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

Multitask Trajectory Planning based on Predictive Control

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 Predictive Control formulation for trajectory planning with multiple target sets is proposed, which solves the problem of performing all tasks in finite time via minimization of a weighted-time-fuel cost function, generating a feasible trajectory. An approach involving a procedure to order the list of the target sets to be visited in terms of the distance between them is used for comparison and it is shown that the proposed technique outperforms this approach in terms of time and fuel spent to accomplish the mission.**

***Index Terms*—Predictive control, trajectory planning, target set.**

I. Introduction

In the context of aircraft guidance and control, the path planning problem becomes more complex than the simple search for a curve that connects the starting point to the goal while avoiding obstacles. Some of the reasons for that 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 [1].

Model-based Predictive Control (MPC) techniques have been increasingly employed in the aeronautical industry [2] due to their ability to handle constraints on inputs and states of the plant [3]. More recently, MPC formulations as the one proposed in [4] have been used to perform trajectory planning for autonomous vehicles. Many particularities of the trajectory planning problem have been addressed in [4], namely the task of reaching a terminal set in finite time, while avoiding obstacles. Through the introduction of a variable horizon, it is possible to calculate the smallest horizon needed to reach the terminal set by solving a *Mixed Integer Linear Programming* (MILP) problem. Thus, a minimum time trajectory between a source point and a target set can be determined by using a kinematic model of the vehicle. Moreover, by enhancing the model with rigid body dynamics and characteristics of actuators, this problem can be extended to a more elaborate guidance and vehicle control framework. The MILP formulation also circumvents the difficulties brought about by the introduction of obstacles, particularly the loss of convexity of the set of admissible solutions.

However, a mission may require that the vehicle visits a number of sets. In this scenario, it may be convenient to consider all sets in the trajectory planning in order to minimize the total time to carry out all the tasks (visit all the target sets), instead of setting a single target set to be reached at every step.

In the present work a formulation to solve the trajectory planning problem with multiple target sets (termed Multitask Trajectory Planning) in the presence of obstacles is proposed. Simulation results with a fictional vehicle are presented in order to illustrate the success of the proposed technique. For comparison, the formulation presented in [4] with a single terminal target set was used with a list of sets to be visited, which was updated upon reaching each one of them. This list was ordered according to a criterion based on minimal distance between the initial position of the vehicle and the first target set as well as between the current target set and the next one.

The remainder of this paper is organized as follows. In Section II, the Predictive Control formulation adopted in the present work and proposed in [4] is presented, which involves minimizing a weighted-time-fuel cost function that penalizes the time to reach a given terminal set in the presence of obstacles. The approach for trajectory planning with multiple target sets is proposed later in Section III. Section IV presents the set ordering technique used for comparison. Simulation results of the application of both approaches are presented and discussed in Section V. Finally, conclusions are drawn and suggestions for future work are given in Section VI.

II. MPC Formulation

The formulation employed in this work is similar to the one adopted in [4], with exception of the contribution related to the inclusion of multiple target sets, which will be introduced in Section III. The problem is recast in a *Mixed-Integer Linear Programming* (MILP) form much in the same way as in [4], again with exception of the inclusion of multiple target sets in Section III.

Figure 1 presents the basic elements of a predictive controller operating in discrete time, namely:

- 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 $\{\hat{u}(k+j|k)\},\ j = 0, \ldots, N-1$ 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.

The notation used is as follows: $u \in \mathbb{R}^p$ and $x \in \mathbb{R}^n$ denote the input and state variables of the plant, respectively. $\hat{\diamond}(k+$

$j|k)$ denotes the predicted value of variable $\diamondsuit$ at time $k+j$ ($j \geq 1$) based on information available up to time k. The optimal control to be applied to the plant at time k is denoted by $\hat{u}^*(k|k)$.

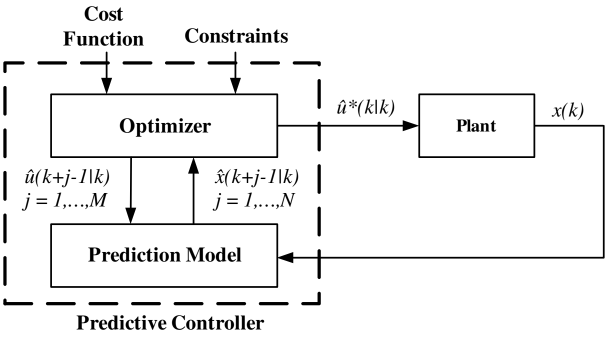

Fig. 1. Predictive control loop using state feedback.

In [4] the cost function is of the form:

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

subject to

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

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

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

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

where $\mathbb{U}(j)$ and $\mathbb{X}(j)$ are the sets of admissible controls and states, respectively, and $\mathbb{Q}(N(k)+1)$ is the terminal set. In [4] the dependence of the sets $\mathbb{X}$, $\mathbb{U}$ and $\mathbb{Q}$ in terms of j and $N(k)$ is inserted in order to provide robustness to an unknown but limited disturbance input. If this disturbance is disregarded, the sets are independent of time.

The first term of the cost function in Eq. (1) penalizes the time necessary to reach the target set. The second term, involving the norm of the control vector at each sampling time, penalizes the fuel spent. Thus, such a formulation provides a compromise between minimizing the time to achieve the goal and the fuel spent, which can be adjusted by the weight γ.

With a small abuse of notation, this cost is usually denoted simply by $J(k)$ in order to clarify that it is a function to be optimized at the k-th sampling time.

Optimization algorithms should be employed to obtain the optimal control sequence $\{\hat{u}^*(k+j|k), j=0,\ldots,N(k)\}$ that minimizes the cost given by Eq. (1) subject to the constraints of Eqs. (2a), (2b), (2c) and (2d). The first element of such a sequence is applied to the plant (i. e., $u(k) = \hat{u}^*(k|k)$) and the optimization is repeated at the next sampling time, making $u(k{+}1) = \hat{u}^*(k{+}1|k{+}1)$. This strategy is known as "receding horizon" [5].

A. Horizon minimization

If the terminal set is given in terms of linear inequalities

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

then the terminal constraints can be rewritten as:

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

where $M > 0$ is a constant and $b(j)$ is a binary decision variable 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 [6].

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

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

subject to 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 of equations (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.

In [4] the state and control constraints are also rewritten up to the horizon $\bar{N}$ using scalars large enough so that they become inactive after $N(k)$. Indeed, let the sets of admissible states and controls be

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

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

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

which makes the constraints inactive for $j > N(k)$ as $b(N(k)) = 1$. $M_x \in \mathbb{R}$ 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. $M_u > 0$ is a scalar large enough to render the control constraints 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

In problems involving the guidance of vehicles, the presence of obstacles is usual, such as buildings, hills, dangerous areas to avoid, among others. In the presence of such obstacles the set of admissible states will no longer be convex. In [7], a form of dealing with polygonal obstacles through the use of MILP was proposed.

The constraint that the trajectory in space does not cross the obstacle can be written as $r = C_r x \notin \mathcal{Z}$, in which matrix C_r extracts the position information from the state vector and $\mathcal{Z} = \{r | P^o r \leq q^o\}$ defines an obstacle in the form of a polygon with N_f sides. It is therefore required that the position r is not in the set $\mathcal{Z}$ at each sampling time, which is equivalent to imposing that the set $\mathcal{I} = \{i \in \{1, \ldots, N_f\} : P_i^o r > q_i^o\} \neq \varnothing$ where P_i is the i-th row of P^o and q_i, the i-th element of q^o. To this end, binary variables can be used as follows:

$$\begin{gathered} - P_i^o r(k+j) \leq -q_i^o + M_o[1 - b_i^o(k+j)] - \epsilon \\ \sum_{i=1}^{N_f} b_i^o(k+j) \geq 1,\, b_i^o \in \{0,1\},\, 1 \leq j \leq N \end{gathered}$$

Thus, when $b_i^o(k+j) = 1$, the constraint is effectively enforced. If $b_i^o(k+j) = 0$, with a large enough scalar $M_o > 0$, the constraint becomes inactive. The condition $\sum_{i=1}^{N_f} b_i^o(k+j) \geq 1$ requires that at least one of the constraints is active at every sampling time, ensuring that the position r is "outside" the 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. Proposed Multitask Trajectory Planning Technique

The contribution of the present paper is the proposition of a novel method to enhance the capability of the trajectory planning technique proposed in [4] to visit $N_{ts} \geq 1$ target sets $\mathbb{Q}_1, \mathbb{Q}_2, \ldots \mathbb{Q}_{N_{ts}}$. This inclusion is in accordance with real-world mission demands, which usually require that the autonomous vehicle visits more than one target. In this context, one alternative is to arrange all target sets in a sequence to be visited. This arrangement may be performed by considering some optimization criteria, such as minimal distance from the starting position of the vehicle to define the first set and then, minimal distance between sets, choosing the next set to be visited as the closest to the last visited one, until all sets have been included in the sequence. With this sequence at hand, one can divide the problem in N_{ts} single-target trajectory planning problems and solve each of them using the framework proposed in [4]. Upon reaching the target set for the current problem, it is replaced with the subsequent one in the pre-established order. This procedure is repeated until the final target set is reached. However, this may not yield the minimum time or minimum fuel solution, since the criteria employed to order the list of target sets do not consider the dynamics of the vehicle and constraints over the variables. Therefore, a framework which is capable of considering multiple target sets within the solution of the Predictive Control trajectory planning problem may bring about interesting results regarding the optimal solution to the trajectory planning problem.

It is assumed that the N_{ts} target sets are defined as in Eq. (3), in terms of linear inequalities:

$$\begin{gathered} \mathbb{Q}_h = \{x : p_{h,i}^T x \leq q_{h,i},\, i = 1, \ldots, N_{Q_h}\}, \\ p_{h,i} \in \mathbb{R}^n,\, q_{h,i} \in \mathbb{R},\, i = 1, \ldots, N_{Q_h}, \\ h = 1, \ldots, N_{ts} \end{gathered} \tag{10}$$

in which $\mathbb{Q}_h$ is the h-th target set and N_{Q_h} is the number of inequalities used to describe it.

Since the vehicle only needs to visit each target set at one sample time, the constraints can be rewritten as:

$$\begin{gathered} p_{h,i}^T \hat{x}(k+j+1|k) \leq q_{h,i} + M[1 - b_h(j)], \\ i = 1, \ldots, N_{Q_h},\, h = 1, \ldots, N_{ts}. \end{gathered} \tag{11}$$

where $M > 0$ is a constant and $b_h(j)$ is a binary decision variable defined as

$$b_h(j) = \begin{cases} 1, \text{ if } j = N_h, \\ 0, \text{ if } j \neq N_h \end{cases} \tag{12}$$

in which N_h is the number of sample times that the vehicle takes to reach the h-th target set from the current position. For example, if the vehicle takes 2 sample times to reach the first target set from the starting position and 3 more to go from the first target set to the second, then $N_1 = 2$ and $N_2 = 5$.

An auxiliary variable has to be introduced in order to consider the minimization of the horizon to visit all of the target sets. This variable in be defined as:

$$N_f = \max{(N_h)},\, h = 1, \ldots, N_{ts} \tag{13}$$

Through the introduction of a new sequence of binary variables $\{b_f(j),\, j = 1, \ldots, \bar{N}\}$, it is possible to penalize this horizon N_f to visit all of the target sets by rewriting the cost function in Eq. (6) as:

$$J(k) = \sum_{j=0}^{\bar{N}} \left(j b_f(j) + \gamma \left\| \hat{u}(k+j|k) \right\|_1\right) \tag{14}$$

with

$$b_f(j) = \begin{cases} 1, \text{ if } j = N_f, \\ 0, \text{ if } j \neq N_f \end{cases} \tag{15}$$

subject to additional constraints

$$\sum_{j=0}^{\bar{N}} b_h(j) = 1,\, h = 1, \ldots, N_{ts} \tag{16}$$

$$\sum_{j=0}^{\bar{N}} b_f(j) = 1, \tag{17}$$

$$b_f(j) \leq \frac{\sum_{h=1}^{N_{ts}} \sum_{i=0}^{j} b_h(i)}{N_{ts}} \tag{18}$$

The constraint in Eq. (16) ensures that each target set is visited at least once, whereas the ones in Eqs. (17) and (18) make sure that the definitions of Eqs. (13) and (15) are applied.

The only step left is to remove the state and control constraints after N_f. This is performed by replacing $b(m)$ in Eq. (9) by $b_f(m)$:

$$\begin{aligned} r_{i,x}^T \hat{x}(k+j|k) &\le q_i^x + M_x \sum_{m=1}^{j-1} b_f(m),\ i = 1, \ldots, N_x \\ r_{l,u}^T \hat{u}(k+j-1|k) &\le q_l^u + M_u \sum_{m=1}^{j-1} b_f(m),\ l = 1, \ldots, N_u \end{aligned} \tag{19}$$

The formulation presented in this section allows for planning a trajectory which visits all the sets while minimizing the cost function penalizing the overall time taken to perform the mission and the amount of fuel spent. However, when receding horizon feedback control has to be implemented, it is important to have a logic which removes a set that has already been visited, in order to avoid visiting the same sets over and over. The algorithm described bellow is proposed to circumvent this issue. This algorithm has to be run at every sampling time before calculating the control sequence:

Algorithm 1: Remove a set from the list after visiting it

```
for $i = 1 \rightarrow N_{ts}$ do
    if $x(k) \in SetList(i)$ then
        $SetList \leftarrow SetList \backslash SetList(i)$
        $N_{ts} \leftarrow N_{ts} - 1$
    end if
end for
```

□

in which the list of sets to be visited is given by $SetList$. The $\Diamond \backslash \spadesuit$ operator removes a set $\spadesuit$ from a set list $\Diamond$. After removing a set from the list, the number of binary variables can be reduced, since there are less sets to visit, which in turn means less constraints to be imposed.

IV. Minimum-distance Trajectory Planning Algorithm

In this Section a trajectory planning algorithm which divides the mission in several tasks is presented. This algorithm will be used for comparison with the technique proposed in Section III. It is basically composed of two parts:

1) Order the list of sets to visit according to the distance, i. e., the first set to be visited is the closest to the initial position, the second is the closest to the first, and so on.
2) Apply the formulation presented in [4] with the terminal set as the next to be visited in the ordered list, starting by the first and changing to the next subsequently after the current target is reached.

In the following, the first part will be detailed, as it is the most cumbersome one, since the second involves only a test of pertinence of a point to a set and an update to the list of target sets.

The first part can be divided into two main algorithms, requiring only a list of the N_{ts} target sets ($SetList$) in arbitrary order. In the first algorithm (Algorithm 2), x_0 is the initial state and C_r, a matrix that extracts position information from the state vector, thus $C_r x_0$ is the initial position of the vehicle. $dist(\alpha,\Omega)$ is a function that returns the minimal distance between a point α and a set Ω (employing 2-norm):

$$\begin{aligned} dist(\alpha,\Omega) = \ & \min_{\alpha,\beta} \|\alpha - \beta\|_2 \\ & s.t.\ \beta \in \Omega \end{aligned} \tag{20}$$

If Ω is a convex polygon, $dist(\alpha,\Omega)$ can be evaluated by using a quadratic programming solver.

Algorithm 2: Determine the closest set to the initial position

```
$d \leftarrow \infty$
for $i = 1 \rightarrow N_{ts}$ do
    if $d \ge dist(C_r x_0, SetList(i))$ then
        $d \leftarrow dist(C_r x_0, SetList(i))$
        $ClosestSet \leftarrow i$
    end if
end for
$OrdList(1) \leftarrow SetList(ClosestSet)$
$SetList \leftarrow SetList \backslash SetList(ClosestSet)$
```

□

In the second algorithm (Algorithm 3), $N_{elem}(\Diamond)$ returns the number of target sets in the list $\Diamond$ and $dist(\Gamma,\Omega)$ is a function that returns the minimal distance (employing the 2-norm) between two sets Γ and Ω (notice that the use of $dist(\cdot,\cdot)$ in the previous algorithm is a particular case in which the point α is the only element of the set Γ).

$$\begin{aligned} dist(\Gamma,\Omega) = \ & \min_{\alpha,\beta} \|\alpha - \beta\|_2 \\ & s.t.\ \alpha \in \Gamma,\ \beta \in \Omega \end{aligned} \tag{21}$$

If Γ and Ω are convex polygons, $dist(\Gamma,\Omega)$ can be again evaluated by using a quadratic programming solver.

Figure 2 shows two examples of the use of $dist(\cdot,\cdot)$ to calculate the minimal distance between a point and a set (a) and between two sets (b).

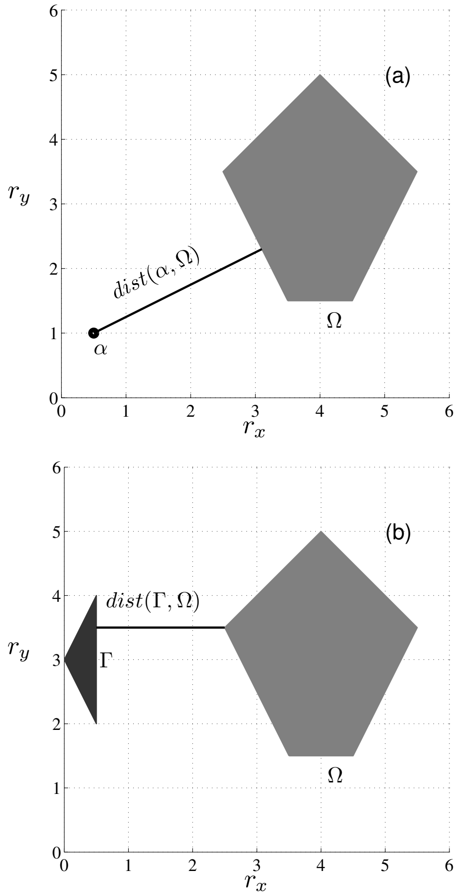

Fig. 2. Illustration of function $dist(\cdot,\cdot)$ involving (a) a point α and a set Ω and (b) two sets Γ and Ω.

Algorithm 3: Order the sets from the first to the last, based on smallest distance between them

```
$N_o \leftarrow 1$
while $N_{elem}(SetList) \neq 0$ do
    $d \leftarrow \infty$
    for $i = 1 \to N_{elem}(SetList)$ do
        if $d \geq dist(OrdList(N_o),SetList(i))$ then
            $d \leftarrow dist(OrdList(N_o),SetList(i))$
            $ClosestSet \leftarrow i$
        end if
    end for
    $OrdList(N_o + 1) \leftarrow SetList(ClosestSet)$
    $SetList \leftarrow SetList \backslash SetList(ClosestSet)$
    $N_o \leftarrow N_o + 1$
end while
```

□

By employing Algorithms 2 and 3, the result is an ordered list of the target sets $OrdList$ containing the set which is the closest to the initial position of the vehicle as its first element and the set that is the closest to the i-th set as the $(i+1)$-th element, for $i = 1,\ldots,N_{ts} - 1$. This list may be used to provide the terminal target set for the trajectory planner formulation presented in [4]. An algorithm similar to Algorithm 1 may then be used to update the ordered list of sets to be visited ($OrdList$).

V. Results

This Section contains two subsections: the simulation scenario is described in Subsection V-A and the simulation results are presented and discussed in Subsection V-B.

A. Simulation Scenario

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

$$\dot{r}_x = v_x,\ \dot{v}_x = a_x,\ \dot{r}_y = v_y,\ \dot{v}_y = a_y \tag{22}$$

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

$$x = [r_x \quad v_x \quad r_y \quad v_y]^T, \quad u = [a_x \quad a_y]^T \tag{23}$$

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

$$A = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0.5T^2 & 0 \\ T & 0 \\ 0 & 0.5T^2 \\ 0 & T \end{bmatrix} \tag{24}$$

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 were $-1 \leq x_2,\, x_4 \leq 1$. As for the accelerations, the imposed limits were $-5 \leq u_1,\, u_2 \leq 5$.

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 obstacle region was represented as a rectangle with $0.6 \leq x_1, x_3 \leq 1$. 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 [8], which involves incorporating restrictions on the transition of the vehicle to each region of the space defined by obstacle inequalities. However, it entails the introduction of more binary variables, increasing the complexity of the MILP problem. As an alternative, 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 obstacle. Therefore, the adopted avoidance constraints were constructed based on the following expanded obstacle: $0.5 \leq x_1, x_3 \leq 1.1$.

The weight γ of the fuel in the cost function was set to 0.1. The maximal horizon was set to $\bar{N} = 35$. Two missions were simulated, each containing $N_{ts} = 3$ rectangular target sets. The inequalities describing the target sets are presented in Table I.

TABLE I
TARGET SETS OF THE TWO SIMULATED EXAMPLES

Target set	Example	
	1	2
1	$0.2 \leq x_1 \leq 0.3$ $0.9 \leq x_3 \leq 1.0$	$1.2 \leq x_1 \leq 1.3$ $0.8 \leq x_3 \leq 0.9$
2	$0.5 \leq x_1 \leq 0.6$ $0.2 \leq x_3 \leq 0.3$	$0.8 \leq x_1 \leq 0.9$ $1.7 \leq x_3 \leq 1.8$
3	$1.2 \leq x_1 \leq 1.3$ $0.9 \leq x_3 \leq 1.0$	$0.2 \leq x_1 \leq 0.3$ $0.7 \leq x_3 \leq 0.8$

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. The MPT toolbox [9] was employed to evaluate the $dist(\cdot,\cdot)$ function in Algorithms 2 and 3.

B. Simulation Results

Figure 3 shows the trajectory in the horizontal plane obtained by employing the multitask planning technique with the target sets of Example 1 in Table I. It can be seen that all targets were visited once and the obstacle region was not crossed. It is worth remarking that the expanded obstacle region is crossed, but not the original obstacle, in agreement with the policy of expanding the obstacle in order to avoid collisions. From Table I, it is possible to note that the first visited target was target set number 2, the second was number 1 and the last, number 3. This shows that the order in which the targets are visited does not depend on the order that they are informed to the planner, because it chooses the visiting order in terms of the solution which provides the minimum cost as a compromise between the overall mission accomplishment time and the fuel spent. Moreover, as shown in Fig. 4, the accelerations a_x and a_y (which correspond to the controls u_1 and u_2, respectively) remained within the ±5 bounds.

The target sets were reached at times: $k = 7$, $k = 13$ and $k = 23$. The fuel cost was 62.50 and the overall cost, 29.25.

For comparison, Fig. 5 shows the trajectory in the horizontal plane generated by the minimum-distance trajectory planning algorithm described in Section IV. It can be seen that the minimum-distance choice for ordering the sets to be visited resulted in the same order as that of the multitask receding horizon planning and control shown in Fig. 3. However, the vehicle is now required to make a sharper turn after reaching the first target set and passes closer to the obstacle as compared to the first case, which is a result of the fact that the optimization is not global in this case, since the presence of a second target set is only informed for the planner/controller at the time the vehicle reaches the first target set. Again, as shown in Fig. 6, the accelerations a_x and a_y remained within the ±5 bounds.

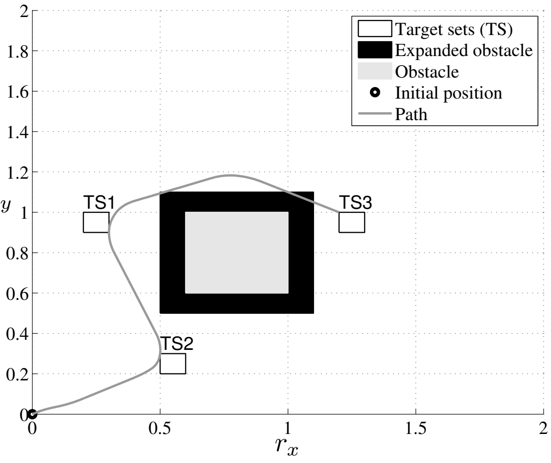

Fig. 3. Trajectory resulting from the use of the proposed multitask planning and control for Example 1.

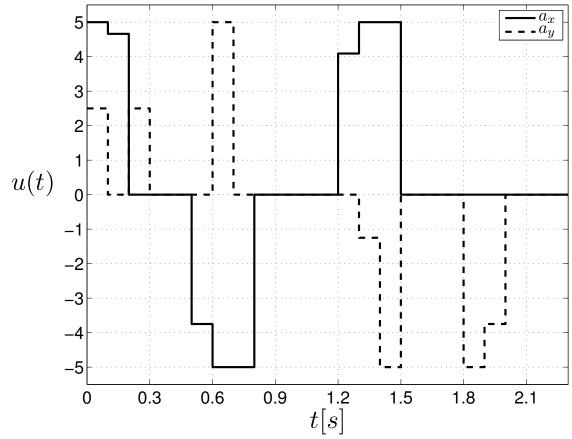

Fig. 4. Control action resulting from the use of the proposed multitask planning and control for Example 1.

The target sets were reached at times $k = 6$, $k = 14$ and $k = 26$. As can be seen, the overall time to accomplish the mission (26 sampling times) was three sampling times larger than the one observed with the multitask planning and control technique (23 sampling times). The fuel cost was 65.47 and the overall cost was 32.55, both larger than the corresponding values obtained with the multitask planning.

Figure 7 shows the trajectory in the horizontal plane obtained by using the multitask planning technique in Example 2. Again, the obstacle was not crossed and all targets were visited once. The first visited target was target set number 3, the second was number 2 and the last, number 1. It can be seen in Fig. 8 that the accelerations a_x and a_y remained within

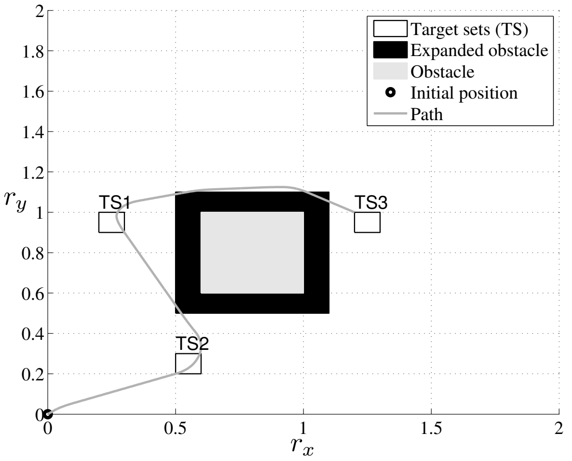

Fig. 5. Trajectory resulting from the use of the minimum-distance target ordering for Example 1.

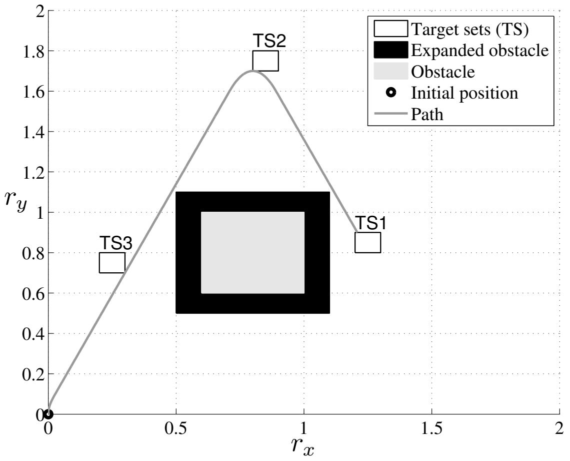

Fig. 7. Trajectory resulting from the use of the proposed multitask planning and control for Example 2.

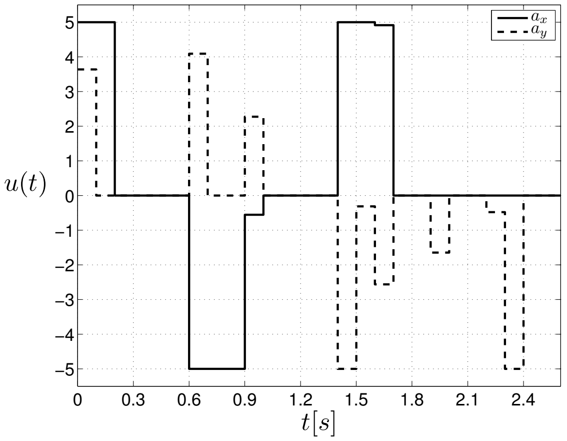

Fig. 6. Control action resulting from the use of the minimum-distance target ordering for Example 1.

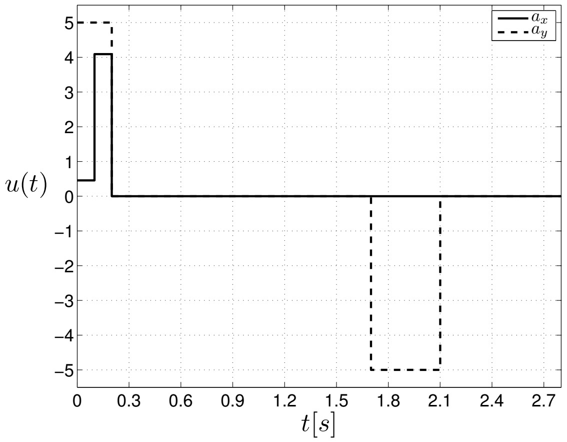

Fig. 8. Control action resulting from the use of the proposed multitask planning and control for Example 2.

the ±5 bounds.

The target sets were reached at times $k = 8$, $k = 19$ and $k = 28$. The fuel cost was 34.55, which is smaller than the corresponding value obtained in Example 1. This can be explained by the fact that the resulting trajectory in this case is composed of two straight lines connected by a curve. The straight lines are more economic regarding the sum of the absolute values of the control signals, since no increment is necessary in these signals in order to travel along straight lines. The overall cost was 31.46, larger than the one obtained in Example 1 due to the larger time necessary to accomplish the maneuver in this case.

For comparison, Fig. 9 depicts the trajectory in the horizontal plane resulting of the employment of the minimum-distance trajectory planning algorithm described in Section IV. It can be seen that the minimum-distance choice for ordering the sets to be visited resulted in a different order as that of the multitask receding horizon planning and control shown in Fig. 7. The first visited target was target set number 3, the second was number 1 and the last, number 2. Figure 10 shows the accelerations a_x and a_y, which once again remained within the bounds ± 5.

The target sets were reached at times $k = 8$, $k = 19$ and $k = 31$ sampling times, which gives an overall time to accomplish the mission three sampling times larger than the one observed with the multitask receding horizon planning and control. The fuel cost was 73.26 and the overall cost was 38.33, both larger than the ones obtained with the multitask planning. It is interesting to note that the trajectory in this case involves more curves than the one in Fig. 7, and even a reversion in the direction after reaching the target set number 1. This explains the larger fuel cost obtained with the minimum-distance order algorithm (73.26) as compared to the one obtained with the multitask algorithm (34.55).

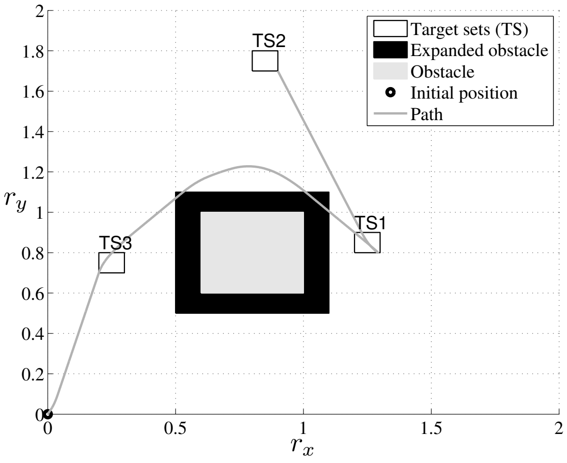

Fig. 9. Trajectory resulting from the use of the minimum-distance target ordering for Example 2.

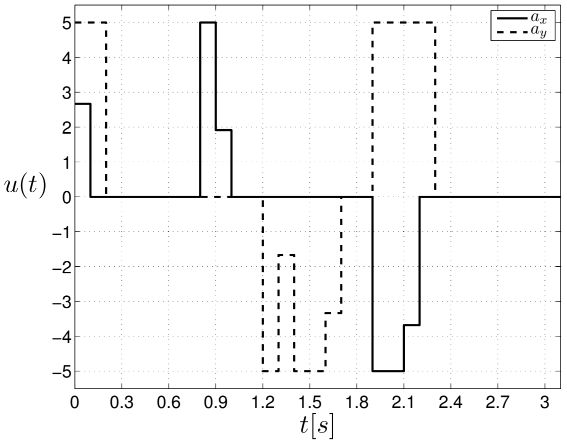

Fig. 10. Control action resulting from the use of the minimum-distance target ordering for Example 2.

VI. Conclusions

This work proposed a novel formulation of a Predictive Control framework which builds upon the one presented in [4] in order to include multiple target sets in the optimization problem. In the context of autonomous vehicles, this is important due to the fact that some missions require the vehicle to visit a number of target sets.

Simulations were employed to validate the proposed technique. Another approach that first orders the list of sets to be visited based on a minimum-distance criterion and then uses each of them as a terminal target set in the formulation presented in [4] was used for comparison. The approach proposed in the present work outperformed the last one in terms of overall cost of the resulting trajectory.

Future works could include robustness regarding an unknown but limited disturbance to this multitask framework by using an adequate constraint tightening approach, as in [4].

Acknowledgments

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

References

[1] 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.

[2] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[3] J. M. Maciejowski, *Predictive Control with Constraints*. Harlow, England: Prentice Hall, 2002.

[4] 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.

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

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

[7] 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.

[8] 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.

[9] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-Parametric Toolbox (MPT)," 2004. [Online]. Available: http://control.ee.ethz.ch/ mpt/