Relaxing the Inevitable Collision State Concept to Address Provably Safe Mobile Robot Navigation with Limited Field-of-Views in Unknown Dynamic Environments

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

***Abstract*— This paper addresses the problem of provably safe navigation for a mobile robot with a limited field-of-view placed in a unknown dynamic environment. In such a situation, absolute motion safety (in the sense that no collision will ever take place whatever happens in the environment) is impossible to guarantee in general. It is therefore settled for a weaker level of motion safety dubbed *passive motion safety*: it guarantees that, if a collision is inevitable, the robot will be at rest. The primary contribution of this paper is a relaxation of the Inevitable Collision State (ICS) concept called *Braking ICS*. A Braking ICS 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 are designed with a passive motion safety perspective for robots with a limited field-of-view in unknown dynamic environments. Braking ICS are formally defined and a number of important properties are established. These properties are then used to design a Braking ICS checker, *i.e.* an algorithm that checks whether a given state is a Braking ICS or not. In a companion paper, it is shown how the Braking ICS checker can be integrated into a reactive navigation scheme whose passive motion safety is provably guaranteed.**

I. Introduction

Robotics technology is now mature and Autonomous Ground Vehicles (AGVs) are becoming a reality: consider the successes of the DARPA Challenges or the VisLab Intercontinental Autonomous Challenge. They demonstrate robotics systems traveling significant distances at high speed in complex and realistic environments. However such systems remains prone to accidents (see [1]). While moving (especially at high speed), AGVs (and other robotic systems as well) can be potentially dangerous should a collision occur; this is a critical issue if such systems are to transport or share space with human beings.

Roboticists have long been aware of the motion safety issue; there is a rich literature on collision avoidance and collision-free navigation. Nonetheless, motion safety has for a long time remained a taken-for-granted and ill-defined notion (see [2]). Demonstrating that a robot avoids collision on a limited set of experiments is not enough. If autonomous robots are ever to be deployed among human beings on a large scale, there is a need to design collision avoidance and navigation schemes for which motion safety can be characterized or even guaranteed. The literature review of §II shows that the Robotics community is displaying a growing interest in designing such provably safe collision avoidance and navigation schemes. 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 a problem, *i.e.* that of provably safe motion for a robot with sensors having a limited field-of-view in an unknown environment featuring moving objects whose future behaviour is unknown; it is merely assumed that their speed is upper-bounded. It can be argued that in such a situation, *absolute motion safety* (in the sense that no collision will ever take place whatever happens in the environment) is impossible to guarantee in general (unless questionable assumptions concerning the robot and its environment are made). In theory, absolute motion safety requires a complete knowledge of the future, up to infinity in some singular situations (see the motion safety criteria laid down in [2] and the discussion on motion safety of [3]). To cope with that issue, a practical stance is taken in this paper; it is settled for a weaker level of motion safety; in other words, it is guaranteed that, if a collision takes place, the robot at hand will be at rest. As per [3], this motion safety level is dubbed *passive motion safety*.

The primary contribution of this paper is a relaxation of the Inevitable Collision State (ICS) concept developed in [4]. An ICS is a state for which, no matter what the future trajectory of the robot is, a collision eventually occurs. ICS were originally defined with an absolute motion safety perspective which is incompatible with the assumptions made in this paper, *i.e.* limited field-of-view and limited knowledge of the future. The relaxed ICS, henceforth called *Braking ICS*, are defined with a passive motion safety perspective. A Braking ICS 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 are formally defined and a number of important properties are established. These properties are then used to derive a Braking ICS checker, *i.e.* the passively safe version of the ICS-checking algorithm presented in [5].

In itself, the central idea behind passive motion safety, *i.e.* using braking trajectories, is not new, it has been used before in different contexts (see Section 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

[†]CDTA (AL); [‡]INRIA (FR); [⋆]Blida Univ. (AL).

it concerns the robot's dynamics, its field-of-view, or the knowledge (or lack thereof) about the future behaviour of the moving objects. As limited as it may appear, passive motion safety is interesting for two reasons: (1) it allows to provide at least one form of motion safety guarantee in such challenging scenarios. , and (2) if every moving object in the environment enforces it then no collision will take place at all.

The paper is organized as follows: a review of the relevant literature is done in §II. §III discusses motion safety issues and defines passive motion safety. The adaptation of the ICS concept to passive motion safety is done in §IV. The passively safe version of the ICS-checking algorithm is detailed in §V. Finally, experimental results obtained in simulation are presented in §VI.

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 [2]).

The earliest relevant works addressed the so-called "Asteroid Avoidance Problem" (wherein objects traveling at a constant linear velocity must be avoided): in the 3D case, [6] shows that collision avoidance is always possible if the robot's velocity is greater than the asteroids' velocities and if the robot is not initially in the "shadow" of an asteroid. In the 2D case, [7] shows that collision avoidance is always possible iff the asteroids appear beyond a "threat horizon", *i.e.* a distance which is a function of the number, size and velocity of the asteroids. Likewise, [8] shows that, for a robot operating in a planar environment with arbitrarily moving objects, collision-free motion 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.* [9], [10]. 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 [4]. 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:

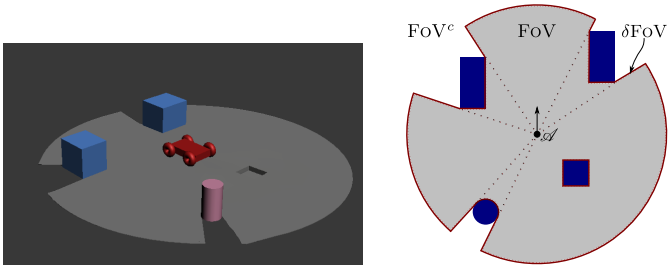
- *ICS approximation*, *e.g.* [11], [12]: such approximations being not conservative, the motion safety guarantee is lost.
- τ-*Safety*, *e.g.* [13], [14]: the robot is guaranteed to remain in states where it is safe for a given duration (hopefully sufficient to compute an updated safe trajectory...).
- *Evasive trajectories*, *e.g.* [15], [16]: 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.* [17], [18], so as to better capture the uncertainty that prevails in real world situations, in particular the uncertainty concerning the future behaviour of the moving objects. These approaches are interesting but they 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 a coarse manner in [19] and in a more principled manner in [20]. The occlusion and the limited field-of-view problems are addressed in [4] and [21]. [4] addresses the case of a mobile robot moving in a static environment; its approach is general and ICS-based. While [21] considers dynamic environments, it does so primarily with a path-velocity decomposition perspective. The contribution of this paper is the extension of [4] to the dynamic environment case.

III. Safety Issues

A. Outline of the Problem



(a) (b)

Fig. 1: (a) Robot with a limited field-of-view in an unknown environment with fixed and moving objects.. (b) Its corresponding field-of-view ∂FoV. FoV is the grey area; ∂FoV and FoV^c have two connected components..

As mentioned in §I, this paper addresses the problem of provably safe motion for a mobile robot with sensors having a limited field-of-view in an unknown environment featuring fixed and moving objects with upper-bounded velocity and unknown future behaviour. Let $\mathscr{A}$ denote the mobile robot at hand. It operates in a 2D workspace $\mathscr{W}$; a state of $\mathscr{A}$ is denoted by s with $s \in \mathscr{S}$, the state space of $\mathscr{A}$. Assuming that $\mathscr{A}$ is equipped with range sensors such as laser telemeters

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

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

This generic field-of-view model can further be enriched if the sensors of $\mathscr{A}$ can differentiate the fixed *vs* the moving objects. In that case, ∂FoV can be partitioned into three parts respectively corresponding to fixed objects, moving objects and so-called "unseen" objects, *i.e.* the sensing limits and the occluding lines:

$$\partial\text{FoV} = \partial\text{FoV}^f \cup \partial\text{FoV}^m \cup \partial\text{FoV}^u \quad (1)$$

When the sensors of $\mathscr{A}$ cannot differentiate between fixed and moving objects, $\partial\text{FoV} = \partial\text{FoV}^u$.

B. Modeling the Future

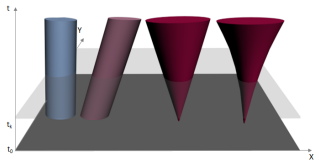


Fig. 2: Models of the future (from left to right): fixed object (1); moving object with constant velocity (known future motion) (2); conservative models for a moving point with unknown future motion and upper-bounded velocity (3), and upper-bounded acceleration (4).

The ICS concept brings to light two things: the first one is that there is more to motion safety than the simple fact that $\mathscr{A}$'s trajectory be collision-free; it must be ICS-free, *i.e.* $\mathscr{A}$ must always be in a state for which an evasive trajectory is available. The second one is that motion safety is always defined *wrt* the model of the future that is used. When dealing with objects whose future behaviour is unknown, what model of the future should be used? The answer is to be *conservative*: one must consider all possible future motions for the object at hand. Consider the case of a point object with upper-bounded velocity whose future behaviour is unknown. Given the initial position of the object, the region of the workspace that is possibly not collision-free is modeled by a disc that grows over time with a growth rate corresponding to the maximal velocity of the object [22]. In space$\times$time, it is represented as an inverted cone (see Fig. 2). Such a cone is the *reachable set* [23, Chap. 14] of a point object whose dynamics is characterized by infinite acceleration and upper-bounded velocity capabilities. In general, reachable sets can be used to represent all possible future motions for object with arbitrary dynamics, *e.g.* an object with upper-bounded velocity *and* acceleration (see Fig. 2).

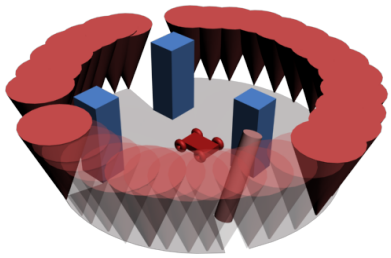


Fig. 3: Conservative model of the future (partially represented for visualisation purposes) for the scenario of Fig. 1a.

Now, in a situation such as the one depicted in Fig. 1b, how does one take into account the unseen parts of $\mathscr{W}$ that belongs to ∂FoV^u or FoV^c? Walking in the footsteps of [4] or [21], the answer is once again to be conservative and to treat every point of ∂FoV^u or FoV^c as a potential moving object with unknown future behaviour. In conclusion, the space$\times$time model of the future for $\mathscr{A}$ can be defined as follows for the different components of $\mathscr{A}$'s field-of-view (see Fig. 3):

- ∂FoV$^u \cup$FoV^C (the unseen objects): every point in this set is modeled as a disc that grows as time passes (*i.e.* a cone in space$\times$time).
- ∂FoV^f (the fixed objects): every point in this set remains constant over time (*i.e.* a vertical line in space$\times$time).
- ∂FoV^m (the moving objects): if the information about their future behaviour is available and reliable, every point in this set is modelled accordingly (*i.e.* a curve in space$\times$time), otherwise it is treated as an unseen object and modeled as a growing disc.

This of course is the case when the sensors of $\mathscr{A}$ can differentiate between fixed and moving objects. If it is not the case then every point in ∂FoV is modeled as a disc that grows as time passes (*i.e.* a cone in space$\times$time). Within such a model of the future, it is worth noting that the region of $\mathscr{W}$ which is free of objects at the sensing time, *i.e.* FoV, gradually shrinks as time passes and eventually vanishes (see Fig. 3). Henceforth, FoV(t) denotes the region of $\mathscr{W}$ which is free of objects at time t in the conservative model of the future. Likewise, ∂FoV(t) denotes its boundary.

C. Absolute vs. Passive Motion Safety

The ICS concept laid down in [4] guarantees *absolute motion safety* in the sense that, for a state not to be an ICS, there must exist a collision-free trajectory of infinite duration. Now, an object with unknown future behaviour is a challenge. If it is modeled conservatively as above then, at some point in the future, the whole workspace is

entirely covered by the growing disc representing it. At that moment, the whole state space of the robot is forbidden and it becomes impossible to find a collision-free trajectory of infinite duration. This is a situation where the ICS concept becomes ineffective. In the authors'opinion, the only answer to this challenge is to settle for a weaker level of motion safety; the rationale being: *better guarantee something than guarantee nothing*. The choice here is to guarantee that, if a collision takes place, the robot will be at rest. This motion safety level, dubbed *passive motion safety* in [3], seems a reasonable choice given the harsh constraints imposed by a limited field-of-view. It yields the following definition:

Def. 1: given a model of the future workspace evolution, a *passively safe* or *p-safe* state for $\mathscr{A}$ is a state s such that there exists one braking trajectory starting at s which is collision-free until $\mathscr{A}$ has stopped.

IV. From ICS to Braking ICS

Using braking trajectories in order to evaluate the safety of a given state has been done before, *e.g.* [15], [16], [24]–[26]. The focus in this paper is to do it in the formal framework of the ICS concept. The concept of *Braking ICS* (ICS^b) is first derived from the original ICS concept. It is then used to design ICS^b-CHECK, *i.e.* the corresponding variant of the ICS checking algorithm proposed in [5]. ICS^b-CHECK is detailed in §V.

A. Notations

The dynamics of the robot $\mathscr{A}$ is generally described by differential equations of the form:

$$\dot{s} = f(s,u) \text{ subject to } g(s,\dot{s}) \leq 0 \tag{2}$$

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

Let $\tilde{u} : [0,t_f] \longrightarrow \mathscr{U}$ 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 $\tilde{\mathscr{U}}$. 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 (2); $\tilde{s}(s_0,\tilde{u},t)$ denotes the state reached at time t.

A control trajectory $\tilde{u}_b \in \tilde{\mathscr{U}}$ such that $\tilde{s}_b(s_0,\tilde{u}_b,t_b)$ is a state where $\mathscr{A}$ 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 $\tilde{\mathscr{U}}_b^{s_0}$.

In a situation such as the one depicted in Fig. 1b, the open subset FoV is the free part of the workspace while $\partial\text{FoV}^f, \partial\text{FoV}^m, \partial\text{FoV}^u$ and FoV^c represent objects (seen and unseen). Let $\mathscr{B}_i$ denote the space×time model of the future evolution of the corresponding object (according to the modeling rules defined in §III-B). At time 0, *i.e.* the sensing time, $\mathscr{B}_i(0)$ corresponds to a subset of ∂FoV^f, ∂FoV^m, ∂FoV^u or FoV^c. $\mathscr{B}_i(t)$ denotes the subset of $\mathscr{W}$ occupied by $\mathscr{B}_i$ at a particular time t in the conservative model of the future. It is assumed that each $\mathscr{B}_i(t)$ is a closed subset of $\mathscr{W}$ and that the total number of objects is n. Likewise $\mathscr{B}_i([t_1,t_2])$ denote the space×time region occupied by the object during the interval $[t_1,t_2]$. To ease notations, it is assumed that $\mathscr{B}_i \equiv \mathscr{B}_i([0,\infty))$.

B. Braking ICS Definition

A Braking ICS (ICS^b) is informally defined as a state for which no matter what the future braking trajectory followed by $\mathscr{A}$ is, a collision occurs before $\mathscr{A}$ is at rest. Hence the following formal definition:

*Def. 2 (**Braking ICS**):* s is a ICS^b iff $\forall \tilde{u}_b \in \tilde{\mathscr{U}}_b^s, \exists t \in [0,t_b[, \tilde{s}(s,\tilde{u}_b,t)$ is a collision state at time t.

It is worth noting that when $\mathscr{A}$ is in a state s where $\mathscr{A}$ is at rest, $\tilde{\mathscr{U}}_b^s$ reduces to $\tilde{u}_b^\circ$ that denotes the braking trajectory where a null control is applied to $\mathscr{A}$. Accordingly, s is always p-safe (even if $\mathscr{A}(s)$ is in collision).

It is then possible to define the set of ICS^b yielding a collision with a particular object $\mathscr{B}_i$:

$$\begin{aligned} \text{ICS}^b(\mathscr{B}_i) &= \{s \in \mathscr{S} | \forall \tilde{u}_b \in \tilde{\mathscr{U}}_b^s, \exists t \in [0,t_b[, \\ &\quad \mathscr{A}(\tilde{s}(s,\tilde{u}_b,t)) \cap \mathscr{B}_i(t) \neq \emptyset\} \end{aligned} \tag{3}$$

Likewise, the ICS^b set yielding a collision with $\mathscr{B}_i$ for a given trajectory $\tilde{u}_b$ (or a given set of trajectories $\mathscr{E} \subseteq \tilde{\mathscr{U}}_b$) is defined as:

$$\begin{aligned} \text{ICS}^b(\mathscr{B}_i,\tilde{u}_b) &= \{s \in \mathscr{S} | \exists t \in [0,t_b[, \\ &\quad \mathscr{A}(\tilde{s}(s,\tilde{u}_b,t)) \cap \mathscr{B}_i(t) \neq \emptyset\} \end{aligned} \tag{4}$$

$$\text{ICS}^b(\mathscr{B}_i,\mathscr{E}) = \bigcap_{\tilde{u}_b \in \mathscr{E}} \text{ICS}^b(\mathscr{B}_i,\tilde{u}_b) \tag{5}$$

C. Braking ICS Properties

The first two ICS^b properties that can be shown are the equivalent of two key ICS properties established in [4] and seminal in the design of an ICS checking algorithm. Let $\mathscr{B} = \bigcup_1^n \mathscr{B}_i$. The first property shows that $\text{ICS}^b(\mathscr{B})$ can be derived from $\text{ICS}^b(\mathscr{B}_i,\tilde{u}_b)$ for every object $\mathscr{B}_i$ and every possible braking trajectory $\tilde{u}_b$.

*Property 1 (**ICS^b Characterization**):*

$$\text{ICS}^b(\mathscr{B}) = \bigcap_{\tilde{u}_b \in \tilde{\mathscr{U}}_b} \bigcup_{i=1}^n \text{ICS}^b(\mathscr{B}_i,\tilde{u}_b)$$

Proof: The two-stage proof of property 1 is done in a straightforward manner as in [4]. It is first established that:

$$s \in \text{ICS}^b(\mathscr{B}) \Leftrightarrow s \in \bigcap_{\tilde{u}_b \in \tilde{\mathscr{U}}_b^s} \text{ICS}^b(\mathscr{B},\tilde{u}_b)$$

and then that:

$$s \in \text{ICS}^b(\bigcup_{i=1}^n \mathscr{B}_i,\tilde{u}_b) \Leftrightarrow s \in \bigcup_{i=1}^n \text{ICS}^b(\mathscr{B}_i,\tilde{u}_b)$$

Combining the two properties above yields property 1. ■

The next property permits to compute a conservative approximation of $\text{ICS}^b(\mathscr{B})$ by using a subset only of the whole set of possible braking trajectories.

*Property 2 (**ICS^b Approximation**):*

$$\text{ICS}^b(\mathscr{B}) \subseteq \text{ICS}^b(\mathscr{B},\mathscr{E})$$

with $\mathscr{E} \subseteq \tilde{\mathscr{U}_b}$, a subset of the whole set of possible braking trajectories.

Proof:

$$\begin{aligned}\mathrm{ICS}^b(\mathscr{B}) &= \bigcap_{\tilde{u}_b \in \mathscr{E}} \mathrm{ICS}^b(\mathscr{B},\tilde{u}_b) \cap \bigcap_{\tilde{u}_b \in \tilde{\mathscr{U}_b} \setminus \mathscr{E}} \mathrm{ICS}^b(\mathscr{B},\tilde{u}_b) \\ &\subseteq \bigcap_{\tilde{u}_b \in \mathscr{E}} \mathrm{ICS}^b(\mathscr{B},\tilde{u}_b)\end{aligned}$$

■

One distinctive feature of the ICS concept is that trajectories of infinite duration are checked for collision, *i.e.* it has an infinite lookahead (it is this infinite lookahead that guarantees safety). While the ICS^b concept also considers trajectories $\tilde{u}_b$ of infinite duration, collision checking is limited to the time interval $[0, t_b[$ where t_b is the braking time of $\tilde{u}_b$. For an arbitrary subset $\mathscr{E}$ of the whole set of possible braking trajectories, a finite lookahead T_h exists:

$$T_h = \max_{\tilde{u}_b \in \mathscr{E}} \{t_b\} \tag{6}$$

T_h is a valid lookahead in the sense that, in order to compute $\mathrm{ICS}^b(\mathscr{B},\mathscr{E})$, it suffices to consider the model of the future up to time T_h. This is established by the following property:

*Property 3 (**ICS^b Lookahead**):*

$$\mathrm{ICS}^b(\mathscr{B},\mathscr{E}) = \mathrm{ICS}^b(\mathscr{B}([0,T_h[),\mathscr{E})$$

Proof: Property 3 stems from the very definition of a Braking ICS which, for a given braking manoeuvre $\tilde{u}_b$, is only concerned with collisions taking place before $t_b < T_h$. ■

Finally, recall from §III-A that, for the case of a robot with a limited field-of-view, $\mathscr{B}$ comprises ∂FoV and FoV^c, *i.e.* the unseen part of $\mathscr{W}$. From a motion safety perspective, the next property is very important since it establishes that FoV^c can be ignored in the computation of ICS$^b(\mathscr{B})$. In other words, considering ∂FoV suffices to guarantee motion safety.

*Property 4 (**Field-of-View Boundary**):*

$$\mathrm{ICS}^b(\mathscr{B}) = \mathrm{ICS}^b(\partial\mathrm{FoV} \cup \mathrm{FoV}^c) = \mathrm{ICS}^b(\partial\mathrm{FoV})$$

Proof: The equality between ICS$^b(\mathscr{B})$ and ICS$^b(\partial$FoV$)$ is done is two stages. Let s denote a collision-free state whose corresponding position is located inside FoV and such that $s \in \mathrm{ICS}^b(\partial\mathrm{FoV})$. As per Definition 2, it stems that:

$$\forall \mathscr{B}_i, \forall \mathscr{B}_j, \mathrm{ICS}^b(\mathscr{B}_i) \subseteq \mathrm{ICS}^b(\mathscr{B}_i \cup \mathscr{B}_j)$$

Accordingly:

$$s \in \mathrm{ICS}^b(\partial\mathrm{FoV}) \Rightarrow s \in \mathrm{ICS}^b(\partial\mathrm{FoV} \cup \mathrm{FoV}^c).$$

It is assumed now that $s \in \mathrm{ICS}^b(\mathrm{FoV}^c)$, it means that $\forall \tilde{u}_b \in \tilde{\mathscr{U}_b}^s, \exists t \in [0,t_b[$ such that $\tilde{s}(s,\tilde{u}_b,t)$ is in collision with a point of FoV$^c(t)$. Since s is located inside FoV, it takes a simple topological argument to realize that $\exists t' < t$ such that $\tilde{s}(s,\tilde{u}_b,t')$ is in collision with a point of ∂FoV(t'). Accordingly $s \in \mathrm{ICS}^b(\partial\mathrm{FoV})$ and the following holds:

$$s \in \mathrm{ICS}^b(\partial\mathrm{FoV} \cup \mathrm{FoV}^c) \Rightarrow s \in \mathrm{ICS}^b(\partial\mathrm{FoV}).$$

■

In other words, it suffices to consider ∂FoV in order to compute ICS$^b(\mathscr{B})$.

V. Braking ICS Checking

ICS^b-Check is an algorithm that checks whether a given state is a Braking ICS or not. It is the passively safe version of the ICS checking algorithm (called ICS-Check) presented in [5]. The passively safe version of ICS-Check can be designed because Properties 1 and 2 are verified for Braking ICS. The steps involved in checking whether a given state s_c is a ICS^b are given in Algorithm 1. Besides the state to be checked, the algorithm takes as input the model of the environment and the conservative space$\times$ model of the future (see §IV-A). Steps 2, 3 and 4 are the direct translation of Property 1.

Algorithm 1: General ICS^b Checking Algorithm.

Input: s_c, the state to be checked; $\mathscr{B}_i, i = 1 \ldots n$.
Output: Boolean value.

1 Select $\mathscr{E} \subset \tilde{\mathscr{U}_b}^{s_c}$, a set of braking trajectories for s_c;
2 Compute ICS$^b(\mathscr{B}_i,\tilde{u}_b)$ for every $\mathscr{B}_i$ and every $\tilde{u}_b \in \mathscr{E}$;
3 Compute $\mathrm{ICS}^b(\mathscr{B},\tilde{u}_b) = \bigcup_{i=1}^n \mathrm{ICS}^b(\mathscr{B}_i,\tilde{u}_b)$ for every $\tilde{u}_b \in \mathscr{E}$;
4 Compute $\mathrm{ICS}^b(\mathscr{B},\mathscr{E}) = \bigcap_{\tilde{u}_b \in \mathscr{E}} \mathrm{ICS}^b(\mathscr{B},\tilde{u}_b)$;
5 **if** $s_c \in \mathrm{ICS}^b(\mathscr{B},\mathscr{E})$ **then**
6 **return** True;
7 **else**
8 **return** False;
9 **end**

As in [5], when $\mathscr{A}$ is planar, it becomes possible to design ICS^b-Check, *i.e.* an efficient version of Algorithm 1. In that case, a state s of $\mathscr{A}$ can be rewritten $s = (x,y,\hat{\mathbf{z}})$ with (x,y) the workspace coordinates of $\mathscr{A}$'s reference point, and $\hat{\mathbf{z}}$ the rest of the state parameters. The primary design principle behind ICS^b-Check is to compute the ICS^b set corresponding to a 2D slice of the state space $\mathscr{S}$ of $\mathscr{A}$ (instead of attempting to perform computation in the fully-dimensioned state space), and then to check if s_c belongs to this set. Assuming the state to be checked is $s_c = (x_c, y_c, \hat{\mathbf{z}}_\mathbf{c})$, the 2D slice considered is the $\hat{\mathbf{z}}_\mathbf{c}$-slice and it is possible to define the ICS^b set of the $\hat{\mathbf{z}}_\mathbf{c}$-slice considered that yields a collision with $\mathscr{B}_i$ at a particular time $t \in [0,t_b[$ for the braking trajectory $\tilde{u}_b$:

$$\begin{aligned}\mathrm{ICS}^b{}_{\hat{\mathbf{z}}_\mathbf{c}}(\mathscr{B}_i,\tilde{u}_b,t) &= \{s \in \hat{\mathbf{z}}_\mathbf{c}\text{-slice} | \\ &\quad \mathscr{A}(\tilde{s}(s,\tilde{u}_b,t)) \cap \mathscr{B}_i(t) \neq \emptyset\}\end{aligned} \tag{7}$$

Likewise:

$$\mathrm{ICS}^b{}_{\hat{\mathbf{z}}_\mathbf{c}}(\mathscr{B}_i,\tilde{u}_b) = \bigcup_{t \in [0,t_b[} \mathrm{ICS}^b{}_{\hat{\mathbf{z}}_\mathbf{c}}(\mathscr{B}_i,\tilde{u}_b,t) \tag{8}$$

Applying this 2D reasoning principle, ICS^b-CHECK is similar to the general ICS^b Checking Algorithm detailed in Algorithm 1 except that, at all steps of the algorithm, ICS$^b_{\hat{\mathbf{z}}_\mathbf{c}}$ is computed instead of ICS^b (see Algorithm 2). It is by keeping all computations in 2D (notwithstanding the actual dimensionality of $\mathscr{S}$) that it is possible to efficiently compute the ICS^b set corresponding to a given $\hat{\mathbf{z}}_\mathbf{c}$-slice.

Algorithm 2: ICS^b-CHECK.

Input: s_c, the state to be checked; $\mathscr{B}_i, i = 1 \dots n$.
Output: Boolean value.

Select $\mathscr{E} \subset \tilde{\mathscr{U}}_b^{s_c}$, a set of braking trajectories for s_c;
forall $\tilde{u}_b \in \mathscr{E}$ **do**
  forall $\mathscr{B}_i$ **do**
    Compute ICS$^b_{\hat{\mathbf{z}}_\mathbf{c}}(\mathscr{B}_i, \tilde{u}_b)$;
  end
  Compute ICS$^b_{\hat{\mathbf{z}}_\mathbf{c}}(\mathscr{B}, \tilde{u}_b) = \bigcup_{i=1}^{n}$ ICS$^b_{\hat{\mathbf{z}}_\mathbf{c}}(\mathscr{B}_i, \tilde{u}_b)$;
end
Compute ICS$^b_{\hat{\mathbf{z}}_\mathbf{c}}(\mathscr{B}, \mathscr{E}) = \bigcap_{\tilde{u}_b \in \mathscr{E}}$ ICS$^b_{\hat{\mathbf{z}}_\mathbf{c}}(\mathscr{B}, \tilde{u}_b)$;
if $s_c \in$ ICS$^b(\mathscr{B}, \mathscr{E})$ **then**
  return TRUE;
else
  return FALSE;
end

For the sake of brevity and because of the similarity between ICS^b-CHECK and ICS-CHECK, the inner workings of ICS^b-CHECK are not detailed here. The reader is referred to [5] instead. Suffice to say that ICS^b-CHECK provides an efficient way to check whether a given state is a ICS^b or not.

VI. SIMULATION RESULTS

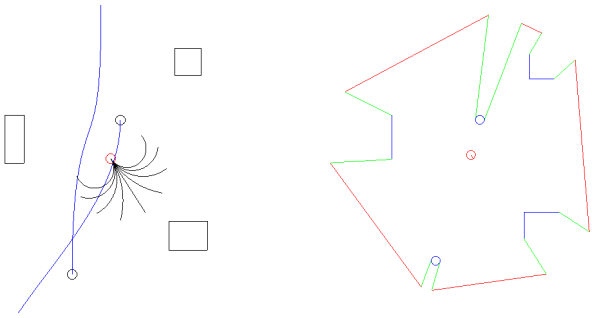
ICS^b-CHECK has been implemented and tested in simulation on scenarios similar to that of Fig. 1a.

A. Model of the Robot

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

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ v\tan\xi/L \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_\alpha + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_\xi \qquad (9)$$

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



(a) (b)

Fig. 4: (a) Test scenario: three fixed rectangles and two moving discs (with their future trajectories). The robot $\mathscr{A}$ is the disc at the center, with $\mathscr{E}$, the set of braking trajectories considered by ICS^b-CHECK. (b) The corresponding Field-of-view ∂FOV.

B. Workspace and Field-of-View

A typical test scenario is depicted in Fig. 4a. The planar workspace $\mathscr{W}$ contains polygonal objects and disks that can be fixed or moving with a given maximum velocity. Assuming that $\mathscr{A}$ is equipped with an omnidirectional laser range finder mounted at the center of $\mathscr{A}$, the field-of-view of $\mathscr{A}$ is depicted in Fig. 4b. The circular arc corresponding to the maximum range of the range finder have been replaced by straight segments; this conservative simplification could easily be lifted. In this particular scenario, it is assumed that the sensors can differentiate between the fixed and the moving objects and that the future motion of the moving objects is known beforehand. In general, it does not have to be the case (see §III-B).

C. ICS^b-CHECK at Work

To briefly illustrate how ICS^b-CHECK works, the scenario in Fig. 4a is used. ICS^b-CHECK is called to determine whether the current state of $\mathscr{A}$ is a Braking ICS or not. This state is $s_c = (0, 0, -1, 0, 20)$. As per Algorithm 2, ICS^b-CHECK computes the ICS^b set for the corresponding $\hat{\mathbf{z}}_\mathbf{c}$-slice with $\hat{\mathbf{z}}_\mathbf{c} = (-1, 0, 20)$. A set $\mathscr{E}$ of braking trajectories must be selected. They can be chosen arbitrarily since they always yield a conservative approximation of the ICS^b set (as per Property 2). In this case, $\mathscr{E}$ comprised nine braking trajectories defined by a constant minimum linear deceleration $u_\alpha = -u_{\alpha_{\max}}$ and a constant steering angle velocity $|u_\xi| \leq u_{\xi_{\max}}$. These braking trajectories are depicted in Fig. 4a.

For each braking trajectory $\tilde{u}_b \in \mathscr{E}$, the set ICS$^b_{\hat{\mathbf{z}}_\mathbf{c}}(\mathscr{B}_i, \tilde{u}_b)$ is computed. Exploiting graphics rendering techniques, ICS$^b_{\hat{\mathbf{z}}_\mathbf{c}}(\mathscr{B}_i, \tilde{u}_b)$ yields a region of a given color on a OpenGL buffer representing the $\hat{\mathbf{z}}_\mathbf{c}$-slice. All the steps of ICS^b-CHECK that involves computing unions and intersections of arbitrary shapes are performed very efficiently (in the order of a few milliseconds) on this OpenGL buffer by taking advantage of the Red-Green-Blue color coding and the bitwise logical operators available; for additional details, the reader is referred to [5]. The final output of this process is illustrated in Figs. 5 and 6 where it appears

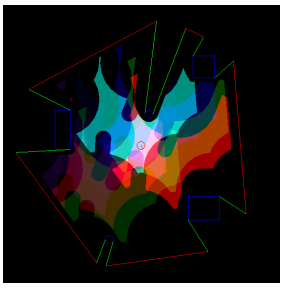


Fig. 5: Two dimensional $\hat{\mathbf{z}}_\mathbf{c}$-slice of the 5D state space of $\mathscr{A}$: a region of a given color indicates that it is a ICS^b for the corresponding braking trajectory. Black regions are ICS^b.

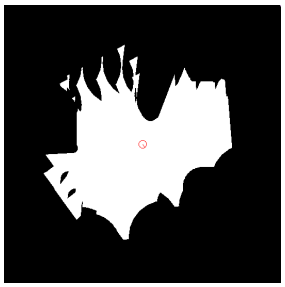


Fig. 6: Black and White version of Fig. 5: white regions correspond to p-safe states.

that $s_c = (0,0,-1,0,20)$ does not belong to the black regions. Thus, s_c is not a ICS^b.

A short video[2] illustrates the evolution of the Braking ICS regions when the robot is moving in the test scenario depicted in Fig. 4a .

VII. Conclusion and Future Work

This paper has addressed the problem of provably safe motion for a mobile robot with a limited field-of-view placed in a unknown dynamic environment. Since absolute motion safety is impossible to guarantee in such a situation, the stance taken herein was to settle for a weaker level of motion safety dubbed *passive motion safety*: it guarantees that, if a collision is inevitable, the robot will be at rest.

The primary contribution of this paper has been to propose a relaxation of the Inevitable Collision State (ICS) concept called *Braking ICS*. A Braking ICS 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 are designed with a passive motion safety perspective for robots with a limited field-of-view in unknown dynamic environments. Braking ICS have been formally defined and a number of important properties have been established. These properties have been then used to design a Braking ICS checker, *i.e.* an algorithm that checks whether a given state is a Braking ICS or not. Simulation results have shown the efficiency of the Braking ICS checker proposed.

In some applications, passive motion safety can be too limited; it could then be interesting to explore more sophisticated levels of motion safety such as the *passive friendly motion safety* introduced in [3]: it guarantees that, if a collision is inevitable, 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). In general, it could be interesting to explore other forms of motion safety depending on the particulars of the navigation problem at hand.

[2]http://emotion.inrialpes.fr/fraichard/films/11-iros-ics-bouraine-etal.mp4

References

[1] 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.

[2] T. Fraichard, "A short paper about motion safety," in *ICRA2007*, Roma (IT), Apr. 2007.

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

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

[5] L. Martinez-Gomez and T. Fraichard, "An Efficient and Generic 2D Inevitable Collision State-Checker," in *IROS2008*, Nice (FR), Sept. 2008.

[6] 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.

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

[8] V. Lumelsky and S. Tiwari, "Velocity bounds for motion planning in the presence of moving planar obstacles," in *IROS1994*, Munchen (DE), Sept. 1994.

[9] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *ICRA2008*, Pasadena (US), May 2008.

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

[11] N. Chan, M. Zucker, and J. Kuffner, "Towards safe motion planning for dynamic systems using regions of inevitable collision," in *Collision-free Motion Planning for Dynamic Systems Workshop*, Rome (IT), Apr. 2007.

[12] M. Kalisiak and M. van de Panne, "Faster motion planning using learned local viability models," in *ICRA2007*, Roma (IT), Apr. 2007.

[13] 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.

[14] R. Vatcha and L. Xiao, "Perceived CT-Space for Motion Planning in Unknown and Unpredictable Environments," in *Workshop on Algorithmic Foundations of Robotics*, Guanajuato (MX), Dec. 2008.

[15] K. Bekris and L. Kavraki, "Greedy but safe replanning under kinodynamic constraints," in *ICRA2007*, Rome (IT), Apr. 2007.

[16] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in *ICRA2007*, Roma (IT), Apr. 2007.

[17] A. Bautin, L. Martinez-Gomez, and T. Fraichard, "Inevitable collision states, a probabilistic perspective," in *ICRA2010*, Anchorage (US), May 2010.

[18] D. Althoff, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic collision state checker for crowded environments," in *ICRA2010*, Anchorage (US), May 2010.

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

[20] 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.

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

[22] J. Van den Berg and M. Overmars, "Planning time-minimal safe paths amidst unpredictably moving obstacles," *Int. Journal of Robotics Research*, vol. 27, no. 11-12, 2008.

[23] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[24] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *IROS2005*, Edmonton (CA), Aug. 2005.

[25] D. Ferguson, T. Howard, and M. Likhachev, "Motion planning in urban environments," *Journal of Field Robotics*, vol. 25, no. 11-12, 2008.

[26] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. How, and G. Fiore, "Real-time motion planning with applications to autonomous urban driving," *IEEE Trans. Control Systems Technology*, vol. 17, no. 5, 2009.