Final Program

IEEE/RSJ International Workshop on Intelligent Robots and Systems '89 (IROS'89)

— The Autonomous Mobile Robots and its Applications —

Tsukuba Center for Institutes

Tsukuba, Japan

Sept. 4-6, 1989

IEEE Catalog Number: 89TH0265-9
IEEE Library of Congress Number: 89-84802

COSPONSORED BY:
IEEE Industrial Electronics Society
Robotics Society of Japan
New Technology Foundation

IN COOPERATION WITH:
IEEE Robotics and Automation Society
Japan Society of Mechanical Engineers (JSME)
Society of Instrument and Control Engineers (SICE)
Institute of Electric Engineers of Japan (JIEE)
Institute of Electronics, Information and Communication Engineers (IEICE)
Information Processing Society of Japan (IPSJ)
The Japan Society of Precision Engineering (JSPE)
Japan Society of Artificial Intelligence (JSAI)
Japan Industrial Robot Association (JIRA)
International Robotics and Factory Automation Center (IROFA)
Tsukuba Research Consortium (TRC)

COOPERATIVE ORGANIZATIONS:
Tsukuba Expo'85 Memorial Foundation
International Information Science Foundation

IEEE/RSJ International Workshop on Intelligent Robots and Systems '89, Sep. 4-6, 1989, Tsukuba, Japan

COGNITIVE COLLISION PREDICTION & PATH PLANNING OF MOBILE ROBOTS: A QUALITATIVE APPROACH

Behrouz HOMAYOUN FAR Takeshi ITOH Mitsuji SAMPEI Matsuroh NAKAMICHI

Graduate School of Science & Technology
Chiba University
1-33 Yayoi-cho,
Chiba-260, Japan

ABSTRACT

In this paper we introduce the preliminaries of qualitative collision prediction, collision avoidance and path planning of intelligent mobile robots by means of nonmonotonic temporal logic and qualitative reasoning techniques. In order to meet the on line limitations, the method of discrete iteration is exploited to produce the path without running the goal seeking algorithm again. The above techniques are implemented in a program. The paper demonstrates our approach through an example.

1. INTRODUCTION

Collision prediction, collision avoidance and path planning problems can all be formulated in a similar way, they all begin with a given present state and some assertions on the goal in a defined scenario, and the robots are required to predict the possible collision and decide upon the control to achieve the goal or subgoals satisfactorily. The abstract knowledge in all cases along with the spatiotemporal coordination remain the same. In the path planning case the concentration is on the spatial argument and in collision prediction avoidance the temporality is mainly concerned. Collision avoidance makes use of both.

1-1. The problem

In order to see what are the problems if one tries to predict events in the future and control them, suppose that the two dimensional scenario is composed of two mobile robots M_1 and M_2 with the preset goals G_1 and G_2 respectively, approaching each other, and their trajectories intersect spatially as in Fig. 1. Each robot has a potential behavior [14] in the lack of any other obstacle interrupting its trajectory (Fig.1-a). In case of collision only parts of the potential behavior until the collision point is manifested and after that there is a new behavior which doesn't necessarily terminate to the previously set goal (Fig.1-b).

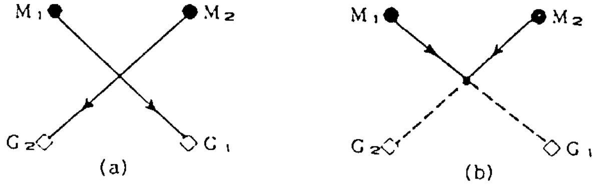

Fig. 1: The two mobile robot in two dimensional space.
a. Potential behaviors (No collision).
b. Manifested behaviors (Collision).

Intuitively, the human observer can predict the possibility of collision when tracking the movement of the robots and adjusting speed or direction to avoid collision. How the robot itself can predict the possible collision and avoid it? It requires a precice model of the situation at the outset. This model may become too extensive if specifying every single item in it, or too rigid if trying to stretch it over time. Specially, there is no guarantee for persistence [10] of the valid facts over extended period of time. These problems are identified as the qualification and extended prediction problems [13,14]. The qualification problem is defined as making sound predictions about the future without taking into account everything about the past [13]. This problem will be faded in a partially idealized world. The extended prediction problem is that in spite of being able to make predictions about short future intervals (immediate intervals), it is required to make a whole lot of them before predicting anything about a substantial part of the future [13]. The solution of both problems can be given by the two major AI techniques: nonmonotonic [14,15] and qualitative [4,7,9] reasoning.

From control point of view, each robot is engaged with two major tasks: state estimation and control. The basic goal of this research is to provide a theoretical basis for handling such tasks. In other words, we look for an underlying theory based on which an intelligent machine can be implemented with the capabilities of: First, deducing valid expressions indicating the robots will possibly collide based on the knowledge about their present spatiotemporal coordinates and final goals. Second, after detecting the possible collision, exploiting proper strategies and control actions to attain the goal (or subgoals) satisfactorily. For the first case we propose a solution based on nonmonotonic temporal logic (NTL). For the second one a knowledge base system is developed.

1-2. Overview

The overview of our proposed method is as follows: It is assumed that the behavior of the system is composed of an ordered set of potential behaviors extended over time intervals. Each potential behavior persists until meeting another one at a distinguished time point [9]. The distinguished time points are not known in advance. Within an interval it is assumed that the behavior is uniform, i.e., the present control strategy and set points persist until the next distinguished time point. Decision upon the next distinguished time point is made based on the collision prediction technique introduced here. Our method is novel specially as it blends the nonmonotonic temporal logic and qualitative inferential techniques uniformly. When a new possible collision is predicted, the new control strategy and control variables settings are decided upon by means of a problem oriented knowledge base. The outlines of developing such knowledge base is explained explicitly.

In this paper as an example we consider the collision prediction, collision avoidance and path planning for a mobile robot entering a garage. The only control variable is ω the angle between the present direction of movement and the one on the next immediate time interval. During each interval ω is fixed and it is supposed that the robot moves on an straight line on the direction indicated by ω. Within each interval the collision prediction technique will be used to predict the next possible collision. If possible collision is found, the interval will be terminated by the decision upon the control strategy and the updated value of ω.

1-3. Qualitative & Nonmonotonic Reasoning

Qualitative Reasoning (QR) method [4,7,9] derives behavior from the structure. The main advantage of using qualitative inferential techniques is to reduce the problem of control of continuous control variables to the one with a finite set of landmarks. Generally a large portion of time must be devoted to the acquisition of causal knowledge that associates a control action [2] and it may lead to infeasible implementation on real time basis. Qualitative technique allows us to cope with the changes of initial conditions without running again the goal seeking procedure, based on the knowledge of a sample behavior by means of the discrete iteration theory [16].

1-4. Time concept

We adopt the simple notation of time as ordered closed intervals. Furthermore, the time points denoting the both ends of an interval are positive integers. Time intervals meet but do not overlap [1,15].

1-5. Paper's Structure

The structure of the paper is as follows: In Section (2) we define the outlines of the method of collision prediction and Section (3) introduces the iterative control. In Section (4) we demonstrate the results through an example: a mobile robot entering a garage. In Section (5) we conclude with the present achievements and some future predictable extensions.

2. COLLISION PREDICTION

Cognitive collision prediction of mobile robots is within the scope of the more general problem of reasoning efficiently about the events over an extended period of time. The collision prediction tasks fall within three categories:

①. IMMEDIATE: Predicting a physical event possibly happening in an immediate future [2].

②. PERSISTENCE: Predicting a chain of physical events in distant future [10].

③. CAUSALITY: Relating the event chain such that each event is the consequence of the immediate predecessor one and cause of the successor [5,8].

Specifically, we are interested in propositions of the form: "If A is true now, then B is true later." The "If...Then" clause denotes causality. The A part encounters the information about the present position and speed of all the moving elements interacting in the scenario as well as all the external world elements affecting the situation. The B part is any valid expression indicating either the collision eventually happens or not.

In [13] two problems in formal temporal reasoning are defined as qualification and extended prediction problems. The qualification problem is making sound predictions about the future without taking into account everything about the past [13]. The extended prediction problem is that in spite of being able to make predictions about short future intervals (immediate intervals), it is required to make a whole lot of them before predicting anything about a substantial part of the future [13]. In [14] a solution to those two problems is suggested. In this paper we adopt the solution and tailor it to our problem of interest: the collision prediction problem. The key point in solution is exploiting nonmonotonic inferences to jump to a conclusion in the lack of a counterfact.

Suppose that the two robots move towards each other on straight lines as in Fig.1. Here one can differentiate two behaviors for each robot: potential and manifested. The potential behavior is the one naturally happens if there is no other interfering element in its course of action. On the other hand the manifested behavior is only parts of the potential behavior that actually happens in an interacting world. By properly defining conditions all the behaviors can be treated as potential. Potential behavior is synonym to the behavioral fragment (BF) [6] for a general physical system and it was learned that each BF is associated with a process. In the collision avoidance case we are engaged with temporally crossover processes [6]. The problem can be formulated either analytically (in CVDS or DEDS form) or logically by means of Nonmonotonic Temporal Logic (NTL). Here we introduce a DEDS model for the problem and formulate it in NTL.

2-1. The Model

State machines [11] are the general modeling constructs for discrete event dynamic systems (DEDS). ESM is a declarative structure which consists of a number of nodes connected by constraints. In ESM, Nodes are from the state set S, and each arc stands for a guarded "event" which show the transition among states. Event set Z is composed of predicates standing for the transition among states. Each event predicate denotes rule of the form:

Z : {Condition ---> Action}

Condition is any logic/analytic expression, when proved true, the action is enabled. Action is any change of data variables, synchronous communication among channels, or cooperation with a similar labeled event in some other ESM. Transition among the states of an ESM may be affected by an external MASK operations specified externally.

Qualitative structure of a set of ESMs is given in the form of shuffle ESM [14] determined by the concurrent action of the ESMs. The behavioral description for the shuffle ESM is a finite set of states on discrete-distinguished time points. Qualitative behavior of a system composed of n-ESMs (QBS) is an N-vector whose component QB_i is the qualitative state of the ESM_i on the distinguished time point.

$QS(\Phi, t_i) = [QS(\phi_1, t_i) \ldots\ldots, QS(\phi_m, t_i)]$

where:

$\Phi = \{\phi_1, \phi_2, \ldots\ldots, \phi_m\}$ is the set of ESMs.

$QS(\Phi, t_i)$ is composed of:

①. states of the ESMs, now are in it.

②. values of the data variables at this point for all ESMs.

③. possible change to next state for all ESMs.

Fig.2 shows the extended state machine (ESM) model of the mobile robot with only one possible collision in its course of action. In a more complicated scenario, there may be multiple collisions along with many MOVE states. Here after STARTing, the robot will stay in the MOVE1 state representing the manifested part of the process until either COLLIDing with another robot or reaching to the end of the process denoted by END. Collide state shifts the behavior from MOVE1 to MOVE2.

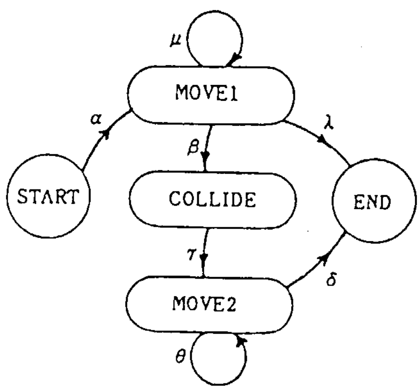

Fig.2 : ESM for the mobile robots.

The events are defined by the temporal expressions in the next paragraphs. In a scenario of two mobile robots, we get to a shuffle product of the two ESMs shown in Fig.3. The critical paths are those including the central node $\langle C_1, C_2 \rangle$.

2－2. NTL scheme

Reasoning over extended period of time is central in collision prediction. In order to build a logical model of our problem we consider the nonmonotonic temporal logic (NTL). In traditional logic the meaning of a formula is the set of interpretations (i.e., truth assignment) that satisfy it or its set of models [14]. In nonmonotonic logic we are interested in only a preferred subset of the models. For our problem, the preference criterion [15] on the models may be taken as the minimum number of deflections of the straight line trajectory or the temporal order of the deflections. The minimum number of deflections can not represent the preference criteria perfectly. For the two robots human observer can predict that they possibly collide and the number of deflections for each robot is only one due to the collision. In a dynamic world with time lag in updating the memory, suppose that obstacles suddenly come from nowhere on the path of the two robots before they get to each other and collide with them. Still the number of deflections will be one for each but apparently the outcomes are completely different. In the temporal ordering preference, the events happened later are the preferred ones. In this example it is clear that the deflection happens later in the first scenario (wanted) than the second one (unwanted). The combination of the two schemes is ideal.

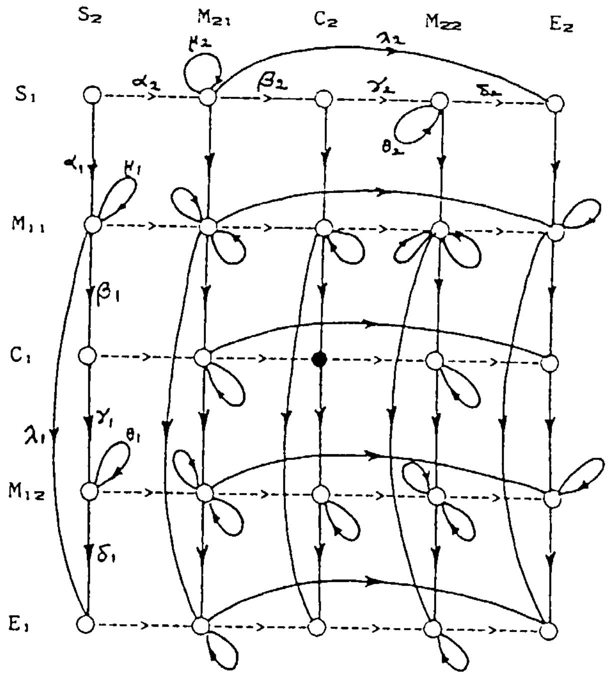

Fig.3 : Shuffle product of the ESMs of the two robots

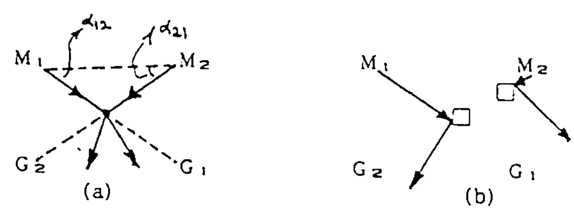

Fig.4 : The minimum deflection:
a. Correct; b. Incorrect.

In the two robot scenario, the propositions assuring the continuity of movement are:

α_1 : TRUE {t, Start (R_1) } →
TRUE {t, Move1 (R_1) } (1)

α_2 : TRUE {t, Start (R_2) } →
TRUE {t, Move1 (R_2) } (2)

μ_1 : TRUE {t, Move1 (R_1) } ∧
¬TRUE {t, Stop (R_1) }
→ TRUE {t+1, Move1 (R_1) } (3)

μ_2 : TRUE {t, Move1 (R_2) } ∧
¬TRUE {t, Stop (R_2) }
→ TRUE {t+1, Move1 (R_2) } (4)

Where the time points are discrete and ordered and represented by integers. Now we should add the conditions of collision and those preventing it. The necessary condition for collision is the mutual distance between the two robots become zero or their mutual sight angle (SA) reduce to zero or 180° simultaneously. The second premise is novel and adopted in this work because it is easier technically to measure and track the angular deflection than the distance. The sight angles α_{12} and α_{21} are the angle between the direction of movement and the straight line joining the two robots (Fig. 4-a). It is visible that when the two robots approach each other, decrease of α_{12} will result in increase of α_{21} and vice-versa. In qualitative terms, the interesting landmark points are those when either α_{12} or α_{21} are zero or 180°. When one intersects the path of the other, its sight angle becomes zero or 180°. Therefore collision happens when both α_{12} and α_{21} reach zero or 180° simultaneously. The collision condition then can be written:

$$\begin{aligned}\beta 1:\ & \text{TRUE}\ \{t,\ \text{Move1}(R_1)\}\ \wedge \\ & \neg\text{TRUE}\ \{t,\ \text{Stop}(R_1)\}\ \wedge \\ & \text{TRUE}\ \{t,\ [\alpha_{12}]\downarrow\}\ \wedge\ \Phi \rightarrow \\ & \text{TRUE}\ \{t+1,\ \text{Collide}(R_1, R_2)\} \qquad (5)\end{aligned}$$

$$\begin{aligned}\beta 2:\ & \text{TRUE}\ \{t,\ \text{Move1}(R_2)\}\ \wedge \\ & \neg\text{TRUE}\ \{t,\ \text{Stop}(R_2)\}\ \wedge \\ & \text{TRUE}\ \{t,\ [\alpha_{21}]\uparrow\}\ \wedge\ \Phi \rightarrow \\ & \text{TRUE}\ \{t+1,\ \text{Collide}(R_2, R_1)\} \qquad (6)\end{aligned}$$

Where "$\downarrow$" stands for the qualitative decrease and "$\uparrow$" represents the qualitative increase. We add up all the physical rules that may affect the movement of the two robots in Φ. This list can be remarkably large and nonsense, for example there is enough fuel to proceed moving, there is no strong wind to prevent movement and etc. The logic of chronological ignorance [14] which is a nonmonotonic logic of temporal knowledge, resolves Φ propositions in a preference scheme.

The movement after collision is expressed by the following expressions:

$$\begin{aligned}\gamma_1:\ & \text{TRUE}\ \{t,\ \text{Collide}(R_1, R_2)\} \rightarrow \\ & \text{TRUE}\ \{t,\ \text{Move2}(R_1)\} \qquad (7)\\ \gamma_2:\ & \text{TRUE}\ \{t,\ \text{Collide}(R_2, R_1)\} \rightarrow \\ & \text{TRUE}\ \{t,\ \text{Move2}(R_2)\} \qquad (8)\\ \theta_1:\ & \text{TRUE}\ \{t,\ \text{Move2}(R_1)\}\ \wedge\ \neg\text{TRUE}\ \{t,\ \text{Stop}(R_1)\} \\ & \rightarrow \text{TRUE}\ \{t+1,\ \text{Move2}(R_1)\} \qquad (9)\\ \theta_2:\ & \text{TRUE}\ \{t,\ \text{Move2}(R_2)\}\ \wedge\ \neg\text{TRUE}\ \{t,\ \text{Stop}(R_2)\} \\ & \rightarrow \text{TRUE}\ \{t+1,\ \text{Move2}(R_2)\} \qquad (10)\\ \lambda_1:\ & \text{TRUE}\ \{t,\ \text{Move1}(R_1)\}\ \wedge\ \neg\beta\ \wedge\ \neg\mu \qquad (11)\\ \lambda_2:\ & \text{TRUE}\ \{t,\ \text{Move1}(R_2)\}\ \wedge\ \neg\beta\ \wedge\ \neg\mu \qquad (12)\\ \delta_1:\ & \text{TRUE}\ \{t,\ \text{Move2}(R_1)\}\ \wedge\ \neg\theta \qquad (13)\\ \delta_2:\ & \text{TRUE}\ \{t,\ \text{Move2}(R_2)\}\ \wedge\ \neg\theta \qquad (14)\end{aligned}$$

The pure temporal logic can only partially predict the collision. The reason is that according to expressions (5) and (6) the collision will certainly happen if the preconditions are satisfied. But as explained above, there are some situations that the two robots just miss each other with a narrow margin ($[\alpha_{12}]=0$ but $[\alpha_{21}] \neq 180°$ yet). Introduction of the modality [3] will remove this drawback. The temporal-modal logic model of the two mobile robot scenario is:

$$\begin{aligned}\alpha_1:\ & \square\ \{t,\ \text{Start}(R_1)\} \rightarrow \\ & \square\ \{t,\ \text{Move1}(R_1)\} \qquad (1')\\ \alpha_2:\ & \square\ \{t,\ \text{Start}(R_2)\} \rightarrow \\ & \square\ \{t,\ \text{Move1}(R_2)\} \qquad (2')\\ \mu_1:\ & \square\ \{t,\ \text{Move1}(R_1)\}\ \wedge \\ & \Diamond\ \{t,\ \neg\text{Stop}(R_1)\} \rightarrow \\ & \square\ \{t+1,\ \text{Move1}(R_1)\} \qquad (3')\\ \mu_2:\ & \square\ \{t,\ \text{Move1}(R_2)\}\ \wedge \\ & \Diamond\ \{t,\ \neg\text{Stop}(R_2)\} \rightarrow \\ & \square\ \{t+1,\ \text{Move1}(R_2)\} \qquad (4')\\ \beta_1:\ & \square\ \{t,\ \text{Move1}(R_1)\}\ \wedge \\ & \Diamond\ \{t,\ \neg\text{Stop}(R_1)\}\ \wedge \\ & \square\ \{t,\ [\alpha_{12}]\downarrow\}\ \wedge\ \Phi \rightarrow \\ & \Diamond\ \{t+1,\ \text{Collide}(R_1, R_2)\} \qquad (5')\\ \beta_2:\ & \square\ \{t,\ \text{Move1}(R_2)\}\ \wedge \\ & \Diamond\ \{t,\ \neg\text{Stop}(R_2)\}\ \wedge \\ & \square\ \{t,\ [\alpha_{21}]\uparrow\}\ \wedge\ \Phi \rightarrow \\ & \Diamond\ \{t+1,\ \text{Collide}(R_2, R_1)\} \qquad (6')\\ \gamma_1:\ & \square\ \{t,\ \text{Collide}(R_1, R_2)\} \rightarrow \\ & \square\ \{t,\ \text{Move2}(R_1)\} \qquad (7')\\ \gamma_2:\ & \square\ \{t,\ \text{Collide}(R_2, R_1)\} \rightarrow \\ & \square\ \{t,\ \text{Move2}(R_2)\} \qquad (8')\\ \theta_1:\ & \square\ \{t,\ \text{Move2}(R_1)\}\ \wedge \\ & \Diamond\ \{t,\ \neg\text{Stop}(R_1)\} \rightarrow \\ & \square\ \{t+1,\ \text{Move2}(R_1)\} \qquad (9')\\ \theta_2:\ & \square\ \{t,\ \text{Move2}(R_2)\}\ \wedge \\ & \Diamond\ \{t,\ \neg\text{Stop}(R_2)\} \rightarrow \\ & \square\ \{t+1,\ \text{Move2}(R_2)\} \qquad (10')\\ \lambda_1:\ & \square\ \{t,\ \text{Move1}(R_1)\}\ \wedge\ \neg\beta\ \wedge\ \neg\mu \qquad (11')\\ \lambda_2:\ & \square\ \{t,\ \text{Move1}(R_2)\}\ \wedge\ \neg\beta\ \wedge\ \neg\mu \qquad (12')\\ \delta_1:\ & \square\ \{t,\ \text{Move2}(R_1)\}\ \wedge\ \neg\theta \qquad (13')\\ \delta_2:\ & \square\ \{t,\ \text{Move2}(R_2)\}\ \wedge\ \neg\theta \qquad (14')\end{aligned}$$

Where "$\Diamond$" means "possibly true" and "$\square$" means "necessarily true".

Propositions (3')-(4') and (9')-(10') stand for the persistence of the states over the next time instant. There are two sources that can defease these propositions: either new specific information or causal information. The former is a new fact proved true, for example, the robot stops. The latter is another fact that causes the removal of the first expression, for instance, the robot's ignition unit is turned down. We can elaborate the model by introducing the upper time bounds on the expressions.

$$\begin{aligned}\alpha_1:\ & \square\ \{t_1, t_1,\ \text{Start}(R_1)\} \rightarrow \\ & \square\ \{t_1, t_1,\ \text{Move1}(R_1)\} \qquad (1'')\\ \alpha_2:\ & \square\ \{t_1, t_1,\ \text{Start}(R_2)\} \rightarrow \\ & \square\ \{t_1, t_1,\ \text{Move1}(R_2)\} \qquad (2'')\\ \mu_1:\ & \square\ \{t, t,\ \text{Move1}(R_1)\}\ \wedge \\ & \Diamond\ \{t \le t_2,\ \neg\text{Stop}(R_1)\} \rightarrow \\ & \square\ \{t+1 \le t_2,\ \text{Move1}(R_1)\} \qquad (3'')\\ \mu_2:\ & \square\ \{t, t,\ \text{Move1}(R_2)\}\ \wedge \\ & \Diamond\ \{t \le t_2,\ \neg\text{Stop}(R_2)\} \rightarrow \\ & \square\ \{t+1 \le t_2,\ \text{Move1}(R_2)\} \qquad (4'')\\ \beta_1:\ & \square\ \{t \le t_2,\ \text{Move1}(R_1)\}\ \wedge \\ & \Diamond\ \{t \le t_2,\ \neg\text{Stop}(R_1)\}\ \wedge \\ & \square\ \{t \le t_2,\ [\alpha_{12}]\downarrow\}\ \wedge\ \Phi \rightarrow \\ & \Diamond\ \{t+1 \le t_2,\ \text{Collide}(R_1, R_2)\} \qquad (5'')\\ \beta_2:\ & \square\ \{t \le t_2,\ \text{Move1}(R_2)\}\ \wedge \\ & \Diamond\ \{t \le t_2,\ \neg\text{Stop}(R_2)\}\ \wedge \\ & \square\ \{t < t_2,\ [\alpha_{21}]\uparrow\}\ \wedge\ \Phi \rightarrow \\ & \Diamond\ \{t+1 \le t_2,\ \text{Collide}(R_2, R_1)\} \qquad (6'')\\ \gamma_1:\ & \square\ \{t, t,\ \text{Collide}(R_1, R_2)\} \rightarrow \\ & \square\ \{t, t,\ \text{Move2}(R_1)\} \qquad (7'')\\ \gamma_2:\ & \square\ \{t, t,\ \text{Collide}(R_2, R_1)\} \rightarrow \\ & \square\ \{t, t,\ \text{Move2}(R_2)\} \qquad (8'')\\ \theta_1:\ & \square\ \{t, t,\ \text{Move2}(R_1)\}\ \wedge \\ & \Diamond\ \{t \le t_2,\ \neg\text{Stop}(R_1)\} \rightarrow\end{aligned}$$

$\square\ \{t+1 \leq t_2,\ Move2\ (R_1)\ \}$ (9'')

$\theta_2 : \square\ \{t, t,\ Move2\ (R_2)\ \}\ \wedge$
$\Diamond\ \{t \leq t_2,\ \neg Stop\ (R_2)\ \} \rightarrow$
$\square\ \{t+1 \leq t_2,\ Move2\ (R_2)\ \}$ (10'')

$\lambda_1 : \square\ \{t, t \leq t2,\ Move1\ (R_1)\ \}\ \wedge\ \neg\beta\ \wedge\ \neg\mu$ (11'')

$\lambda_2 : \square\ \{t, t \leq t2,\ Move1\ (R_2)\ \}\ \wedge\ \neg\beta\ \wedge\ \neg\mu$ (12'')

$\delta_1 : \square\ \{t, t \leq t2,\ Move2\ (R_1)\ \}\ \wedge\ \neg\theta$ (13'')

$\delta_2 : \square\ \{t, t \leq t2,\ Move2\ (R_2)\ \}\ \wedge\ \neg\theta$ (14'')

Where t_2 is the upper time bound and $t_1 \leq t \leq t_2$.
This concludes the issues on modeling and collision prediction. In the following sections we will see that NTL embedded in DEDS can lead to collision prediction satisfactorily.

3. BEHAVIORAL ITERATION

The main advantage of using qualitative inferential technique is the ability of coping with the changes of initial conditions without running again the tedious goal seeking program, based on the knowledge of a sample behavior. Therefore it is ideal for on-line implementation.

Here the control parameters are quantized to a number of levels each denoted by a qualitative predicate. Therefore the control problem with continuous control parameters is simplified to a problem with finite set of problem operators.The behavior of the system then is given in the form of a matrix where the rows are the landmarks and the columns are the distinguished time points that the decision has happened. Both the time points and the landmarks are unknown in advance. There is only one nonzero element in each column of the matrix.

:					
L3			1	1	
L2	1				1
L1		1			
	T1	T2	T3	T4	...

1: If the landmark value is present;
0: Otherwise;

Suppose that again the scenario is composed of a mobile robot and a garage. The robot is assigned to enter the garage satisfactorily no matter where it is left initially. During a sample off-line run we can get a set of distinguished time points and their corresponding landmarks {L}. This behavior serves as the reference for further use. Suppose that now for another initial setting the behavior is given with the landmark set {L'}. If we can find a way to transform {L'} to {L} then there is no need to initiate the path planning and collision avoidance algorithm for the novel situation. The dynamical transformation of the landmarks can be obtained by the theory of discrete iteration [16]. As the set {L} is finite, all of the iterative processes finally end up as a cycle or even a fixed point [16]. This assures the stability of the transformation and accessibility of any other initial condition from the setting at the outset.

4. EXAMPLE

In this section we use the above mentioned methods to predict the collision, avoid it and path planning for a tri-cycle mobile robot. The approach described in this paper has been fully implemented in C programming language on PC-98 compatible personal computers to simulate the behavior. The first part of the program is devoted to predict the possible collision and the second part is composed of a number of problem specific rules for collision avoidance. These rules are quite a few and depend on the nature of the problem. We have developed the original version assuming the robot as a dimensionless point. In the second version we have extended it to four points encompassing the whole body of the mobile robot in rectangular form.

As an example let's think about a mobile robot assigned to enter a garage. This is a simple but intuitive example. Fig.5 shows the scene. For the first approximation we assume the robot as a dimensionless point and furthermore, suppose that the robot on each instant of time is provided with the proper sensory equipment that it can recognize the sight angle between the two edges of the gate. One of the novel points here is that the robot takes as the input data the angular information rather than the spatial. Angular measurement requires less sophisticated sensors. In formulating the problem we have adopted the regional strategy, that is, partitioning the scene into some defined regions with special properties and leading the robot through the regions respectively. The regions are of rectangular shape with perpendicular walls (surfaces) and right angle corners (points). The surfaces are static and either soft or hard. Soft surfaces are those imaginary ones and the robot is allowed to pass through them. While passing through a soft surface, only the qualitative equations representing the situation are changed. Hard surfaces are assumed to represent the real obstacles and facing a hard surface is synonym to collision. Points all are supposed to be hard.

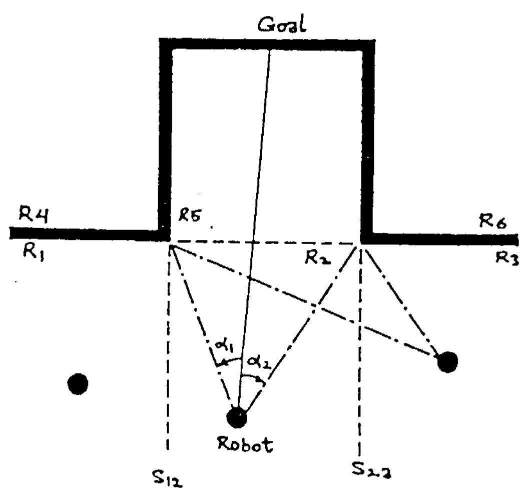

Fig. 5: Mobile robot and the garage

The required input data in our method are:

1. The present sight angle of the robot in a scene of disjunctive regions;
2. The present direction of movement of the robot;

Based on which the robot can keep track of the location of static obstacles by means of measuring and comparing the sight angle with respect to each obstacle. In this example the interesting regions are R_1, R_2, R_3 and R_5. For this representation we can deduce the following propositions:

1. The critical points are those that the sight angles become either zero or 90° . These are taken as the state transition points.
2. The pass or collision with a surface happens if the deviation angle from the surface perpendicular to it becomes zero;
3. The robot is supposed to persist on its present direction of movement until one of the state transition points is found (i.e., it is going to pass a soft surface or collide with a hard surface or a point).
4. If the colliding surface is soft (like S_{12} in region R_1), it is interpreted as acceptable and during passing a soft surface, only the new status of the region (i.e., qualitative formulas representing the situation) and the regional reconfiguration (new environmental conditions such as new surfaces and points) should be reconsidered.
5. If the colliding surface is hard (like S_{14} in region R_1), the control strategy and set points are to be adjusted.
6. When predicting a possible collision with an obstacle, the proper control strategy is selected as either:
   a. DIRECTION: Changes of direction of movement.
   b. SPEED: Changes of speed.
   Here the direction strategy is used to avoid the collision.
7. The set points of the control variables are decided based on the some heuristic rules. The rules governing this decision are quite a few and specific to the problem domain. Table (1) shows typical rules.

Fig. 6 shows a typical simulation result. Here the only control variable is ω and it is quantized and the direction of movement can be changed gradually every 20 degrees.

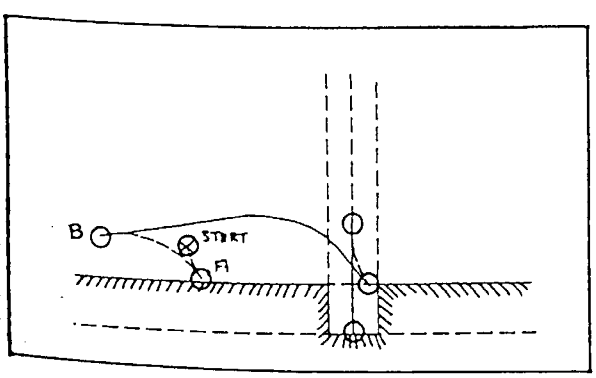

Fig. 6: Typical simulation result

This simple specification can be extended to encounter the followings:

1. Considering the two dimensional body for the robot:
   Fig. 7 shows the simulation results for the robot represented by four-points. The key point here is that as the body is rigid, on each step the same algorithm is run for the four points separately and then the critical one is identified. finally the adjustment action is hold for the critical point.
2. Entering backwards:
   As there is no discrimination among the four points, therefore the mobile robot can enter the garage either forward or backward. Fig.7 shows the robot using both forward and backward strategies to enter the garage.
3. Leaving the present status and trying a new one:
   As mentioned before, our method offers enough flexibility to encounter the heuristic rules implemented in the problem oriented knowledge base. Table (1) shows the two typical rules which govern the next direction of movement, and also moving the robot backward in order to avoid the collision in the four point simulation. Fig.6 shows the typical result of exploiting such rules. It is visible that the robot moves forward till point A, and then marshes backward until point B (dashed line). On the next step it uses the direction adjustment rule to approach the goal once more. After a second trial, it gets to the goal.

```
for (;;)
 {
     go=1;              /*  前進なら 1, 後退 なら -1  */
     ap=0;              /*  +方向に増加なら 1  */
     am=0;              /*  -方向に増加なら 1  */
     for (j=0; j<=3; j++)
     {
          if (ang_ptr->j_q1[j]<=0)
               ap=1;
          if (ang_ptr->j_q2[j]>=0)
               am=1;
          if (dis_ptr->r_dis [j]< dz && cal_ ptr->theta[j]<0)
          {
               kr=1;
               go=-1;
          }
          if (dis_ptr->l_dis [j]< dz && cal_ ptr->theta[j]>0)
          {
               kl=1;
               go=-1;
          }
     }
     if (ap && am)
          ap=am=0;
     if (kr && kl)
          kr=kl=0;
     if (kr)
     {
          for (j=0; j<=3; j ++)
          {
               if (dis_ptr->l_dis[j] <dz)
                    kr=0;
          }
     }
     if (kl)
     {
          for (j=0; j<=3; j ++)
          {
               if (dis_ptr->r_dis[j] <dz)
                    kl=0;
          }
     }
     if (ap || kr)
     {
          da_ptr->alphad = da_ptr->alphad + pi/180*20;
          da_ptr->alphad = judge_ang (da_ptr->alphad);
     }
     if (am || kl)
     {
          da_ptr->alphad = da_ptr->alphad - pi/180*20;
          da_ptr->alphad = judge_ang (da_ptr->alphad);
     }
     if (go == -1)
          vz = (-vz);
 }
```

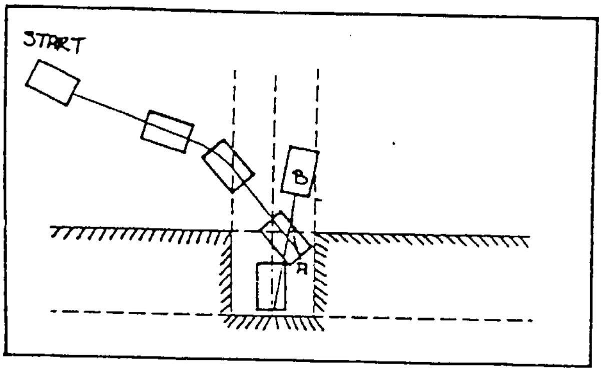

Fig. 7: Typical four point simulation result.

5. CONCLUSION

A method for predicting the collision of mobile robots based on nonmonotonic temporal logic was proposed. In this method we could blend the qualitative inferential technique and nonmonotonic temporal logic by representing the robot's behavior with an extended state machine.

The on-line requirement, is met when the initial conditions are changed. In this case instead of running the goal seeking program again, we could produce the satisfactory results by means of discrete iteration theory.

The above techniques are implemented in a program. In this framework, opposite to the conventional methods, sudden changes of the direction, leaving the present path and exploiting another one, etc., can be included.

REFERENCES

[1] Allen, J.F. : Towards a General Theory of Action and Time: AI, Vol. 23, pp. 123-154 (1984).
[2] Anzai, Y.: Cognitive Control of Real-Time Event Driven Systems, Cognitive Science, Vol. 8, pp. 221-254 (1984).
[3] Chellas, B.: Modal Logic: An Introduction, Cambridge University Press (1980).
[4] DeKleer, J. & Brown, J.S.: A Qualitative Physics Based On Confluences, AI, Vol. 24, pp. 7-83 (1984).
[5] DeKleer, J. & Brown, J.S.: Theories Of Causal Ordering, AI, Vol. 29, pp. 33-61 (1986).
[6] Far, B.H.: Qualitative Control: Qualitative Design & Verification of Industrial Controllers, Proc. MIV-89 Conf., pp. 239-244, (1989).
[7] Forbus, K.D.: Qualitative Process Theory, AI, Vol. 24, pp. 85-168 (1984).
[8] Iwasaki, Y. & Simon, H.A.: Causality In Device Behavior, AI, Vol. 29, pp. 3-32 (1986).
[9] Kuipers, B.: Qualitative Simulation, AI, Vol. 29, pp. 289-338 (1986).
[10] McDermott, D.: A Temporal Logic for Reasoning About Processes & Plans, Cognitive Science, Vol. 6, pp. 101-155 (1982).
[11] Ostroff, J.S. & Wonham, W.M.: State Machines, Temporal Logic and Control: A Framework For Discrete Event Systems, Proc. 26th Conf. on Dec. and Cont., pp. 681-686, Los Angeles, CA (1987).
[12] Ramadge, P.J. & Wonham, W.M.: Supervisory Control of a Class of Discrete Event Processes, SIAM J. Control & Optimization, Vol. 25, No. 1, pp. 206-220 (1987).
[13] Shoham, Y. & McDermott, D.: Problems in Formal Temporal Reasoning: AI, Vol. 36, pp. 49-61 (1988).
[14] Shoham, Y.: Chronological Ignorance: Experiments in Nonmonotonic Temporal Reasoning, AI, Vol. 36, pp. 279-331 (1988).
[15] Shoham, Y.: Temporal Logic in AI: Semantical & Ontological Considerations, AI, Vol. 33, pp. 89-104, (1987).
[16] Soulie, F.F., Robert, Y. & Tchuente, M. (Ed.): Automata Networks in Computer Science, Manchester University Press (1987).