ASTRO: Autonomous, Sensing, and Tetherless netwoRked drOnes

Riccardo Petrolo Yingyan Lin Edward Knightly
ECE Department, Rice University, Houston, TX

ABSTRACT

We propose ASTRO, a drone network that realizes three key features: *(i)* networked drones that coordinate in autonomous flight via software defined radios, *(ii)* off-grid tetherless flight without requiring a ground control station or air-to-ground network, and *(iii)* on-board machine learning missions based on on-drone sensor data shared among drones. We implement ASTRO and present a suite of proof-of-concept experiments based on a mission in which a network of ASTRO drones must find and track a mobile spectrum cheater.

1. INTRODUCTION

In this paper, we present the design, implementation, and experimental evaluation of ASTRO, Autonomous, Sensing, and Tetherless netwoRked drOnes. ASTRO realizes the following three features not previously realized in a single design. First, in contrast to single-drone solutions [10, 17], ASTRO realizes networked drones that communicate and coordinate among themselves. They form a dynamic mesh and employ software defined radios (SDRs) to enable programmability and advanced communication and networking features. ASTRO exploits such capabilities to adapt carrier frequency in order to realize longer range as needed to maintain connectivity with other drones, at the potential cost of less bandwidth being available at lower frequencies. Likewise, ASTRO realizes adaptive network-layer routing implemented on-drone to ensure that drones can communicate with each other as required by the mission.

Second, ASTRO is tetherless, in that we do not employ ground control stations for sending and receiving control signals and/or data. This contrasts with existing systems which use either human or machine control from the ground [10, 12]. As a consequence of tetherless operation, we do not require any communication infrastructure, enabling flight in areas not served by Wi-Fi or cellular networks. Nonetheless, if network infrastructure is available, ASTRO can utilize it to report back mission results while the mission is in progress vs. after completion and landing.

Third, ASTRO realizes data-driven sensing missions via on-line (without prior training) light-weight on-drone machine learning. By data-driven, we refer to the drone's decisions and flight paths being adapted in real-time according to measured sensor data. Namely, in contrast to paths being a pre-defined search pattern or adapted by a human or ground control station, the drones themselves adapt the flight patterns according to sensor data and mission goals. This is realized via on-board machine learning driven by local sensor data, data shared by other drones, and mission objectives (e.g., to find and track a moving target).

To demonstrate ASTRO, we implement all of the aforementioned aspects of ASTRO and report on over 500 hours of test flights over 10 months. We deploy a scenario in which the mission objective is to find and track a mobile spectrum cheater. In particular, the cheater is a mobile device that is transmitting on a frequency for which it lacks regulatory permission. In an exemplary mission, all ASTRO drones are launched from a common location outside of the range of the cheater. Without any prior training data, the drones coordinate with an initial search phase in which they cover the largest possible area given their constraints of sensing capabilities, inter-drone connectivity, battery life time, etc. Once an ASTRO drone has identified the target, it requests that all other drones aid it for the greatest possible tracking accuracy, guided by ASTRO's machine learning methods. Our experiments show that ASTRO drones can collaborate to find and track a target within approximately 10 m, even if the target is mobile or placed among an angular several ton concrete slab surrounded by buildings.

Related work. ASTRO compliments work on autonomous flight for individual drones [6, 13] as our focus is on multi-drone sensing missions. Likewise, prior applications of drones to wireless sensing employed "war driving" in which drone flight patterns were not adapted to sensor data and employed only a single drone [10, 17]. Machine learning has been applied to wireless sensing including learning regression tree [1], support vector machines [18], neural networks [19], and k-nearest neighbors [8]. In con-

DroNet'18, June 10–15, 2018, Munich, Germany

DOI: https://doi.org/10.1145/3213526.3213527

trast to ASTRO, they are all developed for indoor environments, employ static sensor nodes, and employ supervised learning with a pre-labeled training dataset.

2. DESIGN AND IMPLEMENTATION

In this section, we describe ASTRO in the context of the mission scenario of finding and tracking a mobile spectrum cheater, highlighting the three innovative features.

2.1 Mission Scenario

Here, we describe an example mission that we use to motivate both the design and evaluation of ASTRO. As illustrated in Figure 1, we consider a rogue mobile node (which itself can be a drone) that is transmitting without authorization. In particular, the rogue node *(i)* transmits in a way that is not authorized by spectrum regulations, e.g., by transmitting on a frequency that is controlled by a government agency or licensed to a commercial entity; or *(ii)* transmits in a disruptive way, e.g., launching a denial of service attack in either a licensed or unlicensed band.

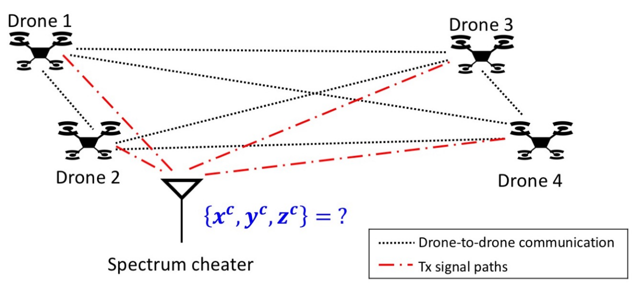

Figure 1: Scenario for ASTRO drones to find and track a spectrum cheater

We consider that the drone mission contains the targeted illicit behavior of the rogue node. In our example, the illicit behavior is any transmission between 563 and 568 MHz to represent the case in which a mobile "spectrum cheater" is transmitting in a prohibited band. The high level objective of this mission is to find and track the mobile spectrum cheater. Namely, ASTRO drones are launched from a location that is assumed to be out of range of the cheater, and they must coordinate to realize the mission without any ground control (tetherless). The definition and realization of this mission are described below.

2.2 On-Drone On-line Light-Weight Machine Learning

2.2.1 Overview

Data processing and control challenges of such a mission reside in *(i)* the lack of prior knowledge or training data about the mission environment; *(ii)* the requirement for real-time response despite the limited processing capability of the on-board computer; and *(iii)* constrained drone flight time due to battery weight constraints ($\sim$10 minutes). As such, the key of designing on-drone algorithms is to balance algorithm complexity and mission performance. To do so, we make use of an on-board spectrum sensor, diverse-spectrum drone-to-drone communications, a signal propagation model, and light-weight machine learning algorithms.

ASTRO operates in two phases: the search and learn phase and the swarm and track phase, depending on whether the target has been initially identified by any drone. Specifically, ASTRO enables networked drones to be launched from an initial arbitrary location that is not necessarily in sensing range of the target. During the search and learn phase, drones fly in accordance pre-computed partitioned zones and planned paths obtained by solving a multiple traveling salesman problem under the constraints of the number of drones, flight time, on-board sensing range, and the target environment [4, 17]. In particular, the zone partition will determine how to divide the target area into zones so that each drone takes charge of one zone, and the planned paths provide guidance for drones to fly within their corresponding zones. Each ASTRO drone works independently during this phase and learns the signal propagation model parameters while trying to identify a potential spectrum cheater (§2.2.2). Once a drone is in range of a target, it informs all other drones. Subsequently, all drones will swarm to the target and switch to the tracking phase in which they work collaboratively to locate and track the target (§2.2.3).

2.2.2 Search and Learn Phase

The goal of this phase is to independently learn model parameters for sensing as well as to search for the target. It therefore features both a machine learning algorithm and an empirical propagation model for receive power at the drone in the cheater's target spectral band as a function of distance.

We select receive power to track the cheater in order to realize it with a single-antenna drone, without any additional hardware, and potentially encountering high vibrations and mobility. Namely, compared to time of arrival, angle of arrival, or time difference of arrival, our method is more suitable for drone based applications, considering drones' mobility and strict constraint on weight. However, locating a transmitter solely based on receive power can encounter high error, as this metric is highly affected by reflections and other interactions of the radio signal with the environment. We then employ a light-weight machine learning algorithm to dynamically adapt model parameters in real-time, aiming to achieve improved modeling and tracking accuracy while maintaining low complexity and minimal hardware requirements.

At the beginning of the search and learn phase, both the environment dependent propagation model and the target location information are unknown. To tackle the challenges of modeling the varied interactions between receive power and corresponding environments and capturing the target position in a short time, we formulate the problem via a machine-learning framework using nonlinear regression [15], and use a batch gradient descent algorithm to solve it. This approach is well suited for the search phase as it is an effective yet simple method for enabling the learning of the model parameters and locating the target simultaneously and in real-time. In the basic radio propagation model, the received signal strength can be simplified as in [9] to:

$$P = \alpha log_{10}(D) + \eta \tag{1}$$

where P denotes the received power strength in dBm and D is the distance between the spectrum sensor (i.e., drone) and the target (i.e., spectrum cheater), and α is proportional to the environment and frequency dependent path-loss exponent. Denoting the positions of the drone and target (cheater) in the Cartesian coordinate system as (x, y, z) and (x^c, y^c, z^c) respectively, the distance is $D = \sqrt{(x^c - x)^2 + (y^c - y)^2 + (z^c - z)^2}$. Note that each ASTRO drone is equipped with an on-board RF receiver and GPS receiver that can obtain P and (x, y, z), respectively. Thus, the goal in the search and learn phase is to utilize these measurements to learn the model parameters (α, η) and the target position (x^c, y^c, z^c).

ASTRO drones continuously sample their own position (x, y, z) and the received signal strength P. In a batch gradient descent algorithm, parameters are updated using a batch of measured data at each step. We formulate the estimation of the five parameters (i.e., $\alpha, \eta, x^c, y^c, z^c$) as a nonlinear regression problem, in which a cost function $J(\alpha, \eta, x^c, y^c, z^c)$ can be defined as:

$$J(\alpha, \eta, x^c, y^c, z^c) = \frac{1}{2N} \sum_{i=1}^{N} (P_i^e(\alpha, \eta, x^c, y^c, z^c) - P_i^m)^2 \tag{2}$$

where N denotes the number of measurements in each batch, P_i^m denotes the measured received signal strength in the i-th measurement, and $P_i^e(\alpha, \eta, x^c, y^c, z^c)$ denotes the i-th estimated signal strength according to Eq. (1) using the estimated path-loss propagation parameters and target location. Essentially, J measures how close the calculated received signal strength based on the estimated parameters and location is to the corresponding measured signal strength. Mathematically, our approach aims to solve the following optimization problem:

$$\underset{\alpha, \eta, x^c, y^c, z^c}{\arg\min}\ J(\alpha, \eta, x^c, y^c, z^c) \tag{3}$$

It is in general intractable to obtain a closed-form solution to Eq. (3). We instead employ a batch gradient descent algorithm [2], which is simple and effective for solving nonlinear regression problems. Specifically, this algorithm iteratively searches possible model parameters and target locations that minimize the cost function in Eq. (2). To do so, it first randomly assigns initial values for the parameters to be learned, and then iteratively performs the following update:

$$\theta_{k+1} = \theta_k - \delta \frac{\partial}{\partial \theta_k} J(\alpha_k, \eta_k, x_k^c, y_k^c, z_k^c) \tag{4}$$

where θ represents one of the five parameters $(\alpha, \eta, x^c, y^c, z^c)$, δ denotes the learning rate, and k denotes the iteration index. Note that all five parameters need to be updated simultaneously. The partial derivative term $\frac{\partial J}{\partial \theta_k}$ corresponding to α in Eq. (4) can be obtained as:

$$\frac{\partial J}{\partial \alpha_k} = \frac{1}{N} \sum_{i=1}^{N} [(P_i^e(\alpha_k, \eta_k, x_k^c, y_k^c, z_k^c) - P_i^m) log10(D_i)] \tag{5}$$

and similarly for other parameters (η, x^c, y^c, z^c).

In summary, our approach for the search and learn phase relies on a single learning model for both propagation parameters and target location based on measured data of each drone independently. Its simplicity enables fast identification of the target within each drone's search area, maximizing the possible covered area and minimizing the mission time, which is critical due to the constrained flight time.

2.2.3 Swarm and Track Phase

Once an ASTRO drone identifies the target, all N drones switch to the collaborative tracking phase. The goal in this phase is to collaboratively locate and track the target.[1] Because the target may be moving, the key challenge lies in ensuring that the tracking algorithm has low latency to realize real-time tracking of the moving target, while at the same time meeting the specified tracking accuracy. To handle this challenge, we utilize a fast and effective tracking algorithm that combines data from multiple drones at different locations to perform collaborative tracking, and a K-means clustering algorithm to suppress potentially large errors that are commonly observed in path-loss based propagation models [14]. Due to space constraints, we sketch our solution in the context of 2-D tracking; we extended to the 3-D tracking case for our implementation.

Using multilateration, a unique solution of the target position (x^c, y^c) at each measured time instance from these N nonlinear equations exists if there are at least three independent measurements/equations, i.e., $N \geq 3$. In other words, this algorithm requires a minimum of three drones for 2-D tracking and four drones for 3-D tracking. Additional drones can further improve the tracking accuracy as noisy tracking results can be ruled out.

As a direct solution to the multilateration location estimation will incur high noise, we utilize a K-means clustering algorithm [11] to improve localization and tracking accuracy. In particular, given a set of adjacent results $(\mathbf{L}_1, \ldots, \mathbf{L}_M)$ obtained with $\mathbf{L}_m = (x_m^c, y_m^c)$, K-means clustering will partition the M results into K ($K \leq M$) sets $\mathbf{S} = \{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_K\}$ so as to minimize the within-cluster sum of squares (i.e., variance). Mathematically, the objective is to find:

$$\underset{\mathbf{S}}{arg\ min} \sum_{k=1}^{K} \sum_{\mathbf{L} \in \mathbf{S}_k} \|\mathbf{L} - \mathrm{c}_k\|^2 \tag{6}$$

In our case, $K = 3$, i.e., the centroid of each M adjacent set of results, is chosen as the estimated target location at each time instance. We choose the size of the adjacent results

[1]For other mission objectives, e.g., multi-target missions, a subset of drones can remain in the search phase. For simplicity of exposition, we consider a single target here.

in each K-means clustering step to optimize the tracking accuracy given the sampling rate and target moving speed.

In summary, our approaches for both phases feature a combination of simple analytical solutions aided with machine learning algorithms in order to ensure fast response time and high localization/tracking accuracy.

2.3 System Architecture and Implementation

We implement ASTRO as a network of hexacopter drones along with all associated hardware and software, including SDRs, to realize the spectrum cheater mission.

2.3.1 *Architecture for Tetherless Operation*

Most existing drone platforms support only a single aerial vehicle and require a ground control station (GCS). The GCS is typically a standard computer that is used as a centralized planning and adaptation point, e.g., to configure mission parameters such as coordinates to cover through waypoint navigation and the action to take at each waypoint [6]. Likewise, GoPro Karma, DJI Phantom, Skydio, realize *Follow Me* applications in which drones autonomously follow a mobile target. However, in these applications, a drone is still tethered to the smartphone which behaves as a mobile GCS [12]. Namely, the drone receives commands from the GCS while executing computer vision algorithms [5], or orders received via human-drone interactions [7].

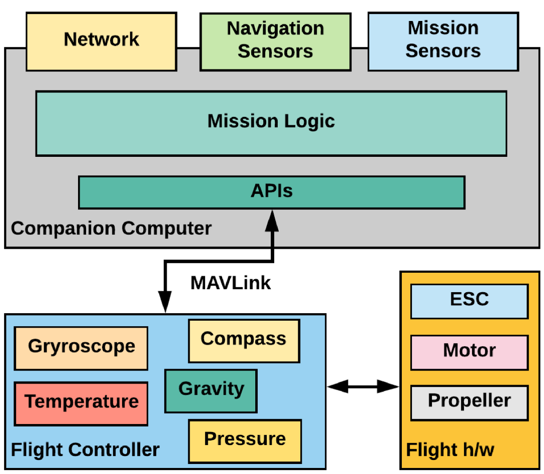

Figure 2: ASTRO software and hardware building blocks

In contrast, ASTRO realizes tetherless operation in which drones communicate with each other but are not required to maintain communication with any ground system and all computing is on-board the drones. To this end, ASTRO drones are enhanced with a software and hardware architecture that supports tetherless and autonomous network sensing missions. Figure 2 depicts the main control building blocks, i.e., Flight Controller (FC) and companion computer, and how they logically connect to the other modules, and their components and functions are described below. Thus, a GC and air-to-ground interfaces are notably not required in the architecture.

2.3.2 *Sensing and Autonomous Flight*

To realize autonomous flight, ASTRO employs a suite of sensors as well as computation and communication resources interconnected to the FC. As shown in Figure 3, ASTRO utilizes a hexadrone structure with a frame that mounts propellers and motors, control for regulating the speed of electric motors, an FC, and an companion computer. The navigation sensors include those required to fly in the environment such as GPS and obstacle avoidance, while the missions sensors are those used to perceive the physical phenomena target of a sensing mission (e.g., an SDR to detect the spectrum cheater). It is critical that each of these components meet the size, weight, and power consumption requirements needed for achieving longer flight times.

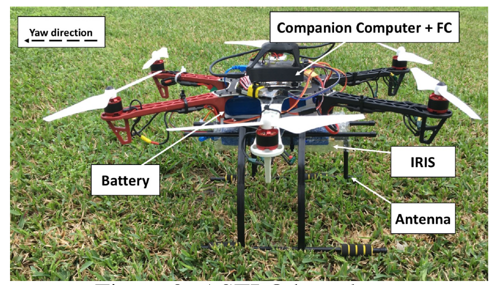

Figure 3: ASTRO hexadrone

The FC is a resource-constrained embedded hardware with built-in sensors including 3 axes gravity sensor, 3 axes gyroscope, 3 axes digital compass, pressure sensor, temperature sensor, and ADC for battery sensing. Together, their purpose is to manage flight dynamics by detecting orientation changes and controlling motors for flight as directed by higher-layer mission objectives. For our targeted outdoor environments, we use GPS to aid navigation and localization. While GPS error can in some cases be as high as 15 m, better accuracy (order of centimeters) can be achieved with methods such as differential-GPS, differential-GNSS, etc., if required. The autopilot software running at this layer is in charge of autonomously driving the drone to the next desired coordinate. The FC is flashed with ArduPilot3, open-source software that supports different boards.

2.3.3 *Communication and Spectrum Sensing*

Each drone is equipped with the IRIS software defined radio, a design based on [3]. IRIS has a transceiver that can be tuned from 50 MHz to 3.8 GHz, with up to 56 MHz of contiguous bandwidth and 12-bit ADC. Baseband functionality is realized via the Zynq 7030 system-on-a-chip, which includes a dual-core ARM Cortex-A9 and an FPGA fabric. Moreover, the IRIS meets the drone's mass budget, weighing in at approximately 160 grams including baseband and RF modules. The IRIS is connected to the drone's companion computer via Ethernet. Figure 4 depicts the on-board SDR (mounted beneath the drone), which is use by both each ASTRO drone as well as the spectrum cheater.

ASTRO drones maintain connectivity with a routing layer employing Better Approach To Mobile Adhoc Networking (BATMAN) [16], a mesh routing operating at Layer 2. BAT-

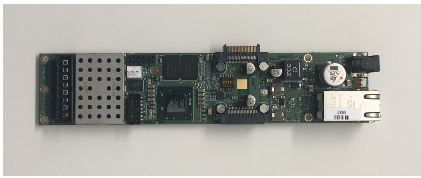
Figure 4: IRIS software defined radio

MAN eliminates the need to spread information concerning network changes to every node in the network and supports multiple network interfaces per node. For the interface card, ASTRO can employ either the SDR, the internal Wi-Fi card of the companion computer, as well as USB Wi-Fi dongles. Note that no air-to-ground network is required to realize tetherless operation.

2.3.4 On-Board Computing

The companion computer is mounted on the drone and has several functions: *(i)* Exposing APIs for the communication with and control of the FC, receiving MAVLink[2] data produced by the FC and using it for decision-making during flight. This enables a broad range of functionality, such as adaptation of flight paths according to post-processed sensor data; *(ii)* Managing the *networking* stack, which is in charge of discovering and maintaining air-to-air links between ASTRO drones, e.g., via the routing protocol above; *(iii)* Processing data received by the *sensing* layer, which is driven by the mission objectives, as the sensors provide the data required by the application. In our initial design, we use SDRs as sensors in order to detect spectrum cheaters. Other sensors such as gas sensors and video cameras can be similarly used for other mission objectives; *(iv)* Receiving from *obstacle avoidance* sensors that perceive possible obstacles in the environment (e.g., buildings) and avoid collision. Detection of an obstacle can be done via different methods, i.e., ultrasonic or infrared sensing, computer vision, and pre-installed maps. While the latter method does not require any additional hardware, it is not flexible since unexpected obstacles not in the map cannot be avoided. We equip ASTRO drones with 360° lidar, a light-weight sensor for obstacle avoidance. If the on-board computing capability permits, computer vision can also be used to enhance this capability.

All of the above functionality is accessible by the *mission logic* block through APIs. The mission logic utilizes such inputs to execute the on-line light-weight machine learning methods described in §2.2. In ASTRO, we use a Raspberry Pi 3 implementing the Raspbian operating system as the companion computer and connect it to the FC via GPIO pins.

3. EXPERIMENTAL EVALUATION

In this section, we present experimental results based on over 500 hours of test flights to explore key features of ASTRO. Due to space constraints, we report on only a small subset of exemplary experiments.

[2]https://mavlink.io/en/

Search and Learn Phase. As described in §2, ASTRO drones are first uploaded with their mission objectives, after which they are launched from an arbitrary location. As in the exemplary mission of §2.1, their first phase after launch is to find the spectrum cheater. In the first experiments, the cheater is placed in random locations of the Rice University football stadium, and transmits between 563 and 568 MHz at 18 dBm. Reception is not amplified to yield a range of approximately 50 m. We configure the drones to have a maximum velocity of 2 m/s, well below their peak capability of approximately 20 m/s, in order to focus on algorithmic dynamics in a relatively small search area.

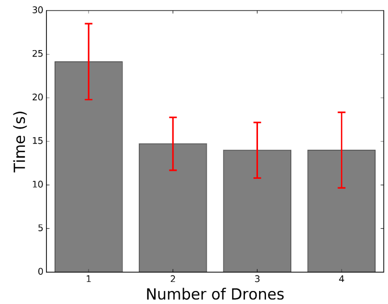

Figure 5: Time to find the cheater

We first evaluate the ability of ASTRO to find the target, as a function of the number of drones, with repeated experiments over different cheater locations. The drones are launched from the edge of the field, and as described in §2.2.2, coordinate to partition the search space into non-overlapping areas. Figure 5 shows the measured time until any drone of the ASTRO network reports to all drones that it has found the cheater (i.e., detected its signal with significant strength). On average, a single drone can find the cheater within 25 s, whereas two drones can cut the average to about 15 s. However, additional drones have only minimal marginal benefit in this stage; further benefits would be expected with a smaller transmission range of the cheater (requiring a more dense search path), a larger field, or slower velocity. Nonetheless, the results show that as the ASTRO network size increases, the drones are able to coordinate without any GC to achieve their mission more efficiently.

Swarm to Localize. After any drone has located the cheater, it informs the others via the drone-to-drone mesh network to enter the next phase to locate and track the cheater. In this phase, the drones cooperatively move towards the cheater. We begin with experiments on a non-mobile cheater in order to evaluate the ability of the drones to cooperatively localize it. For safety, drones keep a minimum 10 m distance from other drones and do not fly lower than 5 m. Figure 6 shows the average localization error as a function of the number of drones. On average, 3 drones cooperatively swarming to localize the cheater have an error of 10 m.

In the above experiments, the cheater was placed in an open football field. We also placed the cheater in a quad surrounded by buildings and placed among a several ton 3.5 m

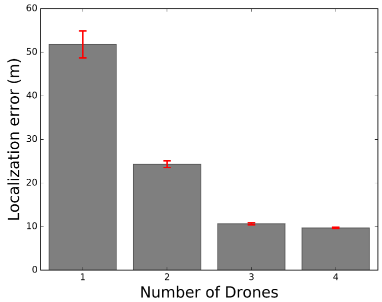

Figure 6: Location error reduction over Number of Drones

by 6 m slab angled at $45°$. Thus, the mission must contend with an irregular propagation environment including multiple reflections off of nearby buildings and the slab. We obtain similar results as above in accuracy, albeit with higher variation due to the new environment.

Tracking a Mobile Cheater. Finally, we perform experiments with a mobile cheater. In this case, a human subject (the first author) walks along the field holding the SDR cheater. He uses the field's hash marks to attempt to maintain an approximate velocity of 1.5 m/s. We use 3 drones with the same mission objective, find and track the cheater.

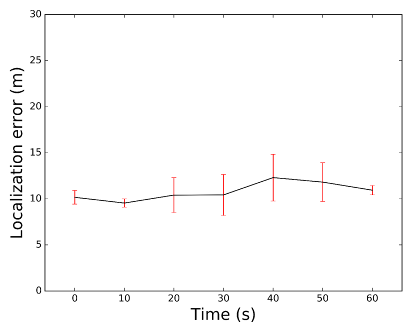

Figure 7: Tracking a mobile cheater

Figure 7 shows the resulting error as a function of time since the "swarm and track" phase. As shown, despite that the cheater's location is changing and the drones must continually move to track it, they maintain similar error throughout the mission. Nonetheless, the drones do not precisely hover over the cheater. While they stay close to the cheater and roughly in the same formation (typically within 20 m total distance with 5 m altitude), signal variations, the machine learning algorithm, and even wind effects, often send a drone several meters "off course." Nonetheless, the prediction error, which combines both drones' measurements and incorporates such errors, remains stable.

4. CONCLUSIONS

We presented the design and evaluation of ASTRO, a system for tetherless and autonomous networked drone sensing missions. We implement the key components of ASTRO and demonstrate ASTRO's capabilities through a series of sensing missions to find and track a mobile spectrum cheater.

Acknowledgments

This research was supported by Cisco, Intel, and by NSF grant CNS-1514285. We thank the Skynet senior design team for their help in building and testing drones.

5. REFERENCES

[1] H. Ahmadi and R. Bouallegue. RSSI-based localization in wireless sensor networks using regression tree. In *Proc. of IEEE IWCMC*, 2015.

[2] E. Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2nd edition, 2010.

[3] N. Anand, R. E. Guerra, and E. W. Knightly. The Case for UHF-band MU-MIMO. In *Proc. of ACM MobiCom*, 2014.

[4] G. S. C. Avellar, G. A. S. Pereira, L. C. A. Pimenta, and P. Iscold. Multi-UAV routing for area coverage and remote sensing with minimum time. *Sensors*, 15(11):27783–27803, 2015.

[5] R. Bartak and A. Vykovsky. Any object tracking and following by a flying drone. In *Proc. of IEEE MICAI*, 2015.

[6] E. Bregu, N. Casamassima, D. Cantoni, L. Mottola, and K. Whitehouse. Reactive control of autonomous drones. In *Proc. of ACM MobiSys*, 2016.

[7] J. R. Cauchard, J. L. E, K. Y. Zhai, and J. A. Landay. Drone & me: An exploration into natural human-drone interaction. In *Proc. of ACM UbiComp*, 2015.

[8] I. T. Haque and C. Assi. Profiling-based indoor localization schemes. *IEEE Systems Journal*, 9(1):76–85, Mar. 2015.

[9] K. Heurtefeux and F. Valois. Is RSSI a good choice for localization in wireless sensor network? In *Proc. of IEEE AINA*, 2012.

[10] Z. Liu, Y. Chen, B. Liu, C. Cao, and X. Fu. HAWK: An unmanned mini-helicopter-based aerial wireless kit for localization. *IEEE Transactions on Mobile Computing*, 13(2):287–298, 2014.

[11] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[12] W. Mao, Z. Zhang, L. Qiu, J. He, Y. Cui, and S. Yun. Indoor follow me drone. In *Proc. of ACM MobiSys*, 2017.

[13] S. Paschall and J. Rose. Fast, lightweight autonomy through an unknown cluttered environment. In *Proc. of IEEE Aerospace Conference*, 2017.

[14] J. Ramiro Martinez-de Dios, A. Ollero, F. J. Fernandez, and C. Regoli. On-line RSSI-range model learning for target localization and tracking. *Journal of Sensor and Actuator Networks*, 6(3), 2017.

[15] G. Seber and C. Wild. *Nonlinear Regression*. Wiley Series in Probability. Wiley, 2003.

[16] D. Seither, A. Konig, and M. Hollick. Routing performance of wireless mesh networks: A practical evaluation of BATMAN advanced. In *Proc. of IEEE LCN*, 2011.

[17] A. Wang, X. Ji, D. Wu, X. Bai, N. Ding, J. Pang, S. Chen, X. Chen, and D. Fang. GuideLoc: UAV-assisted multitarget localization system for disaster rescue. *Mobile Information Systems*, 2017.

[18] Y. Wang, X. Xu, and X. Tao. Localization in wireless sensor networks via support vector regression. In *Proc. of ICGEC*, 2009.

[19] J. Zheng and A. Dehghani. Range-free localization in wireless sensor networks with neural network ensembles. *Journal of Sensor and Actuator Networks*, 1(3):254–271, 2012.