

Passively Safe Partial Motion Planning for Mobile Robots with Limited Field-of-Views in Unknown Dynamic Environments

Sara Bouraine, Thierry Fraichard, Ouahiba Azouaoui, Hassen Salhi

▶ **To cite this version:**

Sara Bouraine, Thierry Fraichard, Ouahiba Azouaoui, Hassen Salhi. Passively Safe Partial Motion Planning for Mobile Robots with Limited Field-of-Views in Unknown Dynamic Environments. IEEE Int. Conf. on Robotics and Automation (ICRA), Jun 2014, Hong Kong, Hong Kong SAR China. hal-01018463

HAL Id: hal-01018463
https://inria.hal.science/hal-01018463

Submitted on 4 Jul 2014

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.

Passively Safe Partial Motion Planning for Mobile Robots with Limited Field-of-Views in Unknown Dynamic Environments

S. Bouraine[†], Th. Fraichard[‡], O. Azouaoui[†] and Hassen Salhi[⋆]

***Abstract*— This paper addresses the problem of planning the motion of a mobile robot with a limited sensory field-of-view in an unknown dynamic environment. In such a situation, the upper-bounded planning time prevents from computing a complete motion to the goal, partial motion planning is in order. Besides the presence of moving obstacles whose future behaviour is unknown precludes *absolute motion safety* (in the sense that no collision will ever take place whatever happens) is impossible to guarantee. The stance taken herein is to settle for a weaker level of motion safety called *passive motion safety*: it guarantees that, if a collision takes place, the robot will be at rest. The primary contribution of this paper is PASSPMP, a partial motion planner enforcing passive motion safety. PASSPMP periodically computes a passively safe partial trajectory designed to drive the robot towards its goal state. Passive motion safety is handled 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. Simulation results demonstrate how PASSPMP operates and handles limited sensory field-of-views, occlusions and moving obstacles with unknown future behaviour. More importantly, PASSPMP is provably passively safe.**

I. INTRODUCTION

Nowadays, the development of autonomous mobile robots is the key issue of several new applications from industry, public transportation to spatial and underwater exploration. Such applications take place in the real world where the term real world implies that:

1) The environment is dynamic and uncertain, it features fixed and moving obstacles whose future behaviour is unknown.
2) The robot has only a partial knowledge of its surroundings because of its sensory limitations.

In order to carry out its task successfully in such situations, an autonomous mobile robot should be able to navigate around while guaranteeing its safety and that of its environment. As of now, the constraints imposed on navigation in dynamic environments are clearly established. As per [1], they can be summarized as follows:

> *In a dynamic environment, one has a limited time to make a motion decision, one has to reason about the future evolution of the environment and do so with an appropriate lookahead*[1].

From a motion safety point of view now, the presence of moving obstacles whose future behaviour is unknown precludes *absolute motion safety* (in the sense that no collision will ever take place whatever happens) is impossible to guarantee [2]. To address that issue, authors such as [3] have advocated the introduction of weaker levels of motion safety level arguing that it is better to guarantee less than to guarantee nothing. It is precisely the position we took in [4] where it was settled for a motion safety level called *passive motion safety* that guarantees that, if a collision takes place, the robot will be at rest. [4] introduced a navigation scheme called PASSAVOID for mobile robots with limited field-of-views placed in unknown dynamic environments. PASSAVOID is reactive in nature, 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. [4] formally establishes that PASSAVOID is provably passively safe.

This paper builds upon [4] and presents an extension of PASSAVOID called PASSPMP. PASSPMP operates according to the Partial Motion Planning (PMP) principle that was introduced in [5]. PMP is motion planning sheme that fulfills the constraints imposed by dynamic environments (limited decision time), and attempts to bridge the gap between reactive approaches (that computes the control to apply at the next time step) and deliberative motion planning approaches (that seeks to compute a complete motion to the goal). At each cycle, PMP expands a search tree rooted at the current state of the robot and when the time available is over, PMP extracts from the tree the best partial motion towards the goal computed so far. The process is repeated until the goal is reached. PASSPMP is based upon the Rapidly-exploring Random Tree (RRT) algorithm [6]. PASSPMP can handle limited field-of-views, occlusions and unknown future behavior of the obstacles. Because, it is PMP-based, PASSPMP has more insights than PASSAVOID. However, similarly to PASSAVOID, PASSPMP is provably passively safe.

The paper is organized as follows. In Section II, an overview of the most relevant works is presented. Section III explains the passive motion safety concept that must be guaranteed by PASSPMP which is later described in Section IV. Finally, the simulation results are given in Section V.

II. RELATED WORKS

Motion safety have been analyzed thanks to the *Inevitable Collision States* (ICS) concept developed in [7]. An ICS is a state for which, no matter what the future trajectory of the robot is, a collision eventually occurs. The key to motion safety is clearly to stay away from ICS. However characterizing the ICS set corresponding to a given current situation is

[†]CDTA (DZ); [‡]INRIA, CNRS-LIG and Grenoble Univ. (FR); [⋆]Blida Univ. (DZ).

[1]The lookahead, *a.k.a.* time horizon, is how far into the future the reasoning is done.

in general challenging since it requires information about the future evolution of the environment with a possibly infinite lookahead. In light of the ICS concept, it appears that, although the robotics literature is rich in works concerned with collision avoidance and safe navigation, most of them do not offer an explicit formulation of the safety guarantee they provide or the conditions under which they must operate [8]. Among them, the most interesting are those that acknowledge the difficulty of getting a meaningful characterization of the ICS set in real-world situation. To cope with that issue, these approaches relies upon a relaxation of the ICS concept: they seek to 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. Example of this kind of approaches can be found in [9], [5], [10], [11].

In order to address the uncertainty that prevails in the real-world, in particular the uncertainty concerning the future behaviour of the moving obstacles, probabilistic version of the ICS concept have been proposed, *e.g.* [12] and [13]. Such approaches are interesting but they do not allow strict motion safety guarantee, they allow instead to minimize the risk.

Concerning sensory limitations, there are only a few research works taking them into account. For instance, the occlusion problem, *i.e.* the existence of regions that are hidden by other obstacles, is addressed in a coarse manner in [14] and in a more principled manner in [15]. The occlusion and the limited field-of-view problems are addressed in [7] and [16]. [7] addresses the case of a mobile robot moving in a static environment; its approach is general and ICS-based. While [16] considers dynamic environments, it does so primarily with a path-velocity decomposition perspective [17].

PASSPMP, the contribution of this paper is an extension of PASSAVOID [4]. PASSPMP deals with limited field-of-views, occlusions and unknown future behavior of the obstacles. Because, it is PMP-based, PASSPMP is not purely reactive, it has more insights. However, similarly to PASSAVOID, PASSPMP is provably passively safe.

III. PASSIVE MOTION SAFETY

Generally, absolute motion safety is impossible to guarantee in dynamic environments given the unknown future behavior of moving objects and with a robot having a limited field-of-view (presence of occlusions, unseen objects, etc.). These harsh constraints impose an alternative solution which is to settle for weaker level of motion safety by guaranteeing at least, if a collision takes place the robot will be at rest. This form of motion safety is dubbed passive motion safety.

Introducing this level of motion safety gives a relaxation to the original ICS concept of [7] namely the braking ICS concept introduced in [4]. A braking ICS (denoted ICS^b) is a state for which no matter what the future trajectory of the robot is, a collision takes place before the robot is at rest. With duality to passive motion safety concept, a state which is braking ICS-free is passively safe (p-safe).

In [4], an efficient Braking ICS-Checker (called ICS^b-CHECK) was designed, it checks whether a given state trajectory is braking ICS or not (i.e. p-safe or not) for a given model of the future. This algorithm is used in the passively safe planner (PASSPMP) (presented in §IV) guaranteeing the generation of p-safe partial trajectories.

However, Braking ICS is defined given an appropriate look-ahead (time horizon) because motion safety guarantee requires to reason about the future evolution of the environment. For absolute motion safety, the lookahead is infinite which is impossible to verify unless the environment is a priori known. But for passive motion safety a finite lookahead called time horizon should be set.

If T_h is a valid lookahead, then computing ICS^b requires to consider the model of the future up to this time.

This notion of lookahead will step in PASSPMP cycles, where an updated model of the future is required.

IV. PASSIVELY SAFE PLANNING

In this paper, the aim is to build a passively safe trajectory from an initial state to a given goal, verifying both safety criterion and best convergence towards the goal. The proposed method uses a replanning process that interleaves planning with execution so that the robot may compute partial plans leading to a passively safe partial motion planner. This planner is developed for a mobile robot $\mathcal{A}$ with a limited field-of-view placed in an unknown dynamic environment. $\mathcal{A}$'s motion is governed by differential equations of the form:

$$\dot{s} = f(s, u) \tag{1}$$

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 $\mathcal{U}$ are respectively the state space and the control space of $\mathcal{A}$.

This planner aims to drive $\mathcal{A}$ from its initial position until it reaches the goal guaranteeing passive motion safety no matter what happens in the environment. In other words, it guarantees two conditions:

1) If a collision takes place, it is guaranteed that $\mathcal{A}$ will be at rest when it occurs (passive safety).
2) $\mathcal{A}$ chooses the optimal trajectory to reach the goal.

To summarize, our purpose is to design a passively safe navigation scheme based on a PMP and guaranteeing passive safety. To do so, a number of points must be verified:

- Kinematic and dynamic constraints of the robot.
- Reasoning about the future.
- React for a time horizon.
- Passive safety guarantee (based on ICS^b-CHECK).
- Drive $\mathcal{A}$ to the goal.

A. Partial Motion Planning (PMP) Principle

Robot navigation requires the interleaving of sensing, decision-making and execution. In an unknown and dynamic environment, it is necessary to continuously sense the environment and regularly update the world model upon which navigational decisions are made. Likewise, the decision-making module must be called frequently and has a finite

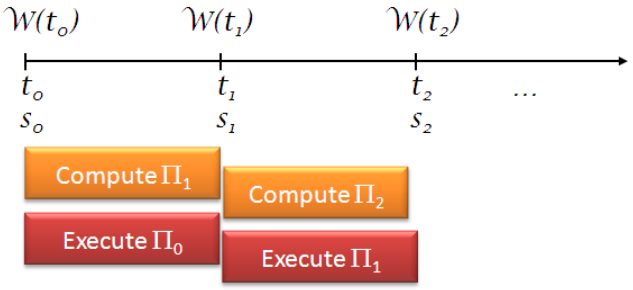


Fig. 1: Timeline of the decision-making and execution processes in a partial motion planner.

time to decide what to do next. In our case, this decision-making module is based upon a partial motion planner. Instead of global motion planners that compute a complete trajectory to the goal, PMP computes partial trajectories. The set of these trajectories drives the robot to the goal. Irrespective of how the decision-making module works, the bottom line is that a trajectory must be available at each decision-making cycle.

The timeline of the decision-making process of $\mathcal{A}$ is illustrated in Fig. 1. At time t_0, $\mathcal{A}$ is in state s_0, it is currently executing a trajectory Π_0 that will drive it to the state s_1 at time t_1, the next cycle time. Let $\mathcal{W}(t_0)$ denotes the world model which is available at time t_0. During the time interval $[t_0, t_1]$, the decision-making module has to compute the trajectory Π_1 that will be executed during the next cycle time $[t_1, t_2]$. At time t_1, the process is repeated based on $\mathcal{W}(t_1)$, the updated model of the environment, and so forth.

From a passive motion safety perspective, the braking ICS concept brings to light two things: the first one is that, it is not sufficient to consider that all the trajectories Π_i are collision-free; they must be braking ICS-free meaning that, at each time instant, $\mathcal{A}$ must always be in a state which is not a braking ICS (if a collision occurs, the robot is at rest). The second one is that, braking ICS-ness is always defined *wrt* the used world model (in particular the model of the future).

Regarding PMP process above, it should be noted that Π_1 is computed using the world model $\mathcal{W}(t_0)$. It can be computed so as to be braking ICS-free but if at time t_1 (when the updated world model $\mathcal{W}(t_1)$ is available), s_2 turns out to be a braking ICS *wrt* $\mathcal{W}(t_1)$, collision occurs. This remark highlights the fact that reasoning about the future evolution of the environment is critical when it comes to motion safety in dynamic environments. Therefore, reasoning about a limited time horizon (lookahead) (T_h) during which safety is guaranteed, is required. In this case, the world model $\mathcal{W}(t_0)$ will be valid during the time interval $[t_0, t_1[\cup[t_1, T_h]$ (with $T_h > t_2$). Consequently, it is always guaranteed that s_2 is braking ICS-free. Introducing this notion of passive motion safety in PMP process leads to the new version PASSPMP, described in next section.

B. Passively Safe PMP (PASSPMP)

PASSPMP is a partial motion planner that operates within

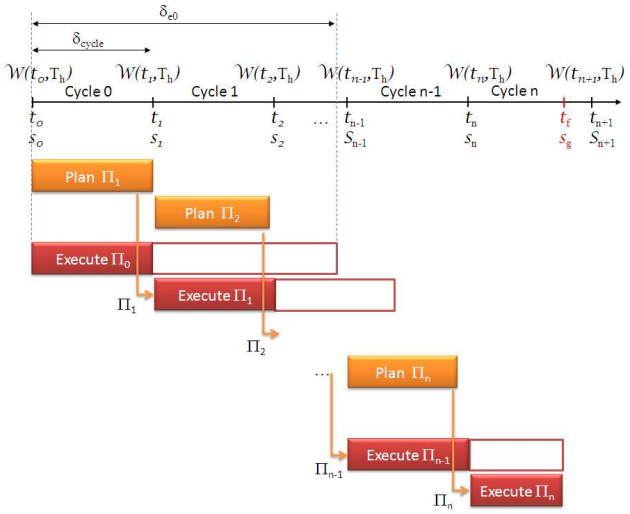


Fig. 2: PASSPMP process.

a given time cycle δ_{cycle}. During this time, a p-safe partial trajectory is computed. It verifies two conditions: (1) corresponding states must be p-safe (*i.e.* ICS^b-free) and (2) convergence towards the desired goal. This trajectory has an execution time duration δ_e. However, only the part corresponding to the duration δ_{cycle} is executed because at the end of each cycle a new computed partial trajectory is available (see Fig. 2). δ_{cycle} should be set carefully in a way that passive safety still always guaranteed. The planning cycle has to return a decision within a bounded time, which depends upon the current model of the environment (model of the future). It is the minimum time to collision named decision time δ_d. Thus, the time cycle must be always lower than this decision time:

$$\delta_{cycle} < \delta_d \tag{2}$$

Assuming that $\mathcal{A}$ is equipped with range sensors such as laser telemeters or range cameras, it can only perceive a subset of the workspace $\mathcal{WS}$; this subset is $\mathcal{A}$'s field-of-view; its shape is arbitrary and its range is $360°$. It represents the region of $\mathcal{WS}$ which is free of objects at the sensing time while the remaining regions of $\mathcal{WS}$ represent the space where there might be objects that can be fixed or moving, seen or unseen. Motion safety requires reasoning about the future motion of the objects in the environment. As it is assumed that $\mathcal{A}$ cannot distinguish between fixed and moving objects (every object observed is treated as a potentially moving object) and that it has no information about their future behavior, the model of the future used herein is conservative: given an upper-bound on the velocity of the objects, every point in the region outside the field-of-view is modeled as a disc that grows as time passes, i.e. a cone in spacetime.

However, as time passes the region of $\mathcal{WS}$ occupied by this growing discs grows too. This is why, the model of the future may be valid up to a finite time; hence an appropriate time horizon (T_h) should be set.

The braking ICS concept considers braking trajectories $\tilde{u}_b$

of finite duration t_b , with t_b the braking time of $\tilde{u}_b$ (braking trajectory definition is given in [4]). t_b is the ratio of the robot's linear velocity and its maximum linear acceleration. For an arbitrary subset $\mathcal{E}$ of the whole set of possible braking trajectories, a finite time horizon T_h exists:

$$T_h = \max_{\tilde{u}_b \in \mathcal{E}} \{t_b\} \tag{3}$$

For motion safety guarantee, it suffices to consider the model of the future up to time T_h. This time denotes how far into the future, objects behavior has been modeled and predicted.

As the executed trajectory during the current PASSPMP's cycle is based on the previous PASSPMP's cycle result, the decision time δ_d cannot exceed the time horizon (otherwise $\mathcal{A}$ would run out of a valid plan for the next execution cycle). However, there is no lower bound for the decision time, it depends on the complexity of the motion planning exploration itself. In fact, we set:

$$\delta_d << T_h \tag{4}$$

Besides, to guarantee that the executed trajectory is still valid, the following condition must be verified:

$$\delta_d + \delta_e \leq T_h \tag{5}$$

where δ_e is the execution time. As a conclusion, PASSPMP cycle is set as a fixed period of time in order to regularly get an update of the environment. Given equations 2 and 5, and knowing that the world model $\mathcal{W}$ is valid during the time interval $[t_k, t_{k+1}] \cup [t_{k+1}, T_h]$ (with $T_h > t_{k+2}$ and $T_h > 2\delta_{cycle}$), the following condition could be set to guarantee passive motion safety:

$$\delta_{cycle} < min(\delta_d, \frac{1}{2} T_h) \tag{6}$$

This condition guarantees that the time horizon is still valid for both trajectory planning and execution.

All the above conditions are important for safety guarantee in the PASSPMP process. However, guarantying passive motion safety requires proving that a collision will never occur while $\mathcal{A}$ is moving. Thus, it must be proved that, at each time cycle PASSPMP would always find a partial trajectory that is p-safe. To that end, some properties are required and henceforth introduced in what follow. Let first define what is a p-safe state.

Def. 1 (P-Safe State): A state s_0 is *passively safe* or *p-safe* (it is not a braking ICS) if it exists at least one braking trajectory $\tilde{u}_b$ starting at s_0 which is collision-free until $\mathcal{A}$ stops.

*Property 1 (**P-Safe trajectory**):* For a given state trajectory Π between s_0 and s_j, if (1) the state trajectory between s_0 and s_j is collision-free (with respect to the conservative model of the future) and (2) s_j is p-safe, then the states belonging to Π are p-safe, therefore, Π is p-safe.

Proof: At first, we define a state s_i, with $0 < i < j$ belonging to a state trajectory between s_0 and s_j, Let assume that: (1) s_2 is p-safe, *i.e.* it exists at least one braking trajectory starting at s_2 which is collision-free until $\mathcal{A}$ has stopped. (2) s_1 is not p-safe, *i.e.* whatever existing braking trajectories, collision occurs (no collision-free braking trajectory).
However, as the state trajectory between s_0 and s_j is collision-free and there exists a braking trajectory for the state s_2, starting from s_1, $\mathcal{A}$ can brake down without a collision occurs. Therefore, s_1 is p-safe: contradiction with (2).

■

*Property 2 (**Passive Safety Guarantee**):* If the state s_0 is p-safe then there exists at least one p-safe state trajectory that drives $\mathcal{A}$ through states that are also p-safe.

Proof: According to definition 1, if s_0 is p-safe then there exists at least one braking trajectory $\tilde{u}_b$ (between s_0 and s_j) which is collision-free until $\mathcal{A}$ stops. At the state s_j, $\mathcal{A}$ is at rest then s_j is p-safe. Applying property 1, since the braking trajectory $\tilde{u}_b$ is collision-free and s_j is p-safe, $\tilde{u}_b$ is henceforth p-safe and every state of this trajectory is also p-safe. Let $\tilde{u}_b$ is a special case of possible trajectories, then it is proved that there exists a p-safe trajectory.

■

Property 2 allows designing a version of PASSPMP that is passively safe *i.e.* at each planning cycle, it is guaranteed that PASSPMP can plan a p-safe partial trajectory to be executed in next cycle. It is an iterative process, repeated until $\mathcal{A}$ reaches its goal.

Algorithm 1 describes PASSPMP behavior in a k cycle.

Algorithm 1 PASSPMP.

Input: Model of the future $\mathcal{W}(t_k, T_h)$, goal s_g, partial trajectory Π_k to execute.
Output: The selected partial trajectory Π_{k+1} to execute in cycle $k+1$.

1: TREE=EXPLORE_STATE_TIME($\mathcal{W}(t_k, T_h)$,s_{k+1}); // Explore the state time space of $\mathcal{A}$, $t \in [t_{k+1}, t_{k+2}]$
2: Π_{k+1}=SELECT_BEST_PTRAJ(TREE,s_g); //Select the best partial trajectory.

Algorithm 1 takes as input the model of the future $\mathcal{W}(t_k, T_h)$ which is available at time t_k and valid up to time T_h (*i.e.* sufficient to plan the partial trajectory Π_{k+1} during the current cycle k and to execute it during the next cycle $k+1$). The algorithm takes also as input the goal to reach, and the partial trajectory computed in the previous cycle $k-1$ to be executed during the current cycle. PASSPMP algorithm operates by first exploring the state time space of $\mathcal{A}$ (with $t \in [t_{k+1}, t_{k+2}]$) through the function EXPLORE_STATE_TIME (line #1 of the algorithm). To do so, a diffusion technique is used, by extending a tree rooted at the state $s(t_{k+1})$ (with $t_{k+1} = t_k + \delta_{cycle}$). For reason of simplicity and facility of adaptation to a partial concept, the adopted technique is inspired from RRT proposed in [6]. As passive safety must be a criterion in trajectories selection, the state time space is explored using what we called p-safe RRT (which is detailed below). During this

step, the set of partial trajectories are computed and then defined with respect to passive safety guarantee (based on ICS^b-CHECK algorithm [4]). In step #2 of the algorithm, the function SELECT_BEST_PTRAJ selects the best partial trajectory verifying passive safety criterion and convergence towards the goal.

The algorithm is repeated iteratively until reaching the goal (s_g).

1) p-safe RRT: In order to explore different possible trajectories, a tree is grown in the state time space of $\mathcal{A}$ using RRT [6], where nodes represent robot states that are related with trajectory primitives. In PASSPMP concept, the expansion is done passively safe, hence the new version of RRT algorithm namely p-safe RRT algorithm (given in algorithm 2). This algorithm describes the function EXPLORE_STATE_TIME of PASSPMP algorithm (algorithm 1). It aims to expand the tree given the model of the future (future behavior of moving objects) and an initialized tree rooted at the initial state of the cycle to plan (s_{k+1}). An exhaustive search based expansion method is used. Each node is expanded according to a selected control space sampling set (line #7) using the function EXPANSION_WITH_SAFETY_CHECK (line #5). This function extends a node (state) s_i with a control u_j generating a new trajectory primitive that is checked for passive safety. Based on property 2, it is guaranteed that if s_{k+1} is p-safe, there exists always a p-safe partial trajectory. As s_{k+1} belongs to previous p-safe partial trajectory (from previous PASSPMP cycle), s_{k+1} is p-safe thus there exists always a p-safe partial trajectory until $\mathcal{A}$ reaches its goal. Using property 1, it is guaranteed that a trajectory primitive $\delta\Pi_{new} = \{s_i, ..., s_{new}\}$ is p-safe if $\delta\Pi_{new}$ is collision-free and s_{new} is p-safe. The state is checked to be p-safe or not using BRAKING_ICS_CHECK function (line #21 of algorithm 2), which is based on ICS^b-CHECK algorithm [4]. If property 1 is verified, the corresponding trajectory primitive ($\delta\Pi_{new}$) and expanded node (s_{new}) are added to the tree (lines #24 and #25 of algorithm 2). In case of no p-safe trajectory primitive is found for a given node s_i, the function GENERATE_BRAKING_TRAJ looks for a collision-free braking trajectory to guarantee that the robot will be at rest if a collision could occur. As the node s_i is p-safe it is guaranteed that it exists at least one braking trajectory $\tilde{u}$ starting at s_i which is collision-free until $\mathcal{A}$ has stopped (verifying passive motion safety property established in [4]).

2) Trajectory selection: In algorithm 1 (PASSPMP algorithm), the function SELECT_BEST_PTRAJ selects from the expanded tree (using algorithm 2) the best partial trajectory during the planning cycles. This selection is based on two conditions: (1) The passive motion safety of the partial trajectory. (2) Convergence towards the goal: a weighted cost function is computed based on a distance metric and time cost. Thus, the selected partial trajectory could be passively safe at first. Using algorithm 2, a p-safe partial trajectory is a concatenation of p-safe trajectory primitives. Doing so, many partial trajectory possibilities are available. To settle

Algorithm 2 p-safe RRT algorithm.

Input: Model of the future $\mathcal{W}(t_k, T_h)$, s_{k+1} (from Π_k), δ_{cycle} (with $\delta_{cycle} < min(\delta_d, T_h/2)$), initialize the tree: TREE $= \{s_0, ..., s_m\} \cup \{\delta\Pi_0, ..., \delta\Pi_{m-1}\}$ (corresponding set of nodes and set of trajectory primitives).
Output: TREE.

1: **while** $t < \delta_{cycle}$ **do**
2: //set of tree nodes
3: **for** i=0 to m **do**
4: //Select the control space sampling set U
5: Sample $\mathcal{U} \rightsquigarrow U = \{u_1 \dots u_n\}$
6: **Forall** $u_j \in U$ **do**
7: $(s_{new}, \delta\Pi_{new}) \leftarrow$ EXPANSION_WITH_SAFETY_CHECK($s_i, u_j, \mathcal{W}$, TREE);
8: **end**
9: **if** $s_{new} = \emptyset$ **then**
10: GENERATE_BRAKING_TRAJ();
11: **end if**
12: **end for**
13: **return** TREE
14: **end while**
15:
16: Procedure EXPANSION_WITH_SAFETY_CHECK (s_i,u_j,$\mathcal{W}$, TREE)
17: //Check p-safety
18: $\delta\Pi_{new} \leftarrow$ GENERATE_TRAJ_PRIM(s_i, u_j)
19: //$\delta\Pi_{new} = \{s_i, ..., s_f\}$
20: $s_{new} = s_f$;
21: **if** BRAKING_ICS_CHECK(s_{new},$\mathcal{W}$) = False **then**
22: //s_{new} is p-safe
23: **if** $\delta\Pi_{new}$ is collision-free **then**
24: TREE = TREE $\cup$ s_{new};
25: TREE = TREE $\cup$ $\delta\Pi_{new}$;
26: **return** s_{new},$\delta\Pi_{new}$
27: **end if**
28: **end if**
29: **EndProcedure**

which p-safe partial trajectory (between s_{k+1} and s_{k+2}) could be selected, a weighted cost function is computed. It is expressed as follows:

$$f_g(s_{k+2}) = w_d||s_{k+2} - s_g|| + w_t\Delta T_\Pi \quad (7)$$

The function f_g determines the best partial trajectory Π_{k+1} *i.e.* the optimal in distance and in time with respect to the goal. It depends on the Euclidean distance between the trajectory final state (s_{k+2}) and the goal state (s_g) associated to a distance weighting factor w_d (minimizing distance to the goal). Besides, this function depends on the trajectory cumulative time (ΔT_Π) associated to a time cost weighting factor w_t (minimizing time to the goal).

V. SIMULATION RESULTS

To demonstrate and validate PASSPMP's passive motion safety, it has been implemented and tested in simulation on

a scenario handling moving obstacles with unknown future behavior, for a mobile robot with a limited field-of-view. The partial motion can be described as a concatenation of several geometrical primitives.

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 $\mathcal{A}$, 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:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ v\tan\xi/L \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_\alpha + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_\xi \quad (8)$$

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

Simulation scenario features 22 objects moving arbitrarily in a 2D workspace. Their motion is unaffected by the other objects.

Fig. 3a shows an example of p-safe RRT construction for the first PASSPMP cycle. The tree is extended through the set of controls defined by a constant maximum linear acceleration $u_\alpha = u_{\alpha_{\max}}$ and a constant steering angle velocity $|u_\xi| \leq u_{\xi_{\max}}$ (namely, the control sets: $[u_\alpha, -u_\xi], [u_\alpha, 0], [u_\alpha, u_\xi]$). It is checked for passive motion safety using ICS^b-CHECK algorithm (line #7 of algorithm 2). As illustrated in Fig. 3a, trajectory primitives represented in blue are p-safe, while those represented in cyan are not p-safe. The tree is extended in a way that each node is extended with at least one p-safe trajectory primitive. In case of no p-safe trajectory primitive is found, the node is extended with a collision free braking trajectory (lines #9 and #10 of algorithm 2) to guarantee that the robot stops before collision occurs (passive motion safety guarantee). Hence, among a set of possible braking trajectories, 21 in this example, (represented in black), the selected braking trajectory is the one that is collision-free and closest to the goal (represented in magenta). During a planning cycle, the best partial trajectory is selected given the two conditions: (1) Passive motion safety of the partial trajectory and (2) the optimal both in distance and in time with respect to the goal (line #2 of algorithm 1). In case of the grown tree of Fig. 3a, the selected partial trajectory is represented in green (see Fig. 3b). To reach the goal in a passively safe manner, PASSPMP drives the robot following a set of concatenated partial trajectories (computed at different cycles). Fig. 4 shows grown trees along PASSPMP cycles until $\mathcal{A}$ reaches the goal.

The obtained results are also illustrated in a short film provided as a multimedia attachment to this paper. The corresponding velocity evolution of $\mathcal{A}$ is depicted in Fig. 5, where $\mathcal{A}$ exhibits the following behavior:

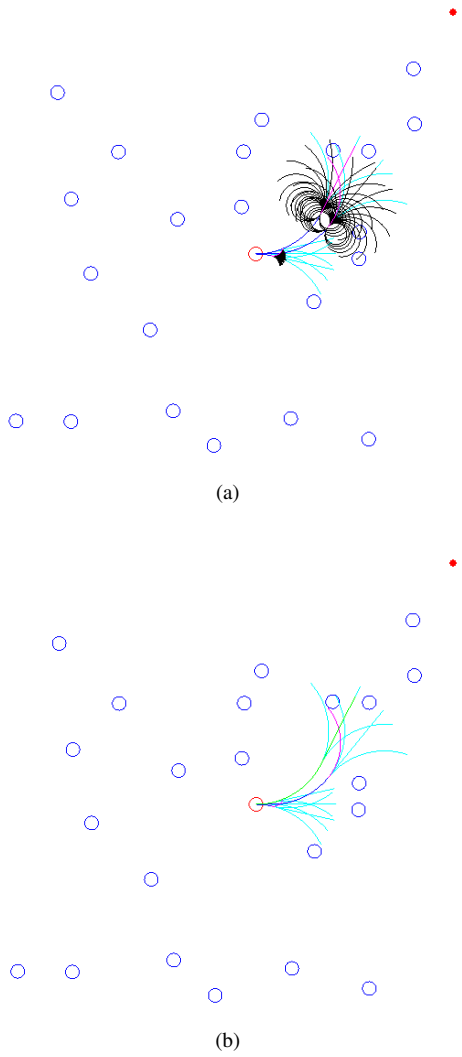


Fig. 3: p-safe RRT at work: (a) p-safe RRT construction during one PASSPMP cycle (b) corresponding selected partial trajectory (in green). The robot $\mathcal{A}$ is the disc at the center, the moving discs in blue are objects and the red mark is the goal to reach by $\mathcal{A}$

.

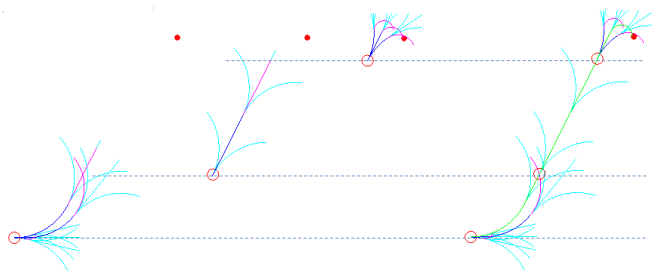


Fig. 4: p-safe RRTs corresponding to different PASSPMP cycles (left) and the overall selected trajectory represented in green (right).

1) $\mathcal{A}$ increases its velocity until it reaches the maximum value. PASSPMP keeps $\mathcal{A}$ moving in p-safe manner (no collision occurs).
2) at the end of the first cycle, even if $\mathcal{A}$ could move without risk of collision, it gradually reduces its velocity (until $v = 14\text{ms}^{-1}$). This is due to the limited field-of-view of $\mathcal{A}$: as unseen objects are considered (the presence possibility of an object forces $\mathcal{A}$ to brake down until it stops in order to keep $\mathcal{A}$ in p-safe state (if collision occurs $\mathcal{A}$ will be at rest).
3) at next cycle, a new plan is considered using a new update of the world model. $\mathcal{A}$ stops reducing its velocity and rises it again until the maximum value and maintains it.Once again it brake down (passive motion guarantee) until $v \simeq 3\text{ms}^{-1}$ (always due to the limited field-of-view consideration).
4) during the last cycle, $\mathcal{A}$ reaches the goal. It selects a collision-free braking trajectory that drives $\mathcal{A}$ to the closest state to the goal with a zero velocity ($\mathcal{A}$ stops) (see Fig. 4).

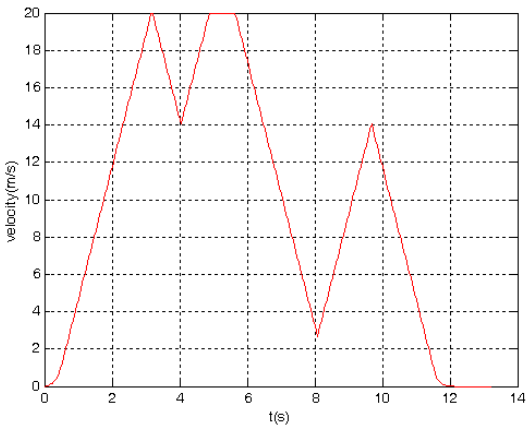


Fig. 5: Velocity profile of $\mathcal{A}$.

In the course of several runs, these experiments have demonstrated the capability of PASSPMP to drive the robot to a predefined goal when enforcing passive motion safety.

VI. Conclusion and Future Work

This paper has addressed the problem of planning the motion of a mobile robot with a limited sensory field-of-view in an unknown dynamic environment. A passively safe motion planner called PASSPMP has been presented: it is a partial motion planner enforcing passive motion safety. PASSPMP periodically computes a passively safe partial trajectory designed to drive the robot towards its goal state. Passive motion safety is enforced thanks to 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. Simulation results have demonstrated how PASSPMP operates and handles limited sensory field-of-views, occlusions and moving obstacles with unknown future behaviour. PASSPMP works better than PASSAVOID, its purely reactive counterpart that was presented in [4] however the analysis of the convergence towards the goal of PASSAVOID remains to be done. It could also be interesting to explore more sophisticated levels of motion safety such as the *passive friendly motion safety* mentioned in [3]: it guarantees that, if a collision takes place, the robot will be at rest and the colliding object could have the time to stop or avoid the collision (if it wants to).

References

[1] T. Fraichard and T. Howard, "Iterative motion planning and safety issue," in *Handbook of Intelligent Vehicles*, A. Eskandarian, Ed. Springer, 2012, pp. 1433–1458.

[2] T. Fraichard, "Will the driver seat ever be empty?" INRIA" Research Report, 2014.

[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] S. Bouraine, T. Fraichard, and H. Salhi, "Provably safe navigation for mobile robots with limited field-of-views in dynamic environments," *Autonomous Robots*, vol. 32, no. 3, pp. 267–283, 2012, *[www]*.

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

[6] S. LaValle and J. Kuffner, "Randomized kinodynamic planning." *International Journal of Robotics Research*, vol. 20, pp. 378–400, May 2001.

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

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

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

[10] K. Bekris and L. Kavraki, "Greedy but safe replanning under kinodynamic constraints," in *IEEE Int. Conf. Robotics and Automation*, Rome (IT), Apr. 2007.

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

[12] D. Althoff, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic collision state checker for crowded environments," in *IEEE Int. Conf. on Robotics and Automation*, Anchorage (US), May 2010.

[13] A. Bautin, L. Martinez-Gomez, and T. Fraichard, "Inevitable collision states: a probabilistic perspective," in *IEEE Int. Conf. on Robotics and Automation*, 2010.

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

[15] 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.

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

[17] K. Kant and S. Zucker, "Toward efficient trajectory planning: the path-velocity decomposition," *Int. Journal Robotics Research*, vol. 5, no. 3, Fall 1986.