Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

Receding Horizon Path Planning with Implicit Safety Guarantees

Tom Schouwenaars, Jonathan How, and Eric Feron

***Abstract*— This paper extends a recently developed approach to optimal path planning of autonomous vehicles, based on mixed integer linear programming (MILP), to account for safety. We consider the case of a single vehicle navigating through a cluttered environment which is only known within a certain detection radius around the vehicle. A receding horizon strategy is presented with hard terminal constraints that guarantee feasibility of the MILP problem at all future time steps. The trajectory computed at each iteration is constrained to end in a so called *basis* state, in which the vehicle can safely remain for an indefinite period of time. The principle is applied to the case of a UAV with limited turn rate and minimum speed requirements, for which safety conditions are derived in the form of loiter circles. The latter need not be known ahead of time and are implicitly computed online. An example scenario is presented that illustrates the necessity of these safety constraints when the knowledge of the environment is limited and/or hard real-time restrictions are given.**

I. INTRODUCTION

In recent years, both military and civilian institutions have expressed increased interest in the use of fully autonomous aircraft or so called Unmanned Aerial Vehicles (UAV's). Such systems need no, or minor, human control from a ground station, thereby reducing operating costs and enabling missions in harsh or remote environments. A significant part of the vehicle autonomy consists of its path planning capabilities: the problem is to guide the vehicle through an obstacle field, while accounting for its dynamic and kinematic properties.

In many applications, a detailed map of the environment is not available ahead of time, and obstacles are detected while the mission is carried out. In this paper, we consider scenarios where the environment is only known within a certain detection radius around the vehicle. We assume that within that region, the environment is static and fully characterized. The knowledge of the environment could either be gathered through the detection capabilities of the vehicle itself, or result from cooperation with another, more sophisticated agent [11].

An approach to optimal path planning based on Mixed Integer Linear Programming (MILP) was recently introduced in [9]. MILP is a powerful mathematical programming framework that extends continuous linear programming to include binary or integer decision variables [3]. These variables can be used to model logical constraints such as obstacle and collision avoidance rules, while the dynamic and kinematic properties of the vehicle are formulated as continuous constraints. As demonstrated by the results in [7] and [11], thanks to the increase in computer speed and implementation of powerful state-of-the-art algorithms in software packages such as CPLEX [1], MILP has become a feasible option for real-time path planning.

Since in the scenarios of interest the environment is explored online, a trajectory from a starting to a destination location typically needs to be computed gradually over time while the mission unfolds. This calls for a receding horizon strategy, in which a new segment of the total path is computed at each time step by solving a MILP over a limited horizon. Such a strategy was proposed in [9], and extended to account for local minima in [2]. In the latter, a cost-to-go function was introduced based on a graph representation of the whole environment between start and end point, that guaranteed stability in the sense of reaching the goal. In this paper, however, we assume that the environment is *not* fully characterized before the mission. As such, we are interested in guaranteeing *safety* of the trajectory rather than *completion* of the mission, which might be infeasible.

As was shown in our previous work [8], despite the hard anti-collision constraints, the receding horizon strategy of [2] and [9] has no safety guarantees regarding avoidance of obstacles in the future. Namely, the algorithm may fail to provide a solution in future time steps due to obstacles that are located beyond the surveillance and planning radius of the vehicle. This translates into the MILP problem becoming infeasible at a certain time step, and indicates that the vehicle is on a collision course. In [8], we proposed a safe receding horizon scheme based on computing two separate paths at each time step: a locally optimal solution without safety guarantees and a feasible "rescue" path to a naturally collision-free state. The latter is executed whenever the MILP problem associated with the optimal path becomes infeasible, thus avoiding future collisions.

That previous work, however, only considered vehicles that can come to a full stop, such as rovers or helicopters. In this paper, we extend the problem to aircraft constrained by a minimum velocity and a limited turn rate. Moreover, we include the safety constraints (*i.e.* future feasibility guarantees) directly in the optimization problem without the need to precompute an invariant set explicitly. In this respect, our work is fundamentally different from the approaches to safety/feasibility presented in [4], [5] and [10].

The paper is organized as follows. Section II briefly recapitulates the MILP formulation for path planning of a single vehicle, and gives an example scenario in which a nominal receding horizon strategy fails to avoid an obstacle.

The authors are with the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology, Cambridge, MA 02139, USA.
T. Schouwenaars (toms@mit.edu) and E. Feron (feron@mit.edu) are Research Assistant and Associate Professor in the Laboratory for Information and Decision Systems. J. How (jhow@mit.edu) is Associate Professor in the Space Systems Laboratory.

Section III introduces additional safety constraints in the MILP, which are specialized to the case of an aircraft in Section IV and applied to the example scenario in Section V. Section VI concludes with topics for future research.

II. OBSTACLE AVOIDANCE USING MILP

A. Basic Formulation

The basic problem discussed in this paper is to guide an autonomous vehicle through an obstacle field while optimizing a certain objective. The latter can be time, fuel or a more sophisticated cost criterion such as to minimize visibility or to maximize the area explored. The vehicle is characterized by a discrete, linear state space model ($\mathbf{A}$,$\mathbf{B}$) in an inertial coordinate frame and additional linear inequalities capturing dynamic and kinematic constraints.

Denote the initial state of the vehicle by $\mathbf{x}_{init}$ and the desired final state by $\mathbf{x}_f$. Consider a planning horizon of T time steps, where the length of the horizon is determined by both the available computational power onboard the vehicle, as well as the distance over which the environment is fully characterized. Let the cost corresponding to the i^{th} time step be defined as a (piecewise) linear expression $\ell_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{x}_f)$, where $\mathbf{u}_i$ is the input vector. Using a terminal (piecewise) linear cost function $f(\mathbf{x}_T, \mathbf{x}_f)$, the optimal path planning problem over T time steps can then be formulated as follows:

$$\min_{\mathbf{x}_i,\mathbf{u}_i} J_T = \sum_{i=0}^{T-1} \ell_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{x}_f) + f(\mathbf{x}_T, \mathbf{x}_f) \tag{1}$$

$$\begin{array}{rrcl} \text{subject to} & \mathbf{x}_{i+1} & = & \mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{u}_i, \ i = 0 \ldots T-1 \\ & \mathbf{x}_0 & = & \mathbf{x}_{init} \\ & \mathbf{x}_i & \in & \mathcal{X} \\ & \mathbf{u}_i & \in & \mathcal{U} \\ & (x_i, y_i) & \in & \mathcal{D} \\ & (x_i, y_i) & \notin & \mathcal{O}, \end{array} \tag{2}$$

where (x_i, y_i) denotes the position of the vehicle in the plane, the set $\mathcal{D}$ represents the current detection region, and the set $\mathcal{O}$ characterizes the obstacles that are located within the known environment. The sets $\mathcal{X}$ and $\mathcal{U}$ represent the dynamic and kinematic constraints of the vehicle, such as maximum turn rate and minimum speed restrictions.

The obstacle avoidance constraints $(x_i, y_i) \notin \mathcal{O}$ can be explicitly formulated as follows [9]. Assume for simplicity of exposition a rectangular obstacle in 2D with lower left corner $(x_{\min}, y_{\min})$ and upper right corner $(x_{\max}, y_{\max})$. Then, for a point-mass vehicle to avoid the obstacle, each trajectory point (x_i, y_i) must satisfy the following set of constraints:

$$\begin{array}{rcrcl} x_i & \leq & x_{\min} & + & Mb_{i1} \\ -x_i & \leq & -x_{\max} & + & Mb_{i2} \\ y_i & \leq & y_{\min} & + & Mb_{i3} \\ -y_i & \leq & -y_{\max} & + & Mb_{i4} \\ \sum_{k=1}^{4} b_{ik} & \leq & 3 & & \\ b_{ik} & \in & \{0, 1\}. & & \end{array} \tag{3}$$

Here, b_{ik} are binary variables and M is a sufficiently large positive number. The last constraint ensures that at least one of the position constraints is active, thereby guaranteeing that the trajectory point (x_i, y_i) lies outside the rectangle. This set of constraints should be formulated for each time step $i = 0 \ldots T$ and for each obstacle in $\mathcal{O}$. Note that this method can be extended to a 3D environment and to arbitrarily shaped obstacles, which are then approximated by a polygon or polyhedron.

The objective function (1) combined with the (linear) constraints (2) and (3) constitutes a mixed integer linear program (MILP) that must be solved at each time step.

B. Aircraft Example

For an aircraft modeled as a double integrator, the constraints $\mathbf{x}_i \in \mathcal{X}$ and $\mathbf{u}_i \in \mathcal{U}$ can be formulated as follows. As discussed in [6], a limit on maximum speed and available turn rate can be expressed by the following set of linear inequalities corresponding to the edges of a K-sided polygon:

$$\begin{array}{rcl} \multicolumn{3}{l}{\forall i \in [0...T-1], \, \forall k \in [1...K]:} \\ v_{xi} \sin\left(\frac{2\pi k}{K}\right) + v_{yi} \cos\left(\frac{2\pi k}{K}\right) & \leq & v_{\max} \\ a_{xi} \sin\left(\frac{2\pi k}{K}\right) + a_{yi} \cos\left(\frac{2\pi k}{K}\right) & \leq & a_{\max}. \end{array} \tag{4}$$

Here (v_{xi}, v_{yi}) and (a_{xi}, a_{yi}) represent the inertial speed and acceleration vector at the i^{th} time step. For a maximum turn rate $\omega_{\max}$, the corresponding maximum acceleration is determined as $a_{\max} = \omega_{\max} v_{\max}$.

By also introducing a minimum speed $v_{\min}$, the constrained double integrator model captures the aircraft dynamics reasonably well. The minimum speed requirement can be handled by ensuring that the speed vector lies *outside* a K-sided polygon by introducing binary variables as follows:

$$\begin{array}{rcl} \multicolumn{3}{l}{\forall i \in [0...T-1], \, \forall k \in [1...K]:} \\ v_{xi} \sin\left(\frac{2\pi k}{K}\right) + v_{yi} \cos\left(\frac{2\pi k}{K}\right) & \geq & v_{\min} - Mc_{ik} \\ \sum_{k=1}^{K} c_{ik} & \leq & K - 1 \\ c_{ik} & \in & \{0, 1\}, \end{array} \tag{5}$$

where M is again a sufficiently large number.

As an illustration of the above, consider the following example of an imaginary autonomous aircraft with the following parameters: $v_{\max} = 4m/s$, $v_{\min} = 2m/s$ and $\omega_{\max} = 30deg/s$. Assume that the detection radius is $30m$, and that the planning horizon T contains 6 time steps of $1s$ each. The scenario is illustrated in Fig. 1: the aircraft is initially in the origin, flying East at $4m/s$, and needs to maneuver to position $(70m, 57m)$. The cost function is adopted from [8] and aims at proceeding towards the goal, while minimizing the applied thrust:

$$\min_{\mathbf{p}_i,\mathbf{u}_i} J_T = \sum_{i=0}^{T-1} \left(\mathbf{q}'|\mathbf{p}_i - \mathbf{p}_f| + \mathbf{r}'|\mathbf{u}_i| \right) + \mathbf{s}'|\mathbf{p}_T - \mathbf{p}_f| \tag{6}$$

Here $\mathbf{p}_i$ denotes the position (x_i, y_i) of the aircraft, and $\mathbf{q}, \mathbf{r}$ and $\mathbf{s}$ are appropriate weighting vectors.

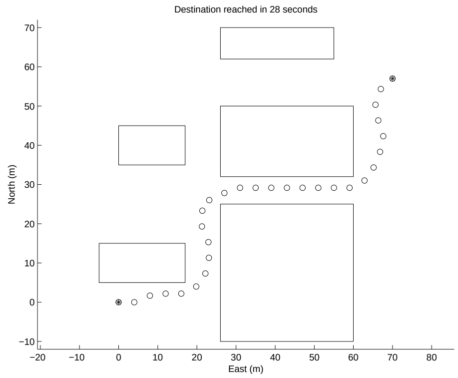

Fig. 1. The aircraft is initially in the origin, flying East at $4m/s$, and has to maneuver to position $(70m, 57m)$. The goal is reached after $28s$.

Consider now the case in which the corridor through which the aircraft is flying, is obstructed as depicted in Fig. 2. Because of its $30m$ detection radius and corresponding limited knowledge of the obstacle field, the MILP optimization guides the vehicle into the concavity, from which it cannot exit. This is due to its limitation on turn rate and minimum speed, resulting in a minimum turn radius that is larger than the available maneuver space. This observation translates into the MILP becoming infeasible after $17s$ and a crash of the aircraft against an obstacle.

It is clear that in such scenarios, the given receding horizon formulation fails. In what follows, we formulate additional constraints that ensure feasibility of the MILP at the next time step, by explicitly solving for a trajectory that ends in a safe state. As such, safety can be guaranteed at each time step. One could claim that by choosing a more sophisticated cost-to-go function that assigns a higher cost to unsafe regions in the state space, problems as the one described above could be avoided automatically. However, our assumption is that nothing is known beyond the detection radius of the vehicle, such that a cost-to-go construction as the one used in [2] can only be applied locally, without guarantees for the future.

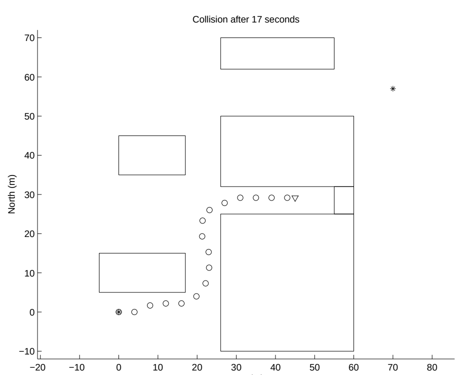

Fig. 2. The aircraft is initially in the origin, flying East at $4m/s$, and has to maneuver to position $(70m, 57m)$. After $17s$, the MILP becomes infeasible, corresponding to the aircraft colliding with the obstacles.

III. SAFETY CONSTRAINTS

To guarantee that the trajectory computed at each time step is safe in the future, we extend the receding horizon problem (1)-(2) to account for terminal safety constraints. To capture the notion of *safety*, we define a *basis state* of the vehicle as a state or motion (*i.e.* a sequence of states) in which the vehicle can remain for an indefinite period of time without violating any of the dynamic, kinematic, detection region or obstacle avoidance constraints. For a helicopter, for instance, the basis state can be defined as hover at any obstacle-free position within the current detection radius. For an airplane, the basis state can be a loiter pattern at a predefined speed. We also define a *safe state* as a state from which there exists a feasible path to a basis state over the length of the planning horizon.

Safety is then ensured at each time step if the receding horizon trajectory terminates in such a basis state, or in other words, if each state along the trajectory is a safe state. As such, when the optimization problem is feasible at the initial time step and assuming that there are no perturbations acting on the vehicle, the MILP will remain feasible at all future steps. Namely, at any given time step other than the initial one, the rest of the trajectory computed at the previous step, augmented with an extra time step in the basis state, is always a feasible solution to the optimization problem defined at that given step. Indeed, since it ends in a basis state in which the vehicle can remain indefinitely, that augmented trajectory is feasible.

More formally, define the set of safe states as $\mathcal{S} \subset (\mathcal{X} \cap \mathcal{O}^c)$ and the set of basis states as $\mathcal{B} \subset \mathcal{S}$, where $\mathcal{X}$ indicates the set of feasible states defined by the kino-dynamic constraints and $\mathcal{O}^c$ denotes the obstacle free states. We now want to ensure that the final state $\mathbf{x}_f$ of the receding horizon trajectory lies in $\mathcal{B}$, or thus that all states $\mathbf{x}_i$ along the trajectory lie in $\mathcal{S}$.

One can add a degree of flexibility to the problem by dividing the planning horizon T in an *optimization* part T_1 and a *feasibility check* part $T_2 = T - T_1$ as follows:

$$\min_{\mathbf{x}_i,\mathbf{u}_i} J_{T_1} = \sum_{i=0}^{T_1-1} \ell_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{x}_f) + f(\mathbf{x}_{T_1}, \mathbf{x}_f) \tag{7}$$

subject to:

$$\left\{ \begin{array}{rcl} \mathbf{x}_{i+1} & = & \mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{u}_i, \; i = 0 \ldots T_1 - 1 \\ \mathbf{x}_0 & = & \mathbf{x}_{init} \\ \mathbf{x}_i & \in & \mathcal{X} \\ \mathbf{u}_i & \in & \mathcal{U} \\ (x_i, y_i) & \in & \mathcal{D} \\ (x_i, y_i) & \notin & \mathcal{O} \end{array} \right. \tag{8}$$

$$\left\{ \begin{array}{rcl} \tilde{\mathbf{x}}_{j+1} & = & \mathbf{A}\tilde{\mathbf{x}}_j + \mathbf{B}\tilde{\mathbf{u}}_j, \; j = T_1 \ldots (T-1) \\ \tilde{\mathbf{x}}_{T_1} & = & \mathbf{x}_{T_1} \\ \tilde{\mathbf{x}}_j & \in & \tilde{\mathcal{X}} \supset \mathcal{X} \\ \tilde{\mathbf{u}}_j & \in & \tilde{\mathcal{U}} \supset \mathcal{U} \\ (\tilde{x}_j, \tilde{y}_j) & \in & \mathcal{D} \\ (\tilde{x}_j, \tilde{y}_j) & \notin & \mathcal{O} \\ \tilde{\mathbf{x}}_T & \in & \mathcal{B} \end{array} \right. \tag{9}$$

The trajectory is thus only optimized over the first T_1 time steps, but is constrained by the feasibility of the remaining T_2 steps and the requirement to terminate in a basis state. This division allows for savings in computation time, without giving up feasibility (*i.e.* safety) guarantees. Moreover, since the terminal cost term in the objective function is defined as a function of $\mathbf{x}_{T_1}$, the optimal trajectory can end in a rather agile state from where there still exists a dynamically feasible path to a basis state. Constraining $\mathbf{x}_{T_1} \in \mathcal{B}$ could restrict the aggressiveness of the trajectories. Moreover, by allowing for more aggressive control in the feasibility check, as expressed by the constraint sets $\tilde{\mathcal{X}} \supset \mathcal{X}$ and $\tilde{\mathcal{U}} \supset \mathcal{U}$, the feasible region of $\mathbf{x}_{T_1}$ is enlarged. For instance, considering hover as the basis state for a helicopter, the maximum feasible speed in $\mathbf{x}_{T_1}$ is larger if a more aggressive braking sequence $\tilde{\mathbf{u}}_j \in \tilde{\mathcal{U}}$ $(j = T_1 \ldots T-1)$ is allowed to bring the helicopter back to hover.

Strictly speaking, for the MILP problem (7)-(9) to remain *feasible* at all future time steps, the constraint sets $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{U}}$ must be subsets of $\mathcal{X}$ and $\mathcal{U}$ rather than supersets. However – still assuming that no disturbances acting on the vehicle– , there will always be a dynamically executable, though possibly aggressive, safe trajectory "*available*" that is the remaining part of the trajectory computed at the previous time step. To avoid this discrepancy between *feasibility* and "*availability*", we will assume in the remainder of this paper that $\tilde{\mathcal{X}} = \mathcal{X}$ and $\tilde{\mathcal{U}} = \mathcal{U}$.

IV. SAFETY CIRCLES

In this section, we specialize the safe receding horizon formulation (7)-(9) to the case of an aircraft constrained by the kino-dynamic inequalities (4)-(5). Because of the minimum speed constraint, a natural basis state $\mathcal{B}$ is a circular loiter pattern. If we can ensure that the trajectory computed at each time step ends in either a left or right turning loiter circle that does not intersect any of the obstacles, the aircraft is guaranteed to be safe at each time step.

A. Linear Loiter Expressions

Assuming that the aircraft behaves like a double integrator, the radius R of the smallest possible loiter circle at a given speed v corresponds to $R = cv^2$. Here c is an aircraft specific parameter associated with the maximum applied side force. As depicted in Fig. 3, to describe the loiter circle as a function of the last state $\mathbf{x}_T = (x_T, y_T, v_{xT}, v_{yT})^T$ in the planning horizon, we need to find the vectors $(\mathbf{p}_T - \mathbf{p}_R)$ and $(\mathbf{p}_T - \mathbf{p}_L)$. Here, $\mathbf{p}_T = (x_T, y_T)^T$ denotes the ingress position of the loiter, and $\mathbf{p}_R$ and $\mathbf{p}_L$ are the center points of the right and left circle respectively. The latter are scaled versions $\alpha \mathbf{v}_T^{\perp}$ and $-\alpha \mathbf{v}_T^{\perp}$ of the orthogonal complement $\mathbf{v}_T^{\perp} = (-v_{yT}, v_{xT})^T$ of $\mathbf{v}_T = (v_{xT}, v_{yT})^T$.

The scaling factor α has a lower and upper bound, corresponding to the minimum and maximum allowed ingress velocity. With $||\mathbf{v}_T||$ the magnitude of $\mathbf{v}_T$, we have $\alpha = \frac{R}{||\mathbf{v}_T||} = c||\mathbf{v}_T||$ and thus $\alpha_{\min} = cv_{\min} \leq \alpha \leq \alpha_{\max} = cv_{\max}$. However, to avoid quadratic constraints in $(v_{xT}, v_{yT})^T$, we use a constant scaling factor α_c. By conservatively setting $\alpha_c = \alpha_{\max}$, the radius of the loiter circles will be larger than necessary for ingress velocities lower than $v_{\max}$, which corresponds to not applying the maximum available lateral thrust. Since this is an over-approximation of a safety condition, however, we are only giving up some performance rather than safety.

By introducing a rotation matrix $\mathbf{R}(\theta)$, any point $\mathbf{p}_{R,\theta}$ along the right loiter circle $\mathcal{C}_R(\mathbf{x}_T)$ can then be expressed as:

$$\begin{aligned} \mathbf{p}_{R,\theta} &= \mathbf{p}_R + \mathbf{R}(\theta)(\alpha_c \mathbf{v}_T^{\perp}) \\ &= (\mathbf{p}_T - \alpha_c \mathbf{v}_T^{\perp}) + \mathbf{R}(\theta)(\alpha_c \mathbf{v}_T^{\perp}) \\ &= \mathbf{p}_T + \alpha_c(\mathbf{R}(\theta) - \mathbf{I})\mathbf{v}_T^{\perp} \end{aligned} \tag{10}$$

Similarly, any point $\mathbf{p}_{L,\theta}$ along the left loiter circle $\mathcal{C}_L(\mathbf{x}_T)$ is given by:

$$\begin{aligned} \mathbf{p}_{L,\theta} &= \mathbf{p}_L - \mathbf{R}(\theta)(\alpha_c \mathbf{v}_T^{\perp}) \\ &= (\mathbf{p}_T + \alpha_c \mathbf{v}_T^{\perp}) - \mathbf{R}(\theta)(\alpha_c \mathbf{v}_T^{\perp}) \\ &= \mathbf{p}_T - \alpha_c(\mathbf{R}(\theta) - \mathbf{I})\mathbf{v}_T^{\perp} \end{aligned} \tag{11}$$

B. Sampling Points Requirements

Maintaining safety now comes down to ensuring that either the left or right loiter circle does not overlap with any of the obstacles that are located within the detection radius of the vehicle. This can be achieved by sampling both circles for fixed values of θ and introducing avoidance constraints similar to (3). However, ensuring obstacle avoidance for sample points along the circle does not guarantee that the loiter circle does not intersect obstacles in the segments between the sample points.

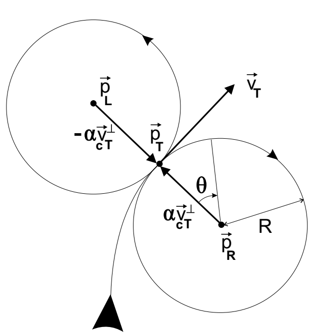

Fig. 3. Safe trajectory ending in either a right or left turning loiter circle.

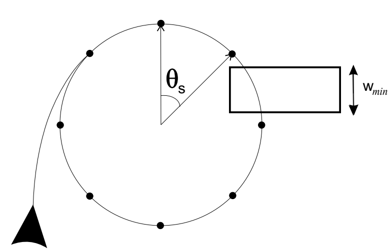

Fig. 4. Situation where undersampling of the loiter circle leads to a safety violation. Although the obstacle avoidance constraints for the sample points are satisfied, the circle intersects the obstacle.

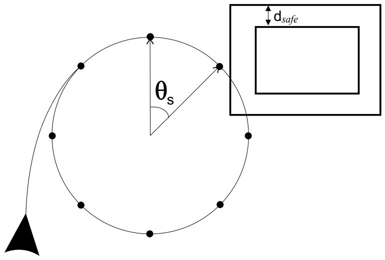

Fig. 5. Situation where the loiter circle cuts the corner of an obstacle. This situation can be avoided by enlarging the obstacles with a safety boundary d_{safe}.

Consider for example the situation depicted in Fig. 4: although the avoidance constraints for all sample points are satisfied, the circle cuts through the obstacle because the sample angle spacing is too coarse. This type of undersampling can be avoided by choosing a minimum number of sampling points N as follows:

$$N \geq N_{\min} = \frac{\pi}{\arcsin(\frac{w_{\min}}{2r_{\max}})} \tag{12}$$

Here $w_{\min}$ denotes the width of the narrowest obstacle, and $r_{\max} = cv_{\max}^2$ is the radius of the largest loiter circle. The derivation of this condition is based on the insight that the maximum spacing in distance between the sample points along the largest circle should not exceed $w_{\min}$. As such, (12) is only necessary when $w_{\min} \leq 2r_{\max}$.

If $w_{\min} > 2r_{\max}$, a situation like the one in Fig. 4 cannot occur, and therefore no minimum number of sample points is required. However, as illustrated in Fig. 5, the loiter circles can now cut the corners of obstacles. Nevertheless, safety can still be guaranteed by enlarging the obstacles with a safety boundary d_{safe} such that the circle can enter the boundary, but does not intersect the actual obstacle. Using basic geometry, one can derive the following expression for d_{safe} as a function of N:

$$d_{\text{safe}}(N) = \frac{\sqrt{2}}{2} r_{\max} \left(1 + \sin\frac{\pi}{N} - \cos\frac{\pi}{N}\right) < \sqrt{2} r_{\max}$$

The enlargement principle also holds for the "cruise" part of the trajectory: due to the time discretization with step Δt, each obstacle must be enlarged by $d_{\text{safe}} = \frac{v_{\max}\Delta t}{\sqrt{2}}$.

C. MILP Formulation

Using the sampling approach from above, we can specify the safety constraint $\mathbf{x}_T \in \mathcal{B}$ from (9) as the following loiter conditions:

$$\left\{\begin{array}{lcl} \mathcal{C}_R(\mathbf{x}_T) = \{(x_{Rj}, y_{Rj})\} & \in & \mathcal{D},\ j = 1 \ldots N \\ \mathcal{C}_R(\mathbf{x}_T) = \{(x_{Rj}, y_{Rj})\} & \notin & \mathcal{O} \\ & \text{OR} & \\ \mathcal{C}_L(\mathbf{x}_T) = \{(x_{Lj}, y_{Lj})\} & \in & \mathcal{D},\ j = 1 \ldots N \\ \mathcal{C}_L(\mathbf{x}_T) = \{(x_{Lj}, y_{Lj})\} & \notin & \mathcal{O} \end{array}\right. \tag{13}$$

where the index j indicates the sample point on the circle. By introducing a binary variable d that selects either the right or left circle, and using (10)-(11) for the coordinates of the sample points, the obstacle avoidance constraints in (13) can be explicitly written as follows:

$\forall l \in [l \ldots L],\ \forall j \in [1 \ldots N]$:

$$\left\{\begin{array}{r} x_T - \alpha_c(\cos j\theta_s - 1)\, v_{yT} - \alpha_c(\sin j\theta_s)\, v_{xT} \\ \leq \quad x_{\min} + Mb_{lj1} + Md \\ -x_T + \alpha_c(\cos j\theta_s - 1)\, v_{yT} + \alpha_c(\sin j\theta_s)\, v_{xT} \\ \leq -x_{\max} + Mb_{lj2} + Md \\ y_T - \alpha_c(\sin j\theta_s)\, v_{yT} + \alpha_c(\cos j\theta_s - 1)\, v_{xT} \\ \leq \quad y_{\min} + Mb_{lj3} + Md \\ -y_T + \alpha_c(\sin j\theta_s)\, v_{yT} - \alpha_c(\cos j\theta_s - 1)\, v_{xT} \\ \leq -y_{\max} + Mb_{lj4} + Md \end{array}\right.$$

AND

$$\left\{\begin{array}{r} x_T + \alpha_c(\cos j\theta_s - 1)\, v_{yT} + \alpha_c(\sin j\theta_s)\, v_{xT} \\ \leq \quad x_{\min} + Mb_{lj1} + M(1-d) \\ -x_T - \alpha_c(\cos j\theta_s - 1)\, v_{yT} - \alpha_c(\sin j\theta_s)\, v_{xT} \\ \leq -x_{\max} + Mb_{lj2} + M(1-d) \\ y_T + \alpha_c(\sin j\theta_s)\, v_{yT} - \alpha_c(\cos j\theta_s - 1)\, v_{xT} \\ \leq \quad y_{\min} + Mb_{lj3} + M(1-d) \\ -y_T - \alpha_c(\sin j\theta_s)\, v_{yT} + \alpha_c(\cos j\theta_s - 1)\, v_{xT} \\ \leq -y_{\max} + Mb_{lj4} + M(1-d) \end{array}\right.$$

$$\left\{\begin{array}{lcl} \sum_{k=1}^{4} b_{ljk} & \leq & 3 \\ b_{ljk},\ d & \in & \{0, 1\} \end{array}\right.$$

Here the index l indicates the (rectangular) obstacles, and $\theta_s = \frac{2\pi}{N}$ is the spacing angle. Note that the obstacle coordinates $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ are those of the obstacles enlarged with d_{safe}.

V. EXAMPLE

We now apply the safe receding horizon formulation to the example of Section II. The planning horizon contains $T = 6$ time steps, of which $T_1 = 3$ are used to optimize, and $T_2 = 3$ are used for the feasibility check. For the loiter circles, we used $N = 8$ sample points. The result is shown in Fig. 6. Thanks to the loiter constraints, the UAV does not fly into the concavity, but chooses an alternative route to reach the goal. As a result, the MILP remains feasible at all time steps, and a collision in the concavity is prevented.

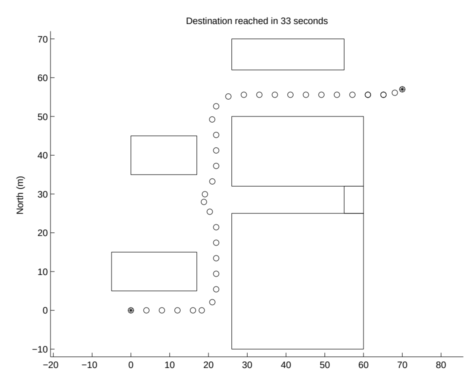

Fig. 6. Receding horizon trajectory with safety constraints. Because of the loiter constraints, the UAV avoids the concavity and chooses an alternative route to reach the goal.

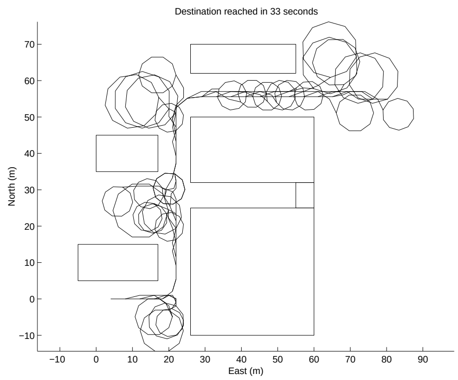

Fig. 7. Sequence of intermediate receding horizon trajectories ending in loiter circles.

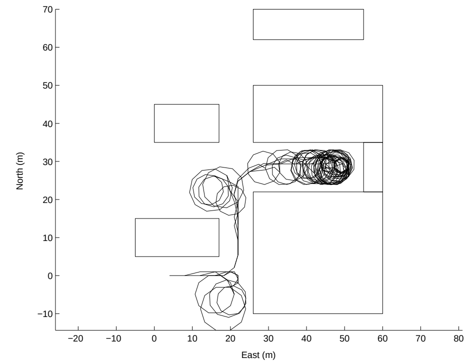

Fig. 8. Sequence of intermediate receding horizon trajectories for a wider concavity. Although the UAV enters the concavity, it does not crash and the trajectory ends in a loiter.

The total trajectory time is now $33s$. The sequence of partial trajectories computed at each time step is depicted in Fig. 7.

Assume now that the width of the concavity is such that the UAV can make a $180°$ turn in it. In this case, the aircraft does fly into the concavity, but can avoid the obstacle by executing a loiter pattern as displayed in Fig. 8. Although the mission was not fulfilled, the aircraft remains safe at all times. While loitering, the vehicle can apply some higher level decision logic or more sophisticated cost-to-go function to compute a path out of the concavity.

It is worthwhile mentioning that the trajectories in this paper were computed in quasi real-time, *i.e.* almost all iterations terminated within the time step duration of $1s$. The simulations were done using MATLAB and CPLEX8.1 on a Pentium4 PC with 2GHz clock speed. At the few time steps where the computation did not finish in time, however, a good suboptimal solution was usually still found within $1s$. This highlights the use of the safety constraints from another perspective: if, in a *hard real-time* system, the optimization does not solve in time at a particular step, the remaining part of the solution from the previous time step can be used. Assuming that a safe trajectory exists at the first time step, and that the vehicle can compensate for perturbations along the way, safety will be maintained at all future steps.

VI. CONCLUSION AND CURRENT WORK

A. Conclusion

We considered the problem of navigating an autonomous vehicle through a cluttered environment that is explored online. We formulated a receding horizon strategy for real-time path planning based on MILP with safety and future feasibility guarantees. The partial trajectory planned at each time step was constrained to terminate in a safe basis condition, in which the vehicle can remain indefinitely. An explicit formulation for a UAV was worked out using the concept of obstacle-free loitering circles.

B. Current Work

To demonstrate the applicability of the presented framework to real-time guidance of UAV's, we are currently integrating it with Boeing's Open Control Platform for flight testing on a T-33 aircraft [11]. This urges us to extend the safety formulation to account for robustness against disturbances in the trajectories. Furthermore, we are applying the loiter principle to guarantee *a priori* collision avoidance in decentralized cooperative path planning of multiple autonomous aircraft.

VII. ACKNOWLEDGEMENTS

This research was funded under the DARPA Software Enabled Control Program, AFRL contract # F33615-01-C-1850.

REFERENCES

[1] *ILOG CPLEX User's guide*. ILOG, 2003.

[2] J. Bellingham, A. Richards, and J. How. "Receding Horizon Control of Autonomous Aerial Vehicles". In *Proc. of the 2002 American Control Conference*, Anchorage, AK, May 2002.

[3] C. A. Floudas. *Nonlinear and Mixed-Integer Programming - Fundamentals and Applications*. Oxford University Press, 1995.

[4] E.C. Kerrigan and J.M. Maciejowski. "Invariant Sets for Constrained Nonlinear Discrete-time Systems with Application to Feasibility in Model Predictive Control". In *Proc. of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, December 2000.

[5] M. Oishi, C. Tomlin, V. Gopal, and D. Godbole. "Addressing Multi-objective Control: Safety and Performance through Constrained Optimization". In M.D. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, number 2034 in LNCS, pages 459–472. Springer Verlag, Berlin Heidelberg, 2001.

[6] A. Richards and J. How. "Aircraft Trajectory Planning With Collision Avoidance Using Mixed Integer Linear Programming". In *Proc. of the 2002 American Control Conference*, Anchorage, AK, May 2002.

[7] A. Richards, Y. Kuwata, and J. How. "Experimental Demonstrations of Real-Time MILP Control". In *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Austin, TX, August 2003.

[8] T. Schouwenaars, E. Feron, and J. How. "Safe Receding Horizon Path Planning for Autonomous Vehicles". In *Proc. of the 40th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, October 2002.

[9] T. Schouwenaars, B. De Moor, E. Feron, and J. How. "Mixed Integer Programming for Multi-Vehicle Path Planning". In *Proc. of the 2001 European Control Conference*, Porto, Portugal, September 2001.

[10] C. Tomlin, I. Mitchell, and R. Ghosh. "Safety Verification of Conflict Resolution Maneuvers". *IEEE Transactions on Intelligent Transportation Systems*, 2(2), June 2001.

[11] M. Valenti, T. Schouwenaars, Y. Kuwata, E. Feron, and J. How. "Implementation of a Parent-Child Mission System in a Partially-Known Environment". Providence, RI, August 2004. To appear at the AIAA Guidance, Navigation and Control Conference.