Trajectory Optimization under Kinematical Constraints for Moving Target Search

Manon Raap, Martin Zsifkovits, Stefan Pickl
Universität der Bundeswehr München
Fakultät für Informatik
Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany
Email:{manon.raap, martin.zsifkovits, stefan.pickl}@unibw.de

***Abstract*—Various recent events in the Mediterranean sea have shown the enormous importance of maritime search-and-rescue missions. By reducing the time to find floating victims, the number of casualties can be reduced. A major improvement can be achieved by employing unmanned aerial systems for autonomous search missions. In this context, the need for efficient search trajectory planning methods arises. Existing approaches either consider K-step-lookahead optimization without accounting for kinematics of fixed-wing platforms or propose a suboptimal myopic method. A few approaches consider both aspects, however only applicable to stationary target search. The contribution of this article consists of a novel method for Markovian target search-trajectory optimization. This is a unified method for fixed-wing and rotary-wing platforms, taking kinematical constraints into account. It can be classified as K-step-lookahead planning method, which allows for anticipation to the estimated future position and motion of the target. The method consists of a mixed integer linear program that optimizes the cumulative probability of detection. We show the applicability and effectiveness in computational experiments for three types of moving targets: diffusing, conditionally deterministic, and Markovian. This approach is the first K-step-lookahead method for *Markovian* target search under *kinematical* constraints.**

I. Introduction

Recent events have shown the enormous importance of search-and-rescue missions. In the spring of 2015, over 1.300 lifes were lost at sea in a single month after refugee ships sunk in the Mediterranean sea [1]. Operational decisions for a search mission by a fleet of aircraft are made by an assigned coordinator of a Maritime Rescue Coordination Center (MRCC). He allocates the search effort by assigning searchers to distinct subareas. This task is already supported by systems based on search theoretical approaches. E.g. the search and rescue optimal planning system [2], which is currently used by the United States Coast Guard. Nevertheless, individual pilots are expected to plan their optimal trajectory by hand, which is tremendously complex; it is proven to be an $\mathcal{NP}$-complete optimization problem by [3] for a single platform searching for a single stationary target. Life-rafts at sea are, moreover, likely to drift due to wind and current. In this case, it is not sufficient just to fully cover the search area, because it is possible that the life-raft drifts into an already observed area. Planning for *moving* target search is considerably more complex in general, and moreover, the kinematical constraints of the aircraft must also be taken into account. Pilots must be ready for take off within the prescribed time to preparedness, which is maximal 30 minutes by international agreement. Executing such a complex task in a stressful situation is susceptible to resulting in a sub-optimal search-trajectory and rescue may come too late. We therefore aim to automatize this task with the outlook towards autonomous search-missions by unmanned aerial vehicles (UAVs). The method presented in this article is applicable for aerial sensor platforms in general, i.e. platforms that are either fixed-winged or rotary-winged, either manned or unmanned and either autonomous or non-autonomous. We refer to an aerial sensor platform by *platform* for short in the remainder of this article. For a general introduction to the method we refer to [4].

The contribution of this article consists of a novel model for Markovian target search-trajectory optimization. This is a unified model for fixed-wing and rotary-wing platforms, taking kinematical constraints into account. It can be classified as K-step-lookahead planning method, which allows for anticipation to the estimated future position and motion of the target. The individual considerations of these two aspects can be found in the related works that we discuss in the literature review. However, these approaches are inherently incompatible. Therefore, a completely new model is required to incorporate both aspects, which we present in the work at hand. Another novel aspect of our approach is the heterogeneous state space for the target and platform, whereas all published K-step-lookahead methods so far consider both to move on a homogeneous grid. This concept is shown in Figure 1 and holds two benefits; a target specific grid allows for a more accurate estimation of the target position, whereas a platform specific grid allows for modeling more natural flight kinematics.

The remainder of this article is structured as follows: Section II provides a review of the related literature. In Section III, the search-trajectory problem is stated, followed by our method for solving this problem in Section IV. Simulations in Section V show the applicability and effectiveness of this method. Finally, the conclusions are presented in Section VI.

II. Related Literature

Research initiated by Koopman [5] has led to the interesting field of *Search Theory*. The standard reference *Theory of Optimal Search* [6] was awarded the Lanchester Prize and continues in [7].

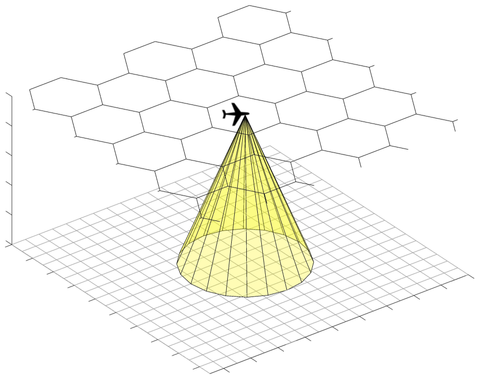


Fig. 1: The heterogeneous state spaces for the target (square grid) and the platform (hexagonal grid).

A problem that is closely related to the problem addressed in the work at hand is the path-constrained search effort allocation (SEA) problem [8]. Here, the search area is typically divided into subareas to which search effort is allocated over time. The search effort is expressed in a number of platforms of a certain type and a duration. For solving the SEA problem, the following exact algorithms have been proposed. A depth-first branch and bound approach was presented in [8], in which lower bound approximations on the probability of non-detection are obtained by relaxing the searcher's path constraints. Most following approaches are also of the branch and bound type [9], [10], [11], [12], where the aim is to find a tightest true lower bound with low computational costs. Computational experiences of several branch and bound procedures are summarized in [13]. Despite vast progression over the years, computation remains intractable for larger instances. This has led to the development of heuristic approaches such as a receding horizon approach [14], cross entropy optimization [15] and constraint programming [16]. This SEA problem is, however, less suitable for autonomous search because the trajectories of the platforms within a subarea are not provided by the solution to this problem. For autonomous search, also the kinematics of the aerial vehicle must be taken into account. The first approach that explicitly considers these aerial vehicle kinematics is presented by Bourgault *et al*. [17], [18]. They showed the effectiveness of a myopic control implementation by simulations. Search-path optimization is clearly a well studied problem in literature. Nevertheless, an import step towards employing fixed-wing platforms for moving target search must be made, since no K-step-lookahead method under *kinematical* constraints has been published so far. To this aim, we propose a novel mixed integer linear program (MILP).

III. The Search-Trajectory Problem

This section describes the search-trajectory problem as introduced in [19]. It consists of the following aspects: the probability map for the target position, the target model, the sensor model, the platform model and, finally, the search objective. We describe these aspects in the subsections below.

A. Probability Map

We consider the search for a moving target in discrete time on the search area $\mathbb{O} \subset \mathbb{R}^2$. The time allocated to a planning stage is defined by a sequence $\mathcal{K} = (1, \ldots, K)$ of K time steps. A grid based probability map partitions the search area uniformly into a finite set of cells $\mathcal{C}$. The target occupies one unknown cell $C_k \in \mathcal{C}$ at time $k \in \mathcal{K}$. For the duration of the search mission, a probability map $\boldsymbol{pc}_k$ is maintained for each k, where the probability of containment $pc_{k,c}$ represents the probability of the target occupying cell c at time k, without being detected prior to time k. Although the initial position of the target is unknown, it is characterized by a known initial probability distribution $\boldsymbol{pc}_1$.

B. Target Model

The target trajectory is modeled by a stochastic process $(C_0, ..., C_K)$, which is assumed to be Markovian [20]. The probability map evolves due to the target motion according to

$$pc_{k+1,c} = \sum_{c' \in \mathcal{C}} d_{c',c}\, pc_{k,c'}, \tag{1}$$

where the transition function $d_{c',c} \in [0, 1]$ represents the probability that the target moves from cell c' to cell c and is assumed to be known for each cell pair c', c.

C. Sensor Model

We assume that the considered aerial sensor platform has a stabilized sensor equipped to make observations. Two results are defined for an observation $Z_{k,c}$ on cell c at time k:

$$Z_{k,c} := \begin{cases} 1, & \text{if the platform detects the target in cell } c \\ & \text{at time } k, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The glimpse probability $pg_{k,o,c}$ represents the probability of target detection, given target occupancy within cell c and platform position $o_k \in \mathbb{O}$ at time k, i.e.

$$pg_{k,o,c} := P(Z_{k,c} = 1 \,|\, C_k = c, o_k). \tag{3}$$

Let us emphasize that the glimpse probability can be positive and variant for multiple cells at once as a result of the heterogeneous state spaces for targets and searchers, as visualized in Figure 1.

When observations are made, the probability map evolves according to the motion model as in Equation (1) and, in addition, evolves according to the observation results as in Equation (2) by the glimpse probability as in Equation (3). Therefore, Equation (1) is extended to account for observation results as follows:

$$pc_{k+1,c} = B \sum_{c' \in \mathcal{C}} d_{c',c}\, pc_{k,c'} \left(1 - pg_{k,o,c'}\right), \tag{4}$$

where the normalization coefficient B is given by

$$B = \left(\sum_{c \in \mathcal{C}} pc_{k,c} \left(1 - pg_{k,o,c}\right) \right)^{-1}. \tag{5}$$

D. Platform Model

The object to control in this problem is an aerial platform. Its motion model, adopted from [19], is given by

$$\begin{aligned} x_{k+1} &= x_k + s_k \cdot \cos(\theta_k + \alpha_k) \\ y_{k+1} &= y_k + s_k \cdot \sin(\theta_k + \alpha_k) \\ \theta_{k+1} &= \theta_k + \alpha_k, \end{aligned} \tag{6}$$

where parameter s_k is the speed of the platform, parameter θ_k is its heading angle and α_k is its change of heading at time k, which are obviously restricted by the laws of physics. Variables x_k and y_k represent the coordinates of the platform on the plane above the search area, i.e. position $o_k = [x_k, y_k] \in \mathbb{O}$.

E. Objective

The objective is to determine a search-trajectory $\boldsymbol{o} = (o_1, ..., o_K)$ maximizing the cumulative probability of detection over time period $\mathcal{K}$, i.e.

$$\max_{\boldsymbol{o}} \sum_{k=1}^{K} \sum_{c \in \mathcal{C}} pd_{k,o,c}, \tag{7}$$

where $pd_{k,o,c}$ is the probability of detecting the target at time k in cell c and is calculated by

$$pd_{k,o,c} = pc_{k,c}\, pg_{k,o,c}. \tag{8}$$

The probability of containment $pc_{k,c}$ is calculated through Equation (4), but with $B = 1$. The normalization from Equation (5) is omitted, so that the probability map is not normalized. Consequently, the probability of containment $pc_{k,c}$ does not represent an actual probability anymore, since it does not sum to unity over the grid cells. It does, however, sum to the probability that the target has not been found up until time k despite the search effort. Therefore, the objective function in Equation (7) yields the cumulative probability of detection over time period $\mathcal{K}$.

IV. METHOD

In this section, we introduce our K-step-lookahead planning method for search-trajectory optimization. First, in Subsection IV-A, we use the platform model to construct a discretized and finite platform state space. We then construct a reduced graph in Subsection IV-B. In Subsection IV-C, we define the search-trajectory problem on the reduced graph. Finally, we formulate this problem as mixed integer linear program in Subsection IV-D.

A. Platform State Space

The platform motion model in (6) results in an infinite and continuous platform state space. We discretize this state space in the following for efficiency purposes. First, we assume a constant speed s in meters per second and, second, we limit the change of heading α to a predefined set of options. For fixed-wing platforms we chose $\{-\gamma, 0, \gamma\}$ and for rotary-wing platforms we chose $\{-2\gamma, -\gamma, 0, \gamma, 2\gamma, 3\gamma\}$. Here, we use $\gamma = \frac{\pi}{3}$ (in radians), such that the resulting platform state space, referred to by $\mathcal{V}'$, is discretized and finite. The Voronoi diagram induced by this platform state space forms a hexagonal grid. Fig. 2 and 3 show a segment of this grid, supplemented by the three fixed-wing (resp. seven rotary-wing) platform options to observe in the next time step. Euclidean distance $l = \|o_k - o_{k+1}\|$ represents the covered distance in one time step, when moving with speed s and the minimum radius of a turn as well. It can be derived from the rate one turn (ROT) and the search speed s. The ROT is a standard rate used to express the time needed for a fixed-wing platform to make a 360 degree turn, which is two minutes. Using basic geometry, it is straightforward to derive the following formula for the radius of a turn l in meters: $l = \frac{60s}{\pi}$.

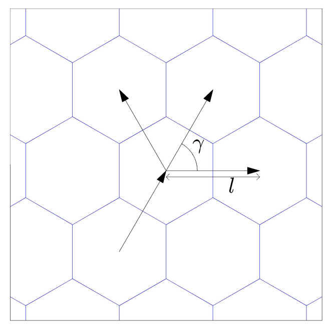


Fig. 2: Three fixed-wing platform control options on the hexagonal grid.

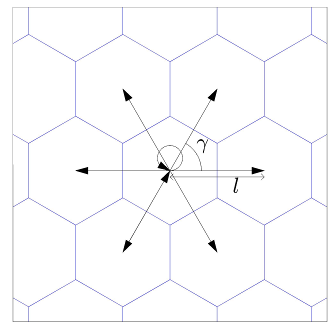


Fig. 3: Seven rotary-wing platform control options on the hexagonal grid.

B. Reduced Graph

The complexity of the search trajectory problem increases exponentially with the number of nodes, so a reduced graph is beneficial to decrease computational costs. We therefore aim to reduce the graph, such that it exclusively contains a fixed number of nodes which are most likely to be contained in the optimal search-trajectory, i.e. the set of connected nodes that yields the highest potential probability of detection. Selecting this subset is equivalent to the maximum weight connected subgraph problem [21], however, a greedy heuristic can be used to approximate this subgraph. The resulting reduced graph $G = (\mathcal{V}, A, R)$ is used for search-trajectory planning and is defined by its nodes $\mathcal{V}$, adjacency matrix A, and reachability matrix R. These elements are described next.

Adjacency Matrix: The binary adjacency matrix A holds information on the adjacency of the nodes. An entry $a_{v,v'}$ is 1 if node v is adjacent to node v' and 0 otherwise. The parameter $\widetilde{a}_{v,v'}$ is its negation and is mentioned here, because it will be used in the linear programming formulation of the search-trajectory problem.

Reachability Matrix: The binary reachability matrix R is used to direct the platform towards a node on the reduced graph. First, recall $o_t \in \mathbb{O}$ to be the platform position at time t and let $s_{reloc} \geq s$ be the relocation speed used to cover a relocation arc. Furthermore, let k' be the minimal number of time steps that the platforms requires to reach any node in $\mathcal{V}$. The goal for the current planning stage is to optimize the search-trajectory over the K time steps $\{t+k', t+k'+1, \ldots, t+k'+K-1\}$. Relocation arcs are added between o_t and each node in $\mathcal{V}$ for each time $k \in \mathcal{K}$ by which it can be reached. The set of relocation arcs is represented by the binary reachability matrix R of size $|\mathcal{V}| \times K$, where entry $r_{k,v}$ is 1 if node v is reachable in $k'+k$ time steps and 0 otherwise. The parameter $\widetilde{r}_{k,v}$ is its negation. Note that the planning horizon recedes by $k'+K$ time steps per planning stage.

This solution has three major benefits. First, it significantly reduces one of the dimensions of the model and thereby its complexity. Second, a drifting uncertainty area can be intercepted by a linear flight approach, due to the ability of anticipation of the predicted target movements. The advantage of this aspect, when compared with a myopic one-step-lookahead method, is emphasized in a simulation shown in Fig. 4. Third, this overcomes a typical problem faced by approaches with a static lookahead horizon of K_{static}; if all feasible trajectories of length K_{static} yield zero reward, no decision on the preceding flight trajectory can be made. By using these relocation arcs, there is at least one feasible search-trajectory yielding positive reward.

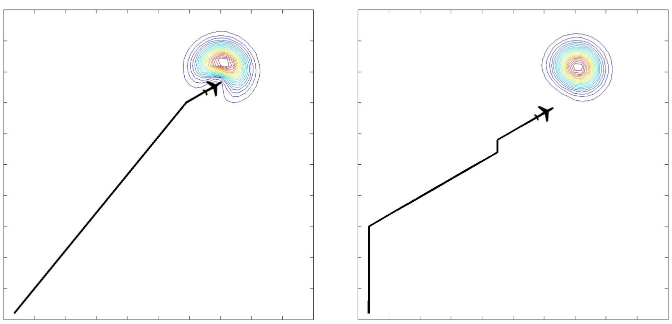


Fig. 4: The K-step-lookahead aspect of the proposed method (a) allows for anticipation to the target movement and finds a direct (shorter) approach towards the east moving target. On the other side, the myopic method used in (b) acts greedy, resulting in a suboptimal detour. As a result, the platform in (a) has a higher probability of detecting the target in K time steps. This aspect intensifies as the approach distance increases.

C. The Search-Trajectory Problem on the Reduced Graph

The search-trajectory problem on the reduced graph G is formulated in the following. It consists of finding a physically feasible trajectory on graph G, such that the cumulative probability of detection is maximized. For a simple graph (which does not have multiple arcs), a trajectory may be specified completely by a sequence of nodes [22]. Formally:

Definition 1 (Trajectory): A *trajectory* is a sequence of nodes $(v_k)_{k\in\mathcal{K}}$, where consecutive nodes in the sequence are adjacent nodes in the graph, i.e. node v_{k+1} is in set $\mathcal{V}(v_k)$, for all $k \in \mathcal{K}$.

The physical feasibility of a trajectory is inherent for a rotary-wing platform. However, a fixed-wing platform can not hover, make sharp turns, or fly backwards. Therefore, for the latter platform type, additional constraints are required for a trajectory to be physically feasible.

Definition 2 (Physically Feasible Trajectory): A trajectory $(v_k)_{k\in\mathcal{K}}$ is *physically feasible* for fixed-wing platforms if and only if node v_{k+1} is in set $\mathcal{V}(v_k) \setminus \mathcal{V}(v_{k-1})$ for all $k \in \mathcal{K}$.

The sets $\mathcal{V}(v_k)$ and $\mathcal{V}(v_k) \setminus \mathcal{V}(v_{k-1})$ are shown schematically in Fig. 5 and Fig. 6 respectively.

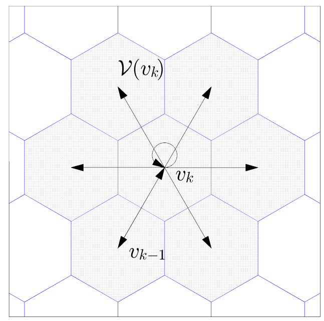


Fig. 5: Kinematical constraints on the trajectory for rotary-wing platforms. The nodes in set $\mathcal{V}(v_k)$ are adjacent to node v_k. A trajectory (v_{k-1}, v_k, v_{k+1}) is physically feasible for a rotary-wing platform if and only if node $v_{k+1} \in \mathcal{V}(v_k)$.

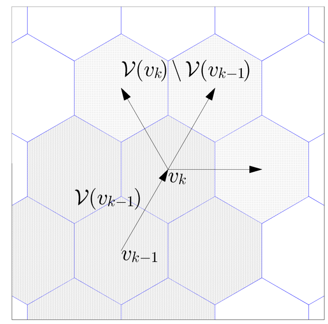


Fig. 6: Kinematical constraints on the trajectory for fixed-wing platforms. A trajectory (v_{k-1}, v_k, v_{k+1}) is physically feasible for a fixed-wing platform if and only if node $v_{k+1} \in \mathcal{V}(v_k) \setminus \mathcal{V}(v_{k-1})$.

D. Mixed Integer Linear Programming Formulation

The search-trajectory problem on the reduced graph can now be formulated as a mixed integer linear program (MILP). Solving the MILP yields the physically feasible search-trajectory that maximizes the cumulative probability of detection.

Decision Variables: Three types of decision variables are necessary. Let $\mathbb{B} = \{0, 1\}$. The decision variable $z_{k,v} \in \mathbb{B}$ is 1 if the platform is at node v at time k and 0 otherwise. These are the main decision variables, since they yield a search-trajectory on the reduced graph. For the accurate calculation of the objective function two types of auxiliary decision variables are used. Auxiliary decision variable $pd_{k,c} \in [0, 1]$ represents the probability of detection in cell c at time k and auxiliary decision variable $pc_{k,c} \in [0, 1]$ represents the probability of containment in cell c at time k. In the remainder of this section, decision variables are written in *italic* font, whereas variables for input data are written in normal font.

Objective: Recall the objective from Subsection III-E. We aim to maximize the cumulative probability of detection by summing over each cell $c \in \mathcal{C}$ and each time step $k \in \mathcal{K}$. Hence the objective is

$$\max \sum_{k=1}^{K} \sum_{c\in\mathcal{C}} pd_{k,c} \tag{9}$$

subject to Constraints (10)-(22).

Objective Constraints: To assure calculation of auxiliary decision variables $pd_{k,c}$ according to Equation (8) we introduce the following objective constraints:

$$\forall c \in \mathcal{C}, \forall v \in \mathcal{V}, \forall k \in \mathcal{K}: \quad pd_{k,c} - \mathrm{pg}_{k,v,c} pc_{k,c} \leq 1 - z_{k,v} \quad (10)$$

This set of constraints only restrict auxiliary decision variables $pd_{k,c}$ sufficiently when node v is selected for time k. The following set of constraint ensure $pd_{k,c}$ to be at most equal to the glimpse probability from the visited node (this is at most one each time), which is less restrictive compared to the constraints (10). But more importantly, it restricts from obtaining search rewards from non-visited nodes, and thereby complementing the constraints (10).

$$\forall c \in \mathcal{C}, \forall k \in \mathcal{K}: \sum_{v \in \mathcal{V}} z_{k,v} \mathrm{pg}_{k,v,c} \geq pd_{k,c} \quad (11)$$

The combination of constraints (10) and (11) ensure accurate calculation of auxiliary decision variables $pd_{k,c}$.

Update Constraints: Auxiliary decision variables $pc_{k,c}$ must evolve over time according to Equation (4), as prescribed by the transition function $\mathrm{d}_{c',c}$. This is ensured by the following constraints:

$$\forall c \in \mathcal{C}, \forall k \in \{2, ..., K\}: \quad \sum_{c' \in \mathcal{C}} \mathrm{d}_{c',c} pc_{k-1,c'} - \sum_{c' \in \mathcal{C}} \mathrm{d}_{c',c} pd_{k-1,c'} = pc_{k,c} \quad (12)$$

Trajectory Constraints: The following constraints are introduced to ensure the necessary structure of a trajectory. First of all, the trajectory must be a sequence of nodes. So for any time k a maximum of one node can be selected, i.e.

$$\forall k \in \mathcal{K}: \sum_{v \in \mathcal{V}} z_{k,v} \leq 1. \quad (13)$$

This sum may be zero, because it is possible for a platform to relocate during a number of time steps, say k', before the search starts. In this case, no nodes are selected for time steps $k < k'$. If the platform is at a node at time $k \geq k'$, it also has to be at a node at time $k+1$, i.e.

$$\forall k \in \{1, ..., K-1\}: \sum_{v \in \mathcal{V}} z_{k,v} - \sum_{v' \in \mathcal{V}} z_{k+1,v'} \leq 0. \quad (14)$$

The next constraint ensures the adjacency of direct successive nodes. When node v is not adjacent to node v', i.e. $\widetilde{a}_{v,v'} = 1$, the platform can either be at node v at time k or at a non-adjacent node v' at time $k+1$ or at neither of the two nodes, i.e.

$$\forall k \in \{1, ..., K-1\}, \forall v \in \mathcal{V}: \quad \sum_{v' \in \mathcal{V}} \tilde{\mathrm{a}}_{v,v'} z_{k+1,v'} + z_{k,v} \leq 1. \quad (15)$$

Kinematical Constraints: The flight kinematic constraints ensure that the trajectory contains no sharp turn, no loop and no cycle of length two for fixed-wing platforms. To this end, we introduce the parameter $\psi \in \mathbb{B}$, which assumes the value zero if the platform is fixed-winged. In this case, the right-hand-side value equals one. When the platform was at node v at time $k-1$ and node v is adjacent to node v', i.e. $a_{v,v'} = 1$, the platform cannot be at node v' at time $k+1$, i.e.

$$\forall k \in \{2, ..., K-1\}, \forall v \in \mathcal{V}: \quad \sum_{v' \in \mathcal{V}} \mathrm{a}_{v,v'} z_{k+1,v'} + z_{k-1,v} \leq 1 + \psi. \quad (16)$$

A rotary-wing platform, however, is able to make sharp turns, hover and fly backwards. Therefore, such platforms should not be restricted by this constraint. The parameter ψ assumes the value one if the platform is rotary-winged. In this case, the right-hand-side value equals two. This constraint is thereby relaxed for rotary-wing platforms.
At the start of a planning stage, the platform either relocates or keeps searching. These scenarios require different constraints; either the relocation constraint or the connecting constraint.

Relocation Constraint: The relocation constraint ensures the reachability of an assigned node. It prevents the assignment of the platform to a node at a time k, when it is physically out of reach. This is achieved by restricting the sum of non-reachable nodes to be zero over the entire planning horizon, i.e.

$$\sum_{v \in \mathcal{V}} \sum_{k=1}^{K} z_{k,v} \tilde{\mathrm{r}}_{k,v} = 0. \quad (17)$$

Recall from Subsection IV-B that the parameter $\tilde{\mathrm{r}}_{k,v}$ is 1 if node v is not reachable in k time steps and 0 otherwise.

Connecting Constraints: The connecting constraints ensure the physical feasibility over the entire duration of the search mission by connecting the walks of the consecutive planning stages. The connection requires an overlap of two nodes, since the first newly assigned node is restricted by the second to last node (denoted by v_{-1}) and by the last node (denoted by v_0). These nodes have been selected in the previous planning stage and are therefore mandatory at the start of the next trajectory, i.e.

$$z_{0,v_0} = 1 \text{ and } z_{-1,v_{-1}} = 1. \quad (18)$$

The trajectory constraints (13), (14), and (15) for time $k = 0$ and kinematical constraints (16) for times $k = -1$ and $k = 0$ are furthermore included to ensure the physical feasibility of the connection between the walks.

To be able to fix the last two nodes v_{-1} and v_0 in the next planning stage, at least two nodes are to be selected in the current planning stage. This is ensured by the following constraint:

$$\sum_{v \in \mathcal{V}} \sum_{k=1}^{K} z_{k,v} \geq 2. \quad (19)$$

Initialization constraints: The following constraint specifies the initial probability of containment.

$$\forall c \in \mathcal{C} : pc_{1,c} = \mathrm{pc}_{1,c}. \tag{20}$$

Binary constraints: The binary constraints for the main decision variables are

$$\forall k \in \mathcal{K}, \forall v \in \mathcal{V} : z_{k,v} \in \mathbb{B}. \tag{21}$$

Probability constraints: The last constraints restrict the auxiliary variables. These represent probabilities and are therefore restricted to the unit interval. For the probability of detection and for the probability of containment we have

$$\forall k \in \mathcal{K}, \forall c \in \mathcal{C} : 0 \leq pd_{k,c}, pc_{k,c} \leq 1. \tag{22}$$

The total number of decision variables, including the auxiliary decision variables, is of order $\mathcal{O}(K|\mathcal{V}||\mathcal{C}|)$.

V. Simulations

In this section, we present the simulation environment and results. The novel MILP is benchmarked against an established method for autonomous UAV trajectory-planning. First, we describe the benchmark method in Subsection V-A, followed by the experimental set-up in Subsection V-B. Finally, we present the results in Subsection V-C.

All simulations were performed on an Intel(R) CoreTM i7-4810MQ CPU processor with 2.80 GHz and a usable memory of 15.6 GB. The simulation platform is written in Matlab, using Gurobi with optimized parameter settings to solve the MILPs.

A. Benchmark Method

We compared the MILP method with the artificial potential field (APF) method [23]. We chose this well established method, because it is often used in control applications including scenarios for autonomous UAV trajectory-planning and, in particular, UAV search for moving targets [24]. The APF method is a myopic method, where the next node v for time $k+1$ is decided on by maximizing the potential $f(v, k+1)$, as follows:

$$\max_{v \in \mathcal{V}(v_k) \setminus \mathcal{V}(v_{k-1})} f(v, k+1) = \sum_{c \in \mathcal{C}} \frac{pc_{k+1,c}}{\sqrt{\|v - c\|}}, \tag{23}$$

where $\|v - c\|$ is the Euclidean distance between node v and cell c. Another reason to use the APF method is its robustness to the challenge of relocation which we would face with other myopic methods.

B. Experimental Set-Up

We describe the set-up for the comparative simulations in the following, starting with the used platform characteristics. For a fair comparison between our anticipatory method and the myopic APF method, we used $s_{reloc} = M$ as the relocation speed, where M is some large value such that each node is reachable within one time step. This way, the disadvantageous greedy character of the APF is canceled out during relocation. The start position $[0, 0]$ is irrelevant due to the large M and the homogeneous environment. We used $s = 2$ as the search speed, resulting in a distance between nodes of $l = 2$. Consequently, the inradius of the hexagons is one. For sensor characteristics, we used the typical glimpse probability function [10]:

$$pg_{k,o,c} = 1 - \exp^{-\omega(k,o,c)}, \tag{24}$$

with $\omega(k, o, c) \geq 0$ being a measure of search effectiveness for cell c. The search effectiveness decreases with the Euclidean distance $||o - c||$ between cell c and the platform at o and with the effect of disturbances δ_k at time k, as follows:

$$\omega(k, o, c) = W \left(||o - c|| + \delta_k\right)^{-1},$$

where W is some sensor quality indicator, drawn from the uniform distribution, i.e. $W \sim \mathcal{U}(0.25, 1.25)$. Disturbances δ_k were fixed at zero for all $k \in \mathcal{K}$. The simulated search missions take place on a square grid of size 80×80. In all tests, the initial target location C_0 was bivariate normally distributed ($C_0 \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$), with $\boldsymbol{\mu} = \left(\begin{smallmatrix} 40 \\ 40 \end{smallmatrix}\right)$ and $\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$, with $\sigma_1^2, \sigma_2^2 \sim \mathcal{U}(3, 7)$.

Three types of target movement models have been considered:

- **Diffusing**: The target moves in either north, east, south, or west direction with equal probability (See Fig. 7);
- **Conditionally Deterministic**: The target moves in east direction with probability one (See Fig. 8);
- **Markovian**: The target moves in either north or east direction with equal probability (See Fig. 9).

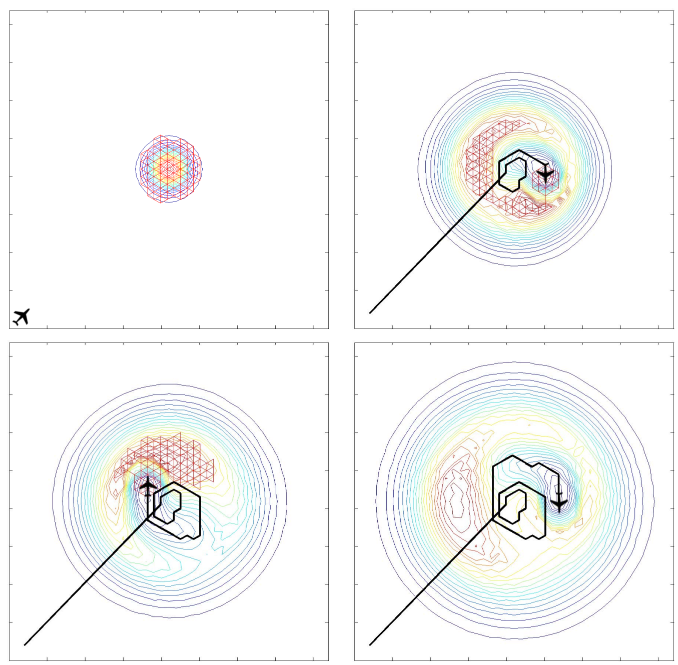


Fig. 7: Snapshots of a simulation of one fixed-wing platform searching for a single target with a diffusing motion model.

For each of the three motion models, we ran a total of 60 test instances on randomized initial probability maps, resulting in 20 for each of the time periods $K \in \{4, 6, 8\}$. Success was measured by the achieved cumulative probability of detection.

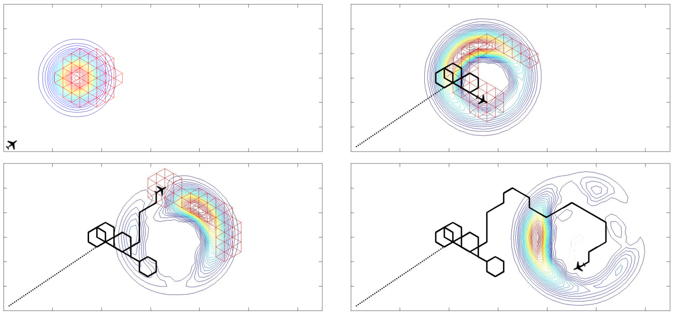


Fig. 8: Snapshots of a simulation of one fixed-wing platform searching for a target with a conditionally deterministic motion model.

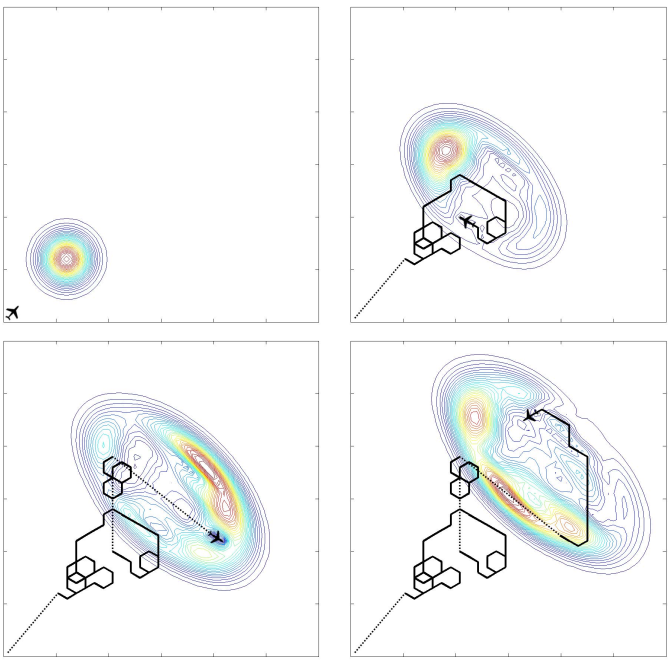
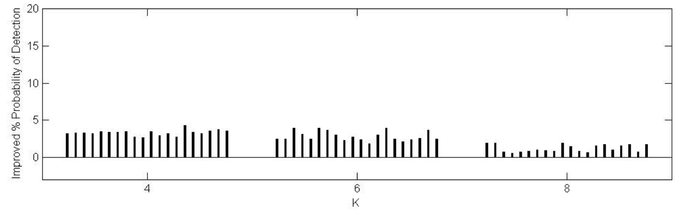


Fig. 9: Snapshots of a simulation of one fixed-wing platform searching for a single target with a Markovian motion model.

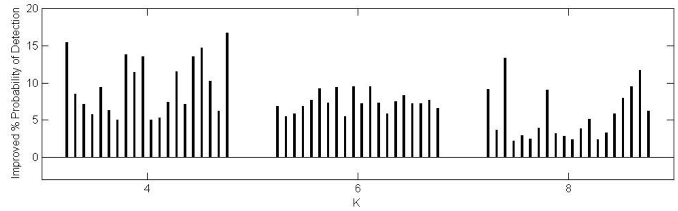
C. Results

The performance results are presented in Fig. 10. Here, the improved probability of detection is plotted relative to the results of the APF method. Fig. 10a shows the results of tests concerning a target with a diffusing motion model. Fig. 10b and Fig. 10c show the results of tests concerning a conditionally deterministic motion model and a Markovian motion model respectively. The results show that the MILP method yields better results when compared to the APF in each test instance. The MILP method shows especially powerful at maximizing the probability of detection in the conditionally deterministic target case. Here, improvements of up to 16.8% are achieved.

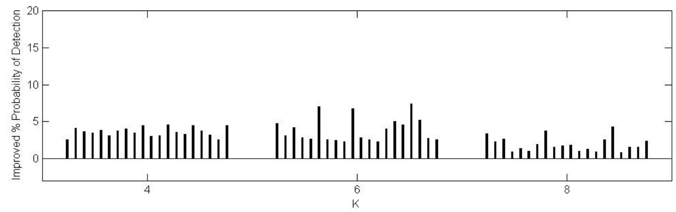
Besides the achieved cumulative probability of detection, we furthermore recorded the computation time. The computation time is effected by K, by the number of nodes $|\mathcal{V}|$, and by the number of cells $|\mathcal{C}|$. Recall from Subsection IV-D, that the number of decision variables is of order $\mathcal{O}(K|\mathcal{V}||\mathcal{C}|)$. In Fig. 11, runtime is plotted as a function of the number of decision variables in the MILP. We set a limit on the computation time at 3600 seconds, which was reached in four out of all test instances. In this case, the solver returns the so far best found solution, which was still better than the solution found by the



(a) Diffusing motion model



(b) Conditionally deterministic motion model



(c) Markovian motion model

Fig. 10: Simulation results in improved probability of detection relative to the results of the APF method.

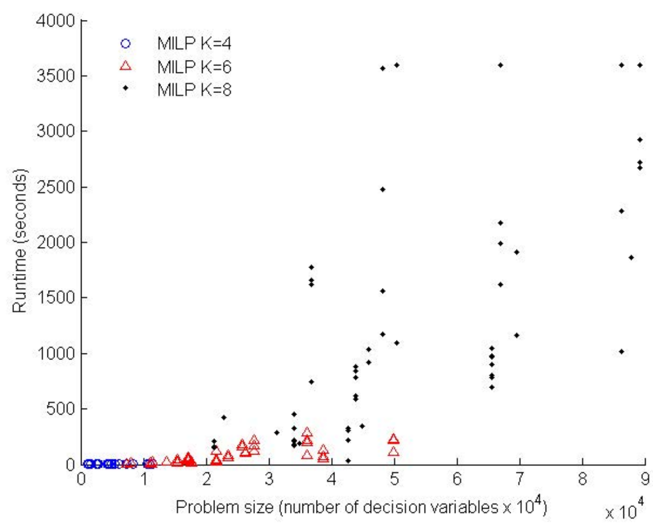


Fig. 11: Runtime plotted as a function of the number of decision variables in the MILP.

APF method in every occurrence.

The advantage of the APF method is the very low computational cost. It required less than one second on all instances. In the MILP method, on the other hand, optimization is performed during the execution of a previously planned search-trajectory. Therefore, it suffices to find a solution within the order of minutes. However, the computation time needed to prove optimality grew rapidly with K. The problem was solved within nine seconds for all instances with $K = 4$ and within 283 seconds for all instances with $K = 6$, but computation time exceeded one hour for some instances with $K = 8$. An estimation on available computation time can be acquired using the standard turn rate in aviation. It takes a platform two minutes to turn $360°$ when turning with the standard turn rate. This takes six nodes on the reduced graph. So a time step would approximately have a duration of 20 seconds in the application. This means that when $K = 12$, the solver has four minutes to plan for the next stage. We conclude that for larger instances, solving the MILP using a commercial solver is insufficient. However, customized algorithms on similar problems yield promising results [11].

VI. Conclusion

Several recent cases, especially emergencies at sea, have shown the high importance of acquiring knowledge about the location of a target. Natural disasters, extreme cases in air transport and the shipping industry as well as terrorist threats, are the basis for possible scenarios of interest. Many of these scenarios can have disastrous consequences when the target is not found (in time). Effective search-trajectory optimization methods are therefore needed. We presented a novel approach, consisting of a mixed integer linear program. This model accounts for the kinematics of the platforms and the fluidity of the trajectory, as well as for the uncertain movements of the target. It furthermore enables K-step-lookahead planning, which yields a higher probability of detection than myopic approaches. We ran simulations to substantiate this proposition. Results show that our approach yields much better results in terms of probability of detection on all test instances, which is crucial in emergency operations.

For now, the main goal of this paper was to introduce and demonstrate the new model. The focus of our future research is on developing a customized algorithm (e.g. [25]), such that optimal search-trajectorys can be planned efficiently for larger instances.

Acknowledgment

This work is supported by Airbus Defence & Space.

References

[1] UN High Commissioner for Refugees (UNHCR), "The sea route to europe: The mediterranean passage in the age of refugees," 2015, last accessed: 24 Feb 2016. [Online]. Available: http://www.refworld.org/docid/5595045a4.html

[2] T. Kratzke, L. Stone, and J. Frost, "Search and rescue optimal planning system," in *Proceedings of the 13th Conference on Information Fusion*, jul 2010, pp. 1–8.

[3] K. E. Trummel and J. R. Weisinger, "The complexity of the optimal searcher path problem," *Operations Research*, vol. 34, no. 2, pp. 324–327, mar 1986.

[4] M. Raap, S. Meyer-Nieberg, S. Pickl, and M. Zsifkovits, "Aerial vehicle search-path optimization: A novel method for emergency operations," *Journal of Optimization Theory and Applications*, pp. 1–19, 2016.

[5] B. O. Koopman, *Search and screening: general principles with historical applications*. Pergamon Press, 1980.

[6] L. D. Stone, *Theory of Optimal Search*, 1975.

[7] L. D. Stone, J. O. Royset, and A. R. Washburn, *Optimal Search for Moving Targets*. Springer, 2016.

[8] T. Stewart, "Search for a moving target when searcher motion is restricted," *Computers & Operations Research*, vol. 6, no. 3, pp. 129–140, 1979.

[9] J. N. Eagle and J. R. Yee, "An optimal branch-and-bound procedure for the constrained path, moving target search problem," *Operations Research*, vol. 38, no. 1, pp. 110–114, feb 1990.

[10] G. H. A. Martins, "A new branch and bound procedure for computing optimal search paths," Master's thesis, Naval Postgraduate School, Monterey, CA, 1993.

[11] H. Lau, S. Huang, and G. Dissanayake, "Discounted MEAN bound for the optimal searcher path problem with non-uniform travel times," *European Journal of Operational Research*, vol. 190, no. 2, pp. 383–397, 2008.

[12] H. Sato and J. O. Royset, "Path optimization for the resource-constrained searcher," *Naval Research Logistics*, vol. 57, no. 5, pp. 422–440, 2010.

[13] A. R. Washburn, "Branch and bound methods for a search problem," *Naval Research Logistics (NRL)*, vol. 45, no. 3, pp. 243–257, 1998.

[14] J. N. Eagle, *The approximate solution of a simple constrained search path moving target problem using moving horizon policies*. Monterey, California. Naval Postgraduate School, 1984.

[15] P. Lanillos, E. Besada-Portas, G. Pajares, and J. Ruz, "Minimum time search for lost targets using cross entropy optimization," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, oct 2012, pp. 602–609.

[16] M. Morin, A. P. Papillon, I. Abi-Zeid, F. Laviolette, and C. G. Quimper, "Constraint programming for path planning with uncertainty," in *Principles and Practice of Constraint Programming*, ser. Lecture Notes in Computer Science, M. Milano, Ed. Springer Berlin Heidelberg, 2012, pp. 988–1003.

[17] F. Bourgault, T. Furukawa, and H. Durrant-Whyte, "Coordinated decentralized search for a lost target in a bayesian world," in *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, oct 2003, pp. 48–53.

[18] ——, "Decentralized bayesian negotiation for cooperative search," in *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, sep 2004, pp. 2681–2686.

[19] ——, "Optimal search for a lost target in a bayesian world," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2006, vol. 24, pp. 209–222.

[20] J. Norris, *Markov Chains*, ser. Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, 1999, no. 2008.

[21] E. Álvarez-Miranda, I. Ljubić, and P. Mutzel, "The maximum weight connected subgraph problem," in *Facets of Combinatorial Optimization*. Springer, 2013, pp. 245–270.

[22] D. B. West *et al.*, *Introduction to graph theory*. Prentice Hall Upper Saddle River, 2001, vol. 2.

[23] J. Baxter, E. Burke, J. Garibaldi, and M. Norman, "Multi-robot search and rescue: A potential field based approach," in *Autonomous Robots and Agents*, ser. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2007, vol. 76, ch. 2, pp. 9–16.

[24] X. Xiao, Z. Dong, J. Wu, and H. Duan, "A cooperative approach to multiple UAVs searching for moving targets based on a hybrid of virtual force and receding horizon," in *INDIN*. IEEE, 2012, pp. 1228–1233.

[25] M. Raap, M. Moll, M. Zsifkovits, and S. Pickl, "Utilizing dual information for moving target search trajectory optimization," in *Proceedings of the Student Conference on Operational Research (SCOR)*, 2016.