[17] T. Oates, M. V. N. Prasad, and V. R. Lesser, "Cooperative information gathering: A distributed problem-solving approach," *Inst. Elect. Eng. Proc. Softw. Eng.*, vol. 144, no. 1, pp. 72–88, Feb. 1997.
[18] J. S. Wong and A. R. Mikler, "Intelligent mobile agents in large distributed autonomous cooperative systems," *J. Syst. Softw.*, vol. 47, no. 2, pp. 75–87, 1999.
[19] W. Caripe *et al.*, "Network awareness and mobile agent systems," *IEEE Commun. Mag.*, vol. 36, pp. 44–49, July 1998.
[20] K. N. Ross *et al.*, "Mobile agents in adaptive hierarchical Bayesian networks for global awareness," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*. Piscataway, NJ: IEEE Press, 1998, pp. 2207–2212.
[21] L. Prasad, S. S. Iyengar, and R. L. Rao, "Fault-tolerant sensor integration using multiresolution decomposition," *Phys. Rev. E*, vol. 49, no. 4, pp. 3452–3461, Apr. 1994.
[22] L. Lamport, "Synchronizing time servers," Digital System Res. Center, Palo Alto, CA, Tech. Rep. 18, 1987.

---

# Fuzzy Temporal Rules for Mobile Robot Guidance in Dynamic Environments

M. Mucientes, R. Iglesias, C. V. Regueiro, A. Bugarín, P. Cariñena, and S. Barro

***Abstract*—This paper describes a fuzzy control system for the avoidance of moving objects by a robot. The objects move with no type of restriction, varying their velocity and making turns. Due to the complex nature of this movement, it is necessary to realize temporal reasoning with the aim of estimating the trend of the moving object. A new paradigm of fuzzy temporal reasoning, which we call fuzzy temporal rules (FTRs), is used for this control task. The control system has over 117 rules, which reflects the complexity of the problem to be tackled. The controller has been subjected to an exhaustive validation process and examples are shown of the results obtained.**

***Index Terms*—Avoidance of moving obstacles, fuzzy control, fuzzy temporal rules (FTRs), robot guidance.**

## I. INTRODUCTION

One of the principal fields of research in robotics is the development of techniques for the guidance of autonomous robots. There are many complex problems in this field, mainly due to the nature of the real world (environments which are difficult to model) and the great uncertainty in these environments: the knowledge about an environment is often incomplete, uncertain and approximated, the information usually supplied by the robot sensors is limited and not totally reliable and the environment in which the robot is located usually has a dynamism which cannot be predicted. For all these reasons, fuzzy logic is a useful tool in the field of robotics [1], as has also been demonstrated in numerous studies carried out for guidance in real environments [2], [3], obstacle avoidance [4], route planning [5], etc.

A number of approaches for tackling the problem of robot navigation in the presence of a moving obstacle have been presented. Some studies deal with estimating the moving object's future positions using either an autoregressive model [6] or neural networks [7]. Reference [8] describes a method based on attractive and repulsive forces. On the other hand, in [9], an approach based on the concept of a collision cone is presented. In [10], a system for the monitoring of trajectories to be followed is described. The trajectories of the robot as well as of the moving objects are made up of linear segments along which they move at a constant speed. In [11] and [12], the avoidance of a moving obstacle is solved in a geometrical manner. Finally, in [13] and [14] the avoidance of moving obstacles is done using a fuzzy control system.

With respect to these solutions, a number of aspects should be pointed out. First, the fact that in some approaches the moving objects have restrictions in their movements. On the other hand, a robot usually acts according to the position of the moving object in the immediate past. In certain cases, this may lead to carrying out precipitated and inadequate actions. For instance, given two identical situations at present time, if one of them has been produced due to a hard brake of the moving object and the other one due to an acceleration of this object, they should be solved in a different way, although at present time both situations may look exactly the same.

Our approach to the problem aims to solve this by taking into account the history of more or less recent values of determined variables, which enable us to reflect the different scenarios through which the obstacle has been passing and, thus, verify what its trend is. In this way, one can deduce what the behavior of the robot should be, and take corresponding actions (modification of its speed and/or turning the robot) in order to obtain a behavior pattern in tune with the recent situations. This system is robust in its working, as it permits the avoidance of collisions even when the moving object behaves 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). The use of conventional fuzzy rules would not permit the direct treatment of this knowledge, since use of average values of variables, would not reflect sharp variations of a variable in a cycle, or it would take a long time to detect a gentle and constant change in a variable. Use of derivatives of variables is even less valid, since it does not permit reasoning with values from the past.

This paper describes a knowledge-based control system for the avoidance of a free-moving mobile object by a robot [16] in a limited environment.[1] The moving objects move varying their speed or turning with no restriction. The system operates in real time (sending the robot three orders/s), it is robust, it 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 a large number of different situations analyzed by means of the simulation software.

In the following section the problem is posed. In Section III the control system is described in detail, along with the presentation of the temporal reasoning model that is used. Section IV analyzes the results obtained for the simulations carried out and conclusions are given in Section V.

## II. POSING OF THE PROBLEM

As has already been mentioned, the movement of a robot in a dynamic environment is an extraordinarily complex problem. Besides avoiding the collision with the moving object, the robot must move

Manuscript received October 14, 1999; revised June 10, 2001. This work was supported by the Secretaría Xeral de I+D of the Xunta de Galicia under Grant PGIDT99PXI20603A and by the European Commission and Spanish Comisiòn Interministerial de Ciencia y Tecnologìa (CICYT) under Grant 1FD97 0183.

M. Mucientes, R. Iglesias, A. Bugarín, P. Cariñena, and S. Barro are with the Department of Electronics and Computer Science, University of Santiago de Compostela, Santiago de Compostela, Spain (e-mail: manuel@dec.usc.es; rober@dec.usc.es; alberto@dec.usc.es; puri@dec.usc.es; senen@dec.usc.es).

C. V. Regueiro is with the Department of Electronics and Systems, University of A Coruña, A Coruña, Spain (e-mail: cvazquez@udc.es).

Publisher Item Identifier S 1094-6977(01)08914-3.

---

[1] The robot used is a Nomad 200 by Nomadic Technologies [15].

in an environment that may have fixed obstacles (walls, etc.) which are restrictions on the movement to be carried out in order to avoid the moving object. To this, one has to add the restrictions imposed by the characteristics of the robot, such as the turn velocity, the linear acceleration, or the range of the sensors.

Let $\overrightarrow{v}_{robot}$ be the velocity of the robot, $\overrightarrow{v}_{obstacle}$ the velocity of the moving object (calculated by simple kinematics based on the positional coordinates in two successive instants $\tau_i$ and $\tau_i + 1$), $R_{robot}$ 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—Fig. 1(a)]. In order to be able to determine in a simpler manner the existence or not of a collision and where it will take place, we carry out a problem transformation [17], which enables us to pass from solving a cinematic problem between two nonpunctual objects to an equivalent static problem. In the equivalent transformed problem [Fig. 1(b)] the velocity of the moving object is null and its size is

$$R = R_{robot} + R_{obstacle} + R_{security} \tag{1}$$

and the robot is a punctual object with the velocity

$$\overrightarrow{v} = \overrightarrow{v}_{robot} - \overrightarrow{v}_{obstacle}. \tag{2}$$

$R_{security}$ 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 which, in any case, avoids the real collision between the object and the robot. Thus the collision test is reduced to verifying the intersection between the straight line that is given by the velocity of the robot relative to the moving object and the circumference that represents the moving object. In the case under study, 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 $R = 4 \times R_{robot}$ ($R = 1$ m).

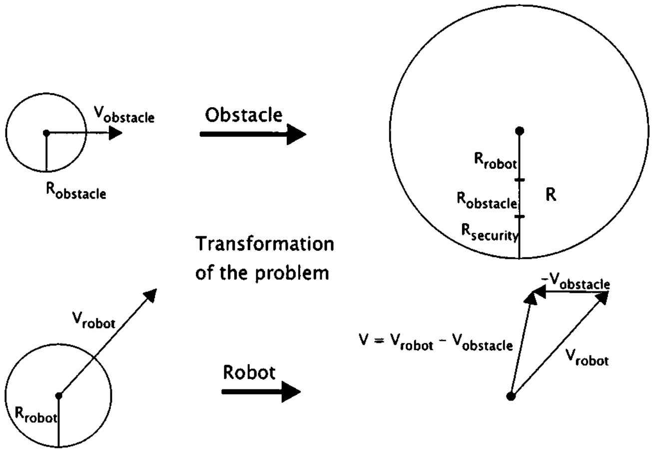
Parameter noncollision index (nci) is defined in this equivalent transformed problem for constantly evaluating the proximity of the current situation with respect to the collision situation[2] as

$$nci = \begin{cases} \dfrac{\sin\alpha}{|\sin\beta|} = \dfrac{d_c}{R}, & \text{if } \alpha \in \left[-\dfrac{\pi}{2}, \dfrac{\pi}{2}\right] \\ \dfrac{-d_o}{R}, & \text{if } \alpha \in \left[-\pi, -\dfrac{\pi}{2}\right) \\ \dfrac{d_o}{R}, & \text{if } \alpha \in \left(\dfrac{\pi}{2}, \pi\right) \end{cases} \tag{3}$$

where, as can be seen in Fig. 2, $d_o$ is the distance between the robot and the moving object and $d_c$ is the distance between the moving object and the point with coordinates $(x_c, y_c)$.

The angle $\alpha$ is formed by the line that joins the robot and the moving object (straight line $d_o$) with $\overrightarrow{v}$, and increases in a clockwise manner. Angle $\beta$ is the one formed by the straight line that is tangential to the circle with radius $R$ and straight line $d_o$.

The nci takes values in $[-d_o/R, d_o/R]$, which reduces as the robot approaches the moving object ($d_o$ decreases in this case). For values of the nci within the interval $[-1, +1]$ there is a collision situation. In order to obtain these values, angle $\alpha$ must be less or equal to $\beta$ (in absolute value), which indicates that there will be an intersection between the straight line given by the relative velocity of the robot with respect to the moving object and the circle with radius $R$. Positive values of the nci indicate that the robot is going to collide on its LHS with the moving object ($0 < nci \leq 1$), or that it is going to pass before the moving object ($nci > 1$), while negative index values reflect a collision on the RHS of the robot ($-1 \leq nci < 0$) or that the robot has let the moving object pass by ($nci < -1$).



(a) (b)

Fig. 1. (a) Original problem. (b) Transformation into an equivalent one, where the robot is a punctual object.

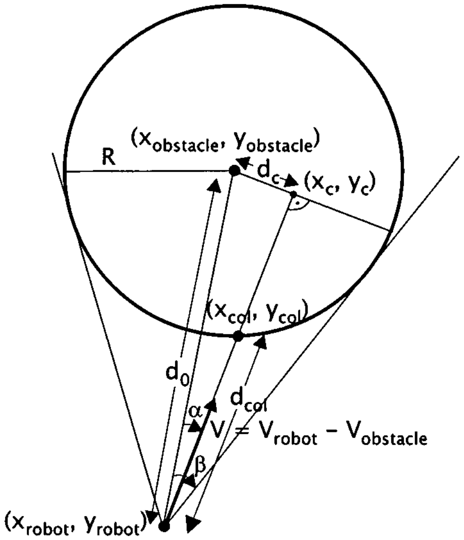


Fig. 2. Definition of the *noncollision index* (*nci*).

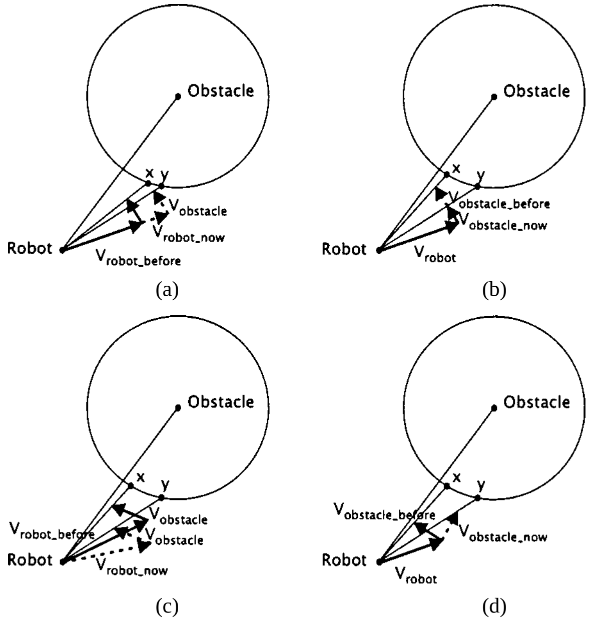
The coordinates of the collision point $(x_{col}, y_{col})$ are given by the point at which the line $\overrightarrow{v}$ intersects with the circle that represents the moving object (Fig. 2), where $d_{col}$ is the distance that separates the robot from this collision point. The robot will be at this point at the end of a certain time (*collision time*) if the velocities of either the robot and the obstacle are not altered (in module and direction). In this situation the robot will actually be at a distance $R_{security}$ from the moving object.

Variations in the value of nci and its temporal evaluation are of great interest for characterizing the dynamic behavior of the obstacle. Thus for merely illustrative purposes, the nci value may increase, due, in general, to the following four causes (Fig. 3).

1) An increase in the robot's velocity.
2) A decrease in the obstacle's velocity.
3) The robot turning to its right.
4) The obstacle turning to its right.

---

[2] It is assumed throughout all the explanations in the next sections that incidence of the moving obstacle is produced from the left-hand side (LHS). This is done for the sake of clarity and without loss of generality, since incidences from the right-hand side (RHS) are treated by means of a simple axis transformation.



(a) (b)

(c) (d)

Fig. 3. Causes for an increase in the nci value. In all of them the collision point goes from being "x" to being situated at "y." (a) Increase of the robot's velocity. (b) Decrease of the obstacle's velocity. (c) Turn of the robot to its right. (d) Turn of the obstacle to its right.

In the same manner, a decrease in the nci may be due to the following.

1) A decrease in the robot's velocity.
2) An increase in the obstacle's velocity.
3) The robot turning to its left.
4) The obstacle turning to its left.

This variable is also 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 behavior of either of them will be clearly reflected.

Having presented the problem, we now describe the intelligent control system. Here the knowledge that is necessary, based on current and past values of the variables, is gathered in order to supply the control orders that are needed to avoid the collision. An aspect that is particularly interesting in this point is the previously mentioned necessity to implement temporal reasoning on the evolution of the *nci*. By analyzing the past and present values of this variable, the current trend of the moving object can be deduced in an intuitive manner. As an example, if an increase in the *nci* had been produced, however in the last few moments there is a decrease, 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. In real situations, it will be necessary to distinguish between true changes in trend as opposed to sporadic movements of the obstacle, a motive due to which, in order to evaluate a situation as being changing, a certain persistence or temporal maintenance is required in the new values of the *nci*. This need to bring temporal intervals into play and to analyze 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. Due to this we have used the FTR's model [18], which is briefly described in Section III.

## III. DESCRIPTION OF THE CONTROL SYSTEM

In this section, each one of the fuzzy knowledge bases (FKBs) that make up the system as well as the FTRs model will be analyzed. The aim of the control system is to obtain those control variables that are sent to the robot with each order: its angular velocity and its linear acceleration. In order to do this a series of steps are followed which initially deal with estimating which maneuver the moving object is intending to carry out (its trend), and going on to select the type of behavior that the robot will require faced with this situation, and lastly to implement this behavior in the optimum manner (i.e., to obtain the most adequate values for the angular velocity and the linear acceleration).

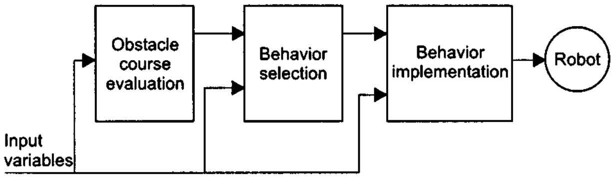


Fig. 4. Schematic diagram of the three modules into which the knowledge base is divided.

The FKB has been modularized into three blocks in order to, first, achieve greater ease in the tuning of the knowledge base, that is made up by 117 rules. Another great advantage is that the different blocks have a high degree of independence amongst themselves, hence modifications in one block do not influence the other blocks.

In order of execution (Fig. 4), the modules making up the knowledge base are the following.

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 wants to pass, or if it is not aware of the robot).
2) *Behavior Selection Module:* The aim of this block is to decide on the optimum behavior that the robot should follow in light of the trend of the moving object.
3) *Behavior Implementation Module:* This final module aims to obtain the angular velocity and linear acceleration with which the robot is going to most suitably implement the desired behavior for the current situation.

### A. Temporal Reasoning: Fuzzy Temporal Rules (FTRs) Model

In the majority of fuzzy control applications, knowledge is modeled by atemporal FKBs, in which the temporal dynamics of the processes are not taken into account, except in certain cases by means of variables defined for the purpose ("increase in velocity," "accumulated error," etc.). In many real-time applications like this one, that supposes a strong restriction on the possibilities of reasoning on the dynamics of the system and, in consequence, conventional fuzzy control is not a valid approach.

Due to the structure of the knowledge that is being modeled, in this application the model described in [18] has been used. The formulation for the propositions is

$$X \text{ is } A \ \langle \text{in } Q \text{ of} \rangle \ T \tag{4}$$

where

| | |
|---|---|
| $X$ | linguistic variable; |
| $A$ | linguistic value of $X$; |
| $T$ | temporal reference or entity; |
| $Q$ | fuzzy quantifier. |

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

Syntactic constructions "*in $Q$ of $T$*" may exhibit well differentiated semantics when $T$ represents a fuzzy temporal interval. The calculation

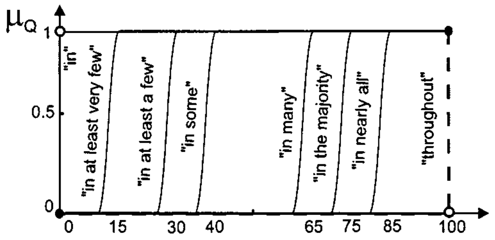


Fig. 5. Membership functions ($\mu_Q$) of the temporal quantifiers used.

of the degree of fulfillment (DOF) for a proposition like (4) is accomplished in that case by taking into account all the points belonging to the support[3] of $T$, in the following manner:

- *Nonpersistence: "$X$ is $A$ in $T$"*

$$\text{DOF} = \bigvee_{\tau_k \in SUPP_T} \mu_A(X(\tau_k)) \wedge \mu_T(\tau_k). \tag{5}$$

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

$$\text{DOF} = \bigwedge_{\tau_k \in SUPP_T} \mu_A(X(\tau_k)) \vee (1 - \mu_T(\tau_k)). \tag{6}$$

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

$$\text{DOF} = \mu_Q \left( \frac{\sum\limits_{\tau_k \in SUPP_T} \mu_A(X(\tau_k)) \wedge \mu_T(\tau_k)}{\sum\limits_{\tau_k \in SUPP_T} \mu_T(\tau_k)} \right) \tag{7}$$

where

| | |
|---|---|
| $\mu_A$ | membership function associated to the value $A$ of the proposition; |
| $X(\tau_k)$ | value observed for the variable $X$ in the instant $\tau_k$; |
| $\mu_T$ | membership function of the temporal reference; |
| $\mu_Q$ | membership function associated with the linguistic quantifier $Q$. |

In Fig. 5 some definitions of the membership functions $\mu_Q$ associated to the temporal persistence quantifiers used in the application are represented. The operators $\wedge$ and $\vee$ are the t-norm minimum and the t-conorm maximum, respectively, and in all of the cases, lower importance is given to the time points outside the kernel of $T$.

Fig. 6 shows an example of the calculation of the DOF for the proposition "*velocity is high throughout the last seconds.*" The process is as follows. First, the membership degree of the variable velocity to the linguistic label *high* for each instant $\tau_k$ is calculated. Thus, five membership degrees are obtained, one for each temporal instant belonging to the support of $T$. Next, and given that the proposition is of a persistent type ("*throughout*"), (6) is applied, which, in the example given, leads to a result of 0.5, which corresponds to the value measured at the temporal point $\tau_{now} - 4\Delta$.

### B. Obstacle Course Evaluation Module

The objective of this module is the estimation of the obstacle's movement tendencies, i.e., to attempt to characterize which is the dynamic scenario in which the robot is placed. Evaluating this situation, the robot will assume that the object that interferes with its trajectory is either trying to pass before it or is letting it pass. In other cases it will not be able to estimate a clear trend in the object's movements. The input variables for this block are

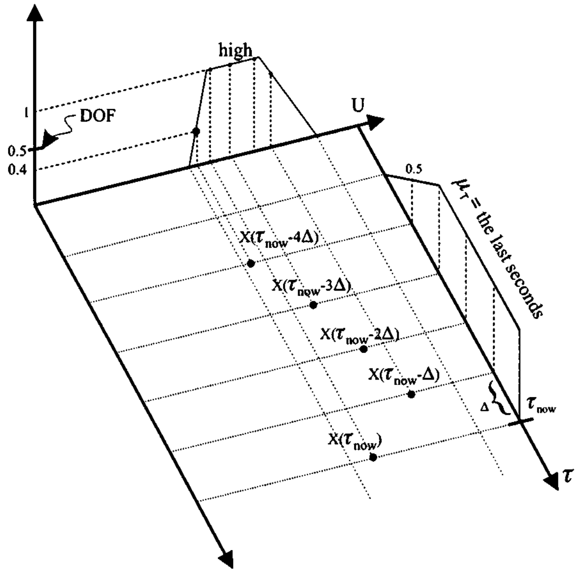


Fig. 6. Calculation of the DOF for the proposition "*velocity is high throughout the last seconds.*"

1) collision time;
2) collision_status_change;
3) nci_trend.

Variable collision time ($t_{col}$) estimates the time available before the robot enters into the obstacle's security radius: a low collision time will suggest a sharp reaction on the part of the robot with the aim of avoiding a collision which seems imminent, while a high collision time enables it to observe the situation and act in a more gradual manner.

The variable collision_status_change helps to detect 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 wishes to pass first, or is letting the robot pass. The possible values of the collision_status_change variable which are going to be considered in this problem are

1) decrease;
2) neutral;
3) 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 nci are evaluated. It is defined as a mean value in order to reduce error due to imprecision

$$\text{nci\_trend} = \frac{\text{nci}(\tau) - \text{nci}(\tau - 2)}{2}. \tag{8}$$

The values that are used for this variable are *decreases a lot, decreases, constant, increases,* and *increases a lot.*

For this module there is one single output variable, the trend of the moving object. It describes the behavior of the moving object in order to then be able to act accordingly. It is a crisp variable that takes the following values.

- *To_give_way*: The moving object intends to let the robot pass.
- *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).

---

[3] The support $SUPP_A$ of a membership function $\mu_A$ defined in a universe of discourse $U$ is defined as $SUPP_A = \{u \in U / \mu_A(u) > 0\}$.

- *To_pass_in_front*: The moving object is attempting to pass first.

The rules of this temporal knowledge base incorporate temporal reasoning and follow the FTR's model which we have presented. We now analyze the most noteworthy aspects of some representative examples. One type of rule is the following.

"IF *collision_time is short* AND *collision_status_change has decreased 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*, e.g., a change in the *nci* from very positive and outside collision into being in collision—"*collision_status_change has decreased 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* was realized.

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), while if subsequent to the *collision_status_change* decrease the *nci* increases, the trend will be *indifferent.*

Another possible situation is that there has been no *collision_status_change.* In this case the trend will be *to_give_way* if the *nci* increases substantially (for a *to_pass_in_front* trend a significant decrease in the *nci* would be required). This increase may be given, for example, by a decrease in the velocity module of the moving object. However, this decrease may take a more gradual form and have practically the same final effect (the braking is not so sharp, thus the moving object will be closer to the robot). It is in order to resolve this type of situation that rules of the following type have been introduced into the knowledge base:

"IF *collision_time is medium* AND *collision_status_change is neutral in the_last_3_seconds* AND *nci_trend is increasing in_at_least_a_few_points_of the_last_3_seconds* THEN *obstacle_aim is to_give_way.*"

In this case the requirement for the increase in the *nci* is not so strict (in this rule it is only stipulated that the *nci* should increase) due to which the change in the index may be lower, but in this rule the increase is needed *in_at_least_a_few_points_of* the interval *the_last_three_seconds* ("*a few*" here represents approximately 30%).

### C. Behavior Selection Module

The objective of this block is to fix the type of behavior that should be adopted by the robot, once considered the trend given by the current situation of the moving obstacle. The input variables of this module are collision time, trend, and limit_situation.

*Limit_situation* is a crisp variable that indicates when the robot is in an extreme situation in which it will attempt to leave the trajectory of the moving object as quickly as possible. The conditions for the robot to be in a *limit_situation* are, first, to be in the trajectory of the moving object,[4] and second that the incidence of the moving obstacle is *extremely frontal* or *extremely rear.*

There are a good number of possible behavior patterns. Moreover, given two equal behavior patterns, they do not necessarily have to be implemented in the same manner, rather this realization of behavior depends on a series of variables and it is the task that is accomplished by the behavior implementation module. The types of behavior that exist and the general description of their implementation are now given in the following.

- *To_give_way*: In this behavior pattern, the robot lets the moving object pass by, and it does so braking and sometimes turning.
- *Observe*: In this situation, the robot maintains its velocity (in module and direction). This is normally due to 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 accelerating.

An example rule of this knowledge base is the following.

"IF *collision_time is high* AND *the obstacle's trend is to_give_way* THEN *the robot's behavior is to_pass_in_front.*"

For obtaining the behavior in this kind of rule, first, one has to pay attention to the trend. If the trend is *to_give_way,* as a general norm, the behavior will be *to_pass_in_front,* while if the trend is *to_pass_in_front* the behavior will be *to_give_way*. For an *indifferent* trend the behavior will be selected taking into account the collision time. For high collision times the robot will act "aggressively," and hence the corresponding behavior will be *to_pass_in_front* while for low collision times the robot will act in a more conservative manner, realizing a *to_give_way* behavior. Lastly, for medium collision times, the robot will adopt intermediate tactics, and the behavior pattern will be *observe,* in which the robot will be waiting for future changes in the trend of the moving object.

Besides these basic behavior patterns, there are other situations in which there is a *limit_situation.* In those, the trajectories of the robot and of the moving object are fairly parallel and it is essential for the robot to turn in order to move out of the path of the moving object. The behavior patterns for *limit_situations* are characterized by their aim of leaving the path of the moving object as quickly as possible. In order to do so it will need to accelerate (as braking will not take it out of the moving object's path) and turn sharply.

Sometimes it may occur that collision cannot be avoided with the selected behavior (e.g., when the robot cannot turn, or its speed has reached the maximum limit and cannot be increased). For these cases, the behavior patterns obtained in this block are modified accordingly.

### D. Behavior Implementation Module

The aim of this block is to obtain the angular velocity and the linear acceleration, which are the commands that will be sent to the robot. The input variables in this module are *collision time, robot's velocity, behavior, deviation*, and vertical component index (vci).

The robot travels to its goal along a path. Any turn that lead the robot to that path is assumed as favorable, while a turn that takes it away from this path would be unfavorable. Variable *deviation* is defined for measuring how favorable a turn in a determined direction is for the robot. It may take the following values: *negative, null*, and *positive.*

*Negative* deviations describe favorable turns that will approach the robot to the path. *Null* values will also permit the turn, although this may also suppose moving slightly away from the path while for positive ones the turn will only be implemented in situations in which it is imperative to avoid a collision.

The sign of the *deviation* is selected taking into account the following. If the robot is moving toward a point situated to the left of the goal point, any turn to the right will be considered favorable since it will take it toward the goal, while a turn to the left will be considered unfavorable since it will take it away from the goal. The behavior *to_pass_in_front* will imply a turn to the right,[5] due to which, when the robot is moving toward a point situated to the right of the goal, the turn will not be favorable in this case, and due to this the sign is positive. For the same behavior, if the robot is moving toward the left of the goal,

---

[4] The trajectory of the moving obstacle is represented by a band whose width is equal to the diameter of the mobile object in the transformed problem.

[5] Remember we are describing LHS incidences.

that turn is considered favorable and thus the sign of the *deviation* is *negative* (for a *to_give_way* behavior the sign of deviation will be the opposite).

The vci indicates whether the incidence of the obstacle is frontal, transversal, or from the rear. It is necessary to differentiate between these situations, as the optimal manner of avoiding the collision is different. The vci is derived from the angle formed between the velocity of the robot ($\overrightarrow{v}_{robot}$) and the relative velocity between the robot and the moving object ($\overrightarrow{v}$). The set of values for the variable is made up of *rear, transversal*, and *frontal.*

The angular velocity that is sent to the robot in an order is obtained as

$$\text{Angular velocity} = \frac{\Omega \times \gamma}{1/3} \tag{9}$$

where

| | |
|---|---|
| $\Omega$ | one of the fuzzy variables of the consequent part of the rules of this module, represents the quantity of the turn that is going to be realized (*very little, a little* or *quite a lot* of turn); |
| 1/3 s | is the time between the control orders; |
| $\gamma$ | maximum number of degrees that it is possible to turn without colliding with the walls. |

The other output variable, the linear acceleration, is calculated as

$$\text{Linear acceleration} = \frac{\sigma \times \|v_{objective} - v_{now}\|}{1/3} \tag{10}$$

where $\sigma$ is the other fuzzy output of the controller, which represents the percentage in which the velocity of the robot is going to vary (*reduce very little, reduce a little, reduce quite a lot, reduce a lot, increase very little, increase a little, increase quite a lot* and *increase a lot*), $v_{now}$ (cm/s) is the current robot's velocity, $v_{objective}$ (cm/s) is the velocity that is desired to reach, and 1/3 is the time between two consecutive cycles. The objective velocity may take only two values: $v_{objective} = 0$ when the aim is to make the robot brake ($\sigma < 0$) or $v_{objective} = 61$ cm/s (maximum attainable robot's velocity) when $\sigma > 0$ and the robot accelerates.

The rules in this block (a total of 72) can be grouped according to the behavior (*to_pass_in_front, to_give_way*, and *limit_situation*) and the incidence (*rear, transversal*, and *frontal*). Thus, a rule for a *to_pass_in_front* behavior and transversal incidence could be the following.

"IF *the robot's behavior 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.*"

The three variables that are going to introduce modifications to the implementation of behavior are the collision time, the robot's velocity and the *deviation.* For high collision times, the reactions should be gentle (light turns and accelerations) while for low collision times the system usually applies maximum turn and acceleration with the aim of avoiding collision. In general, in the behavior 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 favorable (negative deviations). For low collision times, this is not fulfilled (there is no other solution for avoiding the collision), and for positive deviations there will be a turn, although less than those implemented for negative or null deviations.

## IV. ANALYSIS AND DISCUSSION OF RESULTS

The system has been subjected to a large number of simulations with the aim of verifying its validity and effectiveness. These tests have included the whole range of possible velocities, both for the robot and the moving object, as well as different angles of incidence (frontal, rear, left transversal, right transversal, etc.) as shown in Fig. 7.

| | |
|---|---|
| **Maximum translation velocity** | **61 cm/s** |
| **Maximum linear acceleration** | **76 cm/s$^2$** |
| **Maximum angular velocity** | **45° per second** |
| **Range of distances between the robot and the obstacle** | **0.5 to 6 meters** |
| **Range of collision times** | **1 to 30 seconds** |
| **Range of angles of incidence between the robot and the obstacle** | **0 to 2π** |
| **Types of movements of the obstacle** | **Accelerations, decelerations and turns** |
| **Types of trajectories of the obstacle** | **Straight-line, curved-line, zigzag (and their combinations)** |

Fig. 7. Characteristics of the simulations carried out.

Furthermore, with the objective of making the simulations as realistic as possible, the tests were carried out with randomly introduced noise in the position of the moving object received by the robot, in an attempt to simulate the imprecision of the robot's real ultrasound sensors. This noise is a function of the distance between the robot and the moving object (the greater the distance the higher the noise) and tests were carried out for a maximum percentage of 10% error.

The examples are given with graphical representations in which the trajectories of the moving object and the robot are described. Those of the former were chosen in order to show a selection of changes in module and direction of the velocity that face the robot with varied scenarios. A high concentration of marks indicates a lower velocity (of the obstacle or of the robot) whilst a low concentration is a reflection of a greater velocity.

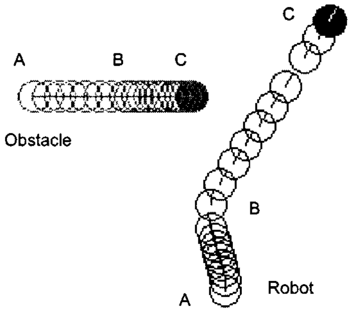


Fig. 8. Example 1. A, B, and C are the positions that the robot and the obstacle occupy in three time instants of the time interval represented.

In the first example (Fig. 8), the initial state is one in which the robot finds itself in a state of collision. Up until this moment, the trend detected by the obstacle course evaluation module was *indifferent,* since the *nci* had not varied.

At point A, there is an increase in the velocity of the moving object of approximately 25% (speed increases from 25 to 31 cm/s). Immediately the following rule is triggered:

"IF *collision_time is medium* AND *collision_status_change is neutral in the_last_3_seconds* AND *nci_trend is decreasing_a_lot in the_last_3_seconds* THEN *obstacle_aim is to_pass_in_front.*"

As the original situation was one of collision, there is no *collision_status_change,* but due to the acceleration of the obstacle, there is a decrease in the *nci* ("*nci_trend is decreasing_a_lot*"). Thus the trend will be that the moving object wants to pass first. Faced with this trend, the robot implements *to_give_way* behavior, which is reflected in a slight deceleration and a turn toward the left. In the two following cycles the same situation is repeated, the robot finally managing to avert the collision situation. In order to do so, the robot has had to turn and vary its velocity from 25 to 19 cm/s.

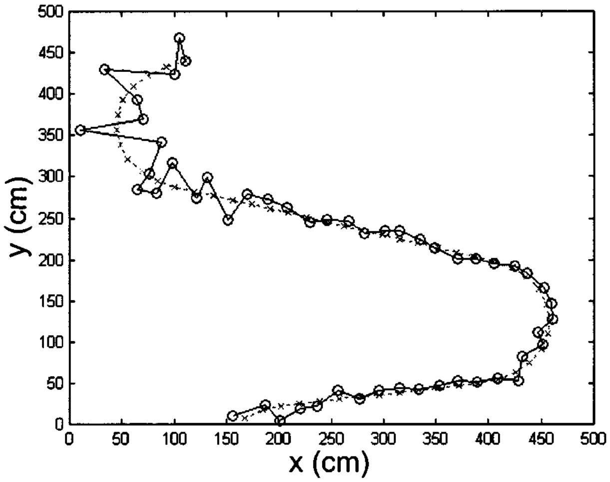


Fig. 9. Trajectory of the moving object. The actual path is represented by symbols $\mathbf{x}\cdot\cdot\mathbf{x}$ and the one sensed by the robot by symbols $\mathbf{o}$-$\mathbf{o}$.

At point B, the robot is in a situation in which there is no danger of a collision. At this instant, the moving object significantly reduces its speed (from 31 to 8 cm/s). Due to the deceleration, the *nci* has increased and the robot has passed from a situation in which there is no collision to one in which collision is possible. The trend that is determined in this situation for the moving obstacle is that it is now giving way to the robot, since there has been a *collision_status_change* (it has passed into a collision situation) and furthermore, the *nci* in the last second has increased or been constant. With the aim of avoiding the collision the robot selects *to_pass_in_front* behavior. This behavior pattern is implemented by accelerating quite a lot and turning a little toward the right (since the collision time is medium). In the three following cycles the situation is analogous but with low collision time, due to which robot accelerates and makes a right turn with greater intensity, thus succeeding in avoiding the collision, having increased its velocity from 19 to 50 cm/s.

In the second example the moving object moves toward the robot in a zigzag fashion. The moving object uses the whole width of the environment, and furthermore its linear velocity is 61 cm/s (which is the maximum speed that can be obtained by the robot). Thus, in this situation we are faced with a critical situation for the robot, as it will not be sufficient to vary its velocity, rather it will have to choose its trajectory with the utmost care. Moreover, the robot is working with an untrue position of the moving obstacle, due to the noise introduced into the system (because of the imprecision of the ultrasound sensors). Fig. 9 shows the actual trajectory of the moving object and the one that is handled by the robot due to the random noise introduced.

A simulation of this example is shown in Fig. 10. It can be seen how, due to the turns that it makes, the moving object's incidence changes. Once the robot has located the moving object (point A), it detects that the *nci* decreases sharply and thus the obstacle course evaluation module indicates that the moving object is attempting to pass first (this is due to the incidence being from the left, and the turn that the moving object is making is toward its left). With this trend the behavior that is decided for the robot is *to_give_way,* however, given that the robot is within the trajectory of the moving object, and in the previous cycles the behavior of the robot was *observe* (since it was not in a collision situation), the selected behavior is changed to *to_pass_in_front.*

The command that is sent to the robot (various rules contribute to the implementation of this behavior) implies a turn to the right (the robot attempts to pass first, and the incidence is from the left) and an acceleration, both with the maximum intensity possible.

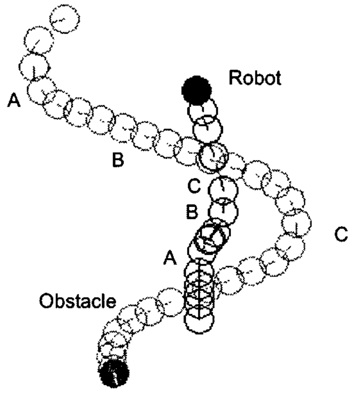


Fig. 10. Example 2. A, B, and C are the positions that the robot and the obstacle occupy in three time instants of the time interval represented.

In the second cycle the same situation is repeated, although the collision time is now extremely low. The situation changes in the following cycle, since due to the moving object's turn, the robot leaves its path and thus the selected behavior is now *to_give_way.* For transversal incidence and low collision time, maximum deceleration is applied to the robot for three cycles (at this moment the moving object has stopped turning) succeeding in avoiding the collision, even when in order to do so the robot has had to come to a halt (point B).

When the moving object reaches the RHS of the environment (point C) it turns and starts to approach the robot once again, which is now stationary. The situation is now one of incidence from the right, and as the moving object is turning to its right the obstacle course evaluation module interprets that the object wants to pass. Due to this change in the incidence, which is now from the right, it is necessary to transform the problem to an equivalent one, but with incidence from the left.

Although the object is attempting to pass, as the robot is faced with a limit situation, the behavior that is chosen is *limit_situation.* This behavior pattern produces a turn to the left, since greater margin to avoid collision exists at this side. The implementation will be a sharp acceleration and a pronounced turn to the left which will be maintained until the robot leaves the collision situation three cycles later.

In spite of the moving object's behavior being so random in this example, with unexpected turns made at high speed, the control system succeeds in avoiding the collision.

## V. CONCLUSION

A fuzzy temporal control system for the guidance of a robot in simulated dynamic environments has been described in this paper. 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 in which the FKB has been divided has been modeled 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 an accurate evaluation of sudden and/or multiple changes in the trend of the mobile object.

An exhaustive process of validation has been made, through simulations in which all type of tests in the most varied conditions of velocity and incidence angles, even for complicated trajectories, have been done. For all of them adequate behaviors on the part of the robot were obtained, even when confronted with sudden changes in the trajectory and velocity of the moving object.

The implementation of the controller is robust, has a low execution time, which is essential in a real-time application like this one, and allows an easy design and tuning of the knowledge base.

## REFERENCES

[1] A. Saffiotti, "The uses of fuzzy logic in autonomous robot navigation," *Soft Comput.*, vol. 1, no. 4, pp. 180–197, 1997.

[2] E. T. Lee, "Applying fuzzy logic to robot navigation," *Kybernetes*, vol. 24, no. 6, pp. 38–43, 1995.

[3] M. Toda *et al.*, "Navigation method for a mobile robot via sonar-based crop row mapping and fuzzy logic control," *J. Agric. Eng. Res.*, vol. 72, pp. 299–309, 1999.

[4] K. Althoefer *et al.*, "Fuzzy navigation for robotic manipulators," *Int. J. Uncertainty, Fuzziness, Knowl.-Based Syst.*, vol. 6, no. 2, pp. 179–188, 1998.

[5] J. de Lope, D. Maravall, and J. G. Zato, *Topological Modeling with Fuzzy Petri Nets for Autonomous Mobile Robots*, ser. No. 1416 in Lecture Notes in Artificial Intelligence. New York: Springer-Verlag, pp. 290–299.

[6] A. Elnagar and K. Gupta, "Motion prediction of moving objects based on autoregressive model," *IEEE Trans. Syst., Man, Cybern. A*, vol. 28, pp. 803–810, Nov. 1998.

[7] C. C. Chang and K. T. Song, "Environment prediction for a mobile robot in a dynamic environment," *IEEE Trans. Robot. Automat.*, vol. RA-13, pp. 862–872, Dec. 1997.

[8] R. Spence and S. Hutchinson, "An integrated architecture for robot motion planning and control in the presence of obstacles with unknown trajectories," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-25, pp. 100–110, Jan. 1995.

[9] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: A collision cone approach," *IEEE Trans. Syst., Man, Cybern. A*, vol. 28, pp. 562–574, Sept. 1998.

[10] J. F. Gil de Lamadrid and M. L. Gini, "Path tracking through uncharted moving obstacles," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-20, pp. 1408–1422, June 1990.

[11] A. Tsoularis and C. Kambahmpati, "On-line planning for collision avoidance on the nominal path," *J. Intell. Robot. Syst.*, vol. 21, no. 4, pp. 327–371, 1998.

[12] ——, "Avoiding moving obstacles by deviation from a mobile robot's nominal path," *Int. J. Robot. Res.*, vol. 18, no. 5, pp. 454–465, 1999.

[13] P. Garnier and T. Fraichard, "A fuzzy motion controller for a car-like vehicle," Institut National de Recherche en Informatique et en Automatique (INRIA), Tech. Rep. 3200, 1997.

[14] D. K. Pratihar, K. Deb, and A. Ghosh, "A genetic-fuzzy approach for mobile robot navigation among moving obstacles," *Int. J. Approx. Reasoning*, vol. 20, no. 2, pp. 145–172, 1999.

[15] Nomadic Technologies, Inc., "The Nomad 200 User's Guide," Nomadic Technologies, Inc., Mountain View, CA, 1997.

[16] M. Mucientes *et al.*, "Use of fuzzy temporal rules for avoidance of moving obstacles in mobile robotics," in *Proc. 1999 Eusflat-Estylf Joint Conf.*, Spain, 1999, pp. 167–170.

[17] P. Fiorini, "Robot motion planning among moving obstacles," Ph.D. dissertation, Dept. Mech., Aerospace, Nucl. Eng., Univ. California, Los Angeles, 1995.

[18] A. Bugarín *et al.*, *Fuzziness in Petri Nets*, ser. Vol. 22 of Studies in Fuzziness and Soft Computing. New York: Physica-Verlag, 1999, pp. 174–202.

---

# Adaptive Action Selection Without Explicit Communication for Multirobot Box-Pushing

Seiji Yamada and Jun'ya Saito

***Abstract*—This paper describes a novel action selection method for multiple mobile robots box-pushing in a dynamic environment. The robots are designed to need no explicit communication and be adaptive to a dynamic environments by changing modules of behaviors. The various control methods for a multirobot system have been studied both in centralized and decentralized approaches, however, they needed explicit communication such as a radio though such communication is expensive and unstable. Furthermore, though it is a significant issue to develop adaptive action selection for a multirobot system to a dynamic environment, few studies have been done on it. Thus, we propose action selection without explicit communication for multirobot box-pushing which changes a suitable behavior set depending on a situation for adaptation to a dynamic environment. First, four situations are defined with two parameters: the existence of other robots and the task difficulty. Next, we propose an architecture of action selection which consists of a situation recognizer and sets of suitable behaviors to the situations and carefully design the suitable behaviors for each of the situations. Using the architecture, a mobile robot recognizes the current situation and activates the suitable behavior set to it. Then it acts with a behavior-based approach using the activated behaviors and can change the current situation when the environment changes. We fully implement our method on four real mobile robots and make various experiments in dynamic environments. As a result, we find out our approach is promising for designing adaptive multirobot box-pushing.**

***Index Terms*—Action selection, behavior-based robots, box-pushing, cooperation, multirobot systems.**

## I. INTRODUCTION

For attacking a task which a single robot cannot achieve, many studies on multiple mobile robots cooperation have been done. They are categorized into two classes: *centralized control* [1]–[3] and *decentralized control* [4]–[11]. In the centralized control, a central system obtains global information on an environment including all the robots by sensing or communication and determines actions for all the robots. Then, the central system sends commands to all the robots and they act according to the commands. Though this approach has the advantage that robots act efficiently, it is less robust than decentralized control because all the robots stop when the central system is down. Thus, a multirobot system in decentralized control has also been investigated. However, both of the two approaches have the following significant issues.

1) *Explicit Communication*: Most multirobot systems [1]–[3], [9] using centralized control need explicit communication using a radio transmitter and a receiver. Even for decentralized control, some systems need explicit communication [12], [13]. Such explicit communication may be expensive and unstable depending on an environment. In contrast, a multirobot system without explicit communication is more robust and inexpensive.
2) *Dynamic Environment*: It is practical that an environment changes due to a fault of a robot, introduction of new robots, or task change, etc. However, most multirobot systems [1]–[11] do not have an effective mechanism to deal with a dynamic environment.

---

Manuscript received October 29, 1999; revised August 30, 2001. This paper was recommended by Associate Editor S. Lakshmivarahan.

The authors are with the CISS, IGSSE, Tokyo Institute of Technology, Yokohama 226-8502, Japan (e-mail: yamada@ymd.dis.titech.ac.jp).

Publisher Item Identifier S 1094-6977(01)10033-7.