SAND2018-1906C

Multiple Pursuer-Based Intercept via Forward Stochastic Reachability

Abraham P. Vinod[1], Baisravan HomChaudhuri[1], Christoph Hintz[1], Anup Parikh[2], Stephen P. Buerger[2], Meeko M. K. Oishi[1], Greg Brunson[1,2], Shakeeb Ahmad[1], and Rafael Fierro[1]

***Abstract*— We discuss the multiple pursuer-based intercept of a threat unmanned aerial system (UAS) with stochastic dynamics via multiple pursuing UASs, using forward stochastic reachability and receding horizon control techniques. We formulate a stochastic model for the threat that can emulate the potentially adversarial behavior and is amenable to the existing scalable results in forward stochastic reachability literature. The optimal state for the intercept for each individual pursuer is obtained via a log-concave optimization problem, and the open-loop control paths are obtained via a convex optimization problem. With stochasticity modeled as a Gaussian process, we can approximate the optimization problem as a quadratic program, to enable real-time path planning. We also incorporate real-time sensing into the path planning by using a receding horizon controller, to improve the intercept probabilities. We validate the proposed framework via hardware experiments.**

I. Introduction

Protecting assets in urban environments has become an increasingly relevant problem with the widespread use of drones. Traditional defense mechanisms, like ballistic defenses and RF/GPS jamming options, are not viable due to the potential for collateral damage. We propose an air-to-air approach, entitled Aerial Suppression of Airborne Platforms (ASAP) (Figure 1), which uses defensive unmanned aerial systems (UASs) in coordination with ground-based systems. The objective of ASAP is to detect, track, and, if needed, neutralize small threat UAS using multiple pursuer UASs. By moving airborne sensors and precision defense systems away from ground interference and near to potential threat vehicles, this approach exploits geometric advantages, such as multiple perspectives, significantly increased angular cross section, and the ability to use short-range precision maneuvers for neutralization.

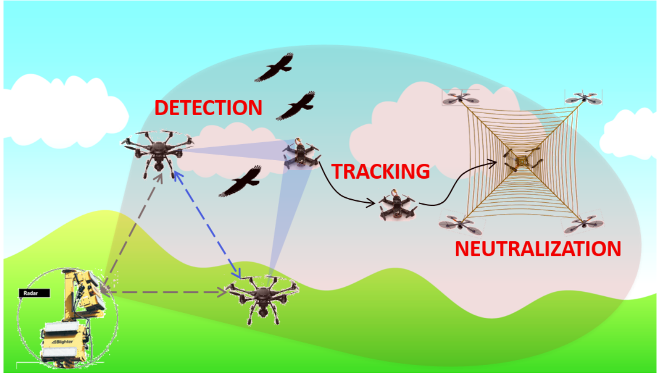


Fig. 1: High-level depiction of Aerial Suppression of Airborne Platform (ASAP) system

In this paper, we focus on the tracking component of ASAP, *i.e*, the interception of a small threat whose intentions may or may not be adversarial. Using a differential game framework [1]–[4], researchers have presumed a worst-case scenario, with purely adversarial threats, and constructed conservative controllers. Backward reachability and viability calculations can also be used to compute the set of states from which the intercept of an adversarial dynamical threat is possible [5], [6]. While these methods provide absolute and probabilistic guarantees of intercept, respectively, they are computationally intractable as the dimensionality of the problem increases, and require large offline computations. Moreover, the worst-case scenario may be overly conservative, depending on the actual intentions of the threat, and create an unintentionally large set of possible threat locations.

Another approach is to use forward reachable sets [7], [8] to compute possible positions of the threat. These methods do not incorporate the stochastic information available on the threat dynamics via sensing, and therefore also tend to be overly conservative as they do not distinguish between locations of the threat with higher likelihood. In [9], the intercept problem is posed as a partially observable Markov decision process (POMDP) and solved approximately using graph theory algorithms. Our paper builds on the results presented in [10], which provided a globally optimal solution, using multiple convex optimization problems, to the path planning problem for a single pursuer in pursuit of a nonadversarial stochastic target. We also leverage recent results on forward stochastic reachability for affine systems [11].

In this paper, we presume the threat attempts to move towards the asset with bounded control authority. We assume the threat dynamics follow a known model derived from our knowledge of the likely targets and the observed trajectory prior to our engagement. To mitigate the computational intractability associated with the traditional differential game approach, we assume that the threat's actions are independent of the pursuers' actions. We incorporate known additive stochastic disturbances to account for the uncertainty in this

This material is based upon work supported by the Laboratory Directed Research and Development (LDRD) program at Sandia National Laboratories, and the National Science Foundation under Grant Number CMMI-1254990 and CNS-1329878. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[1] Electrical and Computer Engineering, University of New Mexico. Corresponding authors: oishi@unm.edu, rfierro@unm.edu.

[2] Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

threat model. We leverage existing *scalable* results in forward stochastic reachability to solve the multiple pursuer-based intercept problem. Using the probability measure associated with the state of the threat at future times of interest, we formulate optimization problems to compute the optimal state for intercept for the pursuers and the associated open-loop controllers. Finally, we incorporate real-time sensing of the position of the threat and pursuers to the intercept planning problem via receding horizon control.

The main contributions of this paper are 1) a stochastic model for a potentially adversarial threat, 2) extension of our solution to the single-pursuer problem [10] to multiple pursuers, 3) incorporation of real-time sensor data into the path-planning algorithm using a receding horizon framework, and 4) experimental validation of the Fourier transform-based forward stochastic reachability techniques. Our approach has two main advantages: 1) it is not overly conservative, as it does not create an unnecessarily large set of possible threat positions, and 2) the resulting convex optimization yields computationally tractable problems and enables computation of a globally optimal solution in the space of open-loop controllers for the pursuers. While we restrict our model to Gaussian stochastic processes for real-time tractability, our approach generalizes to other continuous disturbance processes, for example, an exponential disturbance process [10].

This paper is organized as follows: Section II formulates the problem statement. Section III describes our stochastic model for the threat, the forward stochastic reachability-based solution to the multiple pursuer-based intercept problem, and the implementation of pursuit via receding horizon control. Section IV validates the results in an experimental testbed, and Section V provides conclusions and future directions.

II. Preliminaries

We denote random vectors with bold case, non-random vectors with an overline, and a discrete interval by $\mathbb{N}_{[a,b]}$ for $a, b \in \mathbb{N}$ and $a \leq b$, which inclusively enumerates all natural numbers in between a and b. We denote the Minkowski sum as $\oplus$, the Kronecker product as $\otimes$, $\overline{x}_{p\times q} \in \mathbb{R}^{p\times q}$ as a matrix with all of its elements as $x \in \mathbb{R}$, and I_n as the identity matrix of size n. The Cartesian product of the set $\mathcal{S}$ with itself $k \in \mathbb{N}$ times is $\mathcal{S}^k$. The Fourier transform operator is denoted by $\mathscr{F}$, and the inverse Fourier transform operator is denoted by $\mathscr{F}^{-1}$.

A. System formulation

We describe the lateral and rotational movement of the quadrotor in the inertial frame. We neglect the rotor inertia and analyze the quadrotor at the hovering condition, which allows approximation of the angles measured in the quadrotor frame with the Euler angles that determine the quadrotor attitude. Nonlinear dynamics describing the rigid body quadrotor are obtained using Newton-Euler formalism [12],

$$\ddot{x} = \frac{u_1}{m}(\cos\psi\sin\theta + \cos\theta\sin\phi\sin\psi) \tag{1a}$$

$$\ddot{y} = \frac{u_1}{m}(\sin\psi\sin\theta - \cos\psi\cos\theta\sin\phi) \tag{1b}$$

$$\ddot{z} = \frac{u_1}{m}(\cos\phi\cos\theta) - g \tag{1c}$$

$$\ddot{\phi} = \frac{I_{yy} - I_{zz}}{I_{xx}}\dot{\theta}\dot{\psi} + \frac{u_2}{I_{xx}} \tag{1d}$$

$$\ddot{\theta} = \frac{I_{zz} - I_{xx}}{I_{yy}}\dot{\phi}\dot{\psi} + \frac{u_3}{I_{yy}} \tag{1e}$$

$$\ddot{\psi} = \frac{I_{xx} - I_{yy}}{I_{zz}}\dot{\theta}\dot{\phi} + \frac{u_4}{I_{zz}} \tag{1f}$$

in which x, y and z describe the lateral movement, and ϕ, θ, and ψ approximate the roll, pitch, and yaw angles. The collective thrust is described by u_1, and the moments around the x, y, and z axes created by the difference in the motor speeds are described by u_2, u_3, and u_4, respectively. The quadrotor mass is m and g denotes the acceleration due to gravity. The inertia around x, y, and z axes are described by I_{xx}, I_{yy}, and I_{zz}, respectively.

Let the number of pursuers be N_p and $T \in \mathbb{N}$ be the time horizon. We model each pursuer P_i for $i \in \mathbb{N}_{[1,N_p]}$ as

$$\overline{x}_{P_i}[k+1] = A_P\overline{x}_{P_i}[k] + B_P\overline{u}_{P_i}[k] \tag{2}$$

with state $\overline{x}_{P_i}[k] \in \mathcal{X} \subseteq \mathbb{R}^{12}$, bounded input $\overline{u}_{P_i}[k] \in \mathcal{U}_P \subset \mathbb{R}^4$, and matrices A_P, B_P obtained by linearizing (1) around the hovering point with $u_1 = mg$ and known parameters. Let $\overline{x}_{P_i}[0] \in \mathcal{X}$ denote the known initial state of the pursuer P_i. Given $\tau \in \mathbb{N}_{[1,T]}$, an open-loop control policy $\overline{\pi}_{\tau}^{P_i} : \mathcal{X} \rightarrow \mathcal{U}_P^{\tau}$ for the pursuer depends only the on the initial state $\overline{x}_{P_i}[0]$, that is, $\overline{\pi}_{\tau}^{P_i}(\overline{x}_{P_i}[0]) = \left[(\overline{u}_{P_i}[\tau-1])^\top\ (\overline{u}_{P_i}[\tau-2])^\top\ \ldots\ (\overline{u}_{P_i}[0])^\top\right]^\top$. Let $\overline{\mathcal{M}}_{\tau}^{P_i}$ denote the set of admissible open-loop control policies, and $\mathscr{C}_P(\tau) = [B_P\ A_PB_P\ A_P^2B_P\ \ldots\ A_P^{\tau-1}B_P] \in \mathbb{R}^{12\times(4\tau)}$. From (2), we have for every $\overline{\pi}_{\tau}^{P_i} \in \overline{\mathcal{M}}_{\tau}^{P_i}$,

$$\overline{x}_{P_i}[\tau] = A_P^{\tau}\overline{x}_{P_i}[0] + \mathscr{C}_P(\tau)\overline{\pi}_{\tau}^{P_i}(\overline{x}_{P_i}[0]). \tag{3}$$

The threat G dynamics are given by

$$\boldsymbol{x}_G[k+1] = A_G\boldsymbol{x}_G[k] + B_G(\overline{u}_G[k] + \boldsymbol{w}[k]) \tag{4}$$

with state $\boldsymbol{x}_G[k] \in \mathcal{X}$, bounded input $\overline{u}_G[k] \in \mathcal{U}_G \subset \mathbb{R}^4$, disturbance $\boldsymbol{w}[k] \in \mathcal{W} \subseteq \mathbb{R}^4$, matrices A_G, B_G obtained by linearizing (1) around the hovering point with $u_1 = mg$ with the parameters estimated in the detection phase of ASAP. Similarly to $\overline{\pi}_{\tau}^{P_i}$ and $\overline{\mathcal{M}}_{\tau}^{P_i}$, we define an open-loop control policy $\overline{\pi}_{\tau}^{G} : \mathcal{X} \rightarrow \mathcal{U}_G^{\tau}$ with $\overline{\pi}_{\tau}^{G}(\overline{x}_G[0]) = \left[(\overline{u}_G[\tau-1])^\top\ (\overline{u}_G[\tau-2])^\top\ \ldots\ (\overline{u}_G[0])^\top\right]^\top$, and $\overline{\mathcal{M}}_{\tau}^{G}$ as the set of admissible open-loop control policies. We presume an open-loop controller for the threat to reach a stationary asset at $\overline{x}_a \in \mathcal{X}$, and describe additional assumptions on $\overline{\pi}_{\tau}^{G}(\overline{x}_G[0])$ in Section III-A.

Since our model of the threat's actions is based on sensed data and assumptions about its intent, we perturb the threat's predicted control actions by an IID ran-

$$\Psi_{\boldsymbol{x}_G}\left(\overline{\beta};\tau,\overline{x}_G[0],\overline{\pi}_\tau^G\right)=\exp\left(j\overline{\beta}^\top\overline{x}_G^{\mathrm{certain}}[\tau;\overline{x}_G[0],\overline{\pi}_\tau^G]\right)\Psi_{\boldsymbol{W}_\tau}\left(\left(\mathscr{C}_G(\tau)\right)^\top\overline{\beta}\right),\qquad \overline{\beta}\in\mathbb{R}^{12} \tag{5a}$$

$$\psi_{\boldsymbol{x}_G}\left(\overline{y};\tau,\overline{x}_G[0],\overline{\pi}_\tau^G\right)=\mathscr{F}^{-1}\left\{\Psi_{\boldsymbol{x}_G}\left(\overline{\beta};\tau,\overline{x}_G[0],\overline{\pi}_\tau^G\right)\right\}(-\overline{y}),\qquad \overline{y}\in\mathbb{R}^{12} \tag{5b}$$

dom process with known probability density $\psi_{\boldsymbol{w}}$. As discussed in [10], the independence assumption on the process may be relaxed to a T^{th}-order strict stationarity assumption on the process [13, Section 10.3] when we are provided with the joint density $\psi_{\boldsymbol{W}_T}(\cdot)$ of the concatenated disturbance random vector $\boldsymbol{W}_T$. For $\tau\in\mathbb{N}_{[1,T]}$, $\boldsymbol{W}_\tau=\left[(\boldsymbol{w}[\tau-1])^\top\ (\boldsymbol{w}[\tau-2])^\top\ \ldots\ (\boldsymbol{w}[0])^\top\right]^\top\in\mathbb{R}^{4\tau}$. Let $\overline{x}_G[0]$ denote the known initial state of the threat G, and we define $\mathscr{C}_G(\tau)=[B_G\ A_GB_G\ A_G^2B_G\ \ldots\ A_G^{\tau-1}B_G]\in\mathbb{R}^{12\times(4\tau)}$. From (4), we have for every $\overline{\pi}_\tau^G\in\overline{\mathcal{M}}_\tau^G$,

$$\boldsymbol{x}_G[\tau]=A_G^\tau\overline{x}_G[0]+\mathscr{C}_G(\tau)\overline{\pi}_\tau^G(\overline{x}_G[0])+\mathscr{C}_G(\tau)\boldsymbol{W}_\tau. \tag{6}$$

We denote the unperturbed state of the threat by $\overline{x}_G^{\mathrm{certain}}[\tau;\overline{x}_G[0],\overline{\pi}_\tau^G]$ obtained by setting $\boldsymbol{w}_k=0$ in (4),

$$\overline{x}_G^{\mathrm{certain}}[\tau;\overline{x}_G[0],\overline{\pi}_\tau^G]=A_G^\tau\overline{x}_G[0]+\mathscr{C}_G(\tau)\overline{\pi}_\tau^G(\overline{x}_G[0]). \tag{7}$$

B. Deterministic forward reachable sets for the pursuers

For planning feasible trajectories for the pursuers, we will use the *forward deterministic reach set*, which is the set of all states that can be reached by a deterministic system at a time of interest when starting at a known initial condition. Formally, the forward deterministic reach set for P_i at time $\tau\in\mathbb{N}_{[1,T]}$ is described by

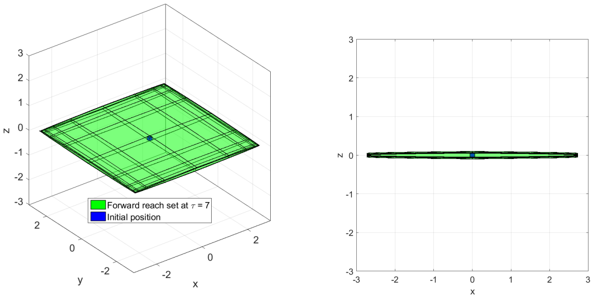
$$\begin{aligned}\mathrm{Reach}_{P_i}(\tau;\overline{x}_{P_i}[0])&=\{\overline{y}\in\mathcal{X}:\exists\overline{\pi}_\tau^{P_i}\in\overline{\mathcal{M}}_\tau^{P_i},\overline{x}_{P_i}[\tau]=\overline{y}\}\\&=\{A_P^\tau\overline{x}_{P_i}[0]\}\oplus\mathscr{C}_P(\tau)\mathcal{U}_P^\tau\end{aligned} \tag{8}$$

where (8) follows from (3). From (8), it follows that $\mathrm{Reach}_{P_i}(\tau;\overline{x}_{P_i}[0])$ is a polytope when $\mathcal{U}_P$ is a polytope. Computation of (8) for linear systems is straightforward with established tools, such as MPT [14], ET [15]. While implementing (8), we can compute the polytopes $\mathscr{C}_P(\tau)\mathcal{U}_P^\tau$ offline and translate by $A_P^\tau\overline{x}_{P_i}[0]$ when $\mathrm{Reach}_{P_i}(\tau;\overline{x}_{P_i}[0])$ is desired. This segregation of computation effort follows from the superposition principle. It eliminates redundant computations and provides significant computational savings due to the high computational costs associated with computing affine maps of high-dimensional polytopes.

Figure 2 shows the forward reach set for the linearized quadrotor dynamics. As expected, the reach set is symmetric in x and y directions. Due to bounded control authority, the displacement in z direction that can be achieved decreases as the desired displacement in x-y plane increases. The computation of Figure 2 took 2.36 seconds when performed using MATLAB on an Intel Core i7 CPU with 3.4 GHz clock rate and 16 GB RAM.

C. Stochastic forward reachability for the threat

For the stochastic system (4), the state $\boldsymbol{x}_G[\tau]$ for $\tau>0$ is a random vector characterized by its support and the probability measure. The support, $\mathrm{FSReach}(\tau,\overline{x}_G[0])$, is referred to as the *forward stochastic reach set*, and the probability measure $\mathbb{P}_{\boldsymbol{x}_G}^{\tau,\overline{x}_G[0],\overline{\pi}_\tau^G}$ as the *forward stochastic reach probability measure* (FSRPM) [10]. The FSRPM also defines a *forward stochastic reach probability density* (FSRPD) $\psi_{\boldsymbol{x}_G}(\cdot;\tau,\overline{x}_G[0],\overline{\pi}_\tau^G)$ for the random vector $\boldsymbol{x}_G[\tau]$ with



(a) (b)

Fig. 2: Forward reach set $\mathrm{Reach}_{P_i}(\tau;\overline{x}_{P_i}[0])$ computed for pursuer P_i (2) at $\tau=7$ and $\overline{x}_{P_i}[0]$ located at the origin. See Section IV-A for numerical values.

$$\mathbb{P}_{\boldsymbol{x}_G}^{\tau,\overline{x}_G[0],\overline{\pi}_\tau^G}\{\boldsymbol{x}_G[\tau]\in\mathcal{S}\}=\int_{\mathcal{S}}\psi_{\boldsymbol{x}_G}(\overline{y};\tau,\overline{x}_G[0],\overline{\pi}_\tau^G)d\overline{y} \tag{9}$$

for any Borel set $\mathcal{S}\subseteq\mathcal{X}$. For a continuous FSRPD, we have

$$\begin{aligned}&\mathrm{FSReach}(\tau,\overline{x}_G[0],\overline{\pi}_\tau^G)\\&=\mathrm{closure}\left(\left\{\overline{y}\in\mathcal{X}:\psi_{\boldsymbol{x}_G}\left(\overline{y};\tau,\overline{x}_G[0],\overline{\pi}_\tau^G\right)>0\right\}\right).\end{aligned} \tag{10}$$

Using Fourier transforms, we have recently developed computationally efficient methods to exactly calculate the FSRPD and the FSRPM for arbitrary disturbances [10], [11]. The characteristic function (CF) $\Psi_{\boldsymbol{x}}$ of the random vector $\boldsymbol{x}$ is defined as the Fourier transform of a probability density function $\psi_{\boldsymbol{x}}$. We define the CF of $\boldsymbol{W}_\tau$ as

$$\Psi_{\boldsymbol{W}_\tau}(\overline{\alpha})=\mathscr{F}\{\psi_{\boldsymbol{W}_\tau}(\cdot)\}(-\overline{\alpha}) \tag{11}$$

where $\overline{\alpha}\in\mathbb{R}^{4\tau}$ and $\psi_{\boldsymbol{W}_\tau}$ is the joint probability density of the random vector $\boldsymbol{W}_\tau$. From [10, Property P2], (6), and (7), we characterize the CF and FSRPD of $\boldsymbol{x}_G[\tau]$ as given in (5). Furthermore, (5b) admits a closed-form expression for the FSRPD when $\boldsymbol{w}$ is a Gaussian random vector which we exploit in Section III-A. This result enables scalable calculations of the threat's possible positions and their associated likelihood.

D. Problem statement

Define a closed convex set $\mathrm{CatchSet}(\overline{x}_{P_i}[\tau])\subseteq\mathcal{X}$ for each pursuer P_i, parameterized by the pursuer's current state $\overline{x}_{P_i}[\tau]$. Pursuer P_i intercepts or catches the threat G at $\tau\in\mathbb{N}_{[1,T]}$ if $\boldsymbol{x}_G[\tau]\in\mathrm{CatchSet}(\overline{x}_{P_i}[\tau])$. The probability of

$$\begin{array}{ll}\underset{\overline{\pi}_T^G(\overline{x}_G[0])}{\text{minimize}} & q\sum_{k=1}^{T}\left\|\overline{x}_G^{\text{certain}}[k;\overline{x}_G[0],\overline{\pi}_T^G]-\overline{x}_a\right\|_2^2+\left(\overline{\pi}_T^G(\overline{x}_G[0])\right)^\top\overline{\pi}_T^G(\overline{x}_G[0]) \\ \text{subject to} & \begin{cases}\overline{x}_G^{\text{certain}}[k;\overline{x}_G[0],\overline{\pi}_T^G]=A_G^k\overline{x}_G[0]+[\mathscr{C}_G(k)\ \overline{0}_{12\times 4(T-k)}]\overline{\pi}_T^G(\overline{x}_G[0]) & k=1,2,\ldots,T \\ \overline{\pi}_T^G(\overline{x}_G[0])\in\overline{\mathcal{M}}_T^G\end{cases}\end{array} \quad (12)$$

intercept by pursuer P_i is described by the catch probability function,

$$\begin{aligned}&\text{CatchPr}(\tau,\overline{x}_{P_i}[\tau];\overline{x}_G[0],\overline{\pi}_\tau^G)\\&\quad=\mathbb{P}_{\boldsymbol{x}_G}^{\tau,\overline{x}_G[0],\overline{\pi}_\tau^G}\{\boldsymbol{x}_G[\tau]\in\text{CatchSet}(\overline{x}_{P_i}[\tau])\} \quad (13)\\&\quad=\int_{\text{CatchSet}(\overline{x}_{P_i}[\tau])}\psi_{\boldsymbol{x}_G}(\overline{y};\tau,\overline{x}_G[0],\overline{\pi}_\tau^G)d\overline{y}. \quad (14)\end{aligned}$$

We define the team state and the *team catch probability* at time $\tau\in\mathbb{N}_{[1,T]}$ as

$$\overline{x}_{\text{team}}[\tau]\triangleq[\overline{x}_{P_1}[\tau],\overline{x}_{P_2}[\tau],...,\overline{x}_{P_N}[\tau]]\in\mathcal{X}^{N_p}, \quad (15)$$

$$\begin{aligned}&\text{TeamCatchPr}(\tau,\overline{x}_{\text{team}}[\tau];\overline{x}_G[0],\overline{\pi}_\tau^G)\\&\quad=\max_{i\in\mathbb{N}_{[1,N_p]}}\text{CatchPr}(\tau,\overline{x}_{P_i}[\tau];\overline{x}_G[0],\overline{\pi}_\tau^G). \quad (16)\end{aligned}$$

We then define the optimal team catch probability as

$$\begin{aligned}&\text{TeamCatchPr}(\tau^*,\overline{x}_{\text{team}}[\tau^*];\overline{x}_G[0],\overline{\pi}_\tau^G)\\&\quad=\max_{\tau\in\mathbb{N}_{[1,T]}}\text{TeamCatchPr}(\tau,\overline{x}_{\text{team}}[\tau];\overline{x}_G[0],\overline{\pi}_\tau^G)\end{aligned} \quad (17)$$

Problem 1: Given the initial states of the pursuers and of the threat, the stochastic dynamics of the threat, and the deterministic dynamics of the pursuer, compute the maximum team catch probability within the time horizon T and admissible open-loop controllers for the team of pursuers P_i, $i\in\mathbb{N}_{[1,N_p]}$ to achieve interception.

Problem 1.a: Model a potentially adversarial threat as a stochastic dynamical system, including design of an open-loop controller for the threat, that is independent of the pursuer's actions and drives the threat to the asset's location, with bounded control authority.

Problem 2: Implement Problem 1 in a receding horizon control framework to incorporate real-time state information.

III. Stochastic threat intercept with multiple pursuers

A. Modeling the threat dynamics

We address Problem 1.a by posing the threat modeling problem as the optimization problem (12) for some $q>0$. The optimization problem computes an open-loop controller that optimizes a quadratic cost function to emulate the potentially adversarial behavior of the threat. We denote the solution to problem (12) as $\overline{\pi}_T^{G,\text{opt}}(\overline{x}_G[0])$. This modeling approach for the threat requires information about the stochastic dynamics of the threat, the bounds on the control authority, and the parameter q in the optimization problem (12).

For real-time tractability, we approximate the problem (12) by solving the corresponding infinite-horizon deterministic linear quadratic regulator. We saturate the control actions when it exceeds the bounded control authority for the threat. From Section II-C and [11, Proposition 1], for $\boldsymbol{w}\sim\mathcal{N}(\overline{\mu}_{\boldsymbol{w}},\Sigma_{\boldsymbol{w}})$, we have $\boldsymbol{x}_G[k]\sim\mathcal{N}(\overline{\mu}_{\boldsymbol{x}_G}[k;\overline{x}_G[0],\overline{\pi}_T^{G,\text{opt}}],\Sigma_{\boldsymbol{x}_G}[k;\overline{x}_G[0],\overline{\pi}_T^{G,\text{opt}}])$ with

$$\begin{aligned}&\overline{\mu}_{\boldsymbol{x}_G}[k;\overline{x}_G[0],\overline{\pi}_T^{G,\text{opt}}]\\&\quad=\overline{x}_G^{\text{certain}}[k;\overline{x}_G[0],\overline{\pi}_T^{G,\text{opt}}]+\mathscr{C}_G(k)(\overline{1}_{k\times1}\otimes\overline{\mu}_{\boldsymbol{w}})\\&\quad=A_G^k\overline{x}_G[0]+\mathscr{C}_G(k)\left(\overline{\pi}_T^{G,\text{opt}}(\overline{x}_G[0])+\overline{1}_{k\times1}\otimes\overline{\mu}_{\boldsymbol{w}}\right),\end{aligned} \quad (18a)$$

$$\begin{aligned}&\Sigma_{\boldsymbol{x}_G}[k;\overline{x}_G[0],\overline{\pi}_T^{G,\text{opt}}]\\&\quad=\mathscr{C}_G(k)(I_k\otimes\Sigma_{\boldsymbol{w}})\big(\mathscr{C}_G(k)\big)^\top.\end{aligned} \quad (18b)$$

We will use $\overline{\mu}_{\boldsymbol{x}_G}[k;\cdot]$ and $\Sigma_{\boldsymbol{x}_G}[k;\cdot]$ when it is unambiguous.

Note that one may use other cost functions in (12) to emulate different threat behavior models. Since (12) needs to be solved frequently, convexity properties or formulations that have good convex approximations are recommended. The optimization problem proposed in (12) may be modified to solve for a closed loop nonlinear controller that can account for more aggressive behaviors. However, it would make the computation difficult and lead to a potential loss of real-time tractability.

B. Multiple pursuer-based threat intercept

For individual pursuer P_i, the problem of maximizing the catch probability can be posed as an optimization problem at time $\tau\in\mathbb{N}_{[1,T]}$,

$$\begin{array}{ll}\underset{\overline{x}_{P_i}[\tau]}{\text{maximize}} & \text{CatchPr}\left(\tau,\overline{x}_{P_i}[\tau];\overline{x}_G[0],\overline{\pi}_T^{G,\text{opt}}\right)\\ \text{subject to} & \overline{x}_{P_i}[\tau]\in\text{Reach}_{P_i}(\tau;\overline{x}_{P_i}[0])\end{array}. \quad (19)$$

Recall that the Gaussian distribution is log-concave [16, Section 2.3]. Further, integration of a log-concave function over a convex set is log-concave [17]. This implies $\text{CatchPr}\left(\tau,\overline{x}_{P_i}[\tau];\overline{x}_G[0],\overline{\pi}_T^{G,\text{opt}}\right)$ is log-concave in $\overline{y}$, from (14). As in [10, Prob. C], we can evaluate (19) for a global optimum in near real-time by converting it into a convex optimization problem,

$$\begin{array}{ll}\underset{\overline{x}_{P_i}[\tau]}{\text{minimize}} & -\log\left(\text{CatchPr}\left(\tau,\overline{x}_{P_i}[\tau];\overline{x}_G[0],\overline{\pi}_T^{G,\text{opt}}\right)\right)\\ \text{subject to} & \overline{x}_{P_i}[\tau]\in\text{Reach}_{P_i}(\tau;\overline{x}_{P_i}[0])\end{array}. \quad (20)$$

By a change of variables and the assumption that the catch set $\text{CatchSet}(\overline{0}_{12\times1})$ is small enough to approximate the

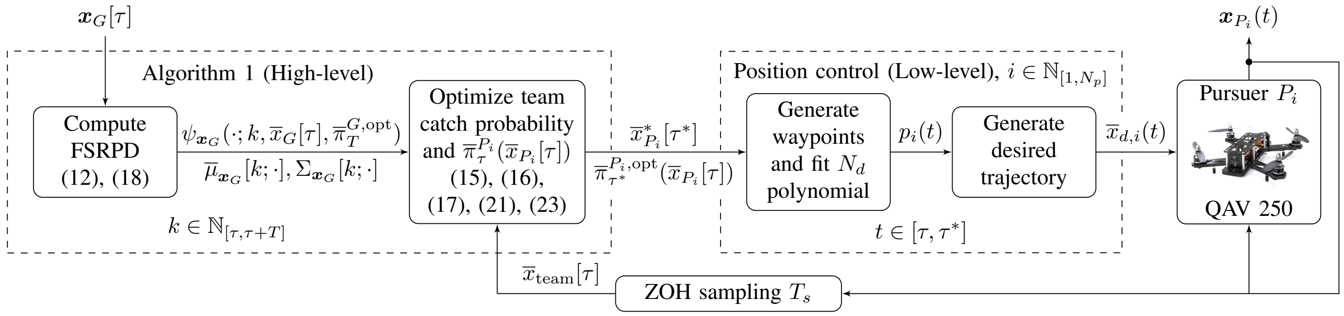

$\boldsymbol{x}_G[\tau]$
Algorithm 1 (High-level)
Compute FSRPD (12), (18)
$\psi_{\boldsymbol{x}_G}(\cdot; k, \overline{x}_G[\tau], \overline{\pi}_T^{G,\mathrm{opt}})$
$\overline{\mu}_{\boldsymbol{x}_G}[k;\cdot], \Sigma_{\boldsymbol{x}_G}[k;\cdot]$
Optimize team catch probability and $\overline{\pi}_\tau^{P_i}(\overline{x}_{P_i}[\tau])$ (15), (16), (17), (21), (23)
$k \in \mathbb{N}_{[\tau,\tau+T]}$
$\overline{x}^*_{P_i}[\tau^*]$
$\overline{\pi}_{\tau^*}^{P_i,\mathrm{opt}}(\overline{x}_{P_i}[\tau])$
Position control (Low-level), $i \in \mathbb{N}_{[1,N_p]}$
Generate waypoints and fit N_d polynomial
$p_i(t)$
Generate desired trajectory
$\overline{x}_{d,i}(t)$
$t \in [\tau, \tau^*]$
Pursuer P_i
QAV 250
$\boldsymbol{x}_{P_i}(t)$
$\overline{x}_{\mathrm{team}}[\tau]$
ZOH sampling T_s

Fig. 3: Flowchart depicting the receding horizon implementation of the multiple pursuer-based intercept problem. The initial states for the pursuers $\overline{x}_{P_i}[0]$ and the threat $\overline{x}_G[0]$ in the mentioned equations should be replaced with the current states $\overline{x}_{P_i}[\tau]$ and the threat $\boldsymbol{x}_G[\tau]$ respectively.

$$\begin{array}{ll}\underset{\overline{x}_{P_i}[\tau]}{\text{minimize}} & \left(\overline{x}_{P_i}[\tau] - \overline{\mu}_{\boldsymbol{x}_G}\left[k; \overline{x}_G[0], \overline{\pi}_T^{G,\mathrm{opt}}\right]\right)^\top \left(\Sigma_{\boldsymbol{x}_G}[\tau; \overline{x}_G[0], \overline{\pi}_T^{G,\mathrm{opt}}]\right)^{-1} \left(\overline{x}_{P_i}[\tau] - \overline{\mu}_{\boldsymbol{x}_G}\left[k; \overline{x}_G[0], \overline{\pi}_T^{G,\mathrm{opt}}\right]\right) \\ \text{subject to} & \overline{x}_{P_i}[\tau] \in \mathrm{Reach}_{P_i}(\tau; \overline{x}_{P_i}[0])\end{array}. \tag{21}$$

integrand in (14) to be a constant, we have

$$\begin{aligned} &-\log\left(\mathrm{CatchPr}\left(\tau, \overline{x}_{P_i}[\tau]; \overline{x}_G[0], \overline{\pi}_T^{G,\mathrm{opt}}\right)\right) \\ &\approx \frac{(\overline{x}_{P_i}[\tau] - \overline{\mu}_{\boldsymbol{x}_G}[\tau;\cdot])^\top (\Sigma_{\boldsymbol{x}_G}[\tau;\cdot])^{-1} (\overline{x}_{P_i}[\tau] - \overline{\mu}_{\boldsymbol{x}_G}[\tau;\cdot])}{2} \\ &\quad - \log\left(\mathrm{volume}(\mathrm{CatchSet}(\overline{0}_{12\times 1}))|2\pi\Sigma_{\boldsymbol{x}_G}[\tau;\cdot]|^{\frac{-1}{2}}\right). \end{aligned} \tag{22}$$

Since $-\log\left(\mathrm{volume}(\mathrm{CatchSet}(\overline{0}_{12\times 1}))|2\pi\Sigma_{\boldsymbol{x}_G}[\tau;\cdot]|^{\frac{-1}{2}}\right)$ is independent of $\overline{x}_{P_i}[\tau]$, we can approximate (20) as a quadratic program (21) for real-time tractability.

We use (3) to synthesize the open-loop controller for each pursuer to reach the state prescribed by (21). Defining a cost function $J_\pi : \mathcal{U}^{\tau^*} \to \mathbb{R}$, we compute the optimal open-loop controller for the pursuer P_i as follows

$$\begin{array}{ll}\underset{\overline{\pi}_{\tau^*}^{P_i}(\overline{x}_{P_i}[0])}{\text{minimize}} & J_\pi(\overline{\pi}_{\tau^*}^{P_i}(\overline{x}_{P_i}[0])) \\ \text{subject to} & \begin{cases} \mathscr{C}_P(\tau^*)\overline{\pi}_{\tau^*}^{P_i}(\overline{x}_{P_i}[0]) = \overline{x}^*_{P_i}[\tau^*] - A_P^{\tau^*}\overline{x}_{P_i}[0] \\ \overline{\pi}_{\tau^*}^{P_i}(\overline{x}_{P_i}[0]) \in \mathcal{U}_P^{\tau^*} \end{cases}\end{array}. \tag{23}$$

The objective function $J_\pi(\overline{\pi}_{\tau^*}^{P_i}) = 0$ provides a feasible open-loop controller, and $J_\pi(\overline{y}) = \overline{y}^\top R\overline{y}, R \in \mathbb{R}^{(4\tau^*)\times(4\tau^*)}$ provides an open-loop controller policy that minimizes the control effort. Let $\overline{\pi}_{\tau^*}^{P_i,\mathrm{opt}}(\overline{x}_{P_i}[\tau])$ denote the optimal solution to (23).

We summarize our solution to Problem 1 in Algorithm 1. We first obtain the FSRPD associated with the threat model for all $\tau \in \mathbb{N}_{[1,T]}$. We then solve (21) and (23) for each pursuer individually. Using the computed $\overline{x}^*_{P_i}[\tau]$, we evaluate the team state $\overline{x}^*_{\mathrm{team}}[\tau]$ via (15), and the optimal team catch probability and the optimal time to intercept via (16) and (17).

Note that depending on the threat and the pursuers dynamics, it is possible that (19) might yield capture locations that have low probability of capture. We remedy this by incorporating real-time sensing information using a receding horizon control as discussed in Section III-C. Also, one can enforce more coordination between the pursuers by changing the team catch probability (16) at the potential loss of structure exploited in Algorithm 1.

Algorithm 1 Multiple pursuer-based threat intercept

```
Input: Initial threat location $\overline{x}_G[0]$, probability density function $\psi_{\boldsymbol{w}}(\cdot)$, initial pursuer locations $\overline{x}_{P_i}[0]$, admissible control sets $\overline{\mathcal{M}}_\tau^{P_i}$ and $\overline{\mathcal{M}}_\tau^{G}$, threat dynamics $A_G, B_G$, pursuer dynamics $A_P, B_P$, and asset location $\overline{x}_a$
Output: Optimal team catch probability $\mathrm{TeamCatchPr}\left(\tau^*, \overline{x}^*_{\mathrm{team}}[\tau^*]; \overline{x}_G[0], \overline{\pi}_T^{G,\mathrm{opt}}\right)$, optimal time to intercept $\tau^*$, optimal team state $\overline{x}^*_{\mathrm{team}}[\tau^*]$, and optimal open-loop controllers for each pursuer $\overline{\pi}_{\tau^*}^{P_i}(\overline{x}_{P_i}[0])$
Compute $\overline{\pi}_T^{G,\mathrm{opt}}(\overline{x}_G[0])$ from (12)
for $\tau = 0, 1, 2, ...., T$ do
    Compute $\psi_{\boldsymbol{x}_G}(\cdot; \tau, \overline{x}_G[0], \overline{\pi}_T^{G,\mathrm{opt}})$ from (18)
    for all $P_i$ do
        Compute $\mathrm{CatchPr}\left(\tau, \overline{x}^*_{P_i}[\tau]; \overline{x}_G[0], \overline{\pi}_T^{G,\mathrm{opt}}\right)$ from (21)
        Compute $\overline{\pi}_\tau^{P_i}(\overline{x}_{P_i}[0])$ from (23)
    end for
    Compute $\overline{x}^*_{\mathrm{team}}[\tau]$ from (15)
    Compute $\mathrm{TeamCatchPr}\left(\tau, \overline{x}^*_{\mathrm{team}}[\tau]; \overline{x}_G[0], \overline{\pi}_T^{G,\mathrm{opt}}\right)$ from (16)
end for
Compute $\mathrm{TeamCatchPr}\left(\tau^*, \overline{x}^*_{\mathrm{team}}[\tau^*]; \overline{x}_G[0], \overline{\pi}_T^{G,\mathrm{opt}}\right)$ and $\tau^*$ from (17)
return $\tau^*$, $\overline{x}^*_{\mathrm{team}}[\tau^*]$, $\overline{\pi}_{\tau^*}^{P_i}$, and $\mathrm{TeamCatchPr}\left(\tau^*, \overline{x}^*_{\mathrm{team}}[\tau^*]; \overline{x}_G[0], \overline{\pi}_T^{G,\mathrm{opt}}\right)$
```

C. Receding horizon control

In Section III-B, we proposed open-loop controllers for the pursuers for the multiple pursuer-based threat interception, *i.e*, control actions that only depend on the initial states of the pursuers and the threat. To address Problem 2, we now incorporate real-time state information in a receding horizon control framework.

The receding horizon control-based planner is described in Figure 3. At every instant $\tau \in \mathbb{N}$, we use the current state information of the threat and the pursuers to solve for the optimal intercept and the open-loop controllers for each pursuer $\overline{\pi}_{\tau^*}^{P_i,\mathrm{opt}}(\overline{x}_{P_i}[\tau])$ using Algorithm 1. We apply the first element of $\overline{\pi}_{\tau^*}^{P_i,\mathrm{opt}}(\overline{x}_{P_i}[\tau])$ and then repeat the process.

We anticipate that in most circumstances, the receding horizon controller will have a higher likelihood of successful interception than the open-loop controller, as demonstrated in Section IV. This is because we incorporate real-time state information and re-solve the optimization problems (21) and (23) at every instant. However, if, for example, variance is low enough and the optimal open-loop controller happens to coincide with a globally optimal state for intercept, the receding horizon controller may actually perform worse. This is because the inherent tracking of the threat done in real-time by the receding horizon control framework may force the pursuers to converge to a locally optimal state for intercept. A rigorous comparison of these controllers is the subject of ongoing work, via a Martingale approach. An additional advantage of the receding horizon control approach is that real-time state information and re-planning of the pursuer trajectory should also provide some robustness to the uncertainty in the threat's model.

IV. Experimental validation

A. Experimental setup

We implemented the proposed solution to multiple pursuer-based threat interception in the MAHRES lab at UNM in a 3m × 3m × 3m environment. We used QAV 250 frame-based quadrotors with Odroid XU-4 micro processors running Robotic Operating System (ROS) for on-board processing, and Pixracer PX4 for attitude and position control. The Odroid XU-4 on each quadrotor communicates to a workstation, an Intel Xeon CPU with 1.6 GHz clock rate and 16GB RAM, which servers as the ROS Master. The workstation implements Algorithm 1 in MATLAB and computes the optimal state for intercept using Gurobi [18]. We transform the optimal state for intercept to a desired trajectory for each pursuer by first constructing a series of waypoints using (23), and then fitting polynomials $p_i(t)$ of degree N_d over these waypoints (Figure 3). The coefficients of these polynomials are published to a ROS topic on the ROS master using MATLAB's Robotic Systems Toolbox. The pursuers subscribe to this topic and track the desired trajectory using their on-board processors.

We choose $T_s = 0.5$ seconds, $T = 10$, $N_p = 1$, $N_d = 5$, $\mathcal{U}_P = [-5, 5]^4$, $\mathcal{U}_G = [-5, 5]^4$, $\overline{\mu}_{\boldsymbol{w}} = \overline{0}_{4\times 1}$, $\Sigma_{\boldsymbol{w}} = 0.05 I_4$, $q = 10$, and the catch set as $\mathrm{Box}(\overline{0}_{12\times 1}, 0.25)$. We restrict each quadrotor to fly at a fixed altitude, since the linearized model of the quadrotor does not permit much displacement along z-direction (see Figure 2b), and it ensures that the quadrotors do not collide.

B. Experiment 1: Robustness to threat's model uncertainty

In this experiment, we controlled the threat manually to produce a real-time trajectory that is not consistent with the threat model described in Section III-A. Figures 5 and 6 show that the pursuer intercepts the threat despite the model mismatch, demonstrating the robustness provided by the receding horizon control framework. Figure 4 shows the optimal probability, optimal time steps to intercept, and the computational time during the experiment run. As expected, the probability of intercept increases as the pursuer comes closer to the threat. An update in the desired trajectory occurs on average every 0.33 seconds. Due to the centralized nature of the algorithm, the computation time will increase linearly with N_p. The significant drop in the computation time at the last update is because the threat was intercepted, implying no additional computation was required.

C. Experiment 2: Open-loop vs receding horizon control

In this experiment, the threat moves along one of the trajectories prescribed by the stochastic model discussed in Section III-A. We compare the performance between the open-loop (Figures 7(a)–(d)) and receding horizon control (Figures 7(e)–(h)) strategies. Figure 8 shows that, due to incorporation of the real-time sensing data, the pursuer using the receding horizon control strategy intercepts the threat 0.9 seconds earlier than the pursuer using an open-loop controller.

Videos of the experiments are posted at `https://youtu.be/eFGg7U7gEQw` and `https://youtu.be/H0BZrk9Goxg` respectively.

V. Conclusion and future work

We proposed a solution to the problem of the multiple pursuer-based intercept of a threat with stochastic dynamics and have validated the results on an experimental testbed. Using existing results from reachability, we demonstrated that the optimal time and state for intercept can be obtained by solving a convex optimization problem that can be approximated via a quadratic program. We obtain corresponding open-loop controllers via an additional linear or quadratic problem. We implemented the proposed solution to multiple pursuer-based intercept in a receding horizon control framework to obtain better intercept probabilities.

Future directions include exploring the relationship between the intercept likelihoods provided by the receding horizon control and the open-loop control, and variations of team catch probability that incorporates more complex forms of coordination.

References

[1] R. Vidal, O. Shakernia, H. Kim, D. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 662–669, Oct. 2002.

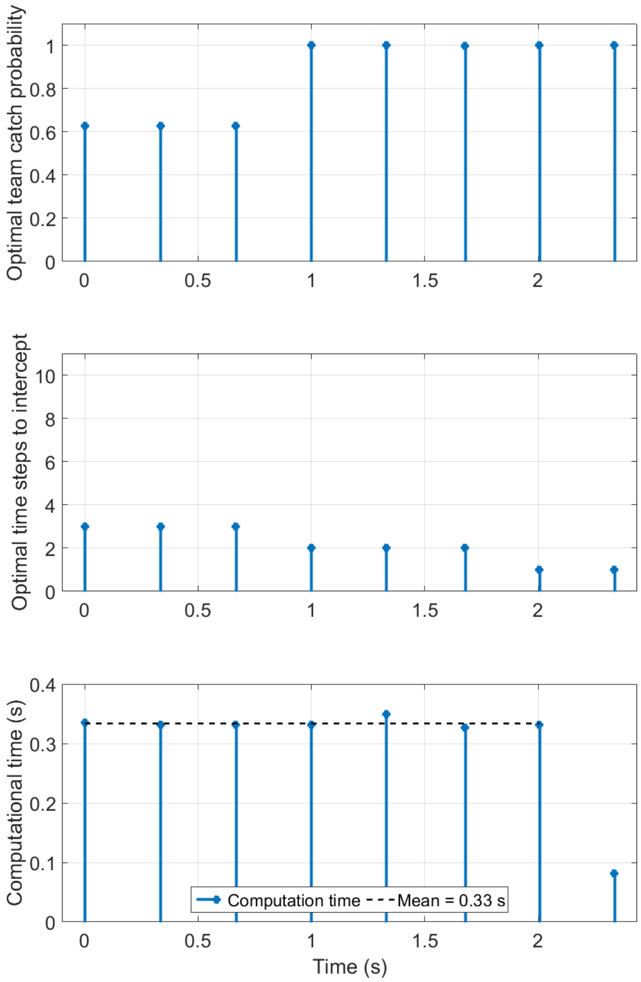


Fig. 4: Optimal team catch probability $\mathrm{TeamCatchPr}(\tau^*, \overline{x}_{\mathrm{team}}[\tau^*]; \overline{x}_G[0], \overline{\pi}_T^{G,\mathrm{opt}})$, time steps to intercept τ^*, and computational time at each update in Experiment 1.

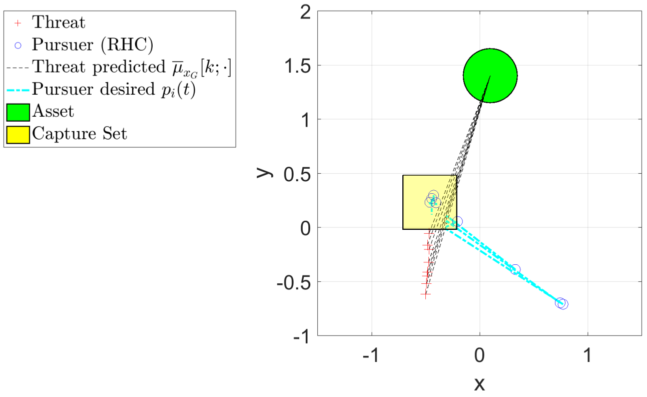


Fig. 5: Predicted threat $\overline{\mu}_{\boldsymbol{x}_G}\left[k; \overline{x}_G[0], \overline{\pi}_T^{G,\mathrm{opt}}\right]$ and the desired pursuer trajectory polynomial $p_i(t)$ in Experiment 1. The pursuer demonstrates robustness by intercepting a threat that follows a path inconsistent with the predictions from the threat model.

[2] O. Bokanowski, N. Forcadel, and H. Zidani, “Reachability and Minimal Times for State Constrained Nonlinear Problems without Any Controllability Assumption,” *SIAM Journal on Control and Optimization*, vol. 48, no. 7, pp. 4292–4316, Jan. 2010.

[3] I. Mitchell, A. Bayen, and C. Tomlin, “A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games,” *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.

[4] H. Huang, W. Zhang, J. Ding, D. Stipanović, and C. Tomlin, “Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers,” in *IEEE Conf. Dec. & Ctrl and Eur. Ctrl Conf.*, 2011, pp. 4835–4840.

[5] S. Summers and J. Lygeros, “Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem,” *Automatica*, vol. 46, no. 12, pp. 1951–1961, 2010.

[6] N. Malone, K. Lesser, M. Oishi, and L. Tapia, “Stochastic reachability based motion planning for multiple moving obstacle avoidance,” in *Proc. of the 17th International Conference on Hybrid Systems: Computation and Control*. Berlin, Germany: ACM, 2014, pp. 51–60.

[7] C. F. Chung, T. Furukawa, and A. H. Goktogan, “Coordinated control for capturing a highly maneuverable evader using forward reachable sets,” in *Proc. of IEEE International Conference on Robotics and Automation*, 2006, pp. 1336–1341.

[8] C. F. Chung and T. Furukawa, “A reachability-based strategy for the time-optimal control of autonomous pursuers,” *Engineering Optimization*, vol. 40, no. 1, pp. 67–93, 2008.

[9] G. Hollinger, S. Singh, J. Djugash, and A. Kehagias, “Efficient multi-robot search for a moving target,” *The International Journal of Robotics Research*, vol. 28, no. 2, pp. 201–219, 2009.

[10] A. P. Vinod, B. HomChaudhuri, and M. M. K. Oishi, “Forward stochastic reachability analysis for uncontrolled linear systems using Fourier Transforms,” in *Proc. of the 20th International conference on Hybrid Systems: Computation and Control*, Pittsburg, PA, USA, 2017.

[11] A. P. Vinod and M. M. K. Oishi, “Scalable underapproximation for stochastic reach-avoid problem for high-dimensional LTI systems using Fourier transforms,” *IEEE Control Systems Letters*, vol. 1, pp. 316 – 321, October, 2017.

[12] R. Mahony, V. Kumar, and P. Corke, “Multirotor aerial vehicles,” *IEEE Robotics and Automation magazine*, vol. 20, no. 32, 2012.

[13] J. A. Gubner, *Probability and random processes for electrical and computer engineers*. New York; Cambridge: Cambridge University Press, 2006.

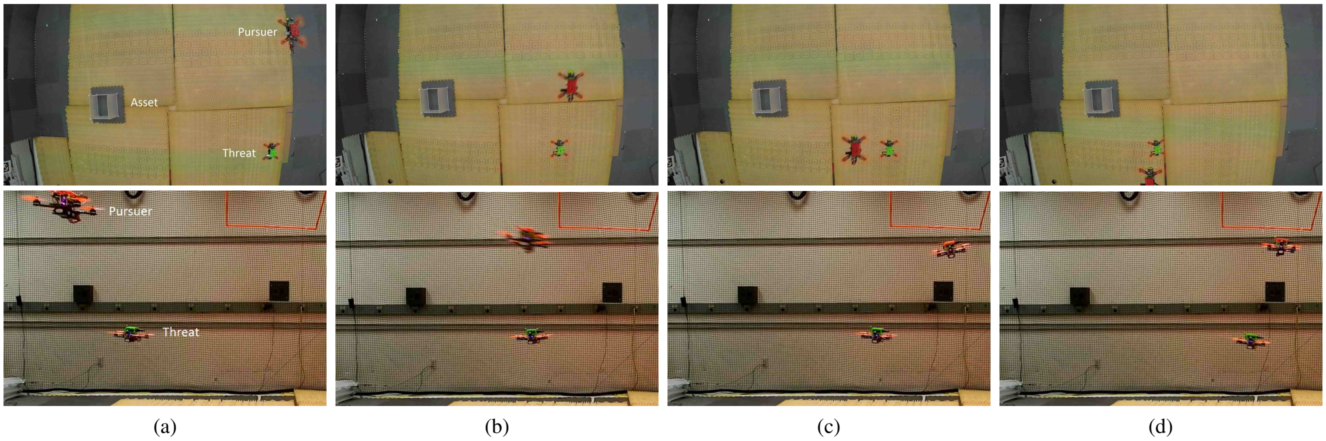
[14] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, “Multi-Parametric Toolbox 3.0,” in *Proc. of the European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510, http://people.ee.ethz.ch/~mpt/3/.

[15] A. A. Kurzhanskiy and P. Varaiya, “Ellipsoidal toolbox,” EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2006-46, 2006.

[16] S. Dharmadhikari and K. Joag-Dev, *Unimodality, convexity, and applications*. Elsevier, 1988.

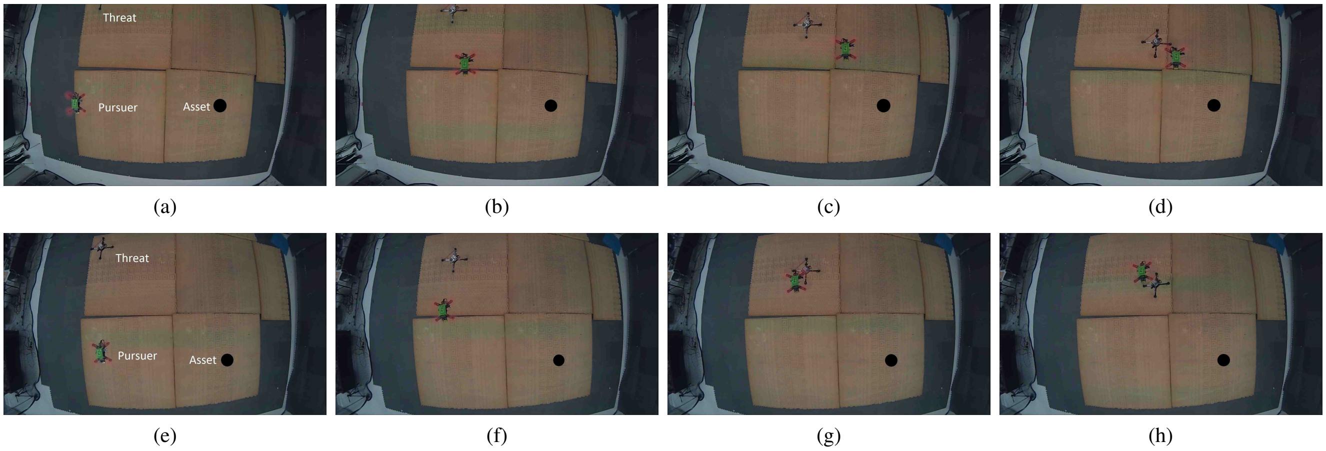
[17] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge, UK ; New York: Cambridge University Press, 2004.

[18] I. Gurobi Optimization, “Gurobi optimizer reference manual,” 2016. [Online]. Available: http://www.gurobi.com



(a) (b) (c) (d)

Fig. 6: Overhead and sideview snapshots of Experiment 1 with receding horizon control. (a) Start of the experiment (b) Pursuer moving towards the optimal location of intercept that was computed online (c) Pursuer at the optimal location of intercept (d) Successful intercept with $\boldsymbol{x}_G[\tau] \in \mathrm{CatchSet}(\overline{x}_{P_i}[\tau])$. See Experiment 1 video at https://youtu.be/eFGg7U7gEQw.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 7: Overhead time-stamped snapshots of Experiment 2 with open-loop control (OLC) (a)–(d) and receding horizon control (RHC) (e)–(h). The black circle marks the location of the asset. The snapshots (a), (d) were taken at 0.0 s; (b), (f) at 0.8 s; (c), (g) at 1.4 s; and, (h), (d) at 2.3 s. We see that the RHC-based pursuer intercepts the threat 0.9 s before the OLC-based pursuer. See Experiment 2 video at https://youtu.be/H0BZrk9Goxg.

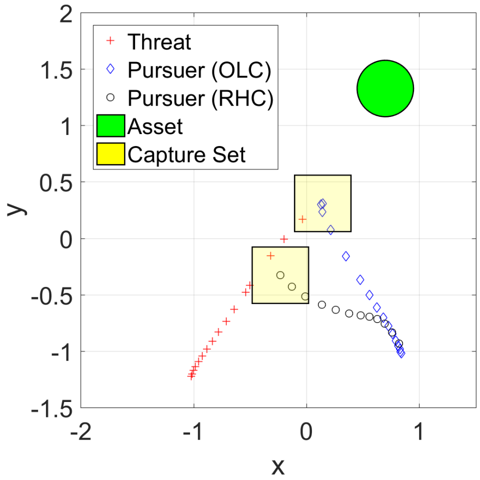


Fig. 8: Comparison of open-loop control (OLC) with receding horizon control (RHC) in Experiment 2.