The obstacle avoidance motion planning problem for autonomous vehicles: a low-demanding receding horizon control scheme

Giuseppe Franzè[a], Walter Lucia[a]

[a]*DIMES - Università degli Studi della Calabria, Rende (CS), 87036, ITALY*
{franze,wlucia}@dimes.unical.it

Abstract

The paper addresses the obstacle avoidance motion planning problem for ground vehicles operating in uncertain environments. By resorting to set-theoretic ideas, a receding horizon control algorithm is proposed for robots modelled by linear time-invariant (LTI) systems subject to input and state constraints and disturbance effects. Sequences of inner ellipsoidal approximations of the exact one-step controllable sets are pre-computed for all the possible obstacle scenarios and then on-line exploited to determine the more adequate control action to be applied to the robot in a receding horizon fashion. The resulting framework guarantees Uniformly Ultimate Boundedness and constraints fulfilment regardless of any obstacle scenario occurrence.

Keywords: Obstacle avoidance, set-theoretic approach, receding horizon control, constraint satisfaction.

1. Introduction

The problem of motion planning and control for autonomous mobile vehicles deals with finding appropriate command inputs such that the resulting vehicle trajectory satisfies the requirements of a specified task [14]. For most real mobile robotics applications, a basic requirement is the capability to safely operate in dynamic and a priori unknown environments [21]. Despite the extensive research, this problem still represents a relevant challenge because of unavoidable uncertainties in the operating scenario, inherent deficiencies in perception abilities and computational capabilities of the robot and restrictions on the vehicle mobility due to nonholonomic kinematic constraints, limited control ranges and under-actuations, see e.g. [12]. Avoidance of collisions with moving obstacles is a key component of the safe navigation whose typical objective is to reach a target through the obstacle-free part of the environment, see [20], [7] and references therein. In this respect, it is well-known that methods based on navigation functions are not capable to properly take care of any physical constraints (see e.g. the specific schemes for navigation of autonomous vehicles [30], [8] and air traffic management [3]) and this represents a *non-negligible* drawback because it makes restrictive such approaches [13].

In particular, during the last decade much attention has been devoted to exploit the possibility of extending road-map and potential functions methods to the case of dynamic obstacle scenarios [22], [24], [26], [28], [29]. In [22] Probabilistic Roadmap Methods (PRM) are considered with the aim to overcome the assumption of static environments properly of this class of strategies. A straightforward and high-demanding solution consists in updating the roadmap after an obstacle has changed its position. Therefore, the proposed algorithm creates a robust roadmap in the preprocessing phase by using the observation that the behaviour of the moving obstacles is often not unconstrained but restricted to pre-specified areas. In [24], the so-called Partial Motion Planner (PMP) mechanism is designed so that uncertainties arising from planning within dynamic environments can be handled. The main idea is to pre-calculate admissible state trajectories by using the Inevitable Collision States (ICS) framework that, though it is capable to generate safe paths, is subject to high computational burdens which could lead to violate the real-time constraints under which the robot must take a decision. By considering dynamic objects characterized by piecewise constant velocities, an explicit kinematic model of the robot is considered in [26]: the family of feasible trajectories and their corresponding steering controls are derived in a closed form. In [28], the path planning problem under nonholonomic constraints is addressed by using the so-called *Follow the Gap Method* (FGM). There by calculating a gap array around the robot, the appropriate gap is selected, the best heading vector through the gap derived and the final angle to the target point computed. Along similar lines is the contribution in [29] where a hybrid approach using *a-priori* knowledge of the environment guarantees that the autonomous vehicle cannot be trapped in deadlocks.

All these contributions share as a common denominator the fact that the path following obstacle avoidance problem is partially tackled: the control unit design is either leaved completely out (e.g. [28], [29]) or it is obtained by considering specific kinematic vehicle descriptions (see [26]).

Moving from this analysis, we are here interested to consider constrained Receding Horizon Control schemes which are an extremely appealing methodology for dealing with the obstacle avoidance motion planning problem in virtue of their intrinsic capability to generate at each time instant feasible trajectories that allow to safely reach a given goal, see to cite a few the recent contributions [23], [31], [11], [27]. In [23] the dynamic window approach (DWA) navigation scheme is recast within

a continuous time nonlinear model control predictive (MPC) framework by resorting to the ideas developed in [25]. In particular the algorithm is based on the jointly use of a model-based optimization scheme and a convergence-oriented potential field method. Although interesting, the approach suffers of unavoidable high computational burdens mainly due to the MPC phase that it is slightly mitigated by splitting the dissipative controls into two subsets with piecewise controls. The authors of [31] consider the collision avoidance as well as navigation towards the destination using a MPC approach. The related on-line optimization is solved by means of nonlinear programming, while a local path planning generator makes use of the so-called *distance* and parallax methods. The main handicap of this method relies on the heavy computational loads pertaining to the computation of the predictive controller action. In this contribution, this difficulty is attacked by considering a very large sensing range that allows to have available sufficient time for solving the underline constrained optimization problem. Along similar lines is also [11] and the references therein. More relevant for our purposes, although limited to a single static obstacle configuration, are instead the developments of [27], where algorithms for the computation of the set of states that can be robustly steered in a finite number of steps via state feedback control to a given target set while avoiding pre-specified zones or obstacles are achieved by exploiting polyhedral algebra concepts. There, it is shown how such regions are necessary to adequately deal with the obstacle avoidance problem even if the computational burdens may become prohibitive in some situations, e.g. when the system is subject to bounded disturbances.

In the sequel, we shall consider the class of dynamic environments defined as follows:

The obstacle locations on the working area are known, but at each time instant it is unpredictable which is the current obstacle configuration.

The consequence of such a formulation is that the resulting setup gives rise to a certain degree of uncertainty that if not properly treated can lead to collisions during the vehicle navigation. To deal with this problem, we develop a novel discrete-time receding horizon control (RHC) strategy based on set-theoretic ideas so that the prescribed saturation and geometric constraints are always fulfilled despite of any obstacle scenario occurrence. The key motivation supporting such an approach relies on the capabilities of the RHC philosophy combined both with set-invariance concepts and the ellipsoidal calculus to guarantee control performance and computational load savings under constraints satisfaction and disturbances effect attenuation requirements, see [2].
Then, the main ingredients of the proposed strategy can be summarized as follows:

- Compute a stabilizing state feedback law and a robust positively invariant ellipsoidal set centred at the goal location;
- Enlarge the set of initial states that, according to the obstacle scenario configurations, can be steered to the target in a finite number of steps;
- At each sampling time, an on-line receding horizon strategy is obtained by deriving the smallest ellipsoidal set complying with the current obstacle configuration. The control move is computed by minimizing a performance index such that the one-step ahead state prediction belongs to the successor set.

A relevant feature of this scheme is the capability to ensure that there exists at each time instant a feasible solution complying with the time-varying obstacle configuration prescriptions. Then, a second important merit relies on the needed computational resources that are significantly modest because the command input computation prescribes at most the solution of a Quadratic Programming (QP) problem under linear constraints.

Finally, the theoretical results are illustrated by means of a simulation campaign on a point mobile robot model whose navigation within a planar environment is limited by the occurrence of moving obstacles.

2. Preliminaries and Notations

Throughout the paper, we consider autonomous vehicles described by discrete-time LTI systems

$$x(t+1) = \Phi x(t) + Gu(t) + G_w w(t) \tag{1}$$

where $t \in \mathbb{Z}_+ := \{0, 1, ...\}$, $x(t) \in \mathbb{R}^n$ denotes the plant state, $u(t) \in \mathbb{R}^m$ the control input and $w(t) \in \mathcal{W} \subset \mathbb{R}^w$, $\forall t \in \mathbb{Z}_+$, an exogenous bounded disturbance. Moreover, the model (1) is subject to the following set-membership state and input constraints:

$$u(t) \in \mathcal{U},\ \forall t \geq 0, \tag{2}$$

$$x(t) \in \mathcal{X},\ \forall t \geq 0, \tag{3}$$

with $\mathcal{U}$, $\mathcal{X}$ compact subsets of $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively.

Definition 1. *A set $\mathcal{T} \subseteq \mathbb{R}^n$ is robustly positively invariant for (1) if there exists a control law $u(t) \in \mathcal{U}$ such that once the closed-loop solution $x_{CL}(t)$ enters inside that set at any given time t_0, it remains in it for all future instants, i.e. $x_{CL}(t_0) \in \mathcal{T} \rightarrow x_{CL}(t) \in \mathcal{T}, \forall w(t) \in \mathcal{W}, \forall t \geq t_0$.*

Given the system (1), it is possible to determine the sets of states i-step controllable to $\mathcal{T}$ via the following recursion (see [4]):

$$\begin{aligned} \mathcal{T}_0 &:= \mathcal{T} \\ \mathcal{T}_i &:= \{x \in \mathcal{X} : \exists\, u \in \mathcal{U} : \forall w \in \mathcal{W} : \Phi x + Gu \\ &\qquad +G_w w \in \mathcal{T}_{i-1}\}, \\ &= \{x \in \mathcal{X} : \exists\, u \in \mathcal{U} : \Phi x + Gu \in \tilde{\mathcal{T}}_{i-1}\} \end{aligned} \tag{4}$$

where $In[\cdot]$ denotes the inner ellipsoidal approximation, whereas $\tilde{\mathcal{T}}_{i-1} := In[\mathcal{T}_{i-1} \sim G_w\mathcal{W}]$ is defined as $In[\{x \in \mathcal{T}_{i-1} : x + w \in \mathcal{T}_{i-1}, \forall w \in G_w\mathcal{W}\}]$ and $\sim$ is known as the *P-difference* operator [18]. Moreover, $\mathcal{T}_0$ is known as the terminal region and $\mathcal{T}_i$ is the set of states that can be steered into $\mathcal{T}_{i-1}$ using a single control move.

Definition 2. *Let S be a neighbourhood of the origin. The closed-loop trajectory of (1) is said to be Uniformly Ultimate Bounded in S if for all $\mu > 0$ there exist $T(\mu) > 0$ and $u(t) \in \mathcal{U}$ such that, for every $\|x(0)\| \leq \mu$, $x_{CL}(t) \in S$ for all $t \geq T(\mu)$.*

Definition 3. *Given a set $W \subset \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$, the distance is defined as:*

$$dist(x, W) = \inf_{w \in W} \|x - w\|_*,$$

where $*$ is any relevant norm.

Definition 4. *Given two sets $W, R \subset \mathbb{R}^n$ the distance is defined as:*

$$dist(W, R) = \inf \{\|w - r\|_* \,|\, w \in W,\ r \in R\}$$

Definition 5. *An oriented graph is an ordered pair $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ such that*

- $\mathbf{V}$ *is the vertex set;*
- $\mathbf{E}$ *is a subset of ordered pairs of $\mathbf{V}$ known as the edge set, i.e.*

$$\mathbf{E} \subset \{\{u, v\} \,|\, u, v \in \mathbf{V}\}$$

Definition 6. *Let $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ an oriented graph, the reachable vertex set $\mathbf{V}_r$ is defined as*

$$\mathbf{V}_r := \{v \in \mathbf{V} | \exists u \in \mathbf{V} : \{u, v\} \in \mathbf{E}\}$$

3. Problem formulation

We shall point out our attention to the evolution of autonomous vehicles within possibly dynamic environments. Specifically, we refer to the following uncertain configuration:

- each moving object can occupy pre-specified positions so that a finite and known set of obstacle scenarios comes out;
- time instants at which an obstacle scenario change occurs are unknown.

This set-up mainly concerns with *partially known* working areas where a specific set of obstacle scenarios may occur, e.g. production/assembly lines [5], restricted areas (buildings or rooms) where surveillance or housekeeping are required tasks [9] and so on.

Hereafter the following definitions will be used:

Definition 7. *Let Ob_j^i be an object. Then an obstacle scenario $\mathcal{O}^i$ is defined as*

$$\mathcal{O}^i := \{Ob_1^i, \ldots, Ob_{n_i}^i\} \tag{5}$$

where n_i denotes the number of the involved objects. □

Definition 8. *Let $\mathcal{O}^i$ be an obstacle scenario. Then the nonconvex obstacle-free region pertaining to $\mathcal{O}^i$ is identified as follows*

$$\mathcal{O}_{free}^i := \{x \in \mathbb{R}^n : h_i(x) > 0\} \tag{6}$$

where $h_i : \mathbb{R}^n \to \mathbb{R}^{n_f}$ and n_f the number of component-wise inequalities.

Then, the problem we want to solve can be stated as follows.
Obstacle Avoidance Motion Planning (OAMP) Problem - *Given a set of obstacle scenarios $\mathcal{O}^i$, $i = 1, \ldots, l$, determine a state-feedback control policy*

$$u(t) = g(x(t)) \tag{7}$$

compatible with (2)-(3) and (6), such that starting from an initial condition $x(0)$ the robot trajectory $x(t)$ is driven to a target position x_f regardless of any obstacle scenario occurrence. □
In the sequel, the problem will be addressed by means of a receding horizon control approach based on set-theoretic concepts which has been successfully used in different contexts, see e.g. [2], [10]. In order to exploit the ideas of [2] for dealing with the **OAMP** problem, the following key questions must be analysed in depth:

- How can the nonconvex obstacle constraints be recast into computationally tractable conditions?
- How can one define a sequence of one-step controllable sets such that there exists at least a feasible path ($x(0) \to x_f$) complying with obstacle avoidance purposes?

Moreover, we assume without loss of generality that each Ob_j^i has a polyhedral convex structure described as the intersection of l_j half-spaces:

$$Ob_j^i : \begin{bmatrix} (H_j^i)_1^T \\ \vdots \\ (H_j^i)_{l_j}^T \end{bmatrix} p \leq \begin{bmatrix} (g_j^i)_1 \\ \vdots \\ (g_j^i)_{l_j} \end{bmatrix} \tag{8}$$

where $p := B\,x \in \mathbb{R}^2$ are the planar components of the state space $x \in \mathbb{R}^n$ and $B \in \mathbb{R}^{2 \times n}$ a projection matrix.

4. A set-theoretic approach

The aim of this section is to provide a set-theoretic based solution to the **OAMP** problem. Notice that the *basic* scheme of [2] can be extended to the proposed framework if the following key aspects are concerned:

a) Given an obstacle scenario $\mathcal{O}^i$, determine a sequence of one-step controllable sets $\{\mathcal{T}_s^i\}_{s=0}^{N_i}$ such that there exists at least a feasible path to the goal x_f;

b) Given the sequences $\{\mathcal{T}_s^i\}_{s=0}^{N_i}$, $i = 1, \ldots, l$, ensure viability properties under time-varying scenarios.

Then, the next developments are devoted to translate these ideas into tractable conditions and computational algorithms.

4.1. One-step controllable sets

Point **a)** relies on the estimation of the domain of attraction (DoA), i.e. all the initial conditions $x(0)$ for which there exists an admissible path to x_f. Note that in fact the *basic* construction (recursions (4)) of $\mathcal{T}_s^i$ may give rise to "small" DoA estimates, i.e. $\bigcup_{i=1}^{l} \mathcal{T}_{N_i}^i$, because a saturation effect may occur on the one-step controllable sets growth. To overcome such a drawback, here we modify the construction of these sets as follows:

---

*One-step Controllable Set Procedure (**OCSP**)*

1: *Given the goal x_f and chosen the initial condition $x(0)$ as follows*

$$x(0) := \arg\max_{x \in \mathcal{X}} \|x - x_f\|_2$$

design the pair $(K_0, \mathcal{T}_0)$, with $\mathcal{T}_0$ a RPI region centred in x_f where K_0 is the stabilizing state-feedback gain complying with the constraints (2), (3), (6). Store the subscript $m = 0$ into an index vector hereafter named IR^i. Let $\mathcal{T}_0^i = \mathcal{T}_0$ and $x_{eq}^0 = x_f$ be the initial terminal region and equilibrium point, respectively;

2: *Derive the sequence $\{\mathcal{T}_s^i\}_{s=1}^{N_m}$ by using recursions (4) under the additional state constrain (6). The integer N_m is the saturation level for the region growth;*

3: *Store the index N_m into an index vector denoted as LR^i;*

4: ***if** $x(0) \notin \mathcal{T}_{N_m}^i$, **then***

5: ***if** there exists a candidate equilibrium x_{eq}^{m+1} such that:*

***a)** $x_{eq}^{m+1} \neq x_{eq}^k$, $k = 0, 1, \ldots, m$;*

***b)** $x_{eq}^{m+1} := \arg\min_{x_{eq} \in \mathcal{T}_{N_m}^i} \|x_{eq} - x(0)\|_2$*

then

6: *Design a new pair $(K_{m+1}, \mathcal{T}_{N_m+1}^i)$, with $\mathcal{T}_{N_m+1}^i$ centered in x_{eq}^{m+1} and K_{m+1} satisfying (2), (3), (6). Store the corresponding index $N_m + 1$ into IR^i;*

7: *Add $\{\mathcal{T}_s^i\}_{s=1}^{N_m}$ to the previous computed sequence;*

8: *$m \leftarrow m + 1$, $\mathcal{T}_0^i := \mathcal{T}_{N_m+1}^i$ and goto Step 2;*

9: ***else** Stop;*

10: ***end if***

11: ***else** Stop;*

12: ***end if***

A graphical description of the **OCSP** *modus operandi* is given in Fig. 1.

Remark 1 - Notice that

- the vector IR^i has the aim to keep trace of all the robust positively invariant regions obtained by the **OCSP** procedure and its relevance will appear clearly in the next Section 5 where the receding horizon scheme is detailed;
- the index vector LR^i stores all the "saturation regions" for each subsequence of $\{\mathcal{T}_s^i\}$ whose aim is to ensure collision avoidance (see the discussion pertaining to the formulas (17)-(21));

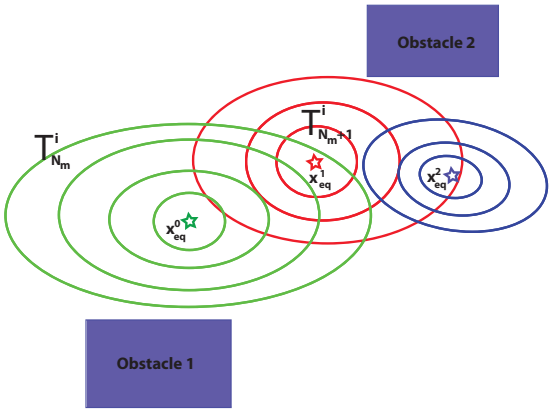

Figure 1: Controllable set sequences construction

- the pairs $(K_m, \mathcal{T}_m^i)$, $m \in IR^i$ are computed via standard LMI techniques, see [17]. □

4.2. Obstacle constraints convexification

Here, the non-convex constraints describing the admissibility regions (6) are translated into computationally tractable requirements. This is achieved via the following arguments:

- by exploiting the polyhedral structure (8) the obstacle-free region (6) can be described as

$$\mathcal{O}_{free}^i := \bigcap_{j=1}^{n_i} \left\{ x \in \mathbb{R}^n : \bigcup_{k=1}^{l_j} \left\{ (H_j^i)_k^T B x > (g_j^i)_k \right\} \right\}, \tag{9}$$

- by denoting as

$$(c_j^i)^a := \left\{ x \in \mathbb{R}^n : \bigcup_{k=1}^{l_j} \left\{ (H_j^i)_k^T B x > (g_j^i)_k \right\} \right\} \tag{10}$$

the *active* constraint region pertaining to each $j - th$ obstacle of $\mathcal{O}^i$ and by using the fact that the construction of controllable set families is based on the equilibrium x_{eq}, (see Fig. 2) the following statement holds true:

Statement 1. *Let $\mathcal{T}_s^i$ be a one-step controllable set, the computation of the predecessor set $\mathcal{T}_{s+1}^i$ via (4) requires that at most two amongst all the constraints $(H_j^i)_k^T B x > (g_j^i)_k$ are imposed for each $j - th$ obstacle in order to define $(c_j^i)^a$.* □

Then, the region (9) can be convexified by means of the following algorithm:

---

*Convexification procedure (**CP**)*

---

0. *Let x_{eq}, Ob_j^i $j = 1 \ldots n_i$, and $\mathcal{B}_\delta$ be the equilibrium point used by the **OCSP** procedure, the obstacles of the $i - th$ scenario and a ball of radius $\delta > 0$, respectively;*

1. *For each* Ob_j^i, *compute*

$$(p_j^i)^{min} = \arg \min_{p \in Ob_j^i} \{dist(p, Bx_{eq})\}$$

and the hyperplane $(h_j^i)^1 := \left\{(H_j^i)_{j_1}^T Bx = (g_j^i)_{j_1}\right\}$ *such that* $(p_j^i)^{min} \in (h_j^i)^1$;

2. *For each* Ob_j^i, *determine the closest hyperplane*

$$(h_j^i)^2 := \left\{(H_i^j)_{j_2}^T Bx = (g_j^i)_{j_2}\right\}, \ (h_j^i)^2 \neq (h_j^i)^1,$$

to the point $(p_j^i)^{min}$;

3. *For each* Ob_j^i, *define the "external" half-spaces as*

$$(sp_j^i)^1 := \left\{(H_j^i)_{j_1}^T Bx > (g_j^i)_{j_1}\right\}$$
$$(sp_j^i)^2 := \left\{(H_j^i)_{j_2}^T Bx > (g_j^i)_{j_2}\right\}$$

4. *For each* Ob_j^i

   a) *if* $Bx_{eq} \in ((sp_j^i)^1 \cap (sp_j^i)^2 \sim B_\delta)$, *then* $(c_j^i)^a = (sp_j^i)^1 \cap (sp_j^i)^2$;

   b) *else if* $Bx_{eq} \in ((sp_j^i)^1 \cap (sp_j^i)^2)$ *and* $d(Bx_{eq}, (sp_j^i)^1) < \delta$, *then* $(c_j^i)^a = (sp_j^i)^2$;

   c) *else if* $Bx_{eq} \in ((sp_j^i)^1 \cap (sp_j^i)^2)$ *and* $d(Bx_{eq}, (sp_j^i)^2) < \delta$, *then* $(c_j^i)^a = (sp_j^i)^1$;

   d) *else* $(c_j^i)^a = (sp_j^i)^1$;

5.

$$(c^i)^a = \ (c_1^i)^a \cap (c_2^i)^a \cap \ldots \cap (c_{n_i}^i)^a = \bigcap_{j=1}^{n_i} \left\{x \in \mathbb{R}^n \mid (H_j^i)^a Bx > (g_j^i)^a\right\} \quad (11)$$

Remark 2 - Note that *Step* 4 allows to achieve a convex relaxation of the union set operator in (9) whose rationale directly comes out from Statement 1 arguments, while *Step* 5 computes the convexification of the whole region (9). Therefore, for each $\mathcal{O}^i$ the following additional geometric constraints must be imposed:

$$(H_j^i)^a Bx > (g_j^i)^a, \ j = 1, \ldots, n_i. \quad (12)$$

Moreover, the vanishing tolerance level δ is instrumental to univocally identify the half-spaces to which the equilibrium x_{eq} belongs and, as a consequence, the constraints to be imposed at *Step* 2 of the **OCSP** procedure in place of (6). □

Finally, Fig. 2 provides a description of the conditions outlined in *Step* 4. There by referring to the obstacle labelled with Ob_1, the point Bx_{eq} belongs to the intersection of the two half-spaces characterized by $(h_1^i)^1$ and $(h_1^i)^2$ under the restriction imposed by the tolerance level δ. This matches the prescriptions of the case *4.a)*, therefore the constraints to impose are

$$(H_1^i)_{j_1} Bx > (g_1^i)_{j_1}$$
$$(H_1^i)_{j_2} Bx > (g_1^i)_{j_2}$$

and the grey zone in Fig. 2 describes the convexified region $(c_1^i)^a$. A similar reasoning applies to the other obstacles.

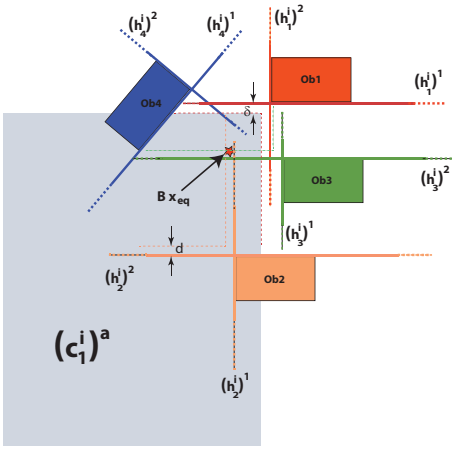

Figure 2: Convexification procedure description

4.3. Time-varying obstacle scenario occurrences

The question **b)** focuses on the difficulties arising when unpredictable obstacle scenario changes occur. Since the l controllable set sequences $\{\mathcal{T}_s^i\}$, $i = 1, \ldots, l$, are computed under the hypothesis that a single obstacle scenario $\mathcal{O}^i$ takes place, the viability retention cannot be ensured when $\mathcal{O}^i \to \mathcal{O}^{i'}$ because a switching to a different set sequence must be imposed.

Here, the idea is to design two further families of one-step controllable sets, hereafter named *Obstacle* and *Scenario Switching* sequences, whose combined use allows to *safely* switch to $\{\mathcal{T}_s^{i'}\}$.

4.3.1. Obstacle controllable sequences

The *Obstacle* sequences $\{\mathcal{T}_s^{Ob_j^{i'}}\}$, $j = 1, \ldots, n_{i'}$, have the aim to encircle the corresponding obstacles $Ob_j^{i'}$, , $j = 1, \ldots, n_{i'}$ and they are introduced because of the following argument:

Let $\mathcal{O}^i$ *be the current scenario, if at* $\bar{t}$ *the obstacle scenario change* $\mathcal{O}^i \to \mathcal{O}^{i'}$ *occurs and the current state* $x(\bar{t}) \notin \mathcal{T}_s^{i'}$, *for some* s, *then collisions could happen because the sequence* $\{\mathcal{T}_s^i\}$ *is no longer admissible.*

In order to avoid such an undesired event, the *Obstacle* sets are designed such that when $x(\hat{t}) \in \{\mathcal{T}_s^{Ob_j^{i'}}\}, \hat{t} \geq \bar{t}$, the trajectory $x(t)$, $\forall t \geq \hat{t}$ remains confined into them until the switching to the correct sequence $\{\mathcal{T}_s^{i'}\}$ is made admissible.

The computation of the sequences $\{\mathcal{T}_s^{Ob_j^{i'}}\}$, $j = 1, \ldots, n_{i'}$, is performed by using the **OCSP** procedure where both $x(0)$ and x_f are substituted with the equilibria $x_{eq}^{Ob_j^{i'}}$, $j = 1, \ldots, n_{i'}$. These equilibrium points are selected as those at the maximum distance from the obstacles $Ob_j^{i'}$, $j = 1, \ldots, n_{i'}$, and satisfying the following condition:

$$dist(Bx_{eq}^{Ob_j^{i'}}, Ob_j^{i'}) < dist(\mathcal{T}_{max_j}^i, Ob_j^{i'}), \ j = 1, \ldots, n_{i'}, \quad (13)$$

where $\mathcal{T}_{max_j}^i$ are the sets corresponding to the greatest indices s of $\{\mathcal{T}_s^i\}$ such that

$$\mathcal{T}_{max_j}^i \cap Ob_j^{i'} = \emptyset, \ j = 1, \ldots, n_{i'} \quad (14)$$

The latter allows to ensure that the equilibrium $x_{eq}^{Ob_j^{i'}}$ lies in the zone bounded by the obstacle and the closer element of $\{\mathcal{T}_s^i\}$ which does not intersect $Ob_j^{i'}$, i.e. $\mathcal{T}_{max_j}^i$. The relevance of this choice will be soon evident in the next Section when the scenario switching will be addressed. For the sake of clarity, an example of the above reasoning is given in Fig. 3.

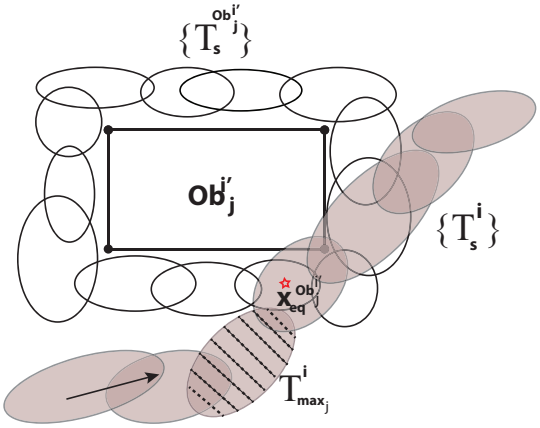

Figure 3: Obstacle sequence construction. The dashed ellipsoid is $\mathcal{T}_{max_j}^i$. The arrow denotes the increasing direction of the subscript s.

4.3.2. Scenario Switching controllable sequences

Although the *Obstacle* sequences have the important merit to avoid collisions when the scenario $\mathcal{O}^{i'}$ occurs, their use leads to the following drawback:

Let $\mathcal{O}^i \to \mathcal{O}^{i'}$ be a generic scenario change occurrence, once the trajectory $x(\cdot)$ enters inside the region defined by $\{\mathcal{T}_s^{Ob_j^{i'}}\}$, it is by construction driven to the terminal region $\mathcal{T}_0^{Ob_j^{i'}}$ centred at the equilibrium $x_{eq}^{Ob_j^{i'}}$ and it will be there confined.

This situation clearly compromises any chance of the vehicle to reach the target x_f and for this reason the trajectory $x(\cdot)$ must be driven towards the controllable sequence $\{\mathcal{T}_s^{i'}\}$. To this aim, the *Scenario Switching* sequences $\{\mathcal{T}_s^{SW_j^{i'}}\}$, $j = 1, \ldots, n_i$, are introduced and designed as follows:

1. *for any admissible scenario change $\mathcal{O}^i \to \mathcal{O}^{i'}$, choose an equilibrium $x_{eq}^{SW_j^{i'}}$ belonging to some $\mathcal{T}_s^{i'}$;*
2. *apply the* **OCSP** *procedure with $x_{eq}^{Ob_j^{i'}}$ in place of $x(0)$ and $x_{eq}^{SW_j^{i'}}$ in place of x_f.*

Fig. 4 provides a graphical interpretation of the generation of *Scenario Switching* sequences. Then, the switching $\{\mathcal{T}_s^i\} \to \{\mathcal{T}_s^{i'}\}$ is accomplished as follows:

- *Let $x(\hat{t}) \in \{\mathcal{T}_s^{Ob_j^{i'}}\}$ be the current state. As soon as*

$$x(t) \in \mathcal{T}_k^{SW_j^{i'}}, \quad \text{for some } k \text{ and for some } t \geq \hat{t}, \tag{15}$$

the set-membership to $\{\mathcal{T}_s^{SW_j^{i'}}\}$ is considered;

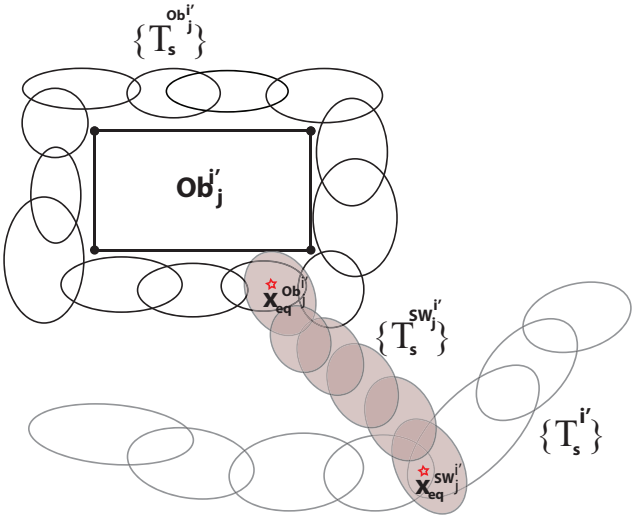

Figure 4: Scenario Switching sequence construction

- *As soon as*

$$x(\bar{t}) \in \mathcal{T}_k^{i'}, \quad \text{for some } k \text{ and for some } \bar{t} \geq \hat{t}. \tag{16}$$

the set-membership to $\{\mathcal{T}_s^{i'}\}$ can be used.

A relevant aspect leaved out in the above developments is to guarantee that under a scenario change the robot does not bump on any obstacle during the switching phase. To this end, the sequences $\{\mathcal{T}_s^i\}$ and $\{\mathcal{T}_s^{SW_j^i}\}$ are derived under the following further condition:

$$\|B(x^+ - x)\| \leq \epsilon, \tag{17}$$

where $x^+ = \Phi x + Gu$ is the disturbance-free one-step evolution. The inequality (17) limits the maximum displacement of the one-step state evolution so that obstacle constraints are satisfied with a tolerance level $\epsilon > 0$. The scalar ϵ can be determined by exploiting the *Obstacle* sequences properties as below detailed:

- *for each scenario $\mathcal{O}^i$ and for each $j-th$ obstacle, consider the sequence $\{\mathcal{T}_s^{Ob_j^i}\}$ and determine the sequence $\{\mathcal{T}_m^{IN_j^i}\}$ as follows:*

$$\begin{cases} \mathcal{T}_m^{IN_j^i} := \mathcal{T}_{LR_j^i(m)}^{Ob_j^i} \bigcap \mathcal{T}_{LR_j^i(m+1)}^{Ob_j^i}, \\ m = 1, \ldots, \dim(LR_j^i) - 1 \\ \mathcal{T}_{\dim(LR_j^i)}^{IN_j^i} := \mathcal{T}_{LR_j^i(1)}^{Ob_j^i} \bigcap \mathcal{T}_{LR_j^i(\dim(LR_j^i))}^{Ob_j^i} \end{cases} \tag{18}$$

where the index vectors LR_j^i store the saturation regions for each sequence $\{\mathcal{T}_s^{Ob_j^i}\}$;

- *compute the minimum d_j^i and maximum D_j^i euclidean distances between the $j-th$ obstacle (polyhedron) and the indexed set sequence $\{\mathcal{T}_m^{IN_j^i}\}$;*

- *compute*

$$d := \max_{\substack{i=1,\dots,l,\\ j=1,\dots,p_i}} d_j^i \tag{19}$$

$$D := \min_{\substack{i=1,\dots,l,\\ j=1,\dots,p_i}} D_j^i \tag{20}$$

$$\epsilon := D - d \tag{21}$$

An illustration of formulas (19)-(20) is provided in Fig. 5.

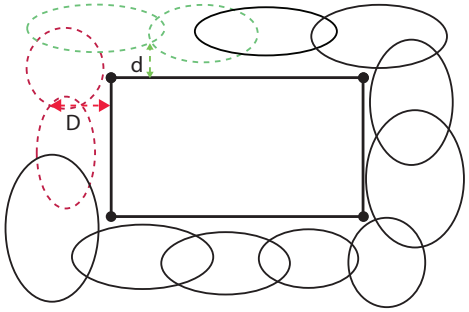

Figure 5: Tolerance level ϵ computation

Remark 3 - The shortest distance d_j^i between two convex sets can be obtained by using the algorithm proposed in [6], whereas an efficient procedure to estimate the maximum distance D_j^i is given in [15] where Hausdorff distance arguments are exploited. □

4.3.3. One-step controllable families computation

The consequence of all the above developments is that the set sequences must be computed by using the following scheme:

---

*Families Construction Procedure (**FCP**)*

---

0. *Let* $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ *be the oriented graph with* $\mathbf{V}$ *the* l *obstacle scenarios and* $\mathbf{E}$ *the set of ordered pairs* $\{i, i'\}, i, i' \in \mathbf{V}$, *such that the switching* $\mathcal{O}^i \to \mathcal{O}^{i'}$ *is admissible;*
1. *Generate the sequences* $\{\mathcal{T}_s^i\}$, $i = 1, \dots, l$;
2. *Generate the Obstacle families* $\{\mathcal{T}_s^{Ob_j^i}\}$, $\forall i \in \mathbf{V}_r$, $j = 1, \dots, n_i$, *according to (13)-(14);*
3. *Estimate the tolerance level* ϵ *by (19)-(21);*
4. *Update the sequences* $\{\mathcal{T}_s^i\}$, $i = 1, \dots, l$ *under the additional constraint (17);*
5. *Generate the sequences* $\{\mathcal{T}_s^{SW_j^i}\}$, $\forall i \in \mathbf{V}_r$, $j = 1, \dots, n_i$ *under the additional constraint (17).*

Finally, Fig. 6 provides a sketch of the switching *modus operandi* under an obstacle scenario change.

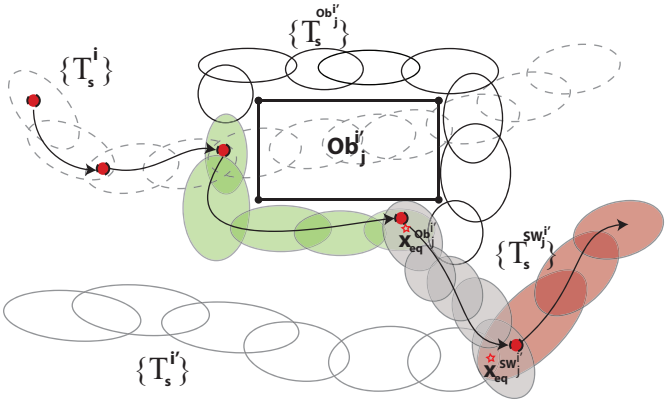

Figure 6: Obstacle scenario change: $\mathcal{O}^i \to \mathcal{O}^{i'}$: $\mathcal{O}^i$ sequence (white), Obstacle sequence (green), Scenario Switching sequence (grey), $\mathcal{O}^{i'}$ sequence (red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5. RHC algorithm

In this section a Receding Horizon Control strategy is outlined by collecting all the above developments. In the sequel the following assumptions are made:

Assumption 1 - At each time instant t the robot is informed about the current obstacle scenario, i.e. $sc(t)$. □

Assumption 2 - At each time instant t, an obstacle scenario change $\mathcal{O}^i \to \mathcal{O}^{i'}$ can occur if the actual robot position $x_p(t) = B\bar{x}(t)$ is such that:

$$\bar{x}(t) \notin \bigcup_{j=1}^{n_{i'}} \{x \in \mathbb{R}^n \mid H_j^{i'} Bx \leq g_j^{i'} + d\} \tag{22}$$

□

The rationale behind the condition (22) is to require that an obstacle configuration change can occur only when the current robot position does not lie in the region pertaining to the new obstacle scenario $\mathcal{O}^{i'}$.

---

*Obstacle Avoidance MPC (**OA-MPC**) Algorithm*

Off-line:

1 *Given the obstacle scenarios* $\mathcal{O}^i$, $i = 1, \dots, l$, *the initial condition* $x(0)$ *and the goal* x_f, *compute the non-empty robust invariant ellipsoidal region* $\mathcal{T}_0$ *and the stabilizing feedback gain* K_0 *complying with the constraints (2), (3), (12);*

2 *Apply the **FCP** scheme in order to generate the sequences* $\{\mathcal{T}_s^i\}$, $i = 1, \dots, l$, $\{\mathcal{T}_s^{Ob_j^i}\}$, $\forall i \in \mathbf{V_r}$, $j = 1, \dots, n_i$, *and* $\{\mathcal{T}_s^{SW_j^i}\}$, $\forall i \in \mathbf{V_r}$, $j = 1, \dots, n_i$ *such that*

$$x(0) \in \left(\bigcup_{\substack{i=1,\dots,l\\ s}} \mathcal{T}_s^i \right) \cup \left(\bigcup_{\substack{\forall i \in \mathbf{V_r}\\ j=1,\dots,n_i,\\ s}} \mathcal{T}_s^{Ob_j^i} \right) \cup \left(\bigcup_{\substack{\forall i \in \mathbf{V_r}\\ j=1,\dots,n_i,\\ s}} \mathcal{T}_s^{SW_j^i} \right)$$

3 *Store the ellipsoidal sequences.*

On-line:

1: ***if***

$$x(t) \in \left(\bigcup_{s} \mathcal{T}_s^{sc(t)} \right) \cup \left(\bigcup_{\substack{j=1,\dots,n_i,\\ s}} \mathcal{T}_s^{Ob_j^{sc(t)}} \right) \cup \left(\bigcup_{\substack{j=1,\dots,n_i,\\ s}} \mathcal{T}_s^{SW_j^{sc(t)}} \right)$$

then $curr := sc(t)$

2: ***else*** $curr := prec$;

3: ***end if***

4: ***if*** $x(t) \in \bigcup_{s} \mathcal{T}_s^{curr}$ ***then***

5: *goto Step 10 by considering* $\mathcal{T}_s^{curr}$ *as the candidate sequence, i.e.* $\mathcal{T}_s^{candidate}$.

6: ***end if***

7: ***if*** $x(t) \in \bigcup_{\substack{j=1,\dots,n_i,\\ s}} \mathcal{T}_s^{SW_j^{curr}}$ ***then*** $\mathcal{T}_s^{candidate} := \mathcal{T}_s^{SW_j^{curr}}$

8: ***else*** $\mathcal{T}_s^{candidate} := \mathcal{T}_s^{Ob_j^{curr}}$

9: ***end if***

10: *Find* $s(t) := \min_{s} \{x(t) \in \mathcal{T}_s^{candidate}\}$

11: ***if*** $s(t) \in IR^{candidate}$ ***then*** $u(t) = K_{s(t)} x(t)$

12: ***else***

13: ***if*** *curr==sc(t)* ***then***

$$u(t) = \arg\min \|\Phi x(t) + Gu\|^2_{\left(P_{s(t)-1}^{candidate}\right)^{-1}} \quad s.t.$$

$$\Phi x(t) + Gu \in \tilde{\mathcal{T}}_{s(t)-1}^{candidate},\ u \in \mathcal{U} \tag{23}$$

$$\tag{24}$$

14: ***else***

$$u(t) = \arg\min \|\Phi x(t) + Gu\|^2_{\left(P_{s(t)-1}^{candidate}\right)^{-1}} \quad s.t. \tag{25}$$

$$\|B(\Phi x(t) + Gu - x(t))\|_2^2 \le \epsilon^2 \tag{26}$$

$$\Phi x(t) + Gu \in \tilde{\mathcal{T}}_{s(t)-1}^{candidate},\ u \in \mathcal{U} \tag{27}$$

15: ***end if***

16: ***end if***

17: *Apply* $u(t)$; *prec:=curr;* $t := t+1$*; goto Step 1*

Note that the running cost $\|\Phi x(t) + Gu\|^2_{\left(P_{s(t)-1}^{candidate}\right)^{-1}}$ characterizes the one-step ahead state prediction with $P_{s(t)-1}^{candidate}$ the shaping matrix of the ellipsoidal region $\{x \in \mathbb{R}^n : x^T (P_{s(t)-1}^{candidate})^{-1} x \le 1\}$.

Remark 4 - In Step 13 the computation of the current command input $u(t)$ is achieved without imposing the one-step constraint (17) as done in Step 14. In fact, note that if the robot state measurement belongs to the current obstacle scenario $sc(t)$ collisions do not occur, therefore the requirement (17) becomes useless. As a consequence, the overall control performance is improved. □

The next proposition shows that the proposed **OA-MPC-Algorithm** enjoys the feasibility retention and closed-loop stability properties.

Proposition 1. *Let the sequences of sets* $\mathcal{T}_s^i$, $\mathcal{T}_s^{SW_j^i}$ *and* $\mathcal{T}_s^{Ob_j^i}$ *be non-empty and*

$$x(0) \in \left(\bigcup_{\substack{i=1,\dots,l\\ s}} \mathcal{T}_s^i \right) \cup \left(\bigcup_{\substack{\forall i \in \mathbf{V_r}\\ j=1,\dots,n_i,\\ s}} \mathcal{T}_s^{Ob_j^i} \right) \cup \left(\bigcup_{\substack{\forall i \in \mathbf{V_r}\\ j=1,\dots,n_i,\\ s}} \mathcal{T}_s^{SW_j^i} \right).$$

Then, the **OA-MPC-Algorithm** *always satisfies the constraints and ensures Uniformly Ultimate Boundedness for all time-varying occurrences of* $\mathcal{O}_i$, $i = 1, \dots, l$.

Proof - Note that existence of solutions at time t implies existence of solutions at time $t+1$, because the optimization problems in Steps 13 and 14 are always feasible. In fact, by construction there exists an input vector u satisfying the constraints (2), (3), (12) and (26) such that the set-membership requirement in (24) holds true. Then thanks to the **FCP** procedure and under the additional constraint constraints (17), at the next time instant $t+1$ the existence of a solution $u(t+1)$ for Steps 13-14 is ensured.

Finally, Uniformly Ultimate Boundedness of the strategy follows by noting that the trajectory is in the worst case confined to

$$\left(\bigcup_{\substack{i=1,\dots,l\\ s}} \mathcal{T}_s^i \right) \cup \left(\bigcup_{\substack{\forall i \in \mathbf{V_r}\\ j=1,\dots,n_i,\\ s}} \mathcal{T}_s^{Ob_j^i} \right) \cup \left(\bigcup_{\substack{\forall i \in \mathbf{V_r}\\ j=1,\dots,n_i,\\ s}} \mathcal{T}_s^{SW_j^i} \right)$$

□

6. Simulations

The aim of this section is to present results on the effectiveness of the proposed obstacle avoidance MPC strategy. Since to the best of authors knowledge, *complete* competitor schemes (planner + controller) are not present in literature, numerical comparisons restricted to the path planning problem are provided with the road-map *Cell decomposition* algorithm [1] properly adapted to comply with dynamic obstacle scenarios. Both the schemes have been compared in terms of performance and computational complexity and implemented in Matlab 7.1, making use of the Multi-Parametric Toolbox [16] over a laptop PC equipped with a Intel Core(TM) 2 Duo CPU.

We shall consider the point mobile robot model, described in [19], whose state consists of position and velocity components $x = [p_x\ p_y\ v_x\ v_y]^T$ and motions are governed by the following discrete-time LTI model:

$$x(t+1) = \Phi x(t) + Gu(t) + G_w w(t)$$

where $u \in \mathbb{R}^2$ is the acceleration vector and

$$\Phi = \begin{bmatrix} I_2 & \Delta t\, I_2 \\ 0_2 & I_2 \end{bmatrix},\ G = \begin{bmatrix} \frac{(\Delta t)^2\, I_2}{2} \\ \Delta t I_2 \end{bmatrix},\ G_d = G$$

with $\Delta t = 1\,s$ and $w(t) \in \mathcal{W} := \{w \in \mathbb{R}^2 : \|w\|_2 \leq 0.01\}, \forall t \geq 0$. Moreover, the following constraint on the acceleration vector is prescribed

$$\|u(t)\|_2^2 \leq 0.028, \ \forall t \geq 0 \tag{28}$$

6.1. Barrier configuration

This example refers to a critical case of study because the working environment is represented by a very narrow area and the single polyhedral obstacle Ob_1 moves along a straight line so that it virtually describes a particular obstacle configuration hereafter denoted as *barrier*, see Fig. 7. The obstacle positions are below reported:

Obstacle	width	heigth	scenario	center of gravity
Ob_1	1	1	1	$[2.5; 0.5]$
Ob_1	1	1	2	$[2.5; 1.5]$
Ob_1	1	1	3	$[2.5; 2.5]$
Ob_1	1	1	4	$[2.5; 3.5]$

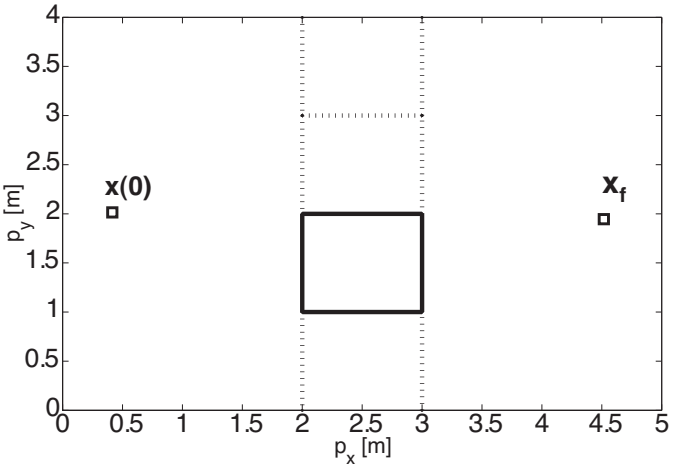

Figure 7: Working planar scenario

In particular, we have considered the following situations:

- **a)** The production line correctly operates and the obstacle circularly moves from the scenario 1 to the scenario 4;
- **b)** At a certain time instant a malfunction is detected, the production line stops its normal behaviour and it is necessary to come back from phase 3 to phase 2. Then, it resumes the prescribed operations.

In order to implement the **OA-MPC-Algorithm**, the terminal pair $(K_0, \mathcal{T}_0)$ has been first determined as follows:

$$K_0 = \begin{bmatrix} -0.1447 & -0.1446 & -0.2780 & -0.2678 \\ -0.1446 & -0.1447 & -0.2678 & -0.2780 \end{bmatrix},$$

$$\mathcal{T}_0 = \{x \in \mathbb{R}^n \mid (x - x_f)^T (P_0)^{-1} (x - x_f) \leq 1\}$$

with

$$(P_0)^{-1} = \begin{bmatrix} 0.1925 & 0.1792 & -0.0595 & -0.594 \\ 0.1792 & 0.1925 & -0.0594 & -0.0595 \\ -0.0595 & -0.0594 & 0.1078 & 0.1078 \\ -0.0594 & -0.0595 & 0.1078 & 0.1078 \end{bmatrix}$$

	Cell Decomposition	OA-MPC
a)	4.60 m	5.68 m
b)	7.05 m	5.59 m

Table 1: Paths length (metres)

Off-line burdens	Basic sequences:	330
	Obstacle sequences:	480
	Scenario switching sequences:	108
		918
On-line burdens		0.05

Table 2: Numerical burdens: average CPU time (seconds)

Then, the ellipsoidal families have been off-line derived by using the **FCP** procedure:

- $\{\mathcal{T}_s^{Ob_1^1}\}_{s=1}^{190}, \{\mathcal{T}_s^{Ob_1^2}\}_{s=1}^{147}, \{\mathcal{T}_s^{Ob_1^3}\}_{s=1}^{149}, \{\mathcal{T}_s^{Ob_1^4}\}_{s=1}^{200}$
- $\{\mathcal{T}_s^{1}\}_{s=1}^{72}, \{\mathcal{T}_s^{2}\}_{s=1}^{87}, \{\mathcal{T}_s^{3}\}_{s=1}^{98}, \{\mathcal{T}_s^{4}\}_{s=1}^{72}$
- $\{\mathcal{T}_s^{SW_1^1}\}_{s=1}^{10}, \{\mathcal{T}_s^{SW_1^2}\}_{s=1}^{2}, \{\mathcal{T}_s^{SW_1^3}\}_{s=1}^{43}, \{\mathcal{T}_s^{SW_1^4}\}_{s=1}^{105}$

with the estimated tolerance level $\epsilon = 0.2867$.

The on-line numerical results are collected in Figs. 8-13. Specifically, Figs. 8-10 refer to **a)** and it is interesting to note how the set-membership signal in Fig. 9 copes with the obstacle configurations occurrences, see Fig. 8. On the other hand the second group of Figs. 11-13 reports the results for the scenario **b)** by assuming that a malfunctioning occurs within the time interval $[55\ 80]\,s$. By taking a look at the grey zones of Figs. 11-12, it is relevant to observe that the **OA-MPC** algorithm prescribes that the current state first belongs to the family $\{\mathcal{T}_s^{Ob_1^2}\}$ and then to $\{\mathcal{T}_s^{Ob_1^3}\}$: therefore this undesired phenomenon is efficiently ridden out.

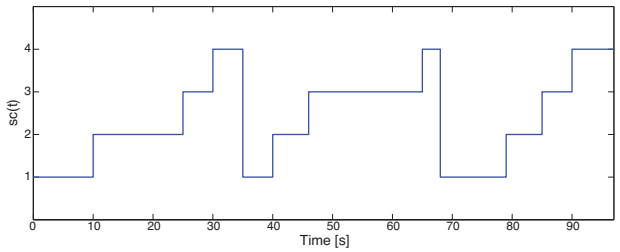

Figure 8: Obstacle scenario switchings

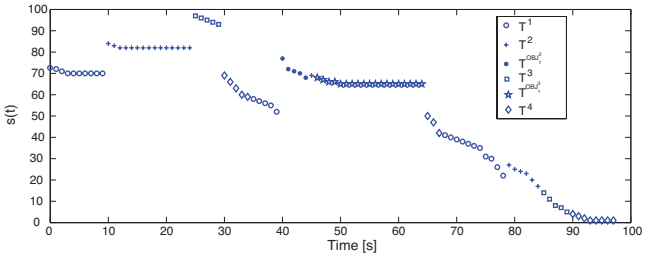

Figure 9: Set-membership signal

Comparisons with the modified *Cell Decomposition* algorithm have been performed by evaluating the path planning length. The results are summarized in Table 1 where it is clearly shown that the proposed strategy, although not oriented to the path planning optimization, provides performance similar to those obtained by the road-map algorithm.

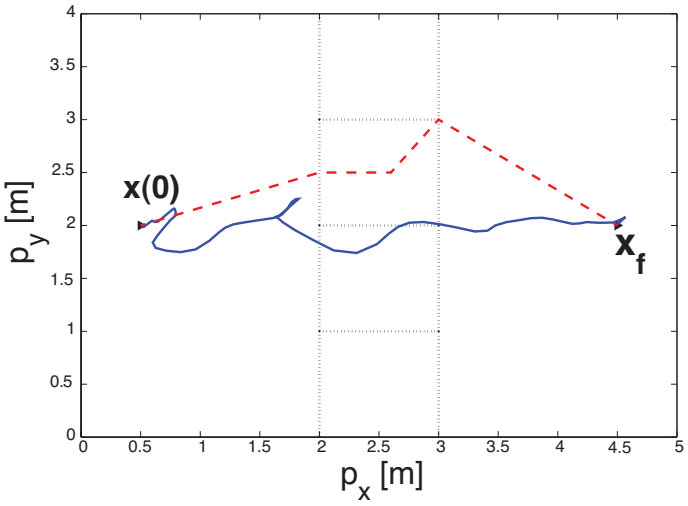

Figure 10: Robot path under time-varying obstacles: **OA-MPC** algorithm (blue continuous line), *Cell decomposition* (red dashed line).(For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

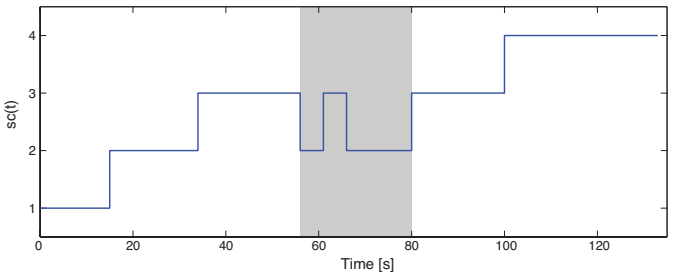

Figure 11: Obstacle scenario switchings

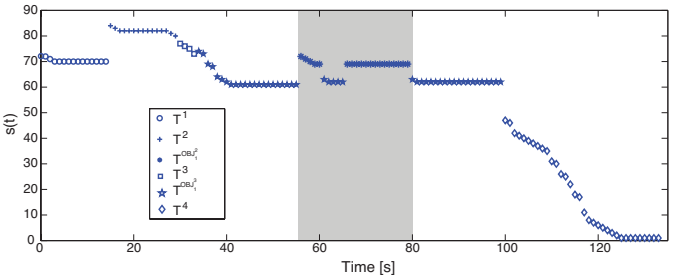

Figure 12: Set-membership signal

Finally, Table 2 reports the computational complexity pertaining to both the **Off-line** and **On-line** phases of the **OA-MPC** algorithm evaluated by computing the average CPU time (seconds). As expected, most of the overall computational loads have been moved to the **Off-line** phase ($918\,s$) so rendering affordable the use of the proposed strategy from a practical point of view.

On the other hand, it is relevant to put in evidence that the modified *Cell decomposition* algorithm, whose cost is only related to the path planning (not to the control action computation), presents an on-line computational load of a magnitude order greater than the proposed **OA-MPC** scheme, i.e. $0.45\,s$. The

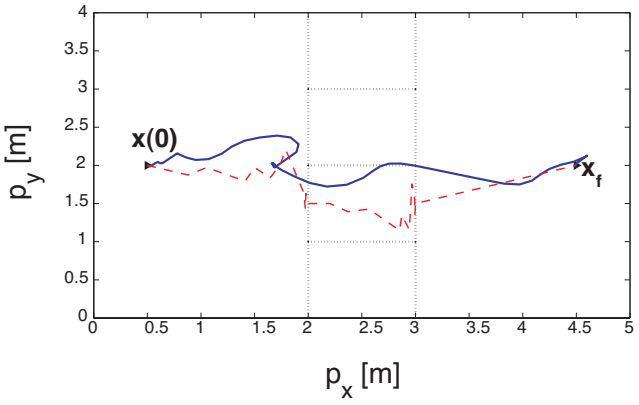

Figure 13: Robot path under time-varying obstacles: **OA-MPC** algorithm (blue continuous line), *Cell decomposition* (red dashed line)

latter is mainly due to the fact that the road-map method must on-line recomputed the visibility graph whenever an obstacle scenario change occurs.

6.2. *Cafeteria*

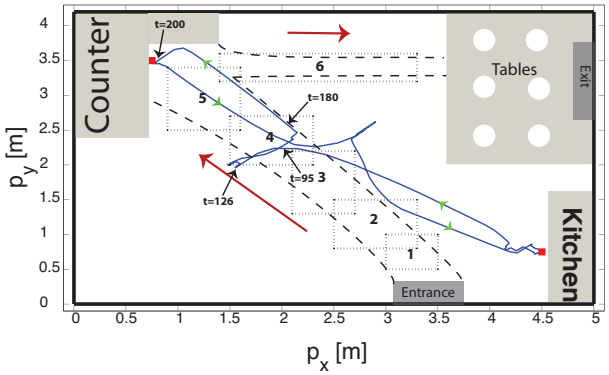

Figure 14: The cafeteria structure

The small cafeteria (size: $21\,m^2$) depicted in Fig. 14 consists of three areas: counter, kitchen and tables. Due to the limited free space, few guests can jointly access and they can move along prescribed lines (dashed lines): entrance-to-counter and counter-to-tables. In order to improve the cost efficiency, an autonomous ground vehicle for carrying plates from the kitchen to the counter and *vice-versa* is used. The main drawback of this idea is related to the motion of the people (time-varying obstacles) that may block the robot transit.

In the sequel, the *cafeteria-guest-robot* architecture is recast within the proposed strategy under the following assumptions:

1. almost two guests at a time are allowed along the lines;
2. people move following the directions indicated by the red arrows;
3. the walkable path is partitioned in six zones (dotted polygons in Fig. 14).

Assumption 1 defines the number of moving obstacles whereas Assumption 3 fixes the obstacle locations within the environ-

ment. Notice that an obstacle scenario change occurs only when a guest trespasses the limits of his current zone.
By considering the combinations with repetition (i, i') of the six polygons, we have that the number of the admissible obstacle scenarios is $l = 21$. Moreover under the Assumption 2, the oriented graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, whose rules are shown in Fig. 15, is obtained. Finally, in order to provide a mapping between the l obstacle scenarios and the nodes $\mathbf{V}$ we have that: *At each node* (i, i'), $i = 1, \dots l$, $i' = i, \dots l$, *corresponds the following scenario index:*

$$\textbf{scenario} \leftarrow 6(i-1) - \sum_{k=1}^{i-1} k + i'$$

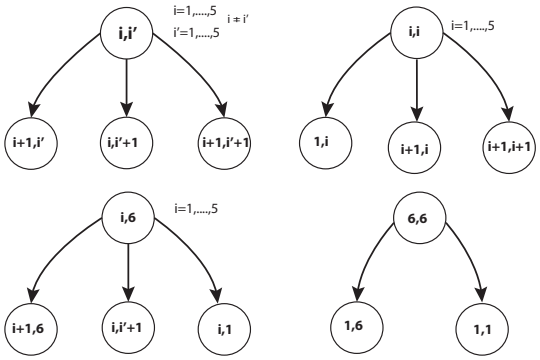

Figure 15: Oriented graph rules

To appreciate the *modus operandi* of the proposed scheme, let us analyse the robot behaviour from the kitchen to the counter shown in Fig. 14. First, at $t = 95\,s$ the obstacle scenario change $\overbrace{(5,6)}^{20} \rightarrow \overbrace{(1,5)}^{5}$ (see Fig. 16) hampers the transit of the robot towards the counter because the occurrence of the obstacle zone $\mathbf{5}$ and the fact that $x(95) \in \mathcal{T}_{55}^{20}$ do not guarantee the obstacle avoidance. Since in principle the sequence $\{\mathcal{T}_s^5\}$ should be used, a switching to $\{\mathcal{T}_s^{Ob_2^5}\}$ is initially performed and, as a consequence, the robot trajectory deviates from the *ideal path*, see Fig. 14.

Later, at $t = 126\,s$ the switching $\overbrace{(2,5)}^{10} \rightarrow \overbrace{(2,6)}^{11}$ occurs, the area $\mathbf{5}$ is walkable and therefore the strategy is capable to select the "optimal" set sequence, i.e. $\{\mathcal{T}_s^{11}\}$.
Finally, from $t = 180\,s$ forward the robot proceeds towards the target by exploiting the ellipsoidal sequence $\{\mathcal{T}_s^8\}$ and at $t = 200\,s$ reaches the terminal region $\{\mathcal{T}_0^8\}$ where the counter target is located. The same reasoning holds true for the opposite path counter-to-kitchen. For the sake of completeness, the command input behaviour is reported in Fig. 17.

Finally the computational burdens are detailed in Table 3. By comparing these results with those related to the modified *Cell decomposition* algorithm, it appears that this road-map method requires $0.94\,s$ per step (very close to the sampling time $T_c = 1\,s$) while the on-line computational load of the proposed scheme is unchanged w.r.t. the *Barrier configuration* example. The

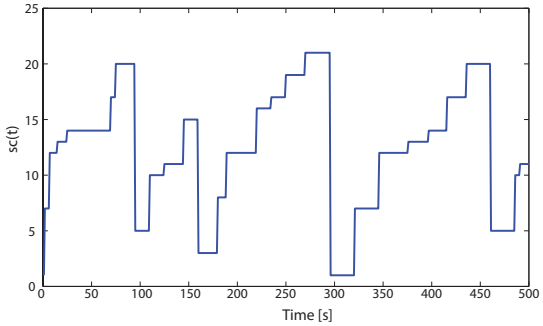

Figure 16: Obstacle scenario switchings

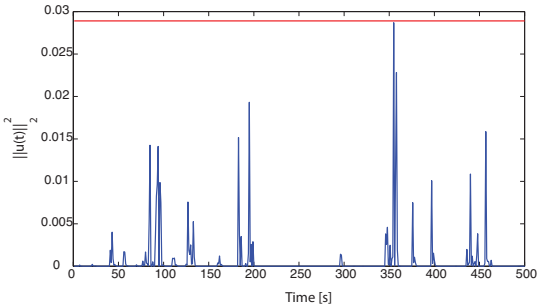

Figure 17: Command input

main consequence of this analysis is that, even if the **OA-MPC** off-line computational burdens grows up with the obstacle scenario number, the on-line phase is always computational affordable. On the other hand road-map methods present the non-negligible disadvantage of increasing on-line loads when complex dynamic environments are taken into consideration.

Table 3: Numerical burdens: average CPU time (seconds)

Off-line burdens	Basic sequences:	1800
	Obstacle sequences:	300
	Scenario switching sequences:	800
		2900
On-line burdens		0.05

7. Conclusions

In this paper we have presented a receding horizon strategy for solving the obstacle avoidance motion planning problem for autonomous vehicles described by linear time-invariant systems within dynamic environments. Set-theoretic ideas have been used to take care of all admissible time-varying obstacle scenarios. From a theoretical point of view, the proposed strategy allows to move in the off-line phase most of computations pertaining to the RHC controller design so that the overall framework becomes appealing in practical applications. On the other hand, the numerical comparisons have put in light that the idea to combine in a *single step* the design of the path planning module and of the predictive controller allows to deal with critical

obstacle scenarios that are not easily worked by dynamic path planning units.

[1] E. U. Acar, H. Choset, A. A. Rizzi, P. Atkar and D. Hull, "Morse decompositions for coverage tasks", *International Journal of Robotics Research*, Vol. 21, pp. 331-344, 2002.

[2] D. Angeli, A. Casavola, G. Franzè and E. Mosca, "An Ellipsoidal Off-line MPC Scheme for Uncertain Polytopic Discrete-time Systems", *Automatica*, Vol. 44, pp. 3113–3119, 2008.

[3] A. Bicchi and L. Pallottino, "On optimal cooperative conflict resolution for air traffic management systems", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, pp. 221-231, 2000.

[4] F. Blanchini and S. Miani. "Set-Theoretic Methods in Control", *Birkäuser*, Boston, 2008.

[5] T. Brogårdh, "Present and future robot control development - An industrial perspective", *Annual Reviews in Control*, Vol. 31, pp. 69-79, 2007.

[6] A. Dax, "The distance between two convex sets", *Linear Algebra and its Applications*, Vol. 416, pp. 184–213, 2006.

[7] N. E. Du Toit and J. W. Burdick, "Robot motion planning in dynamic uncertain environments", *IEEE Transaction on Robotics*, Vol. 28, No. 1, pp. 101-115, 2012.

[8] F. Fadaie and M. E. Broucke, "On the least restrictive control for collision avoidance of two unicycles", *International Journal of Robust and Nonlinear Control*, Vol. 16, No. 12, pp. 553–574, 2006.

[9] T. Fraichard and H. Asama, " Inevitable Collision States A Step Towards Safer Robot? ", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Las Vegas, NV (US), pp. 553–574, 2003.

[10] G. Franzè, D. Famularo and F. Tedesco, "Receding horizon control for constrained networked systems subject to data-losses", *50th IEEE CDC-ECC*, Orlando, FL, USA, pp. 5260-5265, 2011.

[11] Chung-Han Hsieh and Jing-Sin Liu, "Nonlinear Model Predictive Control for Wheeled Mobile Robot in Dynamic Environment", *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Kaohsiung, Taiwan, 2012.

[12] J. Huang, C. Wen, W. Wang and Z.-P. Jiang, "Adaptive stabilization and tracking control of a nonholonomic mobile robot with input saturation and disturbance", *Systems & Control Letters*, Vol. 62, No. 3, pp. 234-241, 2013.

[13] L. Lapierre, R. Zapata and P. Lepinay, "Combined path-following and obstacle avoidance control of a wheeled robot", *International Journal of Robotics Research* Vol. 26, pp. 361–375, 2007.

[14] J. Latombe, "Robot Motion Planning", *Kluwer Academic Publishers*, London, 1991.

[15] B. Llanas, "Efficient Computation of the Hausdorff Distance Between Polytopes by Exterior Random Covering", *Computational Optimization and Applications*, Vol. 30, pp. 161–194, 2005.

[16] M. Kvasnica, P. Grieder, M. Baotic and M. Morari, "Multi-Parametric Toolbox (MPT)", *Hybrid Systems: Computation and Control, Lecture Notes in Computer Science*, Vol. 2993/2004, pp. 121-124, 2004.

[17] M. V. Kothare, V. Balakrishnan and M. Morari, "Robust constrained model predictive control using linear matrix inequalities", *Automatica*, Vol. 32, pp. 1361–1379, 1996.

[18] A. Kurzhanski and I. Valyi, "Ellipsoidal calculus for estimation and control," *Berlin: Birkhäuser*, 1997.

[19] Y. Kuwata, T. Schouwenaars, A. Richards and J. How, "Robust Constrained Receding Horizon Control for Trajectory Planning", *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, California, 2005.

[20] J. Minguez, F. Lamiraux and J.-P. Laumond, "Motion Planning and Obstacle Avoidance", In:B. Siciliano and O. Khatib (Eds), Springer Handbook of Robotics, SpringerVerlag, pp. 827–852, 2008.

[21] J. Minguez and L. Montano, "Robot navigation in very complex, dense and cluttered indoor/outdoor environments", *15th IFAC World Congress*, pp. 218–223, 2002.

[22] D. Nieuwenhuisen, J. van den Berg and Mark Overmars, "Efficient Path Planning in Changing Environments", *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3295–3301, 2007.

[23] P. Ögren and N. E. Leonard, "A Convergent Dynamic Window Approach to Obstacle Avoidance", *IEEE Trans. Robotics and Automation*, Vol. 21, No. 2, pp. 188-195, 2005.

[24] S. Petti and T. Fraichard, "Safe Motion Planning in Dynamic Environments", *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 188-195, 2005.

[25] J. A. Primbs, V. Nevistic and J. C. Doyle, "Nonlinear optimal control: A control Lyapunov function and receding horizon perspective", *Asian J. Control*, Vol. 1, No. 1, pp. 14-24, 1999.

[26] Z. Qu, J. Wang and C. E. Plaisted, "A New Analytical Solution to Mobile Robot Trajectory Generation in the Presence of Moving Obstacles", *IEEE Transaction on Robotics*, Vol. 20, No. 6, pp. 978-993, 2004.

[27] S. Rakovic and D. Mayne, "Robust model predictive control for obstacle avoidance: Discrete time case", *Lecture Notes in Control and Information Sciences*, Vol. 358, No. 1, pp. 617-627, 2007.

[28] V. Sezer and M. Gokasan, "A novel obstacle avoidance algorithm: Follow the Gap Method" ", *Robotics and Autonomous Systems*, Vol. 60, pp. 1123-1134, 2012.

[29] A. Sgorbissa, R Zaccaria, "Planning and obstacle avoidance in mobile robotics", *Robotics and Autonomous Systems*, Vol. 60, pp. 628-638, 2012.

[30] I. Shames, M. Deghat and B.D.O. Anderson, "Safe formation control with obstacle avoidance", *18th IFAC World Congress*, Milano (Italy), pp. 11252-11257, 2001.

[31] Y. Yoon, J. Shin, H.J. Kim, Y. Park and S. Sastry, "Model-predictive active steering and obstacle avoidance for autonomous ground vehicles", *Control Engineering Practice*, Vol. 17, pp. 741-750, 2009.