finally the outputs of the ED-FCM are the product of the factor concepts and the output concepts (OutLeft, OutRigth and OutFront).

As a result, the proposed ED-FCM navigation system confers to the robot/vehicle the following behavior:

- The autonomous robot moves into an unknown environment from an origin point to an end point (planned goal).
- If an obstacle is detected by the front, left and/or right sensors then the robot must make a decision about a new direction to follow.
- A default navigation decision is following a straight line with constant speed, i.e. lateral movements are used only as a result of obstacle detection.
- During movement, if the sensors do not identify any obstacle, the robot accelerates smoothly in a straight trajectory.
- A motion trend is considered by the average values of movement (trajectory/speed) and the same values from a previous time, which prevents any sharp change in of the navigation direction.
- If the robot is turning in a left or right direction and the opposite sensor detects an obstacle, then the motion trend is maintained but the robot is softly stopped until reestablishing a straight movement.

The final intelligent architecture for the navigation system is shown in Fig. 5. The input interface reads the sensor measurements whose values are inversely proportional to the obstacles distance. The ED-FCM model plays the role of an inference engine that must make a decision about the movement of the robot/vehicle based on values of weights w and w_s. The initial values of ED-FCM weights (w and w_s) were heuristically adjusted (step 6 of the algorithm in Table 1) by observing the robot dynamic behavior during several simulations. The Rule Base block represents the heuristic knowledge to make decisions in the presence of conflict-

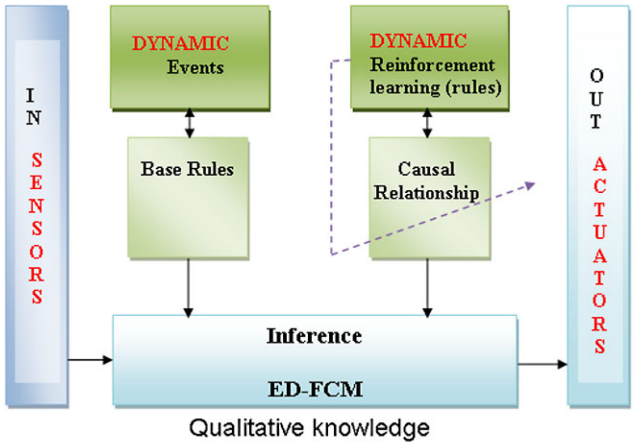


Fig. 5 ED-FCM navigation system architecture

ing events, considering the planned objectives. This base rule represents the deliberative strategy. The Causal Relationship block represents structured knowledge about navigation corresponding to a reactive behavior. Thus, each block represents a source of knowledge/information that may operate simultaneously. According to the inference result that considers the results/decisions from both blocks, control actions are sent to the actuators by means of an output interface.

As discussed above, this architecture is a subsumption that was incrementally built based on robot behavior. This development was facilitated by the use of an ED-FCM model in Table 1 that easily maps a linguistic description of the robot behavior into a graph. This easy translator from behavior to graph is one the advantages of FCM modeling. Unlike neurofuzzy approaches, FCM models do not require a large data set or long hours of training. Moreover, it is easier to incorporate the new knowledge regarding unmodeled situations into the actual navigation system, by modeling the corresponding robot behavior with a new, small FCM, and linking this new FCM with the old navigation system by means of common concepts.

3.3 A fuzzy autonomous navigation system

The first reported uses of fuzzy control in mobile robotics are due to Sugeno and Nishida in 1985 [4]. These authors developed a fuzzy controller able to drive a car along a path delimited by two walls. Since this appearance in the mid-eighties, several journal and conference papers and books have reported successful application of fuzzy logic in mobile robotics with special attention to autonomous navigation systems [4].

In this paper, we recognize the success of fuzzy logic and we also implemented an autonomous navigation system using fuzzy logic. This system is used as a reference to evaluate the performance of the proposed ED-FCM navigation system. Moreover, this type of system can also be developed from expert knowledge as a FCM model [33]. Thus the development of a fuzzy autonomous navigation system (FANS) can assess the difficulty in using the empirical knowledge of specialists to build ED-FCM systems.

The fuzzy navigation system is a generalized fuzzy system (GFS) [33] in which all fuzzy variables are mapped into an interval $[-1, 1]$. The FANS development is based on the same acquired knowledge used to develop the ED-FCM. As the ED-FCM navigation system, FANS has 3 inputs related to the 3 measurements acquired by the left, right and front sensors, and 3 outputs representing the actions left turn, right turn and accelerate (straight movement). Trapezoidal membership functions are used to represent the predicates associated with the intensity of the measurement and actions that are similar to the knowledge represented in the column "intensity" of Table 2: strong, medium and weak. In this point, we report that it is harder for the experts to tune the membership functions of a Fuzzy Controller than to establish the strength of relationships among concepts in an FCM.

An initial rule base with 75 rules was developed to compute the desired speed and direction of the robot. These rules covered all combinations of input variable predicates. Then, a fuzzy rule base reduction and simplification algorithm was applied in order to verify the completeness and the consistency of the fuzzy system [34]. As a result, the retained rule base contains only 23 rules. Examples of some rules are:

- **If** the right sensor measurement is strong **then** the action *turn left* is strong.
- **If** the left sensor measurement is weak **then** the action *turn right* is weak.
- **If** the right sensor measurement is strong **and** the front sensor measurement is strong **then** the action accelerate is "slightly" **and** the action *turn left* is strong.
- **If** all sensor measurements are weak then the action *accelerate* is strong.
- **If** the left sensor measurement is weak **and** the front sensor measurement is weak **then** the action *turn left* is medium.

The Mandani implication is used as inference engine and the Center of Area de-fuzzification method is used for the transformation of the linguistic weight to a numerical value within the range $[-1, 1]$. Figure 6 shows the structure of the fuzzy navigation system implemented with Matlab®. Figure 7 shows the surface corresponding to part of the rule base. This part models the decision about the robot's speed in a straight direction (action *accelerate*) and it is based on the measurements from the right and left sensors.

A fundamental problem of fuzzy systems is the curse of dimensionality. As input dimension increases, the fuzzy rule base increases exponentially [33]. This makes the computational cost, memory, and training data requirements increase. In this sense, the C-code of ED-FCM autonomous navigation system is simple, compact and can be easy embedded into hardware with very limited memory and processing resources. On the contrary, the C-Code of FANS requires additional memory to embed the rule base and a surplus of processor resources to run the Center of Area defuzzification method.

The performance of both navigation systems is expected to be similar. They are designed using the same heuristic knowledge and compute the same outputs from the same inputs. Moreover the decision making of both systems have the same goals (bypass obstacles and reach the arrival point). However the decisions and consequently the trajectories that will be taken by both systems in the same conditions are not necessarily equal, especially in drastic situations. These similarities and differences will be highlighted in the simulated experiments.

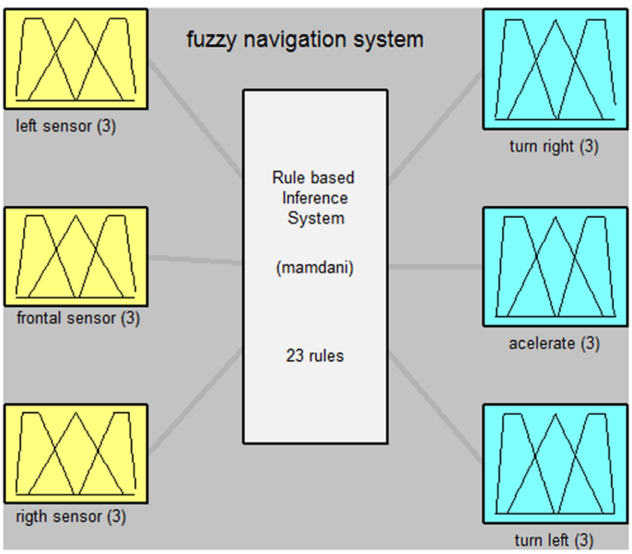


Fig. 6 Fuzzy logic architecture

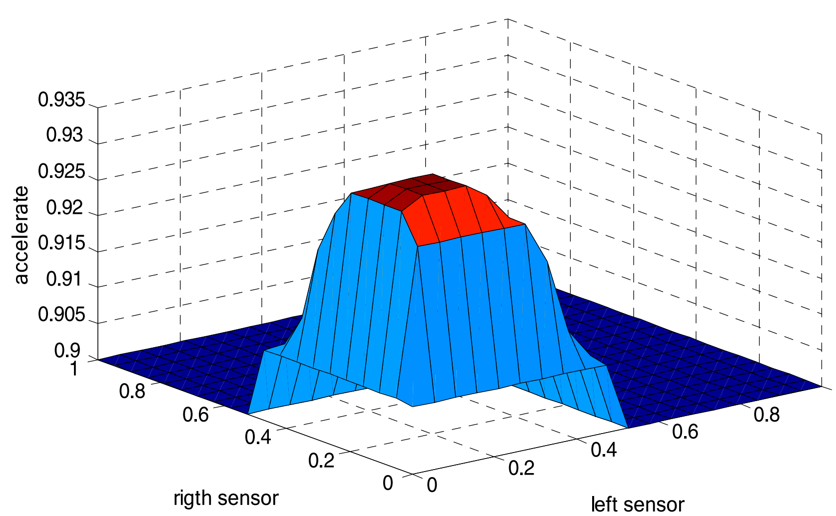
In conclusion, we verify that the development of an ED-FCM system is easier than the development of a fuzzy rule based inference system due to the intuitive nature of knowledge representation by means of cognitive maps rather than membership functions and if-then rules. Moreover the need to check the completeness and the consistency of the fuzzy rule base introduces complexity in the design of the fuzzy logic system. The consistency of an FCM based model is easier to assess, because under a very general condition, fuzzy cognitive maps can be divided into basic FCM modules that are also causally related. Thus, if each small map is consistent the complete map will also be consistent [30].

4 Simulation results

A 2-D animated simulation environment was built using Matlab® to test and validate our proposed navigation systems. The environment corresponds to a 2 m × 2 m square room with fixed and mobile obstacles. The kinematic equations simulating the dynamic behavior of the robot were inspired and adapted from [14]. Indeed, the simulated robot corresponds to a mobile platform with two micro motors, three sensors- one in front and two on each side. These sensors are ultra-sound and only barriers or obstacles in a perception zone can be detected. The intensity of the sensor measurement decreases proportionally to the object's distance. This simulation environment was used initially to acquire knowledge through observation of the robot's behavior in the various situations.

Three experiments in the simulated environment are carried out in order to evaluate the proposed ED-FCM autonomous navigation systems. In these 3 experiments, the performance of the proposed navigation system is compared with the performance of the fuzzy navigation system. Finally, an experiment with a didactic robot [35] is carried out

Fig. 7 Rule base surface associated to the output (accelerate)



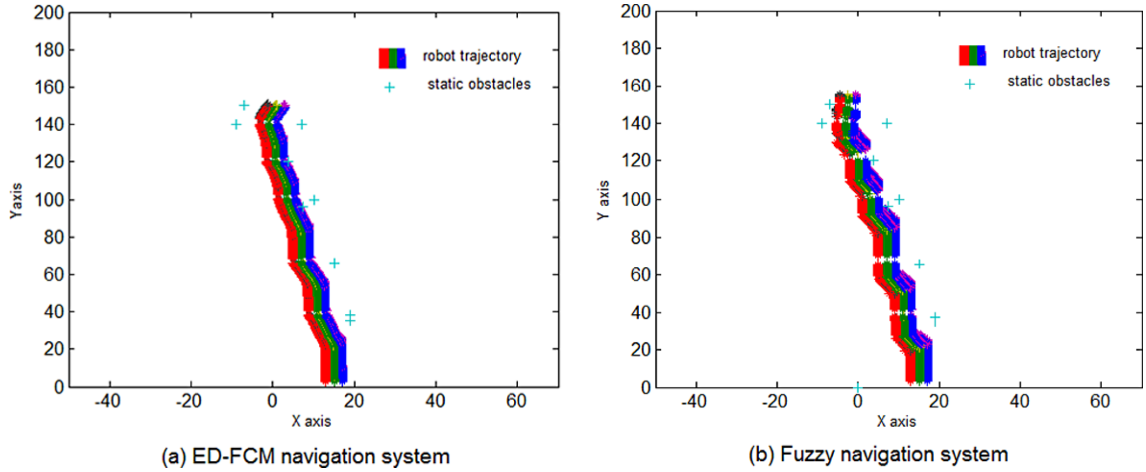


Fig. 8 Robot trajectories in the static scenario (axis scale is in cm)

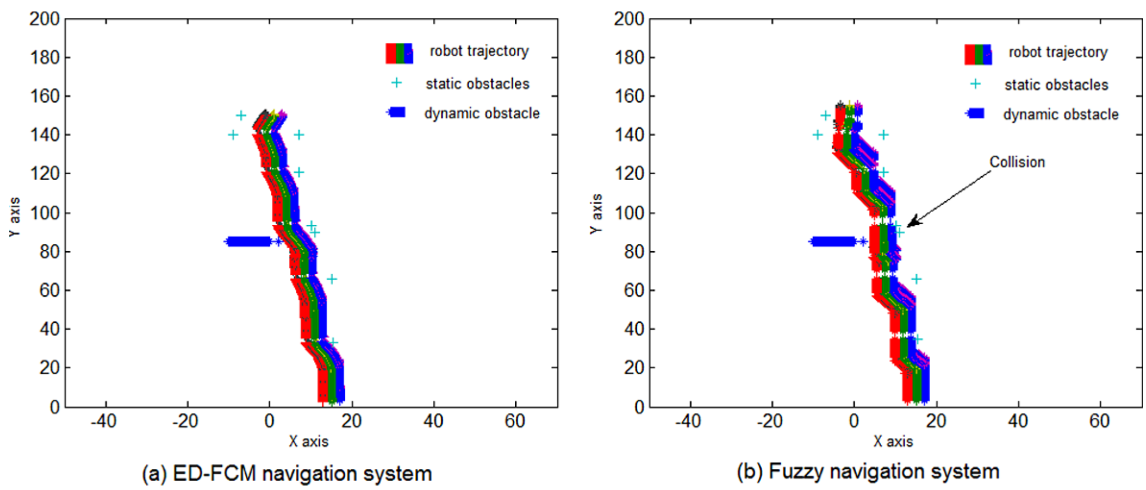


Fig. 9 Robot trajectories in the scenario with a dynamic object (axis scale is in cm)

in order to assess the autonomous navigation system's performance in a real scenario.

In the first experiment, a static unknown scenario is simulated. The goal of the experiment is to validate the reactive strategy of the proposed autonomous navigation system. The robot must cross this scenario from an initial point, corresponding to $(10, 0)$ coordinates in an x–y plan to an end point at $(0, 160)$ coordinates. During the crossing, the robot must detect the fixed obstacles (represented by the cyan plus sign in Fig. 8) and deviate from them, avoiding collision but without departing from the endpoint. The obtained results are presented in Fig. 8 for both FANS and ED-FCM navigation systems. These graphics show the dynamic trajectory (three-color trail) followed by the robot with both systems. The apparent flaws in the trajectory represent the speed-up, when sensors do not "see" an obstacle and the robot accelerates. In this simulated scenario, there is a critical situation around the position $y = 140$ cm. In this point, the robot must make the decision to move straight, pass between two obstacles and immediately turn left to avoid a frontal barrier and attain the target point. By analyzing the trajectories in Fig. 8, we note that the robot makes the correct decisions for both navigation systems and successfully attains the target point around $(0, 160)$. As expected, the developed trajectories in both systems are slightly different, specifically at the end of the path.

The second experiment simulates a dynamic environment in which some obstacles are fixed (cyan plus signs) and another is moving (the dark line represents the obstacle movement) as shown in Fig. 9. This dynamic object suddenly crosses the robot trajectory coming from the left side. This object evolves into a horizontal trajectory from position $(-15, 96)$ to position $(7, 96)$, beginning its movement when the robot is passing around the $(6, 88)$ position. The robot does not know the positions and/or movements of the obstacles. The obtained results are shown in Fig. 9. For both

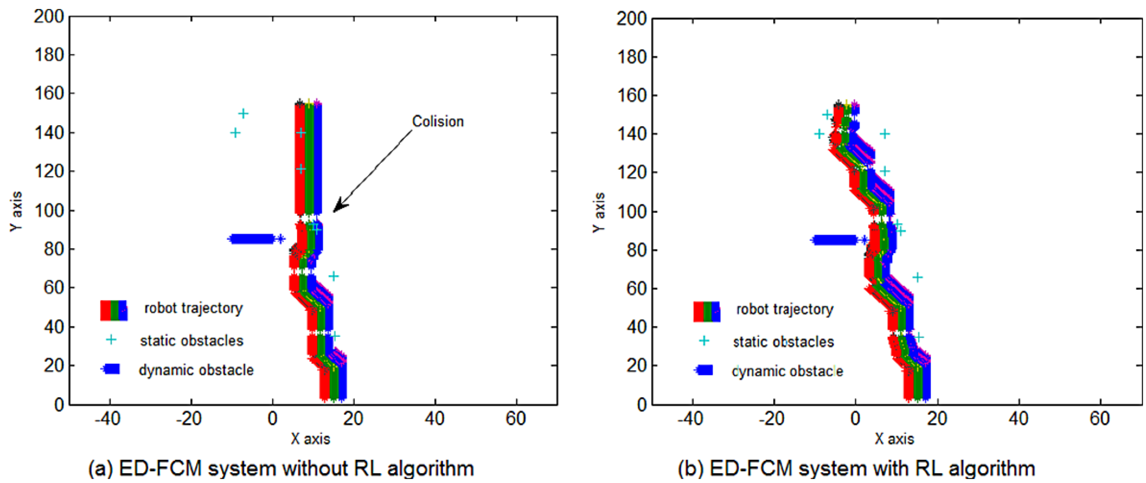


Fig. 10 Robot trajectories in the scenario with a dynamic object and high speed (axis scale is in cm)

navigation systems, the robot makes the correct decision to turn right in order to avoid the collision with the new object. By analyzing the trajectory in this figure, we note that the FANS system has more difficulty to avoid the next obstacle (Fig. 9b) and cannot prevent a slight collision with the obstacle placed in the position $(15, 98)$. With the ED-FCM navigation system, the robot does not collide (Fig. 9a) and it is able to maintain the motion trend of following a straight line in the direction of the target point (planned goal). Thus, in this simulated scenario, the ED-FCM system outperforms the FANS system.

Some other scenarios with different obstacle positions and event occurrences have been tested. Like these 2 illustrated scenarios, the robot always gets the final target position and makes successfully implemented collision avoidance maneuvers. These results validate the proposed navigation systems.

In conclusion, the first two experiments for both systems (Fuzzy and ED-FCM) are very similar with a minimal advantage to the ED-FCM navigation system. After the surprise appearance of an obstacle in the second experiment, the ED-FCM is more able to re-align its trajectory in order to attain the correct end point. This dynamic ability to softly adapt its behavior is due to the use of a reinforcement learning algorithm.

A third experiment is carried out in order to assess the importance of the learning algorithm in the performance of the proposed navigation system. In this experiment, the environment of the second experiment is reproduced, but the robot's speed is increased by approximately 20%, resulting in a difficult control situation for the navigation system. The ED-FCM navigation system is applied with and without the reinforcement learning algorithm. The obtained trajectories are shown in Fig. 10. The navigation system without the reinforcement learning algorithm proceeds with a sudden deviation when it detects the surprise obstacle (Fig. 10a). As a result, the robot cannot quickly implement the maneuvers needed to avoid the next obstacles and so it collides with them. After the collision the navigation system, as part of a security strategy, guides the robot in a straight line directly to the end point (Fig. 10a). This situation didn't occur when the navigation system used the reinforcement learning algorithm (Fig. 10b). In this case, the robot maneuvers to deviate from the surprise object are soft. Although its speed is high, the robot takes the appropriated actions to avoid the next obstacles and to reach the end point (Fig. 10b).

The reinforcement learning algorithm acts by constantly refining the values of ED-FCM weights as shown in Fig. 11. In this figure, the continuous soft lines show the evolution of the weights w_1, w_3 and w_5 that are respectively associated with the measurements of the right (RS), front (FS) and left sensors (LS). With the reinforcement learning algorithm, the values of w_1 and w_5 vary into the interval [0.6 to 1] defined by the expert in accordance with the response of the connected sensor. The value of w_3 associated with the front sensor varies into the interval [0.35 to 0.65] and the learning and discount factors are set to $\alpha = 0.01$ and $\gamma = 1$, respectively. The discount factor γ was set to one because we considered that the probability of the robot to go to the next state is already considered by the rules that compute the prediction of reward W_{lim} in (5). The choice of the learning factor affects the up-date of weights: if α is big (near 1) then sudden and zig-zag movements are allowed. On the contrary, if α is small (near 0) the ability to avoid an obstacle is decreased. The value of $\alpha = 0.01$ indicates that the maximal increment of a weight corresponds to one tenth of its maximum value. During simulation, this value is revealed as a good trade-off between keeping the movement or implementing maneuvers.

Finally the value of reward (punishment) is $r = 0.1$ ($r = -0.1$). Without the reinforcement learning algorithm, the

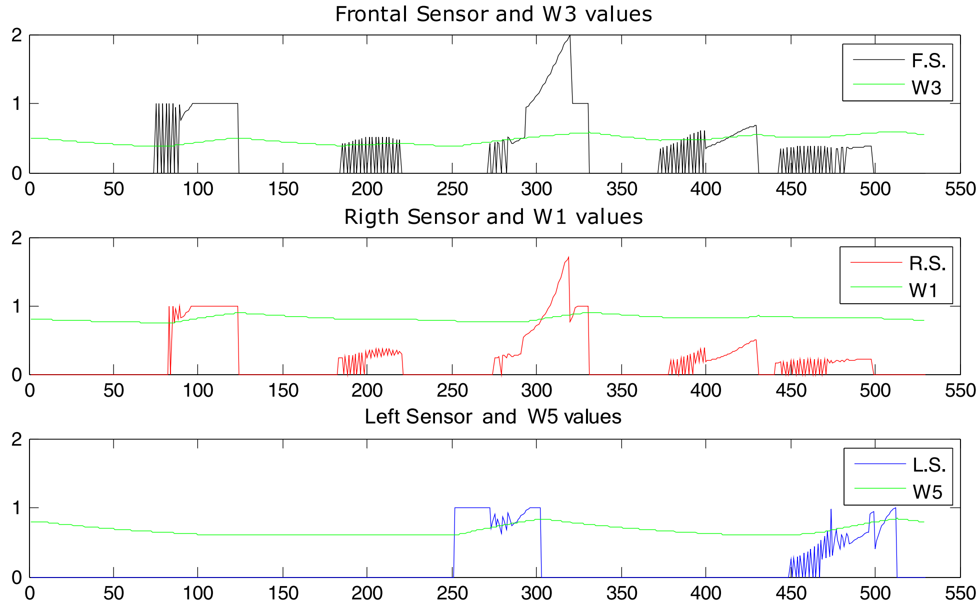


Fig. 11 Sensors measurements and weights computed by RL algorithm

Fig. 12 Fourth experiment: the robot approaches the obstacles

Fig. 13 Fourth experiment: the robot turns away from obstacles

weights w_1, w_3 and w_5 are fixed in the center of the corresponding intervals above.

The results in Figs. 10 and 11 indicate that the reinforcement learning algorithm correctly adjusts the causal relationship weights in accordance with the external stimulus perceived by the sensors. Thus, we can conclude that the use of a reinforcement learning algorithm in the reactive layer allows the online adaptation of the robots' behavior in the presence of some minor variations of robot parameters and/or environments.

The fourth experiment is carried out in order to validate the navigation system in a real didactic robot. The robot is Dr Robot X-80 [35]. The navigation system runs on a netbook Intel Atom N270, 1 GB. The robot's goal is to cross a room containing some boxes scattered on the floor. Specifically, the path's end point is placed among two boxes. Figures 12, 13 and 14 show three different moments during the crossing. Similar to the simulated scenario, the navigation system allows the robot to circumvent the obstacles and reach an end destination point without any prior knowledge

Fig. 14 Fourth experiment: the robot reaches the end of the path

of the scene. Unlike the simulated scenarios, there are many points of uncertainty in the real scenario. Some errors are due to sensor noise, odometers, wheels slipping and other physical components of the robot and the environment.

By analyzing the results of the fourth experiment, we can conclude that the proposed ED-FCM system is able to implement navigation tasks in different scenarios and even in the presence of unknown changes in the environment.

5 Conclusion

This paper developed an autonomous navigation system based on a new type of fuzzy cognitive map, named event-driven fuzzy cognitive map, ED-FCM. The developed ED-FCM approach adds new types of relationships and concepts into a classical FCM that allows for modeling of the human ability to make decisions in the presence of random events. The human knowledge is represented by a rule based system that is triggered when a critical situation occurs. As a result, the inference engine temporarily adds concepts and relationships into the FCM. Moreover, the use of a reinforcement learning algorithm operationally implements an on-line adjustment of knowledge supporting the close interaction of the FCM model with the real system it describes. This approach is a contribution to the intelligent system field. It is not restricted to navigation systems and can be applied to model intelligent systems needing to make decision on line.

An evaluation of the ED-FCM design and performance is carried out having as reference a fuzzy logic navigation system. From discussion in Sect. 3.3, we highlight the development of an ED-FCM system that is easier than the development of a fuzzy rule based inference system due to the intuitive nature of knowledge representation by means of cognitive maps rather than membership functions and if-then rules. From simulation results (Sect. 4), we can confirm that the performances of both systems are very similar with a minimal advantage to ED-FCM navigation system.

In accordance with the results presented in this paper, we can conclude that the proposed ED-FCM architecture constitutes a flexible and robust navigation system able to process vagueness and uncertainty in the environment as well as a fuzzy logic system. One of the main advantages of the proposed approach is that the knowledge acquisition and representation is simplified by the use of FCM models. As the computational complexity of a FCM is reduced if compared with a fuzzy inference system, the resulting fuzzy cognitive maps are also easy to implement and run. Thus, the developed navigation system is easily embedded in the hardware of a small robot.

Some future works include the implementation of additional functionality into the navigation system, such as a power management module vehicle tracking and decision making and refueling control, which targets collection and navigation speed. As an extension of the FCM modeling tool, we are also looking for a mechanism for FCM structure adaptation, by means of constructive algorithms for the creation or deletion of concepts, causal relationships and/or selection factor.

Acknowledgements The authors acknowledge financial support from the Brazilian Petroleum Agency (ANP/FINEP) grant PRH-ANP/MCT PRH10-UTFPR and Brazilian Research Council (CNPq) for grants 311877/2009-5 and 304037/2010-9.

References

1. Schraff RD (1994) Mechatronics and robotics for service applications. IEEE Robot Autom Mag 1(4):31–35
2. Mandow A, Gomes-de-Gabriel JM, Martinéz JL, Muñoz VF, Ollero A, García-Cerezo A (1996) The autonomous mobile robot AURORA for greenhouse operation. IEEE Robot Autom Mag 3(4):18–28
3. Chohra A, Farah A, Benmehrez C (1998) Neural navigation approach for intelligent autonomous vehicles (IAV) in partially structured environments. Appl Intell 8(3):219–233
4. Driankov D, Saffiotti A (2001) Fuzzy logic techniques for autonomous vehicle navigation. Physica-Verlag, Heidelberg
5. Pratihar DK, Deb K, Ghosh A (2002) Optimal path and gait generations simultaneously of a six-legged robot using a GA-fuzzy approach. Robot Auton Syst 41(2):1–20
6. Azouaoui O, Chohra A (2002) Soft computing based pattern classifiers for the obstacle avoidance behavior of intelligent autonomous vehicles (IAV). Appl Intell 16(3):249–272
7. Hui NB, Mahendar V, Pratihar DK (2006) Time-optimal, collision-free navigation of a car-like mobile robot using neuro-fuzzy approaches. Fuzzy Sets Syst 157(16):2171–2204
8. Xuefeng D, Hongmin Z, Yan S (2007) Autonomous navigation for wheeled mobile robots-a survey. In: 2nd international conference on innovative computing, information and control
9. Calvo R, Romero RAF (2006) A Hierarchical self-organizing controller for navigation of mobile robots. In: Proc of int joint conference on neural networks, IEEE World congress computational intelligence
10. Brooks RA (1986) A robust layered control system for a mobile robot. IEEE J Robot Autom 2(1):14–23
11. Murphy R (2000) Introduction to AI robotics. MIT Press, Cambridge
12. Arkin RC (1999) Behavior-based robotics. MIT Press, Cambridge
13. Connell J (1992) A hybrid architecture applied to robot navigation. In: Proc of the IEEE conf on robotics and automation, pp 2719–2724
14. Dickerson JA, Kosko B (1994) Virtual worlds as fuzzy cognitive maps. Presence 3:173–189
15. Stylios DC, Groumpos P (2000) Fuzzy cognitive maps in modeling supervisory control systems. J Intell Fuzzy Syst 8(2):83–98
16. Kosko B (1986) Fuzzy cognitive maps. Int J Man-Mach Stud 24:65–75
17. Axelrod R (1976) Structure of decision: the cognitive maps of political elites. Princeton University Press, Princeton
18. Sutton R, Barto A (1998) Reinforcement learning: an introduction. MIT Press, Cambridge
19. Min HQ, Hui JX, Lu Y-S, Jiang J (2006) Probability fuzzy cognitive map for decision-making in soccer robotics. In: Proc of the IEEE/WIC/ACM int conf on intelligent agent technology
20. Yeap WK, Wong CK, Schmidt J (2006) Initial experiments with a mobile robot on cognitive mapping. In: Proc of the int symp on practical cognitive agents and robots
21. Pipe AG (2000) An architecture for building "potential field" cognitive maps in mobile robot navigation. Adapt Behav 8(2):173–203
22. Gaskett C, Fletcher L, Zelinsky A (2000) Reinforcement learning for a vision based mobile robot. In: Proc of the IEEE/RSJ int conference on intelligent robots and systems

23. Smart WD, Kaelbling LP (2001) Reinforcement learning for robot control. In: Mobile robots XVI proc. SPIE, vol 4573
24. Zhu W, Levinson S (2001) Vision-based reinforcement learning for robot navigation. In: Proc int joint conference on neural networks
25. Yen G, Hickey T (2002) Reinforcement learning algorithms for robotic navigation in dynamic environments. In: Int joint conference on neural networks, vol 2, pp 1444–1449
26. Tunstel E, Jamshidi M (1994) Fuzzy logic and behavior control strategy for autonomous mobile robot mapping. In: Proc of IEEE int conference on fuzzy systems
27. Bhanu B, Leang P, Cowden C, Yin L, Patterson M (2001) Real-time robot learning. In: IEEE international conference on robotics and automation, pp 491–498
28. Macek K, Petrovic I, Peric N (2002) A reinforcement learning approach to obstacle avoidance of mobile robots. In: Int workshop on advanced motion control, pp 462–466
29. Li A, Li Z, Chen J (2011) Microassembly path planning using reinforcement learning for improving positioning accuracy of a 1 cm^3 omni-directional mobile microrobot. Appl Intell 34(2):211–225
30. Miao Y, Miao C, Tao X, Shen Z, Liu Z (2010) Transformation of cognitive maps. IEEE Trans Fuzzy Syst 18:114–124
31. Carvalho JP, Tomé JA (2000) Rule based fuzzy cognitive maps—qualitative systems dynamics. In: Proc 19th int conf North Amer fuzzy inf process soc, pp 407–411
32. Stylios DC, Georgopoulos VC, Malandraki GA, Chouliara S (2008) Fuzzy cognitive map architectures for medical decision support systems. Appl Soft Comput 8(3):1243–1251
33. Passino K, Yurkovich S (1998) Fuzzy control. Addison-Wesley, Reading
34. Gomide F, Pedrycz W (1998) An introduction to fuzzy sets: analysis and design. MIT Press, Cambridge
35. http://www.drrobit.com. Accessed November 11, 2010

Márcio Mendonça received the B.Sc. at Eletronics Engineering from Universidade de Lins (1993), the M.Sc. at Industrial Engineering from Universidade Estadual Paulista Júlio de Mesquita Filho (2003) and Ph.D. degree in Electrical Engineering from the Federal University of Technology—Paraná (2011) where He is currently a Lecturer. He has experience in Electrical Engineering with emphasis on Automation and Industrial Process, acting on the following subjects: computer vision and intelligent systems to autonomous navigation.

Lúcia Valéria Ramos de Arruda was born in Brazil, 1964. She received the Electrical Engineer degree from Federal University of Ceará, Brazil, in 1985, the M.Sc. degree from the Graduate School of Electrical Engineering, Campinas State University (FEE/UNICAMP), Brazil, in 1988, and the Ph.D. degree in Electrical Engineering from the University of Nice-Sophia Antipolis, France, in 1992.
Since 1995, she has been a Lecturer with the Federal University of Technology—Paraná, and in 2005, she was promoted to Associated Professor. Her research interests comprise soft computing methods to model and control of dynamic systems.

Flávio Neves Jr. was born in Pitanga-Pr, Brazil, 1963. He received the B.Sc. and M.Sc. Electrical Engineer degrees from Federal University of Technology—Paraná in 1987 and 1989, respectively and the Ph.D. degree in Electrical Engineering from the University Paul Sabatier (LAAS), France, in 1998.
Since 1992, he has been a Lecturer with the Federal University of Technology—Paraná, and in 2006, he was promoted to Associated Professor. His research interests comprise hardware and software to instrumentation and automation.