2005-6

Forest fire monitoring with multiple small UAVs

David W. Casbeer
Brigham Young University - Provo, casbeer@byu.edu

Randal W. Beard
Brigham Young University - Provo, beard@byu.edu

Timothy W. McLain
Brigham Young University - Provo, mclain@byu.edu

Sai-Ming Li
Scientific Systems Company, Inc.

Raman K. Mehra
Scientific Systems Company, Inc.

Original Publication Citation

Casbeer, D., Beard, R., Li, S., McLain, T., and Mehra, R. Forest Fire Monitoring Using Unmanned Air Vehicles, Proceedings of the American Control Conference, pp. 3530-3535, June 2005, Portland, Oregon. DOI: 10.1109/ACC.2005.1470520

BYU ScholarsArchive Citation

Casbeer, David W.; Beard, Randal W.; McLain, Timothy W.; Li, Sai-Ming; and Mehra, Raman K., "Forest fire monitoring with multiple small UAVs" (2005). *Faculty Publications*. 1506.
https://scholarsarchive.byu.edu/facpub/1506

Forest Fire Monitoring Using Multiple Small Unmanned Air Vehicles

David W. Casbeer
Randal W. Beard
Timothy W. McLain
Brigham Young University

Sai-Ming Li
Raman K. Mehra
Scientific Systems Company, Inc.

Abstract— **Frequent and detailed updates of the development of a forest fire are essential for effective and safe fire fighting. Since a forest fire is typically inaccessible by ground vehicles due to mountainous terrain, small Unmanned Air Vehicles (UAVs) are emerging as a promising solution to the problem of monitoring large forest fires. In this paper we present an effective path planning algorithm for a UAV utilizing infrared images that are collected on-board in real-time. In order to demonstrate the effectiveness of our path planning algorithm in realistic scenarios, we implemented the forest fire propagation model EMBYR to simulate the time evolution of a typical forest fire. We also introduce a new cooperative control mission concept where multiple Low-Altitude, Short-Endurance (LASE) UAVs are used for fire monitoring. By simultaneously deploying multiple UAVs, the effectiveness of the mission in terms of information update rate can be improved dramatically.**

I. 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 [1], [2]. Such images are typically taken by low earth orbiting satellites 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 of a fire's development in order to conduct an effective and safe fire fighting mission. Because forest fire monitoring is a difficult task, fire fighters often have 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.

Low-altitude, short-endurance (LASE) UAVs are expected to be a key technology for closing the fire monitoring gap. 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 [3]. 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 the UAV needs to have the intelligence to know when to return to 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.

In this paper we present an effective path planning algorithm for UAVs tasked to monitor a forest fire. In order to demonstrate the effectiveness of our path planning algorithm in realistic scenarios, we implemented the forest fire propagation model EMBYR in Simulink to generate the time-evolution of a typical forest fire. Then we use this model to verify our path planning algorithm in a simulated forest fire scenario. We also introduce a new cooperative control mission concept utilizing multiple LASE UAVs to monitor the perimeter of the fire. By deploying multiple UAVs at the same time, the effectiveness of the mission in terms of the information update rate can improve dramatically. On the other hand, new problems such as determining how to coordinate the paths of the UAVs to cover the most critical spots, when and which UAV should be taken down for refueling, and how to measure the performance of the entire fleet of UAVs, must be addressed.

The paper is organized as follows: The fire monitoring problem is described in Section II, followed by a description the fire simulation model in Section III. Section IV describes the fire perimeter tracking algorithm used, with simulation results. Section V describes our cooperative control approach using a simplified scenario where the fire is assumed to be of circular shape. Finally, Section VI concludes the paper with comments on future work.

II. 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 a UAV to monitor the propagation of the fire. The objective for the UAV is to image the perimeter of the fire, and to upload the location of the fire perimeter (with associated imagery) to the base station as frequently as possible.

The fire will spread in an unknown fashion, which means the UAV cannot follow a pre-planned path. The UAV is assumed to be equipped with an infrared camera that captures images of a small region beneath it at regular

Fig. 1. Fire monitoring scenario.

sampling instants, indicated by the blue rectangle in Figure 1. The UAV can use the infrared images collected for navigational purposes. The UAV is also assumed to have limited communication range, which means it cannot upload data to the base station unless it is within range of the station. Finally, the UAV is assumed to have limited flight range, which implies that it must return to the base station for refueling after flying for a certain amount of time.

To compare various fire monitoring algorithms, two simple performance metrics are used. Two perpendicular lines crossing at the geometric center of the fire define four points (C1, C2, C3, C4) around the periphery of the fire. The first metric is the update frequency from each checkpoint. The second metric is the time delay required to relay information from each checkpoint to the ground station.

III. FOREST FIRE PROPAGATION MODEL

According to [4] the three main types of models used in computer simulation of forest fires are bond percolation, cellular automata, and elliptical wave propagation.

Bond percolation is a probabilistic technique that divides the terrain into a grid with each element of the grid having defined properties such as moisture, elevation, and fuel type. The fire will spread from one cell to the next with a probability determined by these properties and the direction and speed of the wind [4]. An instantiation of this technique is the EMBYR (Ecological Model for Burning in the Yellowstone Region) model [5], [6].

Similar to bond percolation, the cellular automata technique divides the region into a grid with each cell having certain properties. But instead of spreading according to a probability, the fire will spread according to the cell's initial state, the state of neighboring cells, and the predefined properties of the cell [4]. Two instantiations of this technique are the DYNAFIRE [7] and FIREMAP [8] models.

The last method makes use of a continuous terrain and is referred to as elliptical wave propagation. This method

divides the perimeter of the fire into equally spaced discrete points [4]. According to Huygen's principle, the fire will spread elliptically away from each point [9]. Both the WILDFIRE [10] and FARSITE [11] models use this technique.

The EMBYR model [5], [6] was chosen for this study because it is straightforward to implement and it produces simulations that closely resemble the propagation of actual forest fires.

A. The EMBYR Model

The EMBYR technique divides the landscape into a lattice of cells, in which single or multiple fires may be ignited. The fire propagates according to independent stochastic events at every time step. The fire can spread in a time step by two methods: diffusion from one cell to an adjacent cell or by firebrands being thrown to distant cells. The probability of spread as well as the rate of occurrence for firebrands are a function of the strength and direction of the wind and the combustibility of fuel types due to fuel moisture content.

1) Diffusive spread: The fire will spread from one cell to any of eight adjacent unburned cells with probability I [5], [6], $(0 \le I \le 1)$ in a given time step. It was found in [5] that for $I \le 0.6$ the probability of diffusion does not need to be adjusted for diagonal neighbors because of the increased distance.

2) Wind and Slope Effects: Both the effects of wind [5] and elevation [12] on the probability of diffusive spread I are modeled using a binomial distribution. The parameters b_{wj} or b_{sj} represent the bias value for wind and slope (elevation change) respectively. The bias value is determined by the direction of either wind or slope. The new probability of fire spread is found by $i_j = 1 - (1 - I)^{b_{xj}}$, where the subscript x is used to indicate either wind or slope. The bias value b_{wj} is found for wind as shown in Figure 2. The bias value is adjusted in the direction of wind as well as the two adjacent directions.
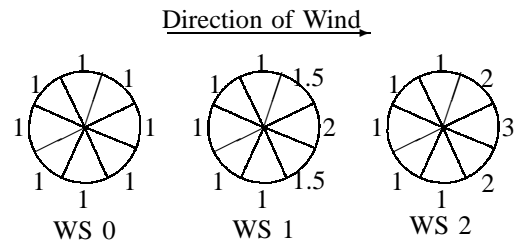

Fig. 2. With the direction of the wind indicated at the top of the figure, the wind bias is determined by the relative weight indicated in the figure where WS0 represents zero wind and WS2 represents strong wind conditions.

The bias due to elevation change is found by $b_{sj} = 2^{(s/s_{\max})}$. Where s is the slope from the burning cell to the neighboring cell, and $s_{\max}$ is the value of s which the fire diffusion probability is a maximum. According to Hargrove [12] a good estimate is $s_{\max} = 1$ or a slope of 45 degrees.

3) Firebrands: Firebrands are carried by updrafts due to the heat of the fire. The number of firebrands produced by a cell depends on its fuel moisture class and fuel type. The distance traveled by the firebrands increases with wind, and the likelihood of the targeted cell igniting depends on its fuel type and moisture class.

(a) t = 2800sec   (b) t = 6800sec   (c) t = 10800sec

Fig. 4. Fire simulation of high wind conditions with an elevation gradient. The The fire spreads in the direction of the wind.

B. Simulated Fire Snapshots

The original EMBYR code is written in FORTRAN and C and is available online [13]. The essential parts of the model have been rewritten in MATLAB for this study. This code was run 100 times and averaged to achieve the four simulations shown. The landscapes are 400 cells wide each with constant a moisture class and each cell's fuel type was generated randomly. In Figure 3, the black regions represent areas that have already burned, the yellow and orange colored areas are currently on fire, and the outside green colored regions are unburned.

We define the maximum speed of the UAVs to be $V_{\max} \triangleq$ 30 mph = 13.4 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. To define the length of a time step (T_{step}), we let the maximum speed of the fire be V_{fmax} = 10 mph = 4.47 m/sec. Because the fastest moving fire advances at about 80 cells per time step each time step is found to be $T_{\text{step}} \approx 200$ sec.

Figure 3 represents a fire on flat ground with no wind with the fire starting in the center of the image. In theory it would burn in the shape of a pure circle if there was a uniform fuel type. It is burning outward from the original ignition point at a rate of about 0.12 m/sec or 0.56 km/hr.

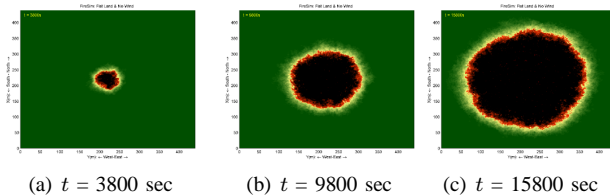(a) t = 3800 sec   (b) t = 9800 sec   (c) t = 15800 sec

Fig. 3. Fire simulation on flat land with no wind. The black pixels represent burned land, the orange and yellow pixels are areas currently on fire, green represents unburned area.

In high wind situations the fire spread takes an elliptical pattern as shown in Figure 4 with the fastest front moving at about 10 m/sec. Here the fire was ignited from the bottom left.
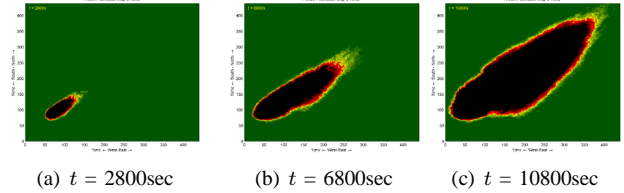
These simulations are realistic enough to allow the study of fire perimeter tracking schemes using single and multiple UAVs.

IV. PATH PLANNING ALGORITHM

In this section, we consider the fire perimeter tracking problem for one UAV. We assume each UAV is equipped with an infrared camera and the on-board processing ability to detect a fire edge in the image using either gradient operators [14] or a thresholding method [15], [16] on the most intense part of the fire. 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.

There are four coordinate systems used in this paper. A vector's coordinate frame can be discerned from its subscript as follows: subscripts i, v, b, and c denote respectively the inertial, vehicle, body, and camera frames. If there is a p present in the subscript this is to denote that this point is the location of the UAV in the respective coordinate frame.

The following control algorithm relies on the UAV's current position, heading, and pixel coordinates belonging to the fire edge. It is also assumed that the image processing algorithm identifies which side of the edge is burning. In the camera frame we define the x-axis to be forward looking, with the y-axis out the right wing, and z-axis pointing out the belly of the UAV with the origin at the center of the image.

The camera has a view angle of 30 deg and is forward looking in the xz-plane with respect to the z-axis by θ_c degrees and produces a picture of size $i_p \times i_p$ pixels. Here ψ is the heading of the UAV, which is flying at a constant altitude of h in meters by means of an autopilot [17]. The camera is kept at this angle with the use of pan and tilt gimbal control [18], [19].

Given these definitions the UAV will be be able to view the ground (assuming flat land) D_f meters forward and D_r meters in reverse, where

$$D_f = h\tan(15 \text{ deg} - \theta_c) \qquad (1)$$
$$D_r = h\tan(15 \text{ deg} + \theta_c) . \qquad (2)$$

Letting the center of the image be the origin, the projected location of the UAV in the image is determined to be

$$\mathbf{x}_{pc} = \begin{bmatrix} x_{pc} \\ y_{pc} \end{bmatrix} = \begin{bmatrix} -\frac{i_p}{2} + \frac{D_r}{k} \\ 0 \end{bmatrix} , \qquad (3)$$

where $k = \frac{D_r + D_r}{i_p}$ is a scaling factor with units of m/pixel.

We can transform a point $\mathbf{x}_c$ in the camera frame to $\mathbf{x}_i$ in the inertial frame by,

$$\mathbf{x}_i = \mathbf{R}_{b \to v} k (\mathbf{x}_c - \mathbf{x}_{pc}) + \mathbf{x}_{pi} , \qquad (4)$$

where $\mathbf{R}_{b \to v}$ is a rotation matrix given by,

$$\mathbf{R}_{b \to v} = \mathbf{R}_{v \to b}^{-1} = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix}^{-1} \qquad (5)$$

and $\mathbf{x}_{pi}$ is the UAV's position in the inertial coordinate frame.

We next describe the path planning algorithm. We assume that image processing stage passes along an image of the fire's edge points along with a label telling us which side of the perimeter is on fire. It is important that we have this second piece of information since folds in the perimeter and sharp edges would cause problems.

The first step is to approximate the fire edge by a straight line. This will help smooth the path of the UAV if there are lakes, rivers, or boulder fields that cause gaps in the fire edge as well as help with noise in the image. We note as a reminder that in the camera frame the y axis is aligned horizontally and the x axis is vertical. Thus the parameters for the straight line $x_c = a y_c + b$ are determined by the Least Square estimate:

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

where the first column of the matrix $\mathbf{Y}_c$ consists of the y coordinates of the perimeter pixels and the second column is ones. $\mathbf{X}_c$ is a vector of x coordinates of fire edge pixels. If the matrix $(\mathbf{Y}_c^\top \mathbf{Y}_c)^{-1}$ 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 essence of our path planning algorithm is to place waypoints along the desired path based on the approximated fire perimeter, and let the autopilot determine the control required to reach the waypoints. At every time instant, a waypoint is placed ahead of the UAV at a fixed distance from the straight line that approximates the fire perimeter. More precisely, let direction parameter $d_p = -1$ if the UAV is to follow a counterclockwise path, and $d_p = 1$ if the path is clockwise. Let $\theta = \tan^{-1}(|a|)$, which is the absolute angle of inclination of the approximated fire perimeter. If $\theta \le 45$ deg we will refer to the fire as being above or below the line. Similarly, for $\theta > 45$ deg, the fire will be referred to as being to the left or to the right (in the camera frame) of the line. If the fire is below or to the right of the line we will assign direction parameter $d_f = 1$, otherwise we will assign $d_f = -1$. The variables d_f and d_p are used to determine the direction to move the next waypoint.

The next waypoint is determined according to theta θ. This waypoint is then converted from the camera frame to the inertial frame so the autopilot can determine a good

path to the waypoint. When $\theta \le 45°$ the next waypoint $[x_{dc}, y_{dc}]^\top$ is:

$$\begin{bmatrix} x_{dc} \\ y_{dc} \end{bmatrix} = \begin{bmatrix} a y_{dc} + b + d_f d_x \\ y_{pc} + d_f d_p d_y \end{bmatrix} \qquad (7)$$

where d_y is measured as the distance along the y axis from the current plane's location and the next waypoint (*i.e.*, $d_y = |y_{pc} - y_{dc}|$). d_x is the distance from the approximated fire perimeter to the next waypoint measured along the x axis. Visually the effects of d_f, d_p, d_y, and d_x can be seen in Figure 5. Here the plane is flying counterclockwise thus $d_p = -1$ and since the fire is above the perimeter approximation $d_f = -1$. From equation 7 we see the next waypoint will move in the positive y direction from the plane's location since $d_f d_p = 1$ and the waypoint will be located below the fire perimeter approximation because $d_f = -1$. In simulations, we have used $d_y = 9$ pixels and $d_x = 13.5$ pixels.
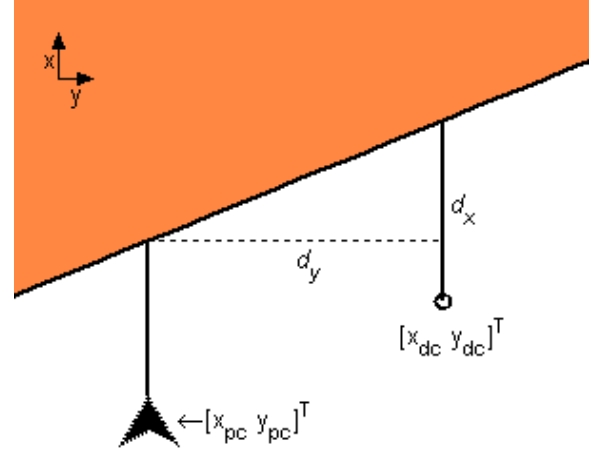

Fig. 5. $\theta \le 45°$. UAV is flying counterclockwise, with the fire above the fire edge. Next waypoint is shown as $[x_{dc}, y_{dc}]^\top$.

When $\theta > 45°$ the next waypoint is:

$$\begin{bmatrix} x_{dc} \\ y_{dc} \end{bmatrix} = \begin{bmatrix} x_{pc} + d_f d_p d_x \\ (x_{dc} - b)/a - d_f d_y \end{bmatrix} . \qquad (8)$$

In Figure 6 the plane is flying clockwise yeilding $d_p = 1$. The fire is found on the right of the fire edge approximation making $d_f = 1$. The next waypoint moves a positive distance of d_x along the x axis since $d_f = d_p = 1$ in equation 8 and because $d_f = 1$ is subtracted, the waypoint is moved to the left of the fire edge by d_y. For simulations, $d_x = 9$ pixels and $d_y = 13.5$ pixels.

V. FIRE MONITORING USING MULTIPLE-UAVS

The problem addressed in this section will be the development of a centralized scheme in which a number of UAV's cooperate to pass information concerning a burning fire to a base station, as frequently as possible, with minimum delay. A brief discussion of a decentralized solution will then be given. The UAVs are assumed to have a communication
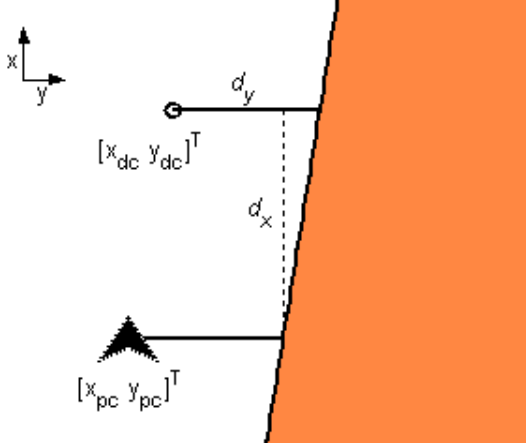
Fig. 6. $|a| > 45°$. UAV is flying clockwise, with the fire burning to the right the fire edge. Next waypoint is shown.

radius of 3 to 5 km which could be significantly reduced in mountainous regions where these UAVs will be used. The region enveloped by the forest fire is much larger, in the neighborhood of 1000 acres or greater, which is equivalent to a circle with a circumference of at least 7 km.

Before proceeding, a few definitions are needed. To simplify the problem for study, the fire is assumed to be a spreading circle with radius R. We will be using the fire center as the origin of an inertial coordinate frame (i.e., X is north, Y is east, and Z points into the ground). There is a single base station located at $p_b = [x_b \, y_b]^\top$ where the data needs to be delivered. There are an even number of UAVs, N of which are labeled with odd numbers and the remaining N UAVs are labeled with even numbers. They are traveling at a constant velocity V_m and are assumed to have the ability to follow the edge of the fire using a vision guidance system.

Since the fire perimeter is moving, it is important that the UAVs can track the edge of the fire autonomously. This allows the explicit path along the fire perimeter to be eliminated from the coordination effort. For example, rendezvous points on the fire edge for the exchange of data do not need to be specified, the rendezvous angles (θ_i, $i = 1, \ldots, 2N$) with respect to the center of the fire can be used. In this paper, we will measure θ_i counterclockwise with respect to east, where the subscript i has been used to distinguish between different rendezvous angles for other sets of UAVs. The UAVs will then follow the fire perimeter until they reach θ_i.

Due to the communication constraint it is evident that the UAVs must either individually deliver the data collected to the base station or pass the data to another UAV that subsequently transmits this information to the base station. The data is considered delivered when the UAV passes the line segment connecting the origin and the base station or when $\theta_{pi} = \theta_b$ where θ_b is the angle of the base station and θ_{pi} is the angle of UAV i both measured counterclockwise with respect to East.

To measure the performance of the fire monitoring algorithm, we assign four checkpoints at equidistant points along the perimeter of the fire (c1 (East) to c4 (South) in Figure 7). The base station will be located at c1 or $\theta_b = 0$. Information about each checkpoint is acquired when a UAV passes over this location and information is exchanged when two UAVs are at the same location on the fire perimeter. Two metrics will be monitored: the update frequency for each checkpoint and the maximum delay to relay information to the groundstation for any of the checkpoints. Three cases will be analyzed: single UAV, two UAVs, and greater than two UAVs.
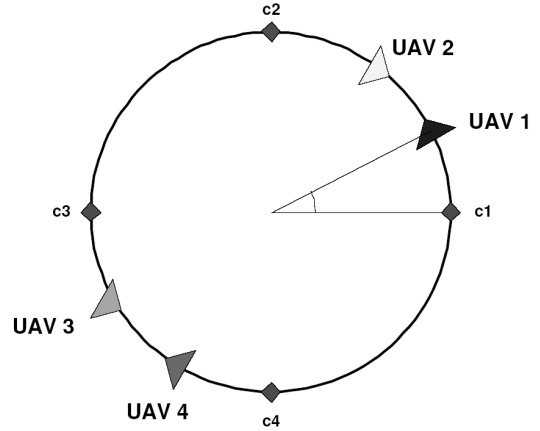

Fig. 7. c1 is East while c2 is North

A. Case 1: Single UAV

This is the trivial case where the best way to monitor the fire is to follow the perimeter in a single direction. It will take

$$T = \frac{2\pi R}{V_m} \qquad (9)$$

seconds to cover the entire fire perimeter once. Hence the update frequency for all checkpoints at the base station is $1/T$ Hz.

For all checkpoints, the maximum delay of information for the UAV is equal to T, as the UAV reaches each of them only once each cycle. The maximum delay of information for the base station, which is located at checkpoint 1, is also equal to $7T/4$ for c2 (assuming the UAV travels in the counter-clockwise direction).

B. Case 2: Two UAVs

In this case, the optimal strategy is to have the two UAVs start from opposite ends of the fire, and travel in opposite directions. When they meet, they can either continue in the same direction, or synchronize their data and reverse direction. Both strategies yield the same performance in this case. However, in the next case where $N > 1$, the latter strategy proves more efficient. Hence we will consider this strategy here as well.

Fig. 8 shows the angular positions of the two UAVs versus time. In this simulation we choose V_m such that $T = 12$ sec. Since the UAVs turn around whenever they meet, each UAV only covers half of the circle (UAV 1 covers $(-\frac{\pi}{2}, \frac{\pi}{2})$, while UAV 2 covers $(\frac{\pi}{2}, \frac{3\pi}{2})$). We will label the two rendezvous angles by $p_{1rv} = \frac{\pi}{2}$ and $p_{2rv} = \frac{3\pi}{2}$. The rendezvous times at these two points are,

$$
\begin{aligned}
t_{1rv} &= (2k+1)T + t_o \\
t_{2rv} &= 2kT + t_o, \qquad k = 0, 1, 2, \ldots
\end{aligned}
$$

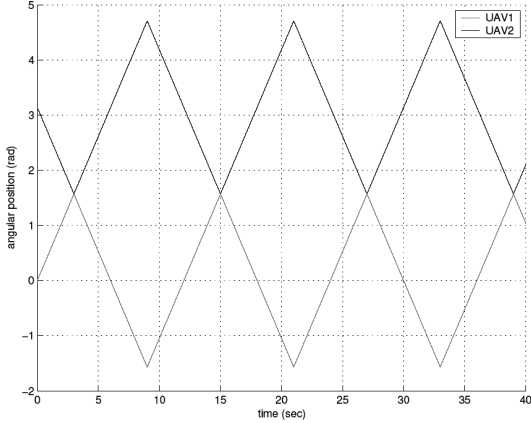where t_o, the initial rendezvous time is set to 3 sec.
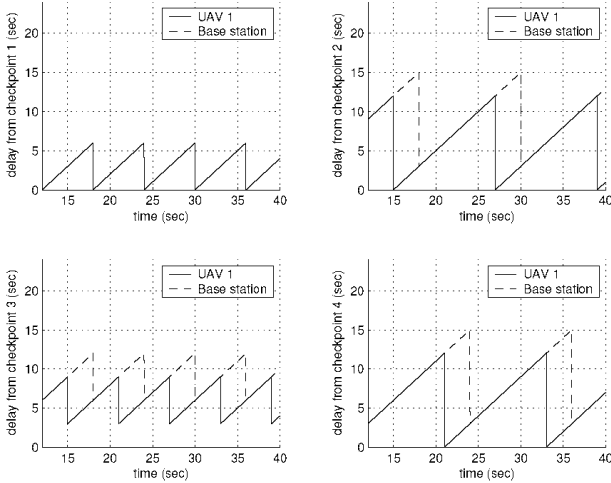

Fig. 8. Two UAV angular position

Fig. 9. Two UAV data delay from checkpoints: top left (c1), top right (c2), bottom left(c3), bottom right (c4). The solid line is the amount of time since UAV 1 has been updated with data from that checkpoint, while the dashed line is the amount of time delay since the base station has been updated.

Fig. 9 shows the amount of time passed since the base station and UAV 1 were last updated with the data from the corresponding checkpoint. Notice that except for checkpoint 1, which coincides with the base station, the data's

delay at the base station for all the checkpoints is always greater than or equal to the delay for UAV 1. This is because UAV 1 obtains updates of the other checkpoints through UAV 2 from time to time. However, until UAV 1 returns to c1, the base station will not have any updates.

As can be seen, the maximum delay (15 sec) of data for the base station occurs from checkpoints 2 and 4. Ironically, the maximum delay for checkpoint 3, which is the farthest from the base station, is smaller (12 sec). The update frequencies for data on checkpoints 1 and 3 has doubled to $2/T$ Hz compared to the single UAV case, while those for checkpoints 2 and 4 remain at $1/T$ Hz.

C. Case 3: $2N$ UAVs, $N > 1$

In this case, the optimal strategy is to start at equidistant points around the perimeter of the fire. Alternate UAVs are assigned to two groups (odd or even). The two groups travel in opposite directions until meeting at rendezvous angles $p_{irv} = \frac{(2i+1)\pi}{2N}$, $i = 0, \ldots, 2N-1$. When UAVs meet, they turn around and travel in the opposite direction. Notice that odd subscripted rendezvous ideally occur simultaneously, as well as the even:

$$
\begin{aligned}
t_{i_o rv} &= (2k+1)T + t_o \\
t_{i_e rv} &= 2kT + t_o, \qquad k = 0, 1, 2, \ldots
\end{aligned}
$$

where i_o indicates odd values in the range $i = [0, \ldots, 2N-1]$ and i_e indicates even values in the same range. In this example $t_o = 1.5$ sec.

Fig. 10 shows the angular position of the UAVs against time when four UAVs are used to track the fire perimeter. The trajectories are similar to those in the two UAV case except the portion of fire covered by each UAV is halved. Figure 11 shows the delay of data from each checkpoint on both UAV 1 and the base station. The frequencies of update for checkpoints 1 and 3 are both $4/T$ Hz, while those for checkpoints 2 and 4 are $2/T$ Hz. The maximum delay for checkpoint 3 is 9 sec, or $3T/4$ sec at the base station, which is a significant improvement from the two-UAV case. Maximum delays for other checkpoints are also shorter in this case.

Figure 12 shows the delay of data from each checkpoint for both UAV 1 and the base station when eight UAVs are deployed. The frequencies of update for checkpoints 1 and 3 are both $8/T$ Hz, while those for checkpoints 2 and 4 are $4/T$ Hz. The maximum delay for checkpoint 3 is 7.5 sec, or $5T/8$ sec at the base station, which is a significant improvement over the two-UAV case.

In general, when $2N$ UAVs are deployed, each UAV only needs to cover an angle of $2\pi/(2N)$ along the perimeter. For checkpoint 1, which coincides with the base station and lies in the midpoint of the path of UAV 1, the time between visits is $T/2N$. Hence the update frequency of checkpoint 1 is $2N/T$ Hz. For checkpoints 2 and 4, we can expect the update frequencies to be half of that for checkpoint 1 at N/T Hz, as UAV 1 gets updates from those checkpoints only once instead of twice per cycle. Finally,

for checkpoint 3, since it lies directly across checkpoint 1 and UAVs from both sides of the fire can communicate its updates to UAV 1, its update frequency is twice that of checkpoints 2 and 4, at $2N/T$ Hz.

Under the scheme described above, the data at each checkpoint can be relayed to the base station in minimum time, *i.e.,* the time it takes for a UAV to travel from the checkpoint to the base station. The maximum delay for each checkpoint at the base station consist of two components: (i) the time it takes for the data to travel between the checkpoint and the base station; (ii) the time between updates for that checkpoint at the base station. The first component for checkpoints 2 and 4 are $T/4$, while that for checkpoint 1 is $T/2$. The second component for checkpoints 2 and 4 are $2 \times T/(2N) = T/N$, as it takes UAV 1 two cycles to update those checkpoints at the base station. For checkpoint 3, the corresponding time is $T/(2N)$. In summary, the maximum delays for checkpoints 2 and 4 are

$$\delta_{2,4} = \frac{T}{4} + \frac{T}{N} \qquad (10)$$

and that for checkpoint 3 is

$$\delta_3 = \frac{T}{2} + \frac{T}{2N} . \qquad (11)$$

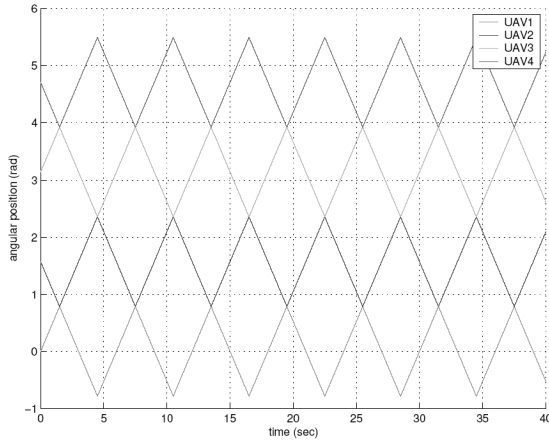This can be verified from Figures 9 through 12.

Fig. 10.   Four UAV angular position

Notice, if R, T, and t_o are known, the centralized solution for UAV cooperative fire monitoring in the steady state is known. Real fires will tend not to be circular due do wind, elevation changes, and fuel availability as was shown in section III. Also, due to the concave nature of a forest fire there might be multiple locations on the fire edge that yield the same angle for a given fire center. Assuming the UAVs can calculate distance traveled, we can measure the distance (L) from the base station along the perimeter as our measure instead of the angle. Here we would use $L = 0, \ldots, P$ (where P is the perimeter length) rather than $\theta = 0, \ldots, 2\pi$. This allows the explicit path to be decoupled from the coordination effort.
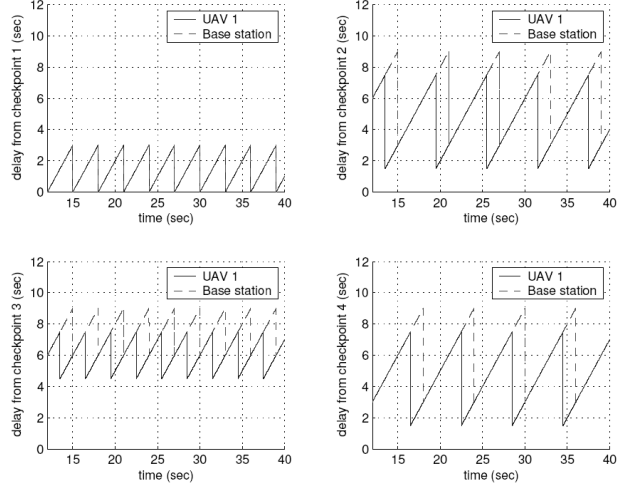

Fig. 11.   Four UAV data delay from checkpoints: top left (c1), top right (c2), bottom left(c3), bottom right (c4). The solid line is the amount of time since UAV 1 has been updated with data from that checkpoint, while the dashed line is the amount of time delay since the base station has been updated.
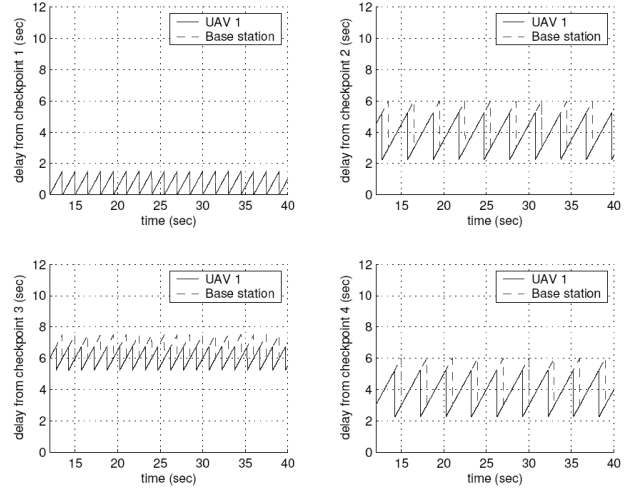

Fig. 12.   Eight UAV data delay from checkpoints: top left (c1), top right (c2), bottom left(c3), bottom right (c4). The solid line is the amount of time since UAV 1 has been updated with data from that checkpoint, while the dashed line is the amount of time delay since the base station has been updated.

Once a pair of UAVs has met after traveling around the circumference of the fire, the perimeter length P and time of travel T (assuming constant velocity) will be known. These values will then be constantly updated to each UAV (with delays for each UAV similar to the examples above). With the UAV locations specified (under the condition that P is updated frequently enough for each plane), the initial rendezvous time for the UAVs, t_o, could then be determined using a cooperative timing [20] or consensus based approach [21].

VI. Conclusions and Future Work

A vision based fire perimeter tracking algorithm has been shown to work effectively on simulated forest fires. This allows a UAV to follow the edge of a forest fire autonomously to obtain images and data concerning the spreading fire. It can be seen that by incorporating multiple UAVs, updates to fire fighters concerning the location and growth of a forest fire can be made more frequently and with greater detail.

While the approach presented has promise, numerous technical issues remain to be resolved including determination of the initial rendezvous time, dealing with fuel contingencies and refueling, implementation with irregular and growing fire shapes, and determining factors that allow the perimeter length to be updated frequently enough.

References

[1] 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.

[2] 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.

[3] White paper on uav over-the-horizon disaster management demonstration projects, Feb 2000. http://geo.arc.nasa.gov/sge/UAVFiRE/whitepaper.html.

[4] D. Albright and B.N. Meisner. Classification of fire simulation systems. *Fire Managment Notes*, 59(2):5–12, Spring 1999.

[5] 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.

[6] 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.

[7] K.D. Kalabokidis, C.M. Hay, and Y.A. Hussin. Spatially resolved fire growth simulation. In *Proceedings of the 11th Conference on Fire and Forest Meteorology*, pages 188–195. Society of American Foresters, Bethesda, Maryland, April 16-19 1991.

[8] G.L. Ball and D.P. Guertin. Improved fire growth modeling. *International Journal of Wildland Fire*, 2(2):47–54, 1992.

[9] D.G. Anderson, E.A. Catchpole, N.J. de Mestre, and T. Parkes. Modeling the spread of grass fires. *Journal of Australian Mathematical Society (series B)*, 23:451–466, 1982.

[10] G. Wallace. A numerical fire simulation model. international journal of wildland fire. *International Journal of Wildland Fire*, 3(2):111–116, 1993.

[11] M.A Finney. Modeling the spread and behavior of prescribed natural fires. In *In proceedings of the 12th Conference on Fire and Forest Meteorology*, volume 59. Jekyll Island, Georgia, October 26-28 1993.

[12] W.W. Hargrove. Personal communication, 2004.

[13] http://research.esd.ornl.gov/hnw/embyr/.

[14] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.

[15] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. PWS Publishing at Brooks/Cole Publishing Co., 2nd edition, 1999.

[16] Rafael C. Gonzales and Richard E. Woods. *Digital Image Processing*. Prentice-Hall, New Jersey, 2nd edition, 2002.

[17] 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.

[18] Randal W. Beard, D.J. Lee, 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).

[19] Morgan Quigley, Michael A. Goodrich, Steve Griffiths, Andrew Eldredge, and Randal W. Beard. Target acquisition, localization, and surveillance using a fixed-wing mini-uav and gimbaled camera. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005. (in review).

[20] Timothy W. McLain and Randal W. Beard. Coordination variables, coordination functions, and cooperative timing missions. *AIAA Journal of Guidance, Control, and Dynamics*, 2004. (to appear).

[21] Derek B. Kingston, Wei Ren, and Randal W. Beard. Cooperative timing using Kalman consensus. In *Proc. of the American Control Conference*, 2005. (Submitted for review.).