

Hunting Drones with Other Drones: Tracking a Moving Radio Target

Louis Dressel and Mykel J. Kochenderfer

Abstract— Unauthorized drone flights near aircraft, airports, and emergency operations compromise the safety of passengers and bystanders. A detection system that can quickly find and track drones could help mitigate the risk of unauthorized drone flights. In this work, we show how a consumer drone outfitted with antennas and commodity radios can autonomously localize another drone by its telemetry radio emissions. We show how a non-myopic planner improves tracking performance over traditionally used greedy, one-step planners. Improved tracking is validated with simulations and the system is demonstrated with real drones in flight tests.

I. INTRODUCTION

Drones, also called unmanned aerial vehicles (UAVs), are pilotless aircraft. They are often controlled by operators on the ground and relay telemetry data and video back to the operator. Consumer drones are typically small, inexpensive, and can carry high-definition cameras, making them popular among hobbyists and researchers. For example, drones have been used to photograph whales [1] and volcanoes [2].

Unfortunately, some users take video of interesting yet dangerous locations such as airports and wildfires. During a three-month span in 2017, the Federal Aviation Administration (FAA) recorded 634 sightings of unmanned aircraft operating near airplanes, helicopters, and airports [3]. In 2017, the UK experienced 92 “Airprox” events in which drones compromised the safety of manned aircraft [4]. The FAA has had to warn pilots not to fly near wildfires, because it forces firefighting aircraft to land [5]. While it is often illegal to fly near airports, aircraft, and emergency operations, some drone pilots are unaware of the laws or ignore them.

Dangerous and illegal drone operations could be curbed by rapidly localizing and tracking the offending drone. One way to track a drone is to home into the radio telemetry it sends back to its operator. This telemetry might consist of drone information such as speed, location, battery status, or even live video. By default, many consumer drones send some sort of telemetry back to their operators, and relaying video in real-time is part of their appeal. Therefore, radio tracking can be a useful localization technique that complements vision-based drone trackers. While a technically competent adversary could avoid detection by programming an autonomous path and maintaining radio silence, radio tracking can deter many consumer drone operators.

When equipped with antennas and radio receivers, a drone can be used to track another drone by its telemetry

emissions. A drone-based tracker is mobile and can track unauthorized drones for longer distances than stationary trackers. The mobility and low cost of consumer drones have made drone-based radio localization an active research topic, and drones have recently localized radio-tagged wildlife [6], GPS jammers [7], and cell phones [8]. However, most existing work assumes radio targets are stationary, leading to algorithm designs that are ill-suited for hunting moving targets.

In this paper, we use a seeker drone to localize a target drone by the emissions of its telemetry radio. The seeker drone carries two antennas and two commodity radios. By successively comparing the signal strengths measured at the antennas, the seeker drone can get a good estimate of the target drone’s location. The seeker drone plans its path—onboard and in real-time—so that it makes informative measurements while avoiding collisions with the target drone. This information-theoretic, belief-space planning uses Monte Carlo tree search. Simulations show that this non-myopic planner outperforms planners that only plan one timestep into the future. Finally, we evaluate our system in flight tests, localizing another drone by its telemetry radio. Summarizing, our key contributions are:

- 1) We analyze the performance improvement when using non-myopic planning for a drone-based drone tracker.
- 2) We demonstrate the drone-based tracking of a moving drone in flight.

II. RELATED WORK

Detection and tracking form a critical layer in a “defense in depth” approach to countering drones [9]. Cameras offer an intuitive solution for tracking drones, but vision-based solutions are not perfect. In particular, it can be difficult to differentiate drones and birds, especially when birds glide [10]. Solutions to this problem include using audio to listen for drone motors [11] or using neural networks to robustly classify drones from images [12]. Other work uses infrared imaging to improve performance in low-light situations [13].

Radio frequency (RF) sensors can detect and analyze the signals a drone sends to its operator. Because these sensors are not fooled by birds or environmental lighting, they complement visual sensors well. However, RF analysis is difficult, so most academic research using RF signals focuses on detection but not tracking [14], [15].

In contrast, commercial products such as ARDRONIS, DroneShield, and Dedrone RF-300 use more complex hardware capable of direction finding [15]. This hardware allows some models to intercept drone video and display what the drone operator sees; other models feature databases of drone

This work was supported by NSF grant DGE-114747.

The authors are with the Aeronautics and Astronautics department at Stanford University, Stanford, CA, 94305 USA. Email: {dressel, mykel}@stanford.edu

RF signatures to identify particular models. These products are large and relatively immobile, making it difficult to track a drone that simply transits a restricted area.

Putting RF sensors on a seeker drone allows measurements to be taken from many locations. This measurement diversity allows for tracking even with simple RF hardware. Further, the mobility of a seeker drone allows it to track moving targets for longer distances than ground-based trackers.

While there has been much research into drone-based radio localization, most work relies on assumptions that are inappropriate for tracking a moving drone. In some work, RF strength measurements are compared to a model that correlates signal strength with distance [16]–[18]. However, this method requires knowing the transmitted power, which is unlikely when tracking an anonymous, intruding drone.

A different technique relies on the maneuverability of a multirotor drone. The drone rotates in place, making measurements with a directional antenna. Each rotation yields a bearing estimate to the source location. The drone then moves to another location and makes another bearing estimate. This technique has been used to track stationary (or slow) radio-tagged wildlife [6] and stationary GPS jammers [7], [19]. Because each rotation takes about half a minute, this method cannot track fast-moving targets like drones unless several seeker drones work together [20].

Recent work incorporates commodity hardware and compares strength measurements made simultaneously on two antennas on the seeker drone [8], [21]. Each comparison yields a bearing-like measurement. While less informative than true bearing estimates, they are made much more quickly, leading to faster localization. The system in this paper is based on one of these works [8].

While these sensor systems are capable of tracking moving targets, they have only been used on stationary targets. As a result, prior work makes algorithmic decisions that are not appropriate for moving targets. For example, almost all drone-based radio localization uses some form of greedy entropy minimization [22]. In a greedy (also called myopic) approach, the seeker drone acts to minimize entropy (a measure of uncertainty) in the target’s location at the next timestep. These methods work well with stationary targets but may fail with a moving target. In order to make advantageous measurements, the seeker drone may need to plan many steps into the future and account for where the target may move.

Additionally, a moving target drone poses more risk to the seeker drone than a stationary target on the ground. The seeker drone should act to reduce its risk of colliding with the target. This feature has not appeared in prior work, which has focused on stationary ground targets. In this work, a non-myopic planner reduces uncertainty in the target’s location while reducing the risk of collision.

III. SYSTEM OVERVIEW

In this work, a seeker drone tracks the position of a target drone by capturing emissions of the target’s telemetry radio. The target and seeker drones can be seen in Figure 1.



Fig. 1. M-100 seeker drone (left) and F550 target drone (right). The Moxon antennas sit under the white foam boards on the seeker drone. The target drone’s telemetry antenna sits between the two white arms pointing left.

A. Target Drone

The target drone is a DJI F550 hexcopter. It is controlled with a Pixhawk flight controller that logs its GPS positions. The target drone also carries a SiK telemetry radio, which can be seen in Figure 2. The radio transmits with 0.1 W and uses frequency hopping spread spectrum (FHSS) technology to radiate in 50 bins between 902 MHz and 928 MHz.

FHSS radios spread their radiation into many frequency bins over a wide frequency range, radiating in a single bin at a time. A receiver paired with the transmitter knows the bin ordering, but an eavesdropper does not and will have difficulty following the transmission. If the eavesdropper has a narrow bandwidth, it will miss much of the transmission.

The transmission power and hopping sequence of the target drone’s telemetry radio are unknown to the seeker drone. The only information available to the seeker drone is that the radio operates in the commonly used 915 MHz ISM band.

B. Seeker Drone

The seeker drone is a DJI Matrice 100 (M-100) quadcopter. The M-100 is 2.4 kg and has a maximum takeoff capacity of 3.4 kg, allowing for a 1 kg payload. When carrying a payload, the M-100 has a flight time of about 15 minutes. The M-100 uses GPS to estimate its position and has a built-in flight controller that maintains stability and provides low-level motor commands.

The seeker drone carries a DJI Manifold computer, which has 2 GB RAM and four ARM Cortex-A15 cores. This onboard computer receives measurements from the sensor system and performs planning. A serial connection with the flight controller lets the Manifold receive drone position and provide velocity commands. The Manifold has a mass of 197 g. It can be seen atop the seeker drone in Figure 1.

C. Sensor System

The seeker drone carries a sensor system developed in prior work [8]. Signal strength is measured simultaneously by two Moxon antennas mounted on opposite sides of the seeker drone. One antenna points along the front of the drone, and the other points rearward. These antennas are directional and receive higher signal strength when pointing at a radio source. If the front-facing antenna measures a higher signal strength, the radio source likely lies in front of the seeker

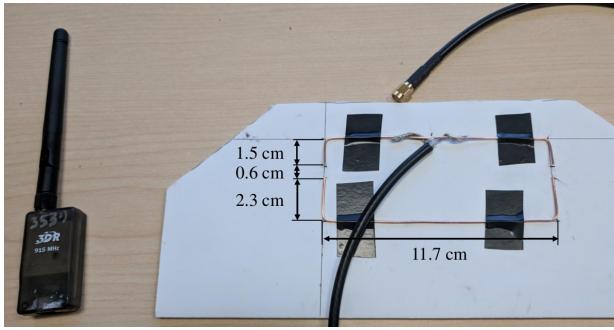


Fig. 2. Target drone telemetry radio (left) and one of the seeker drone's Moxon antennas during construction (right). The coaxial cable is soldered to the antenna before mounting the antenna on the drone.

drone; if the rear-facing antenna measures a higher strength, the radio source likely lies behind the drone.

The lightweight Moxon antennas are easily constructed by bending 18 AWG copper wire, taping it to styrofoam board, and soldering a coaxial cable to the wire. Figure 2 shows one antenna and its dimensions. Including its coaxial cable, each antenna has a mass of 45 g.

Each antenna is connected by its coaxial cable to a HackRF One software-defined radio. The HackRF is larger and more expensive than the RTL-SDR used in prior work (70 g and \$300 USD instead of 30 g and \$20 USD). The weight increase is negligible as the Matrice has enough payload margin to carry the extra 40 g per radio. The increased cost is more concerning, but these commodity radios are still significantly cheaper than research-oriented radios like the USRP.

The HackRF has two critical advantages that justify its increased weight and cost: its bandwidth is 20 MHz instead of 2.4 MHz, and its upper frequency limit is 6 GHz instead of 1.7 GHz. The large bandwidth means the HackRF can observe most of the telemetry radio's frequency hopping range from 902–928 MHz. In contrast, the RTL-SDR must constantly scan and retune to observe a meaningful chunk of this range, and it still might miss emissions as they hop from bin to bin. An upper frequency limit of 6 GHz allows the HackRF to receive in the 2.4 GHz and 5.8 GHz bands, which are commonly used for drone video or control signals. While this work only tracks a telemetry radio at 915 MHz, future work might track drone emissions in these other bands.

Each HackRF radio plugs into a USB port on the Manifold computer. A HackRF can be seen mounted to the front of the seeker drone in Figure 1.

The two Moxon antennas and two HackRF One radios have a combined mass of 230 g. Adding the mass of the Manifold computer yields 427 g, which is less than half the payload capacity of the seeker drone.

IV. MODELS AND FILTERING

Before tracking can be performed, the system described in the previous section must be expressed mathematically.

A. Dynamic Models

At time t , the seeker drone state is $x_t = [x_t^n, x_t^e, x_t^h]^\top$, where x_t^n and x_t^e are the drone's north and east coordinates

and x_t^h is the drone's heading, measured east of north. The drone state does not include altitude because it is constant. The antennas used in the sensor system are insensitive to the elevation angle to the radio target [8], so the seeker and target drones fly at constant altitudes.

The seeker drone state also lacks velocity information because the drone is assumed to operate with single integrator dynamics. This assumption is made because its flight controller accepts velocity commands. Of course, this assumption is only an approximation, as the drone takes time to accelerate to new velocities. However, this effect is likely small, and modeling drone dynamics is not the focus of this work.

The seeker drone state after applying the velocity control input u_t for duration Δt is

$$x_{t+\Delta t} = x_t + u_t \Delta t. \quad (1)$$

The seeker drone is assumed to know its state perfectly. It gets its position from GPS and its heading from a magnetometer. While these measurements have noise, any resulting uncertainty in the seeker's own state is negligible compared to uncertainty in the target's location.

At time t , the target drone location is $\theta_t = [\theta_t^n, \theta_t^e]^\top$, where θ_t^n and θ_t^e are the north and east coordinates. The target moves with constant speed $\dot{\theta}_t = [\dot{\theta}_t^n, \dot{\theta}_t^e]^\top$, so the target update is

$$\theta_{t+\Delta t} = \theta_t + \dot{\theta}_t \Delta t. \quad (2)$$

The assumption of constant target velocity is a limitation, but not a severe one. While the seeker drone knows the target drone moves at a constant velocity, it does not know what this velocity is; the seeker must still infer the velocity from its measurements. The target motion model can also be changed to incorporate any prior knowledge about the target's motion.

B. Sensor Model

Measured east of north, the bearing β_t from the seeker drone to the target drone is

$$\beta_t = \arctan\left(\frac{\theta_t^e - x_t^e}{\theta_t^n - x_t^n}\right). \quad (3)$$

The relative bearing, $\beta_t - x_t^h$, describes how the seeker is oriented with respect to the target. When the relative bearing is 0° , the front of the seeker drone points directly at the target drone; when the relative bearing is 180° , the rear of the seeker points directly at the target.

At time t , the seeker drone makes measurement $z_t \in \{0, 1\}$. If the front-facing antenna measures a stronger signal strength than the rear-facing antenna, $z_t = 1$; if the rear-facing antenna measures a stronger signal, $z_t = 0$. We expect $z_t = 1$ if the target drone lies in front of the seeker drone, and $z_t = 0$ if the target is behind it. However, when the target drone is off to one of the sides, we expect the antennas to measure roughly equal strengths, so we assign equal probability to either measurement. Further, prior work suggests there is a 10% chance the wrong measurement is made when the target lies in front or behind the seeker drone [8]. These wrong measurements are the result of noise in the radio

measurements. Summarizing, the probability of measuring $z_t = 1$ given the seeker and target states is:

$$P(z_t = 1 | x_t, \theta_t) = \begin{cases} 0.9, & \text{if } \beta_t - x_t^h \in [-60^\circ, 60^\circ] \\ 0.1, & \text{if } \beta_t - x_t^h \in [120^\circ, 240^\circ] \\ 0.5, & \text{otherwise.} \end{cases} \quad (4)$$

C. Particle Filter

The distribution over possible target locations is called the belief. A particle filter is used to represent the belief and track the location of the target drone, because the observations are semantic and inappropriate for Gaussian-based filters such as the Kalman filter or its variants.

In prior work, a discrete filter was used to track stationary targets [6]–[8]. In a discrete filter, the search area is split into a grid. The weight assigned to each grid cell represents the probability that the target is in that cell. Discrete filters are simple to use and can represent non-Gaussian distributions.

However, the discrete filter update is less efficient when the target state is not stationary. If a discrete belief consists of N grid cells, the belief update has complexity $O(N^2)$ [23]. In contrast, a particle filter with N particles has complexity $O(N)$ if the sampling is done properly [24].

Therefore, we use a particle filter. The belief at time t , denoted b_t , is represented with a set of N particles. Each particle consists of a potential target location and velocity. A belief update uses the seeker drone state x_t and measurement z_t to update belief b_t to $b_{t+\Delta t}$.

The belief update consists of two steps. First, the position of each particle is updated using its velocity and the model in Eq. (2). A small amount of noise is added to the velocity and position to avoid particle deprivation. In the second step, each particle is weighted according to the probability that the measurement z_t is observed given the drone state and the particle's position (Eq. (4)). Particles are then sampled according to these weights using an efficient low-variance resampler [24]. The set of N sampled particles is $b_{t+\Delta t}$.

V. TRACKING ALGORITHM

Once the seeker drone makes a measurement and updates its belief, it selects a control input. This planning is performed by the seeker drone's onboard computer. The planning algorithm uses the Markov decision process (MDP) framework.

A. Markov Decision Processes

An MDP can be defined with a state space \mathcal{S} , a control space \mathcal{U} , a cost function c , a generative model G , and a timestep horizon T . The model G generates the state at the next timestep, $s_{t+\Delta t} \in \mathcal{S}$, given the current state $s_t \in \mathcal{S}$ and control input $u_t \in \mathcal{U}$. This model can be stochastic as MDPs allow transition uncertainty, so that $s_{t+\Delta t} \sim G(s_t, u_t)$.

A policy $\pi : \mathcal{S} \rightarrow \mathcal{U}$ maps each state to an action. The solution to an MDP is an optimal policy π^* that minimizes the expected total cost to the horizon:

$$\pi^*(s_t) = \operatorname{argmin}_{u_t} \mathbb{E} \sum_{\tau=1}^T c(s_{t+\tau\Delta t}). \quad (5)$$

The expectation accounts for transition uncertainty.

In contrast, a greedy solution minimizes the expected cost at the next timestep:

$$\pi^g(s_t) = \operatorname{argmin}_{u_t} \mathbb{E} c(s_{t+\Delta t}). \quad (6)$$

Greedy policies are generally suboptimal as they value short-term gain at the expense of long-term optimality, but they are easy to implement and computationally inexpensive, so they have been used extensively for drone-based radio localization [6], [8], [21], [22]. However, all of these works assume the target is stationary.

B. Formulation

A traditional formulation of the tracking problem folds the seeker and target drone states into an overall system state [25]. Because the target drone state is unknown, this formulation is actually a partially observable MDP (POMDP). While there has been extensive work in solving POMDPs, they have a critical drawback in localization problems.

The classic definition of a POMDP requires that the cost function be defined in terms of the state. However, a belief-dependent reward often makes sense for tracking, where the goal is to have a belief with low uncertainty, leading to accurate estimates. One way to address this problem is to define an equivalent state-dependent cost; such a cost function might reward the seeker for reaching the target's state [25]. This surrogate cost function encourages the seeker to take information-gathering actions and learn the target state. But in our problem, the seeker avoids the target drone, so there is no obvious surrogate cost. Rewarding the seeker for staying away from the target encourages the seeker to gather only as much information as needed to avoid collisions.

It is possible to modify classic POMDP solvers to handle belief-dependent rewards, but these offline methods are slow even with coarse discretizations [26], [27]. In this work, we convert the POMDP into a belief-MDP, which is an MDP where a belief is part of the state. While the target state is unknown, the belief over possible target states is known. The state at time t is $s_t = (b_t, x_t)$.

The control space is a discrete set of velocity commands that can be given to the seeker drone. Given u_t and s_t , the components of the next state $s_{t+\Delta t} = (b_{t+\Delta t}, x_{t+\Delta t})$ can be obtained with the particle filter update and the seeker drone state update. This update is stochastic because noise in the sensor model affects the resulting belief.

The cost function should encourage the seeker to make measurements that lead to good target estimates while keeping it a safe distance from the target drone. Good target estimates are more likely if there is low uncertainty in the belief. Because the belief is part of the state, the seeker can be penalized when the belief uncertainty is large. The seeker is penalized for near-collisions, which occur if $\|x_t - \theta_t\| < d$, where d is a distance threshold. The following cost function penalizes belief entropy and near-collisions:

$$c(s_t) = H(b_t) + \lambda \mathbb{E}_{b_t} \mathbf{1}(\|x_t - \theta_t\| < d), \quad (7)$$

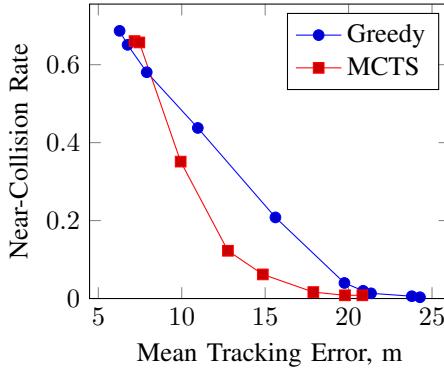


Fig. 3. MCTS outperforms the classic greedy strategy in avoiding collisions and minimizing tracking error, except when the collision penalty is small.

where $H(b_t)$ is the entropy of belief b_t , $\mathbb{1}$ is an indicator function that equals 1 if its argument is true and 0 otherwise, and the weight λ encodes the tradeoff between tracking and collision avoidance. A higher value of λ represents a higher penalty on near-collisions. The collision penalty is the expectation over all particles in the current belief.

Only the particle positions are used when computing belief entropy, capturing position uncertainty. To compute entropy from the particle filter, the particles are binned into M grid cells. The resulting discrete distribution is denoted \tilde{b}_t . Entropy is computed with $H(b_t) = -\sum_{i=1}^M \tilde{b}_t[i] \log \tilde{b}_t[i]$, where $\tilde{b}_t[i]$ is the proportion of particles in bin i .

C. Solution Method

To solve the MDP, we use the UCT variant of Monte Carlo tree search (MCTS) [28]. As its name implies, MCTS generates a tree from the current state s_t by running simulations to evaluate the cumulative cost of different control inputs. After simulating, the lowest-cost control input is selected.

A drawback of using MCTS for belief-MDPs is that each simulation step requires a belief update, which can be computationally expensive [29]. One solution is to use fewer particles, but this can lead to poor target estimation. Our solution is to downsample the particle filters before running MCTS to generate a control input. The seeker maintains the higher-fidelity belief for target estimation, but uses the downsampled belief for efficient planning.

D. Simulations

The planner is validated with simulations. The near-collision threshold is $d = 15$ m. The timestep duration is $\Delta t = 1$ s, after which a new measurement is made and a new control input is generated. The seeker drone can travel at 5 m/s and rotate at 15 °/s. The target drone starts in one corner of a 200 m × 200 m search area and travels across it at 1.7 m/s. The particle filter has 8000 particles and is initialized with random positions and velocities. For MCTS, these beliefs are downsampled to 200 particles before planning the next control input, and 1000 simulations with a timestep horizon of $T = 10$ steps are used to generate the next action.

The value of λ is varied for both the greedy and MCTS methods. For each value of λ , we run 1000 simulations of

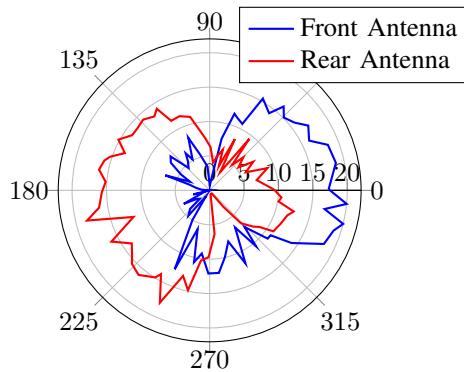


Fig. 4. Relative strength values as a function of relative bearing, measured by seeker drone with target drone 68 m away on the ground.

80 timesteps and log the resulting near-collision rate and mean tracking error. The near-collision rate is the proportion of timesteps that a near-collision has occurred. The mean tracking error is the mean distance per timestep between the particle filter position mean and the true target drone position.

The results are shown in Figure 3. MCTS outperforms the greedy strategy for most values of λ ; for the same near-collision rate, MCTS can offer a tracking error reduction of about 5 m, which is often a reduction of over 20%. The greedy method takes 0.11 s to select an action, while MCTS takes 0.29 s. This delay is acceptable if the sampling rate is 1 Hz, and MCTS can be made faster by reducing the number of simulations or particles in the downsampled belief.

Pareto dominance eludes MCTS because it performs worse when λ is small. One explanation is that the optimal policy is less complicated when near-collisions are not penalized. Both the greedy and MCTS policies lead the seeker drone to fly close to the target drone, where the best measurements are made. No long-term planning is needed as the seeker stays close to the target. Instead, small adjustments are made in the vicinity of the target to get the most information. The greedy method can make more efficient adjustments because it plans with the particle set used for localization. In contrast, the MCTS method plans with the lower-fidelity particle set and must estimate belief transition probabilities from transitions observed in its simulations. Therefore, its adjustments in the vicinity of the target are worse.

VI. FLIGHT TESTS

A. Validating the Measurement Model

To validate the measurement model with the HackRF and FHSS telemetry radios, the seeker drone rotated in place and measured the signal strength at each antenna while the target drone radiated telemetry data from the ground. Ten rotations were made, with the distances between the seeker and target ranging from 13.8 m to 116 m. The strengths measured during one rotation can be seen in Figure 4.

During these rotations, 204 measurements were made when the relative bearing was within $\pm 60^\circ$. In 199 of these, the front measurement was stronger. When the relative bearing was between 120° and 240° , the rear antenna measurement

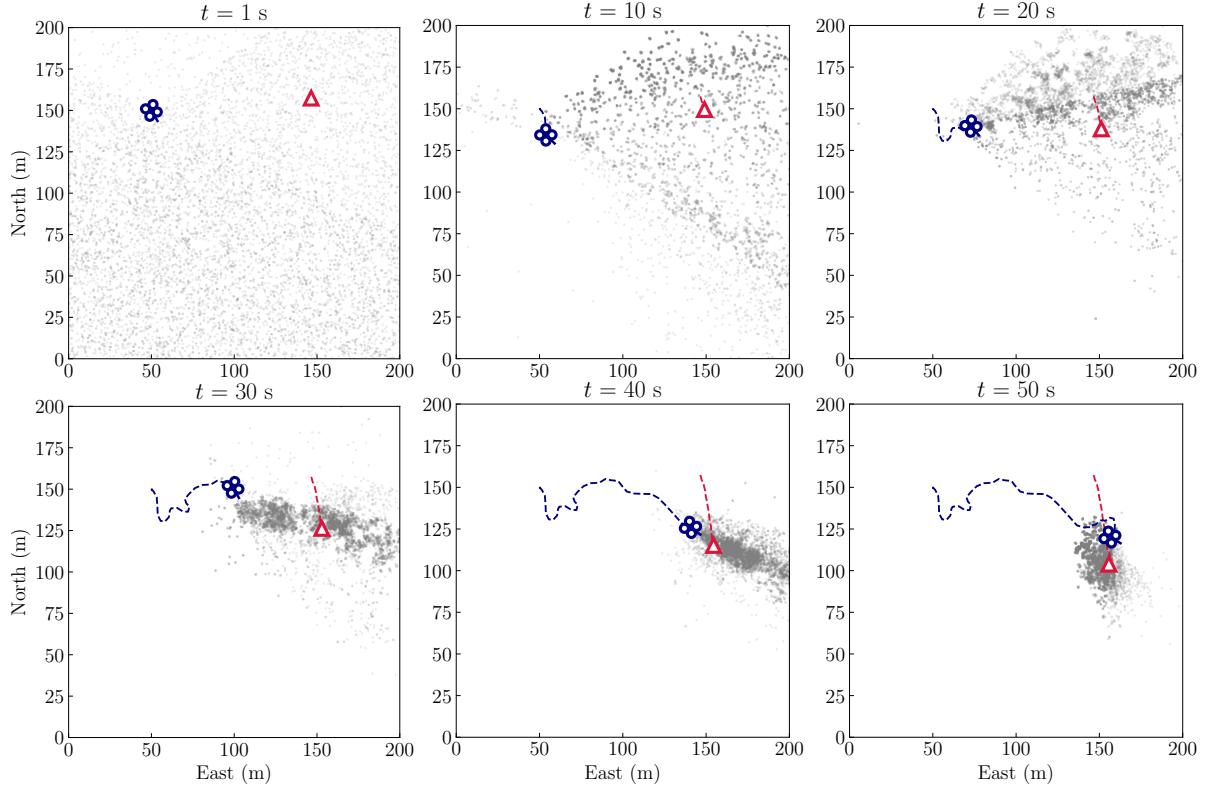


Fig. 5. Flight test: The seeker drone tracks the target drone (triangle) with a mean tracking error of 10.8 m. The seeker successfully avoids near-collisions and never comes within 15 m of the target.

was stronger in 195 of 204 total measurements. For both cases, the error is about 5%, less than the 10% predicted by the model in Eq. (4). When the target was to the side of the seeker, the front measurement was stronger in 106 of 210 measurements, roughly the 50% predicted by the model.

B. Tracking Trials

The seeker tracked the target drone in 20 flight tests. The target drone either flew or was carried along a straight path. In either case, the target drone's GPS positions were logged, and no significant difference in tracking performance was found between walking or flying the target drone. The seeker was limited to 5 m/s and 15 °/s. Measurements were collected and new control inputs were generated at 1 Hz. The trajectories from one flight test can be seen in Figure 5.

The mean tracking error was 38.0 m, which is larger than simulated results in Figure 3. The collision penalty λ was varied across flights; for five flights, λ was set to zero, resulting in a mean error of 27.7 m. This drastic reduction shows how much more information is gathered when the seeker can get close to the target.

Unmodeled noise can help explain the reduced accuracy in flight tests. Of the 272 measurements made with the relative bearing between 120° and 240°, the front measurement was stronger in 46, for an error rate of 16.9%, instead of the roughly 5% found when the drone rotated in place. When the target was to the side of the seeker, 604 of the 992

measurements indicated the front measurement was stronger, yielding a probability of 60.9%, instead of the 50% expected.

One likely source of noise is the coarse discretization of the sensor model. It predicts either measurement is equally likely between the relative bearings of 60° and 120°. While this may be true when averaging measurements over the entire range, the front measurement is more likely to be stronger at 61°, and the rear measurement is more likely to be stronger at 119°. While rotating in place averages these effects out, a less uniform trajectory will not.

VII. CONCLUSION

This work validates the concept of drone-based tracking of another drone by its telemetry radio. Simulations show that improved localization performance can be achieved with non-myopic planning. Flight tests serve as a proof-of-concept, although results indicate there is unmodeled noise. A more complex sensor model may reduce this noise. Likewise, a more complex planner may lead to even further improvement.

A critical area of future work is robustness to other transmissions in the same frequency range—including those from other drones. This sensor system only works when tracking a single drone. Using drones to track multiple transmitters with unknown transmit power is an open problem [8]. The problem is particularly pressing when localizing drones because their telemetry and video systems often use unlicensed ISM bands, which are used by many commercial products.

REFERENCES

- [1] V. Pirotta, A. Smith, M. Ostrowski, D. Russell, I. D. Jonsen, A. Grech, and R. Harcourt, "An economical custom-built drone for assessing whale health," *Frontiers in Marine Science*, vol. 4, p. 425, 2017.
- [2] S. American, *Drones peer inside a volcano*, Accessed: 2018-08-21, Nov. 2017. [Online]. Available: <https://www.scientificamerican.com/article/drones-peer-inside-a-volcano/> (visited on 08/21/2018).
- [3] United States Department of Transportation, *UAS sightings report*, Accessed: 2018-05-08. [Online]. Available: https://www.faa.gov/uas/resources/uas_sightings_report/.
- [4] UK Airprox Board, *Current drone Airprox count and information*, Accessed: 2018-05-10. [Online]. Available: <https://www.airproxboard.org.uk/Topical-issues-and-themes/Drones/>.
- [5] FAA, *Wildfires and drones don't mix*, Accessed: 2018-08-21, 2017. [Online]. Available: https://www.faa.gov/uas/where_to_fly/airspace_restrictions/wildfires/ (visited on 08/21/2018).
- [6] O. M. Cliff, R. Fitch, S. Sukkarieh, D. L. Saunders, and R. Heinsohn, "Online localization of radio-tagged wildlife with an autonomous aerial robot system," in *Robotics: Science and Systems*, 2015.
- [7] A. Perkins, L. Dressel, S. Lo, and P. Enge, "Demonstration of UAV-based GPS jammer localization during a live interference exercise," in *Institute of Navigation (ION) GNSS+*, 2016.
- [8] L. Dressel and M. J. Kochenderfer, "Efficient and low-cost localization of radio signals with a multirotor UAV," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2018.
- [9] R. J. Wallace and J. M. Loffi, "Examining unmanned aerial system threats & defenses: A conceptual analysis," *International Journal of Aviation, Aeronautics, and Aerospace*, vol. 2, no. 4, p. 1, 2015.
- [10] S. R. Ganti and Y. Kim, "Implementation of detection and tracking mechanism for small UAS," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016.
- [11] S. Hu, G. H. Goldman, and C. C. Borel-Donohue, "Detection of unmanned aerial vehicles using a visible camera system," *Applied Optics*, vol. 56, no. 3, B214-B221, 2017.
- [12] A. Schumann, L. Sommer, J. Klatte, T. Schuchert, and J. Beyerer, "Deep cross-domain flying object classification for robust UAV detection," in *IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2017.
- [13] T. Müller, "Robust drone detection for day/night counter-UAV with static VIS and SWIR cameras," in *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR VIII*, 2017.
- [14] P. Nguyen, M. Ravindranatha, A. Nguyen, R. Han, and T. Vu, "Investigating cost-effective RF-based detection of drones," in *Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, 2016.
- [15] X. Shi, C. Yang, W. Xie, C. Liang, Z. Shi, and J. Chen, "Anti-drone system with multiple surveillance technologies: Architecture, implementation, and challenges," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 68–74, 2018.
- [16] A. Posch and S. Sukkarieh, "UAV based search for a radio tagged animal using particle filters," in *Australasian Conference on Robotics and Automation (ACRA)*, 2009.
- [17] F. Körner, R. Speck, A. H. Göktogan, and S. Sukkarieh, "Autonomous airborne wildlife tracking using radio signal strength," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.
- [18] P. Soriano, F. Caballero, and A. Ollero, "RF-based particle filter localization for wildlife tracking by using an UAV," in *International Symposium of Robotics*, 2009.
- [19] A. Perkins, L. Dressel, S. Lo, and P. Enge, "Antenna characterization for UAV based GPS jammer localization," in *Institute of Navigation (ION) GNSS+*, 2015.
- [20] H. Bayram, N. Stefan, K. S. Engin, and V. Isler, "Tracking wildlife with multiple UAVs: System design, safety and field experiments," in *IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, 2017.
- [21] L. Dressel and M. J. Kochenderfer, "Pseudo-bearing measurements for improved localization of radio sources with multirotor UAVs," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [22] G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "Distributed cooperative search using information-theoretic costs for particle filters, with quadrotor applications," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2006.
- [23] M. J. Kochenderfer, *Decision Making Under Uncertainty: Theory and Application*. MIT Press, 2015.
- [24] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
- [25] D. Hsu, W. S. Lee, and N. Rong, "A point-based POMDP planner for target tracking," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008.
- [26] M. Araya, O. Buffet, V. Thomas, and F. Charpillet, "A POMDP extension with belief-dependent rewards," in *Advances in Neural Information Processing Systems (NIPS)*, 2010.
- [27] L. Dressel and M. J. Kochenderfer, "Efficient decision-theoretic target localization," in *International Conference on Automated Planning and Scheduling (ICAPS)*, 2017.

- [28] L. Kocsis and C. Szepesvári, “Bandit based Monte-Carlo planning,” in *European Conference on Machine Learning (ECML)*, 2006.
- [29] Z. N. Sunberg and M. J. Kochenderfer, “Online algorithms for POMDPs with continuous state, action, and observation spaces,” in *International Conference on Automated Planning and Scheduling (ICAPS)*, 2018.