SPEED PLANNING AND GENERATION APPROACH BASED ON THE PATH-TIME SPACE FOR MOBILE ROBOTS.

V. Muñoz, A. Cruz and A. García-Cerezo

Dpto. De Ingeniería de Sistemas y Automática. Universidad de Málaga. Plaza el Ejido s/n, 29013 Málaga (Spain).
Phone: (+34) 5 213-14-06; Fax: (+34) 5 213-14-13; E-mail: victor@ctima.uma.es

Abstract.

This paper presents a speed planning and generation algorithm for mobile robots that work under some kind of speed limitations. The proposed method takes the information about these speed constraints and the moving obstacles in the environment, and provides a safe speed profile which allows to build a trajectory that bypasses such obstacles. The method has been successfully tested in the RAM-2 mobile robot.

1. INTRODUCTION

The control motion system of a mobile robot drives its course by using the position and speed references (trajectory) computed by the planning system. Hence, this trajectory should be defined in such a way that provides a safe navigation with no collisions. Therefore, the obstacle avoidance is defined as the main function of the planning process.

The approach used for performing the above process, depends on the movement state of the obstacle. Some methods contemplate the obstacle's size and trajectory as entries to the local planner algorithm. Classic planning strategies can be adapted for this problem, such as visibility graphs [4][2], configuration space schemes [1][3] and potential fields [5].

The path-velocity decomposition is an efficient method for avoiding mobile obstacles by computing a safe speed function [4]. However, this methodology must be modified for providing the following features:

- First order continuity: The speed function and its first derivative should be continuous in order to reduce the speed tracking error.
- Admissibility from the point of view of the vehicle's kinematic and dynamic behaviour: The speed reference must be defined in such a way that allows the tracking system to follow the trajectory. Therefore, the kinematic and dynamic models of the robot must be taken into account in the speed function definition.

The method for avoiding moving obstacles, proposed in this paper, covers the above points by using piecewise cubic function, which is defined by studying other kind of speed restriction imposed by the vehicle's physical characteristics.

These limitations are commented in section 2. Section 3 is devoted to the previous work where the method presented in this paper leans on [7]. This work involves two stages: a speed planning process, which provides a speed plan that takes into account physical and operational speed limits, and a speed profile generation process, which builds the speed function needed to obtain the vehicle's trajectory. Section 4 details how the proposed method copes with the avoidance of moving obstacles. The speed planning made considering only kinematic and dynamic constraints is combined with the information about the mobile obstacles. In this way, a set of safety zones representing that speed planning, is built. Whenever a hazardous situation is detected, the original planning is modified in order to get a safe navigation. Implementation and experiments on the RAM-2 mobile robot are showed is section 5, and finally, section 6 presents the conclusions of this work.

2. SPEED PLANNING PROBLEM.

A robot path is defined as a set of evenly spaced postures $Q=\{q_1,...,q_m\}$, which are to be executed by the path tracking algorithm. A posture q_i is composed of five basic elements: x_i, y_i, θ_i, κ_i and s_i. The first two elements are position components, the third is the heading with respect to a global work frame, the fourth is the curvature component, and the last one is the distance along the path from the starting posture to the current one.

In order to convert a path Q into a trajectory $\tilde{Q}$ it is necessary to append a speed component to each posture of the path. In other words, the trajectory conversion process must turn each $q_i=(x_i,y_i,\theta_i,\kappa_i,s_i)$ into $\tilde{q}_i=(x_i,y_i,\theta_i,\kappa_i,s_i,v_i)$, where v_i is the posture speed component. This transformation is made by the definition of a parametric arc length speed function V(s). Such a curve is defined in the space-speed plane [10], where the upper speed limits for each posture q_i of the path Q are represented. These limits are obtained by taking into account the speed constraints introduced by the vehicle's features and operational speed limitations. Thus, V(s) is

specified in such a way that it preserves all the posture speed limits, in order to obtain a speed profile with good tracking conditions. That means that V(s) must lie inside a safety area of the space-speed plane defined by the speed limits functions (see Fig. 1.).

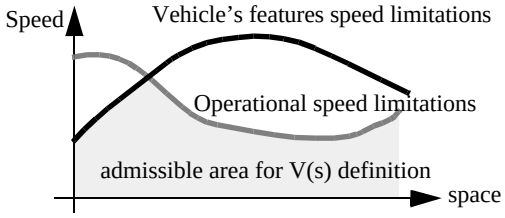

Fig. 1. Speed limitations in the space-speed plane.

The speed constraints considered at speed planning time are shown in table 1.

Table 1: Speed limits classification

Type	Constraint
Physical	Mechanical (ME)
	Kinematics (KI)
	Dynamic (DY)
Operational	Goal Point (DG)
	Known mobile obstacles (KM)

The physical speed restrictions group is due to the kinematic and dynamic behaviour of the mobile robot [9]. These restrictions impose a top speed and acceleration according to the pecularities of the vehicle and the path to be followed.

The second group presents the external speed limitations, which arise because of performing the task in a real environment. In this way, some situations force to set a safe speed value to the vehicle in order to synchronize with other elements in the working environment, or even to stop before getting into a zone with collision hazard.

3. PREVIOUS WORK: SPEED FUNCTION V(s).

The speed function V(s) definition is made in two steps: speed planner and speed generation processes.

3.1. Speed planner process.

This stage chooses a set of control path postures $C=\{q^1, ..., q^p\}$ which divides the path into a set $S=\{S_1,..., S_{p-1}\}$ of path segments, where S_i is composed of the path postures sequence between q^i and q^{i+1}. The choice of the elements which will be belong to C is made depending on the nature of the speed limitations:

- Kinematic and dynamic considerations are a function of the path curvature κ. Therefore, it is necessary to consider the variation law of this magnitude. In this way, the postures which have the local maximum or minimum curvature values are added to C set.
- Secondly, new path postures are chosen in order to satisfy the operational speed limitations. The closest posture to the goal point or known obstacle which satisfies the *distance to a goal point* speed limitation is selected.

A top speed v_i is assigned to each member q^i of C by using the minimum speed value provided by the speed limitations introduced by vehicle's features and operational speed constraints. This operation sets up a speed control set $V=\{0, v_1,...,v_{p-1}, 0\}$ for the path Q, whose first component (always null) is the starting speed for S_1 and the remaining components v_i are speed boundary conditions between segments S_{i-1} and S_i. Sets S and V resulting from the speed planning process, are represented in the space-speed plane, and will be used by the speed profile generator process for building V(s).

3.2. Speed generation process.

The speed function V(s) is modelled as a set of space-time functions $\sigma_i(t)$, as it is shown in expression (1).

$$V(s) = \sum_{i=1}^{p} \frac{d}{dt}\sigma_i(\sigma_i^{-1}(s)) \qquad (1)$$

The i^{th} component of this curve is assigned to the path segment S_i defined by the path postures sequence $\{q^i,...,q^{i+1}\}$. Function $\sigma_i(t)$ is determined by the parameters set $\Pi_i=\{v_i,v_{i+1},s_i,t_i\}$, where v_i and v_{i+1}, components of the set V, are the starting speed at q^i and the ending one at q^{i+1}; s_i is the segment length, and finally t_i is the navigation time assigned to the current segment which must be computed for $\sigma_i(t)$ definition.

The method evaluates $\sigma_i(t)$ by using two cubic polynomials ${}^1\sigma_i(t)$ and ${}^2\sigma_i(t)$, which are obtained from $\sigma_i(t)$:

$$\begin{aligned} {}^1\sigma_i(t) &= {}^1\alpha_{i0}t^3 + {}^1\alpha_{i1}t^2 + {}^1\alpha_{i2}t + {}^1\alpha_{i3} \\ {}^2\sigma_i(t) &= {}^2\alpha_{i0}t^3 + {}^2\alpha_{i1}t^2 + {}^2\alpha_{i2}t + {}^2\alpha_{i3} \end{aligned} \qquad (2)$$

Function ${}^1\sigma_i(t)$, with arc length ${}^1s_i$, covers the first part of the segment , and ${}^2\sigma_i(t)$ the remaining length of S_i (s_i-${}^1s_i$). In this way, by taking ${}^it_m$ as $t_i/2$ and ${}^iv_m$ as the average between v_i and v_{i+1}, the first function ${}^1\sigma_i(t)$ is defined by the set $\{v_i,{}^iv_m,{}^1s_i,{}^it_m\}$, and the second one ${}^2\sigma_i(t)$ by $\{{}^iv_m,v_{i+1},{}^2s_i,{}^it_m\}$. Moreover, the shape detailed at expression (3) is imposed to their second derivatives.

$$ {}^1\sigma''_i(0) = {}^2\sigma''_i({}^it_m) = 0 \qquad {}^1\sigma''_i({}^it_m) = {}^2\sigma''_i(0) \tag{3} $$

These functions are detailed in matrix form as follows:

$$ M \times \begin{bmatrix} {}^1\alpha_{i0} \\ {}^1\alpha_{i1} \\ {}^1\alpha_{i2} \\ {}^1\alpha_{i3} \end{bmatrix} = \begin{bmatrix} 0 \\ {}^1s_i \\ v_i \\ {}^iv_m \end{bmatrix} \qquad M \times \begin{bmatrix} {}^2\alpha_{i0} \\ {}^2\alpha_{i1} \\ {}^2\alpha_{i2} \\ {}^2\alpha_{i3} \end{bmatrix} = \begin{bmatrix} {}^1s_i \\ {}^2s_i \\ {}^iv_m \\ v_{i+1} \end{bmatrix} \tag{4} $$

where M is defined by the expression:

$$ M = \begin{bmatrix} 0 & 0 & 0 & 1 \\ ({}^it_m)^3 & ({}^it_m)^2 & {}^it_m & 1 \\ 0 & 0 & 1 & 0 \\ 3({}^it_m)^2 & 2({}^it_m) & 1 & 0 \end{bmatrix} \tag{5} $$

The values for the elements of these sets are:

$$ {}^it_m = \frac{s_i}{v_i + v_{i+1}} \qquad {}^iv_m = \frac{v_i + v_{i+1}}{2} $$
$$ {}^1s_i = \frac{s_i(5v_i + v_{i+1})}{6(v_i + v_{i+1})} \qquad {}^2s_i = s_i - {}^1s_i \tag{6} $$

Thus, the relationship between the starting and ending speeds for keeping the a given top acceleration constraint a_{max} is defined by the following expression:

$$ \text{if } (v_i < v_{i+1}) \text{ then} $$
$$ v_{i+1} \le \sqrt{v_i^2 + a_{max}s_i}) \qquad \text{else} (v_{i+1} \ge \sqrt{v_i^2 - a_{max}s_i}) \tag{7} $$

Finally, when the parameters sets Π_i have been determined via expressions (6) and (7), the speed profile V(s) is defined in the space-speed plane by using expression (1).

4. MOVING OBSTACLE AVOIDANCE METHOD.

The above speed planning V(s) is made in the speed-space phase plane, and it must be mapped into the path-time $s \times t$ space in order to detect the collisions with known mobile obstacles. The $s \times t$ space contains a set of rectangular forbidden regions, which represents the crossing points of the mobile obstacles trajectories through the robot path Q [4].

The speed planning is rendered in this space implicitly, by means of the space time functions $\sigma_i(t)$ (from i=1 to p). As it was stated in the previous section, every $\sigma_i(t)$ is defined by its parameters set $\Pi_i=\{v_i,v_{i+1},s_i,t_i\}$. Therefore, each $\sigma_i(t)$ is represented in the $s \times t$ by using its (s_i,t_i) elements. Furthermore, expressions at (6) show the relationship between these elements and the initial and final speed values of the segment (v_i and v_{i+1}).

In this way, the computed speed planning is safe when any $\sigma_i(t)$ does not touch any rectangular region. Let OB_j be a moving obstacle in the $s \times t$ space. The planned speed reference provides the maximum speed allowed by the vehicle's kinematic and dynamic features. Therefore, for avoiding the collision with OB_j, the robot must reduce its speed and let the moving obstacle cross the collision path point before the vehicle reaches it.

However, testing the collisions for every $\sigma_i(t)$ and all OB_j is a time-expensive process. Instead of this, the proposed method uses a set of hull-convex areas, called the safety zones Z_i, which encloses every $\sigma_i(t)$. Therefore, the proposed algorithm checks the collisions between every Z_i and OB_j, by using a geometric approach. Whenever this situation is detected, both $\sigma_i(t)$ and Z_i will be modified properly in the $s \times t$ space for avoiding the collision.

The building and modification of the safety zones will be developed in the following subsections.

4.1. Safety Zones

The building of the safety zone Z_i assigned to $\sigma_i(t)$ is based on the following lemma:

Lemma 1: *The shape of the space-time function $\sigma_i(t)$ assigned to S_i is always either concave or convex. In other words, both ${}^1\sigma_i(t)$ and ${}^2\sigma_i(t)$ must be, simultaneously, concave or convex.*

Proof:

Let ${}^1\sigma_i(t)$ and ${}^2\sigma_i(t)$ be the components of $\sigma_i(t)$. Their second derivative are defined in the following expression:

$$ {}^1\sigma''_i(t) = 6\,{}^1\alpha_{i0}t $$
$$ {}^2\sigma''_i(t) = 6\,{}^2\alpha_{i0}t + 2\,{}^2\alpha_{i1} \tag{8} $$

where ${}^1\alpha_{i0}$ is obtained by solving the expression (4):

$$ {}^1\alpha_{i0} = \frac{-(v_i - v_{i+1})(v_i + v_{i+1})^2}{s_i^2} \tag{9} $$

In the same way,:

$$ {}^2\alpha_{i0} = \frac{(v_i - v_{i+1})(v_i + v_{i+1})^2}{6s_i^2} $$
$$ {}^2\alpha_{i1} = \frac{-(v_i - v_{i+1})(v_i + v_{i+1})}{2s_i} \tag{10} $$

Concavity or convexity of these functions are studied by taking into account the sign of their second derivatives, which depends on the relationship between the starting and ending speed of S_i.

A increasing speed segment ($v_i<v_{i+1}$) implies that functions ${}^1\sigma_i(t)$ and ${}^2\sigma_i(t)$ are concave. On the other hand a decreasing speed segment ($v_i>v_{i+1}$) forces that derivatives are negative, and both subsegments are convex.

By taking into account the above lemma, the safety zone Z_i can be modelled as a triangle in the $s \times t$ space as it is shown in Fig. 2.

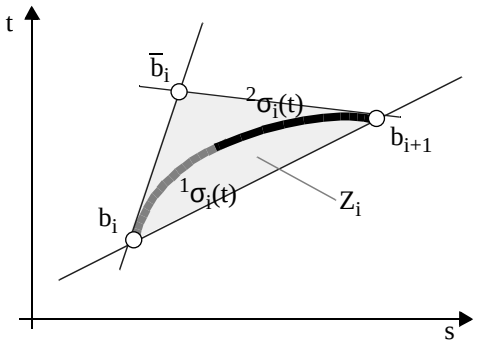

Fig. 2. The safety zone Z_i assigned to $\sigma_i(t)$.

This figure presents the safety zone as a shady triangle defined by the following lines:

- Line segment from the starting point b_i to the ending point b_{i+1}, which are defined as follows:

$$b_n = (gs_n, gt_n) = \left(\sum_{j=1}^{n} s_j, \sum_{j=1}^{n} t_j\right) \quad ;(s_j, t_j) \in \Pi_j \qquad (11)$$

- Tangent line to $^1\sigma_i(t)$ at b_i (slope=v_i).
- Tangent line to $^2\sigma_i(t)$ at b_{i+1}(slope=v_{i+1}).

So, Z_i is defined by b_i, b_{i+1} and the intersection point $\overline{b}_i$ between the tangent lines.

4.2. Safety zones modification.

As it was stated before, the robot must decrease its speed when a collision between a safety zone Z_j and a forbidden region OB_i is detected. In this situation, the path segment S_j corresponding to Z_j is replaced with two new path segments SA_j and SB_j, in such a way that the obstacle OB_i is avoided by using the path-time point b_k (see Fig. 3.).

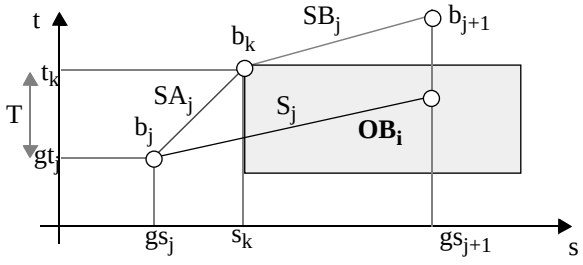

Fig. 3. Collision situation and path segment modification.

Let v_k be the speed reference at b_k. Therefore, the sets of path segments S and speed control set V are updated as shown in expression (12).

$$S = \{S_1, \ldots, S_{j-1}, SA_j, SB_j, S_{j+1}, \ldots, S_p\}$$
$$V = \{0, v_1, \ldots, v_j, v_k, v_{j+1}, \ldots, v_{p-1}, 0\} \qquad (12)$$

The parameters sets for SA_j and SB_j are defined by $\Pi_j^A = \{v_j, v_k, s_j^A, t_j^A\}$ and $\Pi_j^B = \{v_k, v_{j+1}, s_j^B, t_j^B\}$ where v_j and v_{j+1} are the planned speed references at b_j and b_{j+1}. In this way, the value for v_k is computed in order to travel path segment SA_j with a navigation time by using time expression at (6):

$$v_k = \frac{s_j^A}{T} - v_j \qquad (13)$$

where $T=(t_k-gt_j)$. The direct consequence of inserting b_k with a speed reference v_k is the time displacement of point b_{j+1} in the $s \times t$ plane.

The speed components v_k and v_j for SA_j could not assure the acceleration constraint. Therefore, it is necessary to check this condition by using expression (7), and modify v_k when it is needed. If this new value for v_k does not guarantee a navigation time t_j^A for SA_j greater than avoidance time T, the current obstacle is not bypassed.

The correction of this fact is made via the iterative modification of the speed references for the previous segments. This operation is made by the back propagation of the time needed to avoid the obstacle which keeps the acceleration constraint.

This process is detailed in *AvoidMobileObstacle* algorithm which is composed of two different parts. The first one modifies set S by inserting the new path segments and computes the reference speed value v_k. Furthermore, this part calculates the time ΔT as the difference between the time T needed for avoiding the obstacle OB_i and the real time t_j^A for travelling SA_j imposed by the acceleration constraint. The second part (while loop) performs the time back propagation if it is needed.

Function AvoidMobileObstacle(S,V,OB_i, j)
select S_j as the j^{th} element of S.
Divide S_j into SA_j and SB_j and update set S.
Compute v_k by using expression (13) and insert in set V
{V}=SpeedVerification(S,V)
Compute navigation time t_j^A for SA_j with formula (6).
$\Delta T=T- t_j^A$.
while $\Delta T>0$ do
Compute navigation time t_{j-1} for S_{j-1} with formula (6)
required_time=$\Delta T+t_{j-1}$
Compute v_j by using expression (13) with required_time.
{V}=SpeedVerification(S,V)
Compute navigation time t_{j-1} for S_{j-1} with formula (6).
ΔT=required_time-t_{j-1}.
j=j-1
endWhile
return S and V
endProcedure

4.3. Final algorithm.

The complete algorithm integrates both the speed planner and generation processes with the presented mobile obstacle avoidance method. The algorithm is divided into three stages: the speed planning process, the mobile obstacle avoidance and the speed profile generation.

The first step performs the speed planning, which involves the following operations. First, the path Q is divided into a set S of path segments and the speed control set V is computed. The components of this last set assure the acceleration constraint at expression (7).

The second stage is devoted to the moving obstacle avoidance. In this way, sets OB and Z are defined in the path-time space. After this, the algorithm tests the contacts between the elements of OB and Z. Whenever a collision is detected the sets S and V are modified for solving this situation. It is necessary to consider that avoiding a forbidden region OB_j might cause collisions between previous safety zones Z_k with k=0,...,i-1 and rectangles OB_l with l=0,...,j-1; so, the nested loops must be restarted if the function *AvoidMobileObstacle* modifies the previous speed planning.

Finally, the third stage builds the speed profile V(s) and merges this profile with path Q, in order to get the desired trajectory $\tilde{Q}$.

Procedure Path2Trajectory(Q,OT)
/ Speed planning process */*
$\tilde{Q}$ *={}*
{S}=DividePathIntoPathSegments(Q)
{V}=ComputeSpeedControlSet(S)

/ Mobile obstacle avoidance */*
{OB}=ComputeObstaclesSet(S,OT)
{Z}=ComputeSafetyZones(S,V)
For each S_i belonging to S do
For each OB_j belonging to OB do
if Collision(Z_i,OB_j) then
{V,S}=AvoidMobileObstacle(OB_j,S,V,i)
end if
end loop
end loop

/* Speed profile generation and trajectoty building */
For each S_i belonging to S do
Compute ${}^1\sigma_i(t)$ and ${}^2\sigma_i(t)$
$\sigma_i(t)$=*Composition(${}^1\sigma_i(t)$,${}^2\sigma_i(t)$)*
$V_i(s)=\sigma_i'(\sigma_i^{-1}(s))$
$\tilde{Q} = \tilde{Q} \quad \{S_i, V_i(s \quad S_i)\}$
end loop
End Procedure

---

5. IMPLEMENTATION AND EXPERIMENTS

The speed planning and profile generator algorithms have been integrated in the intelligent control architecture of RAM-2 [8].

Figures 4 to 6 show the results of the proposed method with the mobile robot RAM-2. The studied situation involves an environment with a moving obstacle. Figure 4 shows the path to be tracked by the robot in solid line and the trajectory followed by the obstacle in dotted line. Furthermore, the division of the path is marked with circular marks.

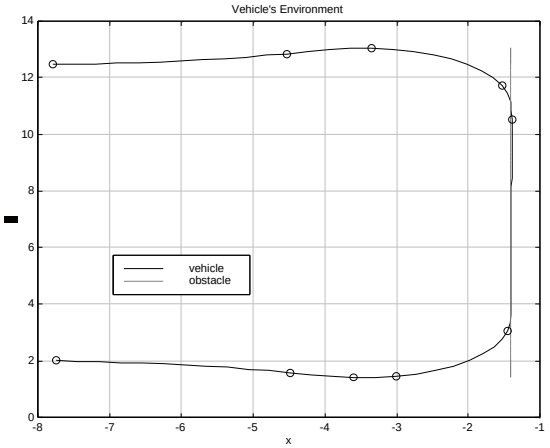

Fig. 4. Robot path and the moving obstacle.

The original speed planning provided by the speed planning process is represented, on space-speed plane, with a dotted line in figure 5. However, the moving obstacle crosses the robot path along its trajectory. This information is used for replanning the original speed profile, and obtaining a safe speed plannification for the mobile robot (solid line) which avoids the collision.

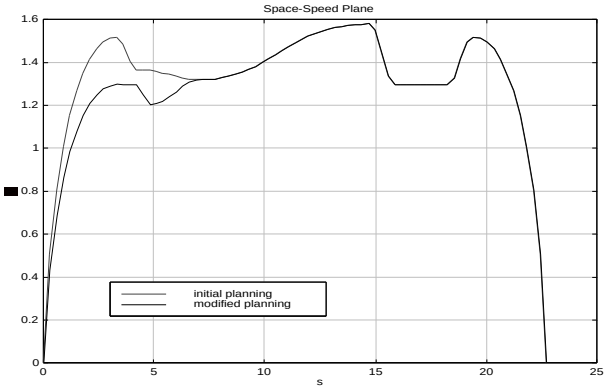

Fig. 5. Original speed planning and its modification for avoiding the mobile obstacle.

The above result is achieved via path-time representation (see Fig. 6.).

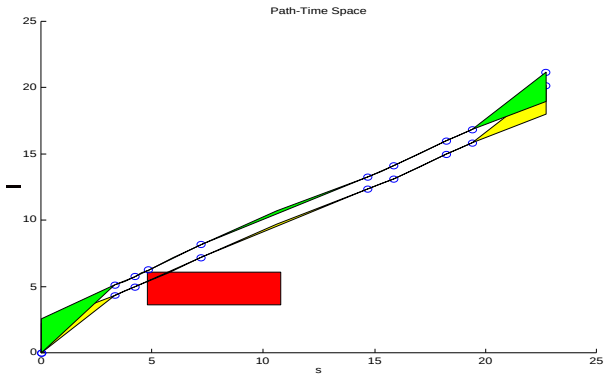

Fig. 6. Path-time representation.

This figure details the moving obstacle as a dark rectangle, and also contains the safety zones for every segment of the path as light shady triangles. The method detects the collision between the first safety zone and the obstacle, and eliminates this situation. The result of this action is shown as the set of dark shady triangles in Fig. 6., which includes the safety zone attached to the new segment needed to avoid the forbidden region. The new speed planning avoids the mobile obstacle and also considers the robot motion constraints due to its physical features.

6. CONCLUSIONS.

This paper presents a speed planning and generation algorithm for mobile robots that work under some kind of speed limitations. The proposed method takes the information about these speed constraints and the moving obstacles in its environment, and provides a safe speed profile which allows to build a trajectory that bypasses such obstacles. The algorithm applies three steps: i) speed planning, ii) mobile obstacles avoidance, and iii) speed profile generation.

The method modifies the speed planning when is required due to a collision situation. Moreover, it does not handle a visibility graph on the $s \times t$ space, and therefore does not use any kind of graph-search algorithm. The result is a speed profile which avoids the moving obstacles and takes into account other kind of speed constraints (i.e. kinematic and dynamic). This feature provides the real time execution for speed replanning which can be also made while the robot is tracking the path.

Finally, working in the presence of mobile obstacles demands a complete knowledge about them, i.e., their sizes and trajectories. Though this situation is usually real in multi-robot systems, the navigator must use a local speed planner which modifies the current speed reference using feedback control [6]. This action solves the uncertainties about the obstacle's trajectory, that the planner does not consider in planning time.

7. ACKNOWLEDGEMENTS

This work has been done within the framework of the projects TAP96-1184-C04-02 and TAP96-0763 of the C.I.C.Y.T. (Spain)

8. REFERENCES.

[1] Fujimura, K. and Samet, H. (1990) Motion planning in a Dynamic Domain. *Proc. IEEE Int. Conf. on Robotics & Automation*, pp. 324-330

[2] Gil de Lamadrid, J. (1994) Avoidance of Obstacles with Unknown Trajectories: Locally Optimal paths and Periodic Sensor Readings. *The International Journal of Robotics Research*, Vol 13,No 6, pp. 496-507.

[3] Griswold, N. C. and Eem, J. (1990) Control of Mobile Robots in the Presence of Moving Objects. IEEE Transactions on Robotics and Automation , Vol. 6, No. 2, pp. 263-268.

[4] Kant K., Zucker S. (1.986). Toward Efficient Trajectory Planning: The Path-Velocity Decomposition. The International Journal of Robotics Research, Vol 5, No. 3.

[5] Kyriakopoulos, K. J. and Saridis, G. N. (1993) An Integrated Collision Prediction and Avoidance Scheme for Mobile Robots in Non-Stationary Environments, Automatica, Vol. 29, No. 2

[6] Mandow A., Muñoz V., Fernandez R., García-Cerezo A. (1997). Dynamic Speed Planning for Safe Navigation. 1997 Proc. IEEE International Conference on Intelligent Robots and Systems.

[7] Muñoz V. (1995). Trajectory Planning for Mobile Robots. Ph. D. Thesis. University of Malaga (Spain).

[8] Ollero A., A. Simón, F. García, and V. E. Torres (1993). Integrated Mechanical Design of a New Mobile Robot. Proc. SICICA´92. Pergamon Press.

[9] Prado M., Simon A., Muñoz V., Ollero A. (1994). Autonomous Mobile Robot Dynamic Constraints due to Wheel-Ground Interaction. Proc. of 1994 European Robotics and Intelligent Systems Conference. Vol 1, pp 347-360. Malaga (Spain)

[10] Shiller Z., Gwo Y. (1991). Dynamic Planning of Autonomous Vehicles. IEEE Transactions on Robotics and Automation, Vol 7 No. 2.