Avoidance of mobile obstacles in real environments

M. Mucientes, M. Rodríguez, R. Iglesias, A. Bugarín and S. Barro
Dept. of Electronics and Computer Science
University of Santiago de Compostela.
15782 Santiago, Spain
{manuel, mrodri, rober, alberto, senen}@dec.usc.es

C.V. Regueiro
Dept. of Electronics and Systems
University of A Coruña.
15071 A Coruña, Spain
cvazquez@des.fi.udc.es

Abstract

This paper describes a fuzzy control system for the avoidance of moving objects by a robot in real environments. The objects move with no type of restriction, varying their velocity and making turns. Due to the complex nature of this movement and the lack of precision in ultrasonic sensors, it is necessary to perform an approximate estimation of the trend of the moving object. This is done by using a new paradigm of fuzzy temporal reasoning, which we call Fuzzy Temporal Rules. The controller has been subjected to an exhaustive validation process, with persons as mobile obstacles. One example showing the results obtained is presented.

Keywords: Avoidance of mobile obstacles, Robot navigation, Fuzzy Temporal Rules based control.

1 Introduction

One of the most complex problems in mobile robotics is the one of mobile obstacles avoidance. This complexity focuses in two aspects: the inherent difficulty of detecting the mobile obstacle/s and, on the other hand, the decision on which behaviour should be implemented for the robot to "intelligently" avoid collision. Only a few papers facing mobile obstacles detection in real environments have been presented in the bibliography. A reduced number of them present ultrasonic sensors-based solutions, in spite of ultrasonic sensors being very widely used in mobile robotics due to its simplicity and low cost. In most occasions, detection is made through more or less sophisticated vision systems (a single wide-angle camera, as in [Nair and Aggarwal, 1998]) or through different types of sensors supported by images (infrared and radar, as in [Cory *et al.*, 1998], ultrasounds with a wide-angle CCD camera [Song and Chang, 1999]). Only a few solutions are exclusively based on information provided by ultrasonic sensors. [de Lamadrid and Gini, 1990] use a ring made up of 24 ultrasound sensors, in order to detect mobile obstacles by means of detecting its extreme borders and [Sabatini and Colla, 1998] use a multiaural sensing head (with three ultrasound transducers) to recognise a person based on the rhythmic features of human walking.

On the other hand, there are a number of approaches for tackling the problem of robot navigation in general [Saffiotti, 1997] and, particularly, for avoiding moving obstacles. Some studies deal with estimating the moving object's future positions placing no restrictions on the trajectory of the obstacle, like the autoregressive model in [Elnagar and Gupta, 1998]. [Chang and Song, 1997] developed a neural network-based environment predictor that provides an estimate of future environment configuration by fusing multisensor data in real time. [Spence and Hutchinson, 1995] use a method based on attractive and repulsive forces, while [Chakravarthy and Ghose, 1998] use an approach based on the concept of a collision cone. Other approaches deal with the problem through making significant reductions in its complexity: [Tsoularis and Kambahmpati, 1999] describe the avoidance of one moving obstacle, in this case assuming a uniform and rectilinear movement, as it happens with the fuzzy approach by [Pratihar *et al.*, 1999]. Also within the frame of fuzzy rule-based approaches, [Arai *et al.*, 1999] determine the direction and speed of the mobile obstacle through message-passing, defining 4 different behaviours: turning, stopping, following and ignoring.

The solution we describe in this paper faces the two previously mentioned problems. Real mobile obstacles detection is made by using a ring of 16 ultrasonic sensors that permits building a real-time grid map onto which existence or not of obstacles is detected. On real environments, and due to the usual problems that arise for using this type of sensors (specular reflection, low angular resolution, ...), the evolution of state variables should be taken into account, in order to produce more robust behaviours, thus reducing the effect of perception failures. Furthermore, when decisions are taken only based on the current state of the robot, precipitated and inadequate actions may be taken. All of these aspects are much more evident in the case of a dynamic environment, when the robot, for example, needs to avoid collision with a number of mobile obstacles that may change their speed and/or direction. With the aim of avoiding collision, it is useful to evaluate the temporal evolution of the moving objects, thus making the handling of temporal information essential. Our approach to the problem aims to solve this by taking into account the history of recent values of determined variables, which enables us to reflect the different scenarios through which the obstacles have been passing and, thus, verify which their trends are. In this way, one can deduce which the behaviour of the robot should be, and take adequate actions (modification of its speed and/or turning the robot) in order to obtain a behaviour pattern in tune with the recent situations. Our system is robust, as it permits the avoidance of collisions even when the moving objects behave in a totally unexpected manner. The need to evaluate past situations and previous values of the variables (which in many cases are fuzzy) and principally, to reason them out, has led us to incorporate a temporal reasoning model which we call Fuzzy Temporal Rules (FTRs). Also, the ability of fuzzy rules to manage imprecise and vague information, like the provided by the ultrasonic sensors, has lead us to use FTRs.

The paper describes a system for the real detection and avoidance of free-moving mobile objects by a robot in a limited environment. The position of fixed obstacles like walls, doors, etc. delimits a variable width passageway where both the robot and the objects move freely. A fuzzy temporal knowledge-based control system has been designed where knowledge involved was modularised into three knowledge bases (figure 1). This makes tuning of the knowledge base easier, as well as allows achieving greater simplicity in each of the modules. The system

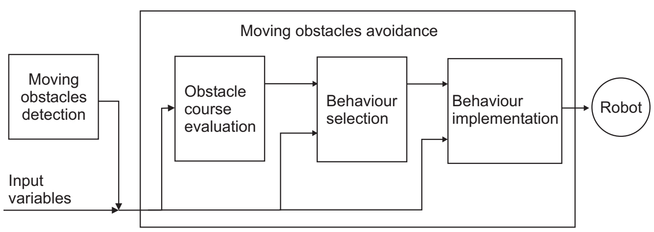

Figure 1: Schematic diagram of the system.

operates in real time (sending the robot three orders per second), enables the robot to operate with imprecise knowledge and takes into account the physical limitations of the environment in which the robot moves, obtaining satisfactory responses for the different situations analysed in real environments.

In the next section, all aspects related to the problem of moving obstacles detection by the real robot are presented. Section 3 is devoted to describe the problem of collision avoidance, whilst in Section 4 a brief introduction to our Fuzzy Temporal Rules (FTRs) model is made. Description of the fuzzy temporal controller for mobile obstacles avoidance is made in Section 5, while in Section 6 an operation example is discussed. Finally, conclusions obtained from this approach are presented.

2 Moving obstacles detection

The controller has been implemented on a Nomad 200 robot, which is endowed with an ultrasound ring comprising 16 sensors (with a range of approximately 6.5 meters) which supply the input data that are necessary in order to detect and avoid the moving obstacles. Detection is based on real-time building-up of a map of the space occupancy basing on the measurements from the ring of ultrasonic sensors. The degree of occupancy to each cell in the map is calculated by taking into account the number of times it is detected as being occupied and the number of times it is detected as being empty (count based model) [Rodríguez *et al.*, 2000]. This method comprises a very simple sensor model and an accumulation method. Each time that a sonar echo is perceived a value $v_t[i, j] = -1$ is assigned to the cells c_{ij} situated between the transmitter and the possible obstacle, and a value $v_t[i, j] = +1$ to the cells situated on the boundary sector of the sonar cone, independently of its position on it (figure 2). The accumulation method chosen is the

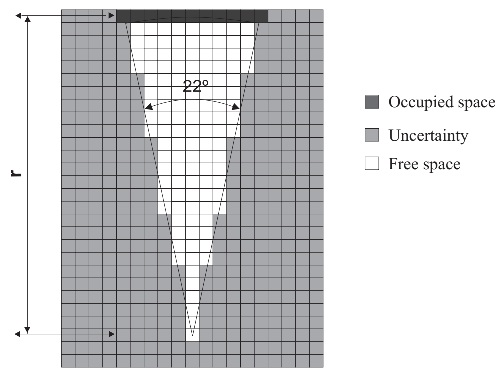

Figure 2: Sensor model used, showing the distribution of free and occupied space with regard to one ultrasound measurement.

SUM model, in which the occupancy value of each cell $C[i, j]$ of the map is obtained by algebraically adding the different observations

$$C_{t+1}[i, j] = C_t[i, j] + v_t[i, j] \qquad (1)$$

truncating to a maximum cell occupancy value. The initial value for all cells is 0, i.e., maximum uncertainty.

Detection of a moving obstacle is carried out by using the current measures from ultrasonic sensors and the existing map at the current instant. For all of the measures provided by sensors, fulfilment of a number of heuristic conditions is requested, in order to assume that a moving obstacle is detected by the sensor. These conditions are based on the following basic idea: a moving obstacle is assumed to be detected at a cell that is currently detected as occupied by the sensor, if that cell was previously detected as empty and is surrounded by empty cells. Fulfilment of the conditions is requested for windows centred in the current cell $C_t[i_0, j_0]$ (the one where ultrasonic measure is currently placed). Conditions that are tested in order to determine if a moving obstacle exists are:

1. Condition for the space surrounding a potential moving obstacle being empty: this is assumed when the number of cells (in a 9-cells side window) exhibiting a low occupancy degree ($C_t[i, j] \leq C_t[i_0, j_0] + 8$), or that are empty ($C_t[i, j] < 0$) is greater than a threshold.
2. Condition for filtering non-moving obstacles noise: the number of occupied cells ($C_t[i, j] > 0$) within a 21-cells side window is less or equal to 5.
3. Condition for filtering non-moving obstacles in high uncertainty situations: the number of unknown cells ($C_t[i, j] = 0$) within a 21-cells side window should be less than a threshold, in order to take a decision. For most real situations, if occupancy situation for all cells was requested, valuable points in the trajectory of the moving obstacle would be missing.

In order to assume that a moving obstacle was detected at $C_t[i_0, j_0]$, fulfilment of all conditions 1, 2 and 3 is requested for $C_t[i_0, j_0]$ and also for any close cell for at least one of the previous three measures.

Once potential moving obstacles were detected, a new moving obstacle is assumed to be found if no previously detected moving obstacle exists in 2 m distance. For all of the other cases the potential moving obstacle is associated to the moving obstacle detected in the previous cycle that is closer to it. Since multiple potential new positions can be produced for the same moving obstacle (as it happens, for example, when a single moving obstacle is detected by two sensors simultaneously), a second stage is accomplished that assigns the arithmetic mean of all potential new positions as the new current position for that obstacle. A sudden change in the moving direction is one important criterion that has to be taken into account in order to filter false detections. Whenever a change in direction is detected (with respect to the one that was previously estimated) that is greater than 90^o, that potential moving obstacle is assumed to be a noisy echo and therefore not taken into account.

After this, speed and moving direction of the obstacle are calculated. This is made only if the new position is at least 75 cm away from the previous one (the position of the moving obstacle is updated although this condition is not verified). Through this filtering, errors due to the lack of precision of the ultrasonic sensors are reduced, since otherwise extremely oscillating values for speed and direction of obstacles are calculated and therefore no clear trends in its movement could be detected.

If it is the case that a moving obstacle that was detected in the previous measure is not detected at the current one, it is assumed it stays at the same position with the same speed and direction as before, for at least three consecutive measures. This is done for avoiding false negative detections. If the moving obstacle is still missing after that three measures, it is assumed it has stopped at the last known position.

Finally, a least square approximation is made for speed and directions of all of the moving obstacles throughout the last three measures, for making sharp changes of these variables smoother. Unless this is done, false changes in direction or speed of the moving obstacle can be assumed, because of the low angular resolution of sensors, and also whenever a change in the sensor that detects the moving obstacle is produced.

3 Collision avoidance problem

Let $\vec{v}_{robot}$ be the velocity of the robot, $\vec{v}_{obstacle}$ the velocity of the moving object, R_{robot} the radius of the robot, and $R_{obstacle}$ the radius of the moving object (it is supposed that both the robot and the moving object are circular, which does not lead to a loss in generality -figure 3a-). In order to be able to determine in a simple

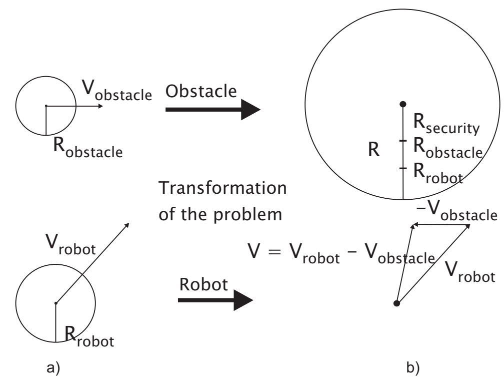

Figure 3: Transformation of the original problem a) in an equivalent one b), where the robot is a punctual object.

manner the existence or not of a collision and where it will take place, a problem transformation [Fiorini, 1995] is carried out into an equivalent static problem, shown in Figure 3: the velocity of the moving object is null and its size is $R = R_{robot} + R_{obstacle} + R_{security}$ and the robot is a punctual object with velocity $\vec{v} = \vec{v}_{robot} - \vec{v}_{obstacle}$. $R_{security}$ is the minimum distance to which the robot is permitted to approach the moving object, and this is established with the aim of maintaining a safety margin. Under this description, collision test is reduced to verifying the intersection between the straight line that is given by $\vec{v}$ and the circumference that represents the moving object.

In our system, it has been assumed that both the robot and the moving object have the same radius (approximately 25 cm), and the diameter of the robot was taken as the security radius, due to which $R = 4 \times R_{robot}$ (R=1 meter). In order to evaluate the proximity of the current situation with respect to the collision situation, parameter *non-collision index* is defined (figure 4):

$$nci = \begin{cases} \frac{\sin\alpha}{|\sin\beta|} = \frac{d_c}{R} & if\ \alpha \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \\ \frac{-d_o}{R} & if\ \alpha \in \left(-\frac{\pi}{2}, -\pi\right] \\ \frac{d_o}{R} & if\ \alpha \in \left(\frac{\pi}{2}, \pi\right) \end{cases} \qquad (2)$$

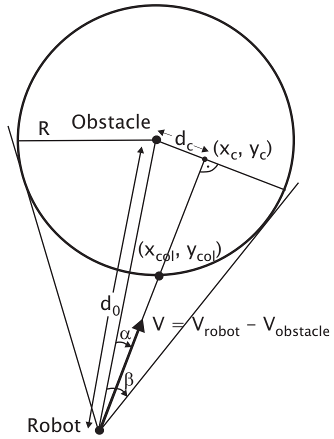

Figure 4: Parameters used in the definition of the *non-collision index*.

For values of the *nci* within interval $[-1, +1]$ a collision will occur. The coordinates of the collision point (x_{col}, y_{col}) are given by the point at which the line $\vec{v}$ intersects with the circle that represents the moving object (figure 4). At this point the robot will be at a distance $R_{security}$ from the moving object. In order to select, among all existing moving obstacles, which of them has to be considered with priority in order to avoid collision, parameter *expected collision time* is defined as the time available before the robot enters into the obstacle's security radius,

Variations in the value of *nci* and its temporal evaluation are of great interest for characterising the dynamic behaviour of the obstacle. The *nci* value may increase due, in general, to four causes: an increase in the robot's velocity, a decrease in the obstacle's velocity, the robot turning to its right and the obstacle turning to its right[1].

nci is used as a basis for the calculation of new parameters related with the evolution of the moving object and/or the robot, since any change in the behaviour of either of them will be reflected.

By analysing the past and present values of this variable, the current trend of the moving object can be deduced in an intuitive manner. Increasing *nci* situations are usually associated with scenarios in which the robot should pass in front (before) the obstacle. On the other hand, a decrease trend in the *nci* values is associated to scenarios where the moving obstacle passes first. Nevertheless, changes in the *nci* trend (as it happens when an increase is followed by a decrease) have to be correctly interpreted (for example, a situation where initially the robot should pass first that changed into a situation where the robot should give way to the obstacle). As an example, if an increase in the *nci* was produced in the past, but a decrease occurs in the last few moments, it is understood that the previous trend of the moving object to let the robot pass has changed, and has become that of passing first. Nevertheless, in real situations, it will be necessary to distinguish between true changes in trend as opposed to sporadic movements of the obstacle. This is accomplished in our approach by requiring a certain persistence or temporal maintenance in the new values of the *nci*, in order to interpret a situation as being changing. This need to bring temporal intervals into play and to analyse their occurrence in the values of the variables does not correspond directly with the usual structure of fuzzy control systems, with regards to both knowledge representational aspects as well as reasoning aspects. Use of average values for variables is not a generally valid alternative, since it cannot reflect, for example, sharp variations within a cycle. Similarly, the use of derivatives does not allow reasoning with past values of variables. Due to this we have used a FTRs model [Bugarín *et al.*, 1999] which is briefly described in the next section.

[1]This is true when incidence occurs from the left side. For right side incidences, an axis transformation is done.
An analogous analysis can be made for explaining *nci* decreases.

4 Temporal reasoning with FTRs

The following formulation for the fuzzy temporal propositions (FTPs) is assumed:

$$X \; is \; A \; < in \; Q \; of > \; T \tag{3}$$

where X is a linguistic variable, A represents a linguistic value of *X*, T is a temporal reference or entity and Q is a fuzzy quantifier.

The temporal entities T may represent both fuzzy temporal instants as well as fuzzy temporal intervals, being in both cases membership functions defined on a discrete set of values $\tau = \{\tau_0, \tau_1, \ldots, \tau_k, \ldots\}$, where each τ_k represents a time point and τ_0 represents the origin.

We assume that the values of this set are evenly spaced, where $\triangle = \tau_j - \tau_{j-1}$ is the unit of time, whose granularity depends on the temporal dynamics of the application[2].

When the fuzzy temporal entity T, represents an interval, it is sometimes necessary to require the fulfilment of *"X is A"* for at least one point of the temporal reference T (a situation which we refer to as *non persistence*, like in *"velocity is high in the last three seconds"*). Other situations require its fulfilment throughout the entire interval (*"throughout T"*: a situation of *persistence*), like in *"velocity is high throughout the last three seconds"*. Finally, fulfilment may be partially required (*"in the majority of T"*, *"in part of T"*).

The execution of a FTP consists on the calculation of the Degree Of Fulfilment (DOF). This is made as follows:

- *Non persistence: "X is A in T"*

$$DOF = \bigvee_{\tau_k \in \tau} \mu_A\left(X(\tau_k)\right) \wedge \mu_T(\tau_k) \tag{4}$$

[2]For this application $\triangle = \frac{1}{3}s$, which is the time that occurs between two consecutive control orders.

- *Persistence: "X is A throughout T"*

$$DOF = \bigwedge_{\tau_k \in \mathcal{T}} \mu_A\left(X(\tau_k)\right) \vee \left(1 - \mu_T(\tau_k)\right) \quad (5)$$

- *Intermediate case: "X is A in Q of T"*

$$DOF = \mu_Q\left(\frac{\sum_{\tau_k \in \mathcal{T}} \mu_A\left(X(\tau_k)\right) \wedge \mu_T(\tau_k)}{\sum_{\tau_k \in \mathcal{T}} \mu_T(\tau_k)}\right) \quad (6)$$

where μ_A is the membership function that is associated to the value A of the proposition, and $X(\tau_k)$ is the value measured for the variable X at the temporal point τ_k.

Operators $\wedge$ and $\vee$ are, respectively, the t-norm minimum and the t-conorm maximum, and μ_Q is the membership function associated to linguistic quantifier Q. It can be seen that, with this model, for all of the three cases, time points τ_k are weighted accordingly to its membership to the temporal reference $\mu_T(\tau_k)$.

5 Description of the control system

The knowledge fed into the system has been structured into a Fuzzy Temporal Knowledge Base (FTKB) made up of 117 rules, that has been modularised into three blocks. This was done for making easier the tuning of the FTKB and for doing rule updating in each block independent from the others.

The modules making up the FTKB are (figure 1):

1. *Obstacle course evaluation module*: its aim is to verify what movement strategy the obstacle is following; if it allows the robot to pass, if it will pass before, or if it is not aware of the robot.
2. *Behaviour selection module*: its aim is to decide on the optimum behaviour that the robot should follow in light of the trend of the moving object.
3. *Behaviour implementation module*: it aims to obtain the angular velocity and linear acceleration with which the robot is going to most suitably implement the desired behaviour for the current situation.

The avoidance of collision is based, to a good degree, on the accurate estimation of the tendency (first block), due to which this task is critical in the process.

5.1 Obstacle course evaluation module

The objective of this module is the estimation of the obstacle's movement tendencies, i.e., to characterise the dynamic scenario in which the robot is placed. By evaluating this situation, it is assumed that the object that interferes with the robot's trajectory is trying to pass before it or is letting it pass. In other cases it is not able to estimate a clear trend in the object's movements. The input variables for this block (which has 31 rules) are the *collision time*, the *collision_status_change* and the *nci_trend*.

The *collision time* (t_{col}) is a variable that estimates the time available before the robot enters into the obstacle's security radius. A low collision time supposes a sharp reaction on the part of the robot with the aim of avoiding a collision which seems imminent, whilst a high collision time enables it to act in a gradual manner.

The derived variable *collision_status_change* is used for detecting situations in which the robot passes from being in collision in one cycle, to not being so in a later cycle, or vice versa. By knowing the *nci* values in these two cycles, it is possible to determine whether the moving object tends to passing first, or to letting the robot pass. The possible values of the *collision_status_change* variable which are going to be considered in this problem are: *decrease*, *neutral* and *increase*.

The *nci_trend* variable gathers, on the contrary, a more precise evolution of the trend of the moving object, in which successive differences in the non-collision index are evaluated. It is defined as:

$$nci\_trend = \frac{nci(\tau) - nci(\tau - 2)}{2} \quad (7)$$

and its associated linguistic values are *decreases a lot*, *decreases*, *constant*, *increases* and *increases a lot* (see figure 5).

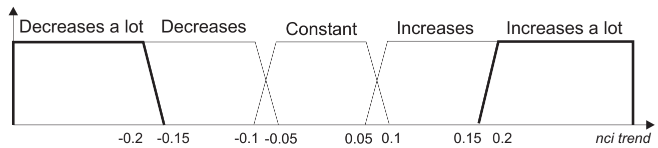

Figure 5: Set of linguistic values for variable *nci_trend*.

For this obstacle course evaluation module one single output variable is defined: *trend* of the moving object. With this we aim to estimate the behaviour of the moving object (the interest of FTRs lies here, since the evaluation of the moving object's movement must take time into account, and not summarise it in a simple average value), in order to then be able to act accordingly. Trend is defined as a crisp variable that may take the following values:

- *To_give_way*: the moving object will pass after the robot passes.
- *Indifferent*: this trend may be due to two reasons: on one hand, that the moving object is moving in a random manner (braking, accelerating or turning without there being any continuity in its movement) or because the moving object is not varying its speed (neither in module nor in direction).
- *To_pass_in_front*: the moving object will pass before the robot.

The rules of this knowledge base incorporate temporal reasoning, as shown in the following example:
"IF *collision_time is short* AND *collision_status_change is decrease in the_last_2_seconds* AND *nci_trend is not_increasing throughout the_last_second*
THEN *obstacle_aim is to_pass_in_front*"

The meaning of this rule is as follows: in a situation of relative proximity between the obstacle and the robot (*"collision_time is short"*), it is assumed that the trend of the former is *to_pass_in_front* if there has recently been at some point a decrease in the *collision_status_change* (for example, a change in the *nci* from very positive and outside collision into being in collision -*"collision_status_change is decrease in the_last_2_seconds"*-) and furthermore, even more recently, the *nci* has been maintained in its value or has decreased (*"nci_trend is not_increasing throughout the_last_second"*). A strict decrease in the *nci* is not required, as this decrease has been produced implicitly when the *collision_status_change* happened.

In general, if *collision_status_change* decreases, and the *nci* decreases or keeps constant, the trend will indicate that the moving object intends to pass (for a *to_give_way* trend and increase in the *nci* is required), whilst if subsequent to the *collision_status_change* decrease the *nci* increases, the trend will be *indifferent.*

5.2 Behaviour selection module

The objective of this block (with only 14 rules) is the determination of the type of behaviour that should be adopted by the robot faced with the trend given by the current situation of the moving obstacle. The most remarkable behaviours the robot implements are:

- *To_give_way*: in this behaviour pattern, the robot lets the moving object pass by, and it does so by braking (and sometimes turning).
- *Observe*: in this situation, the robot maintains its velocity (in module and direction). This is normally due the trend of the moving object not being clear.
- *To_pass_in_front*: here, the robot attempts to pass before the moving object by turning and/or accelerating.

A representative rule of this knowledge base is:
"IF *collision_time is high* AND *the obstacle's trend is to_give_way*
THEN *the robot's behaviour is to_pass_in_front"*

For the determination of the behaviour in this kind of rules firstly one has to pay attention to the trend value. If the detected trend is *to_give_way*, as a general norm, the robot's behaviour will be *to_pass_in_front*, whilst if the trend is *to_pass_in_front* the behaviour will be *to_give_way*. For an *indifferent* trend the behaviour will be selected taking into account the collision time. For high collision times the robot will act "aggressively", and hence the corresponding behaviour will be *to_pass_in_front* whilst for low collision times the robot will act in a more conservative manner, implementing a *to_give_way* behaviour. Lastly, with medium collision times, the robot will adopt intermediate tactics, and the behaviour pattern will be *observe*, due to which the robot will be waiting for future changes in the trend of the moving object. No oscillations between behaviours can occur, since the selection of the behaviour heavily depends on the current trend. Oscillations that may be produced for *nci* will not be propagated to the selected behaviour, since trend variable will be indifferent for this case.

5.3 Behaviour implementation module

In this final block the aim is to obtain the angular velocity and the linear acceleration to be sent to the robot. In order to do this, the selected behaviour, together with collision time and robot speed are considered. Also the type of incidence of the moving obstacle (front, transversal or rear), if the robot is placed or not within the trajectory of the obstacle and deviation are taken into account. Deviation is useful for making an estimation on how much advantageous (for avoiding collision) a turn is. A turn is assumed to be advantageous (and this is represented through negative values for deviation) whenever permits the robot to move towards the goal direction.

A total of 72 rules are used for implementing the different behaviour patterns for each of the situations in a precise and adequate manner. A rule that is representative of this knowledge base is:
"IF *the robot's behaviour is to_pass_in_front* AND *the collision time is medium* AND *the robot's velocity is medium* AND *the deviation is null* AND *the incidence is transversal*
THEN *increase velocity quite a lot* AND *turn a little"*

For high collision times, the reactions should be gentle (light turns and accelerations) whilst for low collision times the system usually applies maximum turn and acceleration with the aim of avoiding collision. In general, in the behaviour implementations the aim is to avoid turns (in order to not move away from the trajectory that the robot was following) except when these are favourable. For low collision times, this is not fulfilled, (there is no other solution for avoiding the collision), and for positive deviations a turn will be produced, although smoother than those implemented for negative or null deviations.

6 Results

The system has been tested for a number of cases, both simulated [Mucientes *et al.*, 1999] and real ones. For real tests (more than 50 were done), and using persons as moving obstacles, different types of incidences (front, rear, transversal) and obstacle tendencies (*to_give_way*, *to_pass_in_front*, *indifferent*) have been forced, showing that the system is robust and reliable, in spite of the low precision that the method for ultrasonic sensors-based moving obstacles detection produces. A real example of collision avoidance[3] is shown in figure 6. Ultrasonic sensors measures are represented by points, robot trajectory by circles (a higher concentration means lower speed), potential moving obstacles by (x) and detected moving obstacles by (+). Lines group all detected moving obstacles that correspond to the same existing obstacle. In general, all potential obstacles correspond to detected ones (represented by (*), as it happens for *M1*),

[3] A video recording of this test is available at *http://www-gsi.dec.usc.es/areas/robotica/rur01.htm.*

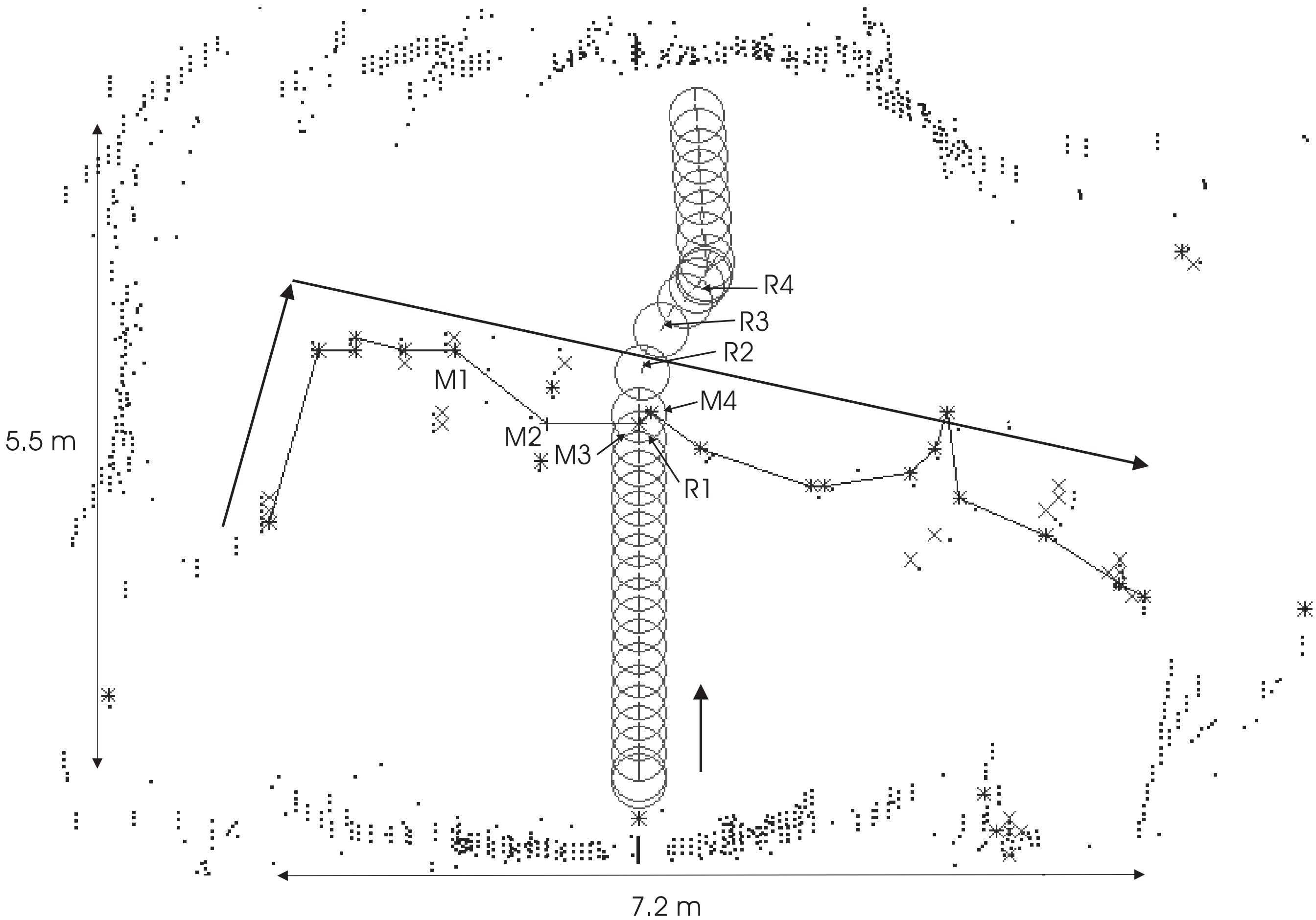

Figure 6: Example of collision avoidance with a person that lets the robot pass.

although for some cases (as in *M2*) this is not the case, since the final position is calculated as the arithmetic mean of two detected positions.

Figure 6 also shows an approximation of the real trace of the moving obstacle (the time spent by the robot in this test has been 17 s). At the beginning, the robot is moving at a constant velocity (18 cm/s) and detects the moving obstacle, but takes no action since no collision will be produced. This happens until the obstacle is placed at *M1* and the robot at *R1*. At this point, the *moving obstacles detection* module detects that, due to the obstacle turn, its tendency is *to_give_way*. This detection is done by means of the following rule:
"IF *collision_time is short* AND *collision_status_change is increase in the_last_2_seconds* AND *nci_trend is not_decreasing throughout the_last_second*
THEN *obstacle_aim is to_give_way*"

Due to this tendency, selected behaviour is *"to pass in front"* and it is implemented by means of an acceleration and turning to the right (since obstacle's incidence is from left to right). This behaviour repeats for the next two operation cycles, until collision is avoided. In order to get this, robot needed to increase speed up to 45 cm/s and turn around 20^o. These values are reached four iterations after the first collision situation was detected, since robot's motors response is not instantaneous.

This example also shows how our controller solves the problem of the obstacle's actual position being unknown, which may arise when the robot assumes it to be the last detected position. When the obstacle is placed at *M2* another collision situation occurs because of the change of obstacle position (1 m), which leads to an approach of the obstacle to the robot. This situation is solved in one cycle, by slightly increasing speed and turning to the right again. Finally, at positions *M3* and *M4*, and due to the change in the position of the mobile obstacle from *M2* to *M3*, a new collision situation occurs. In this case the most adequate action is to reduce the robot's speed, because the obstacle has overtaken the robot and is moving away. From final positions *R4* -for robot- and *M4* -for obstacle-, robot retakes its trajectory towards the goal point, since, moving obstacles avoidance system does not take control anymore.

7 Conclusions

In this paper real moving obstacles detection by means of ultrasonic sensors and a fuzzy temporal-based control has been described. The moving object approaching the robot is not subjected to any type of restriction in its movements, being able to vary its velocity and direction at any moment.

One of the three blocks into which the fuzzy knowledge base was divided has been modelled by means of Fuzzy Temporal Rules (FTRs), with the aim of being able to explicitly handle a history of recent values for correctly estimating the trend of the moving object. The FTRs model has been an essential tool for a correct evaluation of sudden and/or multiple changes in the trend of the mobile object.

An exhaustive process of validation has been made, through experiments that were carried out in real environments in which more than 50 tests in the most varied conditions of velocity and incidence angle have been made. In general, the controller produces adequate behaviours on the part of the robot, even when confronted with sudden changes in the trajectory and velocity of the moving object. The implementation of the controller is robust in spite of the high imprecision of the obstacle position detection system, has a low execution time and allows an easy design and tuning of the knowledge base.

Acknowledgements

Authors wish to acknowledge the support from the *Secretaría Xeral de I+D* of the *Xunta de Galicia* through grants PGIDT99PXI20603A and PGIDT99PXI20601B and from the the Spanish Ministry of Education and Culture (CICYT) and the European Commission through grant 1FD97-0183.

References

[Arai *et al.*, 1999] Y. Arai, T. Fujii, H. Asama, H. Kaetsu, and I. Endo. Collision avoidance in multi-robot systems based on multi-layered reinforcement learning. *Robotics and Autonomous Systems*, 29:21–32, 1999.

[Bugarín *et al.*, 1999] A. Bugarín, P. Cariñena, P. Félix, and S. Barro. *Fuzziness in Petri Nets*, volume 22 of *Studiess in Fuzziness*, chapter Reasoning with Fuzzy Temporal Rules on Petri Nets, pages 174–202. Physica-Verlag, 1999.

[Chakravarthy and Ghose, 1998] A. Chakravarthy and D. Ghose. Obstacle avoidance in a dynamic enviroment: A collision cone approach. *IEEE Trans. on Systems, Man and Cybernetics-A*, 28(5):562–574, 1998.

[Chang and Song, 1997] C.C. Chang and K.T. Song. Environment prediction for a mobile robot in a dynamic environment. *IEEE Trans. on Robotics and Automation*, 13(6):862–872, 1997.

[Cory *et al.*, 1998] P. Cory, H.R. Everett, and T.H. Pastore. Radar-based intruder detection for a robotic security system. In *Mobile Robots XIII and Intelligent Transportation Systems*, volume 3525 of *Proceedings of the SPIE*, pages 62–72, Boston (USA), 1998.

[de Lamadrid and Gini, 1990] J.F. Gil de Lamadrid and M.L. Gini. Path tracking through uncharted moving obstacles. *IEEE Trans. on Systems, Man and Cybernetics*, 20(6):1408–1422, 1990.

[Elnagar and Gupta, 1998] A. Elnagar and K. Gupta. Motion prediction of moving objects based on autoregresive model. *IEEE Trans. on Systems, Man and Cybernetics-A*, 28(6):803–810, 1998.

[Fiorini, 1995] P. Fiorini. *Robot Motion Planning Among Moving Obstacles.* PhD thesis, University of California, Los Ángeles, 1995.

[Mucientes *et al.*, 1999] M. Mucientes, R. Iglesias, C.V. Regueiro, A. Bugarín, P. Cariñena, and S. Barro. Use of fuzzy temporal rules for avoidance of moving obstacles in mobile robotics. In *Proceedings of the 1999 Eusflat-Estylf Joint Conference*, pages 167–170, Mallorca (Spain), 1999.

[Nair and Aggarwal, 1998] D. Nair and J.K. Aggarwal. Moving obstacle detection from a navigating robot. *IEEE Trans. on Robotics and Automation*, 14(3):404–416, 1998.

[Pratihar *et al.*, 1999] D.K. Pratihar, K. Deb, and A. Ghosh. A genetic-fuzzy approach for mobile robot navigation among moving obstacles. *Int. Journal of Approximate Reasoning*, 20(2):145–172, 1999.

[Rodríguez *et al.*, 2000] M. Rodríguez, J. Correa, R. Iglesias, C.V. Regueiro, and S. Barro. Probabilistic and count methods in map building for autonomous mobile robots. In *Advances in Robot Learning*, volume 1812 of *LNAI*, pages 120–137, Lausanne (Switzerland), 2000. European Workshop on Learning Robots, EWLR-8, Springer Verlag.

[Sabatini and Colla, 1998] A.M. Sabatini and V. Colla. A method for sonar based recognition of walking people. *Robotics and Autonomous Systems*, 25:117–126, 1998.

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

[Song and Chang, 1999] K.T. Song and C.C. Chang. Reactive navigation in dynamic environment using a multisensor predictor. *IEEE Trans. on Systems, Man and Cybernetics-B*, 29(6):870–880, 1999.

[Spence and Hutchinson, 1995] R. Spence and S. Hutchinson. An integrated architecture for robot motion planning and control in the presence of obstacles with unknown trajectories. *IEEE Trans. on Systems, Man and Cybernetics*, 25(1):100–110, 1995.

[Tsoularis and Kambahmpati, 1999] A. Tsoularis and C. Kambahmpati. Avoiding moving obstacles by deviation from a mobile robot's nominal path. *Int. Journal of Robotics Research*, 18(5):454–465, 1999.