Proceedings of the 2006 American Control Conference
Minneapolis, Minnesota, USA, June 14-16, 2006

Multi-Vehicle Path Planning for Non-Line of Sight Communication

Tom Schouwenaars, Eric Feron, and Jonathan How

Abstract— **This paper presents the formulation and hardware results of a mixed-integer linear programming approach to online connectivity-constrained trajectory planning of autonomous helicopters through cluttered environments. A lead vehicle must execute a certain mission whereby wireless line of sight communication to its ground station is lost. Relay helicopters are therefore introduced that must position themselves in such a way that indirect line of sight connectivity between the leader and the ground station is always maintained. The corresponding coordinated multi-vehicle trajectory optimization is tackled using both centralized and distributed receding horizon planning strategies. Binary variables are used to capture connectivity, obstacle- and collision avoidance constraints, extending earlier formulations to model nonconvex obstacles more efficiently. Simulation, hardware in the loop, and flight test results are presented for a centralized two-helicopter mission. Simulation results for a distributed scenario are given as well.**

I. INTRODUCTION

The problem of interest in this paper is to guide a small autonomous helicopter through a cluttered (e.g. urban) environment in which wireless line of sight communication with a ground station cannot be maintained due to the presence of obstacles. Depending on the nature of the mission, however, some form of communication with the vehicle is typically required. Examples include urban surveillance, inspection of disaster sites, traffic observation, search and rescue missions, volcanic crater measurements, fire fighting unit support, etc. Nevertheless, a reliable communication link between the ground station and the helicopter executing the mission can be obtained by introducing a set of relay helicopters. Indirect communication with the ground station can then be established by positioning the relay vehicles in such a way that a sequence of direct line of sight links exists between the ground station and the mission helicopter, henceforth called the leader. As the leader is maneuvering through the environment, the relay helicopters should adapt their positions such that visibility/connectivity between the subsequent relay agents is maintained. A cooperative scheme is therefore required that allows the relay formation to coordinate its reconfiguration with the actions of the leader, and vice versa.

This coordination problem can be formulated as a connectivity-constrained multi-vehicle path planning problem. Besides maintaining line of sight, feasibility of the trajectories implies respecting the helicopters' kino-dynamic constraints, as well as avoiding obstacles and collisions. In this paper, we will focus solely on these dynamic and geometric aspects of the connectivity problem and ignore mobile networking factors such as fading, cross-talk, and delay, which can also affect the availability of links between the vehicles.

Various researchers have approached the problem of motion planning for connected multi-agent systems. Nguyen *et al.* [3] describe the use of mobile relay nodes to extend the effective range of a ground robot exploring a complex interior environment. The relay nodes follow the lead robot in convoy and automatically stop where needed to form an ad hoc network guaranteeing a link between the lead robot and the base station. Variations of this algorithm can be found in Sweeney *et al.* [6]. Beard and McLain [1] use dynamic programming to tackle the problem of searching a region of interest using multiple UAVs under communication range and collision avoidance constraints. Obstacles blocking line of sight between vehicles are not accounted for, however. Finally, Spanos and Murray [5] discuss the feasibility aspects of path planning for vehicles connected through a range-constrained wireless network. They define a connectivity robustness measure which quantifies the freedom of individual vehicles to undergo arbitrary motions without disconnecting the network, but do not explicitly construct feasible trajectories.

In this paper, we extend a trajectory planning approach that is based on *mixed-integer linear programming* (MILP). MILP is a powerful mathematical programming framework that extends linear programming to include binary or integer decision variables. It is commonly used in Operations Research and has more recently been introduced to the field of trajectory optimization [4]. Thanks to the increase in computer speed and implementation of powerful state-of-the-art algorithms in software packages such as CPLEX, MILP has become a feasible option for real-time path planning.

II. PROBLEM FORMULATION

A. Problem Setup

This section presents the connectivity-constrained path planning problem more formally. Let the various agents be denoted by an index $i = 0, \ldots, L$, where $i = 0$ is the stationary ground station, $i = L$ indicates the leader, and the remaining index values correspond to the relay helicopters. For optimization purposes, the dynamics of the moving vehicles are characterized by discrete time, linear state space models $\boldsymbol{x}_i(t+1) = \boldsymbol{A}_i \boldsymbol{x}_i(t) + \boldsymbol{B}_i \boldsymbol{u}_i(t),\ i = 1, \ldots, L$ where $\boldsymbol{x}_i(t) \in \mathbb{R}^{N_x}$ is the state vector and $\boldsymbol{u}_i(t) \in \mathbb{R}^{N_u}$ is the input vector at the t^{th} time step. The state vector $\boldsymbol{x}_i(t)$ is typically made up of the position and velocity in a 3D

The authors are with the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology, Cambridge, MA 02139, USA. This work was funded under ONR Grant N00014-03-1-0171 and AFOSR MURI "Complex Adaptive Networks For Cooperative Control" (contract #009628-001-03-132).

inertial coordinate frame (east, north, altitude), respectively denoted by $\boldsymbol{p}_i(t) \equiv [x_i(t)\ y_i(t)\ z_i(t)]' \in \mathbb{R}^3$ and $\boldsymbol{v}_i(t) \equiv [\dot{x}_i(t)\ \dot{y}_i(t)\ \dot{z}_i(t)]' \in \mathbb{R}^3$. A trajectory will then consist of a sequence of states $\boldsymbol{x}_i(t) \equiv [\boldsymbol{p}_i'(t)\ \boldsymbol{v}_i'(t)]'$ or waypoints that the vehicle must track.

Depending on the particular model, the input vector $\boldsymbol{u}_i(t)$ is a 3D inertial acceleration or reference velocity vector. In both cases, however, combined with additional constraints on $\boldsymbol{x}_i(t)$ and $\boldsymbol{u}_i(t)$, the state space model must describe the closed-loop dynamics that result from augmenting the vehicle with a waypoint tracking controller. These constraints should capture kinematic and dynamic properties such as maximum speed, acceleration and turn rate, and will be denoted as follows: $\boldsymbol{x}_i(t) \in \mathcal{X}_i(t)$ and $\boldsymbol{u}_i(t) \in \mathcal{U}_i(t)$. Note that the constraint sets $\mathcal{X}_i(t)$ and $\mathcal{U}_i(t)$ are time dependent: this accommodates the use of robust planning approaches such as constraint tightening. Accounting for the vehicle dynamics in the trajectory planning problem will ensure that the trajectories are dynamically feasible. Feasibility will further be affected by the presence of obstacles in the environment, such as buildings or hills. We define the set $\mathcal{O}_i \subset \mathbb{R}^3$ of all obstacles and unknown areas that are relevant to vehicle i as the regions in the inertial space that the vehicle is forbidden to enter: $\boldsymbol{p}_i(t) \notin \mathcal{O}_i$.

The overall goal of the trajectory planning problem is then for the leader L to perform a certain task while optimizing an associated cost. The latter can be a measure of time, fuel or a more sophisticated criterion such as visibility w.r.t. a threat or radar. The task will typically be specified as having to fly to a certain waypoint of interest, denoted as the final state $\boldsymbol{x}_f \equiv [\boldsymbol{p}_f'\ \boldsymbol{v}_f']'$. In doing so the leader will further be constrained by the network connectivity requirement and the communication capabilities of the relay helicopters. Denoting the broadcasting range of each agent i as $d_{relay,i} \in \mathbb{R}$, we define this more precisely as follows:

Definition 1 (Line of Sight Connection): *We say that there exists a bidirectional* line of sight connection *between two agents i and j, positioned respectively at $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, if their Euclidean separation distance $d_{ij} = \|\boldsymbol{p}_i - \boldsymbol{p}_j\|$ is smaller than the shortest broadcasting range of the two, i.e. $d_{ij} \leq \min(d_{relay,i}, d_{relay,j})$, and the line between them does not intersect any obstacles, i.e., $\boldsymbol{p}_i + \lambda(\boldsymbol{p}_j - \boldsymbol{p}_i) \notin (\mathcal{O}_i \cap \mathcal{O}_j),\ 0 \leq \lambda \leq 1$.*

In the most general case, the topology of the wireless network could be set up ad hoc as part of the optimization problem. In this paper, however, we consider a fixed order in the set of relay vehicles $i = 1, \ldots, L-1$ that corresponds to a linear graph structure: $i = 1$ relays information between $i = 0$ and $i = 2$, $i = 2$ does so between $i = 1$ and $i = 3$, and so forth. As such, there is no robustness against link dropouts at one of the relay agents. At the cost of a more complex optimization problem (and therefore longer computation time) more flexible network structures with redundancy properties could be obtained.

The connectivity requirement for the ordered team of vehicles can then be defined as follows:

Definition 2 (System Connectivity): *We say that there is* system connectivity *in the ordered set of agents if there exist line of sight connections between all pairs of subsequent agents $(i, i+1),\ i = 0, \ldots, L-1$: $d_{i(i+1)} \leq \min(d_{relay,i}, d_{relay,i+1})$ and $\boldsymbol{p}_i + \lambda(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i) \notin (\mathcal{O}_i \cap \mathcal{O}_{i+1}),\ 0 \leq \lambda \leq 1$.*

Trajectory feasibility for all vehicles now also implies that system connectivity is maintained at all times. Lastly, to avoid collisions, the agents should remain at a safe distance d_{safe} from each other at all times: $d_{ij} = \|\boldsymbol{p}_i - \boldsymbol{p}_j\| \geq d_{safe}$ for all pairs $(i, j),\ i = 0, \ldots, L-1,\ j > i,$.

B. Receding Horizon Planning

Depending on the number of vehicles and the distance the leader has to travel, computing complete trajectories from start till finish at once might be computationally too expensive. Moreover, the environment might only be partially-known and further explored in real-time. The trajectories will therefore have to be computed gradually over time while the mission unfolds. This can be accomplished using an online receding horizon strategy, in which partial trajectories from the current states $\boldsymbol{x}_i(t)$ towards the goal $\boldsymbol{x}_f$ (for $i = L$) or feasible relay positions (for $i = 1, \ldots, L-1$) are computed by solving, at each time step, the trajectory optimization problem over a limited horizon of T time steps [4], [2]. Let the sequence of T steps starting at time t be denoted by indices $(t+k|t)$. For each vehicle $i = 1, \ldots, L$, the corresponding state and control sequence is then given by $\boldsymbol{x}_i(t+k|t),\ k = 0, \ldots, T$ and $\boldsymbol{u}_i(t+k|t),\ k = 0, \ldots, T-1$.

C. Centralized Optimization Problem

The primary objective of the trajectory optimization problem is to guide the leader to the goal state $\boldsymbol{x}_f$ while optimizing a certain performance criterion and maintaining connectivity. Since the actions of the leader must be coordinated with those of the relay helicopters, the most straightforward approach is a centralized one, in which one entity, e.g. the ground station, computes trajectories for all vehicles simultaneously. Introducing an objective function J_T, the connectivity-constrained trajectory optimization problem at time t can then be formulated as:

$$J_T^* = \min \sum_{k=0}^{T-1} \ell_{L,k}(\boldsymbol{x}_L(t+k|t), \boldsymbol{u}_L(t+k|t), \boldsymbol{x}_f)\ +$$

$$f_L(\boldsymbol{x}_L(t+T|t), \boldsymbol{x}_f) + \sum_{i=1}^{L-1} \sum_{k=0}^{T-1} \ell_{i,k}(\boldsymbol{x}_i(t+k|t), \boldsymbol{u}_i(t+k|t))$$

subject to: $\quad \forall i = 1, \ldots, L:$

$$\boldsymbol{x}_i(t+k+1|t) = \boldsymbol{A}_i \boldsymbol{x}_i(t+k|t) + \boldsymbol{B}_i \boldsymbol{u}_i(t+k|t), \quad k = 0, \ldots, T-1 \tag{1}$$

$$\boldsymbol{x}_i(t|t) = \boldsymbol{x}_i(t|t-1) \tag{2}$$

$$\boldsymbol{v}_i(t+T|t) = \boldsymbol{0} \tag{3}$$

$$\boldsymbol{x}_i(t+k|t) \in \mathcal{X}_i(k),\ k = 1, \ldots, T \tag{4}$$

$$\boldsymbol{u}_i(t+k|t) \in \mathcal{U}_i(k),\ k = 0, \ldots, T-1 \tag{5}$$

$$\boldsymbol{p}_i(t+k|t) \notin \mathcal{O}_{a,i}(t),\ k=1,\dots,T \quad (6)$$

$$\boldsymbol{p}_i(t+k|t) + \lambda\left(\boldsymbol{p}_i(t+k|t) - \boldsymbol{p}_{i-1}(t+k|t)\right) \notin \mathcal{O}_{v,i}(t) \cap \mathcal{O}_{v,i-1}(t), 0 \le \lambda \le 1, k=1,\dots,T \quad (7)$$

$$\|\boldsymbol{p}_i(t+k|t) - \boldsymbol{p}_{i-1}(t+k|t)\| \le \min(d_{relay,i}, d_{relay,i-1}),\ k=1,\dots,T \quad (8)$$

$$\|\boldsymbol{p}_i(t+k|t) - \boldsymbol{p}_j(t+k|t)\| \ge d_{safe}, \quad j=0,\ j \ge i+1,\ k=1,\dots,T \quad (9)$$

$$\boldsymbol{p}_0(t+k|t) = \boldsymbol{p}_0(0) \quad (10)$$

where the last equality indicates that the ground station does not move.

In the objective function above, $\ell_{L,k}(\cdot)$ indicates the stage cost associated with the leader at the k^{th} time step, and $f_L(\cdot)$ represents a terminal penalty function. The latter should be an estimate of the cost-to-go from the last state $\boldsymbol{x}_L(t+T|t)$ in the planning horizon to the desired waypoint $\boldsymbol{x}_f$. Similarly, $\ell_{i,k}(\cdot)$ is the k^{th} stage cost corresponding to relay vehicle $i,\ i=1,\dots,L-1$. Notice, however, that there is no cost-to-go function associated with the relay vehicles and that their stage costs $\ell_{i,k}(\cdot)$ do not depend on $\boldsymbol{x}_f$. Indeed, reaching the goal waypoint only matters to the leader, and there are no *a priori* known desired locations for the relay helicopters.

Constraint (3) is a terminal hover constraint ensuring that the optimization problem at the next iteration is feasible. Indeed, the remaining segments $k=2,\dots,T$ of the trajectories produced at time t and combined with an extra time step in the respective hover positions $\boldsymbol{p}_i(t+T|t)$ form a feasible solution to the new optimization problem at $t+1$. Furthermore, to prevent the discrete time trajectories and lines of sight from cutting corners of obstacles in between two time steps, the obstacle sets $\mathcal{O}_{a,i}$ and $\mathcal{O}_{v,i}$ are the actual obstacles enlarged with a safety envelope. The trajectories might instead cut through the envelope, but will avoid the actual obstacles. Lastly, since the problem only makes sense if the initial states $\boldsymbol{x}_i(t|t)$ are feasible, the state constraints on the first time step were removed.

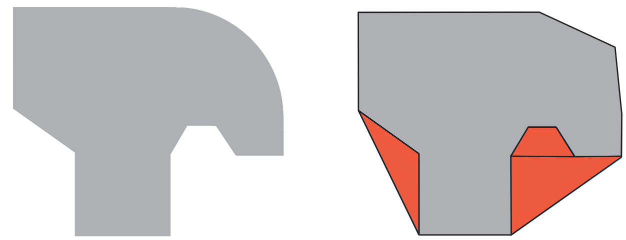
III. MILP Formulation

A. Helicopter Dynamics

Although it is possible to use more complicated models, for the application in this paper it is sufficient to approximate the helicopter dynamics by a double integrator model with limits on planar speed (v_{max}), planar acceleration (a_{max}), climb rate $\dot{z}$ and climb acceleration $\ddot{z}$. For details on the corresponding expressions for constraints (1), (4), and (5) we refer the reader to our earlier work [4] and [2].

B. Obstacle Avoidance

In this paper we extend our earlier obstacle avoidance formulations [4] to arbitrarily shaped obstacles in a compact manner. Let an index o denote the individual obstacles in $\mathcal{O}_{a,i} \subset \mathbb{R}^3$, with $\mathcal{O}_{a,i}$ the set of enlarged obstacles that are within reach of vehicle i at the current planning iteration. Since our objective is to use linear optimization techniques, an arbitrarily shaped obstacle o is first approximated by a



(a) Actual obstacle shape (b) Polytopic approximation

Fig. 1. Approximation of an arbitrary 2D obstacle by its convex hull and with convex polygons filling up the concavities.

(possibly non-convex) polyhedron $\mathcal{P}_{io}$. Next, we construct the convex hull of $\mathcal{P}_{io}$ and denote each of the resulting faces by an index e, $e=1,\dots,H_{io}$. A sufficient condition for obstacle avoidance is then that all points along the planned trajectory of vehicle i lie outside this convex hull, i.e., in at least one of the outer halfspaces determined by the faces e. Let these halfspaces be described by $u_{ioe}x + v_{ioe}y + w_{ioe}z + h_{ioe} \le 0,\ e=1,\dots,H_{io}$. Avoidance of obstacle o by vehicle i can then be expressed by introducing binary variables $b_{ioe}(k) \in \{0,1\},\ e=1,\dots,H_{io}$, as follows: $\forall k=1,\dots,T$:

$$\begin{aligned} u_{io1}x_i(k) + v_{io1}y_i(k) + w_{io1}z_i(k) + h_{io1} &\le Mb_{io1}(k) \\ &\vdots \\ u_{ioH_{io}}x_i(k) + v_{ioH_{io}}y_i(k) + w_{ioH_{io}}z_i(k) &+ h_{ioH_{io}} \\ &\le Mb_{ioH_{io}}(k) \\ \textstyle\sum_{e=1}^{H_{io}} b_{ioe}(k) &\le H_{io}-1 \end{aligned} \quad (11)$$

in which we shortened the time step notation to k. M is an arbitrary constant that is larger than any of the values the left-hand sides of the inequalities can take in the current planning problem. Then, if $b_{ioe}(k)=1$ for a particular e, the corresponding inequality is relaxed and always satisfied. The last constraint ensures that at least one of the binaries $b_{ioe}(k)$ is 0, such that at least one of the halfspace inequalities holds and obstacle avoidance is guaranteed.

If the surrounding polyhedron $\mathcal{P}_{io}$ is non-convex, however, trajectory points can also lie inside concavities (that are reachable from the outside). By partitioning each concavity into convex polyhedral parts c, $c=1,\dots,C_{io}$, an additional binary $\hat{b}_{ioc}(k)$ for each of these parts can be introduced that captures several inequalities at once. Figure 1 shows an example of such a partitioning. Let the faces g and corresponding inner halfspaces of each of the convex parts c be described by $\hat{u}^g_{ioc}x + \hat{v}^g_{ioc}y + \hat{w}^g_{ioc}z + \hat{h}^g_{ioc} \le 0,\ g=1,\dots,G_{ioc}$. The avoidance constraints (11) can then be extended as follows: $\forall k=1,\dots,T$:

$$\begin{aligned} u_{io1}x_i(k) + v_{io1}y_i(k) + w_{io1}z_i(k) + h_{io1} &\le Mb_{io1}(k) \\ \vdots \qquad\qquad &\ \ \vdots \quad \vdots \\ u_{ioH_{io}}x_i(k) + v_{ioH_{io}}y_i(k) + w_{ioH_{io}}z_i(k) + h_{ioH_{io}}(k) & \\ &\le Mb_{ioH_{io}}(k) \end{aligned}$$

$$\begin{cases} \hat{u}^1_{io1} x_i(k) + \hat{v}^1_{io1} y_i(k) + \hat{w}^1_{io1} z_i(k) + \hat{h}^1_{io1} \leq M\hat{b}_{io1}(k) \\ \qquad \vdots \qquad\qquad \vdots \quad \vdots \\ \hat{u}^{G_{io1}}_{io1} x_i(k) + \hat{v}^{G_{io1}}_{io1} y_i(k) + \hat{w}^{G_{io1}}_{io1} z_i(k) + \hat{h}^{G_{io1}}_{io1} \\ \qquad\qquad \leq M\hat{b}_{io1}(k) \end{cases}$$

$$\vdots \qquad \vdots \quad \vdots$$

$$\begin{cases} \hat{u}^1_{ioC_{io}} x_i(k) + \hat{v}^1_{ioC_{io}} y_i(k) + \hat{w}^1_{ioC_{io}} z_i(k) + \hat{h}^1_{ioC_{io}} \\ \qquad\qquad \leq M\hat{b}_{ioC_{io}}(k) \\ \qquad \vdots \qquad\qquad \vdots \quad \vdots \\ \hat{u}^{G_{ioC_{io}}}_{ioC_{io}} x_i(k) + \hat{v}^{G_{ioC_{io}}}_{ioC_{io}} y_i(k) + \hat{w}^{G_{ioC_{io}}}_{ioC_{io}} z_i(k) + \hat{h}^{G_{ioC_{io}}}_{ioC_{io}} \\ \qquad\qquad \leq M\hat{b}_{ioC_{io}}(k) \end{cases}$$

$$\sum_{e=1}^{H_{io}} b_{ioe}(k) + \sum_{c=1}^{C_{io}} \hat{b}_{ioc}(k) \leq H_{io} + C_{io} - 1 \qquad (12)$$

The last inequality now ensures that at time step k vehicle i will either be outside the convex hull or inside one of the concavities of obstacle o.

As mentioned before, the actual obstacles are enlarged with a safety envelope that compensates for the discrete time nature of the trajectories. If this envelope is too large compared to the size of the obstacles, additional avoidance checks can be carried out for linearly interpolated positions between the waypoints along the trajectory. The vehicle coordinates $\boldsymbol{p}_i(k) \equiv [x_i(k)\, y_i(k)\, z_i(k)]'$ in constraints (11)-(12) should then be replaced by:

$$\boldsymbol{p}_i(k-1) + \frac{l}{L_a}\left(\boldsymbol{p}_i(k) - \boldsymbol{p}_i(k-1)\right),\ l = 1, \ldots, L_a \qquad (13)$$

where L_a indicates the number of interpolation points and additional binary variables should be introduced accordingly.

C. Connectivity Constraints

The connectivity constraints (7) can be expressed by substituting the single vehicle coordinates $\boldsymbol{p}_i(k) \equiv [x_i(k)\, y_i(k)\, z_i(k)]'$ in the obstacle avoidance conditions (11)-(12) by the line between subsequent agents $i-1$ and i $(i = 1, \ldots, L)$:

$$\boldsymbol{p}_{i-1}(k) + \frac{l}{L_v}\left(\boldsymbol{p}_i(k) - \boldsymbol{p}_{i-1}(k)\right),\ l = 1 \ldots L_v \qquad (14)$$

where L_v is the number of interpolation points. The latter should be determined in function of how far the vehicles can be separated and how large a safety envelope can be tolerated.

Because they are convex, the maximum separation constraints (8) are easier to formulate and do not require the use of binaries. For simplicity, instead of approximating the 2-norm of the separation distance between vehicles i and $i-1$ $(i = 1, \ldots, L)$ by a collection of tangent planes, we use an outer 1-norm formulation. Constraints (8) then become: $\forall k = 1, \ldots, T$:

$$\begin{array}{rcrcl} x_i(k) & - & x_{i-1}(k) & \leq & \min(d_{relay,i}, d_{relay,i-1}) \\ x_{i-1}(k) & - & x_i(k) & \leq & \min(d_{relay,i}, d_{relay,i-1}) \end{array}$$

and similar for the y- and z-coordinates.

The collision avoidance constraints (9) are the non-convex counterpart of the maximum separation constraints, and do require the use of binary variables. For the detailed expressions, we refer to our earlier work [4].

IV. Implementation

The MILP-based trajectory planning algorithm described in Section III was implemented on a testbed consisting of two autonomous X-Cell miniature helicopters, a ground station and a trajectory planning laptop. The X-Cell's avionics box contains an Aaeon PC-104 computer which runs the QNX real-time operating system. Among other functions, the on-board software features 1) a Kalman filter that combines GPS, IMU, and barometer measurements to estimate the inertial position and velocity, 2) stabilizing control algorithms, 3) a waypoint follower, and 4) health management functions. Waypoints can be uploaded from the Windows-based ground station in real-time, and, through file sharing, the downloaded vehicle states are available to the ground-based trajectory planning laptop. The communication between the ground station and the helicopters is done through Microhard Spectra NT920 wireless modems that were configured for relay operation.

The MILP optimization module was implemented using a combination of commercially available software products. The planning parameters, such as the current state and obstacle information, were processed in Matlab, the optimization model was implemented in AMPL, and CPLEX 9.0 was used as the MILP solver. The module ran on a separate Pentium 4 laptop with 2.2 GHz clock speed, which used standard Windows file sharing with the ground station to obtain state updates from both helicopters (downloaded every second) and to upload waypoints. It was executed every 5 s, resulting in a new waypoint list being uploaded at 5 s intervals. We thereby made use of CPLEX's computation time limit, which was set to 4.7 s. In case no optimal solution was found within that time, the best feasible solution was returned. If no feasible solution was found in time, e.g., because of a structural infeasibility such as having been blown into an obstacle's safety envelope, the previous waypoint list was kept. Since each waypoint plan of the receding horizon strategy terminates in a safe hover state, however, a sequence of such infeasibilities does not jeopardize the helicopters.

V. Results

A. Mission Scenario and Planning Parameters

An office park in Burlington, MA, was chosen as the test side. Using a combination of GPS measurements and satellite image data from Google Earth, we built a Matlab model of the environment, which is shown in Figure 2. There are three large office buildings of about 50 m tall surrounded by parking lots and wooded areas with trees of up to 40 m. The buildings were considered as separate obstacles, whereas the trees were treated as one large no-fly zone. The ground station is located in the origin of the local coordinate frame whose x-, y-, and z-axes are aligned with the east, north, and up directions respectively. The initial states are hover

T	6	d_{relay}	400 m	z_{max}	100 m
Δt	2.5 s	d_{safe}	10 m	$\dot{z}_{min}$	-2 m/s
N	16	v_{max}	2 m/s	$\dot{z}_{max}$	2 m/s
L_a	4	a_{max}	2 m/s^2	$\ddot{z}_{min}$	-2 m/s^2
L_v	10	z_{min}	25 m	$\ddot{z}_{max}$	2 m/s^2

TABLE I
PARAMETERS USED IN THE MILP TRAJECTORY OPTIMIZATION

in respectively $(0, -16, 25)$m for the leads helicopter and $(0, 3, 25)$m for the relay one. The goal, shown as a star, is located at $(176, -206, 25)$m. Unless the lead helicopter is at an altitude that is too high for the type of ISR mission that we envision, flying to the target area clearly involves obscuration of the line of sight between the ground station and the lead helicopter.

Before flight testing the Burlington scenario, we performed Matlab and high-fidelity hardware-in-the-loop (HIL) simulations. In the hardware tests, two identical X-Cell machines or HIL simulators were employed. The parameters used in the trajectory optimization problem are given in Table I. The receding horizon contained $T = 6$ time steps of $\Delta t = 2.5$ s each, corresponding to an actual planning horizon of 15 s. With a maximum velocity of 2 m/s, the 6 waypoints were thus separated by at most 5 m. However, since the X-Cell's waypoint controller provided better tracking results with a 10 m resolution, only the 2nd, 4th and 6th waypoint were uploaded. Furthermore, for safety reasons, a minimum and maximum altitude of respectively $z_{min} = 25$ m and $z_{max} = 100$ m were enforced. The former ensured avoidance of light poles on the terrain and would give the safety pilot enough time to take over in case something went wrong.

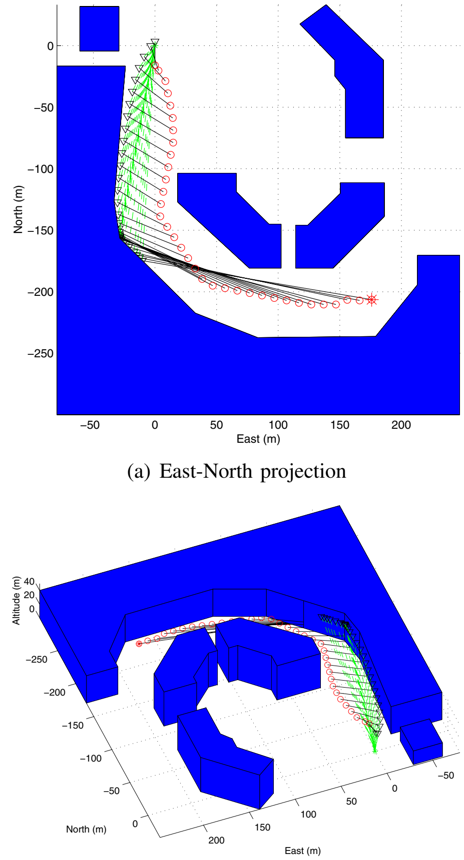
The scenario used the following heuristic cost function:

$$\begin{aligned} J_T &= \sum_{k=1}^{T} 10^{-2}|z_L(k) - z_f| + \sum_{k=0}^{T-1} 10^{-4}|\boldsymbol{a}_L(k)| \\ &+ \sum_{k=1}^{T} \left(10^{-3}|\boldsymbol{v}_R(k)| + 10^{-2}|z_R(k) - z_{min}|\right) \\ &+ \sum_{k=0}^{T-1} 10^{-3}|\boldsymbol{a}_R(k)| + |\boldsymbol{p}_L(T) - \boldsymbol{p}_f| \end{aligned}$$

in which L and R respectively indicate the lead and relay helicopter, and $|\cdot|$ represents the 1-norm. This cost function seeks to guide the lead helicopter to the goal $\boldsymbol{p}_f$ by placing a high weight on the position error $|\boldsymbol{p}_L(T) - \boldsymbol{p}_f|$, while minimizing the effort of the relay helicopter to maintain line of sight. The actions of the latter are expressed in terms of its velocity and acceleration sequence. To prevent the vehicles from flying over the buildings, the cost function also penalizes altitude, as expressed by the $|z_L(k) - z_f|$ and $|z_R(k) - z_{min}|$ terms.

B. Simulation and Flight Test Results

Figure 2 shows a Matlab simulation of the scenario in the nominal case, i.e., without disturbances or model uncertainties. The circles indicate the position of the lead helicopter at every 5 s time step; the triangles represent the relay agent. The goal is reached after 33 receding horizon iterations or 165 s. The line of sight between the lead and relay helicopters at each time step is plotted in dark, the connection between the ground station and the relay in light dashed line. The result shows that as the lead helicopter flies around the buildings to reach its goal, the relay moves south to maintain visibility with both the lead helicopter and the ground station. Connectivity between all agents is thus guaranteed at all times. At every iteration an optimal solution was always found within the time limit of 4.7 s.



(a) East-North projection

(b) 3D plot

Fig. 2. Nominal trajectories of the Burlington scenario computed in Matlab using receding horizon planning. The circles represent the lead helicopter, the triangles show the relay vehicle.

Figure 3 shows the same scenario tested using high fidelity HIL-simulators. It shows the position of the vehicles at a 5 s interval (+/-1 s because of timing inaccuracies in the file sharing). In the HIL-sim test, the MILP algorithm operated on the same platform and with the same data that it would see in an actual flight test. As can be seen from Figure 3, the HIL-sim took 51 receding horizon iterations or 255 s to arrive at the goal instead of 165 s. This was mainly due to the vehicle flying slower than commanded in the straight edge towards the end.

Figure 4 shows the result of an actual flight test. The MILP software was initiated after the helicopters were manually taken off and brought to their starting positions by the waypoint controller. Notice that the first part of the flight looks very similar to the HIL-sim result. About 65% through

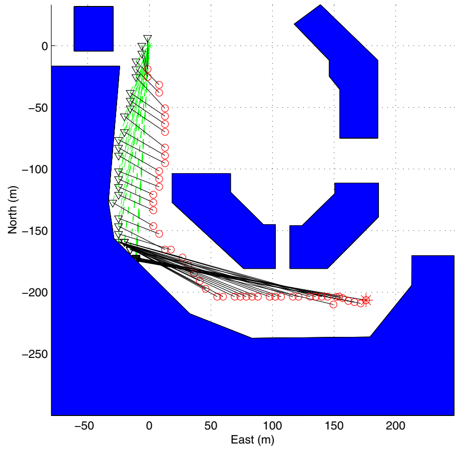


Fig. 3. HIL-simulations of the trajectories computed in real-time.

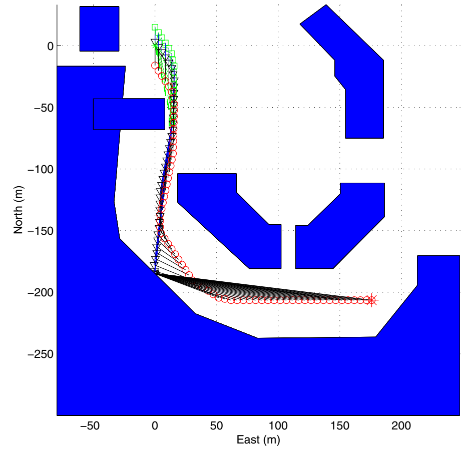


Fig. 5. Simulation of distributed approach with extra obstacle and relay.

the mission, however, a faulty power supply on-board the relay caused the communication with the ground station to be temporarily interrupted. As such, both helicopters ran out of waypoint updates and safely hovered at the last waypoint of their latest plan (approximately at $(-25, -120, 25)$ for the relay and at $(22, -178, 25)$ for the leader). As a safety procedure, the relay helicopter then entered a "safe return home" mode, which consisted of first climbing to an altitude of 100 m and then returning to the position from where it started. However, during this climb out, communication picked up again and the lead helicopter continued receiving and executing the MILP updates. It thereby maintained line of sight with the climbing relay until the mission was manually aborted.

VI. Distributed Algorithm

Because it accounts for the states and inputs of all vehicles as unknowns in the optimization problem, the computation time of the centralized formulation scales exponentially with the number of agents. Our current work therefore focuses on a distributed approach whereby each helicopter only computes its own trajectory. It thereby accounts for the latest plans of the others in a cyclic order corresponding to the relay sequence. For vehicles that just updated their trajectory

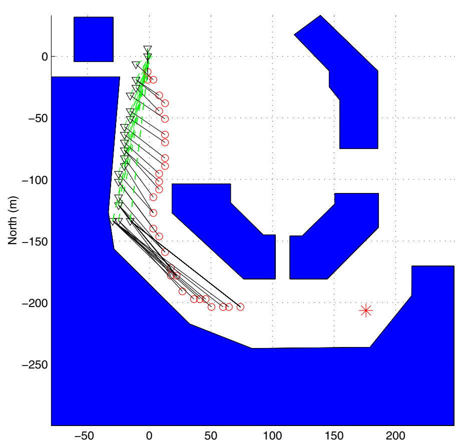


Fig. 4. Actual flight test data of the trajectories computed in real-time.

in the current cycle, the latest plan is the newly optimized trajectory. For vehicles that arrive later in the cycle, the latest plan is the one computed during the previous iteration. The current vehicle thus considers the plans of all others as fixed such that they enter as constraints in the optimization problem and not as unknowns. Feasibility for each vehicle can be guaranteed by terminal hover constraints like Eq. (3). Starting with the leader who optimizes its mission-dependent cost function, a possible objective for each subsequent relay agent is to minimize the distance to the previous vehicle in the cycle. This was done in Figure 5 which shows a second relay helicopter that is introduced to maintain connectivity when an extra obstacle is placed in the field. On average, the total cycle for all vehicles at each iteration took 1.3 s.

VII. Conclusion

We presented a MILP formulation for online connectivity-constrained trajectory planning of autonomous vehicles through cluttered environments. The framework was successfully implemented on two X-Cell helicopters and tested in a real-world scenario. The receding horizon optimization problem, a generalized formulation for obstacle avoidance and maintaining connectivity, and proof-of-concept results were described in detail. The results show that MILP-based path planning can effectively handle obstacle avoidance, collision avoidance, and connectivity constraints in real-time.

References

[1] R. Beard and T. McLain. Multiple uav cooperative search under collision avoidance and limited range communication constraints. In *Proc. 42nd Conf. on Decision and Control*, Maui, HI, December 2003.

[2] J. Bellingham, A. Richards, and J. How. Receding horizon control of autonomous aerial vehicles. In *Proc. 2002 American Control Conference*, pages 3741–3746, Anchorage, AK, May 2002.

[3] H.G. Nguyen, N. Pezeshkian, M. Raymond, A. Gupta, and J.M. Spector. Autonomous communication relays for tactical robots. In *11th Int. Conf. on Advanced Robotics*, Coimbra, Portugal, June 2003.

[4] T. Schouwenaars, B. De Moor, E. Feron, and J. How. Mixed integer programming for multi-vehicle path planning. In *Proc. European Control Conf.*, pages 2603–2608, Porto, Portugal, September 2001.

[5] D. Spanos and R. Murray. Motion planning with wireless network constraints. In *American Control Conf.*, Portland, OR, June 2005.

[6] J. Sweeney, T.J. Brunette, Y. Yang, and R. Grupen. Coordinated teams of reactive mobile platforms. In *Proc. 2002 IEEE Int. Conference on Robotics and Automation*, Washington, D.C., May 2002.