

On the influence of sensor capacities and environment dynamics onto collision-free motion plans

R. Alami

T. Siméon

K. Madhava Krishna

LAAS-CNRS

7, avenue du Colonel-Roche
31077 Toulouse Cedex - France
{nic,rachid,mkrishna}@laas.fr

Abstract: *A methodology for computing the maximum velocity profile for a planned trajectory of the robot is described in this paper. The profile is computed considering the robot and environment dynamics as well as the constraints of the sensing apparatus. The mobile objects can be arbitrary in number and their direction and velocity of motion is not known. The only known information about the moving objects is the maximum velocity they can possess. The robot that moves with the computed velocity profile can assure from its side that it would not collide onto any of the numerous moving objects that could intercept its future trajectory. The methodology has been incorporated into a motion planner for a non holonomous robot and the results presented. The motivation here is to facilitate the process of having safe and understanding robots. Hence the planned velocity profiles are in general conservative though the robot could perhaps do better on-line. However at planning time the robot's immobility before collision is guaranteed.*

1 Introduction

Several strategies exist for planning collision free paths in an environment whose model is known [1]. However many such geometrically planned paths can result in failures during execution if the environment dynamics, parameters of the sensory apparatus and dynamics of the robot are not considered. Generally these factors crop up during execution of the planned paths and one may come up with an ad hoc reactive strategy to alleviate some of these difficulties. Such measures may or may not result in the robot achieving its objective and worse could lead to unpredictable or unforeseen consequences for the robot as well as for the environment. This is especially so if the robot needs to navigate in an environment consisting of humans.

This paper presents a novel pro-active strategy that incorporates robot and environment dynamics as well as

sensory constraints onto a collision free motion plan. By pro-active we mean that the robot is always in a state of expectation regarding the possibility of a mobile object impinging onto its path anywhere from those regions invisible to its sensor. This pro-active state is reflected in the velocity profile of the robot, which guarantees in the worst case scenario, the robot, from its side would not collide with any of the moving objects that can interfere with its path. In fact the ability of the algorithm to a-priori compute velocities for the entire trajectory accounting for moving objects moving in whatsoever direction is the essential novelty of this effort. In other words the robot plans a path and a maximum velocity profile on the path such that, if at any moment, a moving object enters the visibility field, the robot can decelerate to a halt before collision. A similar kind of strategy for the aforementioned objective does not appear to have been confronted in robotic literature so far.

Possible application of this work are in the areas of personal and service robots where the robot needs to move in a typical office environment whose model is available but where there could be frequent interceptions due to mobile objects such as humans onto the robot's path. These objects can emerge from behind doorways, passages and openings. The robot has to plan its velocity profile in a manner such that it guarantees that there would be no collision with the mobile objects from its side. One possibility could be to have a robot that moves very slowly with (without) a horn. However, with a little knowledge of environment dynamics the robot can move for a reasonable duration with a velocity in the vicinity of its maximum possible velocity.

Related work can be cited in the areas of modifying global plans using sensory data obtained during execution for overcoming uncertainty accumulated during motions [2] and those that try to bridge the gap be-

tween planning and uncertainty [3] or planning and control [4], [5]. Work in the area of pursuit evasion [6], [7] that deals with searching objects in an unexplored region whose model is available is of relevance here in relation to memorizing previously cognized scenarios, an issue, which we address briefly. The velocity obstacle concept [8][9] bears resemblance to the current endeavor in that they involve selection of a robot velocity that avoids any number of moving objects. The difference is that in the present approach the only information about the mobile object available is the bound on their velocity. The direction of motion and their actual velocities are not known during computation of the velocity profile.

2 Problem Definition

Given the following:

- an *a priori* known workspace cluttered with static obstacles
- mobile objects: their motion unpredictable, the only available information is their velocity bounds.
- a robot equipped with a sensor allowing to detect the mobile objects.

The paper addresses the two following problems:

Problem 1: Given a robot’s path $\tau(s)$ computed by a standard planner [1], determine the maximal velocity profile $v_\tau(s)$ such that, considering the constraints imposed by its dynamics, the robot can stop before a collision occurs with an unpredictable obstacle detected when the robot is at position $s \in [0, L]$ along path τ . For example the velocity profile dictates that the robot in figure 1 slow down near the doorway in expectation of mobile objects from the other side. We call $MP = (\tau(s), v_\tau(s))$ a **robust motion plan**. The velocity profile allows us to define the time $T(\tau)$ required for the robust execution of path τ :

$$T(\tau) = \int_0^L \frac{ds}{v_\tau(s)}$$

Problem 2: How to determine paths, collision-free wrt. the static obstacles of the workspace, and that optimize the criteria $T(\tau)$. For example the path of figure 2 is traversed in a shorter time though longer than figure 1

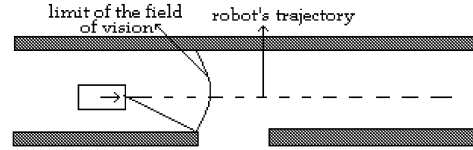


Figure 1: The robot slows down on nearing the doorway

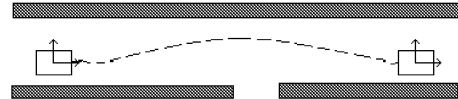


Figure 2: a longer path could be faster due to better velocities

3 From path to robust motion plan

We tackle the first problem by illustrating how the velocity profile can be computed under the following assumptions:

- The robot \mathcal{R} is modelled by a disc and is equipped with an omnidirectional sensor having a limited range R_{vis} . We call \mathcal{C}_{vis} the visibility circle, centered at a given robot’s position and with radius R_{vis} . The paths of \mathcal{R} are sequences of straight segments connected with tangential circular arcs of radius ρ . The robot’s motion is also subject to dynamic constraints simply modelled by a bounded linear velocity $v \in [0, v_M]$ and a bounded acceleration $a \in \{-a_m, a_M\}$.
- The workspace is cluttered by static polygonal obstacles \mathcal{O}_i . It may also contain several mobile obstacles whose motions are not predictable; the only information is their bounded velocity v_{obj} .

3.1 Velocity constraints

Consider that while the robot is executing a planned path τ , a mobile object may enter at any time in the sensor’s field of view. The problem is to determine under what condition the halting distance of the robot will be sufficient to avoid any risk of collision with the moving object.

The halting distance is directly related to the robot’s velocity v_{rob} : after it begins to decelerate (at time $t = 0$) with $a(t) = -a_m$, the velocity $v(t) = v_{rob} - a_m \cdot t$

becomes zero after a time $t_0 = v_{rob}/a_m$ and therefore the robot stops after a halting distance $d_{rob}(v) = v_{rob}^2/2a_m$. Two cases may occur depending on the presence (or not) of static obstacles that may create *shadows* inside \mathcal{C}_{vis} .

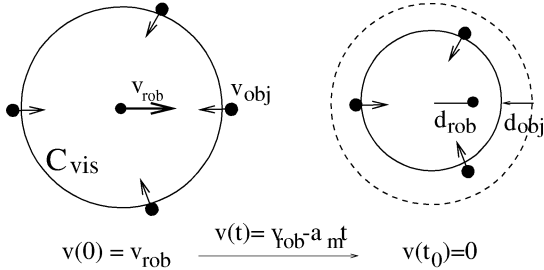


Figure 3: Mobile objects may appear anywhere onto the \mathcal{C}_{vis} 's contour.

No obstacles in \mathcal{C}_{vis} In the simple case where the robot's position is such that no static obstacle lies inside \mathcal{C}_{vis} , a moving object may appear (at time $t = 0$) anywhere onto \mathcal{C}_{vis} 's boundary (Fig. 3). At time t_0 (ie. when the robot is stopped), the distance crossed by the object is $d_{obj}(v) \leq v_{obj}v_{rob}/a_m$. Avoiding any potential collision imposes that $R_{vis} \geq d_{rob}(v) + d_{obj}(v)$. The condition relates the maximal robot's velocity v_{max} to the sensor's range R_{vis} :

$$v_{rob}^{max} = -v_{obj} + \sqrt{v_{obj}^2 + 2a_m R_{vis}} \quad (1)$$

Influence of shadowing corners Static obstacles lying inside \mathcal{C}_{vis} may create shadows (eg. see the grey region of Figure 4) susceptible to contain mobile objects. The worst-case situation occurs when the mobile remains unseen until it arrives at the *shadowing corner* of a polygonal obstacle. Since the mobile's motion direction is not known it is best modeled for a worst case scenario as an expanding circular wave of radius $v_{obj}t$ centered at (d, θ)

$$(X(t) - d \cos \theta)^2 + (Y(t) - d \sin \theta)^2 = v_{obj}^2 t^2$$

Let us first consider that the robot's path τ is a straight segment. Considering that the intersections between the circular wave and the robot's segment path, should never reach the robot before it stops at time t_0 yield to the following velocity constraint:

$$v_{rob}^4 - 4(a_m d \cos \theta + v_{obj}^2)v_{rob}^2 + 4a_m^2 d^2 \geq 0 \quad (2)$$

The solution of eq. 2 gives the maximal robot's velocity as a function of the shadowing corner's position

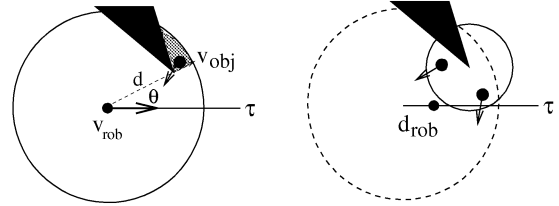


Figure 4: Mobile objects may also appear in the shadows of static obstacles

This solution only exists under the condition $v_{obj} > \sqrt{a_m d(1 - \cos \theta)}$, ie. when the object's velocity v_{obj} is sufficiently high to interfere with the robot's halting path. Otherwise, the shadowing corner does not constrain the robot's velocity which can be set to v_{rob}^{max} obtained from eq. 1 (which corresponds to the particular case $d = R_{vis}$ and $\theta = 0$).

A similar reasoning can be applied to the case where the robot traverses a circular arc path. This case however leads to a non linear equation that needs to be solved numerically to derive the maximal velocity [11]. The expression that needs to be solved for computing the maximum velocity at a given point on a circular arc is of the form

$$\begin{aligned} & ((v_{rob}^2 V_{obj}^2)/a_m^2) + 2\rho^2 \cos(V_{obj}^2/2a_m\rho) + \\ & 2d\rho \sin((V_{obj}^2/2a_m\rho) - \theta) \\ & = d^2 + 2\rho^2 - 2d\rho \sin \theta \end{aligned} \quad (3)$$

3.2 Computing the shadowing corners

We consider now the problem of determining the set of shadowing corners that was needed for the velocity computation in 3.1. The shadowing corners are extracted as those vertices of the polygonal obstacle to which a ray emitted from the robot's center is tangential (Figure 5). The set of shadowing corners can be easily extracted from an algorithm that outputs the visibility polygon [10] as a sorted list of vertices.

3.3 Computing $v(s)$ along a robot's path

Given a path $\tau(s)$ made of straight and arc segments (eg. Reeds and Shepp paths [12]), the velocity profile $v(s)$ is computed by considering discrete positions s_i along τ . For each position s_i , the set of shadowing corners SC_i is determined as explained in the previous section. In the case of an empty set, $v(s_i)$ is simply

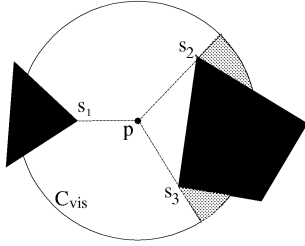


Figure 5: *Shadowing corners: among the three vertices of $V(p)$, only s_2 and s_3 create shadows (the line going through s_1 is not tangent to the left obstacle)*

set to the v_{rob}^{max} given by eq. 1. Otherwise, it is computed from the minimum between this value and the velocities resulting from eq. 2 applied to each corner of set SC_i . The velocities are further adjusted to respect the constraints imposed by robot dynamics and the boundary values, such as zero velocity at the start and ending locations on the path.

3.4 Memorization of Sensor Information

The computation of the velocity profile at a given point on the robot’s trajectory incorporates the robot’s field of vision at that point. This field can change appreciably between two successive instances of computation. For example in figure 6 the robot at position a has full field of vision of the corridor that is transverse to the robot’s trajectory. However at position b the robot is blind to the zone shown in darker shade of gray. Hence it needs to slow down its velocity as it moves further down to c since it envisages the possibility of a moving object approaching it from the corners of the stationary objects. These corners are the starting areas of the robot’s blind zone at b .

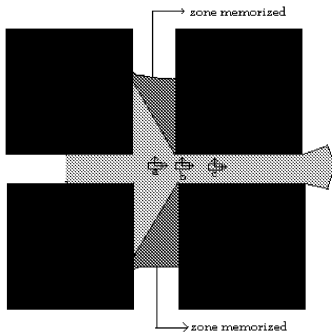


Figure 6: *Memorization of previous scenes*

However if the robot could memorize the scenario cog-

nized earlier it can retain this memorized image for computing its velocity profile during execution of the planned path. In such a case if the robot had not seen any moving objects in close proximity at a it can make use of this information at b to have a velocity profile from b that is greater than the one computed in the absence of such memorization. Fig 6 shows the zone memorized by the robot in darker shade of gray that can facilitate it to have a higher velocity than that computed at the planning stage. The details of the memorization scheme [11] are not mentioned here for brevity. It is worth mentioning nonetheless that the contour of the memorized area represents the blind-zone of the robot at that instant from where mobile objects can emanate. This memorized area overlaps exactly with the actual visibility polygon only during the instant of cognition. With the passage of time the frontier of the memorized area shrinks due to the advancement of the imagined mobiles from the initial frontier.

4 Implementation and Analysis

4.1 Illustration of the Motivations

The motivations of this paper are initially illustrated through a simple example. Figure 7 shows the initial trajectory computed by a non-holonomous planner with the path-length minimized. The robot traverses grazing the corridor as a consequence. Also shown in the figure 7, the robot’s field of vision as it nears the doorway. When environment and robot dynamics are considered the velocity profile of figure 8 entails that the robot decelerate near the doorway. The robot is equipped with a maximum acceleration/deceleration of $2m/s^2$ and a maximum velocity of $2.5m/s$. The maximum range of the sensors was fixed at $5m$ while the mobile objects were ascribed a velocity of $2m/s$. The figure also indicates through a vertical line the position of the robot corresponding to its position in figure 7 when it approaches the doorway. The marked dip in the velocity at that location is seen and the execution time of this path tallies to 21 seconds.

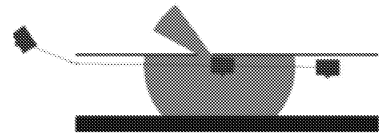


Figure 7: *Path computed by the non holonomous planner*

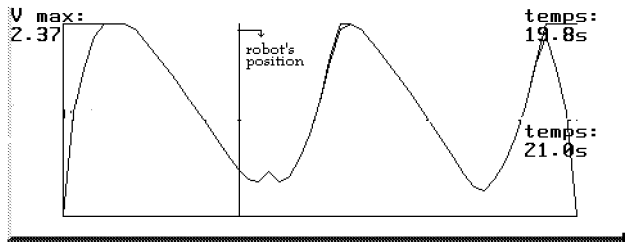


Figure 8: Velocity profile for the path of fig 7

Longer the Better: Minimizing time rather than path length Figure 9 shows the modified path obtained by random minimization of time rather than path length. Though longer than the path of figure 7, it needs an execution time of only 12.1 seconds when compared with 21 seconds in the previous case. The velocity profile of figure 10 shows no dips and the robot can accomplish the path avoiding the deceleration depicted in figure 8

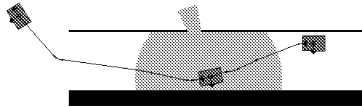


Figure 9: Path obtained after local minimization of time

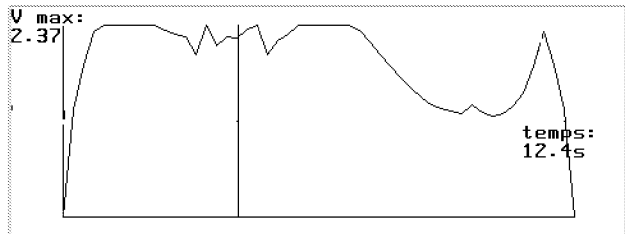


Figure 10: Velocity profile for the path of fig 9

4.2 Analysis of Memorization

Figure 11 shows the path obtained by minimizing time in an environment with four corridors marked 1, 2, 3 and 4. It also depicts the robot's field of vision as it enters the corridor 3. The velocity profile for the above path is shown in figure 12. The location of the robot corresponding to its location on figure 11 is shown through the vertical line. The locations of the robot as it decelerates when its field of view of each of the

corridor vanishes is also marked with the respective numbers on the profile.

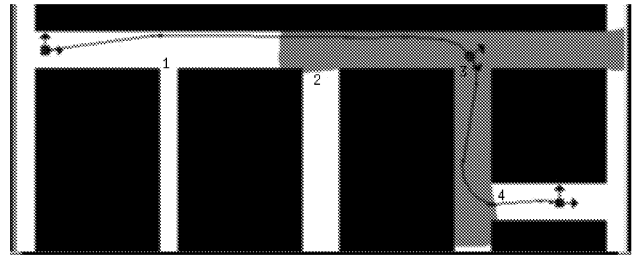


Figure 11: Robot's field of view as it enters corridor 3

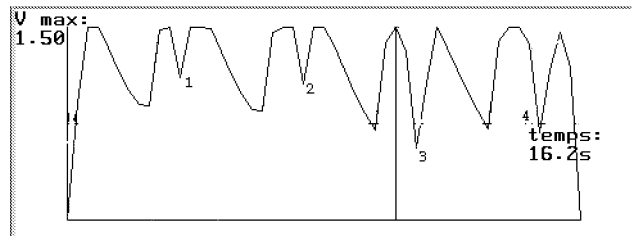


Figure 12: Velocity profile for the figure 11. Corresponding position of the robot shown in vertical line. Decelerations near the corridors are also marked with the same numbers

Though the path of figure 11 is minimized in time its velocity profile still shows decelerations in the vicinity of the corridors. This is due to the phenomenon discussed in section 3.4 where the robot becomes blind to many parts of the environment it had seen at the preceding instant. Figure 13 shows the robot's field of vision at an instant after the instance shown in figure 11. There is a marked decrease in its field of vision at the latter instant that results in robot reducing its velocity in anticipation of moving objects from the blind zones as depicted in the velocity profile. However when the robot is able to memorize its previous images the need to decelerate is nullified and the robot traverses the trajectory in lesser time. Figure 14 illustrates this where the decelerations shown in the velocity profile of figure 12 at locations 1, 2, 3 and 4 are now absent.

5 Conclusions and Possible Extensions

The paper has described a methodology by which given the robot trajectory $\tau(s)$, the maximal velocity profile $v_\tau(s)$ can be computed such that the robot can stop before colliding with any of the several dynamic objects that could be present and that move

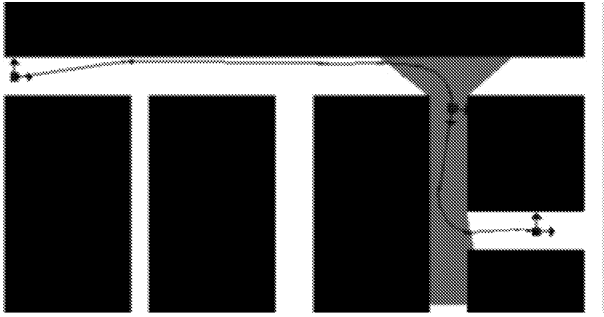


Figure 13: Robot's field of vision at an instant that immediately follows the instance of figure 11

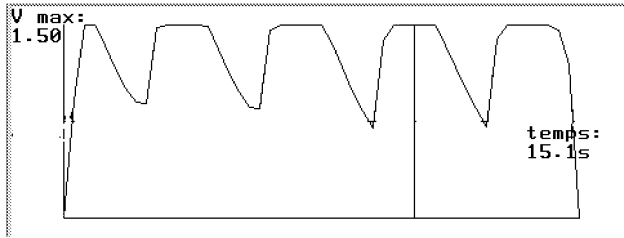


Figure 14: The velocity profile obtained after incorporation of memorization

along any possible direction. The paper also tries to address the problem of minimizing the trajectory time of the robot by randomly selecting the positions in the vicinity of the planned path and rebuilding the path through these positions if a reduced trajectory time is found. Results presented depict that the robot can have a velocity profile that achieves its maximum possible velocity for a sustained duration without many dips provided it stays away from doorways and narrow passages along its path. Memorization of previously cognized scenes also enhance the robot's performance through reduced trajectory time and a more uniform velocity profile. Immediate extension of the work involves implementing the methodology on a real robot along with an online scheme that modifies the velocity profile according to real-time perceptual data.

Acknowledgements We thank profoundly Didier Cruzel for implementing the various aspects of this work through the G2D interface.

References

[1] J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
 [2] B. Bouilly, T. Siméon, and R. Alami. A numerical technique for planning motion strategies of a mobile robot

in presence of uncertainty. In *IEEE International Conference on Robotics and Automation, Nagoya (Japan)*, 1995.
 [3] A. Lazanas and J.C. Latombe. Motion planning with uncertainty: a landmark approach. *Artificial Intelligence*, 76:287–317, 1995.
 [4] M. Khatib, B. Bouilly, T. Simeon, R. Chatila. Indoor Navigation with Uncertainty using Sensor-based Motions. *IEEE International Conference on Robotics and Automation, Albuquerque (USA)*, 1997.
 [5] J.C. Alvarez, A. Skhel, V. Lumelsky. Accounting for mobile robot dynamics in sensor-based motion planning: experimental results. *IEEE International Conference on Robotics and Automation, Leuven (Belgium)*, 1998.
 [6] B. Simov, S. M. LaValle, and G. Slutzki. A Complete Pursuit-Evasion Algorithm for Two Pursuers Using Beam Detection to appear in *IEEE International Conference on Robotics and Automation*, 2002
 [7] S. La Valle, D. Lin, L. Guibas, J.C. Latombe R. Motvani. Finding an unpredictable target in a workspace with obstacles. *IEEE International Conference on Robotics and Automation, Albuquerque (USA)*, 1997.
 [8] Z. Shiller, F. Large and S. Sekhavat. Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories. *IEEE International Conf. on Rob. Automat.*, 3716-3721, 2001.
 [9] P. Fiorini and Z. Schiller. Motion planning in dynamic environments using velocity obstacles *International Journal of Robotics Research*, 17(7):760-772, July 1998.
 [10] S. Suri and J. O'Rourke.. Worst-case optimal algorithms for constructing visibility polygons with holes. *ACM Symp. on Computational Geometry*, 1986..
 [11] D. Cruzel. Planification de mouvements sous contraintes de perception. *Master's report*, LAAS-CNRS, 1998.
 [12] J. A. Reeds and R. A. Shepp, "Optimal paths for a car that goes both forward and backwards," *Pacific Journal of Mathematics*, 145 (2), 1990.