Inevitable Collision States. A Step Towards Safer Robots?

Thierry Fraichard, Hajime Asama

IEEE/RSJ Int. Conf. on Intelligent Robots and Systems
October 27-31, 2003. Las Vegas, NV (US)

Inevitable Collision States
A Step Towards Safer Robots?

Thierry Fraichard
Inria[a] Rhône-Alpes & Gravir[b]
655 av. de l'Europe, Montbonnot, 38334 St Ismier Cedex, France
http://www.inrialpes.fr/sharp/people/fraichard
thierry.fraichard@inria.fr

Hajime Asama
Riken Institute[c]
2-1, Hirosawa, Wako, Saitama 351-0198, Japan
http://celultra.riken.go.jp/~asama
asama@cel.riken.go.jp

University of Tokyo
Komaba 4-6-1, Muguro-ku, Tokyo 153-8904, Japan
http://www.race.u-tokyo.ac.jp/~asama
asama@race.u-tokyo.ac.jp

July 7, 2003

Abstract — An *inevitable collision state* for a robotic system can be defined as a state for which, no matter what the future trajectory followed by the system is, a collision with an obstacle eventually occurs. An inevitable collision state takes into account both the dynamics of the system and the obstacles, fixed or moving. The main contribution of this paper is to lay down and explore this novel concept (and the companion concept of *inevitable collision obstacle*). Formal definitions of the inevitable collision states and obstacles are given. Properties fundamental for their characterisation are established. This concept is very general and can be useful both for navigation and motion planning purposes (for its own safety, a robotic system should never find itself in an inevitable collision state!). The interest of this concept is illustrated by a safe motion planning example.

Keywords — Robotics, safety, navigation, motion planning, sensing constraints.

Acknowledgements — This work was partially supported by the Japan Society for the Promotion of Science and Lafmi, the French-Mexican Computer Science Laboratory.

[a]Institut National de Recherche en Informatique et en Automatique.
[b]Lab. Graphisme, Vision et Robotique.
[c]Institute of Physical and Chemical Research.

Inevitable Collision States
A Step Towards Safer Robots?

Thierry Fraichard
Inria Rhône-Alpes & Gravir Laboratory (FR)
http://www.inrialpes.fr/sharp/people/fraichard

Hajime Asama
Riken Institute & University of Tokyo (JP)
http://www.race.u-tokyo.ac.jp/˜asama

***Abstract*— An *inevitable collision state* for a robotic system can be defined as a state for which, no matter what the future trajectory followed by the system is, a collision with an obstacle eventually occurs. An inevitable collision state takes into account both the dynamics of the system and the obstacles, fixed or moving. The main contribution of this paper is to lay down and explore this novel concept (and the companion concept of *inevitable collision obstacle*). Formal definitions of the inevitable collision states and obstacles are given. Properties fundamental for their characterisation are established. This concept is very general and can be useful both for navigation and motion planning purposes (for its own safety, a robotic system should never find itself in an inevitable collision state!). The interest of this concept is illustrated by a safe motion planning example.**

I. Introduction

The configuration space of a robotic system is the appropriate framework to address path planning problems where the focus is on the geometric aspects of motion planning (no collision between the system and the fixed obstacles of the workspace) [1], [2]. The state space, on the other hand, is more appropriate when it comes to address trajectory planning problems where the dynamics of the system is taken into account [3], [4]. Similarly, the time-state space is appropriate to address trajectory planning problems involving moving obstacles [5], [6], [7].

In the configuration space, the notion of forbidden or *collision configurations*, *ie* configurations yielding a collision, is well-known and so is the notion of *configuration obstacles*, *ie* the set of configurations yielding a collision between the system and a particular obstacle [2]. Transposing these notions in the state space, it is straightforward to define *collision states* and *state obstacles* (idem in the time-state space).

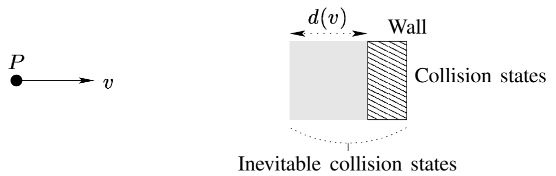

Fig. 1. Collision states *vs* inevitable collision states.

However, be it in state space or time-state space, it takes a simple example such as the one depicted in Fig. 1 to illustrate the interest of extending these notions so as to take into account the dynamics of the system by introducing the concept of *inevitable collision states*.

Consider Fig. 1, P is a point mass that can only move to the right with a variable speed (a state of P is therefore characterised by its position (x, y) and its speed v). If the environment features a wall, the states whose position corresponds to the wall are obviously collision states. On the other hand, assuming that it takes P a certain distance $d(v)$ to slow down and stop, the states corresponding to the wall *and* the states located at a distance less than $d(v)$ left of the wall are such that, when P is in such a state, no matter what it does in the future, a collision will occur. These states are inevitable collision states for P. Clearly, for P's own safety, when it is moving at speed v, it should never be in one of these inevitable collision states.

In general, an *inevitable collision state* for a given robotic system can be defined as a state for which, no matter what the future trajectory followed by the system is, a collision eventually occurs with an obstacle of the environment. Similarly, it is possible to define an *inevitable collision obstacle* as the set of inevitable collision states yielding a collision with a particular obstacle. Except for a brief mention of it in [8], this concept does not seem to have been considered before. However, we believe it can be very useful be it for motion planning or navigation purposes.

Consider navigation first (by navigation, we basically mean the problem of determining the elementary motion that the robotic system should perform during the next time-step). The primary concern of navigation is to ensure the safety of the robotic system. In a environment featuring moving obstacles, this safety concern is critical and it is important to take into account both the dynamics of the robotic system and the future behaviour of the moving obstacles. A number of research works have addressed these issues recently [9], [10], [11], [12], [13]. In this framework, the interest of the inevitable collision state concept is obvious. By design, inevitable collision states integrate both the dynamics of the robotic system and the obstacles, fixed or moving. Besides, it was mentioned earlier that, for its own safety, a robotic system should never end up in an inevitable collision state.

When it comes to motion planning, the inevitable collision state concept can also be useful. Consider the problem of planning motions for a robotic system moving in a partially known environment. The system is subject to sensing constraints (a limited field of view), and it moves in an environment containing obstacles, some of them are known beforehand while others are not (imagine a surveillance robot, it has a map of the building it must patrol but it does not know a priori the position of the small furniture or if people are be moving around). Based on the a priori information available, a nominal trajectory for the robotic system can be computed. However, what if, at execution time, the robotic system finds itself in a situation where an unknown obstacle is detected so late that avoiding it is impossible. The issue here is to compute *safe motions*, *ie* motions for which it is guaranteed that, no matter what happens at execution time, the robotic system never finds itself in a situation where there is no way for it to avoid collision with an unexpected obstacle. This issue is related to the dependency that exists between motion planning and navigation, dependency which is usually ignored by motion planning systems (with the exception of [14]). We show on an example how this issue can be addressed using the inevitable collision state concept and how safe motions (in the sense given above) can be planned.

[0]This work was partially supported by the Japan Society for the Promotion of Science and Lafmi, the French-Mexican Computer Science Laboratory.

The main contribution of this paper is to lay down and explore the concept of inevitable collision states. To begin with, a formal definition of what inevitable collision states and inevitable collision obstacles are is given. Properties that are fundamental for their characterisation are established (§III). To illustrate the use of these properties, a basic example is studied (§IV). Finally, an example of application of the inevitable collision state concept to safe motion planning is given (§V).

II. Notations and Preliminary Definitions

Before defining the inevitable collision states and obstacles, useful definitions and notations are introduced. Let $\mathcal{A}$ denote a robotic system. Its motion is governed by the following differential equation: $\dot{s} = f(s, u)$ 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 ϕ denote a *control input*, *ie* a time-sequence of controls. ϕ represents a trajectory for $\mathcal{A}$. Starting from an initial state s_0 (at time 0) and under the action of a control input ϕ, the state of $\mathcal{A}$ at time t is denoted by $s(t) = \phi(s_0, t)$. Given a control input ϕ and a state s_0 (at time 0), a state s is *reachable from s_0 by ϕ* iff $\exists t, \phi(s_0, t) = s$. Let $\mathcal{R}(s_0, \phi)$ denote the set of states reachable from s_0 by ϕ. Likewise, $\mathcal{R}(s_0)$ denotes the set of states s reachable from s_0, *ie* such that $\exists \phi, s \in \mathcal{R}(s_0, \phi)$:

$$\begin{aligned} \mathcal{R}(s_0, \phi) &= \{s \in \mathcal{S} | \exists t, \phi(s_0, t) = s\} \\ \mathcal{R}(s_0) &= \{s \in \mathcal{S} | \exists \phi, s \in \mathcal{R}(s_0, \phi)\} \end{aligned}$$

Introducing $\phi^{-1}(s_0, \mathrm{t})$ to denote the state s such that $\phi(s, t) = s_0$, it is possible to define $\mathcal{R}^{-1}(s_0)$ (resp. $\mathcal{R}^{-1}(s_0, \phi)$), as the set of states from which it is possible to reach s_0 (resp. to reach s_0 by ϕ):

$$\begin{aligned} \mathcal{R}^{-1}(s_0, \phi) &= \{s \in \mathcal{S} | \exists t, \phi(s, t) = s_0 \Leftrightarrow \phi^{-1}(s_0, t) = s\} \\ \mathcal{R}^{-1}(s_0) &= \{s \in \mathcal{S} | \exists \phi, s \in \mathcal{R}^{-1}(s_0, \phi)\} \end{aligned}$$

Let $\mathcal{W}$ denote the *workspace* of $\mathcal{A}$ ($\mathcal{W} = \mathbb{R}^2$ or $\mathbb{R}^3$), it contains a set of obstacles. An obstacle $\mathcal{B}$, fixed or moving, is defined as a closed subset of $\mathcal{W}$ (time-dependent if $\mathcal{B}$ is moving). A state s is a *collision state* iff $\exists \mathcal{B}, \mathcal{A}(s) \cap \mathcal{B} \neq \emptyset$, where $\mathcal{A}(s)$ denotes the closed subset of $\mathcal{W}$ occupied by $\mathcal{A}$ in state s. In this case, s is a *collision state with $\mathcal{B}$*.

III. Inevitable Collision States and Obstacles

Based on the definitions and notations introduced in the previous section, the inevitable collision states and the inevitable collision obstacles are formally defined.

*Def. 1 (**Inevitable Collision State**):* Given a control input ϕ, a state s is an *inevitable collision state for ϕ* iff $\exists t$ such that $\phi(s, t)$ is a collision state. Now, a state s is an *inevitable collision state* iff $\forall \phi, \exists t$ such that $\phi(s, t)$ is a collision state. Likewise, s is an *inevitable collision state with $\mathcal{B}$* iff $\forall \phi, \exists t$ such that $\phi(s, t)$ is a collision state with $\mathcal{B}$.

*Def. 2 (**Inevitable Collision Obstacle**):* Given an obstacle $\mathcal{B}$ and a control input ϕ, $ICO(\mathcal{B}, \phi)$, the *inevitable collision obstacle of $\mathcal{B}$ for ϕ* is defined as:

$$\begin{aligned} ICO(\mathcal{B}, \phi) &= \{s \in \mathcal{S} | s \text{ is an inev. coll. state with } \mathcal{B} \text{ for } \phi\} \\ &= \{s \in \mathcal{S} | \exists t, \phi(s, t) \text{ is a collision state with } \mathcal{B}\} \end{aligned}$$

Now, $ICO(\mathcal{B})$, the *inevitable collision obstacle of $\mathcal{B}$*, is defined as:

$$\begin{aligned} ICO(\mathcal{B}) &= \{s \in \mathcal{S} | s \text{ is an inevitable collision state with } \mathcal{B}\} \\ &= \{s \in \mathcal{S} | \forall \phi, \exists t, \phi(s, t) \text{ is a collision state with } \mathcal{B}\} \end{aligned}$$

Based upon the two definitions above, the following property can be established. It shows that $ICO(\mathcal{B})$ can be derived from the $ICO(\mathcal{B}, \phi)$ for every possible control input ϕ.

*Property 1 (**Control Inputs Intersection**):*

$$ICO(\mathcal{B}) = \bigcap_{\mathcal{U}} ICO(\mathcal{B}, \phi)$$

Proof:

$$\begin{aligned} s \in ICO(\mathcal{B}) &\Leftrightarrow \forall \phi, \exists t, \phi(s, t) \text{ is a coll. state with } \mathcal{B} \\ &\Leftrightarrow \forall \phi, s \in ICO(\mathcal{B}, \phi) \\ &\Leftrightarrow s \in \textstyle\bigcap_{\mathcal{U}} ICO(\mathcal{B}, \phi) \end{aligned}$$

■

Assuming now that $\mathcal{B}$ is the union of a set of obstacles, $\mathcal{B} = \bigcup_i \mathcal{B}_i$, the following property can be established. It shows that $ICO(\mathcal{B}, \phi)$ can be derived from the $ICO(\mathcal{B}_i, \phi)$ for every subset $\mathcal{B}_i$.

*Property 2 (**Obstacles Union**):*

$$ICO(\bigcup_i \mathcal{B}_i, \phi) = \bigcup_i ICO(\mathcal{B}_i, \phi)$$

Proof:

$$\begin{aligned} s \in ICO(\textstyle\bigcup_i \mathcal{B}_i, \phi) &\Leftrightarrow \exists t, \phi(s, t) \text{ is a coll. state with } \textstyle\bigcup_i \mathcal{B}_i \\ &\Leftrightarrow \exists \mathcal{B}_i, \exists t, \phi(s, t) \text{ is a coll. state with } \mathcal{B}_i \\ &\Leftrightarrow \exists \mathcal{B}_i, s \in ICO(\mathcal{B}_i, \phi) \\ &\Leftrightarrow s \in \textstyle\bigcup_i ICO(\mathcal{B}_i, \phi) \end{aligned}$$

■

Combining the two properties above, the following property is derived. It is the property that permits the formal characterisation of the inevitable collision obstacles for a given robotic system.

*Property 3 (**ICO Characterisation**):* Let $\mathcal{B} = \bigcup_i \mathcal{B}_i$,

$$ICO(\mathcal{B}) = \bigcap_{\mathcal{U}} \bigcup_i ICO(\mathcal{B}_i, \phi)$$

Proof:

$$ICO(\mathcal{B}) \stackrel{1}{=} \bigcap_{\mathcal{U}} ICO(\mathcal{B}, \phi) \stackrel{2}{=} \bigcap_{\mathcal{U}} \bigcup_i ICO(\mathcal{B}_i, \phi)$$

■

Consider property 1 (and property 3), it establishes that $ICO(\mathcal{B})$ can be derived from the $ICO(\mathcal{B}, \phi)$ for every possible control input ϕ. In general, there is an infinite number of control inputs which leaves little hope of being actually able to compute $ICO(\mathcal{B})$. Fortunately, it is possible to establish a property which is of a vital practical value since it shows how to compute a conservative approximation of $ICO(\mathcal{B})$ by using a subset only of the whole set of possible control inputs.

*Property 4 (**ICO Approximation**):* Let $\mathcal{I}$ denote a subset of the set of possible control inputs,

$$ICO(\mathcal{B}) \subset \bigcap_{\mathcal{I}} ICO(\mathcal{B}, \phi)$$

Proof:

$$\begin{aligned} ICO(\mathcal{B}) &\stackrel{1}{=} \bigcap_{\mathcal{I} \cup \overline{\mathcal{I}}} ICO(\mathcal{B}, \phi) \\ &= \bigcap_{\mathcal{I}} ICO(\mathcal{B}, \phi) \cap \bigcap_{\overline{\mathcal{I}}} ICO(\mathcal{B}, \phi) \\ &\subseteq \bigcap_{\mathcal{I}} ICO(\mathcal{B}, \phi) \end{aligned}$$

■

The interest of these properties to characterise inevitable collision obstacles appears in the next sections.

IV. Basic Case Study

The purpose of this section is to illustrate on a simple (and not necessary realistic!) example the notions introduced earlier. A more realistic example is dealt with later in §V

A. "North, North-East" System

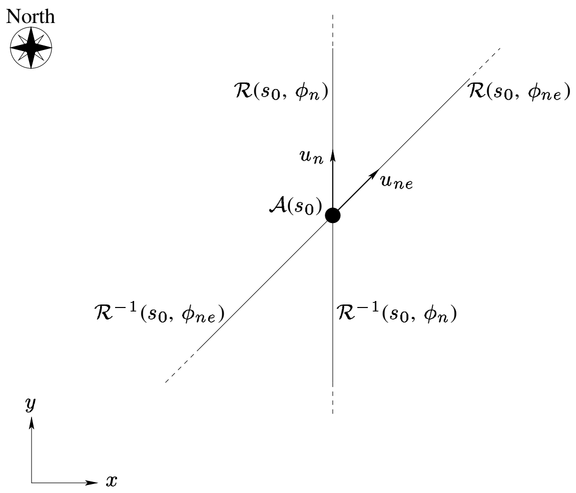

Fig. 2. Reachable states for the "North, North-East" system.

We consider the case of a planar point $\mathcal{A}$ that can move in two directions only (North and North-East) at constant unit speed (Fig. 2). A state of $\mathcal{A}$ is $s = (x, y) \in \mathbb{R}^2$ and a control u can take two values: either $u_n = \pi/2$ (North direction), or $u_{ne} = \pi/4$ (North-East direction). This simple system has only two possible control inputs: ϕ_n and ϕ_{ne}, they respectively correspond to motions in the North and North-East directions.

$\mathcal{R}(s_0)$, *ie* the set of states reachable from an initial state s_0, is easily defined in this case: it is the union of two half-lines starting at s_0 and extending respectively in the North and North-East directions: $\mathcal{R}(s_0) = \mathcal{R}(s_0, \phi_n) \cup \mathcal{R}(s_0, \phi_{ne})$. Likewise, $\mathcal{R}^{-1}(s_0)$, *ie* the set of states from which s_0 is reachable, is the union of two half-lines starting at s_0 and extending respectively in the South and South-West directions: $\mathcal{R}^{-1}(s_0) = \mathcal{R}^{-1}(s_0, \phi_n) \cup \mathcal{R}^{-1}(s_0, \phi_{ne})$ (Fig. 2).

The next sections show how to determine the inevitable collision obstacles corresponding to the "North, North-East" system. We proceed step by step by considering fixed obstacles first and then moving obstacles. In each case, we address point obstacles first before moving to arbitrary obstacles.

B. Fixed Obstacle

1) Point Obstacle: Let $\mathcal{B}$ be a fixed point obstacle. According to property 1, $ICO(\mathcal{B})$ is derived from the characterisations of $ICO(\mathcal{B}, \phi)$ for every possible control input ϕ. In this case, $\mathcal{B}$ is equivalent to a state of $\mathcal{A}$. Accordingly, $ICO(\mathcal{B}, \phi) = \mathcal{R}^{-1}(\mathcal{B}, \phi)$ and the following derivation is made (Fig. 3):

$$\begin{aligned} ICO(\mathcal{B}) &\stackrel{1}{=} ICO(\mathcal{B}, \phi_n) \cap ICO(\mathcal{B}, \phi_{ne}) \\ &= \mathcal{R}^{-1}(\mathcal{B}, \phi_n) \cap \mathcal{R}^{-1}(\mathcal{B}, \phi_{ne}) \\ &= \mathcal{B} \end{aligned}$$

which makes perfect sense: unless $\mathcal{A}$ is already in collision with $\mathcal{B}$, $\mathcal{A}$ can always avoid collision with $\mathcal{B}$. The state corresponding to $\mathcal{B}$ is the only inevitable collision state.

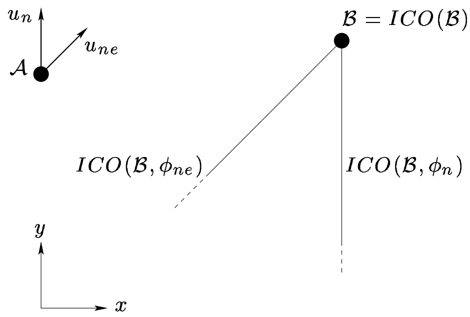

Fig. 3. Inevitable collision obstacle for a fixed point obstacle.

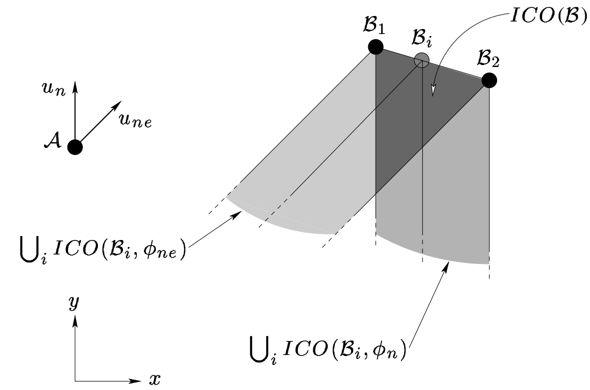

Fig. 4. Inevitable collision obstacle for a fixed linear obstacle.

2) Linear and Arbitrary Obstacle: Let us now assume that $\mathcal{B}$ is a fixed linear obstacle extending from point $\mathcal{B}_1$ to point $\mathcal{B}_2$. $\mathcal{B}$ is the union of a set of fixed point obstacles: $\mathcal{B} = \bigcup_i \mathcal{B}_i$. Now, $ICO(\mathcal{B})$ is derived using both properties 1 and 2:

$$\begin{aligned} ICO(\mathcal{B}) &\stackrel{1}{=} ICO(\mathcal{B}, \phi_n) \cap ICO(\mathcal{B}, \phi_{ne}) \\ &= ICO(\textstyle\bigcup_i \mathcal{B}_i, \phi_n) \cap ICO(\bigcup_i \mathcal{B}_i, \phi_{ne}) \\ &\stackrel{2}{=} \textstyle\bigcup_i ICO(\mathcal{B}_i, \phi_n) \cap \bigcup_i ICO(\mathcal{B}_i, \phi_{ne}) \end{aligned}$$

Consider Fig. 4, $\bigcup_i ICO(\mathcal{B}_i, \phi_n)$ is the region swept by $ICO(\mathcal{B}_i,\, \phi_n)$ for every point $\mathcal{B}_i$ between $\mathcal{B}_1$ and $\mathcal{B}_2$ (idem for

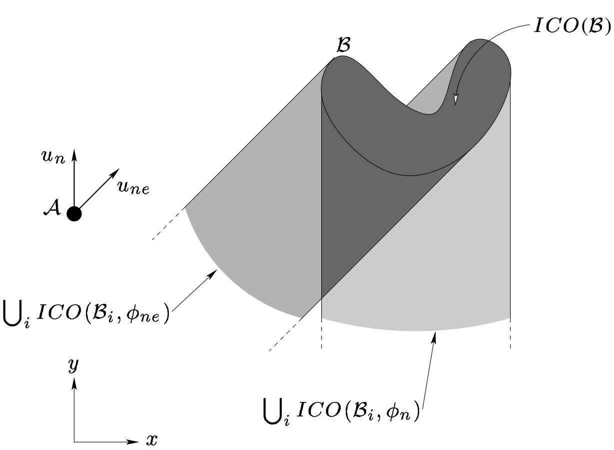

Fig. 5. Inevitable collision obstacle for a fixed arbitrary obstacle.

$\bigcup_i ICO(\mathcal{B}_i, \phi_{nw})$). The intersection between these two regions yields a simple triangular region which is $ICO(\mathcal{B})$. Sure enough, when $\mathcal{A}$ is anywhere inside this region, no matter what it does, it eventually crashes against $\mathcal{B}$.

Likewise, it is possible to characterise $ICO(\mathcal{B})$ for fixed obstacles with arbitrary shape (Fig. 5).

C. Moving Obstacle

As mentioned earlier in §I, the time dimension must be taken into account when dealing with moving obstacles (a state can yield a collision at time t_1 and be collision-free at time t_2). Time-state space becomes the appropriate framework and the different notions introduced so far, inevitable collision states and obstacles, are easily transposed to account for the time dimension.

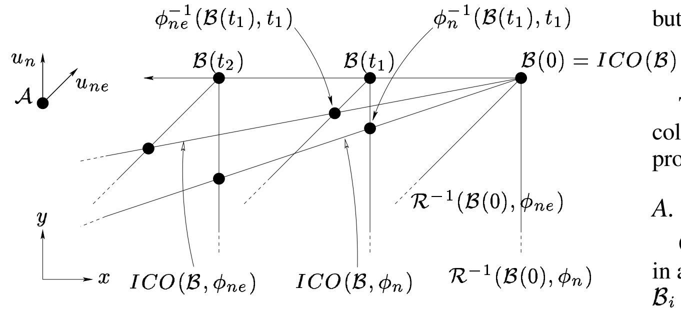

Fig. 6. Inevitable collision obstacle for a moving point obstacle.

1) Point Obstacle: Let $\mathcal{B}$ be a moving point obstacle. $\mathcal{B}(t)$ gives the position of $\mathcal{B}$ at time t (it is also a state of $\mathcal{A}$ at time t). In order to characterise $ICO(\mathcal{B})$, we proceed step by step as we did in the fixed obstacle case. Given a control input ϕ, let us characterise $ICO(\mathcal{B}, \phi)$ first. In time-state space, since $\mathcal{B}$ is a moving obstacle, we can write $\mathcal{B} = \bigcup_t \mathcal{B}(t)$. Therefore $ICO(\mathcal{B}, \phi) = \bigcup_t ICO(\mathcal{B}(t), \phi)$. Now, according to definition 2, $ICO(\mathcal{B}(t), \phi)$ is the set of states s such that if $\mathcal{A}$ starts from s (at time 0) and is subject to the control input ϕ, it reaches the state $\mathcal{B}$(t) (at time t). Such a state s belongs to $\mathcal{R}^{-1}(\mathcal{B}(t), \phi)$ and it is actually the unique solution of the equation $\phi(s,t) = \mathcal{B}(t) \Leftrightarrow s = \phi^{-1}(\mathcal{B}(t), t)$. In conclusion, $ICO(\mathcal{B}, \phi) = \bigcup_t \phi^{-1}(\mathcal{B}(t), t)$ and we have:

$$
\begin{aligned}
ICO(\mathcal{B}) &\stackrel{1}{=} ICO(\mathcal{B}, \phi_n) \cap ICO(\mathcal{B}, \phi_{ne}) \\
&= ICO(\textstyle\bigcup_t \mathcal{B}(t), \phi_n) \cap ICO(\bigcup_t \mathcal{B}(t), \phi_{ne}) \\
&\stackrel{2}{=} \textstyle\bigcup_t ICO(\mathcal{B}(t), \phi_n) \cap \bigcup_t ICO(\mathcal{B}(t), \phi_{ne}) \\
&= \textstyle\bigcup_t \phi_n^{-1}(\mathcal{B}(t), t) \cap \bigcup_t \phi_{ne}^{-1}(\mathcal{B}(t), t)
\end{aligned}
$$

Consider Fig. 6 where it is assumed that $\mathcal{B}$ has a linear motion at constant velocity. For both control inputs ϕ_n and ϕ_{ne}, $ICO(\mathcal{B}, \phi)$ is a linear curve originating at $\mathcal{B}(0)$ whose slope depends upon the relative velocities of $\mathcal{A}$ and $\mathcal{B}$. The application of property 1 yields $ICO(\mathcal{B}) = \mathcal{B}(0)$.

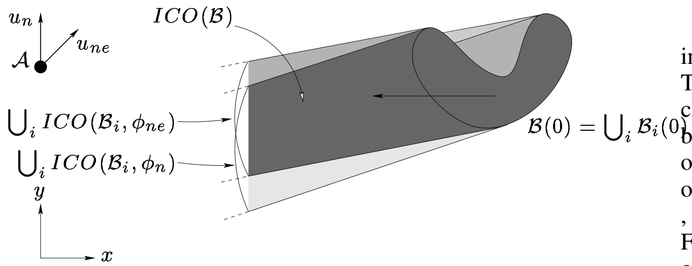

Fig. 7. Inevitable collision obstacle for a moving arbitrary obstacle.

2) Arbitrary Obstacle: Let us now assume that $\mathcal{B}$ is a moving obstacle of arbitrary shape. $\mathcal{B}$ is the union of a set of moving point obstacles and we can write: $\mathcal{B} = \bigcup_i \bigcup_t \mathcal{B}_i(t)$. $ICO(\mathcal{B})$ is derived in the same way as before, *ie* using both properties 1 and 2 plus the result concerning the moving point case presented earlier. Fig. 7 depicts the inevitable collision obstacle obtained for an arbitrary obstacle with a motion at constant velocity similar to that of the point obstacle in §IV-C.1. Whenever $\mathcal{A}$ is inside the region $ICO(\mathcal{B})$ *at time* 0, no matter what it does in the future, it eventually collides with $\mathcal{B}$. Note that, using the same method, one can determine the inevitable collision time-state at an arbitrary time instant in the future.

This simple example has illustrated how, thanks to the inevitable collision state concept, it is possible to characterise forbidden regions of the state-space, the inevitable collision obstacles, this characterisation taking into account both the dynamics of the robotic system but also the future behaviour of the moving obstacles.

V. Safe Motion Planning Application

The purpose of this section is to demonstrate how the inevitable collision state concept can be used to address safe motion planning problems.

A. Statement of the Problem

Consider the problem of planning motions for a vehicle $\mathcal{A}$ moving in a partially known environment that contains a set of fixed obstacles $\mathcal{B}_i$ whose position is a priori known. It also contains *unexpected obstacles*, fixed or moving, whose position is not known beforehand. Finally, $\mathcal{A}$ is subject to sensing constraints, it has a limited field of view. In a given state s, $\mathcal{A}$ perceives only a subset *FoV*(s) of its environment (Fig. 8, left).

In this framework, what does planning a safe motion mean? Safe motions were defined earlier as motions for which it is guaranteed that, no matter what happens at execution time, the vehicle never finds itself in a situation where there is no way for it to avoid collision with an unexpected obstacle.

At execution time, an unsafe situation occurs when an unexpected obstacle suddenly appears in the field of view of $\mathcal{A}$ and $\mathcal{A}$ suddenly finds itself in an inevitable collision state. Safety is therefore closely related to the limited field of view.

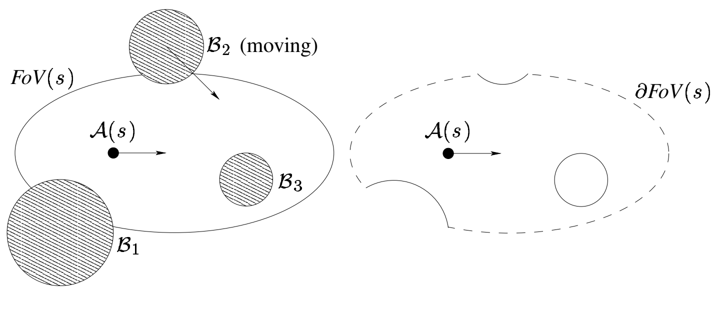

Fig. 8. The field of view of $\mathcal{A}$ (left) and its boundary (right).

At planning time, it is by definition impossible to characterise the inevitable collision states with respect to the unexpected obstacles. This characterisation can be done with respect to the known obstacles only. However, given that unexpected obstacles appear on the boundary of the field of view, something can be done! The boundary of the field of view has two parts: the part corresponding to known obstacles, and the part corresponding to the limit of the field of view , *ie* the sensing range, *eg* the dashed curve in the right-hand side of Fig. 8. Let $\partial FoV(s)$ denote this part. What can be done then is to consider $\partial FoV(s)$ as a potential unexpected obstacle and to determine whether the corresponding state is an inevitable collision state based on this assumption.

This is the key to safe motion planning. A safe motion is a sequence of safe states where a safe state s is defined as a state which is not an inevitable collision state with respect to the known obstacles $\mathcal{B}_i$, and with respect to $\partial FoV(s)$ treated as an unexpected obstacle, in other words:

*Def. 3 (**Safe State**):* s is *safe state* iff $s \notin ICO(\partial FoV(s))$ and $s \notin ICO(\mathcal{B}_i)$ for every known obstacle $\mathcal{B}_i$.

The next sections present a worked-out example of safe motion planning for a car-like vehicle in a partially known environment with fixed obstacles.

B. Car-Like Vehicle

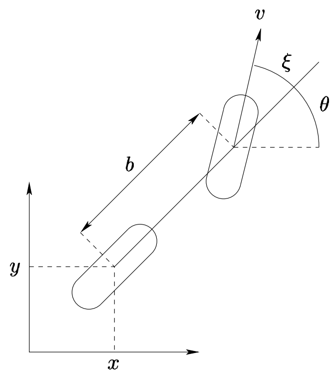

Fig. 9. The car-like vehicle $\mathcal{A}$ (bicycle model).

$\mathcal{A}$ is a car-like vehicle, it is modelled as a bicycle (Fig. 9). A state of $\mathcal{A}$ is defined by the 4-tuple $s = (x, y, \theta, v)$ where (x, y) are the coordinates of the rear wheel, θ is the main orientation of $\mathcal{A}$, and v is the linear velocity of the front wheel. A control of $\mathcal{A}$ is defined by the couple (ξ, a) where ξ is the steering angle and a the linear acceleration. The motion of $\mathcal{A}$ is governed by the following differential equations:

$$\begin{cases} \dot{x} & = & v \cos\theta \cos\xi \\ \dot{y} & = & v \sin\theta \cos\xi \\ \dot{\theta} & = & v \sin\xi / b \\ \dot{v} & = & a \end{cases}$$

with $|\xi| \leq \xi_{\max}$ and $|a| \leq a_{\max}$. b is the wheelbase of $\mathcal{A}$.

C. Inevitable Collision Obstacles

A prerequisite to safe motion planning is to have a characterisation of the inevitable collision states for $\mathcal{A}$, or similarly, a characterisation of the inevitable collision obstacles. The car-like vehicle $\mathcal{A}$ is unfortunately much more complicated a system than the "North, North-East" one. Chiefly, the fact that the number of possible control inputs for $\mathcal{A}$ is infinite makes it difficult to use property 1 directly in order to compute the inevitable collision obstacles.

Fortunately, it is possible to take advantage of property 4 in order to compute a conservative approximation of the inevitable collision obstacles (conservative in the sense that the actual inevitable collision obstacle is included in the approximated one). To do so, property 1 is applied considering a subset of the whole set of possible control inputs.

The subset $\mathcal{I}$ we have chosen contains the control inputs ϕ with a constant steering angle ξ (the acceleration a is allowed to change). Given an obstacle $\mathcal{B}$, the corresponding approximated inevitable collision obstacle $ICO(\mathcal{B})$ is thus defined as: $ICO(\mathcal{B}) = \bigcap_{\mathcal{I}} ICO(\mathcal{B}, \phi)$.

Thanks to this restriction, characterising $ICO(\mathcal{B})$ is straightforward. Due to lack of space, we simply illustrate how it is done and we do so in a step by step manner by considering different families of

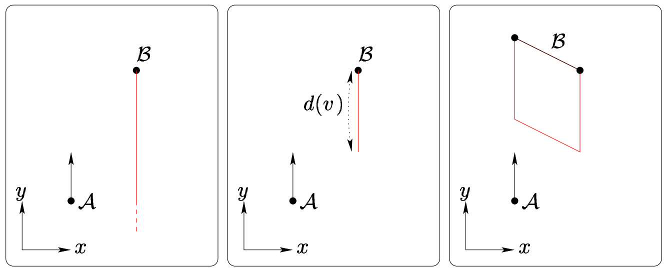

Fig. 10. $ICO(\mathcal{B}, \phi)$ for ϕ such that $\xi = 0$ ($\mathcal{A}$ is moving straight). $a = 0$ (left), a is changing (middle and right).

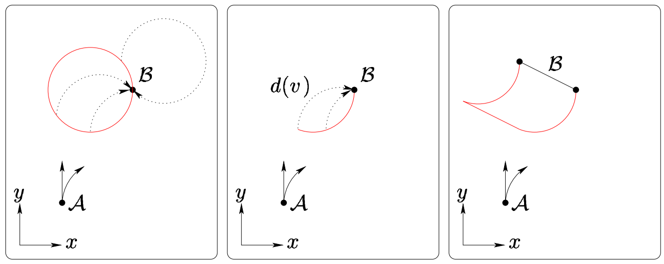

Fig. 11. $ICO(\mathcal{B}, \phi)$ for ϕ such that $\xi \neq 0$ ($\mathcal{A}$ is turning with a constant steering angle). $a = 0$ (left), a is changing (middle and right).

control inputs ϕ. First, Fig. 10 depicts the case where ϕ is such that $\xi = 0$ ($\mathcal{A}$ is moving straight). Then Fig. 11 depicts the case where ϕ is such that $\xi \neq 0$ ($\mathcal{A}$ is turning with a constant steering angle). Finally, Fig 12 depicts how $ICO(\mathcal{B})$ is obtained. Note that what is actually represented on the right-hand side of Fig 12 is only a "slice" of $ICO(\mathcal{B})$. Recall that $ICO(\mathcal{B})$ is defined in the 4-dimensional state-space of $\mathcal{A}$. The slice depicted is the $\{\theta = \pi/2, v\}$-slice. When $\mathcal{A}$ has an orientation $\pi/2$ and a velocity v, it inevitably crashes against $\mathcal{B}$ as soon as it is located in the region $ICO(\mathcal{B})$ depicted. The slices for other values of θ and v are obtained similarly.

D. Safe Motion Planning

Thanks to the results presented above, it is now possible to determine whether a state is safe or not. As far as solving the motion planning problem at hand is concerned, it was decided to use a 'classical' motion planning scheme based on Rapidly-Exploring

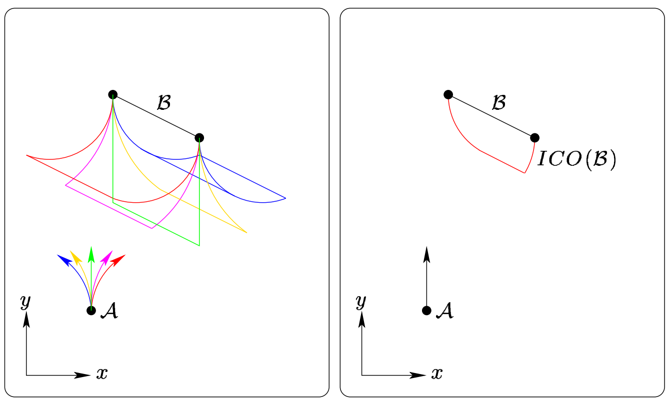

Fig. 12. $ICO(\mathcal{B}, \phi)$ for a number of control inputs $\phi \in \mathcal{I}$ with different ξ values (left). $ICO(\mathcal{B}) = \bigcap_{\mathcal{I}} ICO(\mathcal{B}, \phi)$ (right).

Random Trees [15]. Such an algorithm explores the state space by incrementally expanding a tree rooted at the initial state. The tree is expanded through elementary motions in randomly selected directions. Such an algorithm is very efficient at exploring high-dimensional spaces.

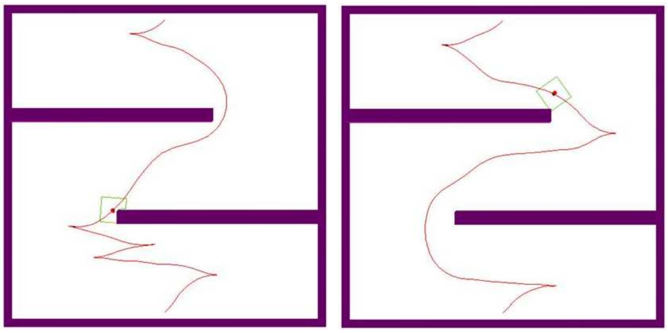

Fig. 13. Safe motion planning: preliminary results.

Fig. 13 presents some preliminary safe motion planning results obtained for the car-like vehicle $\mathcal{A}$ (more results can be found in [16]). The field of view of $\mathcal{A}$ is a rectangular area (visible at a state along the result trajectories).

In the left part of Fig. 13, the trajectory obtained is collision-free only (the sensing constraints and the possible presence of unexpected fixed obstacles is not taken into account). In the right part of Fig. 13, the trajectory obtained is collision-free too but it is also safe, *ie* it is a sequence of safe states (in the sense of Def. 3). It does take into account the limits of the field of view and the possible presence of unexpected fixed obstacles.

Remember that the exploration scheme is random. It accounts for the strange twists and turns of the trajectories obtained. However, it can be noticed how the safe trajectory does not graze the obstacles (especially near the end of the two walls). This makes perfect sense: suppose you have to pass the corner of a wall. The wall prevents you from seeing what is on the other side of the corner. So, if you believe that there may be unexpected obstacles on the other side, you have two strategies possible:

1) Graze the corner while slowing down so that when you pass the corner, your speed is slow enough for you to stop before hitting a possible unexpected obstacle, or
2) Stay away from the corner so as to have a better view of what is on the other side. In this case, you do not have to slow down.

In our experiments, the goal was to optimise the time of the trajectory. It naturally resulted in a solution trajectory following the second strategy and the trajectory obtained is safe. At execution time, no matter how many unexpected fixed obstacles are placed in the environment, it is guaranteed that, when such an unexpected obstacle is detected, $\mathcal{A}$ is not in an inevitably collision state, it can avoid the unexpected obstacle.

Future experiments will concern the safety with respect to unexpected moving obstacles. In this case, it is necessary to have some a priori knowledge about the moving obstacles, *eg* the maximum speed they can have, their expected motion direction, etc. This information is required to compute the inevitable collision obstacle corresponding to a moving obstacle (*cf* §IV-C).

VI. Conclusion

This paper has introduced the novel concept of inevitable collision states for a given robotic system, *ie* states for which, no matter what the future trajectory followed by the system is, a collision eventually occurs with an obstacle of the environment. In terms of collision, an inevitable collision state takes into account both the dynamics of the robotic system and the obstacles, fixed and moving.

The main contribution of this paper was to lay down and explore this novel concept (along with a companion concept, that of inevitable collision obstacle). A formal definition of what inevitable collision states and obstacles are was given. Properties that are fundamental for their characterisation were established. This concept is very general and it can be useful both for navigation and motion planning purposes. To illustrate the interest of this concept, an example of its application to safe motion planning was given.

In the future, it is intended to apply this concept to different robotic systems placed in different kinds of environment (with moving obstacles in particular). The safe motion planning issue remains to be explored in more details (this issue is related to the dependency that exists between motion planning and navigation). The application of this concept for navigation purposes needs to be explored too.

References

[1] T. Lozano-Perez, “Spatial planning, a configuration space approach,” *IEEE Trans. on Computing.*, vol. 32, no. 2, pp. 108–120, Feb. 1983.

[2] J.-C. Latombe, *Robot motion planning*. Kluwer Academic Press, 1991.

[3] J. Canny, B. Donald, J. Reif, and P. Xavier, “On the complexity of kynodynamic planning,” in *Proc. of the Symp. on the Foundations of Computer Science*, White Plains, NY (US), Nov. 1988, pp. 306–316.

[4] P. Xavier, “Provably-good approximation algorithms for optimal kinodynamic robot motion plans,” Ph.D. dissertation, Cornell Univ., Ithaca, NY (US), Apr. 1992.

[5] T. Fraichard, “Dynamic trajectory planning with dynamic constraints: a ‘state-time space’ approach,” in *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 2, Yokohama (JP), July 1993, pp. 1394–1400.

[6] ——, “Trajectory planning in a dynamic workspace: a ‘state-time’ approach,” *Advanced Robotics*, vol. 13, no. 1, pp. 75–94, 1999.

[7] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, “Randomized kinodynamic motion planning with moving obstacles,” in *Proc. of the Workshop on the Algorithmic Foundations of Robotics*, Hanover, NH (US), Mar. 2000.

[8] S. LaValle and J. Kuffner, “Randomized kinodynamic planning,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 1, Detroit, MI (US), May 1999, pp. 473–479.

[9] R. Simmons, “The curvature-velocity method for local obstacle avoidance,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Minneapolis, MN (US), Apr. 1996, pp. 3375–3382.

[10] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23–33, Mar. 1997.

[11] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using velocity obstacles,” *Int. Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, July 1998.

[12] N. Y. Ko and R. Simmons, “The lane-curvature method for local obstacle avoidance,” in *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Victoria, BC (CA), Oct. 1998.

[13] O. Brock and O. Khatib, “High-speed navigation using the global dynamic window approach,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Detroit, MI (US), May 1999.

[14] R. Alami, T. Siméon, and K. Madhava Krishna, “On th einfluence of sensor capacities and environment dynamics onto collision-free motion plans,” in *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Lausanne (CH), Sept.-Oct. 02.

[15] S. Lavalle, “Rapidly-exploring random trees: A new tool for path planning,” Dept. of Computer Science, Iowa State University, Ames, IA (US), Research Report 98-11, Dec. 1998.

[16] S. Blondin, “Planification de mouvements pour véhicule automatisé en environnement partiellement connu,” Inst. Nat. Polytechnique de Grenoble, Grenoble (FR), Mémoire de Diplôme d’Etudes Approfondies, June 2002.