Design of Collision Avoidance System for a Chicken Robot Based on Fuzzy Relation Equations

Alexey Gribovskiy *Student Member, IEEE* and Francesco Mondada, *Member, IEEE*

***Abstract*— Design and study of mixed animal-robot societies are the fields of scientific exploration that can bring new opportunities for research into the group behavior of social animals. Our goal is to develop a Chicken Robot – an autonomous mobile robot, socially acceptable by a group of chicks and able to interact with them using appropriate communication channels. One of the basic requirements to such a robot is the safety of its motion with respect to the chicks, so it has to be endowed with an efficient real-time collision avoidance system. In this paper we present a fuzzy obstacle avoidance system that was designed for the Chicken Robot using the theory of fuzzy relation equations. This approach allows to easily check a consistency of a used rule base and provides a more systematic approach to design of fuzzy control systems comparing with the classical techniques. The experimental results demonstrate that a mobile robot equipped with the presented system is able to successfully avoid obstacles and safely navigate on an experimental arena.**

I. Introduction

The goal of the Chicken Robot project is to develop a mobile robot able to interact with a group of domestic chicks using communication channels relevant for them, and to control specific group behaviors. A mobile robot capable to reply to an animal and to adapt to an animal's behavior offers a promising opportunity to ethology. It can be used in purely scientific experiments to study animal social behavior as well as in farming applications to improve breeding conditions of intensively produced animals.

Design of such a robot and its control system is also an interesting challenge for robotics. There are very few projects in the animal-robot interaction field [1]–[5]. In our research we follow a general methodology developed within the European project Leurre, where a mixed society of cockroaches and robots was created and successfully controlled [5]. We apply and modify this methodology for a case of more complex social animals such as chickens.

In order to overcome the problem of social recognition and acceptance of the robot by chicks we use the mechanism of the filial imprinting [6]. Once the robot is presented to very young chicks, they consider it as their hen and then will follow it, forming a stable group.

The young chicks have fragile bodies, hence one of the main requirements to the Chicken Robot is the safety of its displacements with respect to chicks, i.e. a collision risk must be low and the eventual effects of collision be harmless.

A number of reactive navigation systems have been implemented in robotics. Among them the *potential field* method, where a robot follows to a superposed force field from obstacles and the goal [7]; the *dynamic window approach* that takes into account dynamics of a robot and deals with constraints imposed by robot's limited velocities and accelerations [8]; and the *vector field histogram* method [9], where a motion command is deduced from the polar histogram of obstacles density constructed from the current sensor readings. Finally, a number of mobile navigation approaches based on the principles of fuzzy control were proposed [10]–[12].

In this work we also take a fuzzy control based approach to the task of collision avoidance. In comparison with other techniques the fuzzy control has such advantages as use of expert's heuristic knowledge in a natural linguistic form and ability to provide a satisfactory performance in case of noisy input. This makes the fuzzy control a good choice for the needs of autonomous robotics, where a mathematical model of the environment is often not available, sensory data are noisy and real-time operation is necessary [11].

The conventional way to construct fuzzy control systems is to determine the relationship between input and output sets in form of fuzzy rules and then to use the resulted rule base directly to compute the output of the system. Being in the common use, this approach by default assumes that all of the rules create a consistent control algorithm that is not always the case; design of an efficient control system demands an intense tuning of the system parameters via trial-and-error adjustment. An alternative way is to represent the knowledge about the mapping of the input space to the output space contained in the rule base in the form of a single fuzzy relation by applying the *fuzzy relation equations* theory. Initially proposed by Sanchez [13] in 1976 these equations have been intensively studied since then [14]–[21] and found their applications in different fields [22]–[24]. Use of fuzzy relation equations allows to easily check the consistency of the rule base and provides a more systematic approach to design of the fuzzy control system comparing with other approaches [22].

In this paper we present a fuzzy collision avoidance system designed for the Chicken Robot using the theory of fuzzy relation equations. We represent a relationship between an input set of robot's sensor values and an output set of steering directions in the form of a fuzzy rule base. Then we prove the consistency of the rule base and find a mathematical

Alexey Gribovskiy is with the Laboratoire de Systèmes Robotiques (LSRO), Ecole Polytechnique Fédérale de Lausanne, EPFL STI IMT LSRO1, Station 9, CH-1015 Lausanne, Switzerland (phone: +41 216937730; email: alexey.gribovskiy@epfl.ch).

Francesco Mondada is with the Laboratoire de Systèmes Robotiques (LSRO), Ecole Polytechnique Fédérale de Lausanne, EPFL STI IMT LSRO1, Station 9, CH-1015 Lausanne, Switzerland (phone: +41 216937357; email: francesco.mondada@epfl.ch).

representation of the knowledge that it contains in the form of a fuzzy relation R by solving a corresponding fuzzy relation equation. The fuzzy relation R is further used to compute a robot's steering angle allowing to avoid detected obstacles. To validate efficiency of the proposed obstacle avoidance system we conduct experiments with a real robot.

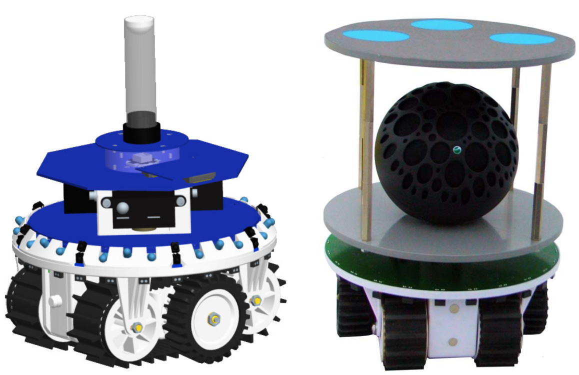
The paper is organized as follows. Section II presents an overview of the Chicken Robot and its control system. Section III explains how the obstacle avoidance system works and discusses efficiency issues, followed by the experimental results in Section IV.

II. The Chicken Robot and its control system

A. The Chicken Robot

In animal-robot experiments we use a modification of a marXbot robot (Modular All-teRrain eXperimentation roBOT) developed in our group (Figure 1a). It is a modular mobile robot that can be expanded with different extensions supporting specific mechanical or electronic features. A base module of the marXbot (Figure 2) guarantees a stable contact with the ground and a remarkable maneuverability. It consists of two treels (tracks plus wheels) with two motors; 24 infrared proximity sensors around the robot used as bumpers; twelve infrared ground and close ground sensors; three dsPIC33 microcontrollers, one for each motor and one to manage the sensors; and space for a battery pack. The infrared bumpers have a distance range about 3 cm.

We extended the base module by adding a speaker module with a Sony Ericsson MBS-100 Bluetooth speaker used to improve quality of the imprinting and a top board with color markers (Figure 1b). The markers are used in the off-line analysis of recorded animal-robot experiments video data to determine the position and orientation of the robot.



(a) The design of the marXbot robot. (b) The Chicken Robot.

Fig. 1: The design of the marXbot robot and its Chicken Robot modification.

The Chicken Robot has a size of an adult chicken, energy autonomy up to several hours and Bluetooth connectivity.

B. The robot's behavior

The control architecture of the Chicken Robot is a behavior-based controller [25]. It consists of a collection of behaviors arranged in a hierarchy where behaviors on higher levels integrate behaviors on lower levels (Figure 3). The highest level behaviors are manually activated through the GUI on the user PC that has a Bluetooth connection with the robot; the behaviors of the second level work on the robot's microcontroller managing sensors.

When a *user control* behavior is activated, a user remote controls robot's displacements through the Bluetooth connection. If a *wandering* behavior is activated, the robot executes a forward move till sensors detect any obstacle. Based on the number of adjoining sensors detecting the obstacle the robot can distinguish two types of obstacles – walls and not walls. If the obstacle is a wall, the robot turns away. The walls avoidance algorithm is simple, the robot goes away from the wall with a turn-away angle equal to an angle of incident.

If the obstacle is not a wall, then the fuzzy collision avoidance system is activated. Based on the positions of obstacles it generates a new heading angle. When the obstacle is not detected any more, odometry data are used to return the robot to the original direction and it continues to move forward.

This behavioral model can be expanded by more complex behaviors depending on results of biological experiments. To combine behaviors we plan to apply the multivalued behavior control approach [10], [26].

Fig. 2: The base module of the marXbot robot.

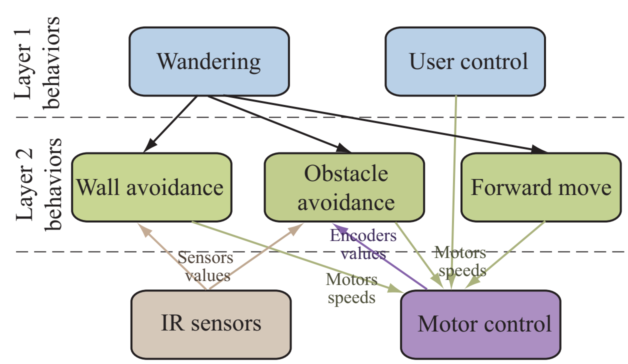


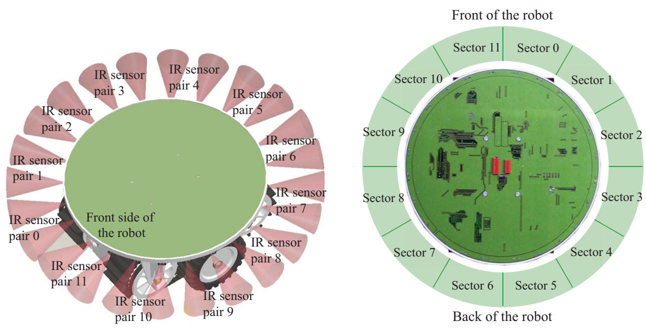
Fig. 3: An overview of the behavioral architecture of the Chicken Robot.

III. The obstacle avoidance method

A. The method's description

To design the fuzzy control system responsible for obstacle avoidance we use the approximate reasoning method. We first construct fuzzy partitions of an input and an output sets, and then determine a relationship between elements of the fuzzy partitions in the form of IF-THEN rules. This relationship is further used to determine the mapping of the input to the output space in the form of a single fuzzy relation.

An input set S is a finite set of infrared sensors of the Chicken Robot. To decrease computational load on the system we reduced the cardinality of S by grouping sensors in pairs (Figure 4a), thus $|S| = 12$. We chose an output set F to be a set of forbidden directions of travel, i.e. directions that lead to a collision with an obstacle. We also reduced its cardinality by partitioning a $360°$ range into sectors of $30°$ each (Figure 4b), $|F| = 12$.



(a) The infrared bumpers grouped in pairs.

(b) The directions of movement around the robot.

Fig. 4: The input and the output sets of the obstacle avoidance system.

A family of fuzzy subsets of X $\mathcal{A}^n = \{A_1, A_2, ..., A_n\}$ is called a *fuzzy partition* of X iff

$$\forall i, j \in \{1, ..., n\}(i \neq j \Rightarrow ((A_i \neq A_j) \wedge (0 < \sum_{k=1}^{m} A_i(x_k) < m)))$$

where m is a cardinality of X [14].

The Figures 5-6 represent the fuzzy partitions $\mathcal{O}^4$ and $\mathcal{D}^4$ of sets S and F correspondingly that we constructed for our fuzzy inference system. For simplicity reasons we restricted the number of elements in each of two partitions to four. We used triangular and trapezoid membership functions.

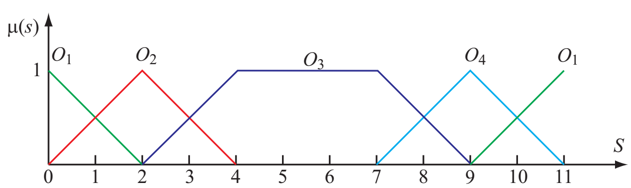


Fig. 5: The fuzzy sets combining the fuzzy partition $\mathcal{O}^4$.

Fuzzy sets of $\mathcal{O}^4$ have the following linguistic representation:

O_1 – *an obstacle is detected in front of the robot*;
O_2 – *an obstacle is detected in the direction of the sensor pair 2*;
O_3 – *an obstacle is detected behind the robot*;
O_4 – *an obstacle is detected in the direction of the sensor pair 10*;

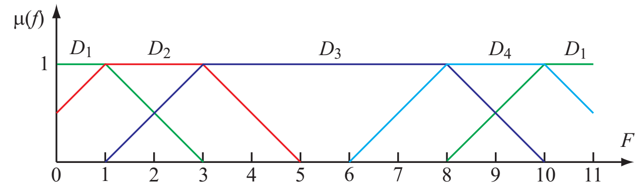


Fig. 6: The fuzzy sets combining the fuzzy partition $\mathcal{D}^4$.

while the fuzzy sets of $\mathcal{D}^4$ have the following linguistic representation:

D_1 – *a movement in directions of sectors 0, 1, 10 and 11 is disallowed*;
D_2 – *a movement in directions of sectors 1-3 is disallowed*;
D_3 – *a movement in directions of sectors 3-8 is disallowed*;
D_4 – *a movement in direction of sectors 8-10 is disallowed.*

The wideness of the fuzzy set O_3 and hence D_3 is explained by the fact that O_3 corresponds to obstacles behind the robot that are less important than obstacles close to the front of the robot.

After the fuzzy partitions on the input and the output sets are chosen, the next step is to establish a relationship between them. Since the elements of $\mathcal{O}^4$ and $\mathcal{D}^4$ have a clear linguistic meaning, empirical linguistic rules can be used in order to describe this relationship. Combining rules with the generalized modus ponens rule of inference, we obtain an approximate reasoning scheme for our obstacle avoidance system in the following form:

$$\begin{array}{lll} \text{RULE 1:} & \text{IF } o \text{ IS } O_1 & \text{THEN } d \text{ IS } D_1 \\ \text{RULE 2:} & \text{IF } o \text{ IS } O_2 & \text{THEN } d \text{ IS } D_2 \\ \text{RULE 3:} & \text{IF } o \text{ IS } O_3 & \text{THEN } d \text{ IS } D_3 \\ \text{RULE 4:} & \text{IF } o \text{ IS } O_4 & \text{THEN } d \text{ IS } D_4 \\ \text{ANTECEDENT:} & o \text{ IS } O & \\ \text{CONSEQUENCE:} & d \text{ IS } D. & \end{array} \quad (1)$$

A mathematical representation of the procedure (1) can be obtained by applying the compositional rule of inference. If we define knowledge contained by the rule base as a fuzzy relation $R : S \times F \rightarrow [0, 1]$ and $\mathbf{R}_{12\times12}$ is a matrix form of R, then any input o (a fuzzy subset on S) can be represented by a vector $\mathbf{o} = [O(s_1), O(s_2), ..., O(s_{12})]$ and an output $\mathbf{d} = [D(f_1), D(f_2), ..., D(f_{12})]$ is given by

$$\mathbf{d} = \mathbf{o} \circ \mathbf{R},$$

where $\circ$ denotes a *max-T* composition with T being a continuous *T*-norm. In this work we use $\min(a, b)$ *T*-norm.

The main problem here is to find the fuzzy relation R, i.e. the matrix $\mathbf{R}$. The conventional approach suggests to use any well-known fuzzy implication to represent each rule [27]. An alternative way is to solve a system of fuzzy relation equations, where each equation represents one rule $O_i \rightarrow D_i, i \in \{1, 2, 3, 4\}$:

$$\begin{cases} \mathbf{d}_1 = \mathbf{o}_1 \circ \mathbf{R} \\ \mathbf{d}_2 = \mathbf{o}_2 \circ \mathbf{R} \\ \mathbf{d}_3 = \mathbf{o}_3 \circ \mathbf{R} \\ \mathbf{d}_4 = \mathbf{o}_4 \circ \mathbf{R}, \end{cases} \quad (2)$$

where $\mathbf{o}_i = [O_i(s_1), O_i(s_2), ..., O_i(s_{12})]$ and $\mathbf{d}_i = [D_i(f_1), D_i(f_2), ..., D_i(f_{12})]$, $i \in \{1, 2, 3, 4\}$.

The selection of one of these two approaches depends on the application and cannot be predetermined, but the fuzzy relation equations based approach has such advantages as theoretical soundness and clearly defined properties [15], providing reliable models of processes and making the whole design process more consistent and systematic. Its main disadvantage is the potential nonexistence of the solution of (2). However in last decades a variety of necessary and sufficient conditions of the solution existence were derived [16], [20] as well as methods allowing to increase the solvability of equations by modifying a rule base [17] and techniques to find approximate solutions [18], [19].

A rule base and a fuzzy inference system using this rule base are said to be *consistent* when there is a common solution for all the equations of (2).

Let an antecedents matrix $\mathbf{O}$ is defined as

$$\mathbf{O} = \begin{bmatrix} \mathbf{o}_1 \\ \mathbf{o}_2 \\ \mathbf{o}_3 \\ \mathbf{o}_4 \end{bmatrix} = \begin{bmatrix} O_1(s_1) & O_1(s_2) & \dots & O_1(s_{12}) \\ O_2(s_1) & O_2(s_2) & \dots & O_2(s_{12}) \\ O_3(s_1) & O_3(s_2) & \dots & O_3(s_{12}) \\ O_4(s_1) & O_4(s_2) & \dots & O_4(s_{12}) \end{bmatrix},$$

and a consequences matrix $\mathbf{D}$ is defined as

$$\mathbf{D} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \\ \mathbf{d}_4 \end{bmatrix} = \begin{bmatrix} D_1(f_1) & D_1(f_2) & \dots & D_1(f_{12}) \\ D_2(f_1) & D_2(f_2) & \dots & D_2(f_{12}) \\ D_3(f_1) & D_3(f_2) & \dots & D_3(f_{12}) \\ D_4(f_1) & D_4(f_2) & \dots & D_4(f_{12}) \end{bmatrix}.$$

Then the system of equations (2) can be rewritten in the form of one fuzzy relation equation

$$\mathbf{O} \circ \mathbf{R} = \mathbf{D}. \tag{3}$$

Thus the rule base is consistent iff equation (3) is solvable.

For the rule base that we designed for our obstacle avoidance system the antecedents and consequences matrices have the following forms:

$$\mathbf{O} = \begin{bmatrix} 1 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1 \\ 0 & 1/2 & 1 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1 & 1 & 1 & 1 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1 & 1/2 & 0 \end{bmatrix},$$

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1 & 1 \\ 1/2 & 1 & 1 & 1 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1 & 1 & 1 & 1 & 1 & 1 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1 & 1 & 1 & 1/2 \end{bmatrix}.$$

For these given values of $\mathbf{O}$ and $\mathbf{D}$ equation (3) is solvable as the following solvability criteria holds true: if the input fuzzy sets A_i are semi-overlapped, satisfying the conditions

$$\text{height}(A_i \wedge A_{i+1}) = 1/2 \text{ and } \forall x \in X \sum_i A_i(x) = 1,$$

where A_i and A_{i+1} are adjacent normal fuzzy sets, then equation $\mathbf{A} \circ \mathbf{R} = \mathbf{B}$ can be exactly solved [28].

If a fuzzy relation equation is solvable, a number of numerical methods can be used to find its solution set [21], [29], [30]. We found $\mathbf{R}$ as

$$\mathbf{R} = \min_{\forall i}(\mathbf{o}_i \varphi \mathbf{d}_i), \tag{4}$$

where φ is a residual implication to a T-norm used in (3). For the min T-norm

$$\varphi(x, y) = \begin{cases} 1 & x \leq y \\ y & x > y \end{cases}.$$

Equation (4) gives a maximum solution of the equation (3) [31]. In our case

$$\mathbf{R} = \begin{bmatrix} 1 & 1 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1 & 1 & 1 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1 & 1 & 1 & 1 & 1 & 1 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1 & 1 & 1 & 1 & 1 & 1 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1 & 1 & 1 & 1 & 1 & 1 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1 & 1 & 1 & 1 & 1 & 1 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1 & 1 & 1 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1 & 1 \end{bmatrix}.$$

After $\mathbf{R}$ is one time computed, we can use it in the control system to calculate the output. The proposed obstacle avoidance system is presented on Figure 7. The fuzzification procedure computes the input vector $\mathbf{o}$ using the following formula

$$o_i = \frac{v_i - T}{V_{max} - T},$$

where v_i – proximity value from the i-th sensor pair (we take the largest of two sensors' values), V_{max} is the known largest proximity value and T is the obstacle avoidance activation threshold. The disallowed directions are calculated as

$$\mathbf{d} = \mathbf{o} \circ \mathbf{R}.$$

To find the steering angle we calculate an allowed direction fuzzy subset as $\overline{\mathbf{d}} = 1 - \mathbf{d}$ and then use the centroid of the largest area defuzzification method [10]. The resulted steering angle is sent to microcontrollers responsible for the motor control, where it is translated to the left and the right motor speed values.

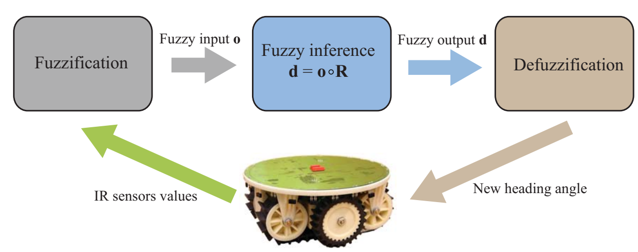


Fig. 7: The overview of the obstacle avoidance system.

B. Computational complexity estimation

We will show that the use of the fuzzy relation R in the inference process allows to reduce the computational load comparing with the conventional approach to fuzzy control. Let N be the cardinality of the input set, M be the cardinality of the output set, $\mathbf{o}' = [O'(s_1), ..., O'(s_N)]$ be an input vector, and vectors $\mathbf{o}_k$ and $\mathbf{d}_k$ represent the left and the right parts of the k-th rule, then the output vector

$\mathbf{d}'_k = [D'_k(f_1), ..., D'_k(f_M)]$ of the k-th rule is calculated as follows [27]

$$D'_k(f_j) = \max_i(T(O'(s_i), I(O_k(s_i), D_k(f_j)))),$$

where T is a T-norm and I is a fuzzy implication used to represent the rule. Hence, a computation of one element of the vector $\mathbf{d}'_k$ requires N calls of function I, N calls of function T and $N-1$ comparisons to find the maximal element. If the rule base contains K rules, then the outputs of all rules have to be computed that requires KMN calls of function I, KMN calls of T and $KM(N-1)$ comparisons.

If the fuzzy relation equation based approach is used, then we only compute a single output vector $\mathbf{d}' = [D'(f_1), ..., D'(f_M)]$:

$$D'(f_j) = \max_i(T(O'(s_i), R(s_i, f_j))), \quad j \in \{1, .., M\}$$

that requires only MN calls of T and $M(N-1)$ comparisons. Thus this approach provides a significant reduction in the computational load.

From the memory requirements point of view, the fuzzy relation matrix $\mathbf{R}$ contains MN elements to store, whereas to store the rule base we need only $(M+N)K$ elements; therefore, when input and output sets are of big cardinalities, storing of $\mathbf{R}$ is less efficient. However the size of $\mathbf{R}$ does not depend on the number of rules in the rule base, hence it is possible to considerably concretize control rules and expand the rule base without increasing the memory demands. For example, for our obstacle avoidance system $M = N = 12$, therefore $\mathbf{R}$ has 144 elements and if the rule base has more than six rules, then $\mathbf{R}$ needs less storage space than the rule base.

Hence, the fuzzy relation equation based approach provides the significant reduction in computational load and in many cases saves memory.

IV. Experimental results

This section validates experimentally the presented approach. We tested the obstacle avoidance system on the real platform that is the base module of the marXbot robot. The experimental set-up is composed of a square arena with sides of 1.5 meters in length, an overhead camera and tennis balls placed on the arena as obstacles. The speed of the robot is 60 mm/s, the obstacle avoidance is a real-time control loop at 10Hz. We used the tennis balls as their sizes approximately correspond to the sizes of young chicks. We recorded the experiments with the help of the color Scout Gigabit Ethernet camera scA1300-32gc from Basler.

Figure 8 shows the result of one of our runs. The trajectory of the robot is extracted from the recorded experimental video data with the help of the SwisTrack software [32]. The tests confirmed that using presented real-time obstacle avoidance system the robot is able to safely navigate on the arena successfully avoiding collisions. Repeating experiment many times we noticed that sometimes the robot gently touched the balls while avoiding them. It can be explained by the fact that in our system we use only four rules; this can be overcome by adding extra rules to make the rule base more accurate.

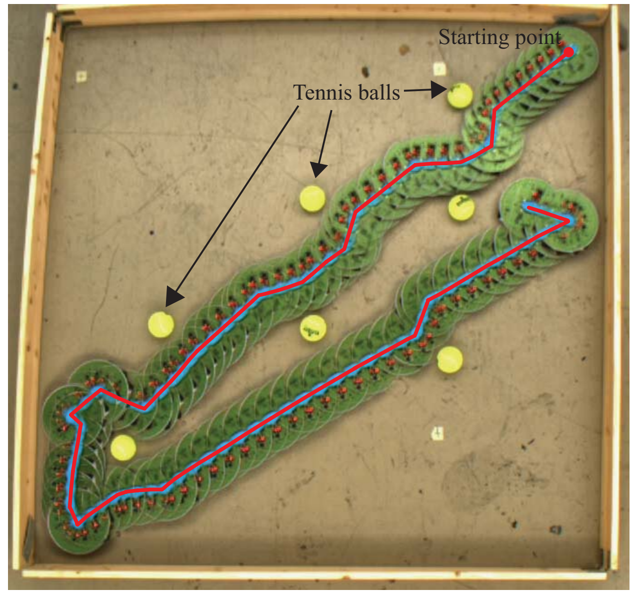


Fig. 8: The experimental results: the robot successfully avoids collisions with obstacles. The blue marker on top of the robot is used by the off-line tracker to estimate the robot's position and orientation.

V. Conclusions

In this paper, we have presented a reactive fuzzy collision avoidance system that was designed using the theory of fuzzy relation equations. Furthermore, we have presented experimental results that show that a mobile robot endowed with this system is able to successfully avoid obstacles and safely navigate on the experimental arena in real-time.

It does worth comparing the used fuzzy behaviors approach with two other popular methods of real-time reactive navigation, namely, the vector field histogram approach and the potential fields approach. The vector field histogram approach [9] uses the local information from the sensors to construct a one-dimension polar histogram of obstacles density that is then analyzed to determine candidate directions of movement, and the one closed to the goal direction is selected. In the potential field method [7] a robot's motion direction vector is computed as a combination of the repulsive fields from obstacles with the attraction fields from goals.

All three methods are not computationally heavy and can be performed in real-time on the dsPIC33 microcontroller, used in the Chicken Robot. However, each of them has features to take into account. The vector field histogram method can only deal with obstacle avoidance while pursuing a single goal location and is therefore less general than the potential field approach that is able to deal with multiple behaviors and to combine their outputs for a variety of different tasks. However, the potential field approach suffers from the well known problem of local minima in the potential field; its another problem is that arbitration of behaviors by vector addition can result in the command, which does not satisfy to any of the contributing behaviors. For example, a robot cannot pass through closely spaced obstacles. The fuzzy behaviors based method does not have this limitation, but similar to the potential field approach it allows to combine different behaviors to carry out complex tasks [33]. At the same time the vector field histogram method is able to produce shorter while less smooth paths than the fuzzy behaviors based one does [34]. Therefore, the approach based on the fuzzy behaviors is an optimal choice for the reactive mobile navigation system in the case when the flexibility to add new behaviors is demanded.

In our work several issues still remain open for improvement. The rule base that we employ is rather simple, using it we intended to test and illustrate the proposed approach. In future we are planning to expand the rule base, providing more detailed relationship between the input and output sets, and also to replace the current wall avoidance system by the fuzzy one, using the same design approach. Finally, the evaluation of our collision avoidance system on real animals in animal-robot experiments will be a part of the future work.

ACKNOWLEDGMENT

This work is supported by the Swiss National Science Foundation grant no. 112150 "Mixed society of robots and vertebrates".

REFERENCES

[1] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron. Experiments in automatic flock control. *Robotics and Autonomous Systems*, 31:109–117, 2000.

[2] N. Correll, M. Schwager, and D. Rus. Social control of herd animals by integration of artificially controlled congeners. In *10th International Conference on Simulation of Adaptive Behavior*, 2008.

[3] H. Ishii, M. Ogura, S. Kurisu, A. Komura, A. Takanishi, N. Iida, and H. Kimura. Development of autonomous experimental setup for behavior analysis of rats. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4152–4157, 2007.

[4] E. Kubinyi, A. Miklosi, F. Kaplan, M. Gacsi, J. Topal, and V. Csanyi. Social behaviour of dogs encountering AIBO, an animal-like robot in a neutral and in a feeding situation. *Behavioural Processes*, 65(3):231–239, March 2004.

[5] J. Halloy, G. Sempo, G. Caprari, C. Rivault, M. Asadpour, F. Tache, I. Said, V. Durier, S. Canonge, J. M. Ame, C. Detrain, N. Correll, A. Martinoli, F. Mondada, R. Siegwart, and J.-L. Deneubourg. Social integration of robots into groups of cockroaches to control self-organized choices. *Science*, 318(5853):1155–1158, 2007.

[6] Lesley J. Rogers. *The Development of Brain and Behaviour in the Chicken*. Oxford University Press, 1996.

[7] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 500–505, 1985.

[8] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *Robotics & Automation Magazine, IEEE*, 4(1):23–33, 1997.

[9] I. Ulrich and J. Borenstein. VFH+: reliable obstacle avoidance for fast mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1572–1577, 1998.

[10] J. Yen and N. Pfluger. A fuzzy logic based extension to Payton and Rosenblatt's command fusion method for mobile robot navigation. *IEEE Transactions on Systems, Man and Cybernetics*, 25(6):971–978, 1995.

[11] A. Saffiotti. The uses of fuzzy logic in autonomous robot navigation. *Soft Computing*, 1(4):180–197, 1997.

[12] K. H. Kim and H. S. Cho. An obstacle avoidance method for mobile robots based on fuzzy decision-making. *Robotica*, 24(5):567–578, 2006.

[13] E. Sanchez. Resolution of composite fuzzy relation equations. *Information and Control*, 30:38–48, 1976.

[14] G. B. Stamou and S. G. Tzafestas. Fuzzy relation equations and fuzzy inference systems: an inside approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 29(6):694–702, 1999.

[15] A. Di Nola, W. Pedrycz, S. Sessa, and E. Sanchez. Fuzzy relation equations theory as a basis of fuzzy modelling: An overview. *Fuzzy Sets and Systems*, 40(3):415–429, April 1991.

[16] S. Gottwald and W. Pedrycz. Solvability of fuzzy relational equations and manipulation of fuzzy data. *Fuzzy Sets and Systems*, 18(1):45–65, January 1986.

[17] K. Hirota and W. Pedrycz. Specificity shift in solving fuzzy relational equations. *Fuzzy Sets and Systems*, 106(2):211–220, September 1999.

[18] X. Li and D. Ruan. Novel neural algorithms based on fuzzy [delta] rules for solving fuzzy relation equations: Part iii. *Fuzzy Sets and Systems*, 109(3):355–362, February 2000.

[19] S. Gottwald. Approximate solutions of fuzzy relational equations and a characterization of t-norms that define metrics for fuzzy sets. *Fuzzy Sets and Systems*, 75(2):189–201, October 1995.

[20] I. Perfilieva and V. Novak. System of fuzzy relation equations as a continuous model of if-then rules. *Information Sciences*, 177(16):3218–3227, August 2007.

[21] B.-S. Shieh. Solutions of fuzzy relation equations based on continuous t-norms. *Information Sciences*, 177(19):4208–4215, October 2007.

[22] A. Di Nola, S. Sessa, W. Pedrycz, and E. Sanchez. *Fuzzy Relation Equations and their Applications to Knowledge Engineering*. Kluwer Academic Publishers, 1989.

[23] A. Di Nola and C. Russo. Lukasiewicz transform and its application to compression and reconstruction of digital images. *Information Sciences*, 177(6):1481–1498, March 2007.

[24] V. Loia and S. Sessa. Fuzzy relation equations for coding/decoding processes of images and videos. *Information Sciences*, 171(1-3):145–172, March 2005.

[25] R.C. Arkin. *Behavior-Based Robotics (Intelligent Robotics and Autonomous Agents)*. MIT Press, 1998.

[26] M. F. Selekwa, D. D. Dunlap, and E. G. Collins. Implementation of multi-valued fuzzy behavior control for robot navigation in cluttered environments. In *IEEE International Conference on Robotics and Automation*, pages 3688–3695, April 2005.

[27] F. Herrera. Searching for basic properties obtaining robust implication operators. In *Fuzzy Control, Fuzzy Sets and Systems 111*, pages 237–251, 2000.

[28] F.-L. Chung and T. Lee. Analytical resolution and numerical identification of fuzzy relational systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 28(6):919–924, 1998.

[29] L. Luoh, W.-J. Wang, and Y.-K. Liaw. Matrix-pattern-based computer algorithm for solving fuzzy relation equations. *IEEE Transactions on Fuzzy Systems*, 11(1):100–108, 2003.

[30] G. B. Stamou and S. G. Tzafestas. Resolution of composite fuzzy relation equations based on archimedean triangular norms. *Fuzzy Sets and Systems*, 120(3):395–407, June 2001.

[31] W. Pedrycz. Processing in relational structures: Fuzzy relational equations. *Fuzzy Sets and Systems*, 40(1):77–106, March 1991.

[32] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli. SwisTrack – a flexible open source tracking software for multi-agent systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008)*, pages 4004–4010, 2008.

[33] J. K. Rosenblatt. *DAMN: A Distributed Architecture for Mobile Navigation*. PhD thesis, Carnegie Mellon University, 1997.

[34] D. Shi, D. Dunlap, and E. G. Collins. A comparison between a fuzzy behavioral algorithm and a vector polar histogram algorithm for mobile robot navigation. In *International Symposium on Computational Intelligence in Robotics and Automation*, pages 260–265, June 2007.