UKACC International Conference on Control 2012
Cardiff, UK, 3-5 September 2012

Predictive Control with Trajectory Planning in the Presence of Obstacles

Rubens Junqueira Magalhães Afonso, Roberto Kawakami Harrop Galvão and Karl Heinz Kienitz
Instituto Tecnológico de Aeronáutica, Divisão de Engenharia Eletrônica,
12228-900 São José dos Campos, SP, Brasil
Emails: rubensjm@ita.br, kawakami@ita.br, kienitz@ita.br

Abstract—**In this work, a trajectory planning technique for an autonomous vehicle is proposed. A Predictive Control formulation is used both to plan a trajectory and control the vehicle in the presence of obstacles and dynamic constraints. However, some particularities of this sort of missions may make the time required for solution of the associated optimization problem prohibitive for a given sampling period. In this context, the possibility of using smaller prediction and control horizons is important to obtain a suitable control sequence within each sampling time. For this purpose, a trajectory planner which distributes waypoints along a previously established path is employed in the present paper. Each waypoint is determined so that it can be reached in a horizon which is smaller than the one necessary to reach the target set from the initial position, thus reducing the computational burden during the control phase. Moreover, during the planning phase the waypoints are chosen under the restriction that the target set should be reached within finite time so that the mission can be accomplished.**

Index Terms—**Predictive control, trajectory planning, waypoint.**

I. Introduction

Model-based Predictive Control (MPC) techniques involve the solution of an optimal control problem within a moving horizon, which is repeated (usually at every sampling time) on the basis of feedback from the sensors of the plant [1], [2]. One of the main advantages of MPC is the explicit treatment of constraints over the outputs and the controls of the plant. In aeronautical applications, it allows the various aircraft actuators to work closer to the limits of saturation, providing an increase in the flight envelope without compromising the safety of operation.

In typical problems of guidance of vehicles the state must reach a given set in finite time for the mission to be completed successfully [3]. Another issue is the existence of constraints that result in a non-convex optimization problem, such as the presence of obstacles which the vehicle must avoid.

In [3] the problem of reaching a terminal set in finite time is addressed by using a variable horizon MPC formulation and minimizing a weighted-time-fuel cost function. The resulting optimization problem is a *Mixed Integer Linear Programming* (MILP) one, because it involves both continuous and logical variables. A kinematic model of the vehicle is used to determine the trajectory, but vehicle guidance and control can be carried out by enhancing the model with rigid body and/or actuator dynamics. The obstacle avoidance constraints can also be encoded in the MILP formulation by using logical variables in conjunction with a "big-M" method, thus circumventing the difficulties brought about by the loss of convexity of the optimization problem.

However, the resulting MILP problem may not be feasible with small horizons, causing the need of larger ones in order to reach the terminal set from the initial state. In turn, larger horizons are associated with a higher number of optimization variables and may render the computational treatment of the optimization problem impracticable.

In this scenario, it may be convenient to split the mission into a series of intermediate goals that can be achieved within smaller horizons. Nevertheless, this division should be done judiciously, considering information about the limits of the actuators and constraints on the states of the plant, as well as the obstacles and the final goal of the mission. One way to accomplish the division is to introduce a sequence of waypoints, i. e., intermediate points through which the vehicle must pass to reach the terminal set. Such waypoints must be followed in a predetermined sequence obeying constraints and leading to the terminal set. Their determination must also take into account the capacity of achieving the waypoints from the current position of the vehicle within a horizon of acceptable size. Thus, the problem is divided into two steps: 1) off-line trajectory planning - involving the calculation of waypoints; 2) online execution of the planned trajectory. The first step is performed before the start of the maneuver.

In this work a technique for trajectory planning via waypoints in the presence of obstacles is proposed. The employment of such a technique along with an MPC-MILP formulation is evaluated regarding the computational burden involved in the control task.

The remainder of this paper is organized as follows. Section II reviews the MPC-MILP formulation adopted in the present work, which involves minimizing the weighted-time-fuel cost function to reach a given terminal set in the presence of obstacles [3]. Next, in Section III, trajectory planning for vehicles in the presence of obstacles is briefly discussed. The main contribution of this work is introduced in Section IV, in which the proposed approach for trajectory planning is presented. The scenarios adopted in the simulations are described in Section V. Section VI presents the simulation results of the proposed approach, which are compared to the direct application of the original MPC-MILP formulation. Finally, conclusions are drawn and suggestions for future work

978-1-4673-1558-6/12/$31.00 ©2012 IEEE

are given in section VII.

A. Notation

- $x \in \mathbb{R}^n$: plant state;
- $x_0 \in \mathbb{R}^n$: initial plant state;
- $u \in \mathbb{R}^p$: control signal;
- $r \in \mathbb{R}^2$: vehicle position;
- $b \in \{0,1\}$: binary variable associated to the horizon minimization;
- $b^{obs}_{i,m} \in \{0,1\}$: binary variables associated to the obstacle avoidance constraints;
- k: current time;
- $\hat{\diamondsuit}(k+i|k)$: predicted value of the variable $\diamondsuit$ at time $k+i$ based on the information available up to time k;
- $\diamondsuit^*$: optimal value of the variable $\diamondsuit$;
- $N(k) \in \mathbb{N}$: MPC control and prediction horizon;
- $C_r \in \mathbb{R}^{2\times n}$: matrix that extracts position information from the state vector;
- $\mathbb{U}(j) \subset \mathbb{R}^p$: set of admissible control values at time j;
- $\mathbb{X}(j) \subset \mathbb{R}^n$: set of admissible state values at time j;
- $\mathbb{Q}(N(k)+1) \subset \mathbb{R}^n$: set of terminal state values at the end of the horizon;
- $\mathcal{Z}_m \subset \mathbb{R}^2$: polygon defining the m-th obstacle;
- $V_i \in \mathbb{R}^2$: i-th vertex in the planned path;
- $\bar{N} \in \mathbb{N}$: maximal horizon in the one-step formulation;
- $\bar{N}_P \in \mathbb{N}$: maximal horizon between waypoints;
- $N_{WP} \in \mathbb{N}$: number of waypoints;
- $N_{obs} \in \mathbb{N}$: number of obstacles;
- $N_f \in \mathbb{N}$: number of sides in each obstacle;
- $N_V \in \mathbb{N}$: number of vertices in the planned path;
- $\alpha_i \in \mathbb{R}$: variable that determines the position of the i-th waypoint along the planned path;
- $M \in \mathbb{R}_+$: constant large enough to make terminal constraints inactive;
- $M_x \in \mathbb{R}_+$: constant large enough to make state constraints inactive;
- $M_u \in \mathbb{R}_+$: constant large enough to make control constraints inactive;
- $M_{obs} \in \mathbb{R}_+$: constant large enough to make obstacle avoidance constraints inactive;
- $M_{WP} \in \mathbb{R}_+$: constant large enough to make waypoint location constraints inactive;
- $r_x \in \mathbb{R}$: position along a coordinate axis in a horizontal plane regarding an arbitrary origin;
- $r_y \in \mathbb{R}$: position along a coordinate axis (perpendicular to the first) in a horizontal plane regarding an arbitrary origin;
- $v_x \in \mathbb{R}$: velocity regarding the r_x position;
- $v_y \in \mathbb{R}$: velocity regarding the r_y position;
- $a_x \in \mathbb{R}$: acceleration regarding the v_x velocity;
- $a_y \in \mathbb{R}$: acceleration regarding the v_y velocity;
- $\gamma \in \mathbb{R}$: weight of the term associated to the fuel consumption in the cost function;
- $\mathbf{1}_\diamondsuit \in \mathbb{R}^\diamondsuit$: column vector of $\diamondsuit$ elements equal to 1;
- $\|\diamondsuit\|_1$: 1-norm of the vector $\diamondsuit$.

II. Predictive Control

As depicted in Fig. 1, the basic elements of a predictive controller operating in discrete time are:

- A model used to predict the state of the plant over a horizon of N steps in the future, based on the current state $x(k)$ and the control sequence $\{\hat{u}(k+j|k)\},\ j = 0,\dots,N-1$ to be applied.
- An algorithm to optimize the control sequence regarding the cost function specified for the problem and the existing constraints on inputs and states of the plant.

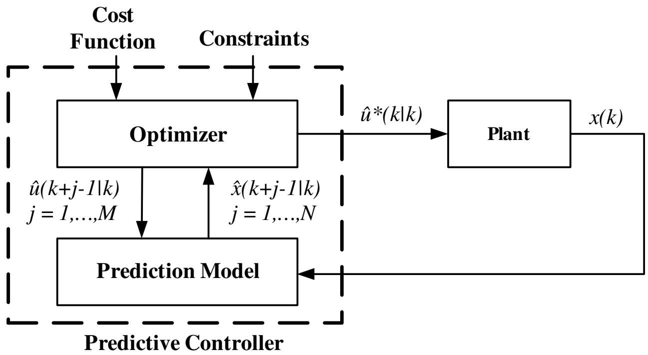

Fig. 1. Predictive control loop using state feedback.

In [3] the cost function is of the form:

$$J\left[\hat{x}(\cdot|k),\hat{u}(\cdot|k),N(k)\right] = \sum_{j=0}^{N(k)} \left(1 + \gamma \left\|\hat{u}(k+j|k)\right\|_1\right),\ \ \gamma > 0 \tag{1}$$

subject to

$$\hat{x}(k+j|k) = \begin{cases} x(k),\ j = 0 \\ A\hat{x}(k+j-1|k) + B\hat{u}(k+j-1|k),\ j > 0 \end{cases} \tag{2a}$$

$$\hat{x}(k+j|k) \in \mathbb{X}(j),\ \ j = 1,\dots,N(k) \tag{2b}$$

$$\hat{u}(k+j|k) \in \mathbb{U}(j),\ \ j = 0,\dots,N(k) \tag{2c}$$

$$\hat{x}(k+N(k)+1|k) \in \mathbb{Q}(N(k)+1) \tag{2d}$$

In the present work, robustness to unknown disturbances is not addressed in order to simplify the presentation of the main contribution, which will be stated in Section III. Therefore, the dependence of the sets $\mathbb{X}$, $\mathbb{U}$ and $\mathbb{Q}$ on j and $N(k)$ is disregarded.

It can be seen from Eq. (1) that a compromise between the time to reach the terminal set and the fuel spent during the task is achieved by penalizing the time in the first term of the cost function and the fuel expense in the second. By manipulating the weight γ, the planner can be adjusted to put more emphasis in time minimization (small values of γ) or fuel expense minimization (large values of γ).

This cost is denoted simply by $J(k)$ to indicate that it is a function to be optimized at the sampling time k.

The optimal control sequence $\{\hat{u}^*(k+j|k), j = 0,\dots,N(k)\}$ that minimizes the cost given by Eq. (1) subject to the constraints in Eqs. (2a), (2b), (2c) and (2d) usually cannot be analytically determined. Therefore, an optimization algorithm has to be used to obtain the control sequence subject to constraints. Customarily, an strategy known as “receding horizon” [2] is applied, i. e., only the first element of the control sequence is applied to the plant ($u(k) = \hat{u}^*(k|k)$) and the optimization is repeated at the next sampling time, making $u(k+1) = \hat{u}^*(k+1|k+1)$.

A. Horizon minimization

If the terminal set is given in terms of linear constraints:

$$\begin{gathered} \mathbb{Q} = \{x : p_i^T x \leq q_i,\, i = 1, \ldots, N_Q\}, \\ p_i \in \mathbb{R}^n,\, q_i \in \mathbb{R},\, i = 1, \ldots, N_Q \end{gathered} \tag{3}$$

then the terminal constraints can be rewritten as:

$$p_i^T \hat{x}(k+j+1|k) \leq q_i + M[1 - b(j)],\, i = 1, \ldots, N_Q \tag{4}$$

with $b(j)$ defined as

$$b(j) = \begin{cases} 1, \text{ if } j = N(k), \\ 0, \text{ if } j \neq N(k) \end{cases} \tag{5}$$

The scalar M must be taken so that $M > p_i^T x - q_i$, $\forall i$ for all admissible x [4].

Thus, the cost can be recast in terms of a maximum preset value $\bar{N}$ for the horizon, that is

$$J(k) = \sum_{j=0}^{\bar{N}} \left(j b(j) + \gamma \left\| \hat{u}(k+j|k) \right\|_1 \right) \tag{6}$$

subject to (4) with the following additional constraints:

$$\sum_{j=0}^{\bar{N}} b(j) = 1 \tag{7}$$

The cost expressed in Eq. (6) coincides with the one in Eq. (1) if the optimal value $N^*(k)$ for the horizon is less than or equal to $\bar{N}$ and the optimal control is null after $N^*(k)$, i. e., $\hat{u}^*(k+j|k) = 0$, $j > N^*(k)$. This last condition is guaranteed as the constraints in Eqs. (2b) and (2c) are imposed only up to the horizon $N^*(k)$. After this horizon, there is no constraint to be satisfied and thus the minimization of $\left\| \hat{u}(k+j|k) \right\|_1$ for $j > N^*(k)$ results in a zero control.

The state and control constraints up to the horizon $\bar{N}$ are rewritten in [3] using scalars large enough so that they become inactive after $N(k)$. Indeed, let the sets of admissible states and controls be

$$\begin{gathered} \mathbb{X} = \{x : r_{i,x}^T x \leq q_i^x,\, i = 1, \ldots, N_x\}, \\ \mathbb{U} = \{u : r_{l,u}^T u \leq q_l^u,\, l = 1, \ldots, N_u\}, \\ r_{i,x} \in \mathbb{R}^n,\, r_{l,u} \in \mathbb{R}^p,\, q_i^x,\, q_l^u \in \mathbb{R}, \\ i = 1, \ldots, N_x,\, l = 1, \ldots, N_u \end{gathered} \tag{8}$$

The constraints on the states and controls can then be rewritten as

$$\begin{aligned} & r_{i,x}^T \hat{x}(k+j|k) \leq q_i^x + M_x \sum_{m=1}^{j-1} b(m),\, i = 1, \ldots, N_x \\ & r_{l,u}^T \hat{u}(k+j-1|k) \leq q_l^u + M_u \sum_{m=1}^{j-1} b(m),\, l = 1, \ldots, N_u \end{aligned} \tag{9}$$

which makes the constraints inactive for $j > N(k)$ as $b(N(k)) = 1$. For this purpose, M_x must be such that $M_x > r_{x,i}^T x - q_i^x$, $\forall i$, for all x reachable in up to $\bar{N}$ steps from the terminal set with null control. M_u is a scalar that renders the inequalities inactive for all admissible values of u.

Therefore the problem is defined with a fixed horizon $\bar{N}$ and a linear cost involving real and integer variables subject to linear constraints. Thus, algorithms for MILP can be used to obtain the optimal control sequence.

B. Obstacle avoidance

Obstacles such as buildings, hills and dangerous areas to be avoided are commonly present in problems of vehicle guidance. The obstacle avoidance constraints lead to the loss of convexity of the optimization problem that has to be solved in order to calculate the control sequence. The present work adopts the formulation for the avoidance of obstacles presented in [5] and [3], which introduces a set of binary variables for each obstacle.

The constraint that the trajectory in space does not cross an obstacle can be written as $r = C_r x \notin \mathcal{Z}_m$, in which $\mathcal{Z}_m = \{r | P_m^{obs} r \leq q_m^{obs}\}$. Without loss of generality, all obstacles will be assumed to have the same number of sides N_f. It is therefore required that the position r is not in the sets $\mathcal{Z}_m$, $1 \leq m \leq N_{obs}$ at each sampling time, which is equivalent to imposing that the sets $\mathcal{I}_m = \{i \in \{1, \ldots, N_f\} : P_{i,m}^{obs} r > q_{i,m}^{obs}\} \neq \varnothing$, where $P_{i,m}$ is the i-th row of P_m^{obs} and $q_{i,m}$, the i-th element of q_m^{obs}. To this end, binary variables can be used as follows:

$$\begin{aligned} & -P_{i,m}^{obs} r(k+j) \leq -q_{i,m}^{obs} + M_{obs}[1 - b_{i,m}^{obs}(k+j)] - \epsilon \\ & \sum_{i=1}^{N_f} b_{i,m}^{obs}(k+j) \geq 1,\, b_{i,m}^{obs} \in \{0,1\}, \\ & 1 \leq j \leq \bar{N},\, 1 \leq m \leq N_{obs} \end{aligned}$$

Thus, with a large enough scalar M_{obs}, when $b_{i,m}^{obs}(k+j) = 0$, the constraint becomes inactive. If $b_{i,m}^{obs}(k+j) = 1$, the constraint is effectively enforced. The condition $\sum_{i=1}^{N_f} b_{i,m}^{obs}(k+j) \geq 1$ requires that at least one of the constraints is active at every sampling time, ensuring that the position r is “outside” the m-th obstacle. $\epsilon > 0$ is chosen arbitrarily small so that the inequality “$\leq$” becomes “$<$”, thus removing the border of the obstacle from the set of allowed positions.

III. TRAJECTORY PLANNING ARCHITECTURE

Path planning refers to the search for a curve in two or three dimensions connecting the starting point to the goal point or terminal set and avoiding obstacles. If the planning is successful, a set of positions that the vehicle must occupy to reach the destination is produced. However, the dynamic constraints of the vehicle are not taken into account. In contrast, the problem of trajectory planning includes dynamic constraints. So the result must also include a sequence of velocity vectors associated to the position of the vehicle.

In the context of aircraft guidance and control, the path planning involves a number of issues in addition to the avoidance of obstacles. Such issues include the presence of dynamic constraints, usually in the form of velocity and acceleration limits, the need for a feedback control strategy in real time in order to make the system robust to atmospheric disturbances, and constraints on the amount of fuel available to execute the maneuver. These factors contribute to increase the complexity of trajectory planning and control of aircraft in the presence of obstacles, often making the problem computationally intractable. This limits the application of established algorithms that have been developed in robotics and path planning for land vehicles [6].

Among the possible solutions for this problem, one that enjoys relative success divides the planning and control in hierarchical levels, counting often with layers that employ heuristics to reduce the computational load [6]. Thus, on the upper level, planning is carried out, in which dynamic constraints may be included and which may rely on some optimization criterion. Then, if the dynamic constraints have not yet been considered, it proceeds to a smoothing of the path to adapt to these constraints when possible and discarding it otherwise. The next step is the addition of time tags to the path, obtaining a trajectory. At this stage, one can employ some kind of optimality criterion to define the trajectory. Finally, this trajectory is used to generate references to the feedback controller. Also, criteria for an optimal control solution can be adopted. In this context, [6] presents a thorough review of the literature.

The technique proposed in the present paper involves three layers, namely:

1) A path planner which produces a path composed of the connection of successive straight-line segments connecting the initial position to the target set, while avoiding obstacles. Dynamics constraints are not considered in this layer.
2) A trajectory planner which determines waypoints along the planned path obtained from the first layer. The determination of the waypoints is done considering the dynamic constraints of the vehicle, the existence of obstacles, the arrival at the target set in finite time and the capacity to reach each waypoint from the previous one within a fixed small horizon.
3) A Predictive Control layer which employs the waypoints determined during the second phase as targets of predictive control problems with small horizon, until the last one is reached and the target set can be reached within the small horizon. In this phase, the dynamic and obstacle avoidance constraints are again enforced.

In the present work, the path planner is not addressed and a path which satisfies the conditions described above is assumed available. For the purpose of obtaining such a path, many techniques may be used, such as Voronoi graphs, probabilistic roadmaps, A^* search [7], and RRTs (Rapidly-exploring Random Trees) [8]. It is further assumed that the path is provided in the form of a sequence of vertices connected by straight-line segments.

The trajectory planning and control architecture adopted herein is depicted in Fig. 2. The dashed lines mark the blocks addressed in the present work. The MPC controller was discussed in the previous section. The trajectory planner provides a list of target waypoints in the order that should be followed to reach the final target set. The "Active target selection logic" simply checks whether the position of the vehicle is equal to the waypoint (up to a certain numerical tolerance); if true, the active target is the next waypoint in the sequence; otherwise, the active target remains the same. After the last waypoint is reached, the logic commutes to the final target set. Since the "Trajectory planner" passes only the waypoints to the control loop, and not every position, velocity and control signal used to reach them, and since the controller cost function and the planner one can be different, the planned trajectory and the one that is actually followed may in general present differences.

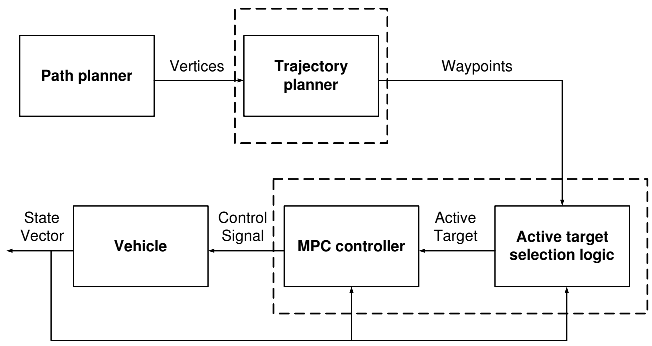

Fig. 2. Trajectory planning and control architecture used in this work.

IV. PROPOSED TRAJECTORY PLANNING TECHNIQUE

The technique proposed in this paper for trajectory planning involves the determination of a preset number of waypoints. These are scattered between the initial position of the vehicle and the terminal set. Their determination considers a horizon shorter than the one necessary to reach the terminal set from the initial state. Therefore, computational burden is expected to be lighter.

In order to limit the search space of solutions, the waypoints are constrained to a previously planned path, given in terms of the vertices of a sequence of straight-line segments which constitute a collision-free path from the initial position of the vehicle to the target set. This turns the problem of searching a two-dimensional space for a solution to a one-dimensional search over a piecewise linear curve.

In addition, the obstacle avoidance constraints are enforced in the optimal determination of the positions of the waypoints along the planned path in order to ensure the existence of collision-free trajectories between the waypoints. The problem of determining waypoint positions along the planned path composed of straight-line segments, avoiding obstacles and leading to the terminal set can be posed as follows:

Problem 4.1: Let N_{WP}, $\bar{N}_P$, N_{obs}, N_f, and $\{V_i\}$, $i = 1,\ldots,N_V+1$ be the preset number of waypoints, the maximal horizon to reach an waypoint from the previous one, the number of obstacles, the number of sides of each obstacle, and the ordered sequence of vertices whose connection via straight-line segments produces the collision-free path (V_1 is the initial position of the vehicle), respectively. The waypoint determination problem is stated as

$$\min_{\hat{u}(k+j|k),\, \alpha_i,\, b_l^{obs},\, b_{i,j}^{WP}} \sum_{i=1}^{N_{WP}} \alpha_i + \mu \sum_{i=1}^{N_{WP}} \sum_{j=1}^{N_V} j b_{i,j}^{WP} \tag{10}$$

s.t.

$$\begin{aligned}&\hat{r}(k+i\bar{N}_P|k) \leq (\alpha_i - N_V + j)V_{j+1} + \\ &+ \left[1-(\alpha_i - N_V + j)\right] V_j + M_{WP}(1-b_{i,j}^{WP}),\\ &1 \leq i \leq N_{WP},\ 1 \leq j \leq N_V\end{aligned} \tag{11a}$$

$$\begin{aligned}&-\hat{r}(k+i\bar{N}_P|k) \leq -\{(\alpha_i - N_V + j)V_{j+1} + \\ &+ \left[1-(\alpha_i - N_V + j)\right] V_j + M_{WP}(1-b_{i,j}^{WP})\},\\ &1 \leq i \leq N_{WP},\ 1 \leq j \leq N_V\end{aligned} \tag{11b}$$

$$\sum_{j=1}^{N_V} (N_V - j) b_{i,j}^{WP} \leq \alpha_i,\ 1 \leq i \leq N_{WP} \tag{11c}$$

$$\sum_{j=1}^{N_V} b_{i,j}^{WP} = 1,\ 1 \leq i \leq N_{WP} \tag{11d}$$

$$b_{i,j}^{WP} \in \{0,1\},\ 1 \leq i \leq N_{WP},\ 1 \leq j \leq N_V$$

$$0 \leq \alpha_{N_{WP}} \leq \alpha_{N_{WP}-1} \leq \cdots \leq \alpha_1 \leq N_V \tag{11e}$$

$$\hat{x}(k+(N_{WP}+1)\bar{N}_P|k) \in \mathbb{Q} \tag{11f}$$

$$\hat{u}(k+j|k) \in \mathbb{U},\ 0 \leq j \leq (N_{WP}+1)\bar{N}_P - 1 \tag{11g}$$

$$\hat{x}(k+j|k) \in \mathbb{X},\ 1 \leq j \leq (N_{WP}+1)\bar{N}_P - 1 \tag{11h}$$

$$P_m^{obs}\hat{x}(k+j|k) \leq -q_m^{obs} + M_{obs}(\mathbf{1}_{N_f} - b_m^{obs}(k+j|k)) \tag{11i}$$

$$\sum_{l=1}^{N_f} b_{l,m}^{obs}(k+j|k) \geq 1,\ b_{l,m}^{obs}(k+j|k) \in \{0,1\}, \tag{11j}$$

$$1 \leq j \leq (N_{WP}+1)\bar{N}_P,\ 1 \leq m \leq N_{obs}$$

where $\mu > 0$ is a scalar, $\hat{r}(k+i\bar{N}_P|k) = C_r\hat{x}(k+i\bar{N}_P|k)$ is the predicted position at the sampling time $(k+i\bar{N}_P)$, and P_m^{obs}, q_m^{obs}, and M_{obs} are defined as in section II-B.

The binary variables $b_{i,j}^{WP}$ are used to make the constraints in Eqs. (11a) and (11b) active or inactive. If $b_{i,j}^{WP} = 1$, then the inequalities (11a) and (11b) are active, which imposes an equality constraint restricting the position of the i-th waypoint to the straight-line segment between the j-th and $(j+1)$-th vertices. Otherwise, if $b_{i,j}^{WP} = 0$, the inequalities (11a) and (11b) are inactive for the particular values of i and j, meaning that the i-th waypoint is not located in the straight-line segment between the j-th and $(j+1)$-th vertices. For this purpose, the scalar M_{WP} is chosen large enough to render the constraints in Eqs. (11a) and (11b) inactive.

The inequalities in Eqs. (11a), (11b), (11c) and (11e) along with the equality in Eq. (11d) impose that the positions of the waypoints remain in one of the straight-line segments that compose the planned path. If $N_V - j \leq \alpha_i \leq N_V - j + 1$, then the i-th waypoint is located in the straight-line segment between vertices V_j and V_{j+1}. For instance, if $\alpha_i = N_V - j$, then the i-th waypoint is exactly at the j-th vertex of the planned path.

The first term of the cost function in Eq. (10) aims at minimizing the values of α_i, $1 \leq i \leq N_{WP}$, which prioritizes solutions that locate the waypoints farther from the initial position and closer to the terminal set, in order to avoid low initial speeds. The second term is introduced in order to obtain the maximal possible value to the term $(N_V - j)b_{i,j}^{WP}$, $1 \leq j \leq N_V$. As a consequence, the value of $(N_V - j)b_{i,j}^{WP}$ resulting from the minimization of this term subject to the constraint in Eq. (11c) is the greatest integer which is smaller or equal to α_i for any positive value of the scalar μ. This, in turn, means that the term $(\alpha_i - N_V + j)$ in the constraints (11a) and (11b) is restricted to the set $[0,1)$, thus resulting in a position between V_j and V_{j+1} for the i-th waypoint.

The resulting values of α_i, $1 \leq i \leq N_{WP}$ correspond to the positions of the waypoints between the initial position and the last vertex, each farther from the initial position than the one before. The last waypoint is chosen so that it is possible to reach the terminal set $\mathbb{Q}$ from this position within the horizon $\bar{N}_P$.

An example is presented in Fig. 3, in which two waypoints were used with a horizon $\bar{N}_P = 10$ for the system dynamics that will be described in Section V. The planned path contains three vertices ($N_V = 2$, since the initial position is an additional vertex): $V_1 = [0\ \ 0]^T$ (initial position), $V_2 = [1.2\ \ 0.4]^T$ (intermediate vertex) and $V_3 = [1.6\ \ 1.5]^T$ (vertex in the border of the target set). The values for α_1 and α_2 were 1.583 and 0.385, respectively. This means that the first waypoint should be between the vertices V_1 and V_2 and the second, between V_2 and V_3, which can be corroborated by the resulting positions of the waypoints depicted in Fig. 3.

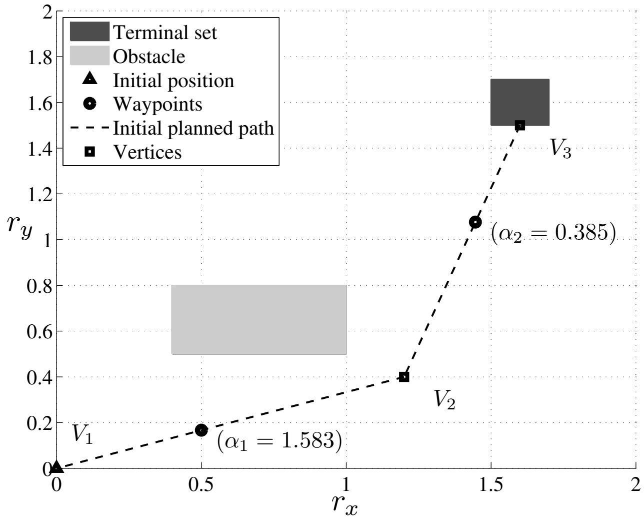

Fig. 3. Example of determination of the waypoints.

V. Simulation scenarios

A kinematic model describing the movement of a vehicle in two dimensions was employed for simulation. The continuous-time model equations are:

$$\dot{r}_x = v_x,\ \dot{v}_x = a_x,\ \dot{r}_y = v_y,\ \dot{v}_y = a_y \tag{12}$$

where r_x and r_y define the position of the vehicle in a horizontal plane with respect to an arbitrary origin. This equation can be recast in state-space form ($\dot{x} = A_c x + B_c u$) by defining the state and control vectors as $x = [r_x \quad v_x \quad r_y \quad v_y]^T$, $u = [a_x \quad a_y]^T$. For use in the proposed MPC approach with trajectory planning, a discrete-time model of the form $x(k+1) = Ax(k) + Bu(k)$ was obtained with

$$A = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0.5T^2 & 0 \\ T & 0 \\ 0 & 0.5T^2 \\ 0 & T \end{bmatrix} \tag{13}$$

in which T is the sampling period. For the simulations in this paper T was normalized to one time unit.

The dynamical constraints imposed on the velocities are $-1 \leq x_2,\ x_4 \leq 1$. As for the accelerations, $-1 \leq u_1,\ u_2 \leq 1$. Constraints $0 \leq x_1,\ x_3 \leq 2$ were also imposed on the position in order to limit it to the known terrain, over which information was assumed to be available.

The initial state of the vehicle was arbitrarily set to $x_0^T = [0\ 0\ 0\ 0]^T$, i. e., it started at rest. The goal was to reach a terminal set in the form of a rectangle described by the following inequalities on the positions $1.5 \leq x_1,\ x_3 \leq 1.7$.

As for the obstacles, they were also represented as rectangles with $0.6 \leq x_1^1, x_1^2 \leq 1$, $0.1 \leq x_3^1 \leq 0.8$, and $1.2 \leq x_3^1 \leq 1.6$, where the superscript refers to each of the two obstacles present in the simulations. It is worth noting that, since only the discrete-time predictions of the position are considered in the inequalities, this does not avoid stretches of the continuous-time trajectory crossing the obstacle. One alternative to handle this issue is proposed in [9], which involves incorporating restrictions on the transition of the vehicle to each region of the space defined by obstacle inequalities. However, it involves the introduction of more binary variables, increasing the complexity of the MILP problem. In this work, the length and width of the obstacle were expanded. To this end, an amount determined through the maximal admissible absolute value of the velocity in each axis was used to expand the borders of the obstacles. Therefore, the adopted avoidance constraints were constructed based on the following expanded obstacles: $0.5 \leq x_1^1, x_1^2 \leq 1.1$, $0 \leq x_3^1 \leq 0.9$, and $1.1 \leq x_3^1 \leq 1.7$.

The weight γ of the fuel in the cost function was set to 0.1. For the one-step solution, the maximal horizon was set to $\bar{N}_{OS} = 35$. Meanwhile, for the planner solution $\bar{N}_P = 8$ was adopted and the number of waypoints was set to $\bar{N}_{WP} = 3$. The computation times were taken as an average of 10 runs of each simulation, in order to eliminate fluctuations due to external factors. All simulations were carried out in a personal computer equipped with a Pentium® Dual-Core E5400 processor with $2.7GHz$ clock. For solution of the MILP, the CPLEX toolbox from IBM ILOG was used in Matlab environment, under an academic license.

VI. Results and discussion

Initially, the simulation was carried out with the controller employing the one-step solution, i. e., trying to reach the terminal set from the beginning. The resulting path is presented in Fig. 4. The terminal set (dark gray rectangle) was reached successfully and the obstacles (light gray rectangles) were avoided. Moreover, as shown in Fig. 5, the accelerations a_x and a_y (which correspond to the controls u_1 and u_2, respectively) remained within the ± 1 bounds. It took 24 sample periods to reach the terminal set from the starting position. The fuel cost was 28.4 and the average computation time was $17.72s$. The highest computational time was $4.22s$ and the mean computational time was $0.74s$. It can also be noted that a stretch of the continuous-time path crosses the prohibited region (black rectangle), but not the original obstacle. This justifies the choice to expand the original obstacle as means to avoid collisions.

The second simulation introduces the waypoint guidance using a previously planned path. An arbitrary path that connects the initial position to the terminal set while avoiding obstacles was employed for illustration, as shown in Fig. 6. The terminal set was reached successfully and the obstacles were avoided. Again, as shown in Fig. 7, the accelerations a_x and a_y remained within the ± 1 bounds. It took 32 sample periods from the initial position to the terminal set. The planning phase lasted $0.33s$. The maximal computational time in the control phase was $0.18s$ and the mean was $0.08s$. Therefore, the total time to plan and execute the trajectory was about $3.03s$, which is much smaller than the time required by the one-step planner. The fuel cost was 52.57, which is larger than the one obtained with the one-step solution. In fact, since the waypoints are restricted to the previously planned

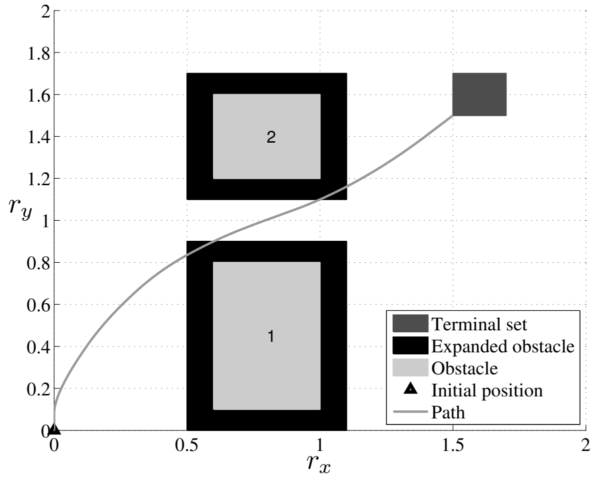

Fig. 4. Path obtained with one-step solution.

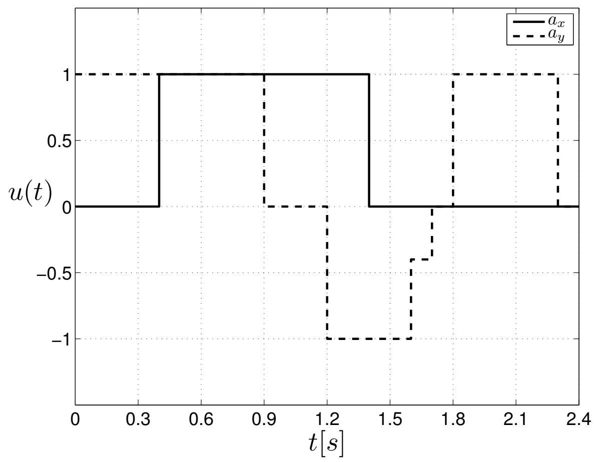

Fig. 5. Control signal obtained with one-step solution.

path that avoids the obstacles, and the horizon is shorter, the minimization of the fuel expense is compromised. It is interesting to note that the path does not cross the prohibited region, due to the fact that the waypoints steer the trajectory to the planned path, which is distant from the expanded obstacles. If the path planner emphasizes a safe path instead of the one that demands less fuel or the shortest one, it is likely to obtain a safer trajectory at the cost of a larger fuel consumption.

VII. Conclusions

The proposed trajectory planning approach conduced to a feasible trajectory and reduced considerably the computational burden. The results showed also that a compromise between optimality and computation time can be inferred. This is suggested by the greater fuel cost and maneuver time obtained with the proposed approach as compared to the one-step MILP approach.

Future works could include a formulation which provides robustness to an unknown but limited disturbance in the trajectory planning phase, as was done in [3] for the MPC MILP formulation.

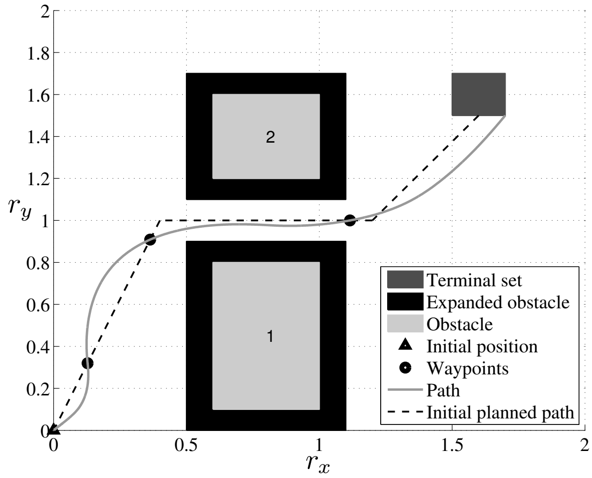

Fig. 6. Path obtained with trajectory planning.

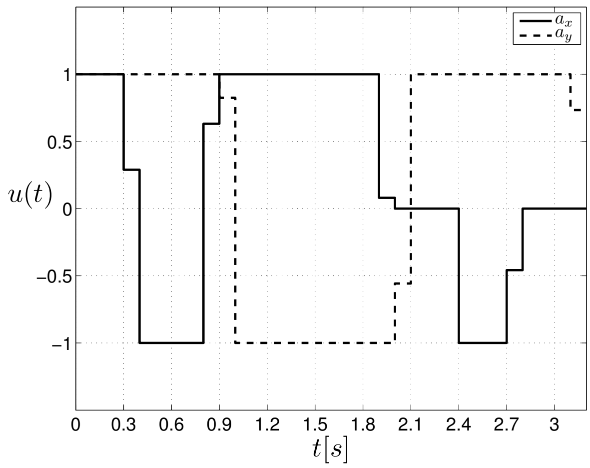

Fig. 7. Control signal obtained with trajectory planning.

Acknowledgments

The authors acknowledge the support of FAPESP (2011/18632-8, 2011/17610-0) and CNPq (research fellowships).

References

[1] E. F. Camacho and C. Bordons, *Model Predictive Control*. London: Springer-Verlag, 1999.

[2] J. A. Rossiter, *Model-based Predictive Control: a practical approach*. Boca Raton: CRC Press, 2003.

[3] A. Richards and J. P. How, "Robust variable horizon model predictive control for vehicle maneuvering," *Int. J. Robust and Non-linear Control*, vol. 16, no. 7, pp. 333 – 351, 2006.

[4] A. Bemporad and M. Morari, "Control of systems intregrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407 – 427, 1999.

[5] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *J. Guid. Control Dyn.*, vol. 25, no. 4, pp. 755 – 764, 2002.

[6] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *J. Intell. and Robotic Systems*, vol. 57, no. 1, pp. 65 – 100, 2010.

[7] S. Russel and P. Norvig, *Artificial Intelligence: a modern approach*, 3rd ed. Prentice Hall, 2009.

[8] S. LaValle, "Motion planning: the essentials," *IEEE Robotics and Automation Society Magazine*, vol. 18, no. 1, pp. 79 – 89, 2011.

[9] M. H. Maia and R. K. H. Galvão, "On the use of mixed-integer linear programming for predictive control with avoidance constraints," *Int. J. Robust and Non-linear Control*, vol. 19, pp. 822 – 828, 2009.