

# Flocking with Fixed-Wing UAVs for Distributed Sensing: A Stochastic Optimal Control Approach

Steven A. P. Quintero, Gaemus E. Collins, and João P. Hespanha

**Abstract**—This work focuses on enabling multiple UAVs to flock together in order to distribute and collectively perform a given sensing task. Flocking is performed in a leader-follower fashion, and the leader is assumed to already have an effective control policy for the particular task. The UAVs are small fixed-wing aircraft cruising at a constant speed and fixed altitude, but experience stochasticity in their dynamics. Accordingly, the control problem for each follower is addressed in the context of stochastic optimal control, wherein the cost is a function of distance and heading with respect to the leader. The problem is solved offline via dynamic programming to minimize the expected cost over a finite horizon and generate a receding horizon optimal control policy. This flocking algorithm was successfully applied in the field, where three camera-equipped UAVs flocked together to perform vision-based target tracking. The experimental results verify the efficacy of the approach and show the benefits of flocking with multiple UAVs to distribute sensing tasks, which include a dramatic reduction in overall sensing error and robustness to individual sensor faults.

## I. INTRODUCTION

In recent years, unmanned aerial vehicles (UAVs) have enhanced the performance of remote sensing tasks such as surveillance, search and rescue, mapping, and real-time monitoring. The UAVs considered in this work are small fixed-wing aircraft that fly at a fixed altitude and constant airspeed, and they are equipped with a global positioning system (GPS), an inertial navigation system (INS), and a sensor payload suited to the given mission. Having several such UAVs perform a mission as a unified group is often preferred to using a single large UAV, as it is less costly and operationally burdensome. Therefore, the purpose of flocking is to overcome the technological limitations imposed by embedded hardware on a small airframe by distributing the sensing and computational tasks amongst flock members. As information is shared and combined within the group, flocking improves the overall performance of the mission and adds robustness to individual sensor faults. In this work, a *flock* (or *swarm*) is regarded as a tight, collision-free formation of UAVs that have a common velocity vector, which is similar to the description in [1].

Perhaps the most well known biologically inspired model for flocking is the *Boids* algorithm developed by Craig

Reynolds in 1987 for creating computer animations of flocks of birds [2]. In the distributed Boids algorithm, each agent *separates* itself from its nearby neighbors, *aligns* itself with the average heading of these same neighbors, and lastly, steers toward the average position of the nearby neighbors for *cohesion*. Inspiration from Reynolds and advancements in mobile robot technologies have led to a significant amount of work designing distributed control laws to enable mobile robots to flock together. Tanner et al. present control laws that guarantee asymptotic flocking for multiple agents having double integrator dynamics under the assumption that the proximity-based graph representing the network topology remains connected at all times [1]. By also working with agents having double integrator dynamics, Olfati-Saber provides and analyzes several distributed control algorithms for flocking [3], one of which prevents flock fragmentation while another enables the flock to split and rejoin to avoid obstacles.

Other works have considered a relaxed objective for flocking where  $N$  nonholonomic agents moving at unit speed only need to align their velocity vectors. Moshtagh and Jadbabaie develop distributed control laws in [4] that enable a group of such agents to align their headings, so long as the proximity-based graph representing the network topology is connected. Similarly, Paley and Peterson assume an all-to-all communication topology and develop distributed control laws in [5] that synchronize the group's collective motion, i.e., align the agent headings, in a known flowfield, e.g., constant-velocity wind. While these two works consider unit-speed nonholonomic vehicles with first-order rotational (heading) dynamics, Mellish and Paley expand upon the work in [5] by using backstepping techniques to synchronize the collective motion of unit-speed nonholonomic vehicles with second-order rotational dynamics in a uniform flowfield [6].

While the works considered thus far have provided distributed control laws that guarantee stable flocking under certain assumptions, none have demonstrated the algorithms with actual hardware. However, Hauer et al. have successfully demonstrated Reynolds flocking with 10 fixed-wing UAVs in the field [7]. Nonetheless, the primary goal of flocking controllers has generally been to have the group simply migrate together. Moreover, while the commonly stated power of flocking lies in the group's ability to distribute communication, sensing, and computational tasks amongst flock members [7], little work has been done to demonstrate its effectiveness at achieving this in practical applications.

We highlight two main contributions of this work. The first is the development of a novel flocking algorithm, wherein we have assumed a leader-follower network topology and

Research supported by the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

Steven A. P. Quintero and João P. Hespanha, Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560, USA, {quintero, hespanha}@ece.ucsb.edu.

Gaemus E. Collins, Toyon Research Corporation, 6800 Cortona Dr., Goleta, CA 93111, USA

a preexisting control policy for the leader suited to the particular mission. Our focus is on the follower’s control policy, and we develop a stochastic kinematic model for the UAVs and a cost function that facilitates flocking. We refer to the problem of generating the follower’s control policy as the *stochastic optimal flocking problem*. This is a stochastic optimal control problem whose solution is an optimal control policy that facilitates flocking among mobile agents with arbitrary dynamics.

The second contribution of the work is a successful field test demonstration of three camera-equipped UAVs flocking to perform the rather demanding sensing task of *vision-based target tracking*. The objective is to have each UAV keep an unpredictable ground vehicle in the camera’s limited sensing region and simultaneously maintain proximity to the vehicle. We assume the leader utilizes a control law that is reasonably effective at achieving this objective so that the followers can inherit this ability by flocking with the leader. In addition, by distributing the sensing task across multiple UAVs, the group dramatically reduces joint target localization error and adds robustness against a single agent losing sight of the target. Field test data illustrate these benefits.

The paper is organized as follows. In Section II, we introduce the overall control approach for a follower and discuss the stochastic kinematic model for a UAV, the cost function for flocking, and the dynamic programming solution. In Section III, we first provide the details of the target tracking application, which include the metric used to judge mission performance, the UAV test platform and its sensing limitations, and the nature of the leader’s control policy. Secondly, we provide the experimental details and flight test results. Conclusions and directions for future work are discussed in Section IV.

## II. CONTROL POLICY FOR FLOCKING

For the purpose of designing effective flocking algorithms, we first develop an empirical characterization of the stochastic UAV dynamics that encompasses the reality of aircraft dynamics. Since the flock has a particular mission other than migration, there is a flock leader that is assumed to already have a control policy suited to the mission, which is known to the entire flock. Furthermore, the leader broadcasts its state (and other relevant mission information) to the remaining flock members, which we call followers.

Next, we present the stochastic optimal flocking problem for the follower, which is a stochastic optimal control problem wherein the cost is a function of distance and heading with respect to the leader. We use dynamic programming to minimize the expected cost over a moderate planning horizon and generate the optimal control policy, which is the same for each follower. In this manner the followers have no regard for one another, yet this is suitable for the target tracking application whose main requirement for any followers is that they remain close to the leader. In practice, the aircraft are flown at different altitudes to deconflict them. Nevertheless, one could add additional terms to the cost function to facilitate desired flock geometries.

### A. UAV Dynamics

While the majority of work on UAV flocking control uses continuous time motion models, this work treats the flocking problem in discrete time. In particular, we assume that each UAV has an autopilot that regulates commanded roll angle, pitch, airspeed, and altitude via internal feedback loops, typically using Proportional-Integral-Derivative (PID) control. Once every two seconds, the roll-angle setpoint  $r$  is updated according to the control policy to achieve flocking.

In practice, UAVs are subject to environmental disturbances, such as wind gusts, that introduce stochasticity into the dynamics. Although a real UAV’s kinematics are most accurately captured by a 6 degree of freedom (DoF) aircraft model, we opt for a 4-state stochastic model of the kinematics, in which stochasticity accounts for the effects of both unmodeled dynamics (arising from the reduced 4th order model) and environmental disturbances.

In the aircraft model, we assume it flies at a nominal airspeed  $s$  and at a fixed altitude. The UAV’s planar position  $(x, y) \in \mathbb{R}^2$  and heading  $\psi \in \mathbb{S}^1$  are measured in a local East-North-Up (ENU) earth coordinate frame while its roll angle  $\phi \in \mathbb{S}^1$  is measured in a local North-East-Down (NED) body frame. In the latter coordinate frame, the  $x$ -axis points out of the nose, the  $y$ -axis points out of the right wing, and the  $z$ -axis completes the right-handed coordinate frame. Hence, we define the UAV state as  $\xi := (x, y, \psi, \phi)$ .

We assume the UAV’s state evolves stochastically according to a Markov Decision Process, where the probability of transitioning from the current state  $\xi$  to the next state  $\xi'$  under the roll command  $r$  is given by the state transition probability function  $p_a(\xi' | \xi, r)$ . Rather than deriving an explicit formula for this state transition probability, we develop an empirical characterization of the UAV kinematics that will allow us to draw random samples  $\xi'_i$ ,  $i \in \{1, \dots, N_p\}$ , from this conditional probability density function. This ability to sample the state transition probability will suffice to effectively approximate the dynamic programming solution.

To develop our stochastic discrete-time kinematic model, we begin with a deterministic continuous-time model, which is given by

$$\dot{\xi} = \frac{d}{dt} \begin{pmatrix} x \\ y \\ \psi \\ \phi \end{pmatrix} = \begin{pmatrix} s \cos \psi \\ s \sin \psi \\ -(\alpha_g/s) \tan \phi \\ f(\phi, r) \end{pmatrix}, \quad (1)$$

where  $\alpha_g$  is the acceleration due to gravity. The function  $f(\phi, r)$  defines the roll dynamics, and could be, for example,  $f(\phi, r) = -\alpha_\phi(\phi - r)$  with  $\alpha_\phi > 0$ . However, we will actually use a much more detailed model for the roll dynamics. To generate a discrete-time model, we will apply a 2-second zero-order hold (ZOH) to (1), where the roll command  $r$  belongs to the set

$$C := \{0^\circ, \pm 10^\circ, \pm 20^\circ, \pm 30^\circ\}. \quad (2)$$

Denoting the roll command over the next ZOH period by  $r'$ ,

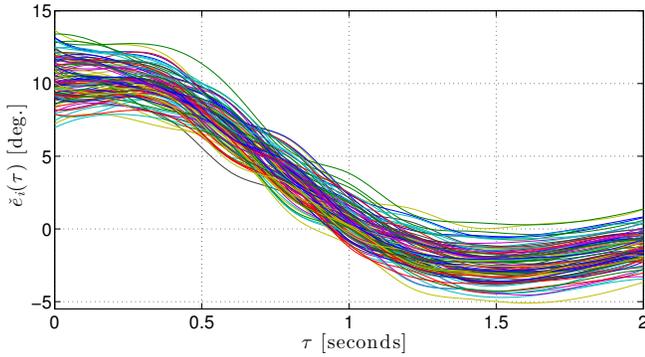


Fig. 1. Error trajectories resulting from a  $-10^\circ$  change in the roll setpoint.

we also stipulate that  $r' \in U(r)$ , where

$$U(r) := \{r, r \pm 10^\circ\} \cap C$$

is the *roll-angle action space*. This allows roll commands to change by at most  $10^\circ$  and avoids sharp changes in roll that would be detrimental to image processing algorithms in the target tracking task [8].

Rather than considering a simple kinematic model with first order roll dynamics for  $f(\phi, r)$  in (1), we sampled roll trajectories over the 2-second ZOH period using Monte Carlo simulations from the high-fidelity flight simulator Aviones [9], which utilizes a 6-DoF aircraft model. In particular, we held constant a given roll setpoint  $r \in C$  over the 2-second ZOH period and then randomly select a new roll setpoint  $r' \in U(r)$ , where all elements of the roll action space  $U(r)$  occur with equal probability. Through this process, we collected families of roll trajectories for which, once every 2 seconds, the roll command increased by  $10^\circ$ , decreased by  $10^\circ$ , or remained constant. Corresponding to each of these roll trajectory families are the collections of reference tracking error trajectories  $\{\check{e}_i(\tau)\}$ ,  $\{\tilde{e}_i(\tau)\}$ , and  $\{\bar{e}_i(\tau)\}$ , respectively, where  $\tau \in [0, 2]$ . Figure 1 shows a collection of reference tracking error trajectories,  $\{\check{e}_i(\tau)\}$ ,  $\tau \in [0, 2]$ , over the 2-second ZOH period for  $N_p = 100$  decreases of  $10^\circ$  in the roll-angle setpoint.

The collection of these Monte Carlos simulations provides us with a large sample of roll-angle trajectories. For example, to generate a sample roll trajectory  $\check{\phi}_i(\tau, r)$ ,  $\tau \in [0, 2]$  corresponding to a  $10^\circ$  decrease in the roll setpoint needed to reach the reference roll angle  $r \in C$ , we take  $\check{\phi}_i(\tau, r) = \check{e}_i(\tau) + r$ . Similarly, the sampled tracking errors  $\{\bar{e}_i(\tau)\}$  and  $\{\tilde{e}_i(\tau)\}$  can be used to generate samples of roll-angle trajectories corresponding to holds and increases in the roll-angle setpoint, respectively. In the sequel, we denote by  $\Phi(r)$  the collection of all the sample roll-angle trajectories so generated for each setpoint  $r \in C$ .

To make the aircraft model more realistic we also introduced stochasticity in the airspeed  $s$ , which was assumed to be normally distributed about a nominal value  $\bar{s}$ , with variance  $\sigma_s^2$ . Specifically, the airspeed  $s$  in (1) was assumed to be constant over each 2-second ZOH period with values drawn from the distribution  $\mathcal{N}(\bar{s}, \sigma_s^2)$ .

Denoting the discrete-time index by  $k \in \mathbb{Z}_{\geq 0}$ , this modeling technique results in samples for the “next” state  $\xi'_i$  at  $t = 2(k + 1)$  seconds, given the “current” state  $\xi$  at  $t = 2k$  seconds and the roll-angle setpoint  $r$ , which is held constant for  $t \in [2k, 2(k + 1))$ . Moreover, the samples have two sources of randomness: stochasticity in the roll-angle dynamics captured by the collection of roll-angle trajectories  $\Phi(r)$  and stochasticity in the airspeed.

While the model introduced above describes the dynamics of one UAV, for the purpose of flocking, all that matters is the relative dynamics between leader and follower. Denoting by the leader’s state by  $\xi_\ell$  and the follower’s state by  $\xi_f$ , we define the system’s state to be the vector  $z := [z_1, z_2, z_3, z_4, z_5, z_6]^\top \in \mathcal{Z}$ , where

$$\mathcal{Z} := \mathbb{R}^2 \times [-\pi, \pi) \times \mathbb{S}^1 \times \mathbb{S}^1 \times C;$$

the pair  $(z_1, z_2)$  denotes the 2D position of the follower relative to the leader, given by

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} := \begin{bmatrix} \cos \psi_\ell & \sin \psi_\ell \\ -\sin \psi_\ell & \cos \psi_\ell \end{bmatrix} \begin{bmatrix} x_f - x_\ell \\ y_f - y_\ell \end{bmatrix}; \quad (3)$$

and the remaining states are

$$(z_3, z_4, z_5, z_6) := (\psi_f - \psi_\ell, \phi_f, \phi_\ell, r_\ell), \quad (4)$$

where  $r_\ell$  is the leader’s roll command.

To arrive at an empirical characterization of the overall state transition probability  $p(z' | z, r)$ , where  $r$  is the follower’s roll command, one simply combines  $\xi'_\ell$  and  $\xi'_f$  according to equations (3) and (4). The leader’s next control action  $z'_6$  is determined by its control policy.

### B. Optimization Criterion for Flocking

Inspired by Reynold’s *Boids*, our flocking optimal controller is based on a cost that is a function of distance and heading with respect to the leader. Consider an annulus  $A$  with inner radius  $a$ , outer radius  $b$ , and centered at the leader’s position:

$$A = \{(z_1, z_2) \in \mathbb{R}^2 : a \leq \rho \leq b\},$$

where  $\rho := \sqrt{z_1^2 + z_2^2}$ . The distance from the follower to this annulus is denoted by  $d$  and is given by

$$d = \max\{a - \rho, 0, \rho - b\}. \quad (5)$$

We consider a cost function that is given by

$$g(z) = \max \left\{ d, \frac{a|z_3|}{\pi(1 + \beta d)} \right\}, \quad (6)$$

where  $\beta$  is a positive tuning parameter. The first argument in the max function is the follower’s distance  $d$  to the annulus and the second argument is the absolute difference of the heading angles weighted inversely by  $d$ . Therefore, while the follower is either very far from the leader or very close, the first argument of the max function is active and regulates the distance from the leader to be within a certain range. This first argument thus combines the *separation* and *cohesion* rules of Reynold’s *Boids* algorithm. When the follower gets inside or fairly close to the annulus, the second argument of the max function becomes active and *aligns* the headings of the two aircraft, just as in the *Boids* algorithm.

### C. Control Objective and Dynamic Programming Solution

The stochastic optimal flocking problem is to determine the optimal control feedback control policy  $F_k^* : \mathcal{Z} \rightarrow \mathcal{C}$ ,  $k \in \{0, \dots, K-1\}$ , that minimizes

$$J(z[0]) = E \left[ \sum_{k=0}^K g(z[k]) \middle| z[0] \right], \quad \forall z[0] \in \mathcal{Z}, \quad (7)$$

where  $K \in \mathbb{N}$ ,  $E[\cdot]$  denotes expectation, and  $z[k]$  refers to  $z(t)$  at discrete time instances, i.e.,  $z[k] = z(2k)$ . Also, in this equation, the state  $z$  is a Markov Decision Process that evolves according to the transition probability  $p(z' | z, r)$  and the feedback law  $r[k] = F_k^*(z)$ , where  $r[k] = r(2k)$ .

Dynamic programming can be used to minimize the criterion (7) by optimizing control actions in reverse chronological order. To compute optimal controls using dynamic programming, we perform *value iteration* as presented in [10]. To describe the method, we introduce the *value function*, or *cost-to-go* from state  $z \in \mathcal{Z}$  at time  $k \in \{0, \dots, K-1\}$ , as

$$V_k(z) := g(z) + \min_{r_k, r_{k+1}, \dots, r_{K-1}} E \left[ \sum_{n=k+1}^K g(z[n]) \right],$$

where  $r_k$  is shorthand for  $r[k]$ . For  $k = K$ , we have  $V_K(z) = g(z)$  and the cost-to-go for previous times is computed (offline) in reverse chronological order, i.e.,  $k = (K-1, \dots, 0)$ , according to the following recursion

$$\begin{aligned} V_k(z) &= g(z) + \min_{r \in U(\phi_f)} E [V_{k+1}(z') | z, r] \\ &= g(z) + \min_{r \in U(\phi_f)} \int V_{k+1}(z') p(z' | z, r) dz'. \end{aligned} \quad (8)$$

This recursion holds due to Bellman's principle of optimality, and performing this sequence of computations yields  $J(z) = V_0(z)$ ,  $\forall z \in \mathcal{Z}$ . Also, as this minimization is performed, the optimal control policy is formed as

$$F_k^*(z) = \arg \min_{r \in U(\phi_f)} \left( g(z) + E [V_{k+1}(z') | z, r] \right), \quad \forall z \in \mathcal{Z}.$$

The main hurdle in computing equation (8) is the expectation, which we overcome by replacing it with an empirical average computed using samples drawn according to the stochastic UAV model developed in Section II-A:

$$V_k(z) \approx g(z) + \min_{r \in U(\phi_f)} \frac{1}{N_p} \sum_{i=1}^{N_p} V_{k+1}(z'_i). \quad (9)$$

Furthermore, to limit the computation of the value function to a finite number of points, we introduce a finite subset  $\mathcal{S}$  of the state space  $\mathcal{Z}$  having  $N_s$  distinct elements and the quantization function  $q : \mathcal{Z} \rightarrow \mathcal{S}$ , defined by

$$q(z) := \arg \min_{w \in \mathcal{S}} \|z - w\|_1.$$

TABLE I

AIRCRAFT MODEL AND COST FUNCTION PARAMETER VALUES

Parameter:	$\bar{s}$	$\sigma_s$	$a$	$b$	$\beta$	$K$	$N_p$
Value:	18	4/5	40	65	0.05	4	1,000

Based on these, we approximate the computation of the value function and optimal policy by

$$\begin{aligned} V_k(z) &\approx g(z) + \min_{r \in U(\phi_f)} \frac{1}{N_p} \sum_{i=1}^{N_p} V_{k+1}(q(z'_i)) \\ F_k^*(z) &= \arg \min_{r \in U(\phi_f)} \left( g(z) + \frac{1}{N_p} \sum_{i=1}^{N_p} V_{k+1}(q(z'_i)) \right), \end{aligned}$$

which only require the evaluation of the value function over the finite set  $\mathcal{S}$ . In practice, to lookup the optimal roll command  $r[k]$  for a state  $z \in \mathcal{Z} \setminus \mathcal{S}$ , we use  $r[k] = F_k^*(q(z))$ .

For the hardware experiment, the discretized state space considered was  $\mathcal{S} = X^2 \times \Psi \times C^3$ , where  $X = \{-300, -295, \dots, 300\}$ ,  $\Psi = \{0^\circ, 15^\circ, \dots, 345^\circ\}$ , and the set  $C$  is defined in (2). The remaining parameters used in the dynamic programming optimization are given in Table I, where the first two columns have units of [m/s]. The control policy is applied in a receding horizon fashion, which entails that  $r[k] = F_0^*(z)$ ,  $\forall k \in \mathbb{Z}_{\geq 0}$ . A planning horizon of  $K = 4$  corresponds to a follower UAV that considers the impact its control decisions have on the flocking cost (7) up to 8 seconds into the future.

### III. HARDWARE EXPERIMENTS

This section presents the field test results of three camera-equipped UAVs flocking together to perform vision-based target tracking as a group. We first introduce the problem by describing the process of geolocation (target localization) and the associated error covariance, which is the metric used to judge mission performance. We then discuss the UAV hardware and corresponding sensing limitations and provide a brief description of how the leader's control policy was designed to address such limitations. Next, we describe the experimental setup for the field tests. Finally, we present and discuss the results that demonstrate not only the effectiveness of the flocking control policy, but also the benefits of flocking for this particular sensing task.

#### A. Vision-based Target Tracking and Geolocation

While vision-based target tracking has been given significant attention in recent years, [11]–[14], for example, to the best of the authors' knowledge, none have explicitly considered flocking with  $N$  UAVs as a means to improve tracking performance. For a single camera-equipped UAV tracking a moving ground vehicle, onboard video processing algorithms are typically used to determine the centroid pixel coordinates of the target, as it moves in the image frame. Using these pixel coordinates, the intrinsic and extrinsic camera parameters, and terrain altitude data, one can estimate the 2D location of the target in inertial coordinates and

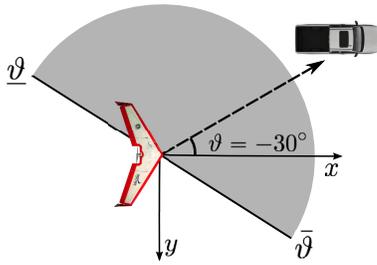


Fig. 2. The *field of regard* is indicated by the shaded region and is the total area visible to the camera as the gimbal is swept through its entire range of pan angles. If the azimuth angle  $\vartheta$  of the line-of-sight vector to the target lies within the upper and lower bounds,  $\bar{\vartheta}$  and  $\underline{\vartheta}$ , respectively, then the target is said to be in the UAV's field of regard.

compute the associated error covariance [15]. This is the process of *geolocation*.

Geolocation error is directly proportional to the distance from the target. However, hovering over the target to fully reduce this error is not only infeasible due to kinematic constraints, but also undesirable from the standpoint of the sensing limitations discussed in the next section. To mitigate the effects of this proximity restriction on geolocation error, one can fuse geolocation estimates across multiple UAVs, as in [14]. Accordingly, we utilize the same metric of fused geolocation error covariance  $\mathcal{P} \in \mathbb{R}^{2 \times 2}$  to assess tracking performance, which can be computed in terms of individual geolocation error covariances  $P_j \in \mathbb{R}^{2 \times 2}$  for  $N$  UAVs, as follows:  $\mathcal{P}^{-1} = \sum_j P_j^{-1}$ , where  $P_j$  is derived in both [14] and [15]. We shall see that flocking with 3 UAVs provides a significant reduction in  $\mathcal{P}$ .

### B. UAV Hardware and Leader Control Policy

To demonstrate flocking for vision-based target tracking, we used Unicorn/Zagi flying wings. The Unicorn comprises a 60" expanded polypropylene foam wing that houses four lithium polymer batteries driving an electric motor attached to a push propeller. The autopilot adjusts the throttle and two elevons to maintain pitch, airspeed, altitude, and the commanded roll angle.

Each UAV carries a gimballed video camera with two degrees of freedom. The visibility region imposed by the particular pan angle limitations of the gimbal mechanism is illustrated in Figure 2 and is subsequently referred to as the *field of regard* (FOR). The most prominent feature of this figure is that there is a large blind spot extending from the right side of the UAV to its back, and hence the UAV must keep the target to its left for visibility. If the target exits UAV  $j$ 's FOR, we take  $P_j^{-1} \equiv 0$ .

The tilt (or elevation) angle of the gimbal mechanism is illustrated in Figure 3. Although its mechanical limitations do not create blind spots, there are still reasons to avoid extreme tilt angles. In particular, a small tilt angle usually means that the airframe is visible to the camera, which can block visibility of the target and/or generate false detections in image processing software. On the other hand, a large tilt angle results in unpredictable movement of the gimbal, as

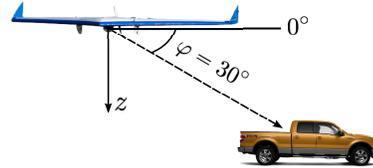


Fig. 3. The tilt (or elevation) angle  $\varphi$  of the gimbal mechanism.

this represents a singularity point in the gimbal geometry, i.e., the pan angle is not unique [8].

Based on these sensor limitations, the control objective for a single UAV performing vision-based target tracking is to maintain a good viewing geometry with respect to an unpredictable target vehicle on the ground. This means that the UAV should not only keep its position as close as possible to the target to reduce geolocation error, but also keep the target in its FOR while simultaneously avoiding extreme gimbal angles. Thus, the control policy for the leader was obtained by solving (offline) a stochastic optimal control problem for a cost function that penalizes the distance from a stochastic target and extreme gimbal angles. This control approach is detailed and compared with an alternative game theoretic control approach in a paper under preparation that includes further details and field test results [16].

### C. Experimental Setup

A flight experiment was conducted in July, 2012 with three UAVs at the McMillan Airfield, Camp Roberts, CA. Toyon Research Corporation was responsible for launching, monitoring, and landing the UAVs.

During the tracking experiment, the UAVs were controlled by the ground control station (GCS) while a truck with GPS was driven along the roads of Camp Roberts. By communicating with the GCS and the GPS receiver onboard the truck, MATLAB<sup>®</sup> was able to acquire the UAV telemetry and target data and determine the optimal roll command for each UAV according to the leader and follower control policies. These roll commands were then sent back to the GCS and relayed to the UAVs.

The UAVs were each set to fly at a nominal airspeed of 18 [m/s]. The leader,  $\mathcal{A}_1$ , flew at an altitude of 170 [m] while the two followers,  $\mathcal{A}_2$  and  $\mathcal{A}_3$ , flew at altitudes of 200 [m] and 230 [m], respectively. Video footage from each camera was sent to the ground on a dedicated radio link, which is shown in Figure 4. The video footage illustrates the diverse viewing angles of the ground target achieved by the flock as the target travels along a dirt road at Camp Roberts.

### D. Experimental Results

We now highlight 3 minutes of the flight experiment that captures the essence of vision-based target tracking with a flock of 3 UAVs. The target and UAV trajectories during this window are provided in Figure 5. In the figure, we observe  $\mathcal{A}_1$  keeping to the right of the target  $\mathcal{T}$  and making counterclockwise turns about the target. Such behavior enables the UAV keep the target in its FOR and stay close to the

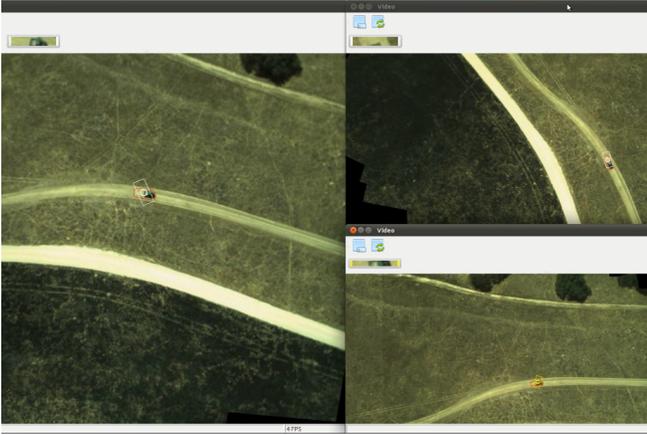


Fig. 4. Live simultaneous video footage from the flock. Each UAV tracks the ground vehicle in the image plane with video processing algorithms developed by Toyon Research Corporation.

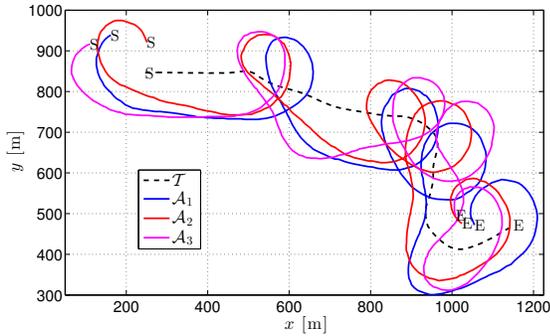


Fig. 5. Trajectories of the UAVs over 3 minutes as they flock to track the target. An “S” indicates a starting position and “E” an ending position.

target in spite of a ground speed differential. Also, the leader never passes over the target to fully minimize geolocation error, which illustrates the tradeoff between minimizing the distance to the target and avoiding a large elevation angle. By virtue of flocking, the followers have similar trajectories and inherit tracking abilities similar to those of the leader, which are influenced by flocking performance.

The performance of the flocking controller for the two followers is given in Figure 6. The first follower  $\mathcal{A}_2$  begins in a somewhat poor configuration relative to the leader, as indicated by  $d$  and  $z_3$ . However, it recovers after nearly 45 seconds and maintains a relatively good flocking cost ( $g(z) < 40$ ) for the remainder of the interval. The second follower  $\mathcal{A}_3$  begins in a good configuration and maintains a good flocking cost for nearly two minutes. When the target begins making major turns, the leader maneuvers to keep sight of the target and disturbs the flocking performance of  $\mathcal{A}_3$ . However, this agent also recovers by the end of the interval. Thus the flock ends in a good configuration after compensating for poor initial conditions and disturbances, which can also be seen in Figure 5.

The tracking performance of the flock is illustrated in Figure 7. To illustrate the advantages of cooperative target tracking via flocking over single-UAV target tracking, we have

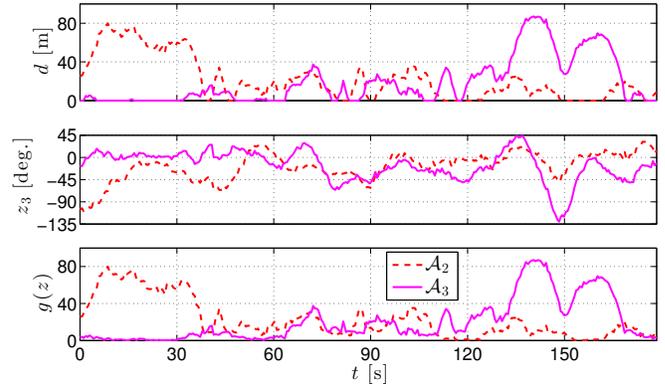


Fig. 6. Performance of the flocking controller. The quantities  $z_3$ ,  $d$ , and  $g(z)$  are given by (4), (5), and (6), respectively. The flocking cost  $g(z)$  is primarily governed by  $d$  and illustrates the controller’s ability to recover from poor initial conditions and disturbances.

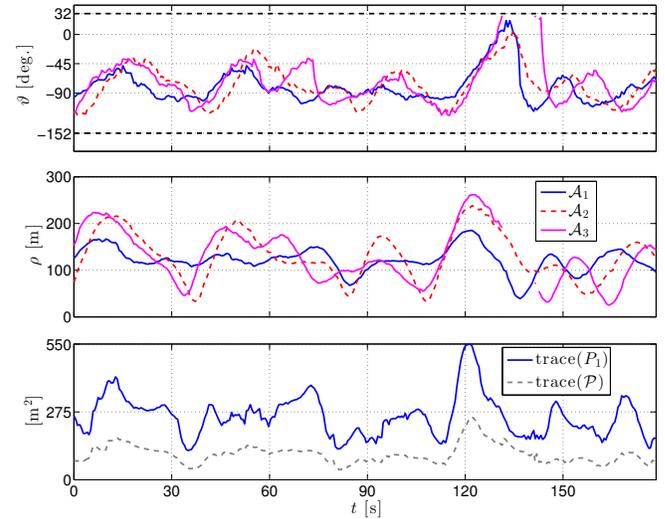


Fig. 7. The tracking performance of the flock. The signals  $\vartheta_j$  and  $\rho_j$  denote each flock member’s individual azimuth angle and planar distance to target, respectively. The FOR limits for the azimuth angles,  $\vartheta$  and  $\vartheta$ , are indicated by dashed lines at  $-152^\circ$  and  $32^\circ$ , respectively. Also, the flock’s fused GEC  $\mathcal{P}$  is compared to the leader’s individual GEC  $\mathcal{P}_1$ .

also provided a comparison of the flock’s fused geolocation error covariance (GEC) to the leader’s individual GEC. The figure highlights two key features of the fused geolocation error. Namely, it is primarily governed by minimum distance to target, and it is often significantly less than that of the leader. In fact, when the trace of the leader’s GEC peaks near  $t = 120$  [s], the trace of the fused GEC is less than half its value. Also, over this 3-minute span, the average value of  $\text{trace}(\mathcal{P})$  was approximately 105 while that of  $\text{trace}(\mathcal{P}_1)$  was approximately 257, which is nearly a 60% reduction in the average value of the trace of the GEC. Moreover, the figure illustrates that performing vision-based target tracking with a flock of even just 3 UAVs dramatically reduces both the peak and average values of the overall geolocation error.

Each UAV generally keeps its azimuth angle centered around  $-90^\circ$  in order to keep the target to its left for visibility; however, we observe all azimuth angles tending

to zero near  $t = 120$  [s] when the target begins to make its second major turn left. This shows the target's two major direction changes disturbing the flock's steady state tracking performance. When  $t \in [132, 142]$ ,  $\vartheta_3$  and  $\rho_3$  vanish to indicate that the target left  $\mathcal{A}_3$ 's FOR. Thus, the perturbation to the flocking performance of  $\mathcal{A}_3$  resulting from the leader's counter maneuvers ultimately caused  $\mathcal{A}_3$  to lose sight of the target completely. This is supported by the observation that  $\vartheta_3$  deviated significantly from  $\vartheta_1$  and exceeded the upper FOR limit during the same time that  $g_3(z)$  hit its peak value. Moreover, we witness the influence of the flocking performance of a follower on its tracking performance.

On a final note, when  $t \in [132, 142]$ ,  $P_3 \equiv 0$ , yet  $\mathcal{P}$  was affected only slightly since  $\mathcal{A}_3$  was not the closest agent to the target. Thus, Figure 7 also illustrates the robustness of vision-based target tracking with a flock to individual sensor faults, as the sensing is distributed across multiple UAVs.

#### IV. CONCLUSION

We have detailed the design of a novel flocking algorithm that enabled multiple, small, fixed-wing UAVs to flock together in order to distribute a given sensing task among group members. Flocking is performed in a leader-follower fashion, where the leader is assumed to already have an effective control policy for the particular sensing task. Since the UAVs experience stochasticity in their dynamics, we developed an empirical characterization of the UAVs' stochastic kinematics. We also presented the stochastic optimal flocking problem for the follower, which is a stochastic optimal control problem wherein the cost is a function of distance and heading with respect to the leader. The solution is an optimal control policy that facilitates flocking among mobile agents with arbitrary dynamics.

We demonstrated the power of flocking in a practical application by testing the flocking control algorithm in the field. In particular, three camera-equipped UAVs were tasked with performing vision-based target tracking. The flight test results verified the effectiveness of the flocking controller and showed the benefits of flocking with multiple UAVs.

Future work involves adapting the cost function to add collision avoidance among followers and even induce certain flock geometries, e.g., having followers distribute uniformly around the leader-centered annulus.

#### ACKNOWLEDGMENTS

The authors would like to thank Mike Wiatt, Chris Stankevitz, and Paul Filitchkin of Toyon Research Corporation for their help with conducting the flight experiments.

#### REFERENCES

- [1] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable flocking of mobile agents part II: Dynamic topology," in *IEEE Conf. on Decision and Control*, Maui, HI, 2003.
- [2] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 25–34, Aug. 1987. [Online]. Available: <http://doi.acm.org/10.1145/37402.37406>
- [3] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *Automatic Control, IEEE Transactions on*, vol. 51, no. 3, pp. 401 – 420, March 2006.
- [4] N. Moshtagh and A. Jadbabaie, "Distributed geodesic control laws for flocking of nonholonomic agents," *Transactions on Automatic Control*, vol. 52, no. 4, pp. 681–686, April 2007.
- [5] D. A. Paley and C. Peterson, "Stabilization of collective motion in a time-invariant flowfield," *Journal of Guidance, Control, and Navigation*, vol. 32, no. 3, pp. 771–779, May-June 2009.
- [6] R. Mellish and D. A. Paley, "Backstepping control design for motion coordination of self-propelled vehicles," in *IEEE Conf. on Decision and Control*, Atlanta, GA, December 2010.
- [7] S. Hauert, S. Leven, M. Varga, F. Ruini, A. Cangelosi, J.-C. Zufferey, and D. Floreano, "Reynolds flocking in reality with fixed-wing robots: Communication range vs. maximum turning rate," in *IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS)*, Sept. 2011.
- [8] G. E. Collins, C. R. Stankevitz, and J. Liese, "Implementation of a sensor guided flight algorithm for target tracking by small UAS," in *Ground/Air Multi-Sensor Interoperability, Integration, and Networking for Persistent ISR II*, vol. 8047. SPIE, April 2011.
- [9] [Online]. Available: <http://aviones.sourceforge.net>
- [10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.
- [11] J. Saunders and R. W. Beard, "Tracking a target in wind using a micro air vehicle with a fixed angle camera," in *American Control Conference*, Seattle, WA, June 2008, pp. 3863–3868.
- [12] Z. Li, N. Hovakimyan, V. Dobrokhodov, and I. Kaminer, "Vision-based target tracking and motion estimation using a small UAV," in *IEEE Conf. on Decision and Control*, Atlanta, GA, December 2010.
- [13] M. Wheeler, B. Schrick, W. Whitacre, M. Campbell, R. T. Rysdyk, and R. A. Wise, "Cooperative tracking of moving targets by a team of autonomous UAVs," in *IEEE / AIAA 25th Digital Avionics Systems Conference*, 2006.
- [14] S. A. P. Quintero, F. Papi, D. J. Klein, L. Chisci, and J. P. Hespanha, "Optimal UAV coordination for target tracking using dynamic programming," in *IEEE Conf. on Decision and Control*, Atlanta, GA, December 2010.
- [15] M. Mallick, "Geolocation using video sensor measurements," in *Proc. IEEE International Conf. on Information Fusion*, Quebec, Canada, July 2007.
- [16] S. A. P. Quintero and J. P. Hespanha, "Vision-based target tracking with a small UAV: Optimization-based control strategies," Note: In preparation.