



All Faculty Publications

2006-01-01

Cooperative Forest Fire Surveillance Using a Team of Small Unmanned Air Vehicles

Randal Beard
beard@byu.edu

David Casbeer

See next page for additional authors

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>

 Part of the [Electrical and Computer Engineering Commons](#)

Original Publication Citation

David W. Casbeer, Derek B. Kingston, Randal W. Beard, and Tim McLain. "Cooperative forest fire surveillance using a team of small unmanned air vehicles". *International Journal of Systems Science* (Impact Factor: 2.1). 05/2006; 37(6):351-360. DOI: 10.1080/00207720500438480

BYU ScholarsArchive Citation

Beard, Randal; Casbeer, David; Kingston, Derek; McLain, Timothy W.; Li, Sai-Ming; and Mehra, Raman, "Cooperative Forest Fire Surveillance Using a Team of Small Unmanned Air Vehicles" (2006). *All Faculty Publications*. 1228.
<https://scholarsarchive.byu.edu/facpub/1228>

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Authors

Randal Beard, David Casbeer, Derek Kingston, Timothy W. McLain, Sai-Ming Li, and Raman Mehra

Cooperative Forest Fire Surveillance Using a Team of Small Unmanned Air Vehicles *

¹David W. Casbeer, ¹Derek B. Kingston, ¹Randal W. Beard,[†]
¹Timothy W. McLain ²Sai-Ming Li, ²Raman Mehra

¹ Brigham Young University, Provo, UT 84602
² Scientific Systems Company, Inc., Woburn, MA 01801

Abstract

The objective of this paper is to explore the feasibility of using multiple low-altitude, short endurance (LASE) unmanned air vehicles (UAVs) to cooperatively monitor and track the propagation of large forest fires. A real-time algorithm for tracking the perimeter of a fire with an on-board infrared sensor is developed. Using this algorithm, we develop a decentralized multiple-UAV approach to monitoring the perimeter of the fire. The UAVs are assumed to have limited communication and sensing range. The effectiveness of the approach is demonstrated in simulation using a 6 DOF dynamic model for the UAV and numerical propagation model for the forest fire. Salient features of the approach include the ability to monitor a changing fire perimeter, the ability to systematically add and remove UAVs from the team, and the ability to supply time-critical information to forest fire fighters.

1 Introduction

Forest fires cause billions of dollars in damage to property and the environment every year. To combat forest fires effectively, their early detection and continuous tracking is vital. With the help of advanced image processing techniques, many methods have been developed to detect a forest fire in remote regions using satellite images [9, 21]. Such images are typically taken by low earth orbiting satellites

*Submitted for review to the International Journal of Systems Science, December 1, 2004

[†]Corresponding author: beard@ee.byu.edu

with an orbital period of about ten hours, and with a resolution that is only sufficient for fire detection. However, fire fighters need frequent and high-quality information updates on the progress of the fire to effectively and safely fight it. Because current forest fire monitoring techniques are deficient, fire fighters are often required to enter a fire region with little knowledge of how and where the fire is propagating, placing their lives at risk. For these reasons, there is an urgent need to develop more effective fire monitoring technologies.

High-altitude, long-endurance (HALE) UAVs such as the ALTAIR have the potential to increase image resolution and update rates over satellite based systems [2]. However, the limited availability of HALE systems during peak fire season may limit their overall effectiveness and highlights the need for lower-cost, locally implementable systems.

Low-altitude, short-endurance (LASE) UAVs are expected to be a key technology for enhanced fire monitoring. Flying at low altitude, these UAVs can capture high resolution imagery and broadcast frequent updates to fire crews. NASA is actively pursuing this possibility with ongoing research projects aimed at tracking the growth of fires using LASE UAVs [1]. However, a number of challenges have to be solved before LASE UAVs can be used for fire monitoring. First, with the fire growing and changing directions, UAVs need to be able to plan their path using limited real-time information. Second, LASE UAVs typically cannot carry enough fuel to endure a long fire fighting mission, which means that the UAV needs to have the intelligence to know when to return to the home base for refueling. Furthermore, for large forest fires, the information update rate may still be too low if only a single LASE UAV is employed.

The objective of this paper is to explore the feasibility of using multiple LASE UAVs to cooperatively monitor and track the propagation of large forest fires. By using teams of inexpensive, rapidly deployable LASE UAVs, the complexity of the system will shift from the hardware platform to the cooperative control strategies employed to coordinate fire monitoring operations. While teams of LASE UAVs will be more robust to single-point failures than a single satellite or HALE UAV, several technical challenges must be addressed to enable their successful implementation. For a large fire tracking mission, several important issues in cooperation of multiple small UAVs arise. Issues addressed in this paper include overcoming limited communication range and flight duration, developing a suitable coordination strategy for fire monitoring, and forming team consensus in the presence of noisy or uncertain information.

Spurred by recent increased interest in UAVs from the military community, research activity in the area of cooperative control of UAV systems has been high. Development of underlying cooperative control theory and strategies has led to advances in cooperative search [3, 18, 32], cooperative path planning [7], and cooperative control strategies [15, 31]. Experimental work with teams of UAVs has been limited, primarily due to the practical challenges of fielding multiple vehicles simultaneously. Several researchers have demonstrated leader following with

two UAVs [8, 19, 29]. Cooperative timing by a team of three UAVs has been demonstrated experimentally [24]. A unique characteristic of this experimental work is that the cooperation occurs in path planning and assignment rather than in trajectory tracking.

Much of the work carried out to date in the field of cooperative control requires centralized implementations of the cooperation algorithms. One of two centralized solution approaches is typically followed: 1) information is gathered from all team members to a central location where cooperative actions are computed and distributed to each member of team, or 2) information is shared among all team members which implement identical cooperation algorithms. These approaches require each team member to be in communication with a central leader or all other members of the team, and to ensure cooperation they require consistent information among all members of the team. For these reasons, decentralized approaches to cooperative control are more attractive.

A key challenge in implementing decentralized cooperation strategies is to form consensus among members of the team when communication links are intermittent or noisy and sensed information is inconsistent among team members. Recent work on consensus algorithms provide a means for convergence to consistent cooperation information among team members. In [16], sufficient conditions are given for consensus of the heading angles of a group of agents under undirected switching interaction topologies. In [28], average consensus problems are solved for a network of integrators using directed graphs. In [23], a set-valued Lyapunov approach is used to consider consensus problems with unidirectional time-dependent communication links. Using directed graphs, Refs. [25] and [27] show necessary and/or sufficient conditions for consensus of information under time-invariant and switching interaction topologies.

In this paper we present a multiple LASE UAV cooperative control solution to the forest fire monitoring problem. As a first step in this process, Section 3 develops a real-time algorithm for tracking the perimeter of a fire given the availability of an on-board infrared sensor. In Section 4 we develop the main result of the paper which is a decentralized multiple-UAV approach to monitor the perimeter of the forest fire. Salient features of this approach include (1) the ability to monitor a changing fire perimeter, (2) the ability to systematically add and remove UAVs from the team (important for re-fueling), and (3) the ability to supply time-critical information to forest fighters. In Section 5 we present simulation results. To demonstrate the effectiveness of our approach in realistic scenarios, we implemented the forest fire propagation model EMBYR [10, 13] in Simulink to generate a realistic simulation of the time-evolution of a typical forest fire. Then we use this model to verify our path planning and cooperation algorithms in a simulated forest fire scenario.

2 Problem Statement

Figure 1 shows the forest fire monitoring scenario considered in this paper. The orange pixels represent the areas where fire is burning, while the area enclosed by them represents the burnt area. A base station, represented by the red truck in Figure 1, sends out one or multiple UAVs to monitor the propagation of the fire. The objective for the UAV(s) is to capture images along the perimeter of the fire, and to upload the location of the fire perimeter (with associated imagery) to the base station as frequently and with as little delay as possible.

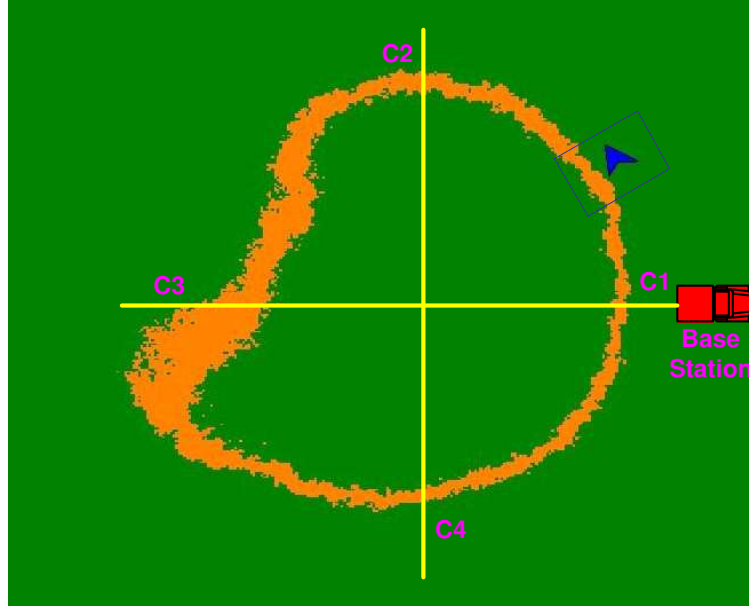


Figure 1: Fire monitoring scenario.

We make the following assumptions. First, we assume each UAV can collect or receive sufficient information on-board to plan and adjust its path autonomously. This allows the UAV to adapt its path according to the fire perimeter. In particular, each UAV is assumed to be equipped with an infrared camera that captures images of a small region beneath it, indicated by the blue rectangle in Figure 1. An infrared camera is particularly suitable for fire monitoring as it detects the regions of the ground with the highest temperature. Second, each UAV is assumed to have limited communication range, which means that it cannot upload data to the base station unless it is within a certain range of the station, and it cannot communicate with other UAVs unless they are within a certain proximity. Finally, each UAV is assumed to have limited fuel, which implies that it must periodically return to the

base station for refueling.

The delay between when the images are collected along the perimeter of the fire and when they are transmitted to the base station, can serve as a measure of the quality of the fire monitoring algorithm. Let $\delta(x, t)$ denote the latency associated with information about the position x along the perimeter at time t . As time passes, the information at the base station grows older (more latent) until a UAV arrives to transmit the latest information it has gathered. For a particular position x along the perimeter, $\delta(x, t)$ will simply increase with time until it is replaced by the data downloaded from a UAV. At each x , $\delta(x, t)$ looks like a sawtooth function where the low points are the transmissions from the UAVs and the edges are the increase in latency due to the time between UAV updates.

In this paper we will consider two performance metrics that attempt to minimize the imaging delay as seen at the base station. The first metric is the **weighted average delay**:

$$J_1(t) = \int_{\tau=t_0}^t \int_{x=0}^{P(\tau)} \beta(x) \delta(x, \tau) dx d\tau,$$

where $x = 0$ corresponds to the base station location, $P(t)$ is the length of the perimeter of the fire at time t , $\beta(x)$ is a weighting function, and t_0 is the initial time. The second metric is the **weighted maximum delay**:

$$J_2(t) = \max_{\tau \in [t-T, t]} \int_{x=0}^{P(\tau)} \beta(x) \delta(x, \tau) dx,$$

where T is a fixed time window. The weighted maximum delay J_2 is a general measure of the maximum latency incurred between updates. The optimization of these performance metrics will be subject to the fuel consumption constraint:

$$\min_j f_j(t) > f_{min},$$

where $f_j(t)$ is the fuel available on UAV j at time t , and f_{min} is the minimum allowed fuel.

3 Fire Perimeter Tracking for a Single UAV

The objective of this section is to develop a real-time algorithm for tracking the perimeter of a forest fire. In Section 4, this algorithm will be used by each UAV on the team. We assume that the UAVs are equipped with an autopilot similar to the one described in [4, 20]. We also assume that the autopilot maintains the height of the UAV at a constant altitude, and that each UAV is given a unique altitude assignment. The autopilot described in [4, 20] has been tuned so that the

closed-loop system exhibits a first order response to roll and velocity commands. Therefore the equations of motion for a single UAV can be written as

$$\dot{r}_x = V \cos \psi + w_x \quad (1)$$

$$\dot{r}_y = V \sin \psi + w_y \quad (2)$$

$$\dot{\psi} = \frac{g}{V} \tan \phi \quad (3)$$

$$\dot{V} = \alpha_V (V^c - V) \quad (4)$$

$$\dot{\phi} = \alpha_\phi (\phi^c - \phi), \quad (5)$$

where $\mathbf{r} = (r_x, r_y)^T$ is the inertial position of the UAV, ψ , ϕ , and V are the heading, roll angle, and airspeed of the UAV, g is the gravitational constant, $\mathbf{w} = (w_x, w_y)^T$ is the windspeed, and V^c and ϕ^c are the airspeed and roll angle commands given to the autopilot. The first order response of the autopilot to airspeed and roll angle commands are quantified by the parameters α_V and α_ϕ .

We will assume that the UAV is equipped with an infrared camera mounted on a pan-and-tilt gimbal similar to the one described in [6]. Our objective for the gimbal control is to servo the pan and tilt angles so that the fire perimeter divides the infrared image in half, as depicted in Figure 2. Doing so requires three steps:

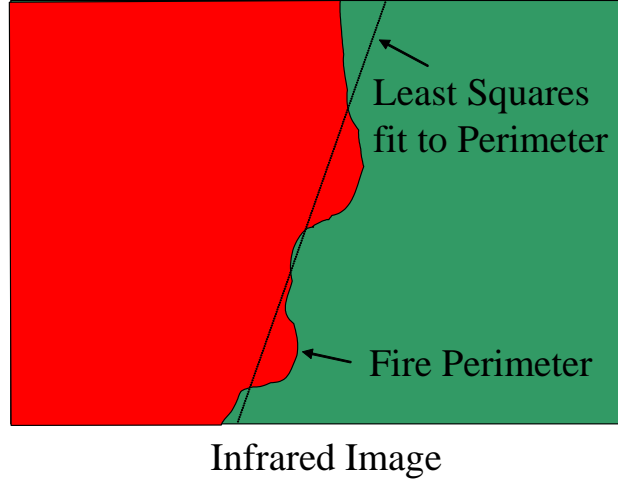


Figure 2: The pan-and-tilt gimbal is controlled such that the fire perimeter divides the infrared image roughly in half.

(1) identifying the fire perimeter in the image, (2) fitting a line to the perimeter, and (3) servoing the gimbal so that the line divides the image in half.

Identifying the fire perimeter in the infrared image can be accomplished using either gradient operators [17] or a thresholding methods [12, 30] on the most intense pixels in the image. Further analysis of actual infrared forest fire images

is needed to identify a robust method of detecting a forest fire's perimeter, and is beyond the scope of this paper. Therefore, for the purposes of this paper, we will assume that location of the fire perimeter in the image is available, and that the image processing algorithm identifies which side of the perimeter has been burned.

We next approximate the fire edge by a straight line. Doing so will make the perimeter tracking algorithm more robust to gaps in the fire due to lakes, rivers, boulder fields, and other terrain that might suppress the fire. The parameters for the straight line $y_c = ax_c + b$ are determined by the least squares estimate [5]:

$$\begin{bmatrix} a \\ b \end{bmatrix} = (\mathbf{X}_c^\top \mathbf{X}_c)^{-1} \mathbf{X}_c^\top \mathbf{Y}_c \quad (6)$$

where the first column of the matrix \mathbf{X}_c consists of the x coordinates of the perimeter pixels and the second column is ones. \mathbf{Y}_c is a vector of y coordinates of fire edge pixels. If the matrix $(\mathbf{X}_c^\top \mathbf{X}_c)^{-1}$ has a large condition number (*i.e.*, the line is almost parallel to the x -axis), we find the line by averaging the y values, giving the equation $y_c = \frac{1}{N} \text{sum}(\mathbf{Y}_c)$, where the function $\text{sum}(\cdot)$ is simply the element-wise summation of the vector.

The next step is to command the gimbal so that the line is in the center of the image. There are four coordinate frames that are involved in the problem of servoing the gimbal. We will use a subscript on a vector to denote its coordinate frame. In particular, the subscripts i , v , b , and c denote the inertial, vehicle, body, and camera frames respectively. In the camera frame, the x -axis points along the view axis of the camera, the y -axis is located in the image plane and points to the right as seen by a person viewing the image. The z -axis is also located in the image plane and points down as seen by a person viewing the image. Assuming that the body and camera frames are originally aligned, define the gimbal azimuth angle α_{az} to be the rotation angle about the z -axis in the camera frame. The gimbal elevation angle is defined to be the subsequent rotation about the y -axis in the rotated camera frame.

We will assume that the UAV is equipped with a height-above-ground sensor that returns the height-above-ground h . Let $(y_{\text{img}}, z_{\text{img}})$ be the image coordinates of the center of the fire perimeter line. If the footprint of the infrared camera is on flat ground, then from projective geometry [22], the relative position vector to the physical location that corresponds to the center of the perimeter line is given by

$$\mathbf{p}_c = \begin{pmatrix} h \sin(\alpha_{az} - \phi) \sin(\alpha_{el} - \theta) \\ y_{\text{img}} f h \sin(\alpha_{az} - \phi) \sin(\alpha_{el} - \theta) \\ z_{\text{img}} f h \sin(\alpha_{az} - \phi) \sin(\alpha_{el} - \theta) \end{pmatrix},$$

where f is the focal length of the camera, and ϕ and θ are the roll and pitch angles

of the UAV respectively. Resolving \mathbf{p}_c in the body frame we get

$$\begin{aligned}\mathbf{p}_b &= R_{c \rightarrow b} \mathbf{p}_c \\ &= \begin{pmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{pmatrix} \cdot \\ &\quad \begin{pmatrix} h \sin(\alpha_{az} - \phi) \sin(\alpha_{el} - \theta) \\ y_{\text{img}} f h \sin(\alpha_{az} - \phi) \sin(\alpha_{el} - \theta) \\ z_{\text{img}} f h \sin(\alpha_{az} - \phi) \sin(\alpha_{el} - \theta) \end{pmatrix}, \end{aligned} \quad (7)$$

where $c\varphi = \cos(\varphi)$ and $s\varphi = \sin(\varphi)$. As shown in [6], the gimbal angles that place \mathbf{p}_b in the center of the image are given by

$$\begin{aligned}\alpha_{az}^c &= \tan^{-1} \left(\frac{p_{by}}{p_{bx}} \right) \\ \alpha_{el}^c &= \sin^{-1} \left(\frac{p_{bz}}{\|\mathbf{p}_b\|} \right),\end{aligned}$$

where $\mathbf{p}_b = (p_{bx}, p_{by}, p_{bz})^T$.

The algorithm described above will fit a line to the fire perimeter detected in the infrared camera image. Given the line, it is trivial to find the two pixel locations where the line intersects the boundary of the image. Let \mathbf{p}_{b1} and \mathbf{p}_{b2} represent the corresponding points in the body frame computed using Equation (7). Therefore, the physical location of the least squares fit to the fire perimeter, referenced to the body frame, is given by $\sigma \mathbf{p}_{b2} + (1 - \sigma) \mathbf{p}_{b1}$ where $0 \leq \sigma \leq 1$.

The objective is to implement a pursuit algorithm that continually tracks this line. The first step in the implementation of the pursuit algorithm is to integrate Equations (1)–(5) from the current position and heading, T seconds into the future. Integration yields

$$\bar{\mathbf{r}}(t, T, \phi) = \mathbf{r}(t) + \frac{V^2}{g \tan \phi} \begin{pmatrix} \sin(\psi(t) + T \frac{g}{V} \tan \phi) - \sin \psi(t) \\ -\cos(\psi(t) + T \frac{g}{V} \tan \phi) + \cos \psi(t) \end{pmatrix} + T \begin{pmatrix} w_x(t) \\ w_y(t) \end{pmatrix}, \quad (8)$$

where $\bar{\mathbf{r}}(t, T, \phi)$ is the estimated position of the UAV at time $t + T$ assuming that it holds a constant roll angle ϕ during that time. The reachable set T seconds into the future is given by

$$\mathcal{R}_T = \bigcup_{-\bar{\phi} \leq \phi \leq \bar{\phi}} \bar{\mathbf{r}}(t, T, \phi),$$

where $\bar{\phi}$ is the roll limit imposed by the airframe. The commanded roll angle is selected to minimize the distance between the reachable set at time T and the least

squares fit to the fire parameter at time t :

$$\phi^c = \arg \min_{-\bar{\phi} \leq \phi \leq \bar{\phi}} \left\{ \min_{0 \leq \sigma \leq 1} \|\bar{\mathbf{r}}(t, T, \phi) - [\sigma \mathbf{p}_{b1} + (1 - \sigma) \mathbf{p}_{b2}]\|^2 \right\}.$$

4 Cooperative Team Tracking

For fire fighters on the perimeter of a fire, frequent updates about the progression of the fire is critical for safety. This section develops a distributed monitoring scheme that allows perimeter information to be transmitted as often as possible to checkpoints along the perimeter of the fire. Our objective is to design a distributed algorithm that makes available the most current information regarding the progression of the perimeter of the fire at a high update rate along the entire fire front.

To aid the development of a cooperative perimeter monitoring scheme, we will first consider fires of fixed perimeter length. For ease of analysis and visualization, the fire perimeter will be assumed to be circular. However, the algorithm applies in the general case. In addition, because fire fighters may be located at any point along the perimeter of the fire, we will assume that all points along the fire have equal weight, i.e. $\beta(x) = c > 0$. Finally, we will assume that all UAVs fly around the perimeter of the fire with constant velocity.

4.1 Performance Metric Minimization

When a UAV transmits its notion of the state of the fire to the base station, an associated latency profile (in terms of time-stamped images) accompanies the data. Let the latency associated with a point x at the time of the base station update be denoted $\rho(x)$. In other words, $\rho(x)$ returns the associated latency for information gathered at the position x along the perimeter of the fire at the time of transmission to the base station. Note that $\delta(x, t) = \rho(x)$ when a UAV updates the base station and $\delta(x, t) = \rho(x) + (t - t_{\text{update}})$ between updates. The objective is to design a cooperative monitoring scheme that minimizes $\rho(x)$ for every x and updates the base station as often as possible.

Consider the latency profile around the perimeter of the fire when only one UAV is used. Figure 3 shows the latency associated with the perimeter of the fire when a counter-clockwise flying UAV arrives back at the base station after a complete tour of the perimeter, where the thickness of the path denotes the latency of the base station update of that point when it arrives back at the base station. Since the state of the fire is transmitted only after the UAV has traversed the entire fire perimeter, the greatest latency is associated with data gathered at the beginning of the flight near the base station. Because the UAV is traveling at constant velocity, the latency profile is a linear function of the distance traveled, $\rho(x) = (P - x)/v$,

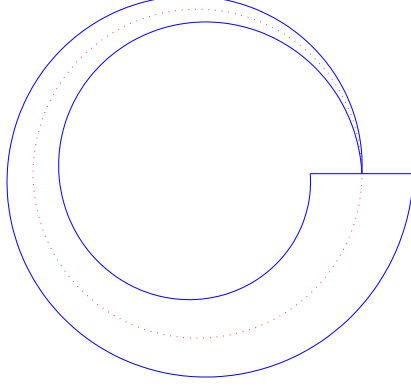


Figure 3: Latency profile for a single UAV monitoring a static circular fire. The thickness of the path denotes the latency of information at that point when it is transmitted to the base station.

where v is the velocity of the UAV. The base station receives updates only as fast as the UAV can traverse the entire fire perimeter. The total latency associated with one traversal is given by $\int_0^P \rho(x) = 0.5P^2/v$.

With a pair of UAVs flying in opposite directions, the update rate is the same (since both UAVs arrive back at the base station at the same time), but the latency associated with the information on both sides of the base station is symmetric and reduced, as can be seen in Figure 4. Here, the latency profile is given by

$$\rho(x) = \begin{cases} \frac{x}{v} & \text{for } 0 \leq x \leq P/2 \\ \frac{P-x}{v} & \text{for } P/2 < x \leq P \end{cases}$$

and the overall latency associated with the scheme is $\int_0^P \rho(x) = 0.25P^2/v$, which is a factor of 1/2 better than the single UAV case.

Lemma 4.1 *For UAVs that follow the perimeter and fly at constant velocity, the latency profile shown in Figure 4 is the minimum possible latency for every x along the perimeter of the fire.*

Proof: The minimum latency associated with data gathered at x on the perimeter is the time needed to travel from x to the base station along the shortest path. Since pairs of UAVs meet at the base station having traversed the entire perimeter in both directions, each transmits the data associated with the half of the perimeter

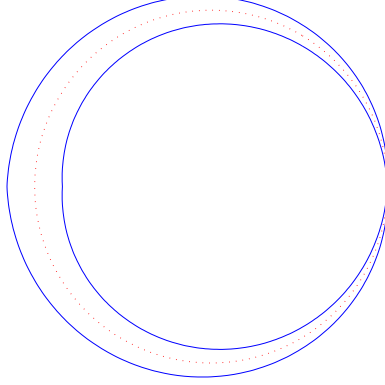


Figure 4: Latency profile with a pair of UAVs monitoring a static circular fire in opposite directions.

that was traversed most recently. The composition of this data is the least latent profile of the entire perimeter since both UAVs gathered data along their respective halves of the perimeter and returned to the base station with the shortest possible paths. ■

A consequence of Lemma 4.1 is that adding more than two UAVs will not improve the overall latency in the update. However, the rate at which updates occur will increase linearly with the number of UAV pairs employed. To maintain the minimum latency profile and to minimize the time between updates at the base station, pairs of UAVs should be launched so that they are uniformly spaced about the perimeter of the fire in each direction.

To minimize the performance metrics introduced in Section 2, we must simultaneously minimize the latency profile and the time between updates. In the two-UAV case, we showed that the minimum possible latency profile is obtained when the UAVs arrive back at the base station at the same time, but the update rate remains the same as in the one-UAV case. What influence does increasing the update rate, at the expense of the latency profile, have on the performance metrics? Consider a scenario where two UAVs are monitoring a fire and the base station is situated between the rendezvous points of the UAVs. The base station will be updated by each UAV at different times, so the update rate will be double that of the scenario shown in Figure 4. However, the latency profile will not be optimal.

To compare the effect that moving the base station has on the performance metrics, let the nominal case be defined as two UAVs arriving at the base station simultaneously. In this case, the latency profile is the same at each update (the minimum possible) and update times occur at $t_n = t_{n-1} + P/v$, where v is the velocity of the UAVs. For comparison, consider the scenario where the base station

is located at $x_0 \in (0, P/4]$. There will be two different latency profiles transmitted to the base station - one by the clockwise moving UAV and one by the counter-clockwise moving UAV. There will also be pairs of update times associated with each of the UAVs, so $t_n = t_{n-2} + P/v$ and $t_n - t_{n-1} = x_0/v$ for n even and $t_n - t_{n-1} = (P/2 - x_0)/v$ for n odd. Figure 5 shows the latency profiles for counter-clockwise and clockwise updates when the base station is located at $P/8$. Note that when the two UAVs meet, each updates its latency profile to the mini-

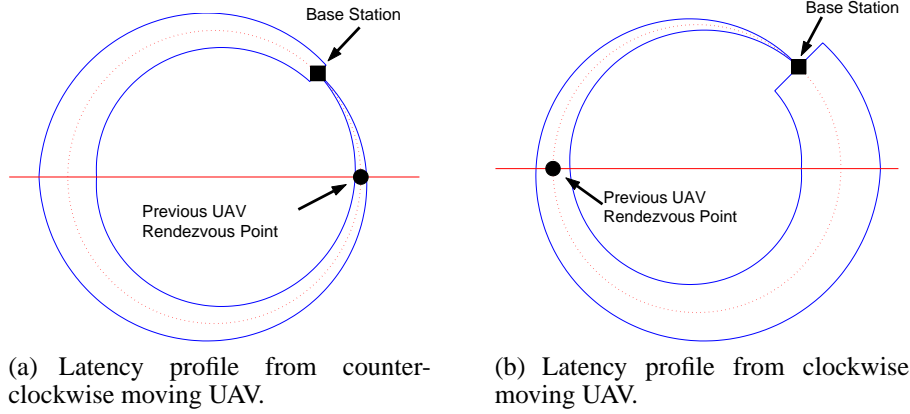


Figure 5: Graphical representation of the latency at each point along the perimeter when base station is located at $P/8$.

mum latency profile. By the time a UAV gets to the base station, it has collected new, low-latency data between the rendezvous point and the base station, but the remaining data is degraded by the time it took to travel from the rendezvous point to the base station.

The overall latency associated with a latency profile is

$$L = \int_0^P \rho(x).$$

For the nominal case, $L_{\min} = 0.25P^2/v$. When the base station is located at $x_0 \in (0, P/4]$, the overall latency can be computed graphically. Consider the case when $x_0 = P/8$. The overall latency can be computed from the area of each triangular region in Figure 6. The geometry extends to an arbitrary $x_0 \in (0, P/4]$ and the overall latency associated with the profile transmitted by the counter-clockwise traveling UAV is

$$L_{\text{ccw}} = L_{\min} + P \frac{x_0}{v} - x_0 \frac{x_0}{v}$$

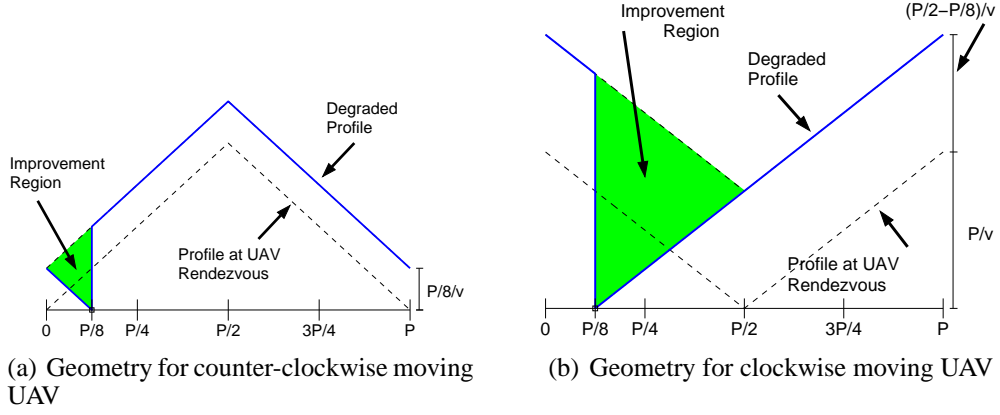


Figure 6: Overall latency geometry when base station is located at $P/8$.

and the overall latency associated with the clockwise traveling UAV is

$$L_{cw} = L_{\min} + P \frac{P/2 - x_0}{v} - \frac{(P/2 - x_0)^2}{v}.$$

The expressions for L_{ccw} and L_{cw} are composed of three terms: (1) a minimum latency profile term associated with the rendezvous of the two UAVs before visiting the base station, (2) the degradation of that minimum profile during the flight to the base station, and (3) the amount of improvement to the latency profile during the flight to the base station. Let $t_0 = 0$ and $t = P/v$ (corresponding to one time around the fire), then the **weighted average delay** for the nominal case (with fixed perimeter length P and equal perimeter weighting $\beta(x) = c > 0$) is

$$\begin{aligned} J_{1_{\text{nom}}} &= \int_{\tau=0}^{\frac{P}{v}} \int_{x=0}^P c \delta(x, \tau) dx d\tau \\ &= c \int_{\tau=0}^{\frac{P}{v}} \int_{x=0}^P [\rho_{\min}(x) + \tau] dx d\tau \\ &= c \int_0^{\frac{P}{v}} [L_{\min} + P\tau] d\tau \\ &= cL_{\min} \frac{P}{v} + cP \int_0^{\frac{P}{v}} \tau d\tau. \end{aligned}$$

For the case where the base station is at $x_0 \in (0, P/4]$, the **weighted average**

delay is

$$\begin{aligned}
J_{1x_0} &= \int_{\tau=0}^{\frac{P}{v}} \int_{x=0}^P c\delta(x, \tau) dx d\tau \\
&= c \int_{\tau=0}^{\frac{P-2x_0}{v}} \int_{x=0}^P [\rho_{\text{ccw}}(x) + \tau] dx d\tau \\
&\quad + c \int_{\tau=\frac{P-2x_0}{v}}^{\frac{P}{v}} \int_{x=0}^P \left[\rho_{\text{cw}}(x) + \left(\tau - \frac{P-2x_0}{v} \right) \right] dx d\tau \\
&= c \int_0^{\frac{P-2x_0}{v}} [L_{\text{ccw}} + P\tau] d\tau \\
&\quad + c \int_{\frac{P-2x_0}{v}}^{\frac{P}{v}} \left[L_{\text{cw}} + P\tau - P \frac{P-2x_0}{v} \right] d\tau \\
&= cL_{\text{ccw}} \frac{P-2x_0}{v} + c \left(L_{\text{cw}} - P \frac{P-2x_0}{v} \right) \frac{2x_0}{v} + cP \int_0^{\frac{P}{v}} \tau d\tau.
\end{aligned}$$

The difference between the nominal and the x_0 case becomes

$$J_{1x_0} - J_{1\text{nom}} = c \frac{2x_0}{v^2} (P - 2x_0) (P - 4x_0),$$

which is greater than 0 for $x_0 \in (0, P/4)$ and equal to 0 for $x_0 = P/4$. The symmetry of the problem allows the same formulation for x_0 in any quadrant and we conclude that the **weighted average delay** is minimum for the nominal case and the case when the base station is located $P/4$ from a rendezvous, i.e. exactly between UAV rendezvous points.

The **weighted maximum delay** expressions can also be compared for $t = T = P/v$.

$$\begin{aligned}
J_{2\text{nom}} &= \max_{\tau \in [0, P/v]} \left\{ \int_{x=0}^{P(\tau)} c\delta(x, \tau) dx \right\} \\
&= c \max_{\tau \in [0, P/v]} \left\{ \int_{x=0}^{P(\tau)} [\rho_{\min}(x) + \tau] dx \right\} \\
&= cL_{\min} + cP \max_{\tau \in [0, P/v]} \{\tau\} \\
&= c \left(L_{\min} + \frac{P^2}{v} \right)
\end{aligned}$$

$$\begin{aligned}
J_{2x_0} &= \max_{\tau \in [0, P/v]} \left\{ \int_{x=0}^{P(\tau)} c\delta(x, \tau) dx \right\} \\
&= c \max \left\{ L_{\text{ccw}} + P \max_{\tau \in [0, \frac{P-2x_0}{v}]} \{\tau\}, L_{\text{cw}} + P \max_{\tau \in [0, 2x_0/v]} \{\tau\} \right\} \\
&= cL_{\min} + \frac{c}{v} \max \left\{ (P - x_0)(P + x_0) - Px_0, 2Px_0 + \left(\frac{P}{2} + x_0\right) \left(\frac{P}{2} - x_0\right) \right\} \\
&= c \left(L_{\min} + \frac{1}{v} [P^2 - x_0^2 - Px_0] \right)
\end{aligned}$$

The difference between the x_0 case and the nominal case is

$$J_{2\text{nom}} - J_{2x_0} = c \frac{x_0}{v} (P + x_0),$$

which achieves a maximum at $x_0 = P/4$ since $x_0 \in (0, P/4]$. Therefore, the **weighted maximum delay** is minimized when the base station is located exactly in between UAV rendezvous points.

Theorem 4.2 *Given N pairs of UAVs flying at constant velocity around a fixed-size fire perimeter, the UAV configuration that minimizes **weighted average delay** and **weighted maximum delay** for $\beta(x) = c$ is the configuration where N uniformly spaced UAVs are traveling clockwise around the perimeter and N uniformly spaced UAVs are traveling counter-clockwise and the base station is located precisely in between adjacent UAV rendezvous points.*

Proof: When $N = 1$, the above arguments show that the performance metrics are minimized. By regarding the segment of the perimeter between adjacent rendezvous points as a fire perimeter in its own right, it is obvious that the performance metrics are minimized in the same fashion. ■

4.2 A Distributed Fire Monitoring Algorithm

Developing a distributed algorithm that causes the members of the UAV team to cover equal lengths of the fire perimeter is complicated by the initial locations of the UAVs and the expansion or contraction of the fire perimeter. The algorithm must adjust the rendezvous locations between two neighboring UAVs without knowledge of the true fire perimeter length. An distributed algorithm has been developed to accomplish this objective. It operates under the assumption that pairs of UAVs are launched in opposite directions simultaneously.

A few definitions are needed before presenting the algorithm. The variable ℓ_c is the distance traveled since the last rendezvous. This variable is updated

by constantly integrating the distance traveled and is assumed to be updated for the algorithm. ℓ_1 is the previous distance traveled between rendezvous (*i.e* from rendezvous ago till the last rendezvous). Let dir indicate the current direction of travel around the perimeter either clockwise or counter-clockwise. Each UAV holds two variables for adjusting rendezvous locations, τ_{cw} and τ_{ccw} . τ_{cw} is used to specify the distance from the last rendezvous the UAV should travel before it begins loitering if it is headed in a clockwise direction, similarly τ_{ccw} is the distance to begin loitering from the last rendezvous if the UAV is traveling in the counter-clockwise direction. To ensure rendezvous between UAVs with a growing fire, we let $\tau_{dir} = \infty$ so the UAV will not loiter, rather it will continue traveling until it rendezvous with its neighbor. Lastly τ is calculated for each rendezvous by averaging ℓ_c . It is the distance that must be traveled after the next rendezvous. The algorithm is shown below.

Algorithm 1: Distributed spread

ℓ_c is the distance traveled (constantly integrated outside the algorithm)

Initialize (when UAV arrives at fire perimeter)

$\ell_c = 0$ distance traveled since last rendezvous

$\ell_1 = 0$ distance traveled between last two rendezvous

To ensure a rendezvous we set distance to loiter to $\tau_{cw} = \infty$, $\tau_{ccw} = \infty$

dir = the direction of launch for this UAV either cw or ccw

while 1 do

```
1   if  $\ell_c \geq \tau_{dir}$  then Wait here for next rendezvous
2   if rendezvous (i.e. my neighbor is within communication range) then
3       if (this is my first rendezvous) then  $\ell_1 = \ell_c$ 
4       Send  $\ell_c$  to neighbor
5        $\ell_p$  = (the  $\ell_c$  received from neighbor)
6       Average the distance traveled since the UAVs last rendezvous
        $\tau = \frac{1}{2}(\ell_c + \ell_p)$ 
7       Determine who should loiter for this pair's next rendezvous
       if  $\ell_c < \ell_p$  (I traveled less) then  $\tau_{dir} = \tau$  (I should loiter)
       else if  $\ell_c < \ell_p$  (I traveled more) then  $\tau_{dir} = \infty$  (neighbor will loiter)
       else ( $\ell_c = \ell_p$  equal distance traveled) They must communicate and
       decide how to set  $\tau_{dir}$ 
8       Switch  $dir$  (For example: if ccw make it cw)
9       Calculate distance to loiter for next rendezvous
        $\tau_{dir} = \tau_{dir} + (\ell_c - \ell_1)$ 
10       $\ell_1 = \ell_c$ ,  $\ell_c = 0$ 
```

The initialization of the algorithm occurs when the UAVs reach the perimeter of the fire. At this time, the UAV defaults to continue until the next rendezvous and not loiter. This is done by setting the distance to loiter in both directions to infinity ($\tau_{cw} = \infty$, $\tau_{ccw} = \infty$). In general operation, when two UAVs rendezvous, they communicate the distance they have traveled since the last rendezvous to their neighbor (step 4). If the distances traveled do not coincide, then the UAV that has traveled the shorter length (smaller ℓ_c) is required to adjust its distance to loiter so that when this pair meets again for their next rendezvous the UAV that travelled the shorter distance will be the one to wait at the designated location. This adjustment is given to the UAV that traveled the shorter distance because in most cases it will arrive at the designated location first. Notice, the actual distance to loiter

for a given direction is calculated from the average of the distances traveled (step 6) and is set only after a rendezvous has occurred in the opposite direction (step 9).

To explain why the distance to loiter must be set after the next rendezvous the following example is provided. Let the current and past rendezvous be called r_0 and r_{-1} respectively. Also, label the next two rendezvous as r_1 and r_2 again respectively with time. At r_0 and r_2 the same UAVs will meet. The distance to loiter, τ from r_1 to r_2 assumes that the rendezvous at r_{-1} and r_1 occurred at the same location. However, this is not always the case. Therefore the term $l_c - l_1$ in step 9 adjusts the distance to loiter to account for this discrepancy.

For the case in step 7, when the UAVs have traveled an equal distance from their last rendezvous, there are multiple ways to decide how to set τ_{dir} that will work. One method is to have both UAVs continue until their next rendezvous without loitering (set $\tau_{dir} = \infty$ for both UAVs). For the simulations provided in Section 5, the UAV that is closest to the base station is set to loiter $\tau_{dir} = \tau$, while the other is set to continue until the next rendezvous.

Figure 7 shows an example in which application of Algorithm 1 allows the UAV team to spread out evenly along the perimeter of the fire. P is the perimeter length of the fire/circle in this figure. Two pairs of UAVs are launched with the first pair leading the second by two time units (Figures 7(a) and 7(b)). After four time units have passed, the first pair of UAVs meet and each transmits $\ell_c = \frac{1}{2}P$ (Figure 7(c)). Since both are equal, the decision was for neither UAV to loiter in the future. After the first pair rendezvous and double-back, the second pair meets the first pair at $t = 5$ (Figure 7(d)). By this time, UAVs 1 and 2 have traveled $\frac{1}{8}P$ and UAVs 3 and 4 have traveled $\frac{3}{8}P$. Since the first pair has traveled the least, they will decide to loiter after traveling $\tau = \frac{1}{4}P$ from their next rendezvous. Both pairs leave the rendezvous location in the opposite direction in which they entered. At $t = 6$, the first pair meets again (Figure 7(e)). Since both have traveled $\frac{1}{N}P$ from their last respective rendezvous points, then neither will loiter after the next rendezvous. However, due to the decision at the previous rendezvous, UAV 1 and 2 must travel τ_{dir} based on line 9 in Algorithm 1 before beginning to loiter. In this case the fire has not expanded or contracted so each will travel for $\frac{1}{4}P$ and then begin to loiter. At $t = 8$, the first pair begins loitering and the second pair meets (Figure 7(f)). UAV 3 and 4 have both traveled the same distance, so neither will loiter after the next rendezvous. Figure 7(g) shows the time when the two pairs of UAVs meet and have equal ℓ_c values. Thereafter, the UAVs are in the optimal configuration to minimize overall latency and provide the fastest possible update rate. This pattern is maintained due to the fact that each UAV will travel exactly $\frac{1}{4}P$ between rendezvous and none will begin to loiter (Figures 7(h) and 7(i)).

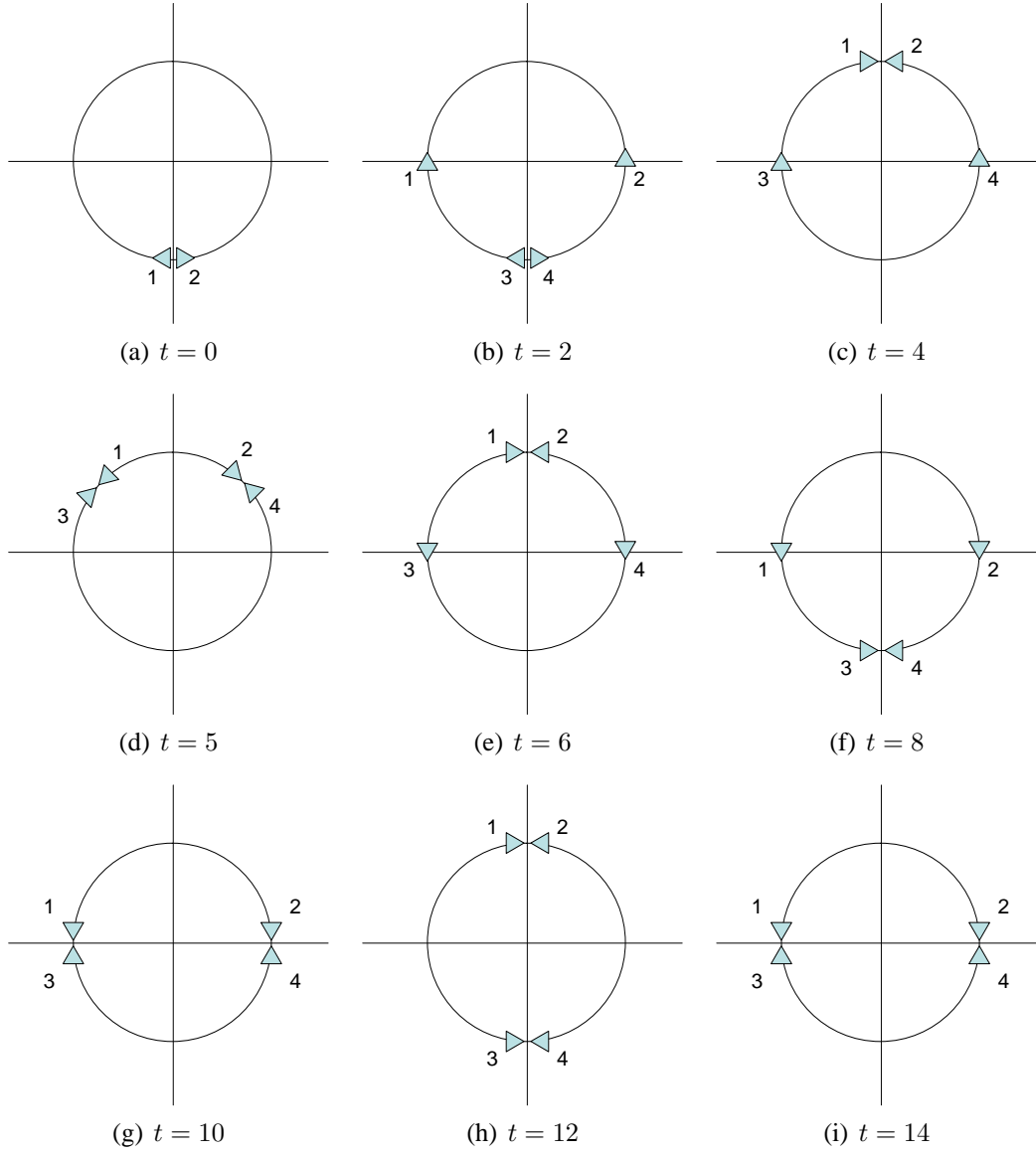


Figure 7: Example scenario in which UAV spread is adjusted by Algorithm 1

4.3 Analysis

In this section we show that Algorithm 1 converges to a configuration where all of the UAVs are equally spaced around the perimeter of the fire. To do so, we will show that the fire surveillance problem reduces to a consensus algorithm and then invoke the results presented in [25, 27].

Throughout this section we will assume that there are an even number of UAVs. For notational simplicity, number the UAVs sequentially around the perimeter of the fire and suppose that during the k^{th} communication event k -odd, there is a bidirectional communication channel between UAV i and UAV $i + 1$, where i is an odd integer, i.e., UAV 1 communicates with UAV 2, UAV 3 communicates with UAV 4, etc. Equivalently, during the k^{th} communication event, k -even, there is a bidirectional communication channel between UAV i and UAV $i + 1$, where i is an even integer, i.e., UAV 2 communicates with UAV 3, UAV N communicates with UAV 1, etc. Each UAV maintains a copy of the distance that they will travel during the next interval, which we denote as $\ell(k)$. According to Algorithm 1 when UAV i communicates with UAV $i + 1$, they update their expected travel distance as

$$\begin{aligned}\ell_i(k+1) &= \frac{\ell_i(k) + \ell_{i+1}(k)}{2} \\ \ell_{i+1}(k+1) &= \frac{\ell_{i+1}(k) + \ell_i(k)}{2}.\end{aligned}$$

To capture the fact that different updates are used for odd and even communication events we can write

$$\ell_i(k+1) = \frac{\ell_i(k) + g_{\text{odd}}(k)\ell_{i+1}(k) + g_{\text{even}}(k)\ell_{i-1}(k)}{2}, \quad (9)$$

where $g_{\text{odd}}(k) = 1$ when k odd and zero otherwise, and $g_{\text{even}}(k) = 1$ when k is even and zero otherwise, and the indices are interpreted modula N (i.e., $N+1 = 1$, $1-1 = N$). Alternatively we could write Equation (9) as

$$\ell_i(k+1) = \frac{\ell_i(k) + g_{\text{odd}}(k)\ell_{i+1}(k) + g_{\text{even}}(k)\ell_{i-1}(k)}{1 + g_{\text{odd}}(k) + g_{\text{even}}(k)}, \quad (10)$$

which is the form of the consensus equations given in [25, 27].

Note that there are two communication topologies where switching occurs between the topologies at every communication event. The two communication topologies are shown in Figure 8. The main result from [25, 27] for discrete-time consensus is the following theorem.

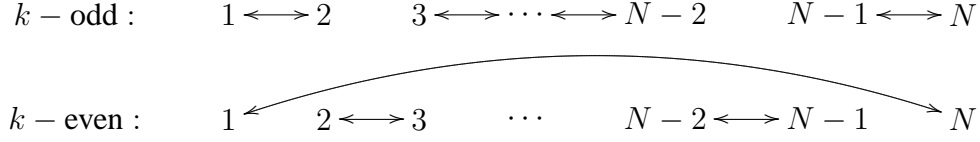


Figure 8: Communication topologies for odd and even communication events.

Theorem 4.3 *Let $\mathcal{G}[k] \in \bar{\mathcal{G}}$ be a switching interaction graph at time $t = kT$. Also let $\alpha_{ij}[k] \in \bar{\alpha}$, where $\bar{\alpha}$ is a finite set of arbitrary positive numbers. The discrete update scheme*

$$\xi_i[k+1] = \frac{1}{\sum_{j=1}^n \alpha_{ij}[k] G_{ij}[k]} \sum_{j=1}^n \alpha_{ij}[k] G_{ij}[k] \xi_j[k], \quad (11)$$

achieves consensus asymptotically for \mathcal{A} if there exists an infinite sequence of uniformly bounded, non-overlapping time intervals $[k_j T, (k_j + l_j) T)$, $j = 1, 2, \dots$, starting at $k_1 = 0$, with the property that each interval $[(k_j + l_j) T, k_{j+1} T)$ is uniformly bounded and the union of the graphs across each interval $[(k_j + l_j) T, k_{j+1} T)$ has a spanning tree.

We can use this theorem to prove that Algorithm 1 causes the UAVs to distribute themselves equally around the perimeter of the fire.

Corollary 4.4 *Given the update strategy described in Equation (10), if P is constant then the UAVs distribute themselves equally around the perimeter of the fire. In other words,*

$$\ell_i(k) \rightarrow P/N, \quad i = 1, \dots, N.$$

Proof:

Note that Equation (10) is a special case of Equation (11) where $\alpha_{ij} = 1$ and $G_{ij}[k]$ is either zero, $g_{\text{odd}}(k)$ or $g_{\text{even}}(k)$. We also note that the union of the communication graphs shown in Figure 8 is a bidirectional ring and therefore has a spanning tree. Since the systems switches between the two graphs at every time interval, all of the conditions of Theorem 4.3 are satisfied, which implies that asymptotic consensus is achieved. In other words $\ell_i(k) \rightarrow \ell_j(k)$ as $k \rightarrow \infty$. Since Algorithm 1 ensures that $\sum_{j=1}^N \ell_j(k) = P$ where P is the length of the perimeter, we must have $\ell_i(k) \rightarrow P/N$. ■

Figure 9 illustrates the operation of the update scheme for eight UAVs and a perimeter of 80 km.

Since the effects of initial conditions die out as time progresses, pairs of UAVs may enter or exit the team for refueling without affecting the long term stability

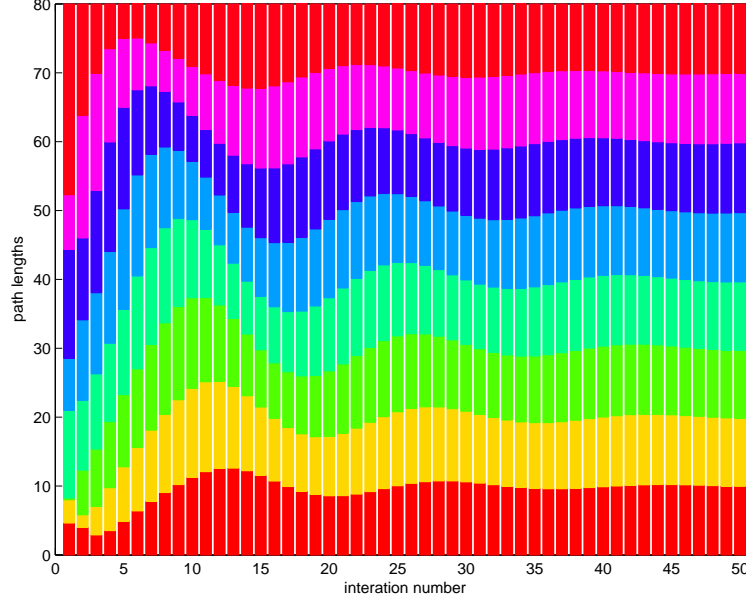


Figure 9: Path length equalization using neighbor averaging.

of the algorithm. Additionally, changes in the size of the fire perimeter enter as a disturbances to the system, however, as shown in [26] the consensus scheme given by Equation (11) is input-to-state stable which implies that finite changes in the perimeter will be tracked by the algorithm.

5 Simulation Results

In this section we describe simulation results that highlight the effectiveness of the algorithms developed in this paper. In Section 5.1 we describe the fire model that is used in all of our simulations. In Section 5.2 we present simulation results for the perimeter tracking algorithm described in Section 3. Section 5.3 presents the main cooperative fire tracking results.

5.1 Fire Model

To effectively test the perimeter tracking and cooperative control algorithms, we have developed a realistic, time-varying fire simulator. The Ecological Model for Burning in the Yellowstone Region (EMBYR) [14, 11] was chosen for this purpose.

EMBYR divides the region of interest into a grid of cells, each with inherent properties that affect the spread of the fire. These properties include the type of foliage, the moisture level, and the elevation. At a given time step, the fire will spread from a burning cell to a non-burning cell according to an independent stochastic event that is a function of the cell's properties.

To determine the length of each time step, we define the maximum speed of the UAVs to be $V_{\max} \triangleq 30 \text{ mph} = 13.4 \text{ m/sec}$. Given V_{\max} , it will take 8.7 min for an UAV to travel 7 km. We also define a circle with a radius of 100 cells to have a circumference of 7 km, which makes one cell 11.1 m wide. Letting the maximum speed of the fire be $V_{\text{fire}} = 10 \text{ mph} = 4.47 \text{ m/sec}$ and since the fastest moving fire advances at about 80 cells per time step each time step is found to be $T_{\text{step}} \approx 200 \text{ sec}$.

By running EMBYR multiple times and averaging the result we can achieve realistic fire simulations like the one shown in Figure 10. Image processing techniques easily enable the edge of the fire and the burning region to be found. This information is then passed to the UAV for perimeter tracking.

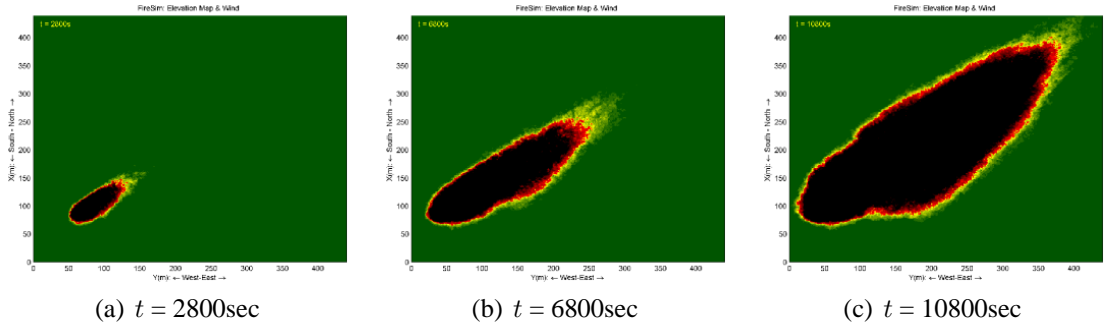


Figure 10: Fire simulation of high wind conditions with an elevation gradient. The fire is spreading in the direction of the wind.

5.2 Perimeter Tracking

As mentioned in Section 3, the only information needed by a UAV, when tracking the perimeter, is the fire's perimeter and side of the edge that is burning. To speed computation, the fire's edge has been computed ahead of time and is two pixels thick. The x, y coordinates of the internal and external parts of the edge are stored separately. An example of a single UAV tracking the fire perimeter is shown in Figure 11.

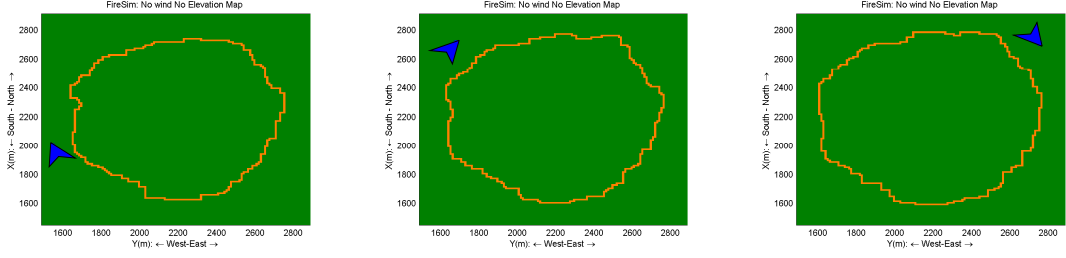


Figure 11: One UAV tracking the fire perimeter (no wind on flat ground).

5.3 Cooperative Tracking

In this section we describe the results of using multiple UAVs to monitor the perimeter of the fire. The communication range for these simulations was set at 1 km (approximately 9 pixels). In Figure 12 the fire perimeter is growing at a rate of about 2.8 m/s. Subfigure (a) contains four UAVs monitoring the fire with two more UAVs approaching (bottom left) from the base station. A short time later, as shown in Subfigure (b) the two new UAVs (also at the bottom left) are loitering while waiting for their next rendezvous. After approximately 600 s have passed the UAVs are in the configuration as shown in Subfigures (c) and (d).

As discussed in Section 4.2, the lengths, $\ell_i(k)$, converge to equal lengths. It is obvious from the figure that convergence in the steady state is only loosely asymptotic. This fact can be attributed to a few factors. First, the UAVs cannot turn instantaneously. The autopilot chooses the closest feasible point to the desired location at each time step. Each turn can differ by a matter of seconds depending on the position of the UAV. For example, a loitering UAV at rendezvous time might be headed in the correct direction and immediately start off after communicating with its neighbor, while the neighbor with whom it has just communicated might need to turn around. Depending on the fire's perimeter and the heading of the UAV, the autopilot, due to tracking the fire and chatter in deciding which way to turn, might take up to 5 s to reverse directions. By the time the second UAV has reversed directions, the first UAV may have already traveled 100 m.

A second reason for the loose convergence is due to the growth of the fire. A UAV will travel the distance it calculated previously, and the fire perimeter will have grown. At $t = 600$ s each UAV will cover one sixth of the perimeter, which will take about 60 s. Meaning the perimeter will have grown by about 170 m since the last meeting time. Each UAV and its neighbor will then differ in the length they have traveled by approximately 60 m.

Despite these factors the lengths $\ell_i(k)$ converge as time progresses. In Figure 13, it can be seen that these lengths are increasing as time progresses due to the growth of the fire.

Figures 14 and 15 also show the convergence of $\ell_i(k)$ for a static fire with

perimeter of length 7.2 km, which is shown in 16. In Figure 14 there are four UAVs initially monitoring the fire and at $t = 1000$ s two more UAVs are introduced. In Figure 15 there are six UAVs monitoring the fire.

The checkpoints in Figure 16 were used to show the latency of information at the base station. A UAV gathers information about a checkpoint when it is within 100 m from that point. This information is passed to its neighbor when the UAVs communicate. When a UAV passes the base station (half way between the left and bottom checkpoints) the information concerning all the checkpoints known by that UAV is passed to the base station. The static perimeter is 72.5 km long, which would take about six minutes to traverse. The maximum latency for one UAV monitoring the fire would be six minutes. However, when more UAVs are used this latency can be cut in half as well as reducing the latency for information near the base station. In the event that fire fighters can communicate with the UAVs, this would enable timely updates about the perimeter of the fire.

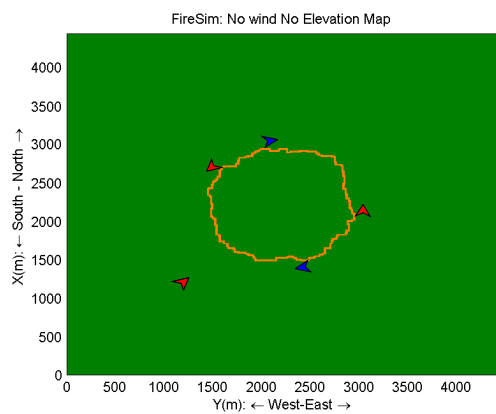
For this simulation six UAVs have been used. The minimum latency for each checkpoint is equivalent to a UAV flying directly from the checkpoint to the base station. The minimum latency of information for checkpoints 1 and 2 is about 45 s and for checkpoints 3 and 4 is about 125 s. The latency for checkpoints 1 and 4 are plotted in Figure 17. The minimum latency for checkpoint 1 is 45 s when information is passed to the base station, and the minimum latency of information from checkpoint 4 is around 125 s.

6 Conclusion

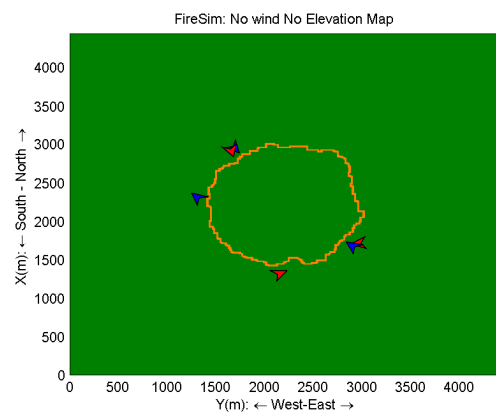
In this paper we have introduced forest fire surveillance as a new cooperative control problem for UAVs. The UAVs are only allowed to communicate when they are within proximity of each other. We have presented an approach for fire surveillance using a single UAV equipped with an infrared camera. We have also introduced a cooperative surveillance scheme that utilizes an even number of UAVs to minimize the information latency and the frequency of update. A simplified version of this algorithm has been analyzed and shown to converge. Simulation results have been presented to demonstrate the effectiveness of the approach.

Acknowledgment

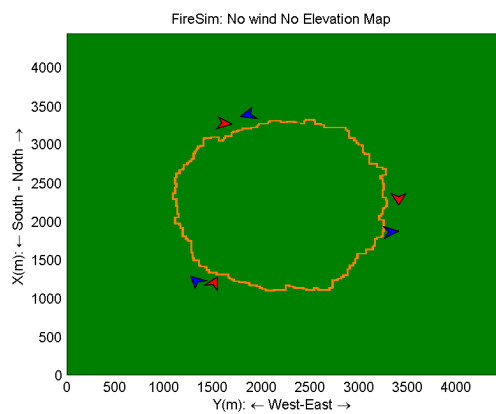
We would like to thank Chad Frost and Francis Enomoto at NASA Ames Research Center for initiating this research problem and pointing out related work in this area. This research was supported by NASA under STTR contract No. NNA04AA19C to Scientific Systems Company, Inc (SSCI) and Brigham Young University (BYU), and by AFOSR grants F49620-01-1-0091 and F49620-02-C-0094.



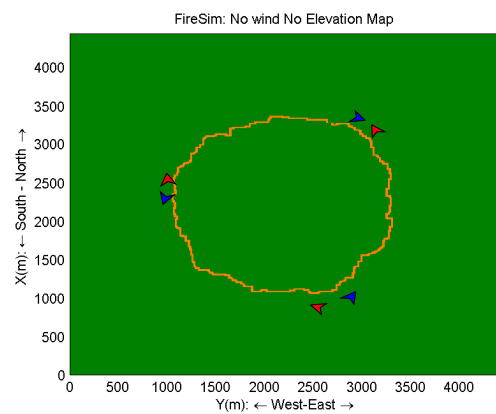
(a)



(b)



(c)



(d)

Figure 12: Six UAVs are shown monitoring a growing fire.

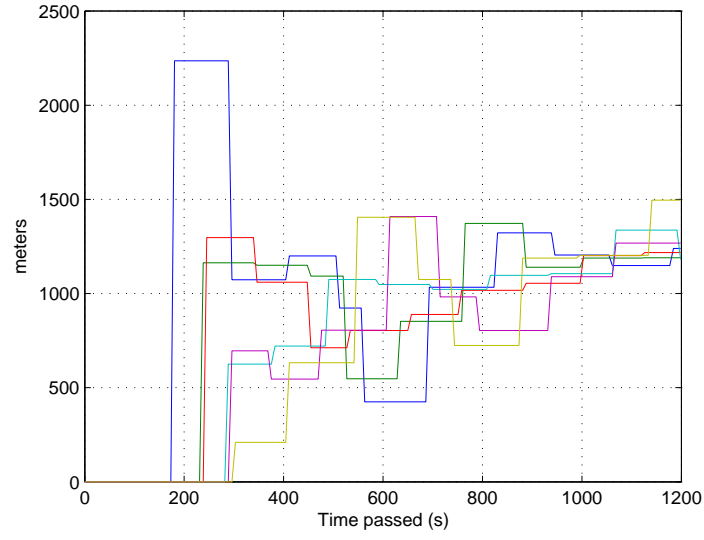


Figure 13: The time histories of $\ell_i(k)$ is for each UAV.

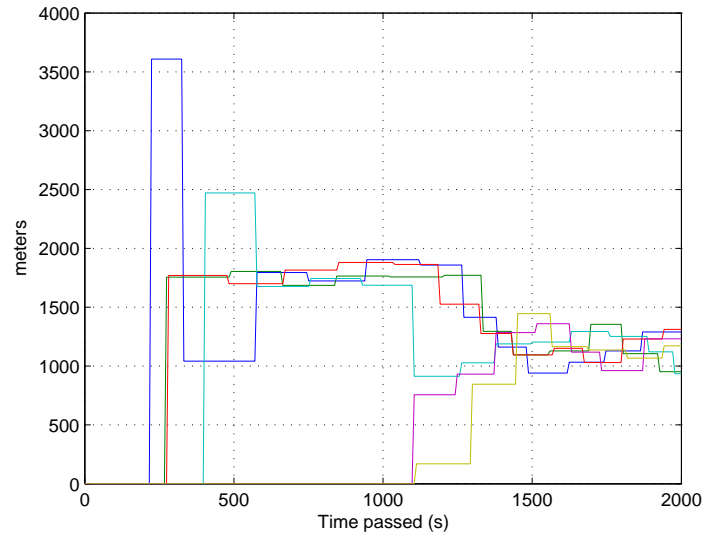


Figure 14: Convergence of $\ell_i(k)$ for four UAVs monitoring the fire with two additional UAVs introduced at $t = 1000$ s.

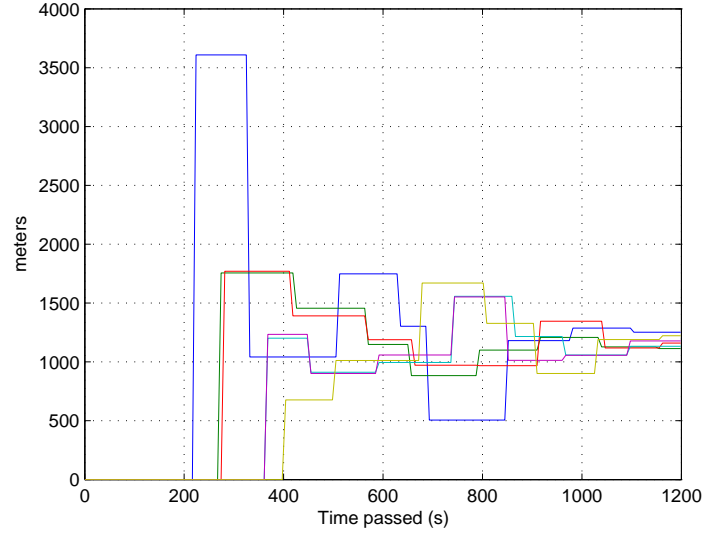


Figure 15: Convergence of $\ell_i(k)$ for six UAVs.

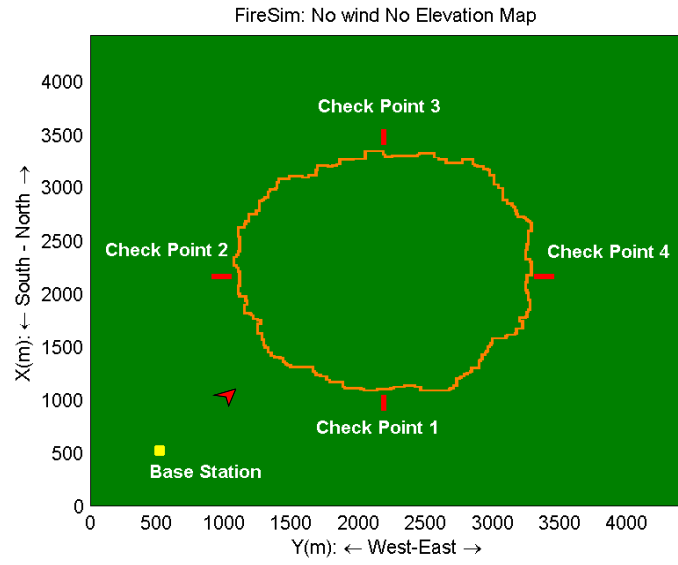


Figure 16: Static fire with checkpoints and base station

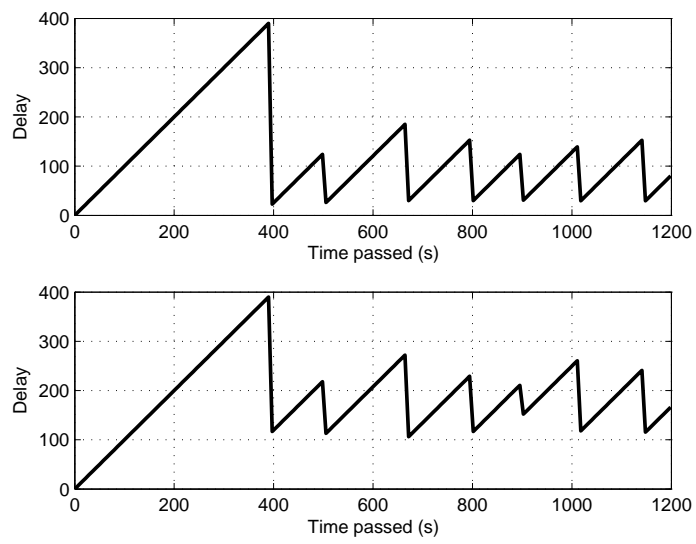


Figure 17: Latency for checkpoints 1 and 4

References

- [1] White paper on UAV over-the-horizon disaster management demonstration projects, Feb 2000. <http://geo.arc.nasa.gov/sge/UAVFiRE/whitepaper.html>.
- [2] V. Ambrosia, S. Wegener, J. Brass, and S. Schoenung. The UAV western states fire mission: Concepts, plans, and developmental advancements. In *Proceedings of the AIAA 3rd Unmanned Unlimited Conference*, September 2004. Paper no. AIAA-2003-6559.
- [3] R. Beard and T. McLain. Multiple UAV cooperative search under collision avoidance and limited range communication constraints. In *Proceedings of the IEEE Conference on Decision and Control*, pages 25–30, December 2003.
- [4] Randal Beard, Derek Kingston, Morgan Quigley, Deryl Snyder, Reed Christiansen, Walt Johnson, Timothy McLain, and Mike Goodrich. Autonomous vehicle technologies for small fixed wing UAVs. *AIAA Journal of Aerospace, Computing, Information, and Communication*, 2004. (to appear).
- [5] Randal W. Beard. Linear operators with application in control and signal processing. *IEEE Control Systems Magazine*, 22(2):69–79, April 2002.
- [6] Randal W. Beard, D.J. Lee, Morgan Quigley, Sarita Thakoor, and Steve Zornetzer. A new approach to observation of descent and landing of future mars mission using bioinspired technology innovations. *AIAA Journal of Aerospace Computing, Information, and Communication*, 2004. (in review).
- [7] J. Bellingham, M. Tillerson, M. Alighanbari, and J. How. Cooperative path planning for multiple UAVs in dynamic and uncertain environments. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2816–2822, December 2002.
- [8] M.E. Campbell, R. D’Andrea, J. Lee, and E. Scholte. Experimental demonstrations of semi-autonomous control. In *Proceedings of the American Control Conference*, pages 5338–5343, July 2004.
- [9] J. Fujiwara, K.; Kudoh. Forest fire detection in 2001 using three-dimensional histogram. In *2002 IEEE International Geoscience and Remote Sensing Symposium, IGARSS ’02*, volume 4, pages 2057 – 2059, June 2002.
- [10] R.H. Gardner, W.W. Hargrove, M.G. Turner, and W.H. Romme. *Climate change, disturbances, and landscape dynamics*, pages 149–172. Global Change and Terrestrial Ecosystems. International Geosphere-Biosphere Programme Book Series - Book #2. Cambridge University Press, Great Britain, 1996.

- [11] R.H. Gardner, W.W. Hargrove, M.G. Turner, and W.H. Romme. *Climate change, disturbances, and landscape dynamics*, pages 149–172. Global Change and Terrestrial Ecosystems. International Geosphere-Biosphere Programme Book Series - Book #2. Cambridge University Press, Great Britain, 1996.
- [12] Rafael C. Gonzales and Richard E. Woods. *Digital Image Processing*. Prentice-Hall, New Jersey, 2nd edition, 2002.
- [13] W.W. Hargrove, R.H. Gardner, M.G. Turner, W.H. Romme, and D.G. Despain. Simulating fire patterns in heterogeneous landscapes. *Ecological Modelling*, 135:243–263, 2000.
- [14] W.W. Hargrove, R.H. Gardner, M.G. Turner, W.H. Romme, and D.G. Despain. Simulating fire patterns in heterogeneous landscapes. *Ecological Modelling*, 135:243–263, 2000.
- [15] G. Inalhan, D. Stipanovic, and C. Tomlin. Decentralized optimization with application to multiple aircraft coordination. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1147–1155, December 2002.
- [16] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, June 2003.
- [17] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.
- [18] Y. Jin, A. Minai, and M. Polycarpou. Cooperative real-time search and task allocation in UAV teams. In *Proceedings of the IEEE Conference on Decision and Control*, pages 7–12, December 2003.
- [19] E. King, Y. Kuwata, M. Alighangari, L. Bertucelli, and J. How. Coordination and control experiments on a multi-vehicle testbed. In *Proceedings of the American Control Conference*, pages 5315–5320, July 2004.
- [20] Derek Kingston, Randal Beard, Timothy McLain, Michael Larsen, and Wei Ren. Autonomous vehicle technologies for small fixed wing UAVs. In *AIAA 2nd Unmanned Unlimited Systems, Technologies, and Operations—Aerospace, Land, and Sea Conference and Workshop & Exhibit*, San Diego, CA, September 2003. Paper no. AIAA-2003-6559.
- [21] J.-I. Kudoh and K. Hosoi. Two dimensional forest fire detection method by using noaa avhrr images. In *2003 IEEE International Geoscience and Remote Sensing Symposium, IGARSS '03*, volume 4, pages 2494 – 2495, July 2003.

- [22] Yi Ma, Stefano Soatto, Jan Kosecka, and Shankar Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer-Verlag, 2003.
- [23] Luc Moreau. Leaderless coordination via bidirectional and unidirectional time-dependent communication. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3070–3075, Maui, Hawaii, December 2003.
- [24] D. Nelson, T. McLain, R. Christiansen, R. Beard, and D. Johansen. Initial experiments in cooperative control of unmanned air vehicles. In *Proceedings of the AIAA 3rd Unmanned Unlimited Conference*, September 2004. AIAA Paper No. 2004-6533.
- [25] Wei Ren and Randal W. Beard. Consensus seeking in multi-agent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, (to appear).
- [26] Wei Ren, Randal W. Beard, and Derek B. Kingston. Kalman consensus strategies and their application to cooperative control. *IEEE Transactions on Automatic Control*, (in review). Technical Report available at <https://dspace.byu.edu/handle/1877/53>.
- [27] Wei Ren, Randal W. Beard, and Timothy W. McLain. Coordination variables and consensus building in multiple vehicle systems. In Vijay Kumar, Naomi Leonard, and A. Stephen Morse, editors, *Cooperative Control*, volume 309 of *Lecture Notes in Control and Information Systems*, pages 171–188. Springer-Verlag, 2004.
- [28] Reza Olfati Saber and Richard M. Murray. Agreement problems in networks with directed graphs and switching topology. In *Proceedings of the IEEE Conference on Decision and Control*, pages 4126–4132, Maui, Hawaii, December 2003.
- [29] B. Seanor, G. Campa, Y. Gu, M. Napolitano, L. Rowe, and M. Perhinschi. Formation flight test results for UAV research aircraft models. In *AIAA 1st Intelligent Systems Technical Conference*, September 2004. AIAA Paper No. 2004-6251.
- [30] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. PWS Publishing at Brooks/Cole Publishing Co., 2nd edition, 1999.
- [31] A. Tiwari, J. Fung, J. Carson, R. Bhattacharya, and R. Murray. A framework for Lyapunov certificates for multi-vehicle rendezvous problems. In *Proceedings of the American Control Conference*, pages 5582–5587, July 2004.

- [32] Y. Yang, A. Minai, and M. Polycarpou. Decentralized cooperative search by networked UAVs in an uncertain environment. In *Proceedings of the American Control Conference*, pages 5558–5563, July 2004.