BOG and Fuzzy Controllers Based Multimodal Collision Avoidance for Industrial Manipulators

Fabrizio Romanelli
Control Engineering
Comau S.p.A., Robotics Business Line
Via Rivalta, 30, 10095 - Grugliasco (TO) - Italy
fabrizio.romanelli@comau.com

Fabio Tampalini
Control Engineering
Comau S.p.A., Robotics Business Line
Via Rivalta, 30, 10095 - Grugliasco (TO) - Italy
fabio.tampalini@comau.com

***Abstract*—In this paper a new approach, utilizing a new paradigm built on the fusion of both Bayesian occupancy grid (*BOG*) and fuzzy logic controller (*FLC*), is presented. The aim of this work is to integrate a probabilistic approach based on the Bayesian Occupancy grids together with the fuzzy logical-based approach. The advantages of this method are several: first of all we can model the behaviours of the obstacles, instant by instant, with a probabilistic model, even if detected from different families of sensors, in order to achieve sensor fusion and robustness to uncertainty of data. On the other hand, the fuzzy logic control helps the algorithm to converge faster to the optimum value of speed override when the obstacle is distant enough, taking also into account the position of the obstacles with respect to the heading of the robot Tool Center Point (*TCP*). Furthermore the FLC has behavioural features, in the sense that it takes into account the behaviours of obstacles: this makes the control system more accurate. Another important aspect of the presented method is that it merges BOG environmental approach to FLC robot-centric one. This gives to the system a more complete vision, since from one side BOG has an "external" view of the scene (utilizing a map of the working area) and the FLC has an "internal" view (utilizing a robot-centric framework). Simulation results show that the robot override speed is adapted constantly to avoid collision with the obstacles, adapting its behaviours to the level of available knowledge with a smooth control law which merges the stochastic approach to the deterministic one. Extensions to control of acceleration of the TCP, integration of the algorithm with the management of robot restricted areas and to the prediction of the trajectories of obstacles are proposed as future development.**

I. Introduction

Occupancy grids [1], [2] are a tool of widespread use in robotics. They are used in order to tessellate the space (i.e. the operating area of the robot) in a regular cells, and to store in each cell a fine grained, quantitative information. With Bayesian occupancy grids [3], the idea is to extend the meaning of the value contained in each cell to the probability of that cell being "occupied" by an object. The nature of the decomposed space may be Euclidean space or a higher dimension state-space which could take into account velocities, accelerations, orientations, etc.

Such maps are extremely useful for robotic applications, such as obstacle avoidance or collision avoidance. In this kind of applications, the problem of the uncertainty of the information given by the sensors (proprioceptive or exteroceptive) is one of the biggest in this field. Such a paradigm, utilizing the Bayesian occupancy grids, face the problem in an very efficient way, as it models the unreliability of the measurements with probability. Another advantage of the use of occupancy grids is that they allow sensor fusion to be performed in a flexible way even if the system presents different typologies of sensor (even with very heterogeneous sensor models).

Fuzzy logic [4], [5], [6] is widely used in several sectors: from control systems to database, modeling to computer programming in order to realize reliable and desirable products. Of particular interest are controls based on fuzzy logic (FLCs) which are used into Antilock Braking Systems (ABS) [7], in camera applications and where robots or automatic systems have to carry out behavioural tasks, such as collision avoidance and path planning [8].

Our system is based on the decisional and behavioural component on a fully reactive system based on FLCs. The information about obstacle position around the working area (related to the pose of robot TCP), are computed in order to establish which kind of behaviour has to be taken and how. It is therefore possible, adjusting the control properly, to synthesize a system capable of acting with complex strategies, based on a simple set of behaviours such as `decelerate, accelerate, stop, turn,...`. The result of this paradigm is a control, which expresses precisely qualitative concepts, defined formally, in terms of mathematical functions between functions (membership functions) [9].

In this paper a new approach to deal with collision avoidance is proposed. In the industry sphere the problem of collision and obstacle avoidance is relevant as the interactions between humans and machines are closer and closer. This is an important aspect which is matter of studies in the field of robotics and automation. In this context the basis idea of this work is to give a first step towards integration between the work of humans and robots; this integration can't be set aside of security which is the most relevant aspect of the problem. This work takes into account this aspect as the first requirement.

The obstacle avoidance paradigm proposed on this paper is based both on a probabilistic framework, such to make the connection between the sensorial perception and the control of the robot, and on a polyvalent logic framework. There are no particular restrictions to the exteroceptive sensorial input model to the system, as the uncertainty of position of the obstacles given from the sensors is modeled in a probabilistic way, and different typologies of sensor can coexist in the same system, as the probabilistic framework also gives a good instrument to obtain sensor fusion. We combined this method, which is efficient for medium-low distances obstacles, with a fuzzy logic engine, which is very efficient for medium-high distances obstacles and it's well adaptable to define politics to decide the reference override speed in function of heading. The advantages of utilizing a combination of the two approaches is that we can control the robot's override speed, acting with both the controls in a continuous and smooth way. This control law takes into account both the trajectory of the obstacles moving around the robot area and the behaviour of the obstacles. This means that if for example there is an obstacle closer to the robot's TCP, this will have more probability to collide than the others objects moving around, the second object closer to the robot's TCP will have less probability, and so on up to the last objects.

Simulation results show how the developed algorithm face the problem of collision avoidance in a robust way, even with several obstacles moving around it apart from the typology of trajectory of the obstacles.

The paper is structured as follows: in Section II the Bayesian occupancy grid and fuzzy logic frameworks are described: in Section III the developed solution is described in detail. In Section IV simulation results are shown and discussed. Section V closes the document with remarks and purposes for future activities.

II. Related works

The developed method combines two existing frameworks: the Bayesian occupancy grid [10], [11], [12] and the fuzzy logic filter [13], [14], [15]. The Bayesian occupancy grid is a tessellated 2D grid in which each cell stores the probability of occupation. Sensor observations are processed from both the BOG algorithm and the fuzzy filter and the results of the computation are given as input to the collision avoidance algorithm.

A. Bayesian occupancy grid

The occupancy grid is based on the division of space (Cartesian or multidimensional) into cells. The probabilistic approach applied to the occupancy grid paradigm gives the possibility to extend the concept of cell value: if applied to obstacle/collision avoidance this value can fit well with the probability that the cell is occupied by an obstacle. Given as input for the algorithm the position $X = [x, y]^T$ of each obstacle, or likewise (ρ, θ), Bayes' theorem states:

$$P_c(Occ|X) \propto P_c(X|Occ) \cdot \hat{P}(Occ) \tag{1}$$

Where $P_c(Occ|X)$ is the probability that the cell is occupied by an obstacle, given the measurement, and the right side member of (1) is a distribution of probability, and it is shaped as a Gaussian multimodal distribution as shown in (2):

$$P_c(X|Occ) \cdot \hat{P}(Occ) \propto \mathcal{N}(\mu, \Sigma) \tag{2}$$

Where $\mu = [\mu_1, \ldots, \mu_N]^T$ and Σ is the covariance matrix (positive-definite real $N \times N$ matrix). The probability density function is:

$$\begin{aligned} f_X(x_1, \ldots, x_N) \;=\; & \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} \cdot \\ & \cdot \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right) \end{aligned} \tag{3}$$

The formula in (3) describes the probability density of an obstacle, in each point of the space $\mathbb{R}^N$ (where $N = 2$).

In order to extend the Bayes' theorem to more than one obstacle, assuming that all the events (obstacles) are independents, we can use the generalized union probability theorem:

$$\begin{aligned} P\left(\bigcup_{i=1}^{n} A_i\right) \;=\; & \sum_i P(A_i) - \sum_{ij}{}' P(A_i \cap A_j) + \\ & + \sum_{ijk}{}'' P(A_i \cap A_j \cap A_k) - \ldots + \\ & + (-1)^{n-1} P\left(\bigcap_{i=1}^{n} A_i\right) \end{aligned} \tag{4}$$

This theorem states that if the probability that an obstacle is occupying a cell is independent from the others (which is reasonable for the problem), we can express the union probability in a closed form. Under this point of view, this method is a good approach to the obstacle avoidance since, besides the possibility to solve the problem of modeling multi-object space occupancy, it also faces the problem of sensor fusion, as the structure of Bayesian occupancy grid is well suited for the integration of different typologies of sensor measurements. The algorithm is structured as follows:

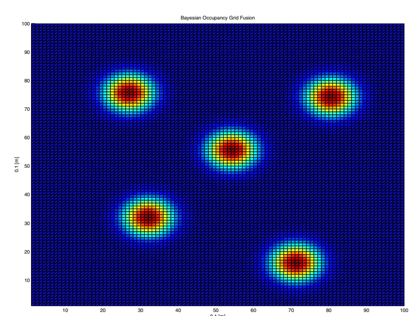

Fig. 1. Bayesian Occupancy Grid: red is high, blue is low probability of occupation.

1) At the beginning, the occupancy grid is initialized with a 0.5 probability of occupation;
2) As a new measurement is available, the grid is updated following the Bayes' rule described in (1);
3) The grid is further updated using the generalized union probability theorem, in order to merge together all the obstacles in the robot area;
4) Back to step 2.

For further details, the interested reader may refer to [16]. Fig. 1 shows an example taken from a simulation of a Bayesian occupancy grid, where 5 obstacles are moving around the robot area.

B. Fuzzy logic

A pure reactive system could be a good solution for "Collision Avoidance" task, requires few computational resources. Other advantages of purely reactive systems are:

- emphasis on the importance of a tight relationship between perception and action;
- vertical decomposition of the problem into subproblem to be executed in parallel;
- modularity of the software.

Following the classis outline [17], we can say that FLC has two components: a functional module and a behavioural module. The functional component (Fig. 2) acquires information to

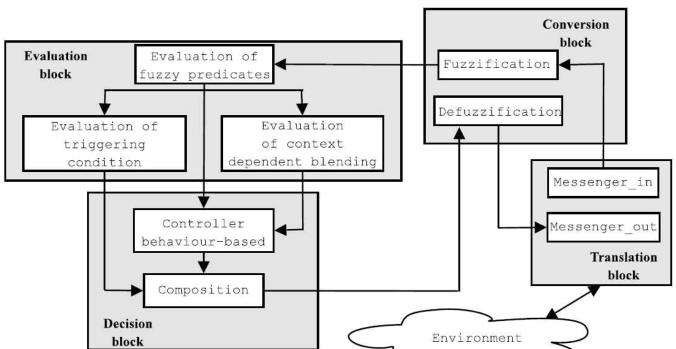

Fig. 2. Functional view.

be used as input to the engine. As these data have been computed, the functional component blends resulting actions and transmits values to actuators. the functional component is composed of:

- translation block: it is an interface between information about the environment surrounding the robot and data processed inside the engine;
- conversion layer: information acquired from translation layer are here transformed into fuzzy values (fuzzification) and, after computation, output fuzzy values are re-transformed into crisp values (defuzzification);
- calculus layer: it is composed by three sub-modules, each one managing a sub-tree: predicates sub-tree, behaviour triggering conditions sub-tree and behaviour evaluation sub-tree;
- decision layer: decides actions to be carried out on the basis of environment information that are provided by previous layer.

The information about obstacles distance and heading should be generated by the BOG subsystem; fuzzy rules are the basis where the operative knowledge of the robot can be built from a human heuristic knowledge. A Fuzzy rules template is the following:

```
IF <antecedent> THEN <conseguent>
```

where the antecedent could consist of an arbitrary large number of precondition combined through logic operators OR, AND and NOT; for example:

$$\texttt{IF ((obstacle} \notin \texttt{North) AND (obstacle} \in \texttt{Far))}$$
$$\texttt{THEN (speed} \in \texttt{Fast)}$$

In natural language this fuzzy rule states that if an obstacle is Far and not on the $North$ of the robot, than the robot must advance $Fast$. In other words, Far, $North$ and $Fast$ are the membership functions of different sets (respectively distance, heading and speed) and the antecedents and consequents represent the credibility values of membership degrees. Obviously, while in the antecedent all the aforementioned logic operation can be used, in the consequent only the AND operator is acceptable. Moreover, credibility value (i.e. the membership degree of a variable to the membership function) range between "0" and "1". The fuzzification, blending and defuzzification blocks in the functional engine scheme are depicted in Fig. 3. The effective engine component is the

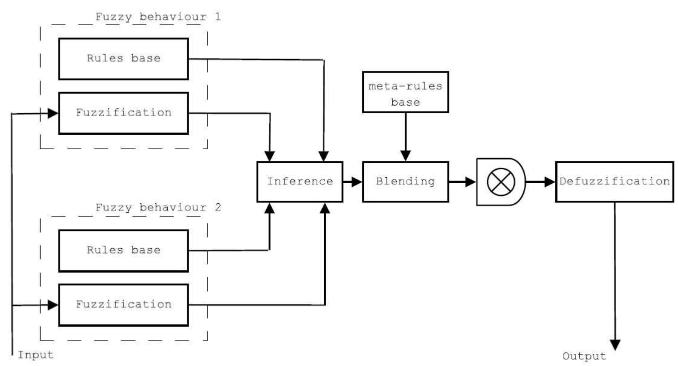

Fig. 3. Fuzzy inferential engine scheme.

one labeled `Inference`. This is the scheme we have chosen for FLC component, in which the block that evaluates the triggering condition is scanned before the behavioural sub-tree (in order to avoid wasting computational resources) and there is a blending block for each behaviour. The purpose of activation threshold is to state the effective possibility that the robot behaviour will not change. In this way the computational load is decreased because the engine is not forced to scan all the rules. Another important component is the blending block that fuses the outputs of the basic behaviours: this block allows the coexistence of behaviours even if there are conflicting tasks to be performed. Blending rules are contained in the component labeled *meta-rules base* (Fig. 3), the interested reader can refer to [18].

III. The Proposed Integrated Solution

In this section we explain in detail the developed algorithm and the proposed solution. The Bayesian occupancy grid explained in Section (II-A) is a probabilistic method that models the space occupied by obstacles on an environment. The input to the occupancy grid, as stated in the previous Section, is the position of each object (given either as a Cartesian position or as a ρ, θ representation).

The grid is relative to the space around the robot, as we are modeling an algorithm for an industrial manipulator which is always in a fixed position. The grid is then mapped on the real working area of the robot, and it is possible to choose a different resolution of thickness of the grid, in order to achieve more accuracy on the possibility to have an obstacle in a cell. As we suppose that the obstacles are humans (i.e. operators that can interact with the manipulator around its working area) and that they are seen as input from cameras (as positions on a plane) or from laser scanners (as distances and angles) to the BOG algorithm, we have to extend the 2D occupancy grid to a 3D Bayesian occupancy grid, since the robot TCP is given each instant as a position and orientation in the space. In order to extend the 2D Bayesian occupancy grid to 3D, we assume that each obstacle can be modeled as a cylinder of probabilities, where the center is given by the mean value of the Gaussian distribution of the obstacle.

At the same time, the BOG override speed is computed by the algorithm. This is quite simple, as the BOG gives a probability framework which is well suited for the problem of controlling override speed of the robot TCP.

$$O_{BOG} = 1 - P_c(Occ|X) \cdot K_s \tag{5}$$

The BOG override speed O_{BOG} is computed as the inverse of the probability that a cell is occupied given the measurement, multiplied by a constant K_s, which is the weight given to the probability that an obstacle is occupying the cell. The algorithm also gives as input for the FLC, the distance and angle of the nearest obstacle to the robot.

The FLC takes the distance and angle computed from the BOG algorithm and, taking into account the behaviours of the obstacles, computes the override speed according to the defuzzification process. The O_{FLC} is computed taking advantage of the behavioural nature of fuzzy logic filter. This override speed is then used by the control in order to compute the override value which has to be assigned to the robot. This value is computed considering the general override at time $t-1$, which is the value we use to weight the BOG and FLC overrides.

$$\begin{aligned} O_{gen}(t) &= O_{gen}(t-1) \cdot O_{FLC}(t) + \\ &+ (1 - O_{gen}(t-1)) \cdot O_{BOG}(t) \end{aligned} \tag{6}$$

The meaning of utilizing the past general override $O_{gen}(t-1)$ as shown in (6) as a weight is that we try to give more importance to the FLC algorithm when the general override speed is high (i.e. the obstacle is far), and more importance to the BOG algorithm when the general override speed is low (i.e. the obstacle is near). This typology of control has two main behaviours: when the obstacle is far, and the general speed override is high, the FLC acts as the main control, since it allows a behaviour based on the heading of the robot TCP towards the obstacle. In this state the decisional policy is submitted to the FLC as the distance from the obstacle is high: for example while the TCP is moving and an obstacle is moving away, behind the heading of the TCP, the FLC override speed will be fixed to the maximum speed, if there is no obstacle in front of the TCP.

On the other hand when the obstacle is near, and the general speed override is low, the BOG algorithm acts as the main control, as it allows a behaviour based on the probability of encountering an obstacle in a portion of space, and adjusting the BOG override with an appropriate control law. In this state the decisional policy is submitted to the BOG algorithm as the distance from the obstacle is low, and a good accuracy is needed by the control constraints.

It is important to emphasize the way the control behaves, as the switch between the two typologies of control techniques is entirely smooth. This means that the BOG and FLC controls act simultaneously in every condition; the advantages of this technique is that it fuses together the advantages of each control law and that, taking advantage of feedback on general speed override, the control is fast, robust and ready. A system scheme is depicted in Figure 4.

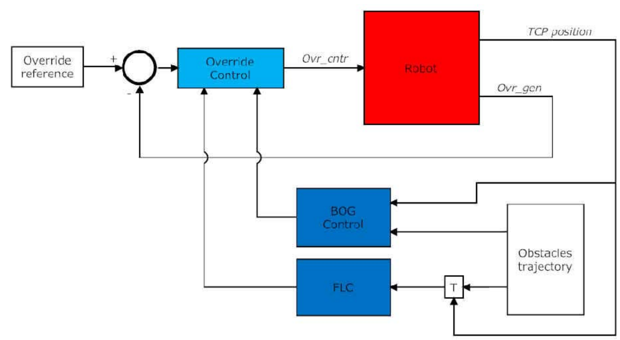

Fig. 4. System layout: robot is controlled by the feedback controller from the output (general speed override).

The proposed algorithm is structured as follows:

1) At the beginning, BOG and FLC variables are initialized;
2) As a new measurement is available, the occupancy grid is updated;
3) The grid is further updated using the generalized union probability theorem, in order to merge together all the obstacles in the robot area;
4) The BOG override, the distance and angle of the nearest obstacle is then computed and passed to the FLC;
5) The FLC computes the speed override according to the defuzzification process;
6) The BOG and FLC partial speed override are used to compute the general speed override as in (6);
7) Back to step 2.

IV. Simulation results

In this section we present some results from the simulation framework. The experimental tests are divided into three sessions:

1) One obstacle;
2) three obstacles;
3) five obstacles;

moving around the robot working area. The control system produces a controlled speed override value in order to control robot movements, taking into account the real override of the robot as a feedback for the control system. The TCP and obstacles are represented as points in space and the relative trajectories are defined a priori inside the simulator. The obstacles speed are constant, while TCP speed value is modified by the feedback control system output (override controlled speed).

In order to describe in detail our system, we produced a general overview of the computed outputs, depicted in Figure 5 for the case of five obstacles moving around the robot area.

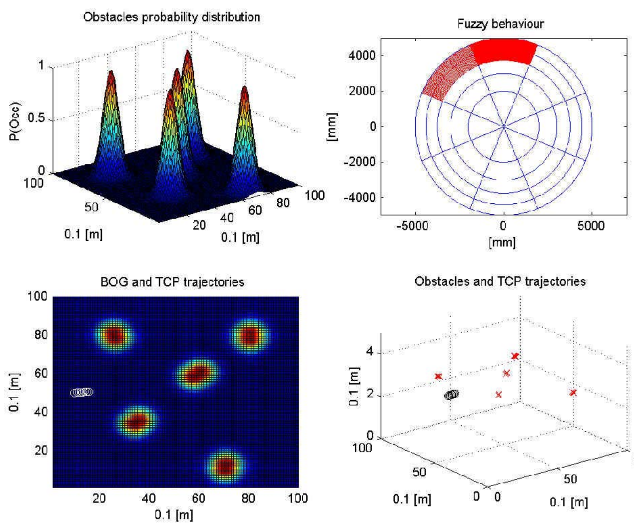

Fig. 5. General overview of simulation results for five obstacles moving around the robot working area.

The Figure shows the distribution of probabilities of the obstacles produced by the BOG filter in the upper left side. In the bottom left side the trajectories of the TCP (in white circles) in 2D representation of the BOG are depicted. The FLC is also showed in the upper right side, with the behaviour related to the closest obstacle. In the last subplot it is depicted the obstacles and TCP 3D trajectories.

The Figure 6 shows the three simulation test-beds. The first one is related to the presence of one obstacle in the robot working area. As we can see from the picture the general override follows the trend of the FLC override, since in this control technique the FLC is more sensible to the far obstacles (as stated in Section III), and the BOG algorithm does not influence the general behaviour of the control paradigm because the obstacles are too far from the robot. The second one is related to the presence of three obstacles. In this case, the BOG filter reacts after about six seconds with the presence of obstacles near to the TCP as expected; the FLC has a general behaviour and acts constantly during all the experiment. In the last subplot the behaviour of control system is depicted for five obstacles. The trends shows clearly the general behaviour of our system. The FLC reacts taking into account all the obstacles in the robot environment, and this can be seen

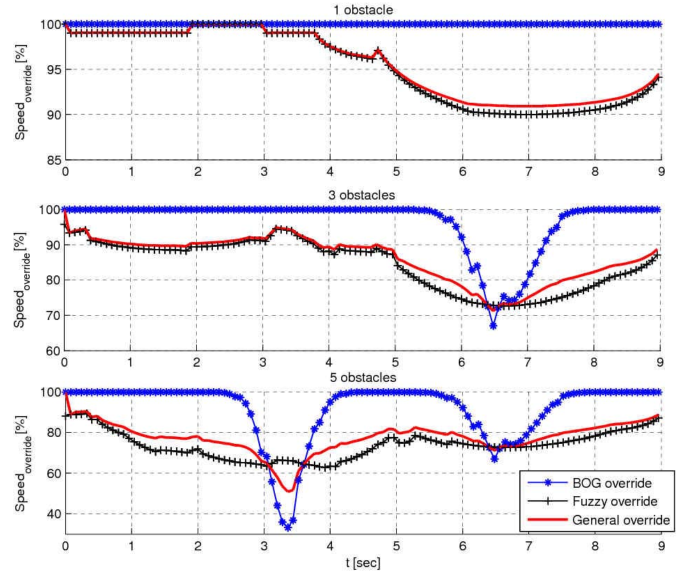

Fig. 6. Override representation for different setups computed by: BOG filter, FLC and controlled override.

considering the slow dynamics, drawn in the subplot. On the other hand BOG filter is more sensible to the close obstacles and influences the general override in a significant way (when general override is low).

The general behaviour produces a control override which is sensible to the environmental dynamical configuration, in a convenient way, as the first plot (one obstacle) shows a general mean override of about 95% while the others shows lower and lower general control override values.

V. Conclusions

We presented a new algorithm, that utilizes an innovative control technique in order to face the problem of collision avoidance. The new paradigm is based on both the Bayesian occupancy grid and fuzzy logic controller. With the presented work we focused our attention on the integration of a probabilistic approach, based on the Bayesian occupancy grids, and a fuzzy logical-based approach.

In this paper we showed several advantages given by the presented method. On the one hand we can model the behaviours of the obstacles, instant by instant, with a probabilistic model, even if detected from different families of sensors. On the other hand, the fuzzy logic control helps the algorithm to converge faster and to take into account obstacle behaviours as well. The combination of the two paradigms gives the benefits of each approach. Furthermore the control is structured as a feedback system from the output (i.e. general speed override): this gives the system a quicker response and makes it more ready. Another important aspect of the presented method is that it merges BOG environmental approach to FLC robot-centric one. This gives to the system a more complete vision, since from one side BOG has an "external" view of the scene (utilizing a map of the working area) and the FLC has an "internal" view (utilizing a robot-centric framework).

Simulation results show that the robot override speed is adapted constantly to avoid collision with the obstacles while adapting its behaviours to the level of available knowledge with a smooth control law. The present system will be soon implemented into an industrial manipulator with a hard real time system in order to confirm the effective feasibility.

The aim of this work is to provide the robotized cells with more and more safety, as the interaction between robots and humans is becoming significant: the support of a dynamic collision avoidance algorithm provides a rough artificial intelligence to the robotized cell which gives the basis to a stronger and safer cooperation between operators and machines.

VI. FUTURE WORK

In order to achieve a better performance and to improve the accuracy of the algorithm on the one hand, we are studying solutions to extend BOG to BOF (Bayesian Occupancy Filters), as suggested in [11] and to adapt this methodology to our system. With this innovative technique the concept of 2-dimensional occupancy grid is extended to a 4-dimensional probabilistic occupancy grid, in order to take into account velocity besides space. This approach generates a dynamic occupancy grid, and considering obstacles velocity allows to reach more accuracy, a quicker response and allows to consider new behaviours for the movement of the obstacles. Another matter of study is about utilizing neural networks (self organizing networks as in [12]) in order to extract objects from the grid and to synthesize a system capable of learning obstacles trajectories (similarly to the one described in [19]) and to use this knowledge in order to improve the readiness of the presented system and to provide the system with a prediction engine for objects which are able to decide their trajectory on the basis of decision process (e.g. humans and robots).

On the FLC side, the future developments are the integration into the system of more behaviours (such as speed, acceleration and the termination point of a move). At the same time it is possible to define obstacles ID in order to classify them as more or less important; this will give the system more behaviours and therefore a better accuracy.

Another improvement to the system is the integration of a trajectory planner, to avoid obstacles and to replan the trajectory in order to continue the task without stopping the motion of the robot.

ACKNOWLEDGMENT

This work has been supported by Comau Robotics S.p.A. and it was partially supported by SME Robot. The authors would also like to thank Rosario Cassano (Control Engineering main chief) and Maela Bertone (Control Engineering) for their help on the experimental work and on the state of the art research.

REFERENCES

[1] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Magazine*, vol. 9, no. 2, pp. 61–74, July August 1988.

[2] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, pp. 46–57, June 1989.

[3] T. Collins, J. Collins, and C. Ryan, "Occupancy grid mapping: an empirical evaluation," in *Proc. of the 15th Mediterranean Conference on Control and Automation*, Athens - Greece, July 2007.

[4] L. Zadeh, "Fuzzy sets," vol. 8, pp. 338–353, 1965.

[5] G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Upper Saddle River, NJ: Prentice Hall, 1995.

[6] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," in *Readings in Fuzzy Sets for Intelligent Systems*, D. Dubois, H. Prade, and R. R. Yager, Eds. San Mateo, CA: Kaufmann, 1993, pp. 283–289.

[7] G. F. Mauer, "A fuzzy logic controller for an abs braking system," *IEEE-FS*, vol. 3, pp. 381–388, Nov. 1995.

[8] B. Kim, O. Kwon, K. Kim, E. Lee, and S. Hong, "A study on path planning for mobile robot based on fuzzy logiccontroller," Cheju Island, South Korea, pp. 1002–1005, Dec 1999.

[9] A. Bonarini, "Sistemi fuzzy (italian)," *Mondo Digitale (AICA)*, vol. anno II, no. 1, pp. 3–14, Mar. 2003.

[10] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings of the IEEE International Conference on Robotics and Automation*. St. Louis, Missouri: IEEE Computer Society Press, 1985, pp. 116–121.

[11] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grid," Rome, April 2007. [Online]. Available: http://emotion.inrialpes.fr/bibemotion/2007/FSL07

[12] D. Vasquez, F. Romanelli, T. Fraichard, and C. Laugier, "Fast object extraction from bayesian occupancy grids using self organizing networks," Singapore (SG), December 2006. [Online]. Available: http://emotion.inrialpes.fr/bibemotion/2006/VRFL06

[13] T. Dong, X. Liao, R. Zhang, Z. Sun, and Y. Song, "Path tracking and obstacles avoidance of uavs - fuzzy logic approach," in *Fuzzy Systems, 2005. FUZZ '05. The 14th IEEE International Conference on*, May 2005, pp. 43–48, iSBN: 0-7803-9159-4.

[14] J. Yen and N. Pfluger, "A fuzzy logic based extension to payton and rosenblatt's command fusion method for mobile robot navigation," in *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 25, no. 6, 1995, pp. 971–978.

[15] R. Malhotra and A. Sarkar, "Development of a fuzzy logic based mobile robot for dynamic obstacle avoidance and goal acquisition in an unstructured environment," in *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*, 2005, pp. 1198–1203, iSBN: 0-7803-9047-4.

[16] P. Stepan, M. Kulich, and L. Preucil, "Robust data fusion with occupancy grid," in *Systems, Man and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 35, Feb 2005, pp. 160–115, iSSN: 1094-6977.

[17] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, March 1986. [Online]. Available: http://jmvidal.cse.sc.edu/library/brooks86a.pdf

[18] E. H. Ruspini, "On the semantics of fuzzy logic," *International Journal of Approximate Reasoning*, vol. 5, no. 1, pp. 45–88, January 1991.

[19] D. Vasquez, T. Fraichard, O. Aycard, and C. Laugier, "Intentional motion on-line learning and prediction," Port Douglas (AU), July 2005. [Online]. Available: http://emotion.inrialpes.fr/bibemotion/2005/VFAL05