

Provably Safe Navigation for Mobile Robots with Limited Field-of-Views in Unknown Dynamic Environments

Sara Bouraine, Thierry Fraichard, Hassen Salhi

▶ **To cite this version:**

Sara Bouraine, Thierry Fraichard, Hassen Salhi. Provably Safe Navigation for Mobile Robots with Limited Field-of-Views in Unknown Dynamic Environments. ICRA 2012 - IEEE International Conference on Robotics and Automation, May 2012, Saint Paul, MN, United States. pp.174-179, 10.1109/ICRA.2012.6224932 . hal-00768527

HAL Id: hal-00768527

https://inria.hal.science/hal-00768527

Submitted on 21 Dec 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Provably Safe Navigation for Mobile Robots with Limited Field-of-Views in Unknown Dynamic Environments

Sara Bouraine[†] and Thierry Fraichard[‡] and Hassen Salhi[⋆]

***Abstract*— This paper addresses the problem of navigating a mobile robot with a limited field-of-view in a unknown dynamic environment. In such a situation, *absolute motion safety*, *i.e.* such that no collision will ever take place whatever happens, is impossible to guarantee. It is therefore settled for a weaker level of motion safety dubbed *passive motion safety*: it guarantees that, if a collision takes place, the robot will be at rest. Passive motion safety is tackled using a variant of the Inevitable Collision State (ICS) concept called *Braking ICS*, *i.e.* states such that, whatever the future braking trajectory of the robot, a collision occurs before it is at rest. Passive motion safety is readily obtained by avoiding Braking ICS at all times. Building upon an existing Braking ICS-Checker, *i.e.* an algorithm that checks if a given state is a Braking ICS or not, this paper presents a reactive collision avoidance scheme called PASSAVOID. The main contribution of this paper is the *formal proof* of PASSAVOID's passive motion safety. Experiments in simulation demonstrates how PASSAVOID operates.**

I. INTRODUCTION

Robotics technology has matured and Autonomous Ground Vehicles are becoming a reality. However such systems remains prone to accidents (see [7]). The literature review of §II shows that the Robotics community is displaying a growing interest in designing navigation schemes for which motion safety can be characterized or even guaranteed (see the 2012 special issue of Autonomous Robots on Guaranteeing Motion Safety for Robots). It also shows that motion safety in the real world remains an open problem as soon as the term real world implies that:

1) The environment features both fixed and moving objects whose future behaviour is unknown.
2) The robot has only a partial knowledge of its surroundings because of its sensory limitations.

The purpose of this paper is precisely to address such problems. It can be argued that absolute motion safety is impossible to guarantee in general unless questionable assumptions concerning the robot and its environment are made, *e.g.* requiring that the velocity of the robot is a multiple of the maximum velocity of the objects [16], or that the moving objects should appear beyond a distance which is a function of their number, sizes and velocities [14]. To cope with that issue, the position taken in this work is: *better guarantee less than guarantee nothing*. To that end, it is settled for a weaker level of motion safety that guarantees that, if a collision takes place, the robot will be at rest. As per [17], this motion safety level is dubbed *passive motion safety*. As limited as it may appear at first sight, passive motion safety is interesting for two reasons: (1) it allows to provide at least one form of motion safety guarantee in the challenging scenarios considered and more important (2) if every moving object in the environment enforces it then no collision ever take place at all. The central idea behind passive motion safety, *i.e.* using braking trajectories, is not new, it has been used before in different contexts (see §II). However, to the best of the authors' knowledge, it is the first time it is given a formal treatment in as general a context as possible whether it concerns the robot's dynamics, its field-of-view, or the knowledge (or lack thereof) about the future behaviour of the moving objects.

Passive motion safety is tackled herein using a variant of the Inevitable Collision State (ICS) concept [10] called *Braking ICS*, *i.e.* states such that, whatever the future braking trajectory followed by the robot, a collision occurs before it is at rest. Passive motion safety is readily obtained by avoiding Braking ICS at all times. Braking ICS have been introduced by the authors of this paper in [3] along with a *Braking ICS-Checker*, *i.e.* an algorithm that determines whether a given state is a Braking ICS or not. To validate the Braking ICS concept and demonstrate its usefulness, the Braking ICS-Checker of [3] is integrated here in a reactive collision avoidance scheme (henceforth called PASSAVOID) for a mobile robot with a limited field-of-view placed in an unknown dynamic environment. It operates with a given time step and its purpose is to compute the control that will be applied to the robot at the next time step. The main contribution of this paper is the *formal proof* of PASSAVOID's passive motion safety: it is guaranteed that the robot will always avoid Braking ICS no matter what happens in the environment.

The paper is organized as follows: a review of the relevant literature is done in §II while the problem addressed is defined in §III. PASSAVOID is then detailed in §IV, the proof of its passive motion safety is established there along with the proof that if every moving object enforces it then no collision ever take place. Finally, experimental results obtained in simulation are presented in §V.

II. RELATED WORKS

As mentioned above, the Robotics literature is teeming with works concerned with collision avoidance but most of them do not offer an explicit formulation of the safety guarantees they provide or the conditions under which they must operate (see [9]). The earliest relevant works addressed the so-called "Asteroid Avoidance Problem": in 3D, [20] shows that collision avoidance is always possible if the

[†]CDTA (AL); [‡]INRIA (FR); ⋆Blida Univ. (AL).
Journal article [4] combines [3] and this paper.

robot's velocity is greater than the asteroids' velocities. In 2D, [14] shows that collision avoidance is always possible iff the asteroids appear beyond a "threat horizon". Likewise, [16] shows that, for a 2D robot among arbitrarily moving objects, collision-avoidance is guaranteed iff the maximum velocity of the robot is a multiple of the maximum velocity of the objects. Such results are very interesting. Unfortunately, they rely on assumptions that rarely occur in the real world. A related family of research works are those seeking to coordinate the motion of a set of robots. Different distributed coordination schemes have been proposed for which collision avoidance is guaranteed, *e.g.* [2], [15]. However, this guarantee is lost if the environment contains uncontrolled moving objects. General motion safety issues have been studied thanks to the *Inevitable Collision States* (ICS) concept developed in [10]. An ICS is a state for which, no matter what the future trajectory of the robot is, a collision eventually occurs. ICS provides insight into the complexity of guaranteeing motion safety since it shows that it requires to *reason about the future* evolution of the environment and to do so with an *appropriate lookahead*[1] that can possibly be infinite. Such conditions being next to impossible to obtain in the real world plus the fact that ICS characterization is very complex has led a number of authors to consider relaxations of ICS such as:

- *ICS approximation*, *e.g.* [13]: such approximations being not conservative, the motion safety guarantee is lost.
- τ*-Safety*, *e.g.* [11]: the robot is guaranteed to remain in states where it is safe for a given duration (hopefully sufficient to compute an updated safe trajectory...).
- *Evasive trajectories*, *e.g.* [12]: they guarantee that the robot can only be in states where it is possible to execute an evasive trajectory, *e.g.* a braking manoeuvre for a car or a circling manoeuvre for a plane.

Recently, authors have proposed probabilistic versions of the ICS concept, *e.g.* [1], so as to better capture the uncertainty that prevails in real world situations. These approaches are interesting but offer no strict motion safety guarantees since probabilistic models are used. There are a few research works taking into account sensory limitations. For instance, the occlusion problem, *i.e.* the existence of regions that are hidden by other objects, is addressed in [21] and [5]. The occlusion and the limited field-of-view problems are addressed in [10] and [18].

The contribution of this paper is an extension of [17] that deals with limited field-of-views, occlusions and unknown future behaviour of the objects. The approach proposed is based upon a relaxation of ICS that falls into the "evasive trajectories" family.

III. Statement of the Problem

Let $\mathcal{A}$ denote the mobile robot at hand. It operates in a 2D workspace $\mathcal{W}$. Its motion is governed by differential equations of the form:

$$\dot{s} = f(s,u) \text{ subject to } g(s,\dot{s}) \leq 0 \qquad (1)$$

[1] *I.e.* how far into the future the reasoning is done.

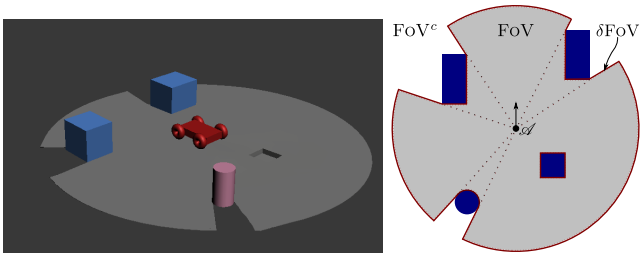


Fig. 1. Robot with a limited field-of-view (left) and its corresponding planar field-of-view FoV (right).

where $s \in \mathcal{S}$ is the state of $\mathcal{A}$, $\dot{s}$ its time derivative and $u \in \mathcal{U}$ a control. $\mathcal{S}$ and $\mathcal{U}$ respectively denote the state space and the control space of $\mathcal{A}$. Let $\mathcal{A}(s)$ denote the closed subset of the workspace $\mathcal{W}$ occupied by $\mathcal{A}$ when it is in s.

Let $\tilde{u} : [0, t_f] \longrightarrow \mathcal{U}$ denote a *control trajectory*, *i.e.* a time-sequence of controls, t_f is the duration of $\tilde{u}$. The set of all possible control trajectories is denoted $\tilde{\mathcal{U}}$. Starting from an initial state s_0 at time 0, a *state trajectory* $\tilde{s}$, *i.e.* a time-sequence of states, is derived from a control trajectory $\tilde{u}$ by integrating (1); $\tilde{s}(s_0, \tilde{u}, t)$ denotes the state reached at time t. A control trajectory $\tilde{u}_b \in \tilde{\mathcal{U}}$ such that $\tilde{s}_b(s_0, \tilde{u}_b, t_b)$ is a state where $\mathcal{A}$ comes to a halt (and remains so) is a *braking trajectory* for s_0 and t_b is its *braking time*. The set of all possible braking trajectories for s_0 is denoted $\tilde{\mathcal{U}}_b^{s_0}$.

Assuming that $\mathcal{A}$ is equipped with range sensors such as laser telemeters or range cameras, it can only perceive a subset of $\mathcal{W}$; this subset is $\mathcal{A}$'s *field-of-view*; its shape is arbitrary. It is henceforth denoted FoV. Accordingly, $\mathcal{W}$ is partitioned in three subsets: (1) FoV, (2) FoV^c, the part which is unseen (FoV$^c = \mathcal{W} \setminus cl(\text{FoV})$) and (3) ∂FoV, the boundary between the two. Both FoV and FoV^c are open sets. It seems reasonable to assume that $\mathcal{A}$ is "looking around itself"; in other words that $\mathcal{A}(s) \subset$ FoV where $\mathcal{A}(s)$ denotes the region of $\mathcal{W}$ occupied by $\mathcal{A}$ when it is in s. To account for the existence of 3D range sensors, *e.g.* Velodyne LIDAR, FoV can contain holes representing objects entirely perceived by the sensory system of $\mathcal{A}$. Accordingly, ∂FoV and FoV^c are not necessarily singly connected (see Fig. 1).

FoV represents the region of $\mathcal{W}$ which is free of objects at the sensing time while ∂FoV $\cup$ FoV^c represent objects (fixed or moving, seen and unseen). Recall that motion safety requires reasoning about the future motion of the objects in the environment. The model of the future used herein is conservative: it is assumed that $\mathcal{A}$ cannot distinguish the fixed from the moving objects (hence every object observed is treated as a potentially moving object), and that it has no information whatsoever about their future behaviour. Accordingly, given an upper-bound on the velocity of the objects, every point in ∂FoV $\cup$ FoV^{C}d is modeled as a disc that grows as time passes, *i.e.* a cone in space$\times$time (see [3]).

IV. Passively Safe Navigation

A. Braking ICS

Ref. [3] introduces a relaxation of the original ICS concept called Braking ICS. A Braking ICS (henceforth denoted

ICS^b) is a state for which, no matter what the future trajectory of the robot is, it is impossible to stop before a collision takes place. Braking ICS and passive safety are two dual concepts: a state which is not a Braking ICS is p-safe. An efficient Braking ICS-Checker (henceforth called ICS^b-CHECK) is also presented in [3], it checks whether a given state is a Braking ICS or not for a given model of the future.

B. PASSAVOID

In order to demonstrate passive motion safety and to validate the Braking ICS concept, a navigation scheme (henceforth called PASSAVOID) has been developed for a mobile robot $\mathcal{A}$ with a limited field-of-view placed in a unknown dynamic environment. PASSAVOID's primary task is to keep $\mathcal{A}$ in p-safe states, or equivalently, to drive $\mathcal{A}$ away from Braking ICS. PASSAVOID *guarantees passive motion safety* no matter what happens in the environment. In other words, if a collision takes place, it is guaranteed that $\mathcal{A}$ will be at rest when it occurs. PASSAVOID relies upon ICS^b-CHECK to operate. It is a reactive navigation scheme that operates with a given time step δt. At each time step, its purpose is to compute the constant control u that will be applied to $\mathcal{A}$ during the next time step; u must be *admissible*, *i.e.* the corresponding state trajectory must be p-safe (in other words, it must be ICS^b-free).

PASSAVOID operates like most standard reactive collision avoidance schemes, (*e.g.* [6], [8]). In all cases, their operating principle is to first characterize forbidden regions in a given control space and then select an admissible control, *i.e.* one which is not forbidden. Accordingly collision avoidance also depends on the ability of the collision avoidance scheme at hand to find such an admissible control. In the absence of a formal characterization of the forbidden regions, all schemes resort to some form of sampling of the control space with the inherent risk of missing the admissible regions. PASSAVOID also resorts to sampling in order to find an admissible control. However, in contrast with standard collision avoidance schemes, PASSAVOID is designed in such a way that it is guaranteed that, if an admissible control exists, it will be part of the sampling set.

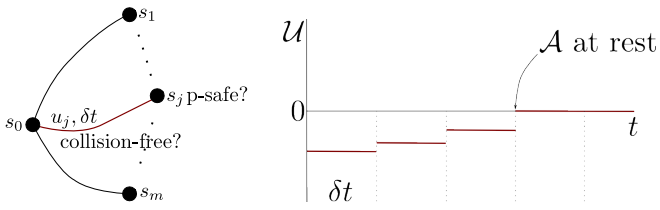


Fig. 2. PASSAVOID's operating principle (left), and example of a δ-braking trajectory (right).

The operating principle of PASSAVOID is illustrated in Fig. 2. Let s_0 denote the current state of $\mathcal{A}$ and U a sampled set of controls: $U = \{u_1 \dots u_m\}$. A given control $u_j \in U$ is applied to $\mathcal{A}$ for a duration δt. It takes $\mathcal{A}$ from the state s_0 to the state $s_j = \tilde{s}(s_0, u_j, \delta t)$. If the state trajectory between s_0 and s_j is p-safe then u_j is admissible. Using the Sufficient Safety Condition established in [19], the admissibility of u_j can equivalently be verified by checking that (1) the state trajectory between s_0 and s_j is collision-free (with respect to the model of the future), and that (2) s_j is p-safe, *i.e.* it is not a Braking ICS. This procedure is applied for every control in U; it yields a set of admissible controls denoted U^* from which PASSAVOID can pick the control to apply during the next time step. This selection can be made arbitrarily if one is only concerned with the survival of $\mathcal{A}$ or it can be made so as to ensure convergence towards a given goal (using for instance a global navigation function, a potential field, or even a partial motion planning scheme).

Such a scheme works well as long as an admissible control can be found in U. But if, at the end of the day, U^* is empty, it means that every control in U takes $\mathcal{A}$ to a Braking ICS. In other words, passive motion safety will not be achieved and a collision will take place while $\mathcal{A}$ is still moving. To address this issue, it is necessary to guarantee that $U = \{u_1 \dots u_m\}$ contains at least one admissible control. It is possible to achieve this by carefully designing PASSAVOID. To that end, a number of definitions and properties are required. They are introduced now. The concepts of δ-braking trajectory and δ-passive safety are defined first. They are just specific types of braking trajectory and passive safety:

Def. 1 (δ-Braking Trajectory): A braking trajectory $\tilde{u}^* \in \tilde{\mathcal{U}}_b^{s_0}$ of duration t^* is a *δ-braking trajectory* if it is constant over intervals of fixed duration δt.

A δ-braking trajectory is just a special type of braking trajectory (see Fig. 2). It yields a corresponding type of passive motion safety:

Def. 2 (δ-Passive Safety): A state s_0 is *δ-passively safe* or *δ-p-safe* if it exists one δ-braking trajectory $\tilde{u}^*$ starting at s_0 which is collision-free until $\mathcal{A}$ has stopped.

Then two useful properties are established:

*Property 1 (**P-Safe States**):* If the state s_0 is p-safe and the braking trajectory $\tilde{u}_b \in \tilde{\mathcal{U}}_b^{s_0}$ starting at s_0 is collision-free until $\mathcal{A}$ has stopped then every state $\tilde{s}(s_0, \tilde{u}_j, t), 0 < t \leq t_b$ is also p-safe.

Proof: Suppose that $\exists t_i \in]0, t_b]$ such that $\tilde{s}(s_0, \tilde{u}_b, t_i)$ is not p-safe then, by definition, $\forall \tilde{u}_j \in \tilde{\mathcal{U}}_b^{s_i}$, $\tilde{u}_j$ yields a collision before $\mathcal{A}$ stops. This also applies to the braking trajectory corresponding to the restriction of $\tilde{u}_b$ to the time interval $[t_i, t_b]$ which yields a contradiction. ■

Note that Property 1 also applies to δ-p-safe states.

*Property 2 (δ-**Passive Safety Guarantee**):* If the state s_0 is δ-p-safe then there exists at least one admissible control u^* that PASSAVOID can use to drive $\mathcal{A}$ to a state which is also δ-p-safe.

Proof: Since s_0 is δ-p-safe, there exists at least a one δ-braking trajectory $\tilde{u}^*$ starting at s_0 which is collision-free until $\mathcal{A}$ has stopped. As per Property 1, the state $\tilde{s}(s_0, \tilde{u}^*, \delta t)$ is δ-p-safe. Let u^* denote the value of $\tilde{u}^*$ over the time interval $[0, \delta t[$, u^* is an admissible control. ■

Property 2 is fundamental for the design of a version of PASSAVOID whose passive motion safety can be guaranteed. PASSAVOID simply has to drive $\mathcal{A}$ from one δ-p-safe state to the next. Now, assuming that s_0 is δ-p-safe, property 2

Algorithm 1: PASSAVOID.

Input: s_0, the current δ-p-safe state of $\mathcal{A}$; δt, the time step; model of the future.
Output: u

1 Sample $\mathcal{U} \rightsquigarrow U = \{u_1 \dots u_m\}$// [
2]Select the control space sampling set U
3 $U^* = K(s_0)$; // Initialize adm. controls
4 **forall** $u_j \in U$; // Compute adm. controls
5 **do**
6 $s(\delta t) = \tilde{s}(s_0, u_j, \delta t)$;
7 **if** $\tilde{s}(s_0, u_j, [0, \delta t[)$ *is collision-free and* $s(\delta t)$ *is* δ*-p-safe* **then**
8 $U^* = U^* \cup \{u_j\}$; // u_j admissible
9 **end**
10 **end**
// Select and return one adm. control
11 Select $u \in U^*$;
12 **return** u;

guarantees the existence of at least one admissible control u^* which, if applied to $\mathcal{A}$ for the duration δt, will take it to another δ-p-safe state. In general, a δ-p-safe state s has more than one admissible control. Let $K(s)$ denote this set of admissible controls, it is dubbed the *kernel* $K(s)$. Now, in order to guarantee its passive motion safety, PASSAVOID must include $K(s_0)$ in its control space sampling set. This is precisely what PASSAVOID does (see Algorithm 1, line #2). PASSAVOID features two important steps: computing the kernel $K(s_0)$ (line #2) and checking whether the state $s(\delta t)$ is δ-p-safe (line #6). It turns out that these two procedures are related and can be done by a straightforward adaptation of ICS^b-CHECK which is not detailed here due to lack of space (see [4]).

Now, provided that the initial state of the system $\mathcal{A}$ is δ-p-safe, Property 2 allows PASSAVOID to have at its disposal at each time step an admissible control that can be used to drive $\mathcal{A}$ from one δ-p-safe state to the next (forever if need be). Concerning the assumption on the initial state being δ-p-safe, it is satisfied when $\mathcal{A}$ is at rest, and the null control is admissible. In other words, starting with $\mathcal{A}$ at rest, PASSAVOID has an admissible control readily available that can be used right away if the situation demands it (this is true even if δt is very small). At the end of the day, PASSAVOID is *provably passively safe* in the sense that it is guaranteed that $\mathcal{A}$ will always stay away from Braking ICS no matter what happens in the environment.

C. Passively Safe Multi-Robot Navigation

In the introduction, it was stated that, if every moving object in a given environment was passively safe, then no collision should take place at all. It turns out that this property is straightforward to demonstrate. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ denote two robots that are driven by a provably passively safe navigation scheme such as PASSAVOID. As per Properties 1 and 2, both $\mathcal{A}_1$ and $\mathcal{A}_2$ are in a δ-p-safe state at all times. In other words, the following holds:

$$\forall t, s_1(t) \notin \text{ICS}_1^b \text{ and } s_2(t) \notin \text{ICS}_2^b \tag{2}$$

where $s_i(t)$ and ICS_i^b respectively denote the state at time t and the corresponding Braking ICS set for robot $\mathcal{A}_i, i = 1, 2$. Assuming that a collision can take place between $\mathcal{A}_1$ and $\mathcal{A}_2$ with one of them having a non zero velocity yields a contradiction. It cannot happen.

V. SIMULATION RESULTS

To validate the Braking ICS concept and demonstrate its usefulness, ICS^b-CHECK and PASSAVOID have both been implemented and tested in simulation.

A. Model of the Robot

The model of $\mathcal{A}$ is that of a standard car-like vehicle with two fixed rear wheels and two orientable front wheels. A state of $\mathcal{A}$ is a 5-tuple $s = (x, y, \theta, v, \xi)$ with (x, y) the coordinates of the rear axle midpoint, θ the orientation of $\mathcal{A}$, v the linear velocity of system, and ξ the orientation of the front wheels (steering angle). A control of $\mathcal{A}$ is a couple $u = (u_\alpha, u_\xi)$ with u_α the linear acceleration of the rear wheels and u_ξ the steering angle velocity. Let L denote the wheelbase of $\mathcal{A}$. The motion of $\mathcal{A}$ is governed by the following differential equations:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ v\tan\xi/L \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_\alpha + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_\xi \tag{3}$$

with $|v| \leq v_{\max}, |\xi| \leq \xi_{\max}, |u_\alpha| \leq u_{\alpha_{\max}}$ and $|u_\xi| \leq u_{\xi_{\max}}$.

B. PASSAVOID at Work

To illustrate how PASSAVOID works, two scenarios have been selected. The first one is called the *1D Compactor scenario*, it is simple but it helps to understand the kind of behaviour that PASSAVOID will yield when $\mathcal{A}$ is confronted to a clearly identified dangerous situation. The second one is called the *Blind Crowd scenario*; its primary purpose is to illustrate the performances of PASSAVOID in complex situations. The results obtained are also illustrated in a short film provided as a multimedia attachment to this paper[2]. In both cases, PASSAVOID had no information regarding the future trajectories of the moving objects. PASSAVOID did not attempt to drive $\mathcal{A}$ to a given goal. Its primary purpose was to keep $\mathcal{A}$ in p-safe states. Its secondary purpose was to keep $\mathcal{A}$ moving. In other words, the admissible control selection (line #10 of Algorithm 1) was biased towards controls yielding a non-zero linear velocity. This choice was made so as to avoid the straightforward answer to the passive motion safety problem which is simply to brake down and stop forever (by doing so, $\mathcal{A}$ reaches and stays in a p-safe state).

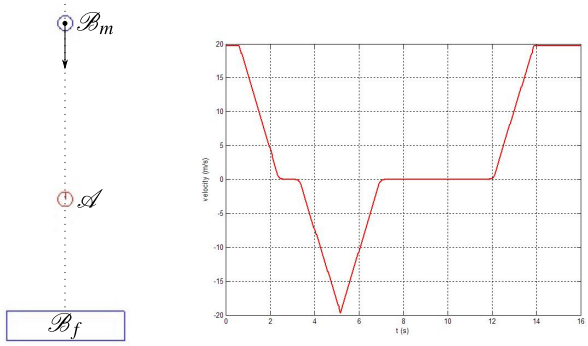


Fig. 3. 1D compactor scenario (left); corresponding velocity profile of $\mathcal{A}$ (right).

1) 1D Compactor Scenario: The 1D Compactor scenario features one fixed object $\mathcal{B}_f$ and one moving object $\mathcal{B}_m$. The moving object is moving towards the fixed object (see Fig. 3-left). $\mathcal{B}_f$ and $\mathcal{B}_m$ are like the two jaws of a compactor (hence the name of the scenario). $\mathcal{A}$ is placed between $\mathcal{B}_f$ and $\mathcal{B}_m$ and it is further assumed that $\mathcal{A}$ can only move along the vertical line connecting $\mathcal{B}_f$ and $\mathcal{B}_m$. At the beginning, $\mathcal{A}$ is moving upward with a positive linear velocity. In such a situation, the initial state s_0 of $\mathcal{A}$ is clearly an ICS (no matter what $\mathcal{A}$ does it will end up being crushed by $\mathcal{B}_m$). It is however possible to select $\mathcal{A}$'s initial position and linear velocity such that s_0 is p-safe. The parameters for this scenario were set as follows: $v_{\max} = 20\text{m.s}^{-1}$ (maximum velocity of $\mathcal{A}$ and $\mathcal{B}_m$), $u_{\alpha_{\max}} = 7\text{m.s}^{-2}$. The radius of $\mathcal{A}$ and $\mathcal{B}_m$ was 2.5m and the sensor range, *i.e.* the maximum radius of the field-of-view, was 80m. The control space sampling set U was obtained through a regular discretization of the control set $[-u_{\alpha_{\max}}, u_{\alpha_{\max}}]$. The set of braking trajectories $\mathcal{E}$ used by ICS^b-CHECK comprised one δ-braking trajectory defined by a constant minimum linear deceleration $u_\alpha = -u_{\alpha_{\max}}$.

In this scenario, when driven by PASSAVOID, $\mathcal{A}$ exhibits the following behaviour in order to always remain in p-safe states:

1) the increasing approach of $\mathcal{B}_m$ forces $\mathcal{A}$ to gradually decrease its velocity until it stops.
2) $\mathcal{A}$ backs up in order to avoid collision with $\mathcal{B}_m$ (recall that PASSAVOID is biased towards keeping $\mathcal{A}$ in motion).
3) while backing up, $\mathcal{A}$ gets closer to $\mathcal{B}_f$. At some point, it forces $\mathcal{A}$ to reduce its velocity.
4) $\mathcal{A}$ is now at rest next to $\mathcal{B}_f$, it will soon be hit by $\mathcal{B}_m$.
5) $\mathcal{A}$ is in collision with $\mathcal{B}_m$ ($t = 7s$).
6) when the collision with $\mathcal{B}_m$ is over[3], $\mathcal{A}$ resumes its upward motion.

The evolution of $\mathcal{A}$'s velocity in this scenario is depicted in Fig. 3-right. As simple as it may appear, this scenario shows how PASSAVOID seeks to avoid collision with $\mathcal{B}_m$ in a natural way (by braking down and shifting in reverse). However, when $\mathcal{A}$ is trapped, PASSAVOID guarantees that the robot will be at rest when the collision occurs.

2) Blind Crowd Scenario: The blind crowd scenario is more challenging. It features 22 moving objects moving arbitrarily in a 2D workspace. The objects are blind in the sense that their motion is unaffected by the other objects. The parameters for this scenario were set as follows: $v_{\max} = 15\text{m.s}^{-1}$ (maximum velocity of $\mathcal{A}$ and of the moving objects), $\xi_{\max} = \pi/3\text{rad}, u_{\alpha_{\max}} = 7\text{m.s}^{-2}, u_{\xi_{\max}} = 1.54\text{rad.s}^{-1}$. The radius of the disk objects was 2.5m and the sensor range, *i.e.* the maximum radius of the field-of-view, was 80m. The control space sampling set U was obtained through a regular discretization of the 2D control set $[-u_{\alpha_{\max}}, u_{\alpha_{\max}}] \times [u_{\xi_{\max}}, u_{\xi_{\max}}]$, and the set of braking trajectories $\mathcal{E}$ used by ICS^b-CHECK comprised 9 δ-braking trajectories defined by a constant minimum linear deceleration $u_\alpha = -u_{\alpha_{\max}}$ and a constant steering angle velocity $|u_\xi| \leq u_{\xi_{\max}}$.

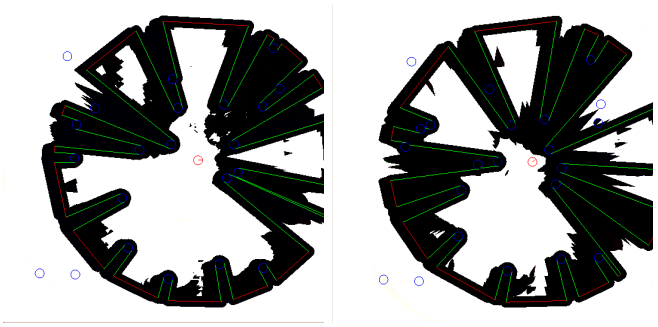


Fig. 4. Snapshots of PASSAVOID at work in the blind crowd scenario (the black region represents the ICS^b).

Fig. 4 presents snapshots taken at different time instants of one run of PASSAVOID in this scenario. Each snapshot feature $\mathcal{A}$ (at the center), the moving objects and the corresponding field-of-view. The set of ICS^b are also overlaid on the figure (black region).In the sequence, $\mathcal{A}$ is generally moving to the right. In the course of several runs, these experiments have demonstrated the capability of PASSAVOID to enforce passive motion safety: whenever a collision took place, $\mathcal{A}$ was at rest.

Note: looking at the provided video for the two scenarios considered, it may appear that PASSAVOID could do better in terms of collision avoidance and overall behaviour. Recall however that PASSAVOID is just a reactive collision avoidance scheme, it has no foresight and is not concerned with driving $\mathcal{A}$ to a given goal. In the authors' opinion, the important result concerning PASSAVOID is the formal proof of its passive motion safety. These simulations merely serve as a proof of concept and to illustrate how PASSAVOID operates. Ways to improve PASSAVOID are discussed in §VI.

C. Complexity and Performance

The computational time complexity of PASSAVOID grows linearly with n_s, the size of the control space sampling set U (**forall** loop of Algorithm 1), and the complexity

[2]Downloadable from http://emotion.inrialpes.fr/fraichard/films/11-auro-passavoid.wmv.

[3]Assuming that $\mathcal{B}_m$ sort of passes through $\mathcal{A}$.

of one iteration depends primarily on the complexity of ICS^b-CHECK (δ-p-safety test in line #6 of Algorithm 1). Given that the complexity of ICS^b-CHECK grows linearly with n_b (the size of the set of braking trajectories), n_o (number of objects) and n_t (number of the time steps used to represent the model of the future), the final time complexity of PASSAVOID is $O(n_s n_b n_o n_t)$.

TABLE I
AVERAGE RUNNING TIME OF ICS^b-CHECK *wrt* n_o, THE NUMBER OF OBJECTS ($n_b = 9, n_t = 71$).

n_o	4	10	17	22
Running time (ms)	49	101	123	138

The current implementation of both ICS^b-CHECK and PASSAVOID has been done in C++ on an average laptop computer (Intel Core i7 1.6GHz CPU, 4GB RAM, ATI Mobility Radeon HD 4500 GPU). Table I gives the average running times of ICS^b-CHECK *wrt* n_0, the number of objects. These running times are encouraging and could further be improved thanks to code optimization (a CUDA implementation is underway), or the use of a more powerful desktop.

VI. CONCLUSION AND FUTURE WORK

This paper has addressed the problem of navigating in a provably safe manner a mobile robot with a limited field-of-view placed in a unknown dynamic environment. The position taken in this paper was to guarantee level of motion safety dubbed *passive motion safety*: if a collision takes place, the robot will be at rest. Passive motion safety has been tackled using a variant of the Inevitable Collision State (ICS) concept called *Braking ICS* [3], *i.e.* states such that, whatever the future braking trajectory followed by the robot, a collision occurs before it is at rest. To validate the Braking ICS concept and demonstrate its usefulness, the Braking ICS-Checker of [3] has been integrated in a reactive collision avoidance scheme called PASSAVOID. The main contribution of this paper has been to *formally prove* PASSAVOID's passive motion safety. This work could be extended in the following directions:

In certain situations, PASSAVOID may drive the robot to a collision state although such a collision could have been avoided. This is due to PASSAVOID's lack of foresight[4]. Besides, PASSAVOID is not concerned with driving the robot to a given goal. These issues could be addressed by turning PASSAVOID into a Partial Motion Planner à la [19]. Such an extension would yield a navigation scheme better able to avoid collisions and to reach a given goal while retaining the passive motion safety guarantee.

It could also be interesting to explore more sophisticated levels of motion safety such as the *passive friendly motion safety* mentioned in [17]: it guarantees that, if a collision takes place, the robot will be at rest and the colliding object could have had the time to stop or avoid the collision (if it wanted to). Such a motion safety level assume that the moving objects have cognitive abilities and are not hostile (which happens to be true in many situations).

[4]PASSAVOID is purely reactive, its sole purpose is to compute the control to apply for the next time step.

REFERENCES

[1] A. Bautin, L. Martinez-Gomez, and T. Fraichard, "Inevitable collision states, a probabilistic perspective," in *IEEE Int. Conf. Robotics and Automation*, Anchorage (US), May 2010.

[2] K. Bekris, K. Tsianos, and L. Kavraki, "Safe and distributed kino-dynamic replanning for vehicular networks," *Mobile Networks and Applications*, vol. 14, no. 3, June 2009.

[3] S. Bouraine, T. Fraichard, and H. Salhi, "Relaxing the inevitable collision state concept to address provably safe mobile robot navigation with limited field-of-view in unknown dynamic environments," in *IEEE-RSJ Int. Conf. Intelligent Robots and Systems*, San Francisco (US), 2011.

[4] ——, "Provably safe navigation for mobile robots with limited field-of-views in dynamic environments," *Autonomous Robots*, 2012.

[5] W. Chung, S. Kim, M. Choi, J. Choi, H. Kim, C. Moon, and J. Song, "Safe navigation of a mobile robot considering visibility of environment," *IEEE Trans. Industrial Electronics*, vol. 56, no. 10, Oct. 2009.

[6] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. Journal of Robotics Research*, vol. 17, no. 7, July 1998.

[7] L. Fletcher, S. Teller, E. Olson, D. Moore, Y. Kuwata, J. How, J. Leonard, I. Miller, M. Campbell, D. Huttenlocher, A. Nathan, and F.-R. Kline, "The MIT-Cornell collision and why it happened," *Int. Journal of Field Robotics*, vol. 25, no. 10, Oct. 2008.

[8] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, Mar. 1997.

[9] T. Fraichard, "A short paper about motion safety," in *IEEE Int. Conf. Robotics and Automation*, Roma (IT), Apr. 2007.

[10] T. Fraichard and H. Asama, "Inevitable collision states. a step towards safer robots?" *Advanced Robotics*, vol. 18, no. 10, 2004.

[11] E. Frazzoli, E. Feron, and M. Dahleh, "Real-time motion planning for agile autonomous vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, Jan.-Feb. 2002.

[12] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kin-odynamic motion planning with moving obstacles," *Int. Journal of Robotics Research*, vol. 21, no. 3, Mar. 2002.

[13] M. Kalisiak and M. van de Panne, "Faster motion planning using learned local viability models," in *IEEE Int. Conf. Robotics and Automation*, Roma (IT), Apr. 2007.

[14] R. Kohout, J. Hendler, and D. Musliner, "Guaranteeing safety in spatially situated agents," in *AAAI Nat. Conf. Artificial Intelligence*, Portland (US), Aug. 1996.

[15] E. Lalish and K. Morgansen, "Decentralized reactive collision avoidance for multivehicle systems," in *IEEE Conf. Decision and Control*, Cancun (MX), Dec. 2008.

[16] V. Lumelsky and S. Tiwari, "Velocity bounds for motion planning in the presence of moving planar obstacles," in *IEEE-RSJ Int. Conf. Intelligent Robots and Systems*, Munchen (DE), Sept. 1994.

[17] K. Macek, D. Vasquez-Govea, T. Fraichard, and Siegwart, "Towards safe vehicle navigation in dynamic urban scenarios," *Automatika*, vol. 50, no. 3-4, 2009.

[18] K. Madhava Krishna, R. Alami, and T. Simeon, "Safe proactive plans and their execution," *Robotics and Autonomous Systems*, vol. 54, no. 3, Mar. 2006.

[19] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton (CA), Aug. 2005.

[20] J. Reif and M. Sharir, "Motion planning in the presence of moving obstacles," in *IEEE Int. Symp. Foundations of Computer Science*, Portland (US), Oct. 1985.

[21] M. Sadou, V. Polotski, and P. Cohen, "Occlusion in obstacle detection for safe navigation," in *IEEE Intelligent Vehicles Symp.*, Parma (IT), June 2004.